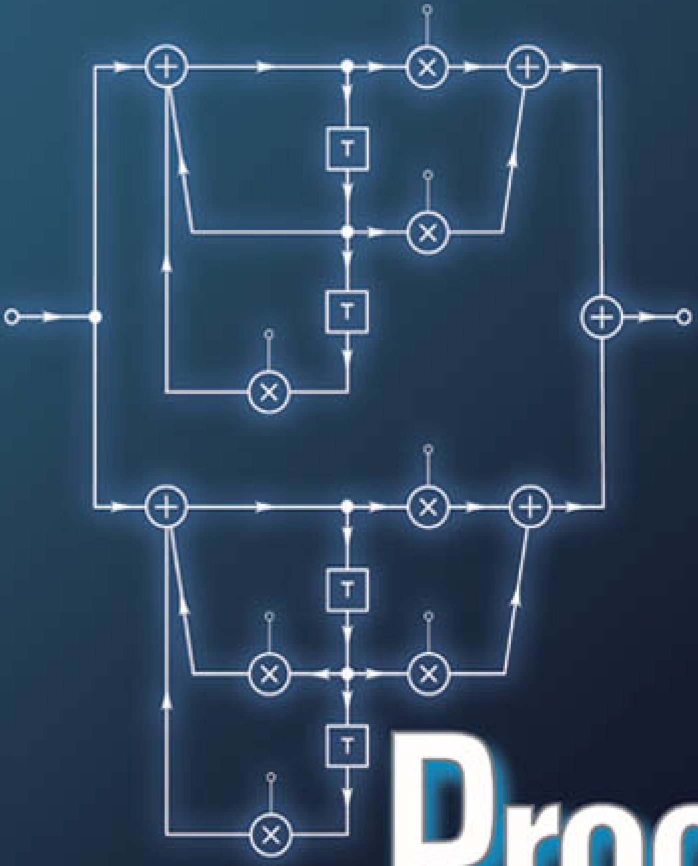


Digital Signal Processing

SIGNALS, SYSTEMS, AND FILTERS

Andreas Antoniou

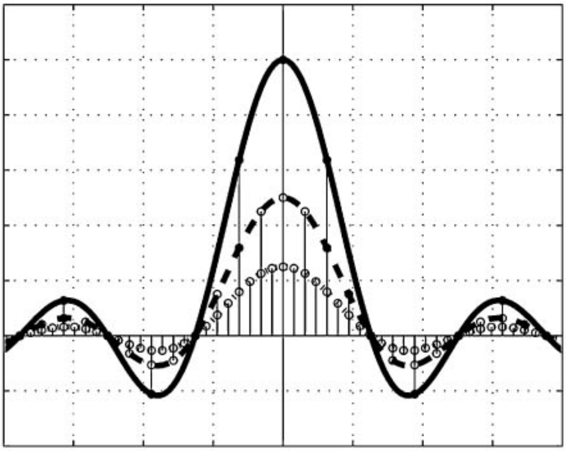
Includes Free License to the DSP Software Package D-Filter

Digital Signal Processing

This page intentionally left blank

Digital Signal Processing

SIGNALS SYSTEMS AND FILTERS



Andreas Antoniou

University of Victoria

British Columbia

Canada

McGraw-Hill

New York Chicago San Francisco Lisbon London
Madrid Mexico City Milan New Delhi San Juan
Seoul Singapore Sydney Toronto

The McGraw-Hill Companies

0-07-158904-X

The material in this eBook also appears in the print version of this title: 0-07-145424-1.

McGraw-Hill eBooks are available at special quantity discounts to use as premiums and sales promotions, or for use incorporate training programs. For more information, please contact George Hoare, Special Sales, at george_hoare@mcgraw-hill.com or (212) 904-4069.

DOI: 10.1036/0071454241

McGraw Hill **Professional**

Want to learn more?

We hope you enjoy this McGraw-Hill eBook! If you'd like more information about this book, its author, or related books and websites, please click here.

In memory of
my wife Rosemary
my mother Eleni
and my father Antonios

This page intentionally left blank

ABOUT THE AUTHOR

Andreas Antoniou received the B.Sc. (Eng.) and Ph.D. degrees in Electrical Engineering from the University of London, U.K., in 1963 and 1966, respectively, and is a Fellow of the Institution of Electrical Engineers and the Institute of Electrical and Electronics Engineers. He taught at Concordia University from 1970 to 1983 serving as Chair of the Department of Electrical and Computer Engineering during 1977–83. He served as the founding Chair of the Department of Electrical and Computer Engineering, University of Victoria, B.C., Canada, from 1983 to 1990, and is now Professor Emeritus in the same department. His teaching and research interests are in the areas of circuits and systems and digital signal processing. He is the author of *Digital Filters: Analysis, Design, and Applications* (McGraw-Hill), first and second editions, published in 1978 and 1993, respectively, and the co-author with W.-S Lu of *Two-Dimensional Digital Filters* (Marcel Dekker, 1992).

Dr. Antoniou served as Associate Editor and Chief Editor for the *IEEE Transactions on Circuits and Systems* (CAS) during 1983–85 and 1985–87, respectively; as a Distinguished Lecturer of the IEEE Signal Processing Society in 2003; and as the General Chair of the *2004 IEEE International Symposium on Circuits and Systems*.

He received the Ambrose Fleming Premium for 1964 from the IEE (best paper award), a CAS Golden Jubilee Medal from the IEEE Circuits and Systems Society in 2000, the B.C. Science Council Chairman's Award for Career Achievement for 2000, the Doctor Honoris Causa degree from the Metsovio National Technical University of Athens, Greece, in 2002, and the IEEE Circuits and Systems Society Technical Achievements Award for 2005.

This page intentionally left blank

TABLE OF CONTENTS

PREFACE

The great advancements in the design of microchips, digital systems, and computer hardware over the past 40 years have given birth to *digital signal processing* (DSP) which has grown over the years into a ubiquitous, multifaceted, and indispensable subject of study. As such DSP has been applied in most disciplines ranging from engineering to economics and from astronomy to molecular biology. Consequently, it would take a multivolume encyclopedia to cover all the facets, aspects, and ramifications of DSP, and such a treatise would require many authors. This textbook focuses instead on the fundamentals of DSP, namely, on the representation of signals by mathematical models and on the processing of signals by discrete-time systems. Various types of processing are possible for signals but the processing of interest in this volume is almost always linear and it typically involves reshaping, transforming, or manipulating the frequency spectrum of the signal of interest. Discrete-time systems that can reshape, transform, or manipulate the spectrum of a signal are known as *digital filters*, and these systems will receive very special attention as they did in the author's previous textbook *Digital Filters: Analysis, Design, and Applications*, McGraw-Hill, 1993.

This author considers the processing of continuous- and discrete-time signals to be different facets of one and the same subject of study without a clear demarcation where the processing of continuous-time signals by analog systems ends and the processing of discrete-time signals by digital systems begins. Discrete-time signals sometimes exist as distinct entities that are not derived from or related to corresponding continuous-time signals. The processing of such a signal would result in a transformed discrete-time signal, which would be, presumably, an enhanced or in some way more desirable version of the original signal. Obviously, reference to an underlying continuous-time signal would be irrelevant in such a case. However, more often than not discrete-time signals are derived from corresponding continuous-time signals and, as a result, they inherit the spectral characteristics of the latter. Discrete-time signals of this type are often processed by digital systems and after that they are converted back to continuous-time signals. A case in point can be found in the recording industry where music is first sampled to generate a discrete-time signal which is then recorded on a CD. When the CD is played back, the discrete-time signal is converted into a continuous-time signal. In order to preserve the spectrum of the underlying continuous-time signal, e.g., that delightful piece of music, through this series of signal manipulations, special attention must be paid to the spectral relationships that exist between continuous- and discrete-time signals. These relationships are examined in great detail in Chapters 6 and 7. In the application just described, part of the processing must be performed by analog filters. As will be shown in Chapter 6, there is often a need to use a bandlimiting analog filter before sampling and, on the other hand, the continuous-time signal we hear through our stereo systems is produced by yet another analog filter. Therefore, knowledge of analog filters is prerequisite if we are called upon to design DSP systems that involve continuous-time signals in some way. Knowledge of analog filters is crucial in another respect: some of the better recursive digital filters can be designed only by converting analog into digital filters, as will be shown in Chapters 10–12 and 17.

The prerequisite knowledge for the book is a typical undergraduate mathematics background of calculus, complex analysis, and simple differential equations. At certain universities, complex analysis may not be included in the curriculum. To overcome this difficulty, the basics of complex analysis are summarized in Appendix A which can also serve as a quick reference or refresher. The derivation of the elliptic approximation in Section 10.6 requires a basic understanding of elliptic functions but it can be skipped by most readers. Since elliptic functions are not normally included in undergraduate curricula, a brief but adequate treatment of these functions is included in Appendix B for the sake of completeness. Chapter 14 requires a basic understanding of random variables and processes which may not be part of the curriculum at certain universities. To circumvent this difficulty, the prerequisite knowledge on random variables and processes is summarized in Chapter 13.

Chapter 1 provides an overview of DSP. It starts with a classification of the types of signals encountered in DSP. It then introduces in a heuristic way the characterization of signals in terms of frequency spectrums. The filtering process as a means of transforming or altering the spectrum of a signal is then described. The second half of the chapter provides a historical perspective of the evolution of analog and digital filters and their applications. The chapter concludes with two specific applications that illustrate the scope, diversity, and usefulness of DSP.

Chapter 2 describes the Fourier series and Fourier transform as the principal mathematical entities for the spectral characterization of continuous-time signals. The Fourier transform is deduced from the Fourier series through a limiting process whereby the period of a periodic signal is stretched to infinity.

The most important mathematical tool for the representation of discrete-time signals is the *z transform* and this forms the subject matter of Chapter 3. The z transform is viewed as a Laurent series and that immediately causes the z transform to inherit the mathematical properties of the Laurent series. By this means, the convergence properties of the z transform are more clearly understood and, furthermore, a host of algebraic techniques become immediately applicable in the inversion of the z transform. The chapter also deals with the use of the z transform as a tool for the spectral representation of discrete-time signals.

Chapter 4 deals with the fundamentals of discrete-time systems. Topics considered include basic system properties such as linearity, time invariance, causality, and stability; characterization of discrete-time systems by difference equations; representation by networks and signal flow graphs and analysis by node-elimination techniques. Time-domain analysis is introduced at an elementary level. The analysis is accomplished by solving the difference equation of the system by using induction. Although induction is not known for its efficiency, it is an intuitive technique that provides the newcomer with a clear understanding of the basics of discrete-time systems and how they operate, e.g., what are initial conditions, what is a transient or steady-state response, what is an impulse response, and so on. The chapter continues with the representation of discrete-time systems by convolution summations on the one hand and by state-space characterizations on the other.

The application of the z transform to discrete-time systems is covered in Chapter 5. By applying the z transform to the convolution summation, a discrete-time system can be represented by a transfer function that encapsulates all the linear properties of the system, e.g., time-domain response, stability, steady-state sinusoidal response, and frequency response. The chapter also includes stability criteria and algorithms that can be used to decide with minimal computational effort whether a discrete-time system is stable or not. The concepts of amplitude and phase responses and their physical significance are illustrated by examples as well as by two- and three-dimensional MATLAB plots that show clearly the true nature of zeros and poles. Chapter 5 also delineates the standard first- and second-order transfer functions that can be used to design lowpass, highpass, bandpass, bandstop, and allpass digital filters. The chapter concludes with a discussion on the causes and elimination of signal distortion in discrete-time systems such as amplitude distortion and delay distortion.

Chapter 6 extends the application of the Fourier transform to impulse and periodic signals. It also introduces the class of impulse-modulated signals which are, in effect, both sampled and continuous in time. As such, they share characteristics with both continuous- as well as discrete-time signals. Therefore, these signals provide a bridge between the analog and digital worlds and thereby facilitate the DSP practitioner to interrelate the spectral characteristics of discrete-time signals with those of the continuous-time signals from which they were derived. The chapter also deals with the sampling process, the use of digital filters for the processing of continuous-time signals, and the characterization and imperfections of analog-to-digital and digital-to-analog converters.

Chapter 7 presents the discrete Fourier transform (DFT) and the associated fast Fourier-transform method as mathematical tools for the analysis of signals on the one hand and for the software implementation of digital filters on the other. The chapter starts with the definition and properties of the DFT and continues with the interrelations that exist between the DFT and (1) the z transform, (2) the continuous Fourier transform, and (3) the Fourier series. These interrelations must be thoroughly understood, otherwise the user of the fast Fourier-transform method is likely to end up with inaccurate spectral representations for the signals of interest. The chapter also deals with the window method in detail, which can facilitate the processing of signals of long or infinity duration.

Chapters 1 to 7 deal, in effect, with the characterization and properties of continuous- and discrete-time, periodic and nonperiodic signals, and with the general properties of discrete-time systems in general. Chapters 8 to 18, on the other hand, are concerned with the design of various types of digital filters. The design process is deemed to comprise four steps, namely, approximation, realization, implementation, and study of system imperfections brought about by the use of finite arithmetic. Approximation is the process of generating a transfer function that would satisfy the required specifications. Realization is the process of converting the transfer function or some other characterization of the digital filter into a digital network or structure. Implementation can take two forms, namely, software and hardware. In a software implementation, a difference equation or state-space representation is converted into a computer program that simulates the performance of the digital filter, whereas in a hardware implementation a digital network is converted into a piece of dedicated hardware. System imperfections are almost always related to the use of finite-precision arithmetic and manifest themselves as numerical errors in filter parameters or the values of the signals being processed.

Although the design process always starts with the solution of the approximation problem, the realization process is much easier to deal with and for this reason it is treated first in Chapter 8. As will be shown, several realization methods are available that lead to a great variety of digital-filter structures. Chapter 8 also deals with a special class of structures known as systolic structures which happen to have some special properties that make them amenable to integrated-circuit implementation.

Chapter 9 is concerned with closed-form methods that can be used to design nonrecursive filters. The chapter starts by showing that constant-delay (linear-phase) nonrecursive filters can be easily designed by forcing certain symmetries on the impulse response. The design of such filters through the use of the Fourier series in conjunction with the window method is then described. Several of the standard window functions, including the Dolph-Chebyshev and Kaiser window functions, and their interrelations are detailed. The chapter includes a step-by-step design procedure based on the Kaiser window function that can be used to design standard nonrecursive filters that would satisfy prescribed specifications. It concludes with a method based on the use of classical numerical analysis formulas which can be used to design specialized nonrecursive filters that can perform interpolation, differentiation, and integration.

The approximation problem for recursive filters can be solved by using direct or indirect methods. In direct methods, the discrete-time transfer function is obtained directly in the z domain usually through iterative optimization methods. In indirect methods, on the other hand, the discrete-time transfer function is obtained by converting the continuous-time transfer function of an appropriate analog filter through a series of transformations. Thus the need arises for the solution of the approximation problem in analog filters. The basic concepts pertaining to the characterization of analog filters and the standard approximation methods used to design analog lowpass filters, i.e., the Butterworth, Chebyshev, inverse-Chebyshev, elliptic, and Bessel-Thomson methods, are described in detail in Chapter 10. The chapter concludes with certain classical transformations that can be used to convert a given lowpass approximation into a corresponding highpass, bandpass, or bandstop approximation.

Chapter 11 deals with the approximation problem for recursive digital filters. Methods are described by which a given continuous-time transfer function can be transformed into a corresponding discrete-time transfer function, e.g., the invariant impulse-response, matched-z transformation, and bilinear-transformation methods. The chapter concludes with certain transformations that can be used to convert a given lowpass digital filter into a corresponding highpass, bandpass, or bandstop digital filter. A detailed procedure that can be used to design Butterworth, Chebyshev, inverse-Chebyshev, and elliptic filters that would satisfy prescribed specifications, with design examples, is found in Chapter 12.

The basics of random variables and the extension of these principles to random processes as a means of representing random signals are introduced in Chapter 13. Random variables and signals arise naturally in digital filters because of the inevitable quantization of filter coefficients and signal values. The effects of finite word length in digital filters along with relevant up-to-date methods of analysis are discussed in Chapter 14. The topics considered include coefficient quantization and methods to reduce its effects; signal scaling; product quantization and methods to reduce its effects; parasitic and overflow limit-cycle oscillations and methods to eliminate them.

Chapters 15 and 16 deal with the solution of the approximation problem using iterative optimization methods. Chapter 15 describes a number of efficient algorithms based on the Remez exchange algorithm that can be used to design nonrecursive filters of the standard types, e.g., lowpass, highpass, bandpass, and bandstop filters, and also specialized filters, e.g., filters with arbitrary amplitude responses, multiband filters, and digital differentiators. Chapter 16, on the other hand, considers the design of recursive digital filters by optimization. To render this material accessible to the reader who has not had the opportunity to study optimization before, a series of progressively improved but related algorithms is presented starting with the classical Newton algorithm for convex problems and culminating in a fairly sophisticated, practical, and efficient quasi-Newton algorithm that can be used to design digital filters with arbitrary frequency responses. Chapter 16 also deals with the design of recursive equalizers which are often used to achieve a linear phase response in a recursive filter.

Chapter 17 is in effect a continuation of Chapter 8 and it deals with the realization of digital filters in the form of wave digital filters. These structures are derived from classical analog filters and, in consequence, they have certain attractive features, such as low sensitivity to numerical errors, which make them quite attractive for certain applications. The chapter includes step-by-step procedures by which wave digital filters satisfying prescribed specifications can be designed either in ladder or lattice form. The chapter concludes with a list of guidelines that can be used to choose a digital-filter structure from the numerous possibilities described in Chapters 8 and 12.

Chapter 18 deals with some of the numerous applications of digital filters to digital signal processing. The applications considered include downsampling and upsampling using decimators and interpolators, the design of quadrature-mirror-image filters and their application in time-division to frequency-division multiplex translation, Hilbert transformers and their application in single-sideband modulation, adaptive filters, and two-dimensional digital filters.

The purpose of Appendix A is twofold. First, it can be regarded as a brief review of complex analysis for readers who have not had the opportunity to take a course on this important subject. Second, it can serve as a reference monograph that brings together those principles of complex analysis that are required for DSP. Appendix B, on the other hand, presents the basic principles of elliptic integrals and functions and its principal purpose is to facilitate the derivation of the elliptic approximation in Chapter 10.

The book can serve as a text for undergraduate or graduate courses and various scenarios are possible depending on the background preparation of the class and the curriculum of the institution. Some possibilities are as follows:

- *Series of Two Undergraduate Courses.* First-level course: Chapters 1 to 7, second-level course: Chapters 8 to 14
- *Series of Two Graduate Courses.* First-level course: Chapters 5 to 12, second-level course: Chapters 13 to 18
- *One Undergraduate/Graduate Course.* Assuming that the students have already taken relevant courses on signal analysis and system theory, a one-semester course could be offered comprising Chapters 5 to 12 and parts of Chapter 14.

The book is supported by the author's DSP software package *D-Filter* which can be used to analyze, design, and realize digital filters, and to analyze discrete-time signals. See D-Filter page at the end of the book for more details. The software can be downloaded from D-Filter's website: www.d-filter.com or www.d-filter.ece.uvic.ca. In addition, a detailed *Instructor's Manual* and PDF slides for classroom use are now being prepared, which will be made available to instructors adopting the book through the author's website: www.ece.uvic.ca/~ andreas.

I would like to thank Stuart Bergen, Rajeev Nongpiur, and Wu-Sheng Lu for reviewing the reference lists of certain chapters and supplying more up-to-date references; Tarek Nasser for checking certain parts of the manuscript; Randy K. Howell for constructing the plots in Figures 16.12 and 16.13;

Majid Ahmadi for constructive suggestions; Tony Antoniou for suggesting improvements in the design of the cover and title page of the book and for designing the installation graphics and icons of D-Filter; David Guindon for designing a new interface for D-Filter; Catherine Chang for providing help in updating many of the illustrations; Lynne Barrett for helping with the proofreading; Michelle L. Flomenhoft, Development Editor, Higher Education Division, McGraw-Hill, for her many contributions to the development of the manuscript and for arranging the reviews; to the reviewers of the manuscript for providing useful suggestions and identifying errata, namely, Scott T. Acton, University of Virginia; Selim Awad, University of Michigan; Vijayakumar Bhagavatula, Carnegie Mellon University; Subra Ganesan, Oakland University; Martin Haenggi, University of Notre Dame; Donald Hummels, University of Maine; James S. Kang, California State Polytechnic University; Takis Kasparis, University of Central Florida; Preetham B. Kumar, California State University; Douglas E. Melton, Kettering University; Behrooz Nowrouzian, University of Alberta; Wayne T. Padgett, Rose-Hulman Institute of Technology; Roland Priemer, University of Illinois, Chicago; Stanley J. Reeves, Auburn University; Terry E. Riemer, University of New Orleans; A. David Salvia, Pennsylvania State University; Ravi Sankar, University of South Florida; Avtar Singh, San Jose State University; Andreas Spanias, Arizona State University; Javier Vega-Pineda, Instituto Tecnologico de Chihuahua; Hsiao-Chun Wu, Louisiana State University; Henry Yeh, California State University. Thanks are also due to Micronet, Networks of Centres of Excellence Program, Canada, the Natural Sciences and Engineering Research Council of Canada, and the University of Victoria, British Columbia, Canada, for supporting the research that led to many of the author's contributions to DSP as described in Chapters 12 and 14 to 17. Last but not least, I would like to express my thanks and appreciation to Mona Tiwary, Project Manager, International Typesetting and Composition, and Stephen S. Chapman, Editorial Director, Professional Division, McGraw-Hill, for seeing the project through to a successful conclusion.

Andreas Antoniou

CHAPTER 1

INTRODUCTION TO DIGITAL SIGNAL PROCESSING

1.1 INTRODUCTION

The overwhelming advancements in the fabrication of microchips and their application for the design of efficient digital systems over the past 50 years have led to the emergence of a new discipline that has come to be known as *digital signal processing* or DSP.

Through the use of DSP, sophisticated communication systems have evolved, the Internet emerged, astronomical signals can be distilled into valuable information about the cosmos, seismic signals can be analyzed to determine the strength of an earthquake or to predict the stability of a volcano, computer images or photographs can be enhanced, and so on.

This chapter deals with the underlying principles of DSP. It begins by examining the types of signals that are encountered in nature, science, and engineering and introduces the sampling process which is the means by which analog signals can be converted to corresponding digital signals. It then examines the types of processing that can be applied to a signal and the types of systems that are available for the purpose. The chapter concludes with two introductory applications that illustrate the nature of DSP for the benefit of the neophyte.

1.2 SIGNALS

Signals arise in almost every field of science and engineering, e.g., in astronomy, acoustics, biology, communications, seismology, telemetry, and economics to name just a few. Signals arise naturally through certain physical processes or are man-made. Astronomical signals can be generated by huge cosmological explosions called supernovas or by rapidly spinning neutron stars while seismic signals are the manifestations of earthquakes or volcanos that are about to erupt. Signals also abound in biology, e.g., the signals produced by the brain or heart, the acoustic signals used by dolphins or whales to communicate with one another, or those generated by bats to enable them to navigate or catch prey. Man-made signals, on the other hand, occur in technological systems, as might be expected, like computers, telephone and radar systems, or the Internet. Even the market place is a source of numerous vital signals, e.g., the prices of commodities at a stock exchange or the Dow Jones Industrial Average.

We are very interested in natural signals for many reasons. Astronomers can extract important information from optical signals received from the stars, e.g., their chemical composition, they can decipher the nature of a supernova explosion, or determine the size of a neutron star from the periodicity of the signal received. Seismologists can determine the strength and center of an earthquake whereas volcanologists can often predict whether a volcano is about to blow its top. Cardiologists can diagnose various heart conditions by looking for certain telltale patterns or aberrations in electrocardiographs.

We are very interested in man-made signals for numerous reasons: they make it possible for us to talk to one another over vast distances, enable the dissemination of huge amounts of information over the Internet, facilitate the different parts of a computer to interact with one another, instruct robots how to perform very intricate tasks rapidly, help aircraft to land in poor weather conditions and low visibility, or warn pilots about loss of separation between aircraft to avoid collisions. On the other hand, the market indices can help us determine whether it is the right time to invest and, if so, what type of investment should we go for, equities or bonds.

In the above paragraphs, we have tacitly assumed that a signal is some quantity, property, or variable that depends on time, for example, the light intensity of a star or the strength of a seismic signal. Although this is usually the case, signals exist in which the independent parameter is some quantity other than time, and the number of independent variables can be more than one occasionally. For example, a photograph or radiograph can be viewed as a two-dimensional signal where the light intensity depends on the x and y coordinates which happen to be lengths. On the other hand, a TV image which changes with time can be viewed as a three-dimensional signal with two of the independent variables being lengths and one being time.

Signals can be classified as

- *continuous-time*, or
- *discrete-time*.

Continuous-time signals are defined at each and every instant of time from start to finish. For example, an electromagnetic wave originating from a distant galaxy or an acoustic wave produced by a dolphin. On the other hand, discrete-time signals are defined at discrete instants of time, perhaps every millisecond, second, or day. Examples of this type of signal are the closing price of a particular commodity at a stock exchange and the daily precipitation as functions of time.

Nature's signals are usually continuous in time. However, there are some important exceptions to the rule. For example, in the domain of quantum physics electrons gain or lose energy in discrete amounts and, presumably, at discrete instants. On the other hand, the DNA of all living things is constructed from a ladder-like structure whose ranks are made from four fundamental distinct organic molecules. By assigning distinct numbers to these basic molecules and treating the length of the ladder-like structure as if it were time, the genome of any living organism can be represented by a discrete-time signal. Man-made signals can be continuous- or discrete-time and typically the type of signal depends on whether the system that produced it is analog or digital.

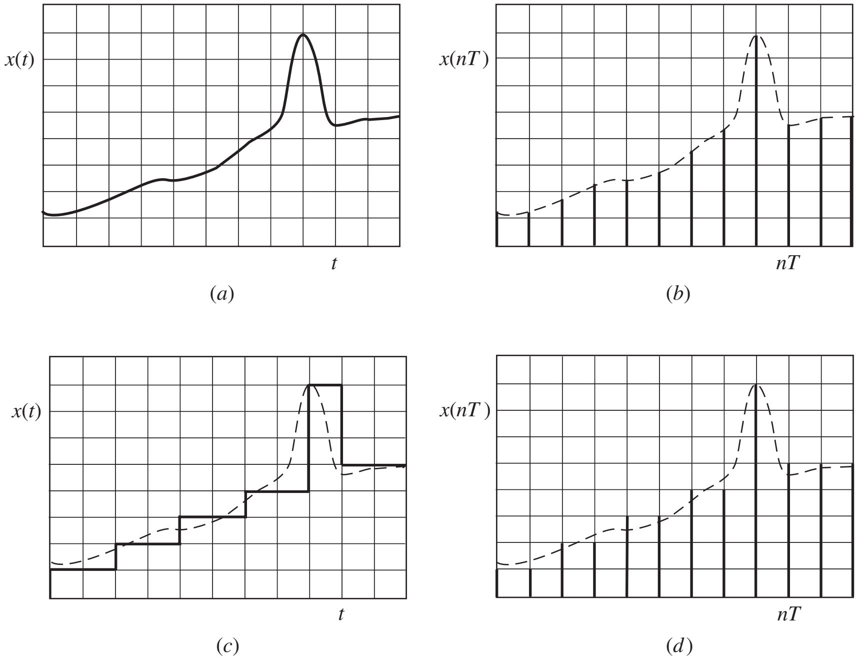
In mathematical terms, a continuous-time signal can be represented by a function $x(t)$ whose domain is a range of numbers $(t_1,\ t_2)$, where $-\infty < t_1$ and $t_2 < \infty$, as illustrated in Fig. 1.1*a*. Similarly, a discrete-time signal can be represented by a function $x(nT)$, where T is the period between adjacent discrete signal values and n is an integer in the range $(n_1,\ n_2)$ where $-\infty < n_1$ and $n_2 < \infty$, as shown in Fig. 1.1*b*. Discrete-time signals are often generated from corresponding continuous-time signals through a sampling process and T is, therefore, said to be the *sampling period*. Its reciprocal, i.e., $f_s = 1/T$, is known as the *sampling frequency*.

Signals can also be classified as

- *nonquantized*, or
- *quantized*.

A nonquantized signal can assume any value in a specified range, whereas a quantized signal can assume only discrete values, usually equally spaced. Figure 1.1*c* and *d* shows quantized continuous-time and quantized discrete-time signals, respectively.

Signals are sometimes referred to as *analog* or *digital* in the literature. By and large, an analog signal is deemed to be a continuous-time signal, and vice versa. Similarly, a digital signal is deemed to be a discrete-time signal, and vice versa. A pulse waveform, like the numerous waveforms found in a typical digital system, would be regarded as a digital signal if the focus were on its two-level idealized representation. However, if the exact actual level of the waveform were of interest, then the pulse waveform would be treated as a continuous-time signal as the signal level can assume an infinite set of values.



(*a*)

(*b*)

(*c*)

(*d*)

Figure 1.1 Types of signals: (*a*) Nonquantized continuous-time signal, (*b*) nonquantized discrete-time signal, (*c*) quantized continuous-time signal, (*d*) quantized discrete-time signal.

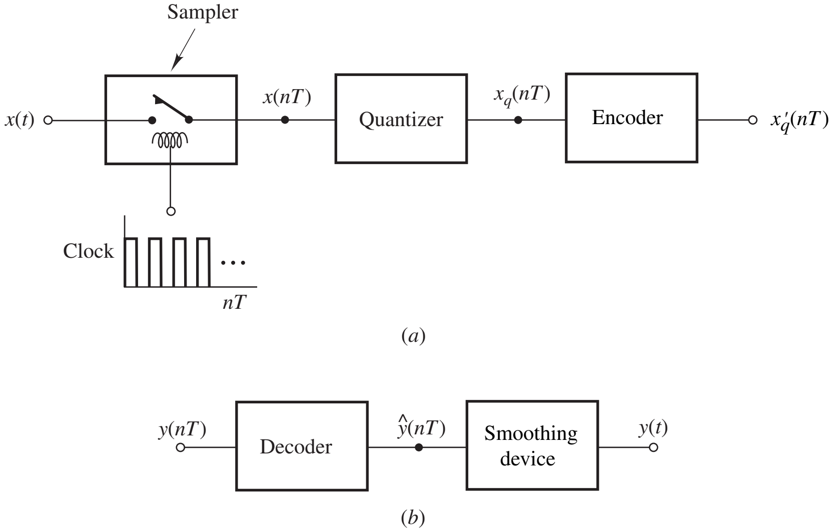


Figure 1.2 Sampling system: (*a*) A/D interface, (*b*) D/A interface.

Discrete-time signals are often generated from corresponding continuous-time signals through the use of an analog-to-digital (A/D) interface and, similarly, continuous-time signals can be obtained by using a digital-to-analog (D/A) interface. An A/D interface typically comprises three components, namely, a sampler, a quantizer, and an encoder as depicted in Fig. 1.2*a*. In the case where the signal is in the form of a continuous-time voltage or current waveform, the sampler in its bare essentials is a switch controlled by a clock signal, which closes momentarily every T s thereby transmitting the level of the input signal $x(t)$ at instant nT, that is, $x(nT)$, to the output. A quantizer is an analog device that will sense the level of its input and produce as output the nearest available level, say, $x_q(nT)$, from a set of allowed levels, i.e., a quantizer will produce a quantized continuous-time signal such as that shown in Fig. 1.1*c*. An encoder is essentially a digital device that will sense the voltage or current level of its input and produce a corresponding number at the output, i.e., it will convert a quantized continuous-time signal of the type shown in Fig. 1.1*c* to a corresponding discrete-time signal of the type shown in Fig. 1.1*d*.

The D/A interface comprises two modules, a decoder and a smoothing device as depicted in Fig. 1.2*b*. The decoder will convert a discrete-time signal into a corresponding quantized voltage waveform such as that shown in Fig. 1.1*c*. The purpose of the smoothing device is to smooth out the quantized waveform and thus eliminate the inherent discontinuities.

The A/D and D/A interfaces are readily available as off-the-shelf components known as A/D and D/A converters and many types, such as high-speed, low-cost, and high-precision, are available.

1.3 FREQUENCY-DOMAIN REPRESENTATION

Signals have so far been represented in terms of functions of time, i.e., $x(t)$ or $x(nT)$. In many situations, it is useful to represent signals in terms of functions of frequency. For example, a

Table 1.1 Parameters of signal in Eq. (1.1)

k	ω_k	A_k	ϕ_k
1	1	0.6154	0.0579
2	2	0.7919	0.3529
3	3	0.9218	−0.8132
4	4	0.7382	0.0099
5	5	0.1763	0.1389
6	6	0.4057	−0.2028
7	7	0.9355	0.1987
8	8	0.9169	−0.6038
9	9	0.4103	−0.2722

continuous-time signal made up of a sum of sinusoidal components such as

$$x(t) = \sum_{k=1}^{9} A_k \sin(\omega_k t + \phi_k) \tag{1.1}$$

can be fully described by two sets,[1] say,

$$A(\omega) = \{A_k\colon \omega = \omega_k \text{ for } k = 1, 2, \ldots, 9\}$$

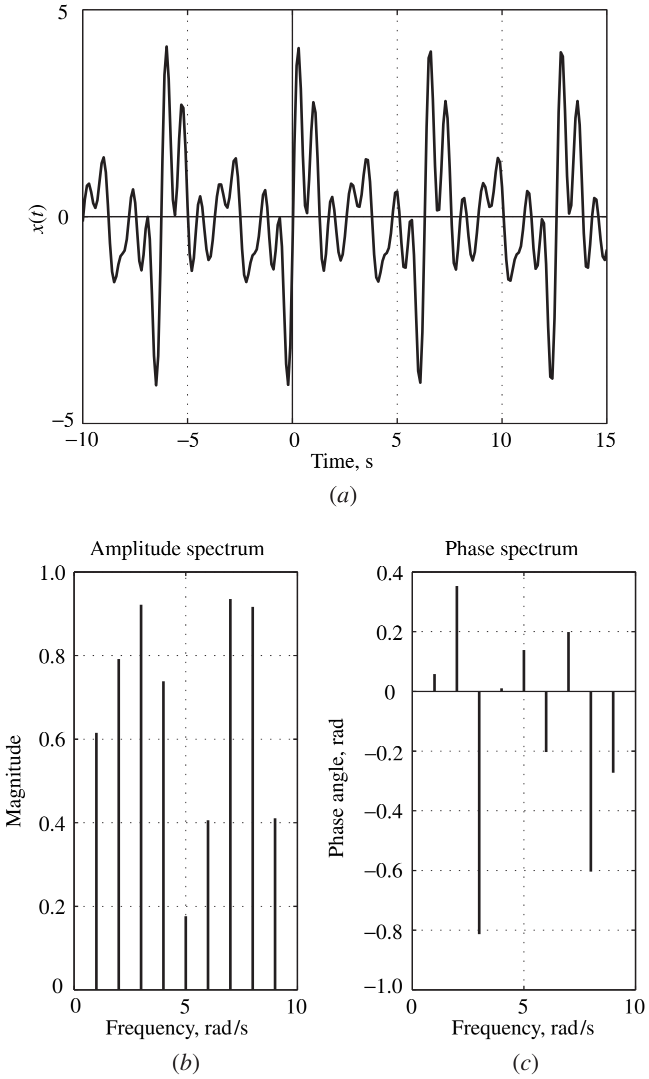
and

$$\phi(\omega) = \{\phi_k\colon \omega = \omega_k \text{ for } k = 1, 2, \ldots, 9\}$$

that describe the amplitudes and phase angles of the sinusoidal components present in the signal. Sets $A(\omega)$ and $\phi(\omega)$ can be referred to as the *amplitude spectrum* and *phase spectrum* of the signal, respectively, for obvious reasons, and can be represented by tables or graphs that give the amplitude and phase angle associated with each frequency. For example, if A_k and ϕ_k in Eq. (1.1) assume the numerical values given by Table 1.1, then $x(t)$ can be represented in the time domain by the graph in Fig. 1.3*a* and in the frequency domain by Table 1.1 or by the graphs in Fig. 1.3*b* and *c*.

The usefulness of a frequency-domain or simply *spectral* representation can be well appreciated by comparing the time- and frequency-domain representations in Fig. 1.3. The time-domain representation shows that what we have is a noise-like periodic signal. Its periodicity is to be expected as the signal is made up of a sum of sinusoidal components that are periodic. The frequency-domain representation, on the other hand, provides a fairly detailed and meaningful description of the individual frequency components, namely, their frequencies, amplitudes, and phase angles.

[1] *This representation of a set will be adopted throughout the book.*



(*a*)

(*b*)

(*c*)

Figure 1.3 Time- and frequency-domain representations of the periodic signal represented by Eq. (1.1) with the parameters given in Table 1.1: (*a*) Time-domain representation, (*b*) amplitude spectrum, (*c*) phase spectrum.

The representation in Eq. (1.1) is actually the *Fourier series* of signal $x(t)$ and deriving the Fourier series of a periodic signal is just one way of obtaining a spectral representation for a signal. Scientists, mathematicians, and engineers have devised a variety of mathematical tools that can be used for the spectral representation of different types of signals. Other mathematical tools, in addition to the Fourier series, are the *Fourier transform* which is applicable to periodic as well as nonperiodic continuous-time signals; the *z transform* which is the tool of choice for discrete-time nonperiodic signals; and the *discrete-Fourier* transform which is most suitable for discrete-time periodic signals.

The Fourier series and Fourier transform will be reviewed in Chap. 2, the z transform will be examined in detail in Chap. 3, and the discrete-Fourier transform will be treated in Chap. 7.

1.4 NOTATION

The notation introduced in Sec. 1.2 for the representation of discrete-time signals, i.e., $x(nT)$, preserves the exact relation between a discrete-time signal and the underlying continuous-time signal $x(t)$ for the case where the former is generated from the latter through the sampling process. The use of this notation tends to be somewhat cumbersome on account of the numerous Ts that have to be repeated from one equation to the next. For the sake of simplicity, many authors use $x(n)$ or x_n instead of $x(nT)$. These simplified notations solve one problem but create another. For example, a discrete-time signal generated from the continuous-time signal

$$x(t) = e^{\alpha t}\sin(\omega t)$$

through the sampling process would naturally be

$$x(nT) = e^{\alpha nT}\sin(\omega nT)$$

If we were to drop the T in $x(nT)$, that is,

$$x(n) = e^{\alpha nT}\sin(\omega nT)$$

then a notation inconsistency is introduced as evaluating $x(t)$ at $t = n$ does not give the correct expression for the discrete-time signal. This problem tends to propagate into the frequency domain and, in fact, it causes the spectral representation of the discrete-time signal to be inconsistent with that of the underlying continuous-time signal.

The complex notation can be avoided while retaining consistency between the continuous- and discrete-time signals through the use of *time normalization*. In this process, the time axis of the continuous-time signal is scaled by replacing t by t/T in $x(t)$, that is,

$$x(t)|_{t\to t/T} = x\left(\frac{t}{T}\right) = e^{\alpha(t/T)}\sin\left(\omega\cdot\frac{t}{T}\right)$$

If t is now replaced by nT, we get

$$x(n) = e^{\alpha(nT/T)}\sin\left(\omega\cdot\frac{nT}{T}\right) = e^{\alpha n}\sin(\omega n)$$

In the above time normalization, the sampling period is, in effect, changed from T to 1 s and, consequently, T disappears from the picture. Time normalization can be reversed by applying *time denormalization* by simply replacing n by nT where T is the actual sampling period.

In this book, the full notation $x(nT)$ will be used when dealing with the fundamentals, namely, in Chaps. 3–6. In later chapters, signals will usually be assumed to be normalized with respect to time and, in such cases, the simplified notation $x(n)$ will be used. The notation x_n will not be used.

It was mentioned earlier that the independent variable can be some quantity other than time, e.g., length. Nevertheless, the symbol T will be used for these situations as well, for the sake of a consistent notation. In certain situations, the entity to be processed may well be just a sequence of numbers that are independent of any physical quantity. In such situations, $x(n)$ is the correct notation. The theories presented in this book apply equally well to such entities but the notions of *time domain* and *frequency domain* lose their usual physical significance. We are, in effect, dealing with mathematical transformations.

1.5 SIGNAL PROCESSING

Signal processing is the science of analyzing, synthesizing, sampling, encoding, transforming, decoding, enhancing, transporting, archiving, and in general manipulating signals in some way. With the rapid advances in very-large-scale integrated (VLSI) circuit technology and computer systems, the subject of signal processing has mushroomed into a multifaceted discipline with each facet deserving its own volume. This book is concerned primarily with the branch of signal processing that entails the spectral characteristics and properties of signals.

The spectral representation and analysis of signals in general are carried out through the mathematical transforms alluded to in the previous section, e.g., the Fourier series and Fourier transform. If the processing entails modifying, reshaping, or transforming the spectrum of a signal in some way, then the processing involved will be referred to as *filtering* in this book.

Filtering can be used to select one or more desirable and simultaneously reject one or more undesirable bands of frequency components, or simply frequencies. For example, one could use lowpass filtering to select a band of preferred low frequencies and reject a band of undesirable high frequencies from the frequencies present in the signal depicted in Fig. 1.3, as illustrated in Fig. 1.4; use highpass filtering to select a band of preferred high frequencies and reject a band of undesirable low frequencies as illustrated in Fig. 1.5; use bandpass filtering to select a band of frequencies and reject low and high frequencies as illustrated in Fig. 1.6; or use bandstop filtering to reject a band of frequencies but select low frequencies and high frequencies as illustrated in Fig. 1.7.

In the above types of filtering, one or more undesirable bands of frequencies are rejected or filtered out and the term *filtering* is quite appropriate. In some other types of filtering, certain frequency components are strengthened while others are weakened, i.e., nothing is rejected or filtered out. Yet these processes transform the spectrum of the signal being processed and, as such, they fall under the category of filtering in the broader definition of filtering adopted in this book. Take differentiation, for example. Differentiating the signal in Eq. (1.1) with respect to t gives

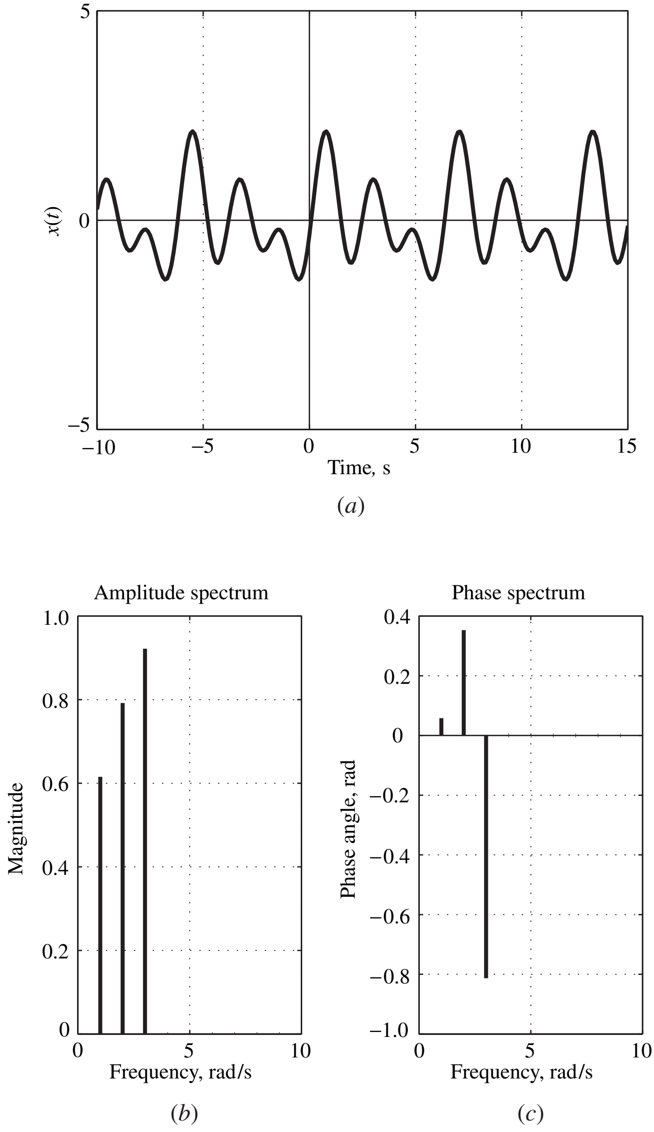
$$\begin{aligned}\frac{dx(t)}{dt} &= \sum_{k=1}^{9} \frac{d}{dt}[A_k \sin(\omega_k t + \phi_k)] = \sum_{k=1}^{9} \omega_k A_k \cos(\omega_k t + \phi_k)\\ &= \sum_{k=1}^{9} \omega_k A_k \sin\left(\omega_k t + \phi_k + \tfrac{1}{2}\pi\right)\end{aligned}$$

The amplitude and phase spectrums of the signal have now become

$$A(\omega) = \{\omega_k A_k \colon \omega = \omega_k \text{ for } k = 1,\ 2,\ \ldots,\ 9\}$$

and

$$\phi(\omega) = \left\{\phi_k + \tfrac{1}{2}\pi \colon \omega = \omega_k \text{ for } k = 1,\ 2,\ \ldots,\ 9\right\}$$



(*a*)

(*b*)

(*c*)

Figure 1.4 Lowpass filtering applied to the signal depicted in Fig. 1.3: (*a*) Time-domain representation, (*b*) amplitude spectrum, (*c*) phase spectrum.

respectively. The effect of differentiating the signal of Eq. (1.1) is illustrated in Fig. 1.8. As can be seen by comparing Fig. 1.3*b* and *c* with Fig. 1.8*b* and *c*, differentiation scales the amplitudes of the different frequency components by a factor that is proportional to frequency and adds a phase angle of $\frac{1}{2}\pi$ to each value of the phase spectrum. In other words, the amplitudes of low-frequency components are attenuated, whereas those of high-frequency components are enhanced. In effect, the process of differentiation is a type of highpass filtering.

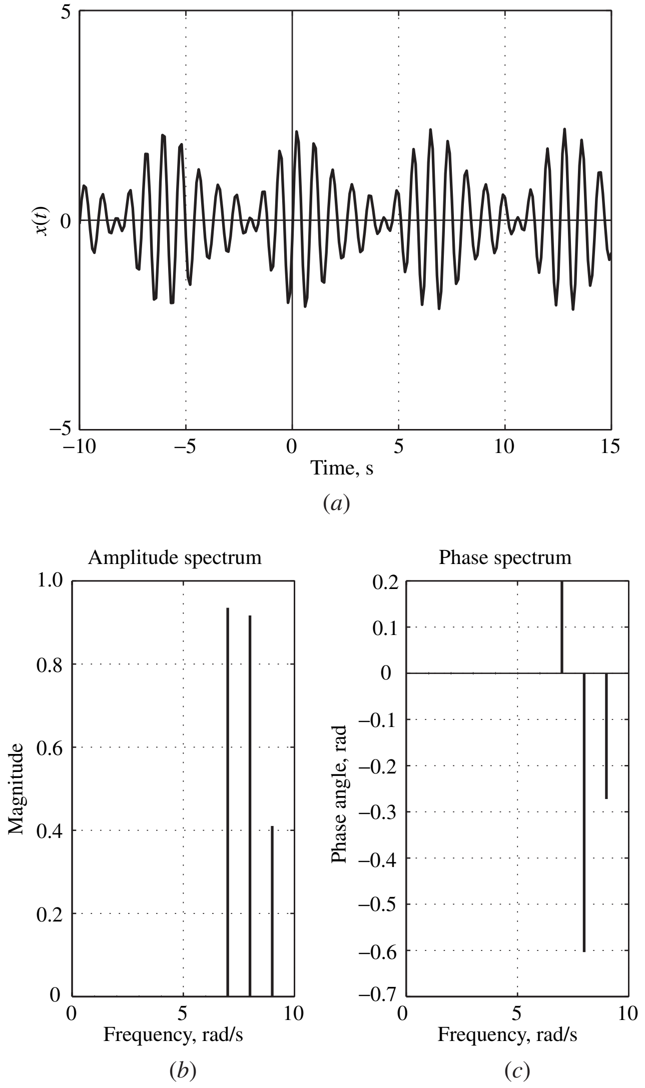


Figure 1.5 Highpass filtering applied to the signal depicted in Fig. 1.3: (*a*) Time-domain representation, (*b*) amplitude spectrum, (*c*) phase spectrum.

Integrating $x(t)$ with respect to time, on the other hand, gives

$$\int x(t)\,dt = \sum_{k=1}^{9} \int A_k \sin(\omega_k t + \phi_k)\,dt = \sum_{k=1}^{9} \left[-\frac{A_k}{\omega_k} \cos(\omega_k t + \phi_k) \right]$$

$$= \sum_{k=1}^{9} \frac{A_k}{\omega_k} \sin\left(\omega_k t + \phi_k - \tfrac{1}{2}\pi\right)$$

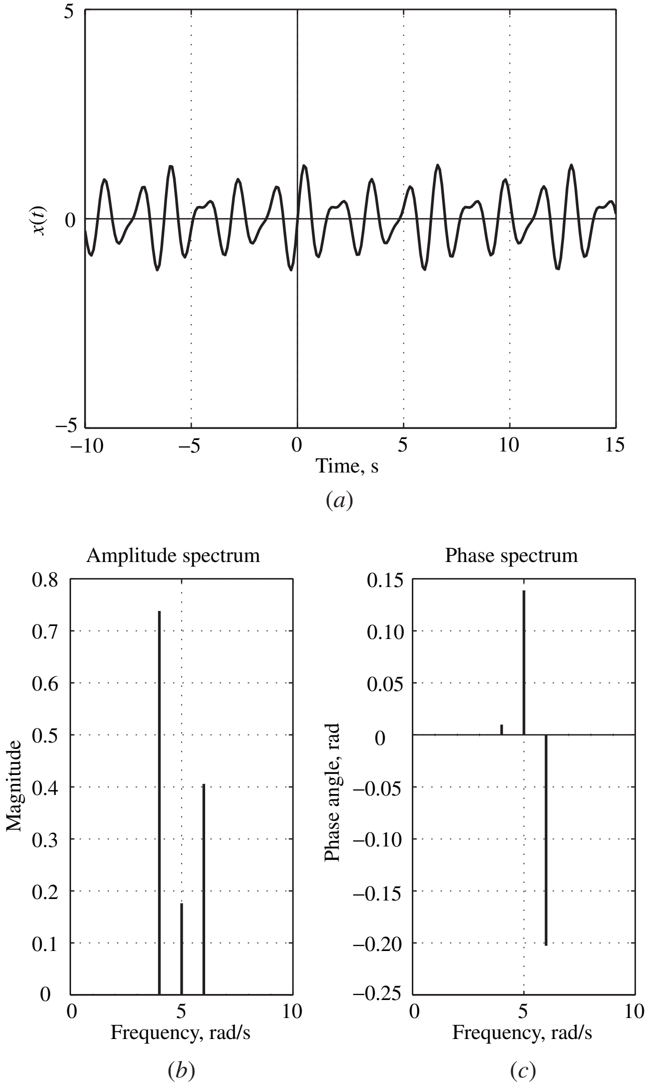


Figure 1.6 Bandpass filtering applied to the signal depicted in Fig. 1.3: (*a*) Time-domain representation, (*b*) amplitude spectrum, (*c*) phase spectrum.

In this case, the amplitude and phase spectrums become

$$A(\omega) = \{A_k/\omega_k \colon \omega = \omega_k \text{ for } k = 1,\ 2,\ \ldots,\ 9\}$$

and

$$\phi(\omega) = \left\{\phi_k - \tfrac{1}{2}\pi \colon \omega = \omega_k \text{ for } k = 1,\ 2,\ \ldots,\ 9\right\}$$

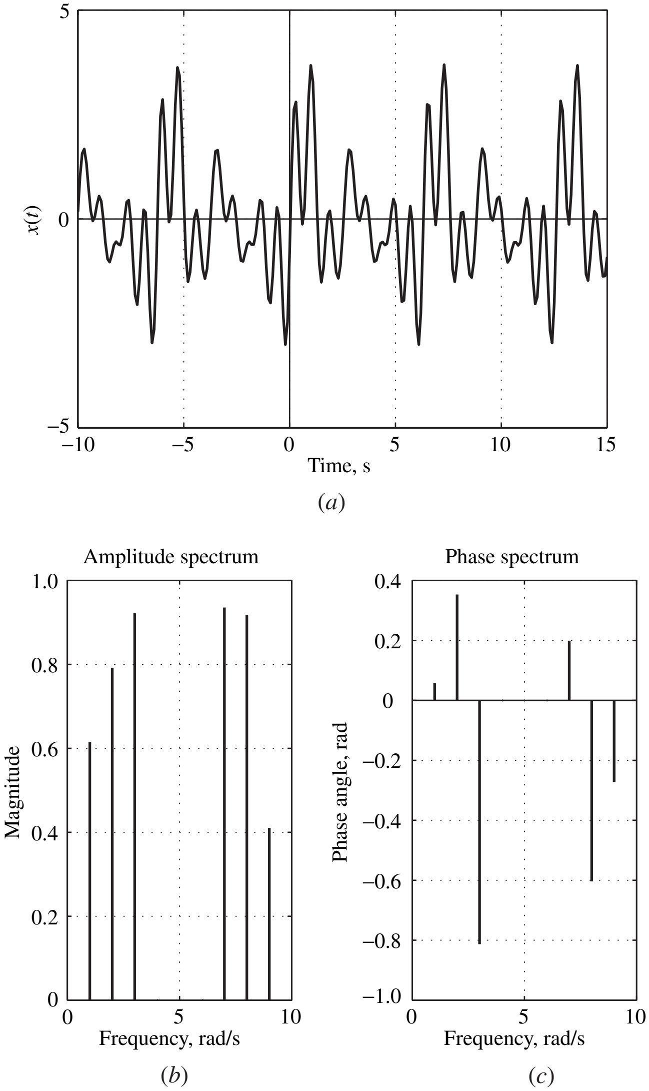
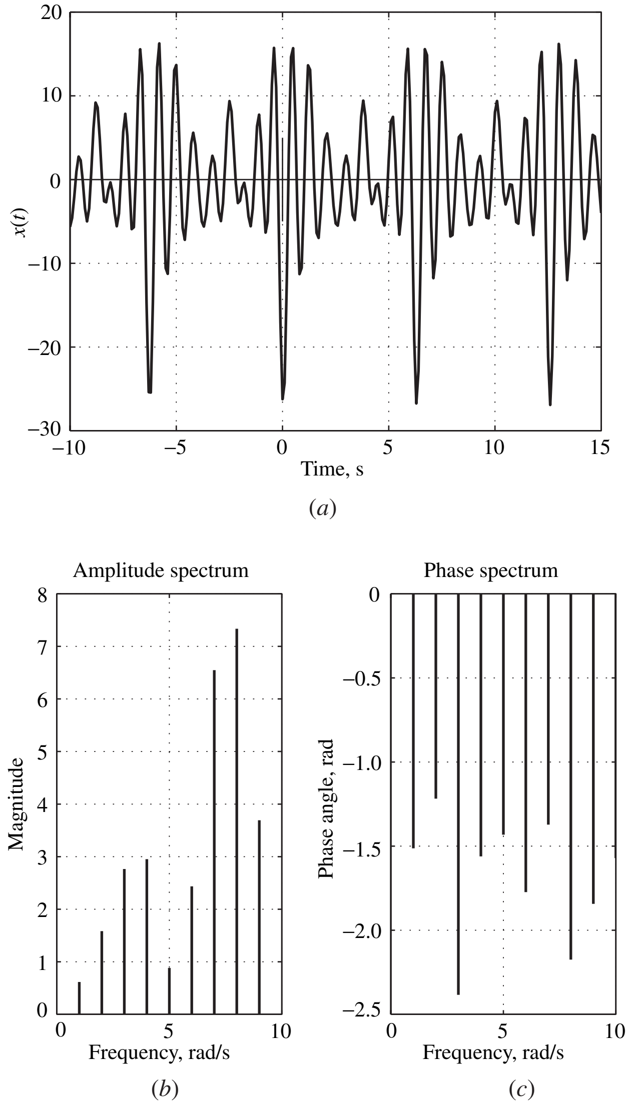


Figure 1.7 Bandstop filtering applied to the signal depicted in Fig. 1.3: (*a*) Time-domain representation, (*b*) amplitude spectrum, (*c*) phase spectrum.

respectively, i.e., the amplitudes of the different frequency components are now scaled by a factor that is inversely proportional to the frequency and a phase angle of $\frac{1}{2}\pi$ is subtracted from each value of the phase spectrum. Thus, integration tends to enhance low-frequency and attenuate high-frequency components and, in a way, it tends to behave very much like lowpass filtering as illustrated in Fig. 1.9.

In its most general form, *filtering is a process that will transform the spectrum of a signal according to some rule of correspondence.* In the case of lowpass filtering, the rule of correspondence



(*a*)

(*b*)

(*c*)

Figure 1.8 Differentiation applied to the signal depicted in Fig. 1.3: (*a*) Time-domain representation, (*b*) amplitude spectrum, (*c*) phase spectrum.

might specify, for example, that the spectrum of the output signal be approximately the same as that of the input signal for some low-frequency range and approximately zero for some high-frequency range.

Electrical engineers have known about filtering processes for well over 80 years and, through the years, they invented many types of circuits and systems that can perform filtering, which are known collectively as *filters*. Filters can be designed to perform a great variety of filtering tasks, in addition, to those illustrated in Figs. 1.4–1.9. For example, one could easily design a lowpass filter

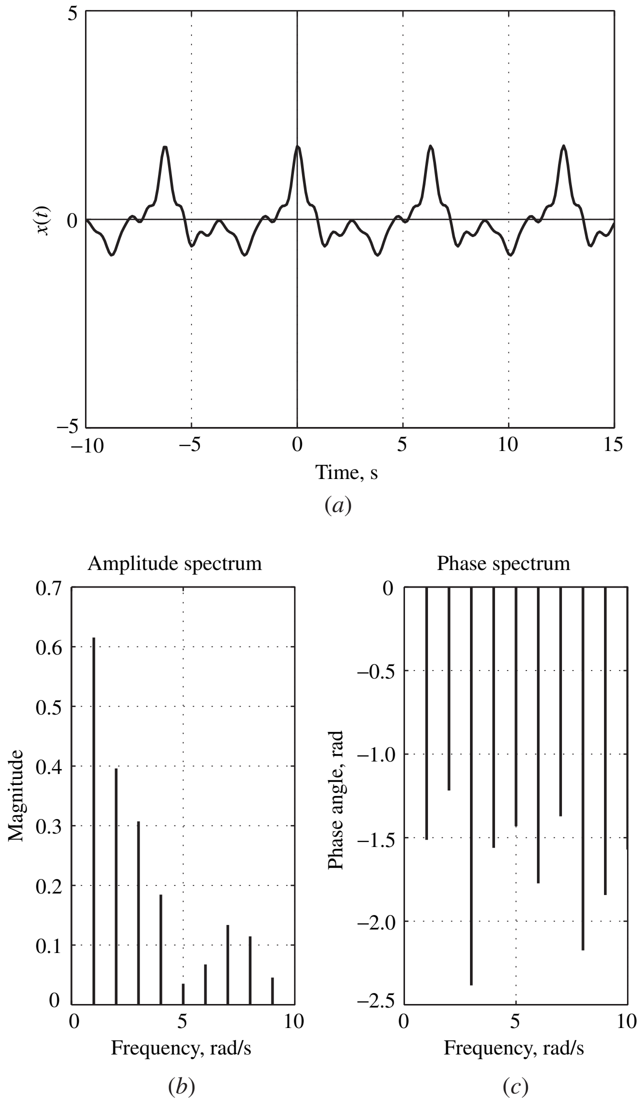


Figure 1.9 Integration applied to the signal depicted in Fig. 1.3: (*a*) Time-domain representation, (*b*) amplitude spectrum, (*c*) phase spectrum.

that would select low frequencies in the range from 0 to ω_p and reject high frequencies in the range from ω_a to ∞. In such a filter, the frequency ranges from 0 to ω_p and ω_a to ∞, are referred to as the *passband* and *stopband*, respectively.

Filters can be classified on the basis of their operating signals as *analog* or *digital*. In analog filters, the input, output, and internal signals are in the form of continuous-time signals, whereas in digital filters they are in the form of discrete-time signals.

1.6 ANALOG FILTERS

This book is concerned mainly with DSP and with discrete-time systems that can perform DSP, such as digital filters. Since digital filters evolved as a natural extension of analog filters and are often designed through the use of analog-filter methodologies, a brief outline of the historical evolution and applications of analog filters are worthwhile.

Analog filters were originally invented for use in radio receivers and long-distance telephone systems and continue to be critical components in all types of communication systems. Various families of analog filters have evolved over the years, which can be classified as follows on the basis of their constituent elements and the technology used [1, 2].[2]

- Passive RLC[3] filters
- Discrete active RC filters
- Integrated active RC filters
- Switched-capacitor filters
- Microwave filters

Passive RLC filters began to be used extensively in the early twenties. They are made of interconnected resistors, inductors, and capacitors and are said to be *passive* in view of the fact that they do not require an energy source, like a power supply, to operate. Filtering action is achieved through the property of electrical resonance which occurs when an inductor and a capacitor are connected in series or in parallel. The importance of filtering in communications motivated engineers and mathematicians between the thirties and fifties to develop some very powerful and sophisticated methods for the design of passive RLC filters.

Discrete active RC filters began to appear during the mid-fifties and were a hot topic of research during the sixties. They comprise discrete resistors, capacitors, and amplifying electronic circuits. Inductors are absent and it is this feature that makes active RC filters attractive. Inductors have always been bulky, expensive, and generally less ideal than resistors and capacitors particularly for low-frequency applications. Unfortunately, without inductors, electrical resonance cannot be achieved and with just resistors and capacitors only crude types of filters can be designed. However, through the clever use of amplifying electronic circuits in RC circuits, it is possible to simulate resonance-like effects that can be utilized to achieve filtering of high quality. These filters are said to be *active* because the amplifying electronic circuits require an energy source in the form of a power supply.

Integrated-circuit active RC filters operate on the basis of the same principles as their discrete counterparts except that they are designed directly as complete integrated circuits. Through the use of high-frequency amplifying circuits and suitable integrated-circuit elements, filters that can operate at frequencies as high as 15 GHz can be designed [3, 4].[4] Interest in these filters has been strong during the eighties and nineties and research continues.

Switched-capacitor filters evolved during the seventies and eighties. These are essentially active RC filters except that switches are also utilized along with amplifying devices. In this family

[2] *Numbered references will be found at the end of each chapter.*

[3] *R, L, and C are the symbols used for the electrical properties of resistance, inductance, and capacitance, respectively.*

[4] *One GHz equals* 10^9 *Hz.*

of filters, switches are used to simulate high resistance values which are difficult to implement in integrated-circuit form. Like integrated active RC filters, switched-capacitors filters are compatible with integrated-circuit technology.

Microwave filters are built from a variety of microwave components and devices such as transverse electromagnetic (TEM) transmission lines, waveguides, dielectric resonators, and surface acoustic devices [5]. They are used in applications where the operating frequencies are in the range 0.5 to 500 GHz.

1.7 APPLICATIONS OF ANALOG FILTERS

Analog filters have found widespread applications over the years. A short but not exhaustive list is as follows:

- Radios and TVs
- Communication and radar systems
- Telephone systems
- Sampling systems
- Audio equipment

Every time we want to listen to the radio or watch TV, we must first select our favorite radio station or TV channel. What we are actually doing when we turn the knob on the radio or press the channel button on the remote control is tuning the radio or TV receiver to the broadcasting frequency of the radio station or TV channel, and this is accomplished by aligning the frequency of a bandpass filter inside the receiver with the broadcasting frequency of the radio station or TV channel.

When we tune a radio receiver, we select the frequency of a desirable signal, namely, that of our favorite radio station. The signals from all the other stations are undesirable and are rejected. The same principle can be used to prevent a radar signal from interfering with the communication signals at an airport, for example, or to prevent the communication signals from interfering with the radar signals.

Signals are often corrupted by spurious signals known collectively as *noise*. Such signals may originate from a large number of sources, e.g., lightnings, electrical motors, transformers, and power lines. Noise signals are characterized by frequency spectrums that stretch over a wide range of frequencies. They can be eliminated through the use of bandpass filters that would pass the desired signal but reject everything else, namely, the noise content, as in the case of a radio receiver.

We all talk daily to our friends and relatives through the telephone system. More often than not, they live in another city or country and the conversation must be carried out through expensive communication channels. If these channels were to carry just a single voice, as in the days of Alexander Graham Bell,[5] no one would ever be able to afford a telephone call to anyone, even the very rich. What makes long-distance calls affordable is our ability to transmit thousands of conversations through one and the same communications channel. And this is achieved through the use of a so-called *frequency-division multiplex* (FDM) *communications system* [6]. A rudimentary

[5] *(1847–1921) Scottish-born scientist and inventor who spent most of his career in the northeast US and Canada. He invented the telephone between 1874 and 1876.*

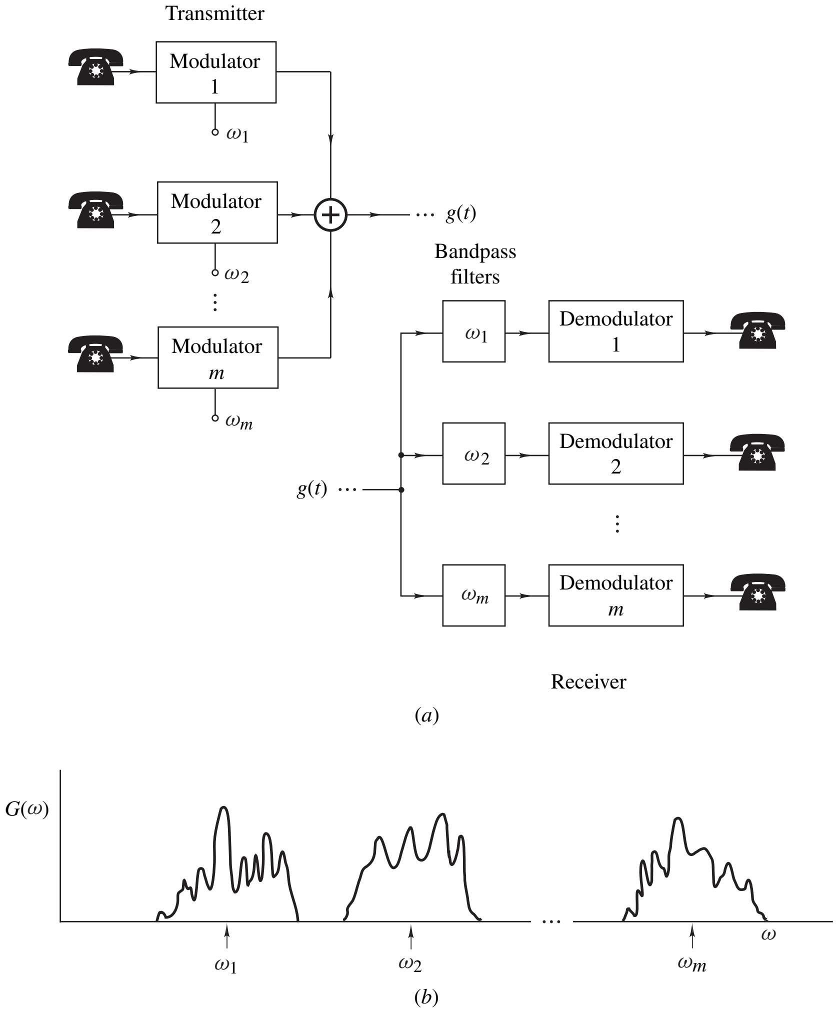


Figure 1.10 Frequency-division multiplex communications system: (a) Basic system, (b) frequency spectrum of $g(t)$.

version of this type of system is illustrated in Fig. 1.10a. The operation of an FDM communications system is as follows:

1. At the transmit end, the different voice signals are superimposed on different carrier frequencies using a process known as modulation.
2. The different carrier frequencies are combined by using an adder circuit.
3. At the receive end, carrier frequencies are separated using bandpass filters.
4. The voice signals are then extracted from the carrier frequencies through demodulation.
5. The voice signals are distributed to the appropriate persons through the local telephone wires.

What the transmit section does in the above system is to add the frequency of a unique carrier to the frequencies of each voice signal, thereby shifting its frequency spectrum by the frequency of the carrier. In this way, the frequency spectrums of the different voice signals are arranged contiguously one after the other to form the composite signal $g(t)$ which is referred to as a *group* by telephone engineers. The frequency spectrum of $g(t)$ is illustrated in Fig. 1.10*b*. The receive section, on the other hand, separates the translated voice signals and restores their original spectrums.

As can be seen in Fig. 1.10*a*, the above system requires as many bandpass filters as there are voice signals. On top of that, there are as many modulators and demodulators in the system and these devices, in their turn, need a certain amount of filtering to achieve their proper operation. In short, communications systems are simply not feasible without filters.

Incidentally, several groups can be further modulated individually and added to form a *supergroup* as illustrated in Fig. 1.11 to increase the number of voice signals transmitted over an intercity cable or microwave link, for example. At the receiving end, a supergroup is subdivided into the individual groups by a bank of bandpass filters which are then, in turn, subdivided into the individual voice signals by appropriate banks of bandpass filters. Similarly, several supergroups can be combined into a *master group*, and so on, until the bandwidth capacity of the cable or microwave link is completely filled.

An important principle to be followed when designing a sampling system like the one illustrated in Fig. 1.2 is that the sampling frequency be at least twice the highest frequency present in the spectrum of the signal by virtue of the sampling theorem (see Chap. 6). In situations where the sampling frequency is fixed and the highest frequency present in the signal can exceed half

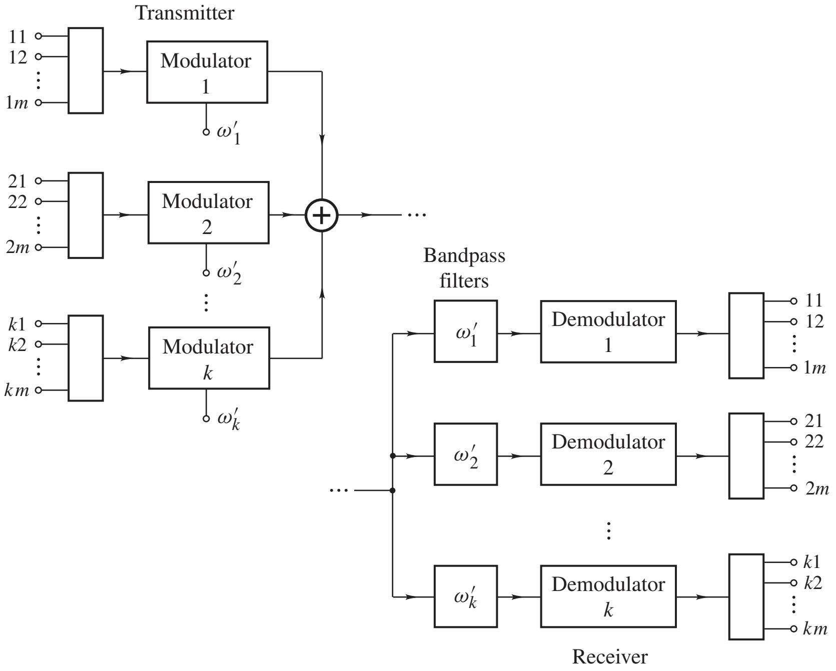


Figure 1.11 Frequency-division multiplex communications system with two levels of modulation.

the sampling frequency, it is crucial to *bandlimit* the signal to be sampled to prevent a certain type of signal distortion known as *aliasing*. This bandlimiting process, which amounts to removing signal components whose frequencies exceed half the sampling frequency, can be carried out through the use of a lowpass filter.

Discrete-time signals are often converted back to continuous-time signals. For example, the signal recorded on a compact disk (CD) is actually a discrete-time signal. The function of a CD player is to reverse the sampling process illustrated in Fig. 1.2, that is, it must read the discrete-time signal, decode it, and reproduce the original continuous-time audio signal. As will be shown later on in Chap. 6, the continuous-time signal can be reconstructed through the use of a lowpass filter.

Loudspeaker systems behave very much like filters and, consequently, they tend to change the spectrum of an audio signal. This is due to the fact that the enclosure or cabinet used can often exhibit mechanical resonances that are superimposed on the audio signal. In fact, this is one of the reasons why different makes of loudspeaker systems often produce their own distinct sound which, in actual fact, is different from the sound recorded on the CD. To compensate for such imperfections, sound reproduction equipment, such as CD players and stereos, are often equipped with *equalizers* that can be used to reshape the spectrum of the audio signal. These subsystems typically incorporate a number of sliders that can be adjusted to modify the quality of the sound reproduced. One can, for example, strengthen or weaken the low-frequency or high-frequency content (bass or treble) of the audio signal. Since an equalizer is a device that can modify the spectrum of a signal, equalizers are filters in the broader definition adopted earlier. What the sliders do is to alter the parameters of the filter that performs the equalization. In the same way, one can also compensate for the acoustics of the room. For example, one might need to boost the treble a bit if there is a thick carpet in the room because the carpet could absorb a large amount of the high-frequency content.

Transmission lines, telephone wires, and communication channels often behave very much like filters and, as a result, they tend to reshape the spectrums of the signals transmitted through them. The local telephone lines are particularly notorious in this respect. We often do not recognize the voice of the person at the other end only because the spectrum of the signal has been significantly altered. As in loudspeaker systems, the quality of transmission through communication channels can be improved by using suitable equalizers. In fact, it is through the use of equalizers that it is possible to achieve high data transmission rates through local telephone lines. This is achieved by incorporating sophisticated equalizers in the modems at either end of a telephone line.

1.8 DIGITAL FILTERS

In its most general form, *a digital filter is a system that will receive an input in the form of a discrete-time signal and produce an output again in the form of a discrete-time signal,* as illustrated in Fig. 1.12. There are many types of discrete-time systems that fall under this category such as digital control systems, encoders, and decoders. What differentiates digital filters from other digital systems is the nature of the processing involved. As in analog filters, there is a requirement that the spectrum of the output signal be related to that of the input by some rule of correspondence.

The roots of digital filters go back in history to the 1600s when mathematicians, on the one hand, were attempting to deduce formulas for the areas of different geometrical shapes, and astronomers, on the other, were attempting to rationalize and interpret their measurements of planetary orbits. A need arose in those days for a process that could be used to interpolate a function represented by numerical data, and a wide range of numerical interpolation formulas were proposed over the

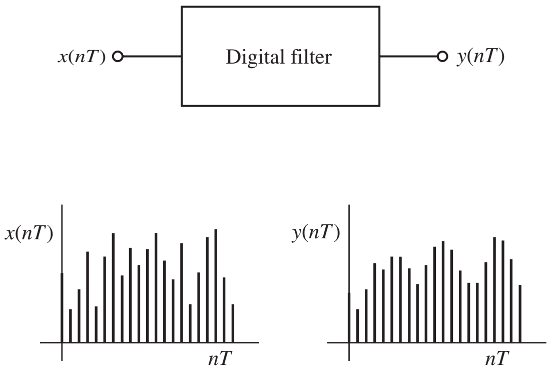


Figure 1.12 The digital filter as a discrete-time system.

years by Gregory (1638–1675), Newton (1642–1727), Taylor (1685–1731), Stirling (1692–1770), Lagrange (1736–1813), Bessel (1784–1846), and others [7, 8]. On the basis of interpolation formulas, formulas that will perform numerical differentiation or integration on a function represented by numerical data can be generated. These formulas were put to good use during the seventeenth and eighteenth centuries in the construction of mathematical, scientific, nautical, astronomical, and a host of other types of numerical tables. In fact, it was the great need for accurate numerical tables that prompted Charles Babbage (1791–1871) to embark on his lifelong quest to automate the computation process through his famous difference and analytical engines [9], and it is on the basis of numerical formulas that his machines were supposed to perform their computations.

Consider the situation where a numerical algorithm is used to compute the derivative of a signal $x(t)$ at $t = t_1, t_2, \ldots, t_K$, and assume that the signal is represented by its numerical values $x(t_1), x(t_2), \ldots, x(t_M)$. In such a situation, the algorithm receives a discrete-time signal as input and produces a discrete-time signal as output, which is a differentiated version of the input signal. Since differentiation is essentially a filtering process, as was demonstrated earlier on, an algorithm that performs numerical differentiation is, in fact, a digital filtering process.

Numerical methods have found their perfect niche in the modern digital computer and considerable progress has been achieved through the fifties and sixties in the development of algorithms that can be used to process signals represented in terms of numerical data. By the late fifties, a cohesive collection of techniques referred to as *data smoothing and prediction* began to emerge through the efforts of pioneers such as Blackman, Bode, Shannon, Tukey [10, 11], and others. During the early sixties, an entity referred to as the *digital filter* began to appear in the literature to describe a collection of algorithms that could be used for spectral analysis and data processing [12–17]. In 1965, Blackman described the state of the art in the area of data smoothing and prediction in his seminal book on the subject [18], and included in this work certain techniques which he referred to as *numerical filtering*. Within a year, in 1966, Kaiser authored a landmark chapter, entitled “Digital Filters” [19] in which he presented a collection of signal processing techniques that could be applied for the simulation of dynamic systems and analog filters. From the late sixties on, the analysis and processing of signals in the form of numerical data became known as *digital signal processing*, and algorithms, computer programs, or systems that could be used for the processing of these signals became fully established as digital filters [20–22].

With the rapid advances in integrated-circuit technology during the sixties, a trend toward digital technologies began to emerge to take advantage of the classical merits of digital systems in general, which are as follows:

- Component tolerances are uncritical.
- Accuracy is high.
- Physical size is small.
- Reliability is high.
- Component drift is relatively unimportant.
- The influence of electrical environmental noise is negligible.

Owing to these important features, digital technologies can be used to design cost-effective, reliable, and versatile systems. Consequently, an uninterrupted evolution, or more appropriately *revolution*, began to take place from the early sixties on whereby analog systems were continuously being replaced by corresponding digital systems. First, the telephone system was digitized through the use of pulse-code modulation, then came long-distance digital communications, and then the music industry adopted digital methodologies through the use of compact disks and digital audio tapes. And more recently, digital radio and high-definition digital TV began to be commercialized. Even the movie industry has already embarked on large-scale digitization of the production of movies.

Digital filters in hardware form began to appear during the late sixties and two early designs were reported by Jackson, Kaiser, and McDonald in 1968 [23] and Peled and Liu in 1974 [24].

Research on digital filters continued through the years and a great variety of filter types have evolved, as follows:

- Nonrecursive filters
- Recursive filters
- Fan filters
- Two-dimensional filters
- Adaptive filters
- Multidimensional filters
- Multirate filters

The applications of digital filters are widespread and include but are not limited to the following:

- Communications systems
- Audio systems such as CD players
- Instrumentation
- Image processing and enhancement
- Processing of seismic and other geophysical signals
- Processing of biological signals
- Artificial cochleas
- Speech synthesis

It is nowadays convenient to consider computer programs and digital hardware that can perform digital filtering as two different implementations of digital filters, namely,

- software
- hardware.

Software digital filters can be implemented in terms of a high-level language, such as C++ or MATLAB, on a personal computer or workstation or by using a low-level language on a general-purpose digital signal-processing chip. At the other extreme, hardware digital filters can be designed using a number of highly specialized interconnected VLSI chips. Both hardware and software digital filters can be used to process real-time or nonreal-time (recorded) signals, except that the former are usually much faster and can deal with real-time signals whose frequency spectrums extend to much higher frequencies. Occasionally, digital filters are used in so-called *quasi-real-time applications* whereby the processing appears to a person to be in real time although, in actual fact, the samples of the signal are first collected and stored in a digital memory and are then retrieved in blocks and processed. A familiar, quasi-real-time application involves the transmission of radio signals over the Internet. These signals are transmitted through data packets in a rather irregular manner. Yet the music appears to be continuous only because the data packets are first stored and then properly sequenced. This is why it takes a little while for the transmission to begin.

Hardware digital filters have an important advantage relative to analog filters, in addition to the classical merits associated with digital systems in general. The parameters of a digital filter are stored in a computer memory and, consequently, they can be easily changed in real time. This means that digital filters are more suitable for applications where programmable, time-variable, or adaptive filters are required. However, they also have certain important limitations. At any instant, say, $t = nT$, a digital filter generates the value of the output signal through a series of computations using some of the values of the input signal and possibly some of the values of the output signal (see Chap. 4). Once the sampling frequency, f_s, is fixed, the sampling period $T = 1/f_s$ is also fixed and, consequently, a basic limitation is imposed by the amount of computation that can be performed by the digital filter during period T. Thus as the sampling frequency is increased, T is reduced, and the amount of computation that can be performed during period T is reduced. Eventually, at some sufficiently high sampling frequency, a digital filter will become computation bound and will malfunction. In effect, digital filters are suitable for low-frequency applications where the operating frequencies are in some range, say, 0 to ω_{max}. The upper frequency of applicability, ω_{max}, is difficult to formalize because it depends on several factors such as the number-crunching capability and speed of the digital hardware on the one hand and the complexity of the filtering tasks involved on the other.

Another basic limitation of digital filters comes into play in situations where the signal is in continuous-time form and a processed version of the signal is required, again in continuous-time form. In such a case, the signal must be converted into a discrete-time form, processed by the digital filter, and then converted back to a continuous-time form. The two conversions involved would necessitate various interfacing devices, e.g., A/D and D/A converters, and a digital-filter solution could become prohibitive relative to an analog-filter solution. This limitation is, of course, absent if we are dealing with a digital system to start with in which the signals to be processed are already in discrete-time form.

Table 1.2 summarizes the frequency range of applicability for the various types of filters [1]. As can be seen, for frequencies less than, say, 20 kHz digital filters are most likely to offer the best

Table 1.2 Comparison of filter technologies

Type of technology	Frequency range
Digital filters	0 to $\omega_{\max}$
Discrete active RC filters	10 Hz to 1 MHz
Switched-capacitor filters	10 Hz to 5 MHz
Passive RLC filters	0.1 MHz to 0.1 GHz
Integrated active RC filters	0.1 MHz to 15 GHz
Microwave filters	0.5 GHz to 500 GHz

engineering solution whereas for frequencies in excess of 0.5 GHz, a microwave filter is the obvious choice. For frequencies between 20 kHz and 0.5 GHz, the choice of filter technology depends on many factors and trade-offs and is also critically dependent on the type of application.

To conclude this section, it should be mentioned that software digital filters have no counterpart in the analog world and, therefore, for nonreal-time applications, they are the only choice.

1.9 TWO DSP APPLICATIONS

In this section, we examine two typical applications of filtering, namely, its use for the processing of an electrocardiogram (EKG), on the one hand, and the processing of stock exchange data, on the other.

1.9.1 Processing of EKG Signals

The EKG of a healthy individual assumes a fairly well-defined form although significant variations can occur from one person to the next as in fingerprints. Yet certain telltale patterns of an EKG enable a cardiologist to diagnose certain cardiac ailments or conditions. An EKG is essentially a graph representing a low-level electrical signal picked up by a pair of electrodes attached to certain well-defined points on the body and connected to an electrical instrument known as the *electrocardiograph*. These machines are used in clinics and hospitals where a multitude of other types of electrical machines are utilized such as x-ray machines and electrical motors. All these machines along with the power lines and transformers that supply them with electricity produce electrical 60-Hz noise, which may contaminate an EKG waveform. A typical noise-free EKG signal is shown in Fig. 1.13a. An EKG signal that has been contaminated by electrical 60-Hz noise is illustrated in Fig. 1.13b. As can be seen, the distinct features of the EKG are all but obliterated in the contaminated signal and are, therefore, difficult, if not impossible, to discern. A diagnosis based on such an EKG would be unreliable.

As electrical noise originating from the power supply has a well-defined frequency, i.e., 60 Hz, one can design a bandstop filter that will reject the electrical noise. Such a filter has been designed using the methods to be studied in later chapters and was then applied to the contaminated EKG signal. The filtered signal is shown in Fig. 1.13c and, as can be seen, apart from some transient artifacts over the interval $n = 0$ to 100, the filtered signal is a faithful reproduction of the original noise-free signal. As another experiment, just to illustrate the nature of filtering, the contaminated

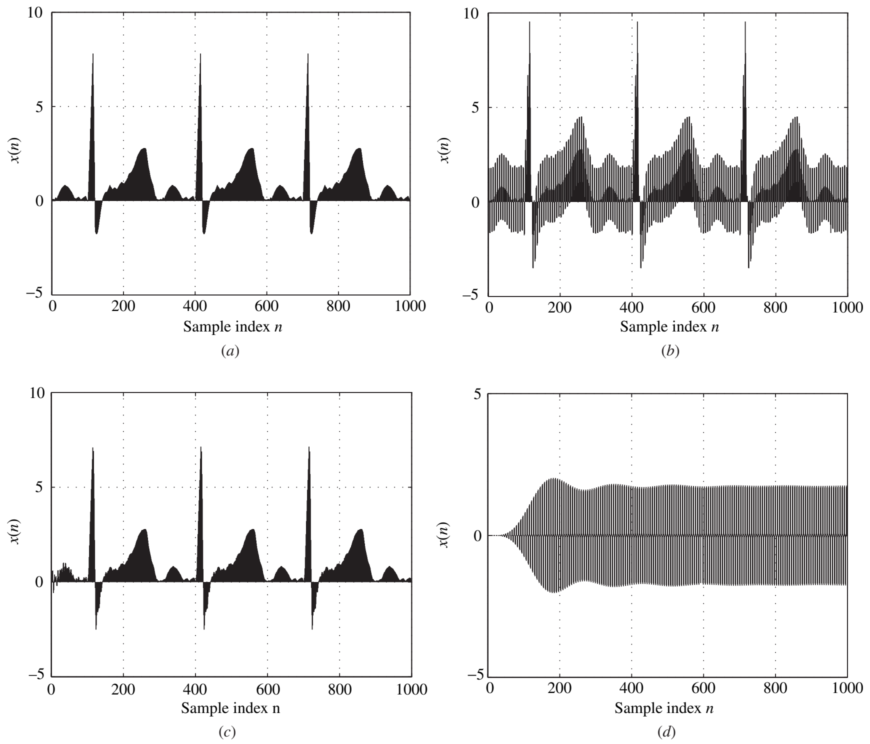


Figure 1.13 Processing of EKG waveform: (*a*) Typical EKG, (*b*) noisy EKG, (*c*) noisy EKG processed with a bandstop filter, (*d*) noisy EKG waveform processed with a bandpass filter.

EKG signal was passed through a bandpass filter which was designed to select the 60-Hz noise component. The output of the bandpass filter is illustrated in Fig. 1.13d. After an initial transience over the interval $n = 0$ to 150, a steady noise component is isolated by the bandpass filter. This is actually a sinusoidal waveform. It does not appear to be so because there are only six samples per cycle with the approximate values of 0, 1.7, 1.7, 0, -1.7, and -1.7.

1.9.2 Processing of Stock-Exchange Data

We are all interested in the health of the market place for various reasons. We would all like, for example, to put aside some funds for another day and, naturally, we would prefer to invest any such funds in secure low-risk stocks, bonds, or mutual funds that provide high returns. To make financial decisions such as these, we read the business section of our daily newspaper or browse the Web for numerical stock-exchange data. Naturally, we would like to make investments that grow steadily from year to year at a steady rate and never devalue. However, this is not what happens in real life. The prices of stocks change rapidly with time and once in a while, for example, when a market recession occurs, they can actually lose a large proportion of their values.

Typically, there are many economic forces that cause the value of a stock to change. Some of these forces are of short duration while others reflect long-term economic pressures. As long-term investors, we should perhaps ignore the day-to-day variations and focus as far as possible on the underlying changes in the stock price. An investor with a sharp eye may be able to draw conclusions by simply comparing the available stock-exchange data of two competing stocks. For most of us this is not an easy task. However, through the use of DSP the task can be greatly simplified, as will be demonstrated next.

The price of a company's stock is a signal and, as such, it possesses a spectrum that can be manipulated through filtering. Day-to-day variations in a stock constitute the high-frequency part of the spectrum whereas the underlying trend of the stock is actually the low-frequency part. If we are interested in the long-term behavior of a stock, then perhaps we should filter out the high-frequency part of the spectrum. On the other hand, if we cannot tolerate large day-to-day variations, then, perhaps we should attempt to check the volatility of the stock. Measures of volatility are readily available, for example, the variance of a stock. Another way to ascertain the volatility would be to remove the low-frequency and retain the high-frequency content of a stock through a highpass filter.

To illustrate these ideas, two actual mutual funds, a bond fund and a high-tech fund, were chosen at random for processing. One year's worth of data were chosen for processing and to facilitate the comparison, the scaled share prices of the two funds were normalized to unity at the start of the year. The normalized share prices of the two funds are plotted in Fig. 1.14*a*. As can be seen, the bond fund has remained rather stable throughout the year, as may be expected, whereas the high-tech one was subjected to large variations. The day-to-day variations, i.e., the high-frequency content, in the two mutual funds can be eliminated through the use of a lowpass filter to obtain the smooth curves shown in Fig. 1.14*b*. On the other hand, the underlying trend of the mutual fund or the low-frequency spectrum can be removed through the use of a highpass filter to obtain the high-frequency content shown in Fig. 1.14*c*. In this figure, the filter output is depicted as a percentage of the unit value.

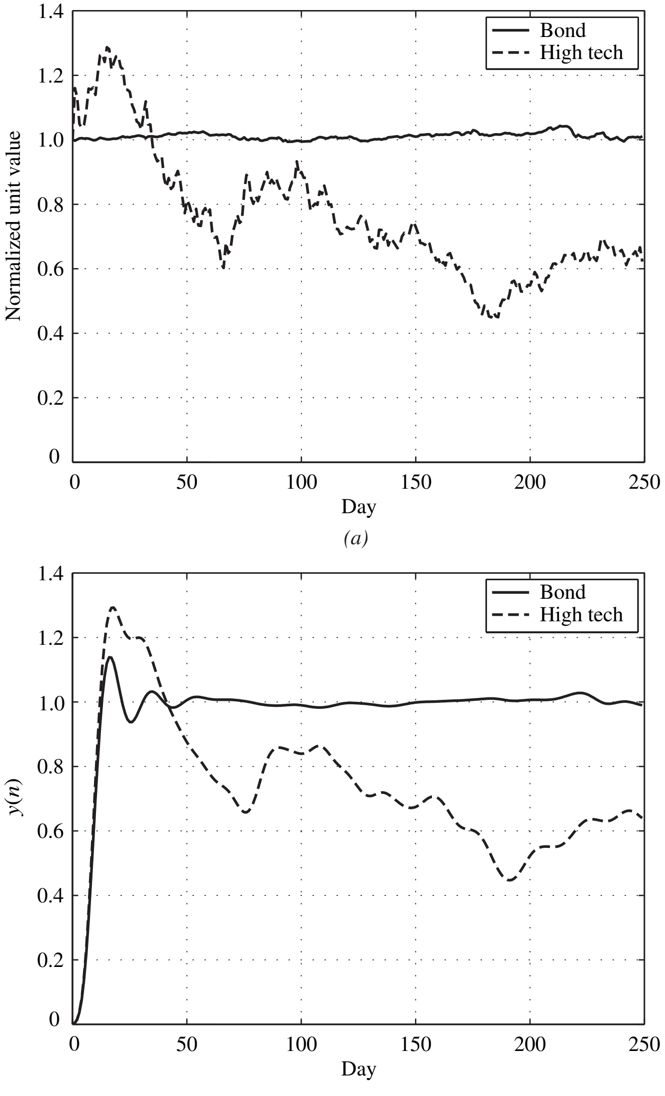
In the plots obtained, certain anomalies are observed during the first 50 or so sample values. These are due to certain initial transient conditions that exist in all types of systems including filters, which will be explained in Chap. 4, but they can be avoided in practice by using a suitable initialization. Ignoring this initial phase, we note that the lowpass-filtered version of the data shown in Fig. 1.14*b* provides a less cluttered view of the funds whereas Fig. 1.14*c* gives a much clearer picture of their relative volatilities. In this respect, note the 5 to 1 difference in the scale of the y axis between the two funds.

Quantitative measures of volatility analogous to the variance of a stock can also be deduced from plots like those in Fig. 1.14*c*. One could, for example, obtain the mean-square average (MSA) of $y(n)$ which is defined as

$$\text{MSA} = \frac{1}{N}\sum_{n=1}^{N}[y(n)]^2$$

or the average of $|y(n)|$ or some other norm. The value of the MSA for the bond and high-tech funds for values of $50 \leq n \leq 250$ can be readily computed as 0.0214 and 1.2367, respectively, i.e., a ratio of 1 to 57.7 in favor of the bond fund. Evidently, the message is very clear as to what type of fund one should buy to avoid sleepless nights.

Another intriguing possibility that deserves a mention is the use of extrapolating filters. Filters of this type can be used to predict tomorrow's stock prices but if we pursue the subject any further, we will find ourselves in the domain of what is known in the business world as *technical analysis.*



(a)

(b)

Figure 1.14 Processing of stock-exchange data: (*a*) Unit values of bond and high technology mutual funds, (*b*) data processed with a lowpass filter.

REFERENCES

[1] R. Schaumann and M. E. Van Valkenburg, *Design of Analog Filters,* New York: Oxford University Press, 2001.

[2] K. L. Su, *Analog Filters,* London: Chapman & Hall, 1996.

[3] C. Rauscher, "Two-branch microwave channelized active bandpass filters," *IEEE Trans. Microwave Theory Tech.,* vol. MTT-48, pp. 437–444, Mar. 2000.

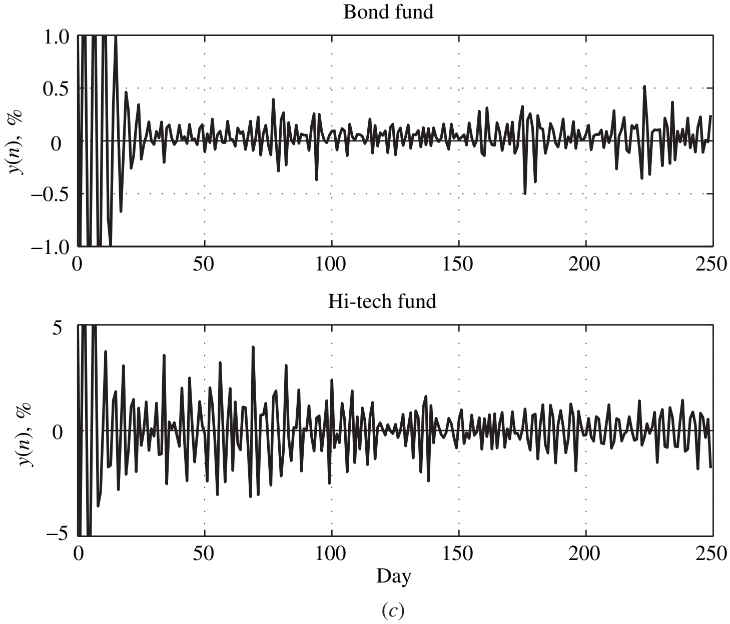


Figure 1.14 Cont'd (*c*) Processing of stock-exchange data: Data processed with a highpass filter.

[4] C.-H. Lee, S. Han, and J. Laskar, "GaAs MESFET dual-gate mixer with active filter design for Ku-band applications," *IEEE Radio Frequency Integrated Circuits Symposium,* pp. 203–206, 1999.

[5] I. C. Hunter, *Theory and Design of Microwave Filters,* London: The Institution of Electrical Engineers, 2001.

[6] B. P. Lathi, *Modern Digital and Analog Communication Systems,* New York: Holt, Reinhart and Winston, 1983.

[7] R. Butler and E. Kerr, *An Introduction to Numerical Methods,* London: Pitman, 1962.

[8] C.-E. Fröberg, *Introduction to Numerical Analysis,* 2nd ed., Reading, MA: Addison-Wesley, 1969.

[9] D. D. Swade, "Redeeming Charles Babbage's mechanical computer," *Scientific American*, vol. 268, pp. 86–91, Feb. 1993.

[10] R. B. Blackman, H. W. Bode, and C. E. Shannon, "Data smoothing and prediction in fire-control systems," *Summary Technical Report of Division 7,* NDRC, vol. 1, pp. 71–160. Reprinted as Report Series, MGC 12/1 (August 15, 1948), National Military Establishment, Research and Development Board.

[11] R. B. Blackman and J. W. Tukey, *The Measurement of Power Spectra from the Point of View of Communications Engineering,* New York: Dover, 1959.

[12] M. A. Martin, *Digital Filters for Data Processing,* General Electric Co., Missile and Space Division, Tech. Inf. Series Report No. 62-SD484, 1962.

[13] K. Steiglitz, *The General Theory of Digital Filters with Applications to Spectral Analysis,* AFOSR Report no. 64–1664, New York University, New York, May 1963.

[14] E. B. Anders et al., *Digital Filters,* NASA Contractor Report CR-136, Dec. 1964.

[15] H. H. Robertson, “Approximate design of digital filters,” *Technometrics*, vol. 7, pp. 387–403, Aug. 1965.

[16] J. F. Kaiser, “Some practical considerations in the realization of digital filters,” *Proc. Third Allerton Conf. on Circuits and Systems,* pp. 621–633, Oct. 1965.

[17] K. Steiglitz, “The equivalence of digital and analog signal processing,” *Information and Control,* vol. 8, pp. 455–467, Oct. 1965.

[18] R. B. Blackman, *Data Smoothing and Prediction,* Reading, MA: Addison-Wesley, 1965.

[19] F. F. Kuo and J. F. Kaiser, *System Analysis by Digital Computer,* New York: Wiley, 1966.

[20] B. Gold and C. M. Rader, *Digital Signal Processing*, New York: McGraw-Hill, 1969.

[21] R. E. Bogner and A. G. Constantinides (eds.), *Introduction to Digital Filtering*, New York: Wiley, 1975.

[22] A. Antoniou, *Digital Filters: Analysis and Design*, New York: McGraw-Hill, 1979.

[23] L. B. Jackson, J. F. Kaiser, and H. S. McDonald, “An approach to the implementation of digital filters,” *IEEE Trans. Audio and Electroacoust.,* vol. 16, pp. 413–421, Sept. 1968.

[24] A. Peled and B. Liu, “A new hardware realization of digital filters,” *IEEE Trans. Acoust. Speech, Signal Process.,* vol. 22, pp. 456–462, Dec. 1974.

CHAPTER 2

THE FOURIER SERIES AND FOURIER TRANSFORM

2.1 INTRODUCTION

Spectral analysis has been introduced in a heuristic way in Chap. 1. In the present chapter, the spectral analysis of continuous-time signals is developed further. The basic mathematical tools required for the job, namely, the *Fourier series* and the *Fourier transform*, are described in some detail. The Fourier series, which provides spectral representations for periodic continuous-time signals, is treated first. Then the Fourier transform is derived by applying a limiting process to the Fourier series. The properties of the Fourier series and the Fourier transform are delineated through a number of theorems. The chapter also deals with the application of the Fourier series and Fourier transform to a variety of standard continuous-time signals.

The reader may question the extent of the treatment of the spectral representation of continuous-time signals in a book that claims to deal with DSP. However, as was emphasized in Chap. 1, most of the signals occurring in nature are essentially continuous in time, and it is, therefore, reasonable to expect the spectrums of discrete-time signals to be closely related to those of the continuous-time signals from which they are derived. This indeed is the case, as will be shown in Chaps. 3, 6, and 7.

2.2 FOURIER SERIES

In Chap. 1, the concept of frequency spectrum of a signal was introduced as an alternative to time-domain representation. As was demonstrated, a periodic signal that comprises a weighted sum of sinusoids such as that in Eq. (1.1) can be represented completely in the frequency domain in terms of the amplitudes and phase angles of its individual sinusoidal components. Below, we demonstrate that through the use of the Fourier[1] series, the concept of frequency spectrum can be applied to arbitrary periodic signals.

In the next and subsequent sections, periodic signals are typically represented in terms of nonperiodic signals. To avoid possible confusion between the two types of signals we will use the notation $\tilde{x}(t)$ for a periodic signal and simply $x(t)$ for a nonperiodic one. Signals will be assumed to be real unless otherwise stated.

2.2.1 Definition

A periodic continuous-time signal, namely, a signal that satisfies the condition

$$\tilde{x}(t + r\tau_0) = \tilde{x}(t) \qquad \text{for } |r| = 1, 2, \ldots, \infty$$

where τ_0 is a constant called *the period* of the signal, can be expressed as

$$\tilde{x}(t) = \sum_{r=-\infty}^{\infty} x(t + r\tau_0) \tag{2.1}$$

where $x(t)$ is a nonperiodic signal given by

$$x(t) = \begin{cases} \tilde{x}(t) & \text{for } -\tau_0/2 < t \le \tau_0/2 \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

The time interval $-\tau_0/2 < t \le \tau_0/2$ will be referred to as the *base period* hereafter.

What the above formulas are saying is this: If a nonperiodic signal $x(t)$ is available that fully describes the periodic signal $\tilde{x}(t)$ with respect to the base period, then the periodic signal $\tilde{x}(t)$ can be generated by creating time-shifted copies of $x(t)$, that is, $x(t + r\tau_0)$ for $r = 1, 2, \ldots, \infty$ and $r = -1, -2, \ldots, -\infty$, and then adding them up. This replication process occurs frequently in DSP and it will be referred to as *periodic continuation*, that is, $\tilde{x}(t)$ is the periodic continuation of $x(t)$ in the present context.

A periodic signal $\tilde{x}(t)$ that satisfies certain mathematical requirements as detailed in Theorem 2.1 (see Sec. 2.2.3) can be represented by the Fourier series (see Chap. 7 of Ref. [1] or Chap. 5 of Ref. [2]). The most general form of this important representation is given by

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t} \qquad \text{for } -\tau_0/2 \le t \le \tau_0/2 \tag{2.3}$$

[1] *Jean Baptiste Joseph Fourier (1768–1830) was a French mathematician who was taught by Lagrange and Laplace. He got himself involved with the French Revolution and in due course joined Napoleon's army in the invasion of Egypt as a scientific advisor. The series named after him emerged while Fourier was studying the propagation of heat in solid bodies after his return from Egypt.*

where $\omega_0 = 2\pi/\tau_0$. The coefficients[2] $\{X_k\}$ can be deduced by multiplying both sides of Eq. (2.3) by $e^{-jl\omega_0 t}$ and then integrating over the base period $-\tau_0/2 < t \le \tau_0/2$. Thus

$$\int_{-\tau_0/2}^{\tau_0/2} \tilde{x}(t)e^{-jl\omega_0 t}\,dt = \int_{-\tau_0/2}^{\tau_0/2} \sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t} e^{-jl\omega_0 t}\,dt$$

$$= \sum_{k=-\infty}^{\infty} X_k \int_{-\tau_0/2}^{\tau_0/2} e^{j(k-l)\omega_0 t}\,dt$$

The change in the order of integration and summation between the first and second equations is allowed for signals that satisfy the conditions in Theorem 2.1. Now

$$\int_{-\tau_0/2}^{\tau_0/2} e^{j(k-l)\omega_0 t}\,dt = \begin{cases} \tau_0 & \text{if } l = k \\ 0 & \text{if } l \neq k \end{cases} \tag{2.4}$$

(see Prob. 2.1) and as $\tilde{x}(t) = x(t)$ over the base period $-\tau_0/2 < t \le \tau_0/2$ according to Eq. (2.2), we have

$$X_k = \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t)e^{-jk\omega_0 t}\,dt \tag{2.5}$$

As $e^{jk\omega_0 t}$ is a complex quantity, coefficients $\{X_k\}$ are complex in general but can be purely real or purely imaginary.[3] X_k can be represented in terms of its real and imaginary parts or its magnitude and angle as

$$X_k = \Re e\, X_k + j\, \Im m\, X_k = |X_k| e^{j \arg X_k}$$

where

$$|X_k| = \sqrt{(\Re e\, X_k)^2 + (\Im m\, X_k)^2} \qquad \text{and} \qquad \arg X_k = \tan^{-1} \frac{\Im m\, X_k}{\Re e\, X_k}$$

2.2.2 Particular Forms

The Fourier series can also be expressed in terms of a sum of sines and cosines, or just sines or just cosines, as will now be demonstrated. Equation (2.3) can be expanded into a sum of two series plus a constant as

$$\tilde{x}(t) = \sum_{k=-\infty}^{-1} X_k e^{jk\omega_0 t} + X_0 + \sum_{k=1}^{\infty} X_k e^{jk\omega_0 t}$$

and by letting[4] $k \to -k$ in the first summation and then noting that

$$\sum_{k=\infty}^{1} X_{-k} e^{-jk\omega_0 t} \equiv \sum_{k=1}^{\infty} X_{-k} e^{-jk\omega_0 t}$$

[2] *The notation $\{X_k\}$ is used to represent the set of coefficients X_k for $-\infty \le k \le \infty$, which can also be represented more precisely by the notation $\{X_k\colon\ -\infty \le k \le \infty\}$.*

[3] *See Appendix A for the basic principles of complex analysis.*

[4] *The notation $k \to -k$ here and in the subsequent chapters represents two variable transformations carried out in sequence one after the other, that is, $k = -k'$ and $k' = k$.*

we get

$$
\begin{aligned}
\tilde{x}(t) &= \sum_{k=1}^{\infty} X_{-k} e^{-jk\omega_0 t} + X_0 + \sum_{k=1}^{\infty} X_k e^{jk\omega_0 t} \\
&= X_0 + \sum_{k=1}^{\infty} X_{-k}(\cos k\omega_0 t - j \sin k\omega_0 t) \\
&\quad + \sum_{k=1}^{\infty} X_k(\cos k\omega_0 t + j \sin k\omega_0 t) \\
&= X_0 + \sum_{k=1}^{\infty}(X_k + X_{-k}) \cos k\omega_0 t + \sum_{k=1}^{\infty} j(X_k - X_{-k}) \sin k\omega_0 t
\end{aligned} \tag{2.6}
$$

Now from Eq. (2.5), we have

$$
X_0 = \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t)\, dt \tag{2.7a}
$$

$$
\begin{aligned}
X_k &= \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) e^{-jk\omega_0 t}\, dt \\
&= \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t)(\cos k\omega_0 t - j \sin k\omega_0 t)\, dt \\
&= \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) \cos k\omega_0 t\, dt - j \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) \sin k\omega_0 t\, dt
\end{aligned} \tag{2.7b}
$$

$$
\begin{aligned}
X_{-k} &= \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) e^{jk\omega_0 t}\, dt \\
&= \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t)(\cos k\omega_0 t + j \sin k\omega_0 t)\, dt \\
&= \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) \cos k\omega_0 t\, dt + j \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) \sin k\omega_0 t\, dt
\end{aligned} \tag{2.7c}
$$

and hence Eqs. (2.7b) and (2.7c) give

$$
X_k + X_{-k} = \frac{2}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) \cos k\omega_0 t\, dt \tag{2.8a}
$$

$$
j(X_k - X_{-k}) = \frac{2}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) \sin k\omega_0 t\, dt \tag{2.8b}
$$

$$
X_{-k} = X_k^* \tag{2.8c}
$$

where X_k^* is the complex conjugate of X_k. On using Eqs. (2.7a), (2.8a), and (2.8b), Eq. (2.6) can also be expressed as

$$\tilde{x}(t) = \tfrac{1}{2}a_0 + \sum_{k=1}^{\infty}(a_k \cos k\omega_0 t + b_k \sin k\omega_0 t) \tag{2.9}$$

where

$$a_0 = 2X_0 = \frac{2}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} x(t)\,dt \tag{2.10a}$$

$$a_k = X_k + X_{-k} = \frac{2}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} x(t)\cos k\omega_0 t\,dt \tag{2.10b}$$

$$b_k = j(X_k - X_{-k}) = \frac{2}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} x(t)\sin k\omega_0 t\,dt \tag{2.10c}$$

The 1/2 in the constant term of Eq. (2.9) is used to make the formula for a_0 in Eq. (2.10a) a special case of the formula for a_k in Eq. (2.10b). These equations are often referred to as the *Euler* or *Euler-Fourier formulas*.

Equation (2.9) gives the Fourier series in terms of sines and cosines. As sines can be converted to cosines and vice versa, a representation of the Fourier series in terms of just sines or just cosines can be readily obtained. If we let

$$a_k = A_k \cos\phi_k \qquad \text{and} \qquad b_k = -A_k \sin\phi_k \tag{2.11}$$

then parameters $\{A_k\}$ and $\{\phi_k\}$ can be expressed in terms of $\{a_k\}$ and $\{b_k\}$ or $\{X_k\}$ as

$$A_0 = |a_0| = 2|X_0| \tag{2.12a}$$

$$\phi_0 = \begin{cases} 0 & \text{if } a_0 \text{ or } X_0 \geq 0 \\ -\pi & \text{if } a_0 \text{ or } X_0 < 0 \end{cases} \tag{2.12b}$$

$$A_k = \sqrt{a_k^2 + b_k^2} = 2|X_k| \tag{2.12c}$$

$$\phi_k = \tan^{-1}\left(-\frac{b_k}{a_k}\right) = \arg X_k \tag{2.12d}$$

(see Prob. 2.2). Now on eliminating coefficients a_0, a_k, and b_k in Eq. (2.9) using Eq. (2.11), the Fourier series can be put in the form

$$\begin{aligned}\tilde{x}(t) &= \tfrac{1}{2}A_0\cos\phi_0 + \sum_{k=1}^{\infty} A_k(\cos\phi_k \cos k\omega_0 t - \sin\phi_k \sin k\omega_0 t) \\ &= \tfrac{1}{2}A_0\cos\phi_0 + \sum_{k=1}^{\infty} A_k(\cos k\omega_0 t\cos\phi_k - \sin k\omega_0 t \sin\phi_k) \\ &= \tfrac{1}{2}A_0\cos\phi_0 + \sum_{k=1}^{\infty} A_k\cos(k\omega_0 t + \phi_k) & (2.13a) \\ &= \tfrac{1}{2}A_0\sin(\phi_0 + \tfrac{1}{2}\pi) + \sum_{k=1}^{\infty} A_k \sin\left(k\omega_0 t + \phi_k + \tfrac{1}{2}\pi\right) & (2.13b)\end{aligned}$$

In summary, *the Fourier series can be used to express a periodic signal in terms of an infinite linear combination of exponentials as in Eq. (2.3), in terms of sines and cosines as in Eq. (2.9), just cosines as in Eq. (2.13a), or just sines as in Eq. (2.13b).* Engineers often refer to the sinusoidal component of frequency ω_0 as the *fundamental* and to those of frequencies $k\omega_0$ for $k = 2, 3, \ldots,$ as the *harmonics.* The terms $\frac{1}{2}a_0$ in Eq. (2.9), $\frac{1}{2}A_0 \cos \phi_0$ in Eq. (2.13a), and $\frac{1}{2}A_0 \sin(\phi_0 + \pi/2)$ in Eq. (2.13b) are alternative ways of representing the zero frequency component and can assume positive or negative values.

The set of coefficients $\{X_k\colon -\infty \le k \le \infty\}$ in Eq. (2.5), the sets of coefficients $\{a_k\}$ and $\{b_k\}$ in Eq. (2.9), and the corresponding amplitudes and phase angles of the sinusoids in Eq. (2.13b), that is, $\{A_k\colon\ 0 \le k \le \infty\}$ and $\{\phi_k\colon\ 0 \le k \le \infty\}$, respectively, constitute alternative but complete descriptions of the*frequency spectrum* of a periodic signal. The coefficients $\{X_k\}$ are closely related to the Fourier transform of the nonperiodic signal $x(t)$, as will be demonstrated in Sec. 2.3.1, and, for this reason, they will receive preferential treatment in this book, although the alternative representations in terms of $\{a_k\}$ and $\{b_k\}$ or $\{A_k\}$ and $\{\phi_k\}$ will also be used once in a while. The magnitude and phase angle of X_k, that is, $|X_k|$ and $\arg X_k$, viewed as functions of the discrete frequency variable $k\omega_0$ for $-\infty < k\omega_0 < \infty$, will henceforth be referred to as the *amplitude spectrum* and *phase spectrum*, respectively.

The periodic signal $\tilde{x}(t)$ in the above analysis can be symmetrical or antisymmetrical with respect to the vertical axis. If it is symmetrical, signal $x(t)$ in Eq. (2.2) is an even function of time since

$$x(-t) = x(t)$$

We know that $\cos \omega t$ is an even and $\sin \omega t$ is an odd function of time, that is,

$$\cos(-\omega t) = \cos \omega t \qquad \text{and} \qquad \sin(-\omega t) = -(\sin \omega t)$$

and hence $x(t) \cos k\omega_0 t$ is even and $x(t) \sin k\omega_0 t$ is odd. Consequently, Eqs. (2.7a)–(2.7c) give

$$X_0 = \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t)\, dt = \frac{2}{\tau_0} \int_0^{\tau_0/2} x(t)\, dt \tag{2.14a}$$

$$\begin{aligned} X_k &= \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) \cos k\omega_0 t\, dt - j\frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) \sin k\omega_0 t\, dt \\ &= \frac{2}{\tau_0} \int_0^{\tau_0/2} x(t) \cos k\omega_0 t\, dt \qquad \text{for } k = 1, 2, \ldots \end{aligned} \tag{2.14b}$$

$$\begin{aligned} X_{-k} &= \frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) \cos k\omega_0 t\, dt + j\frac{1}{\tau_0} \int_{-\tau_0/2}^{\tau_0/2} x(t) \sin k\omega_0 t\, dt \\ &= \frac{2}{\tau_0} \int_0^{\tau_0/2} x(t) \cos k\omega_0 t\, dt \qquad \text{for } k = 1, 2, \ldots \end{aligned} \tag{2.14c}$$

that is, $$X_{-k} = X_k \qquad \text{for } k = 1, 2, \ldots$$

and from Eqs. (2.10a)–(2.10c), we get

$$a_0 = 2X_0 = \frac{4}{\tau_0}\int_0^{\tau_0/2} x(t)\,dt \tag{2.15a}$$

$$a_k = X_k + X_{-k} = \frac{4}{\tau_0}\int_0^{\tau_0/2} x(t)\cos k\omega_0 t\,dt \qquad \text{for } k = 1, 2, \ldots \tag{2.15b}$$

$$b_k = j(X_k - X_{-k}) = 0 \qquad \text{for } k = 1, 2, \ldots \tag{2.15c}$$

On the other hand, if $\tilde{x}(t)$ is antisymmetrical about the vertical axis, then $x(t)$ is an odd function and thus $x(t)\cos k\omega_0 t$ is an odd function and $x(t)\sin k\omega_0 t$ is an even function. In this case, Eqs. (2.7a)–(2.7c) give

$$X_0 = \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} x(t)\,dt = 0 \tag{2.16a}$$

$$\begin{aligned} X_k &= \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} x(t)\cos k\omega_0 t\,dt - j\frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} x(t)\sin k\omega_0 t\,dt \\ &= -j\frac{2}{\tau_0}\int_0^{\tau_0/2} x(t)\sin k\omega_0 t\,dt \qquad \text{for } k = 1, 2, \ldots \end{aligned} \tag{2.16b}$$

$$\begin{aligned} X_{-k} &= \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} x(t)\cos k\omega_0 t\,dt + j\frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} x(t)\sin k\omega_0 t\,dt \\ &= j\frac{2}{\tau_0}\int_0^{\tau_0/2} x(t)\sin k\omega_0 t\,dt \qquad \text{for } k = 1, 2, \ldots \end{aligned} \tag{2.16c}$$

that is,

$$X_{-k} = -X_k \qquad \text{for } k = 1, 2, \ldots$$

and from Eqs. (2.10a)–(2.10c), we get

$$a_0 = 2X_0 = 0 \tag{2.17a}$$

$$a_k = X_k + X_{-k} = 0 \qquad \text{for } k = 1, 2, \ldots \tag{2.17b}$$

$$b_k = j(X_k - X_{-k}) = 2jX_k \tag{2.17c}$$

$$= \frac{4}{\tau_0}\int_0^{\tau_0/2} x(t)\sin k\omega_0 t\,dt \qquad \text{for } k = 1, 2, \ldots \tag{2.17d}$$

In effect, if $x(t)$ is antisymmetrical, then the DC component, which is the average value of the waveform, is zero.

2.2.3 Theorems and Properties

Fourier series have certain theoretical properties that are often of considerable practical interest. A few of the most important ones are described below in terms of a number of theorems.

To start with, we are quite interested in the circumstances under which the substitution of the coefficients given by Eq. (2.5) in the Fourier series of Eq. (2.3) would yield the periodic signal $\tilde{x}(t)$.

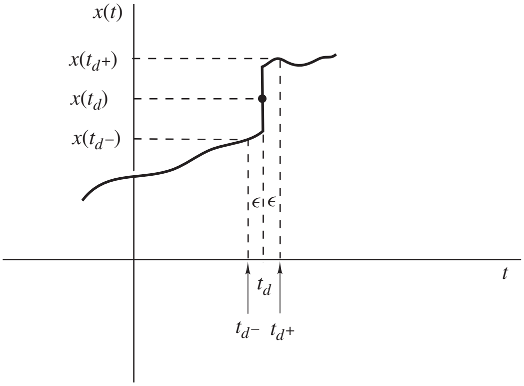


Figure 2.1 A signal $x(t)$ with a discontinuity.

Theorem 2.1 Convergence *If $\tilde{x}(t)$ is a periodic signal of the form*

$$\tilde{x}(t) = \sum_{r=-\infty}^{\infty} x(t + r\tau_0)$$

where $x(t)$ is defined by Eq. (2.2), and over the base period $-\tau_0/2 < t \le \tau_0/2$ $x(t)$

- *has a finite number of local maxima and minima*
- *has a finite number of points of discontinuity*
- *is bounded, that is,*

$$|x(t)| \le K < \infty$$

for some positive K, then the substitution of coefficients $\{X_k\}$ given by Eq. (2.5) in the Fourier series of Eq. (2.3) converges to $\tilde{x}(t)$ at all points where $x(t)$ is continuous.

At points where $x(t)$ is discontinuous, the Fourier series converges to the average of the left- and right-hand limits of $x(t)$, namely,

$$x(t_d) = \tfrac{1}{2}[x(t_d-) + x(t_d+)]$$

as illustrated in Fig. 2.1 where the left- and right-hand limits of $x(t)$ at $t = t_d$ are defined as

$$x(t_d-) = \lim_{\epsilon \to 0} x(t_d - |\epsilon|) \qquad \text{and} \qquad x(t_d+) = \lim_{\epsilon \to 0} x(t_d + |\epsilon|)$$

Proof (See pp. 225–232 of Ref. [3] for proof.) ▲

The prerequisite conditions for convergence as stated in Theorem 2.1 are known as the *Dirichlet*[5] *conditions (see Ref. [4]).*

[5] *Johann Peter Gustave Lejeune Dirichlet (1805–1859) was born in Düren, a town between Aachen and Cologne. In addition to his work on the Fourier series, he contributed a great deal to differential equations and number theory. He married one of the two sisters of the composer Felix Mendelssohn.*

In the above analysis, we have tacitly assumed that the periodic signal $\tilde{x}(t)$ is real. Nevertheless, the Fourier series is applicable to complex signals just as well. Furthermore, the variable need not be time. In fact, the Fourier series is often used to design certain types of digital filters, as will be demonstrated in Chap. 9, and in that application a function is used, which is periodic with respect to frequency, that is, the roles of time and frequency are interchanged. In the following theorem, signal $\tilde{x}(t)$ is deemed to be complex but the theorem is, of course, valid for real signals as well. The theorem provides a relation between the power associated with a periodic signal and the Fourier-series coefficients of the signal.

Theorem 2.2 Parseval's Formula for Periodic Signals ***The mean of the product $\tilde{x}(t)\tilde{x}^*(t)$, where $\tilde{x}^*(t)$ is the complex conjugate of $\tilde{x}(t)$, can be expressed in terms of the Fourier-series coefficients $\{X_k\}$ as***

$$\begin{aligned}\overline{\tilde{x}(t)\tilde{x}^*(t)} &= \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} \tilde{x}(t)\tilde{x}^*(t)\,dt = \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} |\tilde{x}(t)|^2\,dt \\ &= \sum_{k=-\infty}^{\infty} X_k X_k^* = \sum_{k=-\infty}^{\infty} |X_k|^2 \end{aligned} \tag{2.18}$$

See footnote on Parseval[6].

Proof The mean of the product $\tilde{x}(t)\tilde{x}^*(t)$ is defined as

$$\overline{\tilde{x}(t)\tilde{x}^*(t)} = \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} \tilde{x}(t)\tilde{x}^*(t)\,dt \tag{2.19}$$

Hence, Eqs. (2.3) and (2.19) give

$$\begin{aligned}\overline{\tilde{x}(t)\tilde{x}^*(t)} &= \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} \left(\sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t}\right)\left(\sum_{l=-\infty}^{\infty} X_l e^{jl\omega_0 t}\right)^* dt \\ &= \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} \left(\sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t}\right)\left(\sum_{l=-\infty}^{\infty} X_l^* e^{-jl\omega_0 t}\right) dt \\ &= \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} \left(\sum_{k=-\infty}^{\infty} X_k \sum_{l=-\infty}^{\infty} X_l^* e^{j(k-l)\omega_0 t}\right) dt \end{aligned}$$

For signals that satisfy Theorem 2.1, the order of summation and integration can be interchanged and thus

$$\overline{\tilde{x}(t)\tilde{x}^*(t)} = \sum_{k=-\infty}^{\infty} X_k \sum_{l=-\infty}^{\infty} X_l^* \cdot \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} e^{j(k-l)\omega_0 t}\,dt$$

[6] *Marc-Antoine Parseval de Chenes (1755–1836) was a French mathematician of noble birth who lived in Paris during the French Revolution. He published some poetry against Napoleon's regime, which nearly got him arrested.*

Now the value of the integral is equal to τ_0 if $l = k$ and zero otherwise, according to Eq. (2.4). Therefore,

$$\overline{\tilde{x}(t)\tilde{x}^*(t)} = \sum_{k=-\infty}^{\infty} X_k X_k^* = |X_k|^2 \quad \blacktriangle$$

For a real $\tilde{x}(t)$, we have $\tilde{x}^*(t) = \tilde{x}(t)$ and hence Parseval's formula in Eq. (2.18) assumes the simplified form

$$\overline{\tilde{x}^2(t)} = \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} \tilde{x}^2(t)\,dt = \sum_{k=-\infty}^{\infty} X_k X_k^* = \sum_{k=-\infty}^{\infty} |X_k|^2 \tag{2.20}$$

where $\overline{\tilde{x}^2(t)}$ is the mean square value of the periodic signal $\tilde{x}(t)$.

If $\tilde{x}(t)$ represents a voltage across or a current through a resistor then the mean square of $\tilde{x}(t)$ is proportional to the average power delivered to the resistor. In effect, Parseval's theorem provides a formula that can be used to calculate the average power by using the Fourier-series coefficients.

Theorem 2.3 Least-Squares Approximation *A truncated Fourier series for a real periodic signal $\tilde{x}(t)$ of the form*

$$\tilde{x}'(t) = \sum_{k=-N}^{N} X_k e^{jk\omega_0 t} \tag{2.21}$$

is a least-squares approximation of $\tilde{x}(t)$ independently of the value of N.

Proof Let

$$\tilde{y}(t) = \sum_{k=-N}^{N} Y_k e^{jk\omega_0 t} \tag{2.22}$$

be an approximation for $\tilde{x}(t)$ and assume that $e(t)$ is the error incurred. From (2.3) and (2.22), we can write

$$\begin{aligned}\tilde{e}(t) &= \tilde{x}(t) - \tilde{y}(t)\\ &= \sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t} - \sum_{k=-N}^{N} Y_k e^{jk\omega_0 t}\\ &= \sum_{k=-\infty}^{\infty} E_k e^{jk\omega_0 t}\end{aligned} \tag{2.23}$$

where

$$E_k = \begin{cases} X_k - Y_k & \text{for } -N \le k \le N \\ X_k & \text{for } |k| > N \end{cases}$$

On comparing Eq. (2.23) with Eq. (2.3), we conclude that Eq. (2.23) is the Fourier series of the approximation error, $\tilde{e}(t)$, and by virtue of Parseval's theorem (that is, Eq. (2.20)), the mean-square error is given by

$$\overline{\tilde{e}^2(t)} = \sum_{k=-\infty}^{\infty} |E_k|^2$$

$$= \sum_{k=-N}^{N} |X_k - Y_k|^2 + \sum_{|k|>N} |X_k|^2 \tag{2.24}$$

The individual terms at the right-hand side of Eq. (2.24) are all positive and, therefore, $\overline{e^2(t)}$ is minimized if and only if

$$Y_k = X_k \qquad \text{for } -N \le k \le N$$

that is,

$$\tilde{y}(t) = \sum_{k=-N}^{N} Y_k e^{jk\omega_0 t} = \sum_{k=-N}^{N} X_k e^{jk\omega_0 t} = \tilde{x}'(t)$$

That is, the approximation $\tilde{y}(t)$ of $\tilde{x}(t)$ that minimizes the mean-square error incurred is the truncated Fourier series of Eq. (2.21). Such an approximation is said to be a *least-squares approximation.* ▲

Theorem 2.4 Uniqueness ***If two periodic signals $\tilde{x}_1(t)$ and $\tilde{x}_2(t)$ are continuous over the base period and have the same Fourier-series coefficients, that is, $\{X_{k1}\} = \{X_{k2}\}$, then they must be identical, that is, $\tilde{x}_1(t) = \tilde{x}_2(t)$.***

Proof (See p. 487 in Ref. [1] for proof.) ▲

The theorem also applies if the signals have a finite number of discontinuities over the base period provided that the values of $x_1(t)$ or $x_2(t)$ at each discontinuity are defined as the average of the left- and right-hand limits as in Theorem 2.1.

A consequence of the uniqueness property is that an arbitrary linear combination of sines, or cosines, or both, such as Eq. (1.1), for example, is a unique Fourier series of a corresponding unique periodic signal.

The application of the Fourier series will now be illustrated by analyzing some typical periodic waveforms.

Example 2.1 The periodic pulse signal $\tilde{x}(t)$ shown in Fig. 2.2*a* can be represented by Eq. (2.1) with $x(t)$ given by[7]

$$x(t) \equiv p_\tau(t) = \begin{cases} 0 & \text{for } -\tau_0/2 < t < -\tau/2 \\ 1 & \text{for } -\tau/2 \le t \le \tau/2 \\ 0 & \text{for } \;\; \tau/2 < t \le \tau_0/2 \end{cases}$$

[7] *The values of the pulse function at the points of discontinuity $t = -\tau/2$ and $\tau/2$ should, in theory, be defined to be $\frac{1}{2}$ to make the function consistent with Theorem 2.1. However, this more precise but more complicated definition would not change the Fourier series of the function since the integral in Eq. (2.5) would assume an infinitesimal value when evaluated over an infinitesimal range of t, ϵ.*

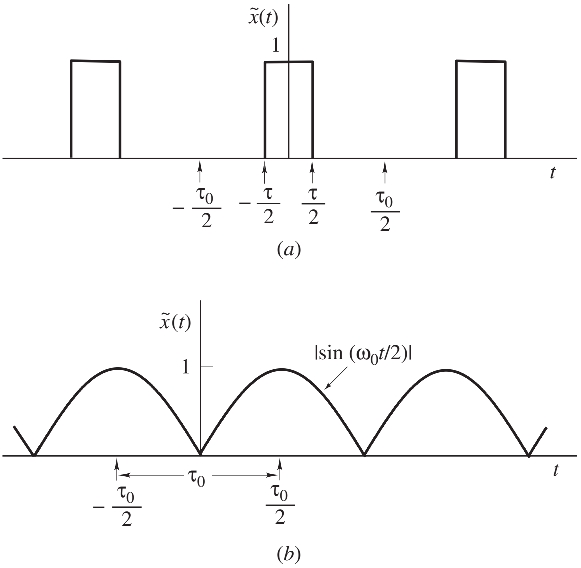


Figure 2.2 Periodic signals: (*a*) Pulse signal, (*b*) rectified sinusoid.

with $\tau_0 = 2\pi/\omega_0$. (*a*) Obtain the Fourier series of $\tilde{x}(t)$ in terms of Eq. (2.3). (*b*) Obtain and plot the amplitude and phase spectrums.

■ **Solution**

As $x(t)$ is symmetrical with respect to the vertical axis, it is an even function of t and Eqs. (2.14a) and (2.14b) apply. We note that Eq. (2.14a) can be obtained from Eq. (2.14b) by letting $k = 0$ and hence, for any k, we have

$$\begin{aligned} X_k &= \frac{2}{\tau_0}\int_0^{\tau_0/2} x(t)\cos k\omega_0 t\, dt = \frac{2}{\tau_0}\int_0^{\tau/2} \cos k\omega_0 t\, dt \\ &= \frac{2}{\tau_0}\left[\frac{\sin k\omega_0 t}{k\omega_0}\right]_0^{\tau/2} \\ &= \frac{\tau}{\tau_0}\,\frac{\sin k\omega_0\tau/2}{k\omega_0\tau/2} \quad \blacksquare \end{aligned} \tag{2.25}$$

Thus

$$|X_k| = \begin{cases} \dfrac{\tau}{\tau_0} & \text{for } k = 0 \\ \left|\dfrac{\tau}{\tau_0}\,\dfrac{\sin k\omega_0\tau/2}{k\omega_0\tau/2}\right| & \text{otherwise} \end{cases} \quad \blacksquare$$

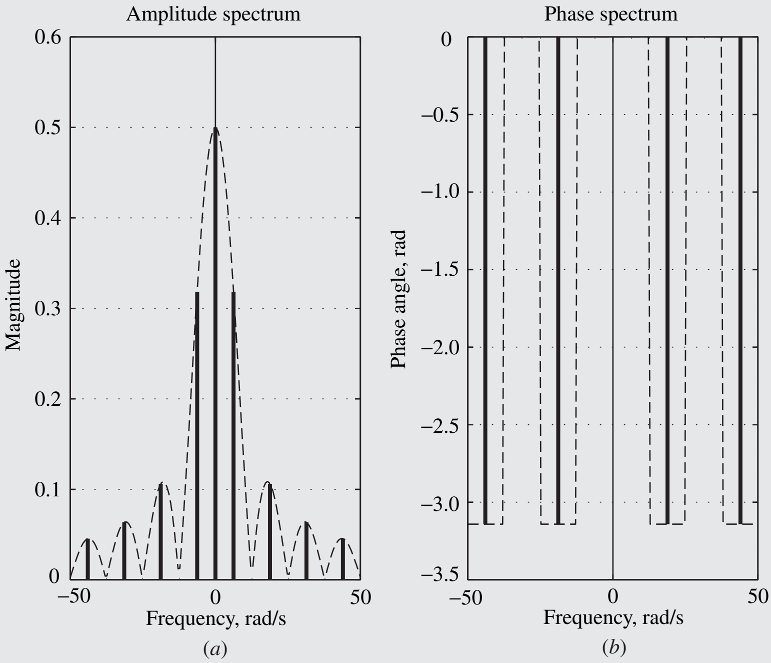


Figure 2.3 Frequency spectrum of periodic pulse signal (Example 2.1): (*a*) Amplitude spectrum, (*b*) phase spectrum.

and

$$\arg X_k = \begin{cases} 0 & \text{if } X_k \geq 0 \\ -\pi & \text{if } X_k < 0 \end{cases} \qquad \blacksquare$$

The amplitude and phase spectrums of the signal for $\tau_0 = 1$ s and $\tau = \tau_0/2$ are plotted in Fig. 2.3*a* and *b*.

The periodic pulse signal analyzed in the above example is of interest in a number of applications, for example, in A/D and D/A converters (see Chap. 6).

Example 2.2 The periodic waveform depicted in Fig. 2.2*b* can be represented by Eq. (2.1) with $x(t)$ given by

$$x(t) = \left|\sin \tfrac{1}{2}\omega_0 t\right| \qquad \text{for } -\tfrac{1}{2}\tau_0 < t \leq \tfrac{1}{2}\tau_0$$

and $\tau_0 = 2\pi/\omega_0$. (*a*) Obtain its Fourier-series representation in terms of Eq. (2.3). (*b*) Obtain and plot the amplitude and phase spectrums of $\tilde{x}(t)$. (*c*) Express the Fourier series obtained in part (*a*) as a linear combination of sines.

■ **Solution**

(*a*) As $x(t)$ is an even function of t, Eqs. (2.14a) and (2.14b) give

$$X_k = \frac{2}{\tau_0}\int_0^{\tau_0/2} x(t)\cos k\omega_0 t\,dt = \frac{2}{\tau_0}\int_0^{\tau_0/2} \sin \tfrac{1}{2}\omega_0 t \cos k\omega_0 t\,dt$$

$$= \frac{2}{\tau_0}\int_0^{\tau_0/2} \cos k\omega_0 t \sin \tfrac{1}{2}\omega_0 t\,dt$$

From trigonometry

$$\cos\theta \sin\psi = \tfrac{1}{2}[\sin(\theta+\psi) - \sin(\theta-\psi)]$$

and hence we obtain

$$X_k = \frac{1}{\tau_0}\int_0^{\tau_0/2} \left[\sin\left(k+\tfrac{1}{2}\right)\omega_0 t - \sin\left(k-\tfrac{1}{2}\right)\omega_0 t\right] dt$$

$$= \frac{1}{\tau_0}\left[\frac{-\cos\left(k+\frac{1}{2}\right)\omega_0 t}{\left(k+\frac{1}{2}\right)\omega_0} - \frac{-\cos\left(k-\frac{1}{2}\right)\omega_0 t}{\left(k-\frac{1}{2}\right)\omega_0}\right]_0^{\tau_0/2}$$

On evaluating the limits, straightforward manipulation gives

$$X_k = \frac{2}{\pi\left(1-4k^2\right)} \quad \blacksquare$$

Thus for any value of k including zero, we have

$$|X_k| = \left|\frac{2}{\pi\left(1-4k^2\right)}\right|$$

and

$$\arg X_k = \begin{cases} 0 & \text{if } k = 0 \\ -\pi & \text{otherwise} \end{cases}$$

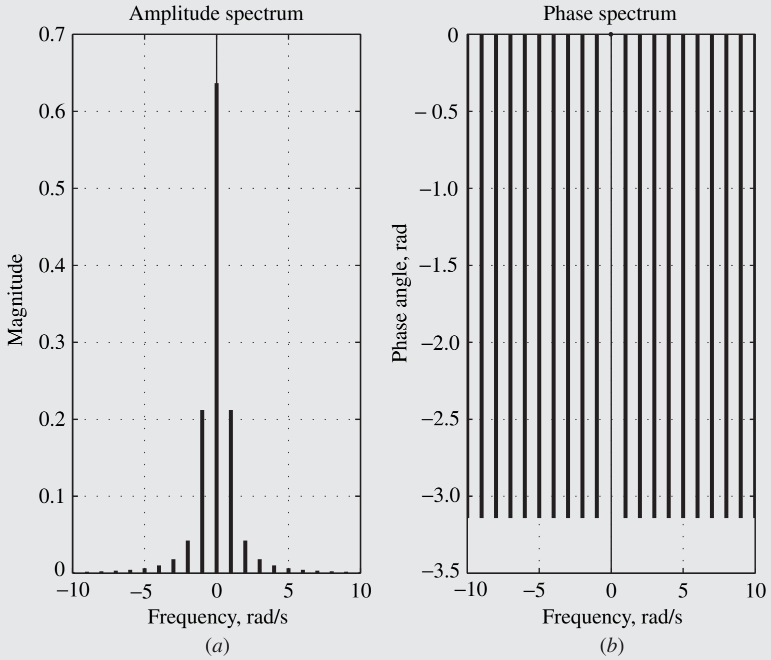


Figure 2.4 Frequency spectrum of rectified waveform (Example 2.2): (*a*) Amplitude spectrum, (*b*) phase spectrum.

(*b*) From Eqs. (2.12a)–(2.12d), we have

$$A_k = 2|X_k| = \left|\frac{4}{\pi\left(1-4k^2\right)}\right| \qquad \text{for } k \ge 0 \quad \blacksquare$$

$$\phi_k = \arg X_k = \begin{cases} 0 & \text{if } k = 0 \\ -\pi & \text{for } k > 0 \end{cases} \quad \blacksquare$$

The amplitude and phase spectrums of the waveform are illustrated in Fig. 2.4 for the case where $\omega_0 = 1$ rad/s.

(*c*) Now Eq. (2.13b) yields

$$\begin{aligned}\tilde{x}(t) &= \frac{2}{\pi} + \sum_{k=1}^{\infty}\left|\frac{4}{\pi\left(1-4k^2\right)}\right| \sin\left(k\omega_0 t - \pi + \tfrac{1}{2}\pi\right)\\ &= \tfrac{2}{\pi} + \tfrac{4}{3\pi}\sin\left(\omega_0 t - \tfrac{1}{2}\pi\right) + \tfrac{4}{15\pi}\sin\left(2\omega_0 t - \tfrac{1}{2}\pi\right)\\ &\quad + \tfrac{4}{35\pi}\sin\left(3\omega_0 t - \tfrac{1}{2}\pi\right) + \tfrac{4}{63\pi}\sin\left(4\omega_0 t - \tfrac{1}{2}\pi\right) + \cdots \quad \blacksquare\end{aligned}$$

The waveform analyzed in Example 2.2 is essentially a sinusoidal waveform with its negative half cycles reversed and is the type of waveform generated by a so-called *full-wave rectifier circuit*. Circuits of this type are found in AC-to-DC adaptors such as those used to power laptop or handheld computers and modems. The Fourier series obtained shows that an AC supply voltage of amplitude 1 V[8] would produce a DC output voltage of $2/\pi$ V. Hence, an AC voltage of amplitude 170 V would produce a DC voltage of 108.23 V. We note also that there would be an infinite number of residual AC components with frequencies $\omega_0, 2\omega_0, 3\omega_0, 4\omega_0, \ldots$, namely, the fundamental and harmonics, with amplitudes, of 72.15, 14.43, 6.18, 3.44 V, . . . , respectively. In good-quality AC-to-DC adaptors, the amplitudes of the harmonics are reduced to insignificant levels through the use of analog filter circuits.

Example 2.3 (a) Obtain the Fourier series of the periodic signal shown in Fig. 2.5a in terms of Eq. (2.3). (b) Obtain and plot the amplitude and phase spectrums of $\tilde{x}(t)$.

■ **Solution**

(a) The signal in Fig. 2.5a can be modeled by using shifted copies of $p_{\tau/2}(t)$ for the representation of signal $x(t)$ in Eq. (2.2), where $p_\tau(t)$ is the pulse signal of Example 2.1, that is, we can write

$$x(t) = p_{\tau/2}\left(t + \tfrac{1}{4}\tau\right) - p_{\tau/2}\left(t - \tfrac{1}{4}\tau\right)$$

As $x(t)$ is antisymmetrical with respect to the vertical axis, it is an odd function of time. Hence, from Eq. (2.16a), we get

$$X_0 = 0$$

Now from Eq. (2.16b), we have

$$\begin{aligned}
X_k &= -j\frac{2}{\tau_0}\int_0^{\tau_0/2} x(t)\sin k\omega_0 t\, dt \\
&= -j\frac{2}{\tau_0}\int_0^{\tau_0/2} p_{\tau/2}\left(t - \tfrac{1}{4}\tau\right)\sin k\omega_0 t\, dt \\
&= -j\frac{2}{\tau_0}\int_0^{\tau/2} -\sin k\omega_0 t\, dt \\
&= j\frac{2}{\tau_0}\left[\frac{-\cos k\omega_0 t}{k\omega_0}\right]_0^{\tau/2} = j\frac{2}{\tau_0}\left[\frac{1-\cos k\omega_0\tau/2}{k\omega_0}\right] \\
&= j\frac{4\sin^2 k\omega_0\tau/4}{k\omega_0\tau_0} \qquad \text{for } k = 1,\ 2,\ \ldots
\end{aligned}$$

[8] *The symbol V stands for volts.*

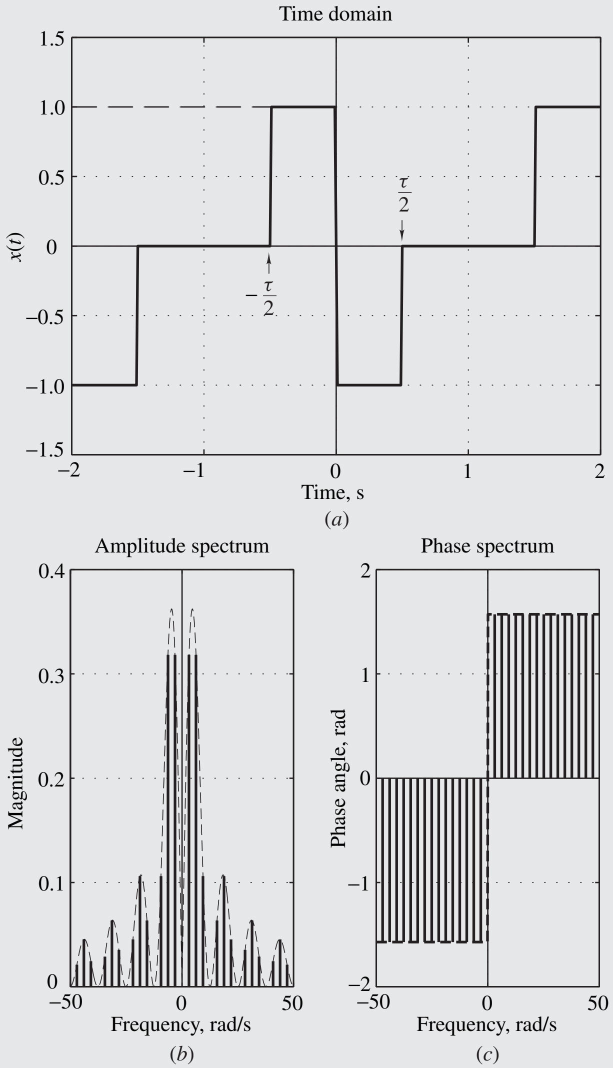


Figure 2.5 Frequency spectrum (Example 2.3): (*a*) Time-domain representation, (*b*) amplitude spectrum, (*c*) phase spectrum.

that is,

$$X_k = \begin{cases} 0 & \text{for } k = 0 \\ j\dfrac{4\sin^2 k\omega_0\tau/4}{k\omega_0\tau_0} & \text{for } k = 1,\ 2,\ \ldots \end{cases} \qquad \blacksquare$$

(b) The amplitude and phase spectrums are obtained as

$$|X_0| = 0 \qquad \text{and} \qquad \arg X_0 = 0$$

$$|X_k| = \left| \frac{2}{k\pi} \sin^2 \frac{k\pi\tau}{2\tau_0} \right| \qquad \text{for } k = 1, 2, \ldots \quad \blacksquare$$

$$\arg X_k = \begin{cases} \tfrac{1}{2}\pi & \text{if } k > 0 \\ 0 & \text{if } k = 0 \\ -\tfrac{1}{2}\pi & \text{if } k < 0 \end{cases} \qquad \text{for } k = 1, 2, \ldots \quad \blacksquare$$

(See Fig. 2.5b and c for plots.)

Example 2.4 Deduce the Fourier series of the following periodic signal

$$\tilde{x}(t) = \sin^4 \omega_0 t$$

in terms of cosines.

■ **Solution**

We can write

$$\begin{aligned} \tilde{x}(t) &= (\sin^2 \omega_0 t)^2 = \left[\tfrac{1}{2}(1 - \cos 2\omega_0 t)\right]^2 \\ &= \tfrac{1}{4}\left(1 - 2\cos 2\omega_0 t + \cos^2 2\omega_0 t\right) \\ &= \tfrac{1}{4}\left[1 - 2\cos 2\omega_0 t + \tfrac{1}{2}\left(1 + \cos 4\omega_0 t\right)\right] \\ &= \tfrac{1}{4}\left(\tfrac{3}{2} - 2\cos 2\omega_0 t + \tfrac{1}{2}\cos 4\omega_0 t\right) \\ &= \tfrac{3}{8} - \tfrac{1}{2}\cos 2\omega_0 t + \tfrac{1}{8}\cos 4\omega_0 t \quad \blacksquare \end{aligned}$$

The above is a Fourier series and by virtue of Theorem 2.4, it is the unique Fourier series for the given signal.

2.3 FOURIER TRANSFORM

The Fourier series described in the previous section can deal quite well with periodic signals but, unfortunately, it is not applicable to nonperiodic signals. Periodic signals occur in a number of applications but more often than not signals tend to be nonperiodic, e.g., communications, seismic, or music signals, and for signals of this type some mathematical technique other than the Fourier series must be used to obtain useful spectral representations.

A mathematical technique that can deal quite effectively with nonperiodic signals is the *Fourier transform* [5]. This can be defined as an independent transformation. Alternatively, it can be deduced from the Fourier series by treating a nonperiodic signal as if it were periodic and then letting the period approach infinity [2]. The latter approach provides a fairly accurate physical interpretation of a somewhat abstract mathematical technique and will, therefore, be pursued.

2.3.1 Derivation

Let us consider the nonperiodic pulse signal of Fig. 2.6*a*, which comprises just a single pulse. This signal can, in theory, be deemed to be the special case of the periodic pulse signal $\tilde{x}(t)$ shown in Fig. 2.2*a* when the period τ_0 is increased to infinity, that is,

$$x(t) = \lim_{\tau_0 \to \infty} \tilde{x}(t) = p_\tau(t)$$

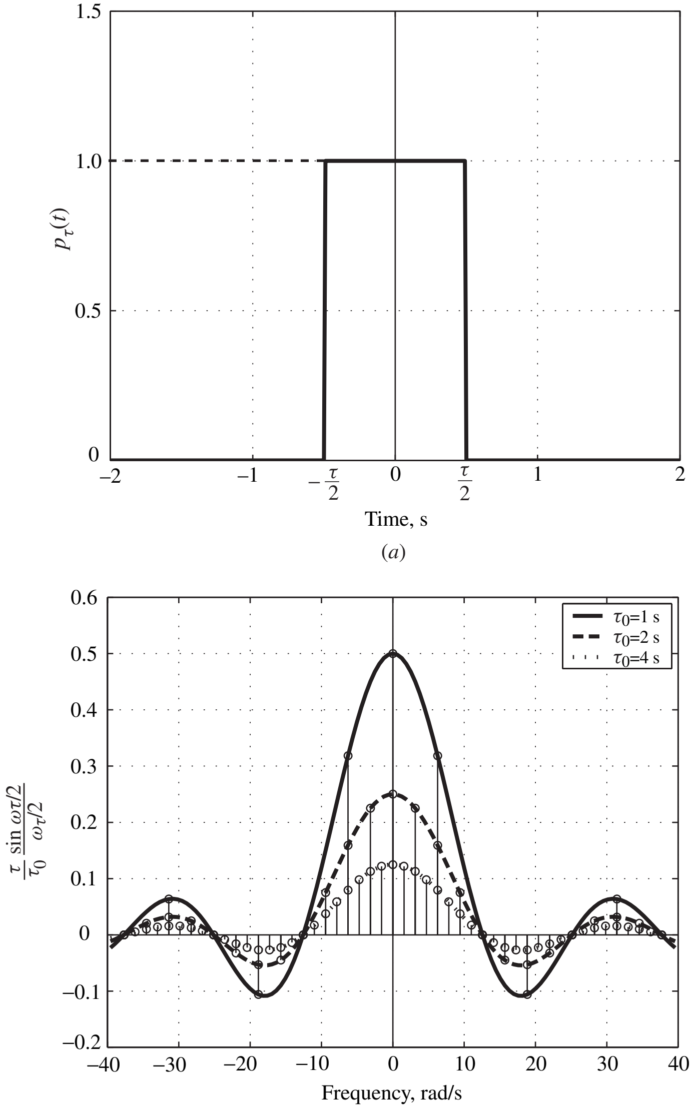
The Fourier series of the periodic pulse signals was obtained in Example 2.1 and its Fourier-series coefficients $\{X_k\}$ are given by Eq. (2.25). If we replace $k\omega_0$ by the continuous variable ω, Eq. (2.25) assumes the form

$$X_k = \frac{\tau}{\tau_0} \frac{\sin k\omega_0\tau/2}{k\omega_0\tau/2} = \frac{\tau}{\tau_0} \frac{\sin \omega\tau/2}{\omega\tau/2}$$

Let us examine the behavior of the Fourier series as the period τ_0 of the waveform is doubled to $2\tau_0$, then doubled again to $4\tau_0$, and so on, assuming that the duration τ of each pulse remains fixed. Using a simple MATLAB program with $\tau_0 = 1$ and $\tau = \frac{1}{2}$ s, the plots in Fig. 2.6*b* can be readily obtained. Two things can be observed in this illustration, namely, the magnitudes of the Fourier-series coefficients $\{X_k\}$ are progressively halved because X_k is proportional to τ/τ_0, whereas the number of frequency components is progressively doubled because the spacing between adjacent harmonics is halved from ω_0 to $\frac{1}{2}\omega_0$, then to $\frac{1}{4}\omega_0$, and so on. Evidently, if we were to continue doubling the period *ad infinitum*, the coefficients $\{X_k\}$ would become infinitesimally small whereas the number of harmonics would become infinitely large and the spacing between adjacent harmonics would approach zero. In effect, *applying the Fourier series to the periodic pulse signal of Fig. 2.2a and letting* $\tau_0 \to \infty$ *would transform the signal in Fig. 2.2a to the nonperiodic pulse signal of Fig. 2.6a but as* $X_k \to 0$, *the approach does not yield a meaningful spectral representation for the nonperiodic pulse signal of Fig. 2.6a.* The same problem would arise if one were to apply the Fourier series to any nonperiodic signal and, therefore, an alternative spectral representation must be sought for nonperiodic signals.

The previous analysis has shown that as $\tau_0 \to \infty$, we get $\omega_0 \to 0$ and $X_k \to 0$. However, the quantity

$$X(j\omega) = \lim_{\tau_0 \to \infty} X(jk\omega_0) \triangleq \lim_{\tau_0 \to \infty} \frac{X_k}{f_0} \tag{2.26}$$



(*a*)

(*b*)

Figure 2.6 (*a*) Pulse function. (*b*) Fourier-series representation of the pulse signal shown in Fig. 2.2*a* for pulse periods of τ_0, $2\tau_0$, and $4\tau_0$.

where $\omega = k\omega_0$ and $f_0 = 1/\tau_0 = \omega_0/2\pi$ is the frequency between adjacent harmonics in Hz, assumes a finite value for a large class of signals and, furthermore, it constitutes a physically meaningful spectral representation for nonperiodic signals. As will be shown in Theorem 2.16, $|X(j\omega)|^2$ is proportional to the energy density of signal $x(t)$ per unit bandwidth in Hz at frequency $f = \omega/(2\pi)$ in Hz.

From Eqs. (2.26) and (2.5), we can write

$$\begin{aligned} X(jk\omega_0) &= \frac{X_k}{f_0} = \tau_0 X_k \\ &= \int_{-\tau_0/2}^{\tau_0/2} x(t)e^{-jk\omega_0 t}\,dt \end{aligned}$$

and therefore,

$$X(j\omega) = \lim_{\tau_0 \to \infty} X(jk\omega_0)$$

or

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt \tag{2.27}$$

The quantity $X(j\omega)$ is known universally as the *Fourier transform* of nonperiodic signal $x(t)$.

If the Fourier-series coefficients of a periodic signal are known, then the signal itself can be reconstructed by using the formula for the Fourier series given by Eq. (2.3), namely,

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t} \qquad \text{for } -\tau_0/2 \le t \le \tau_0/2$$

As before, a nonperiodic signal can be generated from a periodic one by letting $\tau_0 \to \infty$ in $\tilde{x}(t)$, that is,

$$x(t) = \lim_{\tau_0 \to \infty} \tilde{x}(t) = \lim_{\tau_0 \to \infty} \sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t} \qquad \text{for } -\tau_0/2 \le t \le \tau_0/2 \tag{2.28}$$

From Eq. (2.26), we have

$$X(jk\omega_0) = \frac{X_k}{\omega_0/2\pi}$$

or

$$X_k = \frac{X(jk\omega_0)\omega_0}{2\pi}$$

and hence Eq. (2.28) assumes the form

$$x(t) = \lim_{\tau_0 \to \infty} \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} X(jk\omega_0)e^{jk\omega_0 t}\omega_0 \qquad \text{for } -\tau_0/2 \le t \le \tau_0/2$$

If we now let $k\omega_0 = \omega$ and $\omega_0 = \Delta\omega$, then as $\tau_0 \to \infty$ the above summation defines an integral. Therefore,

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}\,d\omega \tag{2.29}$$

This is referred to as the *inverse Fourier transform* of $X(j\omega)$ because it can be used to recover the nonperiodic signal from its Fourier transform.

A nonperiodic signal can be represented by a Fourier transform to the extent that the integrals in Eqs. (2.27) and (2.29) can be evaluated. The conditions that would assure the existence of the Fourier transform and its inverse are stated in Theorem 2.5 in Sec. 2.3.3.

The Fourier transform and its inverse are often represented in terms of operator notation as

$$X(j\omega) = \mathcal{F}x(t) \qquad \text{and} \qquad x(t) = \mathcal{F}^{-1}X(j\omega)$$

respectively. An even more economical notation, favored by Papoulis [5], is given by

$$x(t) \leftrightarrow X(j\omega)$$

and is interpreted as: $X(j\omega)$ is the Fourier transform of $x(t)$, which can be obtained by using Eq. (2.27), and $x(t)$ is the inverse Fourier transform of $X(j\omega)$, which can be obtained by using Eq. (2.29). The choice of notation depends, of course, on the circumstances.

Like the Fourier series coefficients of a periodic signal, $X(j\omega)$ is, in general, complex and it can be represented in terms of its real and imaginary parts as

$$X(j\omega) = \Re e\, X(j\omega) + j\,\Im m\, X(j\omega)$$

Alternatively, it can be expressed in terms of its magnitude and angle as

$$X(j\omega) = A(\omega)e^{j\phi(\omega)}$$

where

$$A(\omega) = |X(j\omega)| = \sqrt{[\Re e\, X(j\omega)]^2 + [\Im m\, X(j\omega)]^2} \tag{2.30}$$

and

$$\phi(\omega) = \arg X(j\omega) = \tan^{-1}\frac{\Im m\, X(j\omega)}{\Re e\, X(j\omega)} \tag{2.31}$$

As physical quantities, the magnitude and angle of the Fourier transform are the *amplitude spectrum* and *phase spectrum* of the signal, respectively, and the two together constitute its *frequency spectrum*.

A fairly standard practice, is to use lower-case symbols for the time domain and upper-case symbols for the frequency domain. This convention will as far as possible be adopted throughout this textbook to avoid confusion.

2.3.2 Particular Forms

In the above analysis, we have implicitly assumed that signal $x(t)$ is real. Although this is typically the case, there are certain applications where $x(t)$ can be complex. Nevertheless, the Fourier transform as defined in the previous section continues to apply (see Ref. [5]), that is,

$$\begin{aligned}\mathcal{F}x(t) &= \mathcal{F}[\Re e\, x(t) + j\,\Im m\, x(t)]\\ &= \int_{-\infty}^{\infty} [\Re e\, x(t) + j\,\Im m\, x(t)]e^{-j\omega t}\, dt\\ &= \int_{-\infty}^{\infty} [\Re e\, x(t) + j\,\Im m\, x(t)][\cos\omega t - j\sin\omega t]\, dt\\ &= \Re e\, X(j\omega) + j\,\Im m\, X(j\omega) \qquad (2.32a)\end{aligned}$$

where

$$\Re e\, X(j\omega) = \int_{-\infty}^{\infty} \{\Re e[x(t)]\cos\omega t + \Im m[x(t)]\sin\omega t\}\, dt \qquad (2.32b)$$

$$\Im m\, X(j\omega) = -\int_{-\infty}^{\infty} \{\Re e[x(t)]\sin\omega t - \Im m[x(t)]\cos\omega t\}\, dt \qquad (2.32c)$$

If $x(t)$ is real, then Eqs. (2.32b) and (2.32c) assume the forms

$$\Re e\, X(j\omega) = \int_{-\infty}^{\infty} x(t)\cos\omega t\, dt \qquad (2.33a)$$

$$\Im m\, X(j\omega) = -\int_{-\infty}^{\infty} x(t)\sin\omega t\, dt \qquad (2.33b)$$

As the cosine is an even function and the sine is an odd function of frequency, we conclude that the real part of the Fourier transform of a real signal is an even function and the imaginary part is an odd function of frequency. Hence,

$$\begin{aligned}X(-j\omega) &= \Re e\, X(-j\omega) + j\,\Im m\, X(-j\omega)\\ &= \Re e\, X(j\omega) - j\,\Im m\, X(j\omega)\\ &= X^*(j\omega) \qquad (2.34)\end{aligned}$$

that is, *$X(-j\omega)$ is equal to the complex conjugate of $X(j\omega)$*. It also follows that the amplitude spectrum given by Eq. (2.30) is an even function and the phase spectrum given by Eq. (2.31) is an odd function of frequency.

For a real $x(t)$, the inverse Fourier transform can be expressed as

$$\begin{aligned}x(t) &= \mathcal{F}^{-1}[\Re e\, X(j\omega) + j\,\Im m\, X(j\omega)]\\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} [\Re e\, X(j\omega) + j\,\Im m\, X(j\omega)]e^{j\omega t}\, d\omega\\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} [\Re e\, X(j\omega) + j\,\Im m\, X(j\omega)][\cos\omega t + j\sin\omega t]\, d\omega\\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \{\Re e[X(j\omega)]\cos\omega t - \Im m[X(j\omega)]\sin\omega t\}\, d\omega \qquad (2.35a)\end{aligned}$$

since the imaginary part is zero. The product of two odd functions such as $\Im m[X(j\omega)]\sin\omega t$ is an even function and thus we can write

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Re e[X(j\omega)e^{j\omega t}]\,d\omega$$

$$= \frac{1}{\pi}\,\Re e\left[\int_{0}^{\infty} X(j\omega)e^{j\omega t}d\omega\right] \tag{2.35b}$$

If the signal is both real and an even function of time, that is, $x(-t) = x(t)$, then Eqs. (2.33a) and (2.33b) assume the form

$$\Re e\, X(j\omega) = 2\int_{0}^{\infty} x(t)\cos\omega t\,dt \tag{2.36a}$$

$$\Im m\, X(j\omega) = 0 \tag{2.36b}$$

that is, the Fourier transform is real. As the imaginary part of the Fourier transform is zero in this case, Eq. (2.35a) assumes the form

$$x(t) = \frac{1}{\pi}\int_{0}^{\infty} \Re e[X(j\omega)]\cos\omega t\,d\omega \tag{2.36c}$$

The converse is also true, i.e., if the Fourier transform is real, then the signal is an even function of time.

If the signal is both real and an odd function of time, that is, $x(-t) = -x(t)$, then Eqs. (2.33a) and (2.33b) assume the form

$$\Re e\, X(j\omega) = 0 \tag{2.37a}$$

$$\Im m\, X(j\omega) = -2\int_{0}^{\infty} x(t)\sin\omega t\,dt \tag{2.37b}$$

and from Eq. (2.35a), we get

$$x(t) = -\frac{1}{\pi}\int_{0}^{\infty} \Im m[X(j\omega)]\sin\omega t\,d\omega \tag{2.37c}$$

The above principles can be extended to arbitrary signals that are neither even nor odd with respect to time. Such signals can be expressed in terms of even and odd components, $x_e(t)$ and $x_o(t)$, respectively, as

$$x(t) = x_e(t) + x_o(t) \tag{2.38a}$$

where

$$x_e(t) = \tfrac{1}{2}[x(t) + x(-t)] \tag{2.38b}$$

$$x_o(t) = \tfrac{1}{2}[x(t) - x(-t)] \tag{2.38c}$$

From Eq. (2.38a)

$$X(j\omega) = \Re e\, X(j\omega) + j\, \Im m\, X(j\omega) = X_e(j\omega) + X_o(j\omega) \tag{2.39}$$

and as $X_e(j\omega)$ is purely real and $X_o(j\omega)$ is purely imaginary, we have

$$x_e(t) \leftrightarrow \Re e\, X(j\omega) \tag{2.40a}$$

where

$$\Re e\, X(j\omega) = 2\int_0^\infty x_e(t)\cos\omega t\, dt \tag{2.40b}$$

$$x_e(t) = \frac{1}{\pi}\int_0^\infty \Re e[X(j\omega)]\cos\omega t\, d\omega \tag{2.40c}$$

and

$$x_o(t) \leftrightarrow j\, \Im m\, X(j\omega) \tag{2.41a}$$

where

$$\Im m\, X(j\omega) = -2\int_0^\infty x_o(t)\sin\omega t\, dt \tag{2.41b}$$

$$x_o(t) = -\frac{1}{\pi}\int_0^\infty \Im m[X(j\omega)]\sin\omega t\, d\omega \tag{2.41c}$$

Occasionally, signals are ‘right-sided’ in the sense that their value is zero for negative time,[9] that is, $x(t) = 0$, for $t < 0$. For such signals, $x(-t) = 0$ for $t > 0$ and hence Eqs. (2.38b) and (2.38c) give

$$x(t) = 2x_e(t) = 2x_o(t) \tag{2.42a}$$

and from Eqs. (2.40c) and (2.41c), we have

$$x(t) = \frac{2}{\pi}\int_0^\infty \Re e[X(j\omega)]\cos\omega t\, d\omega \tag{2.42b}$$

$$= -\frac{2}{\pi}\int_0^\infty \Im m[X(j\omega)]\sin\omega t\, d\omega \tag{2.42c}$$

For this particular case, the real and imaginary parts of the Fourier transform are dependent on each other and, in fact, one can readily be obtained from the other. For example, if $\Re e\, X(j\omega)$ is known, then $x(t)$ can be obtained from Eq. (2.42b) and upon eliminating $x(t)$ in Eq. (2.33b) $\Im m\, X(j\omega)$ can be obtained.

[9] *Such signals have often been referred to as causal signals in the past but the word is a misnomer. Causality is a system property as will be shown in Chap. 4.*

It should be emphasized here that the relations in Eq. (2.42) are valid only for $t > 0$. For the case $t = 0$, $x(t)$ must be defined as the average of its left- and right-hand limits at $t = 0$, to render $x(t)$ consistent with the convergence theorem of the Fourier transform (see Theorem 2.5), that is,

$$\begin{aligned} x(0) &= \tfrac{1}{2}[x(0-) + x(0+)] = \tfrac{1}{2}x(0+) \\ &= \frac{1}{\pi}\int_0^\infty \Re e[X(j\omega)]\,d\omega \end{aligned}$$

The Fourier transform will now be used to obtain spectral representations for some standard nonperiodic waveforms.

Example 2.5 (*a*) Obtain the Fourier transform of the nonperiodic pulse signal shown in Fig. 2.6*a*. (*b*) Obtain and plot the amplitude and phase spectrums of $x(t)$.

■ **Solution**

(*a*) From Fig. 2.6*a*, the pulse signal can be represented by

$$x(t) \equiv p_\tau(t) = \begin{cases} 1 & \text{for } -\tau/2 \le t \le \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

Hence

$$\begin{aligned} X(j\omega) &= \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt = \int_{-\tau/2}^{\tau/2} e^{-j\omega t}\,dt \\ &= \left[\frac{e^{-j\omega t}}{-j\omega}\right]_{-\tau/2}^{\tau/2} = \frac{2\left(e^{j\omega\tau/2} - e^{-j\omega\tau/2}\right)}{2j\omega} \\ &= \frac{2\sin\omega\tau/2}{\omega} \end{aligned}$$

or

$$p_\tau(t) \leftrightarrow \frac{2\sin\omega\tau/2}{\omega} \quad ■$$

where $(2\sin\omega\tau/2)/\omega$ is often referred to as a *sinc function*.

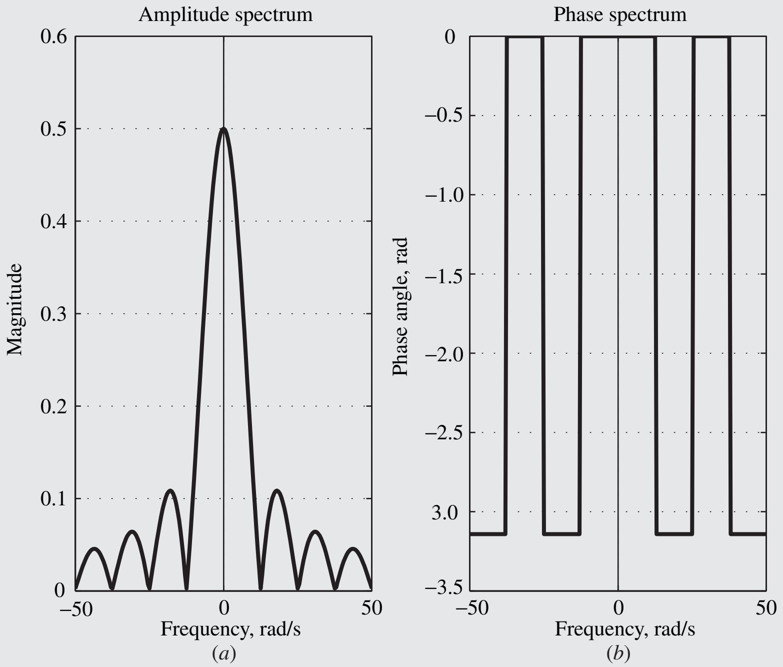


Figure 2.7 Frequency spectrum of pulse (Example 2.5): (*a*) Amplitude spectrum, (*b*) phase spectrum.

(*b*) The amplitude and phase spectrums are given by

$$A(\omega) = |X(\omega)| = \left|\frac{2\sin\omega\tau/2}{\omega}\right| \quad \blacksquare$$

$$\phi(\omega) = \arg X(\omega) = \begin{cases} 0 & \text{if } \dfrac{2\sin\omega\tau/2}{\omega} \geq 0 \\ -\pi & \text{if } \dfrac{2\sin\omega\tau/2}{\omega} < 0 \end{cases} \quad \blacksquare$$

and are illustrated in Fig. 2.7. Note the continuous frequency spectrum of the nonperiodic pulse signal in contrast to the discrete frequency spectrum of the periodic pulse signal depicted in Fig. 2.3.

Example 2.6 (*a*) Obtain the Fourier transform of the decaying exponential signal

$$x(t) = u(t)e^{-\alpha t}$$

where α is a positive constant and

$$u(t) = \begin{cases} 1 & \text{for } t \geq 0 \\ 0 & \text{for } t < 0 \end{cases}$$

is known as the *unit-step* function. (b) Obtain and plot the amplitude and phase spectrums of $x(t)$.

■ **Solution**

(a) We can write

$$X(j\omega) = \int_{-\infty}^{\infty} u(t)e^{-\alpha t}e^{-j\omega t}\,dt = \int_{0}^{\infty} e^{-(\alpha+j\omega)t}\,dt$$

$$= \left[-\frac{e^{-(\alpha+j\omega)t}}{\alpha + j\omega}\right]_0^{\infty}$$

For $\alpha > 0$, we note that

$$\lim_{t\to\infty} e^{-\alpha t} \to 0$$

and as a result

$$\lim_{t\to\infty} e^{(-\alpha-j\omega)t} = \lim_{t\to\infty} \left(e^{-\alpha t}\cdot e^{-j\omega t}\right) \to 0$$

Thus

$$X(j\omega) = \frac{1}{\alpha + j\omega}$$

or

$$u(t)e^{-\alpha t} \leftrightarrow \frac{1}{\alpha + j\omega} \quad \blacksquare$$

(b) The amplitude and phase spectrums of the signal are given by

$$A(\omega) = \frac{1}{\sqrt{\alpha^2 + \omega^2}} \qquad \text{and} \qquad \phi(\omega) = -\tan^{-1}\frac{\omega}{\alpha} \quad \blacksquare$$

respectively. Note that certain ambiguities can arise in the evaluation of the phase spectrum as the above equation has an infinite number of solutions due to the periodicity of the tangent function (see Sec. A.3.7).

(See Fig. 2.8 for plots.)

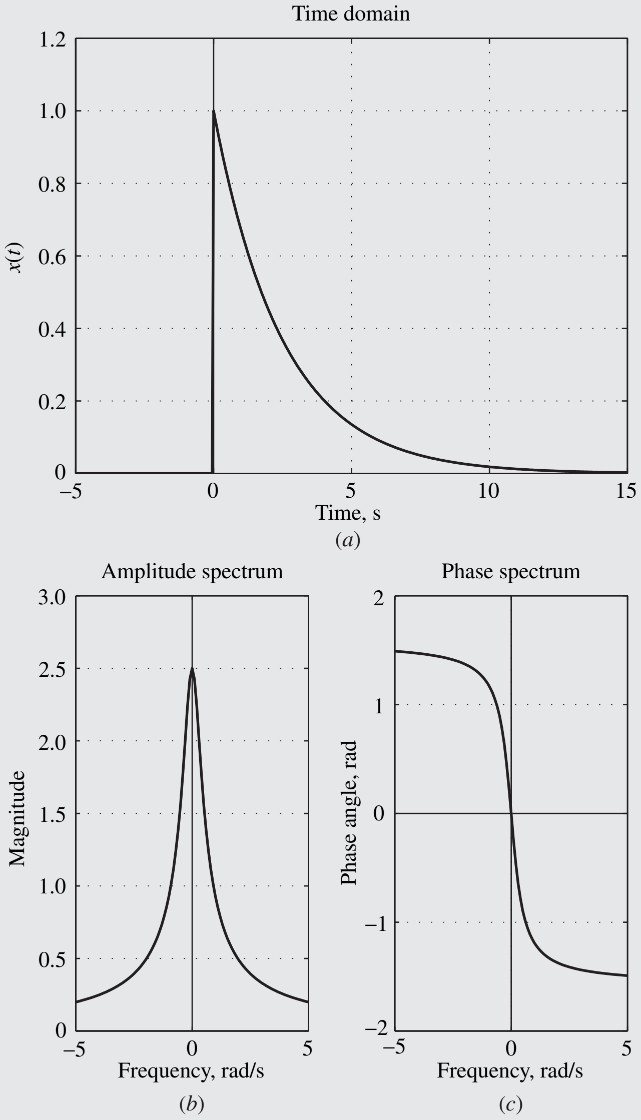


Figure 2.8 Frequency spectrum of decaying exponential (Example 2.6 with $\alpha = 0.4$): (a) Amplitude spectrum, (b) phase spectrum.

2.3.3 Theorems and Properties

The properties of the Fourier transform, like those of the Fourier series, can be described in terms of a small number of theorems as detailed below.

Theorem 2.5 Convergence ***If signal $x(t)$ is piecewise smooth in each finite interval and is, in addition, absolutely integrable, i.e., it satisfies the inequality***

$$\int_{-\infty}^{\infty} |x(t)|\, dt \le K < \infty \tag{2.43}$$

where K is some positive constant, then the integral in Eq. (2.27) converges. Furthermore, the substitution of $X(j\omega)$ in Eq. (2.29) converges to $x(t)$ at points where $x(t)$ is continuous; at points where $x(t)$ is discontinuous, Eq. (2.29) converges to the average of the left- and right-hand limits of $x(t)$, namely,

$$x(t) = \tfrac{1}{2}[x(t+) + x(t-)]$$

Proof (See pp. 471–473 of [6] for proof.) ▲

The convergence theorem essentially delineates sufficient conditions for the existence of the Fourier transform and its inverse and is analogous to the convergence theorem of the Fourier series, i.e., Theorem 2.1. Note that periodic signals are not absolutely integrable as the area under the graph of $|\tilde{x}(t)|$ over the infinite range $-\infty \le t \le \infty$ is infinite, and a similar problem arises in connection with impulse signals which comprise infinitely tall and infinitesimally thin pulses. The application of the Fourier transform to signals that do not satisfy the convergence theorem will be examined in Sec. 6.2.

The following theorems hold if $x(t)$, $x_1(t)$, and $x_2(t)$ are absolutely integrable, which would imply that

$$x(t) \leftrightarrow X(j\omega) \quad x_1(t) \leftrightarrow X_1(j\omega) \quad x_2(t) \leftrightarrow X_2(j\omega)$$

The parameters $a,\ b,\ t_0$, and ω_0 are arbitrary constants which could be complex in theory.

Theorem 2.6 Linearity ***The Fourier transform and its inverse are linear operations, that is,***

$$ax_1(t) + bx_2(t) \leftrightarrow aX_1(j\omega) + bX_2(j\omega)$$

Proof See Prob. 2.18. ▲

Theorem 2.7 Symmetry[10] ***Given a Fourier transform pair***

$$x(t) \leftrightarrow X(j\omega)$$

the Fourier transform pair

$$X(jt) \leftrightarrow 2\pi x(-\omega)$$

can be generated.

[10] *Also referred to as the duality property.*

Proof By letting $t \to -t$ in the inverse Fourier transform of Eq. (2.29), we get

$$2\pi x(-t) = \int_{-\infty}^{\infty} X(j\omega)e^{-j\omega t}\, d\omega$$

and if we now let $t \to \omega$ and $\omega \to t$, we have

$$2\pi x(-\omega) = \int_{-\infty}^{\infty} X(jt)e^{-j\omega t}\, dt$$

that is, the Fourier transform of $X(jt)$ is $2\pi x(-\omega)$ and the inverse Fourier transform of $2\pi x(-\omega)$ is $X(jt)$. ▲

Theorem 2.8 Time Scaling

$$\boldsymbol{x(at) \leftrightarrow \frac{1}{|a|}X\left(\frac{j\omega}{a}\right)}$$

Proof Assuming that $a > 0$, letting $t = t'/a$, and then replacing t' by t in the definition of the Fourier transform, we get

$$\begin{aligned}\int_{-\infty}^{\infty} x(at)e^{-j\omega t}\, dt &= \frac{1}{a}\int_{-\infty}^{\infty} x(t')e^{-j(\omega/a)t'}\, dt' \\ &= \frac{1}{a}\int_{-\infty}^{\infty} x(t)e^{-j(\omega/a)t}\, dt \\ &= \frac{1}{a}X\left(j\frac{\omega}{a}\right) \qquad (2.44a)\end{aligned}$$

If $a < 0$, proceeding as above and noting that the limits of integration are reversed in this case, we get

$$\begin{aligned}\int_{-\infty}^{\infty} x(at)e^{-j\omega t}\, dt &= \frac{1}{a}\int_{\infty}^{-\infty} x(t')e^{-j(\omega/a)t'}\, dt' \\ &= -\frac{1}{a}\int_{-\infty}^{\infty} x(t)e^{-j(\omega/a)t}\, dt \\ &= \frac{1}{|a|}X\left(j\frac{\omega}{a}\right) \qquad (2.44b)\end{aligned}$$

Now, if we compare Eqs. (2.44a) and (2.44b), we note that Eq. (2.44b) applies for $a < 0$ as well as for $a > 0$, and hence the theorem is proved. ▲

One often needs to normalize the time scale of a signal to a more convenient range to avoid awkward numbers in the representation. For example, the time scale of a signal that extends from 0 to 10^{-6} s could be scaled to the range 0 to 1 s. Occasionally, the available signal is in terms of a normalized time scale and it may become necessary to 'denormalize' the time scale, say, from the normalized range 0 to 1 s to the actual range. In either of these situations, time scaling is required, which changes the Fourier transform of the signal.

Theorem 2.9 Time Shifting

$$x(t - t_0) \leftrightarrow e^{-j\omega t_0} X(j\omega)$$

Proof See Prob. 2.19, part (*a*). ▲

The time-shifting theorem is handy in situations where a signal is delayed or advanced by a certain period of time. Evidently, delaying a signal by t_0 s amounts to multiplying the Fourier transform of the signal by the exponential of $-j\omega t_0$.

Theorem 2.10 Frequency Shifting

$$e^{j\omega_0 t} x(t) \leftrightarrow X(j\omega - j\omega_0)$$

Proof See Prob. 2.19, part (*b*). ▲

The similarity of Theorems 2.9 and 2.10 is a consequence of the similarity between the Fourier transform and its inverse.

Theorem 2.11 Time Differentiation

$$\frac{d^k x(t)}{dt^k} \leftrightarrow (j\omega)^k X(j\omega)$$

Proof The theorem can be proved by obtaining the kth derivative of both sides in Eq. (2.29) with respect to t. ▲

Theorem 2.12 Frequency Differentiation

$$(-jt)^k x(t) \leftrightarrow \frac{d^k X(j\omega)}{d\omega^k}$$

Proof The theorem can be proved by obtaining the kth derivative of both sides in Eq. (2.27) with respect to ω. ▲

Theorem 2.13 Moments Theorem ***For a bounded signal $x(t)$, the relation***

$$(-j)^k m_k = \frac{d^k X(0)}{d\omega^k} \tag{2.45}$$

holds where

$$m_k = \int_{-\infty}^{\infty} t^k x(t)\, dt$$

is said to be the kth moment of $x(t)$.

Proof See Ref. [5] for proof. ▲

The moments theorem will be found useful in the derivation of Fourier transforms for Gaussian functions (see Example 2.11).

Theorem 2.14 Time Convolution

$$\boldsymbol{x_1(t) \otimes x_2(t) \leftrightarrow X_1(j\omega)X_2(j\omega)}$$

where

$$\boldsymbol{x_1(t) \otimes x_2(t) = \int_{-\infty}^{\infty} x_1(\tau)x_2(t-\tau)\,d\tau} \tag{2.46a}$$

$$\boldsymbol{= \int_{-\infty}^{\infty} x_1(t-\tau)x_2(\tau)\,d\tau} \tag{2.46b}$$

Proof From Eq. (2.46b) and the definition of the Fourier transform, we have

$$\begin{aligned}\mathcal{F}[x_1(t)\otimes x_2(t)] &= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} x_1(t-\tau)x_2(\tau)\,d\tau\right]e^{-j\omega t}\,dt\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1(t-\tau)x_2(\tau)e^{-j\omega t}\,d\tau dt\end{aligned}$$

As $x_1(t)$ and $x_2(t)$ are deemed to be absolutely integrable, they are bounded and hence the order of integration can be reversed. We can thus write

$$\begin{aligned}\mathcal{F}[x_1(t)\otimes x_2(t)] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1(t-\tau)x_2(\tau)e^{-j\omega t}\,dtd\tau\\ &= \int_{-\infty}^{\infty} x_2(\tau)e^{-j\omega\tau}\int_{-\infty}^{\infty} x_1(t-\tau)e^{-j\omega(t-\tau)}\,dtd\tau\end{aligned}$$

By applying the variable substitution $t = t' + \tau$ and then replacing t' by t, we get

$$\begin{aligned}\mathcal{F}[x_1(t)\otimes x_2(t)] &= \int_{-\infty}^{\infty} x_2(\tau)e^{-j\omega\tau}\left[\int_{-\infty}^{\infty} x_1(t')e^{-j\omega t'}\,dt'\right]d\tau\\ &= \int_{-\infty}^{\infty} x_2(\tau)e^{-j\omega\tau}\left[\int_{-\infty}^{\infty} x_1(t)e^{-j\omega t}\,dt\right]d\tau\\ &= \int_{-\infty}^{\infty} x_2(\tau)e^{-j\omega\tau}X_1(j\omega)\,d\tau\end{aligned}$$

and as $X_1(j\omega)$ is independent of τ, we can write

$$\begin{aligned}\mathcal{F}[x_1(t)\otimes x_2(t)] &= X_1(j\omega)\int_{-\infty}^{\infty} x_2(\tau)e^{-j\omega\tau}\,d\tau\\ &= X_1(j\omega)X_2(j\omega)\end{aligned}$$

The same result can be obtained by starting with Eq. (2.46a) (see Prob. 2.21). ▲

The above theorem is stating, in effect, that *the Fourier transform of the time convolution is equal to the product of the Fourier transforms of the two signals*. Equivalently, *the time convolution is equal to the inverse Fourier transform of the product of the Fourier transforms of the two signals.*

Therefore, if a Fourier transform $X(j\omega)$ can be factorized into two Fourier transforms $X_1(j\omega)$ and $X_2(j\omega)$, that is,

$$X(j\omega) = X_1(j\omega)X_2(j\omega)$$

whose inverse Fourier transforms $x_1(t)$ and $x_2(t)$ are known, then the inverse Fourier transform of the product $X(j\omega)$ can be deduced by evaluating the time convolution.

Theorem 2.15 Frequency Convolution

$$\boldsymbol{x_1(t)x_2(t) \leftrightarrow \frac{1}{2\pi} X_1(j\omega) \otimes X_2(j\omega)}$$

where

$$\boldsymbol{X_1(j\omega) \otimes X_2(j\omega) = \int_{-\infty}^{\infty} X_1(jv)X_2(j\omega - jv)\,dv} \tag{2.47a}$$

$$\boldsymbol{= \int_{-\infty}^{\infty} X_1(j\omega - jv)X_2(jv)\,dv} \tag{2.47b}$$

Proof The proof of this theorem would entail using the definition of the inverse Fourier transform and then reversing the order of integration as in the proof of Theorem 2.14. The second formula can be obtained from the first through a simple change of variable. (See Prob. 2.22, part (*b*).) ▲

Theorem 2.16 Parseval's Formula for Nonperiodic Signals

$$\boldsymbol{\int_{-\infty}^{\infty} |x(t)|^2\,dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(j\omega)|^2\,d\omega}$$

Proof From Theorem 2.15,

$$\int_{-\infty}^{\infty} x_1(t)x_2(t)e^{-j\omega t}\,dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_1(jv)X_2(j\omega - jv)\,dv$$

By letting $\omega \to 0$, then replacing v by ω, we have

$$\int_{-\infty}^{\infty} x_1(t)x_2(t)\,dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_1(j\omega)X_2(-j\omega)\,d\omega$$

Now if we assume that $x_1(t) = x(t)$ and $x_2(t) = x^*(t)$, then $X_2(-j\omega) = X^*(j\omega)$ (see Prob. 2.23, part (*b*)). Hence, from the above equation, we obtain

$$\int_{-\infty}^{\infty} x(t)x^*(t)\,dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)X^*(j\omega)\,d\omega$$

or

$$\int_{-\infty}^{\infty} |x(t)|^2\,dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(j\omega)|^2\,d\omega \quad \blacktriangle$$

If $x(t)$ represents a voltage or current waveform, the left-hand integral represents the total energy that would be delivered to a 1-Ω resistor, that is,

$$E_T = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(j\omega)|^2\, d\omega \tag{2.48}$$

and if $\omega = 2\pi f$ then the energy of the signal over a bandwidth of 1 Hz, say, with respect to the frequency range

$$f_0 - \frac{1}{2} < f < f_0 + \frac{1}{2}$$

can be obtained from Eq. (2.48) as

$$\begin{aligned}\Delta E_T &= \frac{1}{2\pi}\int_{-(f_0-\frac{1}{2})}^{(f_0+\frac{1}{2})} |X(j\omega)|^2\, d(2\pi f)\\ &\approx |X(j\omega_0)|^2 \int_{-(f_0-\frac{1}{2})}^{(f_0+\frac{1}{2})} df = |X(j\omega_0)|^2\end{aligned}$$

In effect, the quantity $|X(j\omega)|^2$ represents the energy density per unit bandwidth (in Hz) of the signal at frequency $f = \omega/2\pi$ (in Hz) and is often referred to as the *energy spectral density*. As a function of ω, $|X(j\omega)|^2$ is called the *energy spectrum* of $x(t)$.

Parseval's formula is the basic tool in obtaining a frequency-domain representation for random signals, as will be shown in Chap. 13.

The application of the above theorems is illustrated through the following examples.

Example 2.7 Show that

$$\frac{\sin \varpi t/2}{\pi t} \leftrightarrow p_{\varpi}(\omega)$$

where

$$p_{\varpi}(\omega) = \begin{cases} 1 & \text{for } |\omega| \le \varpi/2 \\ 0 & \text{otherwise} \end{cases}$$

■ **Solution**

From Example 2.5, we have

$$p_\tau(t) \leftrightarrow \frac{2\sin \omega\tau/2}{\omega}$$

where

$$p_\tau(t) = \begin{cases} 1 & \text{for } |t| \le \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

By using the symmetry theorem (Theorem 2.7), we get

$$\frac{2\sin\tau t/2}{t} \leftrightarrow 2\pi p_\tau(-\omega)$$

where

$$\begin{aligned} p_\tau(-\omega) &= \begin{cases} 1 & \text{for } |-\omega| \le \tau/2 \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} 1 & \text{for } |\omega| \le \tau/2 \\ 0 & \text{otherwise} \end{cases} \\ &= p_\tau(\omega) \end{aligned}$$

Now if we let $\tau = \varpi$, we get

$$\frac{\sin\varpi t/2}{\pi t} \leftrightarrow p_\varpi(\omega) \quad \blacksquare$$

where

$$p_\varpi(\omega) = \begin{cases} 1 & \text{for } |\omega| \le \varpi/2 \\ 0 & \text{otherwise} \end{cases}$$

Example 2.8 Obtain the Fourier transform of the signal shown in Fig. 2.9*a*.

■ **Solution**

From Fig. 2.9*a*, the given signal can be modeled as

$$x(t) = p_{\tau/2}\left(t + \tfrac{1}{4}\tau\right) - p_{\tau/2}\left(t - \tfrac{1}{4}\tau\right) \tag{2.49}$$

by using shifted copies of the pulse $p_{\tau/2}(t)$ which is obtained by replacing τ by $\tau/2$ in the pulse of Example 2.5.

On using the linearity and time-shifting theorems, we get

$$\begin{aligned} X(j\omega) &= \mathcal{F}\left[p_{\tau/2}\left(t + \tfrac{1}{4}\tau\right) - p_\tau\left(t - \tfrac{1}{4}\tau\right)\right] \\ &= \mathcal{F}p_{\tau/2}\left(t + \tfrac{1}{4}\tau\right) - \mathcal{F}p_{\tau/2}\left(t - \tfrac{1}{4}\tau\right) \\ &= e^{j\omega\tau/4}\mathcal{F}p_{\tau/2}(t) - e^{-j\omega\tau/4}\mathcal{F}p_{\tau/2}(t) \end{aligned}$$

and from Example 2.5, we deduce

$$X(j\omega) = (e^{j\omega\tau/4} - e^{-j\omega\tau/4})\mathcal{F}p_{\tau/2}(t) = \frac{4j\sin^2\omega\tau/4}{\omega} \tag{2.50}$$

Hence

$$p_{\tau/2}\left(t + \tfrac{1}{4}\tau\right) - p_{\tau/2}\left(t - \tfrac{1}{4}\tau\right) \leftrightarrow \frac{4j\sin^2\omega\tau/4}{\omega} \quad \blacksquare$$

and $$A(\omega) = \left| \frac{4\sin^2 \omega\tau/4}{\omega} \right| \qquad \text{and} \qquad \phi(\omega) = \begin{cases} \frac{1}{2}\pi & \text{for } \omega > 0 \\ -\frac{1}{2}\pi & \text{for } \omega < 0 \end{cases}$$

(See Fig. 2.9*b* and *c* for plots.)

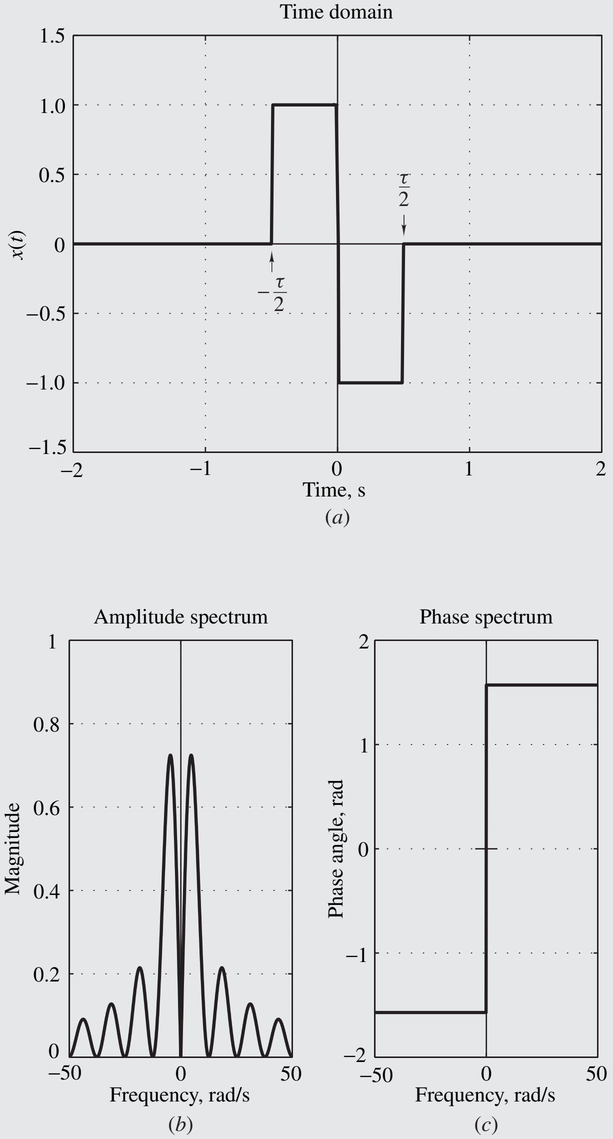


Figure 2.9 Frequency spectrum of the function in Eq. (2.49) (Example 2.8 with $\alpha = 0.4$): (*a*) Time-domain representation, (*b*) amplitude spectrum, (*c*) phase spectrum.

Example 2.9 Obtain the Fourier transform of the triangular pulse $q_\tau(t)$ shown in Fig. 2.10a.

Solution

From Fig. 2.10a, we have

$$q_\tau(t) = \begin{cases} 1 - \frac{2|t|}{\tau} & \text{for } |t| \le \tau/2 \\ 0 & \text{for } |t| > \tau/2 \end{cases}$$

We note that the given triangular pulse can be generated by performing the integration

$$q_\tau(t) = \int_{-\infty}^{t} \frac{2x(t)}{\tau}\,dt \tag{2.51}$$

where

$$x(t) = p_{\tau/2}\left(t + \tfrac{1}{4}\tau\right) - p_{\tau/2}\left(t - \tfrac{1}{4}\tau\right)$$

(see Example 2.8). If we differentiate both sides in Eq. (2.51), we get

$$\frac{dq_\tau(t)}{dt} = \frac{2x(t)}{\tau} \tag{2.52a}$$

If we apply the time-differentiation theorem (Theorem 2.11) with $k = 1$ to the left-hand side of Eq. (2.52a), we have

$$\mathcal{F}\left[\frac{dq_\tau(t)}{dt}\right] = j\omega Q_\tau(j\omega) \tag{2.52b}$$

On the other hand, if we apply the Fourier transform to the right-hand side of Eq. (2.52a), we get

$$\mathcal{F}\left[\frac{2x(t)}{\tau}\right] = \frac{2X(j\omega)}{\tau} \tag{2.52c}$$

Therefore, from Eqs. (2.52a)–(2.52c)

$$\mathcal{F}\left[\frac{dq_\tau(t)}{dt}\right] = \mathcal{F}\left[\frac{2x(t)}{\tau}\right]$$

or

$$Q_\tau(j\omega) = \frac{2X(j\omega)}{j\omega\tau}$$

and from Eq. (2.50), we get

$$Q_\tau(j\omega) = \frac{8\sin^2\omega\tau/4}{\omega^2\tau}$$

or

$$q_\tau(t) \leftrightarrow \frac{8\sin^2 \omega\tau/4}{\omega^2\tau} \quad \blacksquare$$

The amplitude spectrum is illustrated in Fig. 2.10*b*. The phase spectrum is zero for all frequencies since the signal is a real, even function of time (see Eqs. (2.36a) and (2.36b)).

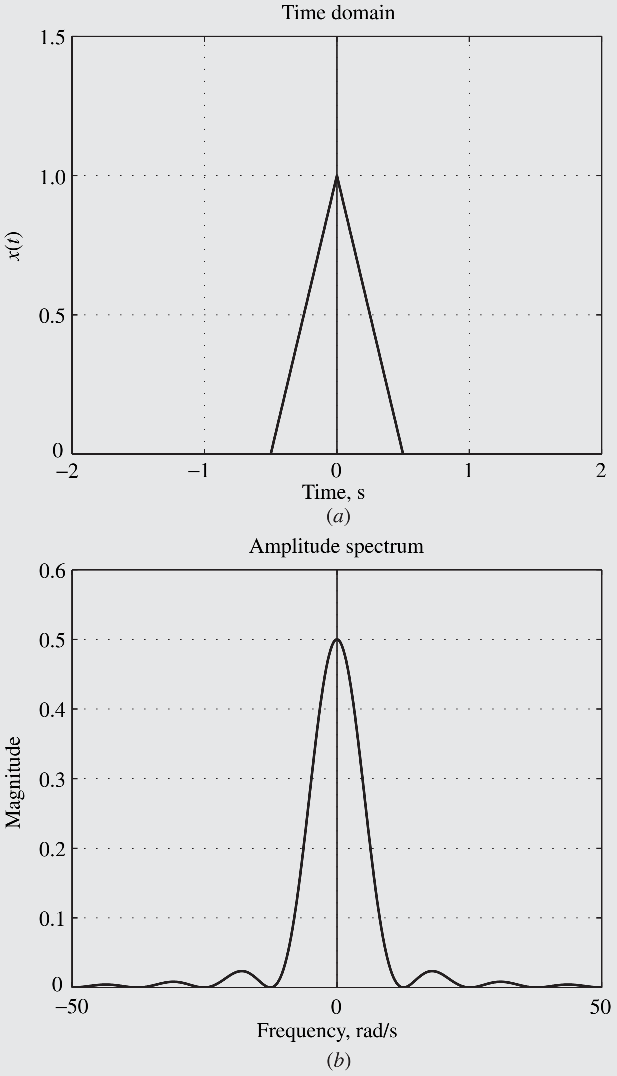


Figure 2.10 Frequency spectrum of triangular function (Example 2.9 with $\tau = 1.0$): (*a*) Time-domain representation, (*b*) amplitude spectrum.

Example 2.10 Obtain the Fourier transform of the decaying sinusoidal signal

$$x(t) = u(t)e^{-\alpha t} \sin \omega_0 t$$

(see Fig. 2.11*a*) where α and ω_0 are positive constants.

■ **Solution**

We can write

$$x(t) = u(t)e^{-\alpha t} \sin \omega_0 t = \frac{u(t)}{2j}\left(e^{j\omega_0 t} - e^{-j\omega_0 t}\right)e^{-\alpha t}$$

$$= \frac{u(t)}{2j}\left(e^{-(\alpha - j\omega_0)t} - e^{-(\alpha + j\omega_0)t}\right) \tag{2.53a}$$

From Example 2.6, we have

$$u(t)e^{-\alpha t} \leftrightarrow \frac{1}{\alpha + j\omega}$$

and if we replace α first by $\alpha - j\omega$ and then by $\alpha + j\omega$, we get

$$u(t)e^{-(\alpha - j\omega_0)t} \leftrightarrow \frac{1}{\alpha - j\omega_0 + j\omega} \tag{2.53b}$$

and

$$u(t)e^{-(\alpha + j\omega_0)t} \leftrightarrow \frac{1}{\alpha + j\omega_0 + j\omega} \tag{2.53c}$$

respectively. Now from Eqs. (2.53a)–(2.53c)

$$\frac{u(t)}{2j}\left(e^{-(\alpha - j\omega_0)t} - e^{-(\alpha + j\omega_0)t}\right)$$

$$\leftrightarrow \frac{1}{2j}\left[\frac{1}{\alpha - j\omega_0 + j\omega} - \frac{1}{\alpha + j\omega_0 + j\omega}\right]$$

or

$$u(t)e^{-\alpha t} \sin \omega_0 t \leftrightarrow \frac{\omega_0}{\alpha^2 + \omega_0^2 - \omega^2 + j2\alpha\omega} = \frac{\omega_0}{(a + j\omega)^2 + \omega_0^2} \quad ■$$

Hence, the amplitude and phase spectrums of the decaying sinusoidal signal are given by

$$A(\omega) = \frac{\omega_0}{\sqrt{(\alpha^2 + \omega_0^2 - \omega^2)^2 + 4\alpha^2\omega^2}}$$

and

$$\phi(\omega) = -\tan^{-1}\frac{2\alpha\omega}{\alpha^2 + \omega_0^2 - \omega^2}$$

respectively. (See Fig. 2.11*b* and *c* for the plots.)

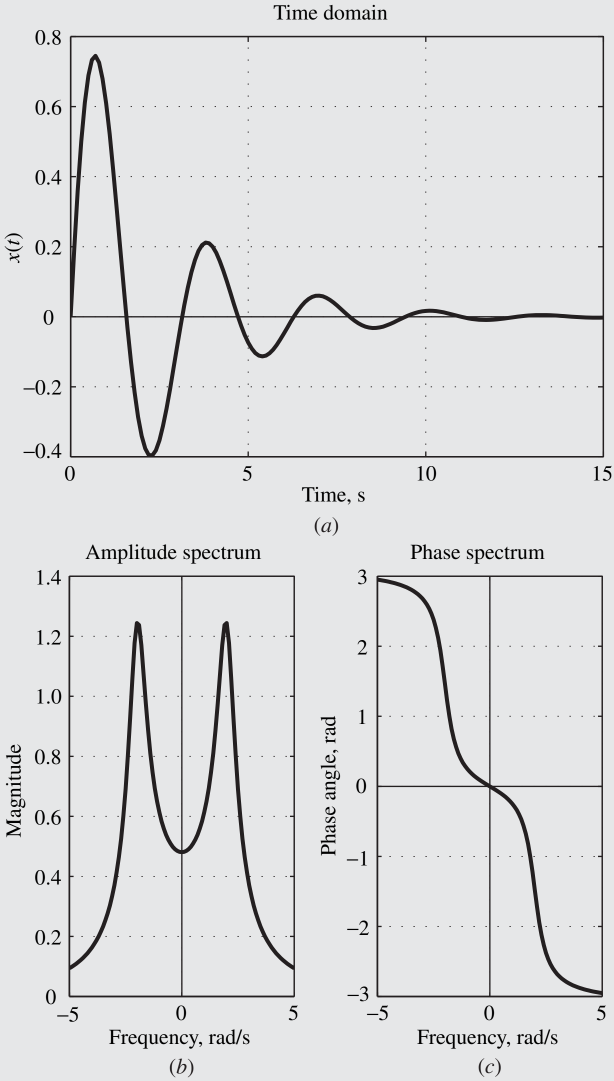


Figure 2.11 Frequency spectrum of continuous-time decaying sinusoidal signal (Example 2.10, $a = 0.4$, $\omega_0 = 2.0$): (*a*) Time-domain representation, (*b*) amplitude spectrum, (*c*) phase spectrum.

Example 2.11 Obtain the Fourier transform of the Gaussian function $x(t) = e^{-at^2}$ (see Fig. 2.12*a*).

■ **Solution**

A solution for this example found in Ref. [5] starts with the standard integral

$$\int_{-\infty}^{\infty} e^{-at^2}\, dt = \sqrt{\frac{\pi}{\alpha}}$$

which can be obtained from mathematical handbooks, for example, Ref. [7]. On differentiating both sides of this equation k times with respect to α, we can show that

$$\int_{-\infty}^{\infty} t^{2k} e^{-at^2}\, dt = \frac{1 \cdot 3 \cdots (2k-1)}{2^k} \sqrt{\frac{\pi}{\alpha^{2k+1}}} \tag{2.54a}$$

On the other hand, if we replace $e^{-j\omega t}$ by its series representation (see Eq. A.11a) in the definition of the Fourier transform, we get

$$\begin{aligned} X(j\omega) &= \int_{-\infty}^{\infty} x(t) e^{-j\omega t}\, dt = \int_{-\infty}^{\infty} x(t) \left[\sum_{k=0}^{\infty} \frac{(-j\omega t)^k}{k!} \right] dt \\ &= \sum_{k=0}^{\infty} \frac{(-j\omega)^k}{k!} \int_{-\infty}^{\infty} t^k x(t)\, dt = \sum_{k=0}^{\infty} \frac{(-j\omega)^k}{k!} m_k \end{aligned} \tag{2.54b}$$

where

$$m_k = \int_{-\infty}^{\infty} t^k x(t)\, dt$$

is the kth moment of $x(t)$ (see Theorem 2.13). As $x(t)$ is an even function, the moments for odd k are zero and hence Eq. (2.54b) can be expressed as

$$\begin{aligned} X(j\omega) &= m_0 + \frac{(-j\omega)^2}{2!} m_2 + \frac{(-j\omega)^4}{4!} m_4 + \cdots \\ &= \sum_{k=0}^{\infty} \frac{(-j\omega)^{2k}}{(2k)!} m_{2k} \end{aligned}$$

where

$$m_{2k} = \int_{-\infty}^{\infty} t^{2k} e^{-\alpha t^2}\, dt = \frac{1 \cdot 3 \cdots (2k-1)}{2^k} \sqrt{\frac{\pi}{\alpha^{2k+1}}}$$

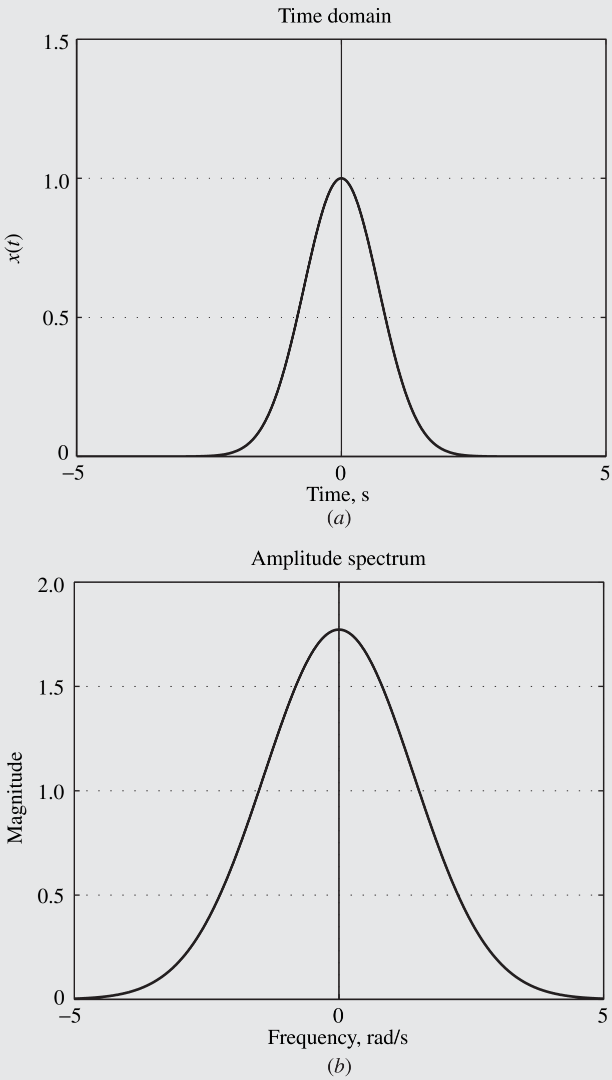


Figure 2.12 Frequency spectrum of continuous-time Gaussian function: (Example 2.11, $\alpha = 1.0$): (*a*) Time-domain representation, (*b*) amplitude spectrum.

according to Eq. (2.54a) or

$$X(j\omega) = \sum_{k=0}^{\infty} \frac{(-j\omega)^{2k}}{(2k)!} \cdot \frac{1 \cdot 3 \cdots (2k-1)}{2^k} \sqrt{\frac{\pi}{\alpha^{2k+1}}}$$

$$= \sqrt{\frac{\pi}{\alpha}} \sum_{k=0}^{\infty} \frac{1 \cdot 3 \cdots (2k-1)(-j\omega)^{2k}}{(2k)!(2\alpha)^k}$$

The summation in the above equation is actually the series of $e^{-\omega^2/4\alpha}$, as can be readily verified and, therefore,

$$X(j\omega) = \sqrt{\frac{\pi}{\alpha}}e^{-\omega^2/4\alpha}$$

or

$$e^{-at^2} \leftrightarrow \sqrt{\frac{\pi}{\alpha}}e^{-\omega^2/4\alpha} \quad \blacksquare$$

The Gaussian function and its Fourier transform are plotted in Fig. 2.12 and as can be seen, the frequency-domain function has the same general form as the time-domain function. (See Eq. (6.26) for another transform pair that has this property.)

The Fourier transform pairs obtained in this chapter are summarized in Table 2.1. We have not dealt with impulse functions or periodic signals so far because these types of signals require special attention. It turns out that the applicability of the Fourier transform to periodic signals relies critically on the definition of impulse functions. Impulse functions and periodic signals are, of course, very important in DSP and they will be examined in detail in Chap. 6.

Table 2.1 Standard Fourier transforms

$x(t)$	$X(j\omega)$
$p_\tau(t) = \begin{cases} 1 & \text{for } \|t\| \le \tau/2 \\ 0 & \text{for } \|t\| > \tau/2 \end{cases}$	$\dfrac{2\sin\omega\tau/2}{\omega}$
$\dfrac{\sin\varpi t/2}{\pi t}$	$p_\varpi(\omega) = \begin{cases} 1 & \text{for } \|\omega\| \le \varpi/2 \\ 0 & \text{for } \|\omega\| > \varpi/2 \end{cases}$
$q_\tau(t) = \begin{cases} 1 - \dfrac{2\|t\|}{\tau} & \text{for } \|t\| \le \tau/2 \\ 0 & \text{for } \|t\| > \tau/2 \end{cases}$	$\dfrac{8\sin^2\omega\tau/4}{\tau\omega^2}$
$\dfrac{4\sin^2\varpi t/4}{\pi\varpi t^2}$	$q_\varpi(\omega) = \begin{cases} 1 - \dfrac{2\|\omega\|}{\varpi} & \text{for } \|\omega\| \le \varpi/2 \\ 0 & \text{for } \|\omega\| > \varpi/2 \end{cases}$
$e^{-\alpha t^2}$	$\sqrt{\frac{\pi}{\alpha}}e^{-\omega^2/4\alpha}$
$\dfrac{1}{\sqrt{4\alpha\pi}}e^{-t^2/4\alpha}$	$e^{-\alpha\omega^2}$
$u(t)e^{-\alpha t}$	$\dfrac{1}{a + j\omega}$
$u(t)e^{-\alpha t}\sin\omega_0 t$	$\dfrac{\omega_0}{(a + j\omega)^2 + \omega_0^2}$

REFERENCES

[1] W. Kaplan, *Operational Methods for Linear Systems*, 3rd ed., Reading, MA: Addison-Wesley, 1984.

[2] R. J. Schwarz and B. Friedland, *Linear Systems*, New York: McGraw-Hill, 1965.

[3] H. S. Carslaw, *Fourier Series*, New York: Dover, 1930.

[4] C. R. Wylie, Jr. *Advance Engineering Mathematics*, 3rd ed., New York: McGraw-Hill, 1966.

[5] A. Papoulis, *The Fourier Integral and Its Applications*, New York: McGraw-Hill, 1962.

[6] W. Kaplan, *Advanced Calculus*, 3rd ed., Reading, MA: Addison-Wesley, 1962.

[7] M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*, New York: McGraw-Hill, 1965.

PROBLEMS

2.1. Derive Eq. (2.4).

2.2. Derive Eqs. (2.12c) and (2.12d).

2.3. A periodic signal $\tilde{x}(t)$ is described by Eq. (2.1) with

$$x(t) = \begin{cases} 1 & \text{for } -\tau_0/2 < t < -\tau_0/4 \\ 2 & \text{for } -\tau_0/4 \le t < \tau_0/4 \\ 1 & \text{for } \tau_0/4 \le t < \tau_0/2 \end{cases}$$

(a) Obtain the Fourier series of $\tilde{x}(t)$ in the form of Eq. (2.3).
(b) Express the Fourier series in the form of Eq. (2.9).
(c) Express the Fourier series in the form of Eq. (2.13b).
(d) Obtain the amplitude and phase spectrums of $\tilde{x}(t)$.

2.4. A periodic signal $\tilde{x}(t)$ is described by Eq. (2.1) with

$$x(t) = \begin{cases} 0 & \text{for } -\tau_0/2 < t < -3\tau_0/8 \\ 1 & \text{for } -3\tau_0/8 \le t < -\tau_0/4 \\ 2 & \text{for } -\tau_0/4 \le t < \tau_0/4 \\ 1 & \text{for } \tau_0/4 \le t < 3\tau_0/8 \\ 0 & \text{for } 3\tau_0/8 \le t \le \tau_0/2 \end{cases}$$

(a) Obtain the Fourier series of $\tilde{x}(t)$ in the form of Eq. (2.3).
(b) Express the Fourier series in the form of Eq. (2.9).
(c) Express the Fourier series in the form of Eq. (2.13b).
(d) Obtain the amplitude and phase spectrums of $\tilde{x}(t)$.

2.5. A periodic signal $\tilde{x}(t)$ is described by Eq. (2.1) with

$$x(t) = \begin{cases} 1 & \text{for } -\tau_0/2 < t \le -\tau/2 \\ 0 & \text{for } -\tau/2 < t < \tau/2 \\ 1 & \text{for } \tau/2 \le t \le \tau_0/2 \end{cases}$$

where $\tau < \tau_0$.
(a) Obtain the Fourier series of $\tilde{x}(t)$ in the form of Eq. (2.3).
(b) Express the Fourier series in the form of Eq. (2.9).
(c) Express the Fourier series in the form of Eq. (2.13b).
(d) Obtain the amplitude and phase spectrums of $\tilde{x}(t)$.

2.6. A periodic signal $\tilde{x}(t)$ is described by Eq. (2.1) with

$$x(t) = \begin{cases} 1 & \text{for } -\tau_0/2 < t \le -\tau/2 \\ 0 & \text{for } -\tau/2 < t < \tau/2 \\ -1 & \text{for } \tau/2 \le t \le \tau_0/2 \end{cases}$$

where $\tau < \tau_0$.

(*a*) Obtain the Fourier series of $\tilde{x}(t)$ in the form of Eq. (2.3).

(*b*) Express the Fourier series in the form of Eq. (2.9).

(*c*) Express the Fourier series in the form of Eq. (2.13b).

(*d*) Obtain the amplitude and phase spectrums of $\tilde{x}(t)$.

2.7. A periodic signal $\tilde{x}(t)$ is described by Eq. (2.1) with

$$x(t) = \begin{cases} 0 & \text{for } -\tau_0/2 < t < -\tau_2 \\ 1 & \text{for } -\tau_2 \le t \le -\tau_1 \\ 0 & \text{for } -\tau_1 < t < \tau_1 \\ 1 & \text{for } \tau_1 \le t \le \tau_2 \\ 0 & \text{for } \tau_2 < t \le \tau_0/2 \end{cases}$$

where $\tau_1 < \tau_2 < \tau_0/2$.

(*a*) Obtain the Fourier series in the form of Eq. (2.3).

(*b*) Obtain the amplitude and phase spectrums of $\tilde{x}(t)$.

2.8. A periodic signal $\tilde{x}(t)$ is described by Eq. (2.1) with

$$x(t) = \begin{cases} 0 & \text{for } -\tau_0/2 < t < -\tau_2 \\ -1 & \text{for } -\tau_2 \le t \le -\tau_1 \\ 0 & \text{for } -\tau_1 < t < \tau_1 \\ 1 & \text{for } \tau_1 \le t \le \tau_2 \\ 0 & \text{for } \tau_2 < t \le \tau_0/2 \end{cases}$$

where $\tau_1 < \tau_2 < \tau_0/2$.

(*a*) Obtain the Fourier series in the form of Eq. (2.3).

(*b*) Obtain the amplitude and phase spectrums of $\tilde{x}(t)$.

2.9. A periodic signal $\tilde{x}(t)$ is described by Eq. (2.1) with

$$x(t) = \begin{cases} 1 & \text{for } -\tau_0/2 < t < -\tau_2 \\ 0 & \text{for } -\tau_2 \le t \le -\tau_1 \\ 1 & \text{for } -\tau_1 < t < \tau_1 \\ 0 & \text{for } \tau_1 \le t \le \tau_2 \\ 1 & \text{for } \tau_2 < t \le \tau_0/2 \end{cases}$$

where $\tau_1 < \tau_2 < \tau_0/2$.

(*a*) Obtain the Fourier series in the form of Eq. (2.3).

(*b*) Obtain the amplitude and phase spectrums of $\tilde{x}(t)$.

2.10. A periodic signal is given by

$$\tilde{x}(t) = \cos^2 \omega t + \cos^4 \omega t$$

(*a*) Obtain the Fourier series of $\tilde{x}(t)$ in the form of a linear combination of cosines.

(*b*) Obtain the amplitude and phase spectrums of $\tilde{x}(t)$.

2.11. A periodic signal is given by

$$\tilde{x}(t) = \tfrac{1}{2} + \sin \omega t + \tfrac{1}{4} \sin^2 \omega t + \cos^4 \omega t$$

(a) Obtain the Fourier series of $\tilde{x}(t)$ in the form of a linear combination of sines.

(b) Obtain the amplitude and phase spectrums of $\tilde{x}(t)$.

2.12. Find the Fourier series of

(a) $x(t) = \alpha t \qquad \text{for } -\tau_0/2 \le t \le \tau/2$

(b) $x(t) = \begin{cases} -\alpha t & \text{for } -\tau_0/2 \le t < 0 \\ \alpha t & \text{for } 0 \le t \le \tau_0/2 \end{cases}$

2.13. Find the Fourier series of

(a) $x(t) = |\cos \omega_0 t| \qquad \text{for } -\tau_0/2 \le t \le \tau_0/2$

where $\omega_0 = 2\pi/\tau_0$.

(b) $x(t) = \begin{cases} 0 & \text{for } -\tau_0/2 \le t < 0 \\ |\sin \omega_0 t| & \text{for } 0 \le t \le \tau_0/2 \end{cases}$

where $\omega_0 = 2\pi/\tau_0$.

2.14. Find the Fourier series of

(a) $x(t) = jt \qquad \text{for } -\tau_0/2 \le |t| \le \tau_0/2$

(b) $x(t) = j|t| \qquad \text{for } -\tau_0/2 \le |t| \le \tau_0/2$

2.15. Find the Fourier series of

(a) $x(t) = t/\tau_0 + 1/2 \quad \text{for } -\tau_0/2 \le t \le \tau_0/2$

(b) $x(t) = \begin{cases} 0 & \text{for } -\tau_0/2 \le t < -\tau_0/4 \\ e^{\omega_0 t} & \text{for } -\tau_0/4 \le t < 0 \\ e^{-\omega_0 t} & \text{for } 0 \le t < \tau_0/4 \\ 0 & \text{for } \tau_0/4 \le t \le \tau_0/2 \end{cases}$

2.16. Assuming that $x(t)$ is a real signal which can be either an even or odd function of time, show that

(a) $X(-j\omega) = -X^*(j\omega)$

(b) $|X(-j\omega)| = |X(j\omega)|$

(c) $\arg X(-j\omega) = -\arg X(j\omega)$

2.17. Assuming that $x(t)$ is purely imaginary show that

(a)

$$\Re e\, X(j\omega) = \int_{-\infty}^{\infty} \Im m\, x(t) \sin \omega t \, dt \qquad \text{and} \qquad \Im m\, X(j\omega) = \int_{-\infty}^{\infty} \Re e\, x(t) \cos \omega t \, dt$$

(b) Assuming that $x(t)$ is purely imaginary and an even function of time, show that $\Re e\, X(j\omega)$ is an odd function and $\Im m\, X(j\omega)$ is an even function of frequency.

(c) Assuming that $x(t)$ is purely imaginary and an odd function of time, show that $\Re e\, X(j\omega)$ is an even function and $\Im m\, X(j\omega)$ is an odd function of frequency.

2.18. (a) Prove Theorem 2.6 (linearity) for the Fourier transform.

(b) Repeat part (a) for the inverse Fourier transform.

2.19. (a) Prove Theorem 2.9 (time shifting).

(b) Prove Theorem 2.10 (frequency shifting).

2.20. Show that

$$\int_{-\infty}^{\infty} x_1(\tau)x_2(t-\tau)\,d\tau = \int_{-\infty}^{\infty} x_1(t-\tau)x_2(\tau)\,d\tau$$

2.21. Prove Theorem 2.14 (time convolution) starting with Eq. (2.46a).

2.22. (*a*) Prove Theorem 2.15 (frequency convolution) starting with Eq. (2.47a).

(*b*) Show that Eq. (2.47b) is equivalent to Eq. (2.47a).

2.23. A complex signal $x_2(t)$ is equal to the complex conjugate of signal $x(t)$. Show that

(*a*) $X(j\omega) = X^*(-j\omega)$

(*b*) $X(-j\omega) = X^*(j\omega)$

2.24. (*a*) Find the Fourier transform of $x(t) = p_\tau(t - \tau/2)$ where $p_\tau(t)$ is a pulse of unity amplitude and width τ.

(*b*) Find the Fourier transform of

$$x(t) = \begin{cases} 1 & \text{for } -\tau_0/2 \le t < -\tau_0/4 \\ 2 & \text{for } -\tau_0/4 \le t < \tau_0/4 \\ 1 & \text{for } \tau_0/4 \le t < \tau_0/2 \\ 0 & \text{otherwise} \end{cases}$$

(*c*) Find the amplitude and phase spectrums for the signal in part (*b*).

2.25. (*a*) Find the Fourier transform of

$$x_1(t) = [u(t + \tau/2) - u(t - \tau/2)]$$

where $u(t)$ is the continuous-time unit-step function defined as

$$u(t) = \begin{cases} 1 & \text{for } t \ge 0 \\ 0 & \text{for } t < 0 \end{cases}$$

(*b*) Sketch the waveform of

$$x_2(t) = \sum_{n=-\infty}^{\infty} x_1(t - n\tau)$$

(*c*) Using the result in part (*a*), find the Fourier transform of $x_2(t)$.

2.26. Find the Fourier transform of $x(t) = u(t - 4.5T) - u(t - 9.5T)$.

(*b*) Obtain the amplitude and phase spectrums of $x(t)$.

2.27. (*a*) Find the Fourier transform of

$$x(t) = \begin{cases} (1 + \cos\omega_0 t)/2 & \text{for } |t| \le \tau_0/2 \\ 0 & \text{otherwise} \end{cases}$$

where $\omega_0 = 2\pi/\tau_0$.

(*b*) Obtain the amplitude and phase spectrums of $x(t)$.

2.28. (*a*) Find the Fourier transform of $x(t) = u(t)e^{-at}\cos\omega_0 t$.

(*b*) Obtain the amplitude and phase spectrums of $x(t)$.

2.29. (*a*) Find the Fourier transform of

$$x(t) = \begin{cases} e^{-at}\cosh\omega_0 t & \text{for } 0 \le t \le 1 \\ 0 & \text{otherwise} \end{cases}$$

(*b*) Obtain the amplitude and phase spectrums of $x(t)$.
(See Prob. 2.25 for the definition of $u(t)$.)

2.30. (*a*) Find the Fourier transform of

$$x(t) = \begin{cases} \sin\omega_0 t & \text{for } -\tau_0/4 \le t \le \tau_0/4 \\ 0 & \text{otherwise} \end{cases}$$

where $\omega_0 = 2\pi/\tau_0$

(*b*) Obtain the amplitude and phase spectrums of $x(t)$.

2.31. (*a*) Find the Fourier transform of

$$x(t) = \begin{cases} e^{-at}\sinh\omega_0 t & \text{for } -1 \le t \le 1 \\ 0 & \text{otherwise} \end{cases}$$

(*b*) Obtain the amplitude and phase spectrums of $x(t)$.

2.32. Find the Fourier transforms of

(*a*) $x(t) = \begin{cases} |\cos\omega_0 t| & \text{for } -\tau_0/2 \le t \le \tau_0/2 \\ 0 & \text{otherwise} \end{cases}$

where $\omega_0 = 2\pi/\tau_0$

(*b*) $x(t) = \begin{cases} |\sin\omega_0 t| & \text{for } 0 \le t \le \tau_0/2 \\ 0 & \text{otherwise} \end{cases}$

where $\omega_0 = 2\pi/\tau_0$

2.33. Find the Fourier transforms of

(*a*) $x(t) = \begin{cases} 1 & \text{for } -\tau_2 \le t \le -\tau_1 \\ 1 & \text{for } \tau_1 \le t \le \tau_2 \\ 0 & \text{otherwise} \end{cases}$

where τ_1 and τ_2 are positive constants and $\tau_1 < \tau_2$

(*b*) $x(t) = \begin{cases} e^{\omega_0 t} & \text{for } -\tau_0/4 \le t < 0 \\ e^{-\omega_0 t} & \text{for } 0 \le t < \tau_0/4 \\ 0 & \text{otherwise} \end{cases}$

2.34. (*a*) Using integration by parts, show that

$$\int \alpha t e^{\beta t}\,dt = \alpha(\beta t - 1)e^{\beta t}/\beta^2$$

(*b*) Using the result in part (*a*) find the Fourier transform of

$$x(t) = \begin{cases} \alpha t & \text{for } -\tau_0/2 \le t \le \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

(*c*) Find the Fourier transform of

$$x(t) = \begin{cases} -\alpha t & \text{for } -\tau_0/2 \le t < 0 \\ \alpha t & \text{for } 0 \le t \le \tau_0/2 \\ 0 & \text{otherwise} \end{cases}$$

2.35. Find the Fourier transforms of

$$(a)\ \ x(t)=\begin{cases} t/\tau_0+1/2 & \text{for } -\tau_0/2\le t\le \tau_0/2 \\ 0 & \text{otherwise} \end{cases}$$

$$(b)\ \ x(t)=\begin{cases} 1+t & \text{for } \tau_0/2\le t<0 \\ 1-t & \text{for } 0\le t\le \tau_0/2 \\ 0 & \text{otherwise} \end{cases}$$

2.36. Find the Fourier transforms of

$$(a)\ \ x(t)=\begin{cases} jt & \text{for } -\tau_0/2\le |t|\le \tau_0/2 \\ 0 & \text{otherwise} \end{cases}$$

$$(b)\ \ x(t)=\begin{cases} j|t| & \text{for } -\tau_0/2\le |t|\le \tau_0/2 \\ 0 & \text{otherwise} \end{cases}$$

2.37. Obtain the Fourier transforms of the following:

(a) $x(t)=e^{-\alpha t}\cos^2\omega_0 t$
where $\alpha>0$

(b) $x(t)=\cos at^2$

CHAPTER 3

THE *Z* TRANSFORM

3.1 INTRODUCTION

Chapter 2 has dealt with the Fourier series and transform. It has shown that through these mathematical tools, spectral representations can be obtained for a given periodic or nonperiodic continuous-time signal in terms of a frequency spectrum, which is composed of the amplitude and phase spectrums. Analogous spectral representations are also possible for discrete-time signals. The counterpart of the Fourier transform for discrete-time signals is the z transform [1]. The Fourier transform will convert a real continuous-time signal into a function of complex variable $j\omega$. Similarly, the z transform will convert a real discrete-time signal into a function of a complex variable z. The transform name is based on nothing more profound than the consistent use of the letter z for the complex variable involved over the years.

The z transform, like the Fourier transform, comes along with an inverse transform, namely, the inverse z transform. As a consequence, a signal can be readily recovered from its z transform. The availability of an inverse makes the z transform very useful for the representation of digital filters and discrete-time systems in general. Though the most basic representation of discrete-time systems is in terms of difference equations, as will be shown in Chap. 4, through the use of the z transform difference equations can be reduced to algebraic equations which are much easier to handle.

In this chapter, the z transform is first defined as an independent mathematical entity and it is immediately shown that it is actually a particular type of Laurent series. The inverse z transform is then introduced as a means of recovering the discrete-time signal from its z transform. This turns out to be an exercise in constructing Laurent series. The properties of the z transform are then described through a number of fundamental theorems as was done for the Fourier transform in Chap. 2. The z transform is then used for the representation of some typical discrete-time signals.

The chapter concludes with the application of the z transform as a tool for the spectral representation of discrete-time signals. The application of the z transform for the representation of digital filters and discrete-time systems in general will be treated in Chap. 5 and certain fundamental interrelations between the z transform and the Fourier transform will be investigated in Chap. 6.

3.2 DEFINITION OF Z TRANSFORM

Given an arbitrary discrete-time signal that satisfies the conditions

$$\text{(i)} \qquad x(nT) = 0 \qquad \text{for } n < -N_1$$

$$\text{(ii)} \qquad |x(nT)| \le K_1 \qquad \text{for } -N_1 \le n < N_2$$

$$\text{(iii)} \qquad |x(nT)| \le K_2 r^n \qquad \text{for } n \ge N_2$$

where $N_1,\ N_2$ are positive integers and $K_1,\ K_2$, and r are positive constants, the infinite series

$$X(z) = \sum_{n=-\infty}^{\infty} x(nT) z^{-n} \tag{3.1}$$

can be constructed where z is a complex variable. In mathematical terms, this is a Laurent series (see Sec. A.6) but in the digital signal processing (DSP) literature it is referred to as *the z transform of $x(nT)$*. As will be shown in Sec. 3.4, this turns out to be a unique representation of $x(nT)$ for all the values of z for which it converges.

Infinite series are not convenient to work with in practice but for most well- behaved discrete-time signals that can be represented in terms of analytical expressions, the z transform can be expressed as a *rational function* of z of the form

$$X(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{M} a_i z^{M-i}}{z^N + \sum_{i=1}^{N} b_i z^{N-i}} \tag{3.2a}$$

By factorizing the numerator and denominator polynomials, namely, $N(z)$ and $D(z)$, $X(z)$ can be put in the form

$$X(z) = \frac{N(z)}{D(z)} = H_0 \frac{\prod_{i=1}^{M}(z - z_i)}{\prod_{i=1}^{N}(z - p_i)} \tag{3.2b}$$

where z_i and p_i are the zeros and poles of $X(z)$. Thus the z transform of a discrete-time signal can be represented by a zero-pole plot. For example, the z transform

$$X(z) = \frac{(z^2 - 4)}{z(z^2 - 1)(z^2 + 4)} = \frac{(z - 2)(z + 2)}{z(z - 1)(z + 1)(z - j2)(z + j2)} \tag{3.3}$$

can be represented by the zero-pole plot shown in Fig. 3.1.

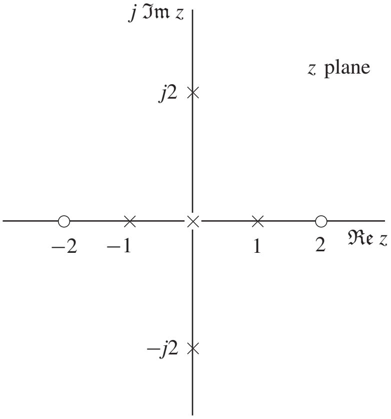


Figure 3.1 Zero-pole plot of z transform $X(z)$ in Eq. (3.3).

3.3 CONVERGENCE PROPERTIES

The infinite series in Eq. (3.1) is meaningful if it converges and, as in the case of the Fourier transform, convergence theorems exist that specify the circumstances under which the series converges. Two such theorems pertaining to absolute and uniform convergence are examined next. An infinite series is said to *converge absolutely* if the sum of the magnitudes of its terms has a finite value. An infinite series that involves an independent complex variable is said to *converge uniformly* in a given region of convergence if it converges absolutely everywhere in that region.

Theorem 3.1 Absolute Convergence ***If***

(i) $\boldsymbol{x(nT) = 0 \qquad \text{for } n < -N_1}$

(ii) $\boldsymbol{|x(nT)| \le K_1 \qquad \text{for } -N_1 \le n < N_2}$

(iii) $\boldsymbol{|x(nT)| \le K_2 r^n \qquad \text{for } n \ge N_2}$

where N_1 and N_2 are positive constants and r is the smallest positive constant that will satisfy condition (iii), then the z transform as defined in Eq. (3.1) exists and converges absolutely (see Theorem A.2) if and only if

$$\boldsymbol{r < |z| < R_\infty \qquad \textit{with } R_\infty \to \infty} \tag{3.4}$$

Proof If we let $z = \rho e^{j\theta}$, we can write

$$\begin{aligned}\sum_{n=-\infty}^{\infty} |x(nT)z^{-n}| &= \sum_{n=-\infty}^{\infty} |x(nT)| \cdot |z^{-n}| \\ &= \sum_{n=-\infty}^{\infty} |x(nT)| \cdot |\rho^{-n} e^{jn\theta}|\end{aligned}$$

Noting that the magnitude of $\rho^{-n}e^{jn\theta}$ is simply ρ^{-n} (see Eq. (A.13b)) and then substituting conditions (i) to (iii) of the theorem in the above equation, we get

$$\sum_{n=-\infty}^{\infty} |x(nT)z^{-n}| \le \sum_{n=-N_1}^{N_2-1} K_1\rho^{-n} + \sum_{n=N_2}^{\infty} K_2\left(\frac{r}{\rho}\right)^n$$

$$\le K_1 \sum_{n=-N_1}^{N_2-1} \rho^{-n} + K_2 \sum_{n=N_2}^{\infty} \left(\frac{r}{\rho}\right)^n \tag{3.5}$$

The first term at the right-hand side is the sum of a finite number of negative powers of ρ and since ρ is implicitly assumed to be finite, the first term is finite. The second term is the sum of a geometric series and if $\rho > r$, that is, $r/\rho < 1$, it is finite by virtue of the ratio test (see Theorem A.3). Hence

$$\sum_{n=-\infty}^{\infty} |x(nT)z^{-n}| \le K_0$$

where K_0 is finite, i.e., $X(z)$ converges absolutely. If $\rho < r$, then $r/\rho > 1$ and $(r/\rho)^n \to \infty$ as $n \to \infty$; consequently, the right-hand summation in Eq. (3.5) becomes infinite. If $\rho = r$, then $(r/\rho)^n = 1$ for all n. However, the right-hand summation in Eq. (3.5) entails an infinite number of ones and again it is infinite. In effect, $X(z)$ converges absolutely if $\rho > r$ and diverges if $\rho \le r$.

There is one more situation that needs to be taken into account before the proof can be considered complete, namely, the behavior of $X(z)$ as $z \to \infty$. If $x(nT) \ne 0$ for one or more negative values of n, then

$$\lim_{z\to\infty} \sum_{n=-\infty}^{\infty} x(nT)z^{-n} \to \infty$$

That is, $X(z)$ diverges if $z = \infty$ and, therefore, it converges if and only if

$$r < |z| < \infty \qquad \text{or} \qquad r < |z| < R_\infty \qquad \text{with } R_\infty \to \infty$$

In terms of the usual mathematical language, this is the *necessary and sufficient condition* for absolute convergence. ▲

Summarizing the essence of the above theorem, if $x(nT)$ is bounded by the shaded region in Fig. 3.2*a*, then its z transform converges absolutely if and only if z is located in the shaded region of the z plane depicted in Fig. 3.2*b* where $R_\infty \to \infty$. The area between the two circles is referred to as an *annulus* and the radius of the inner circle, namely, r, as the *radius of convergence* of the function since the inner circle separates the regions of convergence and divergence (see Sec. A.5).

Theorem 3.2 Uniform Convergence ***$X(z)$ converges uniformly and is analytic in the region defined by Eq. (3.4).*** ▲

This theorem follows readily from Theorem 3.1. Since $X(z)$ converges absolutely at any point in the region defined by Eq. (3.4), it has a limit and a derivative at any point in the region of

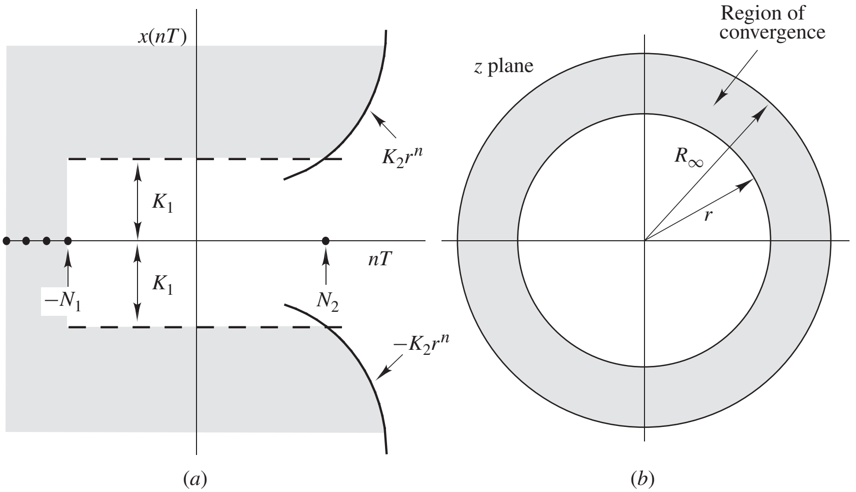


Figure 3.2 Convergence of z transform: (a) Bounds in time domain, (b) region of convergence in z domain.

convergence. Therefore, $X(z)$ is analytic in the annulus of Eq. (3.4). In addition, $X(z)$ converges uniformly in that annulus, which is a way of saying that the convergence $X(z)$ is independent of z (see Sec. A.5).

3.4 THE Z TRANSFORM AS A LAURENT SERIES

If we compare the series in Eq. (3.1) with the Laurent series in Eq. (A.49), we note that the z transform is a Laurent series with

$$X(z) = F(z) \qquad x(nT) = a_n \qquad a = 0$$

Therefore, the z transform inherits properties (a) to (d) in Theorem A.4.

From property (a), if $X(z)$, for example, the function represented by the zero-pole plot in Fig. 3.3a is analytic on two concentric circles C_1 and C_2 with center at the origin and in the area between them as depicted in Fig. 3.3b, then it can be represented by a series of the type shown in Eq. (3.1) where $x(nT)$ is given by the contour integral

$$x(nT) = \frac{1}{2\pi j} \oint_\Gamma X(z) z^{n-1}\, dz \tag{3.6}$$

The contour of integration Γ is a closed contour in the counterclockwise sense enclosing all the singularities of $X(z)$ inside the inner circle, i.e., C_1.

From property (b), a Laurent series of $X(z)$ converges and represents $X(z)$ in the open annulus obtained by continuously increasing the radius of C_2 and decreasing the radius of C_1 until each of C_2 and C_1 reaches one or more singularities of $X(z)$, as shown in Fig. 3.3c.

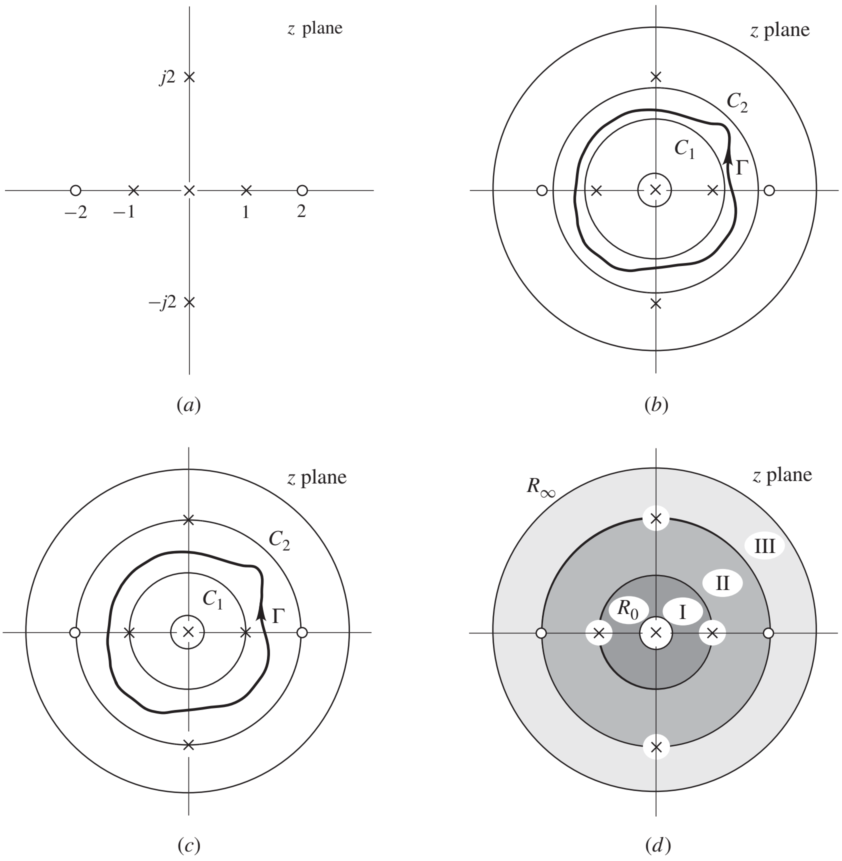


Figure 3.3 Laurent series of $X(z)$ with center at the origin of the z plane: (*a*) Zero-pole plot of $X(z)$, (*b*), (*c*), and (*d*) Properties (*a*), (*b*), and (*c*), respectively, of the Laurent series (see Theorem A.4).

From property (*c*), $X(z)$ can have several, possibly many, annuli of convergence about the origin; and from property (*d*), the Laurent series for a given annulus of convergence is unique. For example, function $X(z)$ given by Eq. (3.3) has three distinct annuli of convergence, namely,

$$A_{\mathrm{I}} = \{z : R_0 < |z| < 1\}$$

$$A_{\mathrm{II}} = \{z : 1 < |z| < 2\}$$

$$A_{\mathrm{III}} = \{z : 2 < |z| < R_\infty\}$$

as illustrated in Fig. 3.3*d* where $R_0 \to 0$ and $R_\infty \to \infty$, and a unique Laurent series can be obtained for each one of them.

3.5 INVERSE *Z* TRANSFORM

According to Theorem 3.1, the z transform of a discrete-time signal is a series that converges in the annulus defined by Eq. (3.4), that is,

$$r < |z| < R_\infty \qquad \text{with } R_\infty \to \infty$$

where r is specified in condition (iii) of the theorem. On the other hand, the Laurent theorem states that function $X(z)$ can have several Laurent series, possibly many, about the origin that converge in different annuli but each one is unique to its annulus of convergence.

The only annulus of convergence that is consistent with the annulus in Eq. (3.4) is the *outermost annulus* of $X(z)$, which is defined as

$$R < |z| < R_\infty \qquad \text{with } R_\infty \to \infty \tag{3.7}$$

where R is the radius of a circle passing through the most distant singularity of $X(z)$ from the origin. Therefore, we must have $r = R$, that is, if the pole locations of $X(z)$ are known, the template that bounds the discrete-time signal in Fig. 3.2*a* can be constructed and if the template in Fig. 3.2*a* is known, then the radius of the most distant pole of $X(z)$ from the origin can be deduced.

On the basis of the above discussion, a discrete-time signal $x(nT)$ can be uniquely determined from its z transform $X(z)$ by simply obtaining the Laurent series that represents $X(z)$ in its outermost annulus of convergence as illustrated in Fig. 3.4. This can be accomplished by evaluating the coefficients of the Laurent series using the contour integral in Eq. (3.6), that is,

$$x(nT) = \frac{1}{2\pi j}\oint_\Gamma X(z)z^{n-1}\,dz$$

where Γ is a closed contour in the counterclockwise sense enclosing all the singularities of function $X(z)z^{n-1}$. Equation (3.6) is, in effect, the formal definition of the *inverse z transform.*

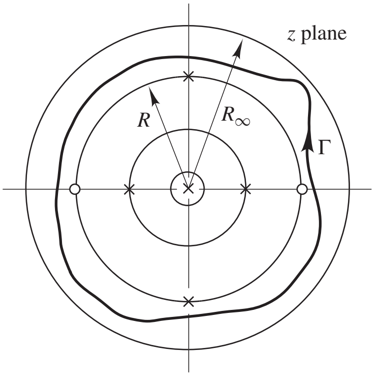


Figure 3.4 Evaluation of inverse z transform.

Like the Fourier transform and its inverse, the z transform and its inverse are often represented in terms of operator format as

$$X(z) = \mathcal{Z}x(nT) \qquad \text{and} \qquad x(nT) = \mathcal{Z}^{-1}X(z)$$

respectively.

At first sight, the contour integration in Eq. (3.6) may appear to be a formidable task. However, for most DSP applications, the z transform turns out to be a rational function like the one in Eq. (3.3) and for such functions the contour integral in Eq. (3.6) can be easily evaluated by using the residue theorem (see Sec. A.7). According to this theorem,

$$x(nT) = \frac{1}{2\pi j}\oint_{\Gamma} X(z)z^{n-1}\,dz = \sum_{i=1}^{P} \underset{z\to p_i}{\Re es}\left[X(z)z^{n-1}\right] \tag{3.8}$$

where $\Re es_{z\to p_i}\left[X(z)z^{n-1}\right]$ is the *residue* of $X(z)z^{n-1}$ at pole p_i. P is the number of poles in $X(z)z^{n-1}$. The residue at a pole of order m_i is given by

$$\underset{z=p_i}{\Re es}\left[X(z)z^{n-1}\right] = \frac{1}{(m_i-1)!}\lim_{z\to p_i}\frac{d^{m_i-1}}{dz^{m_i-1}}\left[(z-p_i)^{m_i}X(z)z^{n-1}\right] \tag{3.9a}$$

which simplifies to

$$\Re\text{es}_{z=p_i}\left[X(z)z^{n-1}\right] = \lim_{z\to p_i}\left[(z-p_i)X(z)z^{n-1}\right] \tag{3.9b}$$

for a simple pole since no differentiation is needed and $0! = 1$. Evidently, the residue at a first-order pole p_i can be readily obtained by simply deleting the factor $(z - p_i)$ from the denominator of $X(z)z^{n-1}$ and then evaluating the remaining part of the function at pole p_i.

The above method of inversion is known as the *general inversion method* for obvious reasons and its application will be examined in Sec. 3.8.

3.6 THEOREMS AND PROPERTIES

The general properties of the z transform can be described in terms of a small number of theorems, as detailed below. To facilitate the exposition we assume that

$$\mathcal{Z}x(nT) = X(z) \qquad \mathcal{Z}x_1(nT) = X_1(z) \qquad \mathcal{Z}x_2(nT) = X_2(z)$$

The symbols a, b, w, and K represent constants which may be complex.

Most of the z transform theorems are proved by applying simple algebraic manipulation to the z transform definition in Eq. (3.1).

Theorem 3.3 Linearity

$$\boldsymbol{\mathcal{Z}[ax_1(nT) + bx_2(nT)] = aX_1(z) + bX_2(z)}$$

and

$$\boldsymbol{\mathcal{Z}^{-1}[aX_1(z) + bX_2(z)] = ax_1(nT) + bx_2(nT)}$$

Proof See Prob. 3.5. ▲

Theorem 3.4 Time Shifting ***For any positive or negative integer m,***

$$\boldsymbol{\mathcal{Z}x(nT+mT)=z^m X(z)}$$

Proof From the definition of the z transform

$$\begin{aligned}\mathcal{Z}x(nT+mT)&=\sum_{n=-\infty}^{\infty}x(nT+mT)z^{-n}\\&=z^m\sum_{n=-\infty}^{\infty}x[(n+m)T]z^{-(n+m)}\end{aligned}$$

If we now make the variable substitution $n+m=n'$ and then replace n' by n, we have

$$\mathcal{Z}x(nT+mT)=z^m\sum_{n=-\infty}^{\infty}x(nT)z^{-n}=z^m X(z)\quad\blacktriangle$$

If m is negative, then $x(nT+mT)=x(nT-|m|T)$ and thus the signal is delayed by $|m|T$ s. As a consequence, the z transform of a discrete-time signal which is delayed by an integer number of sampling periods is obtained by simply multiplying its z transform by the appropriate negative power of z. On the other hand, multiplying the z transform of a signal by a positive power of z causes the signal to be advanced or shifted to the left with respect to the time axis.

Theorem 3.5 Complex Scale Change ***For an arbitrary real or complex constant w,***

$$\boldsymbol{\mathcal{Z}[w^{-n}x(nT)]=X(wz)}$$

Proof

$$\begin{aligned}\mathcal{Z}[w^{-n}x(nT)]&=\sum_{n=-\infty}^{\infty}[w^{-n}x(nT)]z^{-n}\\&=\sum_{n=-\infty}^{\infty}x(nT)(wz)^{-n}\\&=X(wz)\end{aligned}$$

Evidently, multiplying a discrete-time signal by w^{-n} is equivalent to replacing z by wz in its z transform. If the signal is multiplied by v^n, then we can write $v^n=(1/v)^{-n}$ and thus

$$\mathcal{Z}[v^n x(nT)]=\mathcal{Z}\left[\left(\frac{1}{v}\right)^{-n}x(nT)\right]=X(z/v)\quad\blacktriangle$$

Theorem 3.6 Complex Differentiation

$$\boldsymbol{\mathcal{Z}[nT_1x(nT)]=-T_1z\frac{dX(z)}{dz}}$$

Proof

$$\begin{aligned}\mathcal{Z}[nT_1x(nT)] &= \sum_{n=-\infty}^{\infty} nT_1x(nT)z^{-n} = -T_1z\sum_{n=-\infty}^{\infty} x(nT)(-n)z^{-n-1}\\ &= -T_1z\sum_{n=-\infty}^{\infty} x(nT)\frac{d}{dz}(z^{-n})\\ &= -T_1z\frac{d}{dz}\left[\sum_{n=-\infty}^{\infty} x(nT)z^{-n}\right] = -T_1z\frac{dX(z)}{dz}\end{aligned}$$

Changing the order of summation and differentiation is allowed in the last equation for values of z for which $X(z)$ converges. ▲

Complex differentiation provides a simple way of obtaining the z transform of a discrete-time signal that can be expressed as a product $nT_1x(nT)$ by simply differentiating the z transform of $X(z)$.

Theorem 3.7 Real Convolution

$$\begin{aligned}\mathcal{Z}\sum_{k=-\infty}^{\infty} \boldsymbol{x_1(kT)x_2(nT-kT)} &= \mathcal{Z}\sum_{k=-\infty}^{\infty} \boldsymbol{x_1(nT-kT)x_2(kT)}\\ &= \boldsymbol{X_1(z)X_2(z)}\end{aligned}$$

Proof This theorem can be proved by replacing $x(nT)$ in the definition of the z transform by either of the above sums, which are known as *convolution summations*, then changing the order of summation, and after that applying a simple variable substitution, as follows:

$$\begin{aligned}\mathcal{Z}\sum_{k=-\infty}^{\infty} x_1(kT)x_2(nT-kT) &= \sum_{n=-\infty}^{\infty}\left[\sum_{k=-\infty}^{\infty} x_1(kT)x_2(nT-kT)\right]z^{-n}\\ &= \sum_{k=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} x_1(kT)x_2(nT-kT)z^{-n}\\ &= \sum_{k=-\infty}^{\infty} x_1(kT)z^{-k}\sum_{n=-\infty}^{\infty} x_2(nT-kT)z^{-(n-k)}\\ &= \sum_{n=-\infty}^{\infty} x_1(nT)z^{-n}\sum_{n=-\infty}^{\infty} x_2(nT)z^{-n}\\ &= X_1(z)X_2(z)\end{aligned}$$

Changing the order of the two summations in the above proof is valid for all values of z for which $X_1(z)$, and $X_2(z)$ converge. ▲

Convolution summations arise naturally in the representation of digital filters and discrete-time systems as will be shown in Chap. 4. Consequently, the real-convolution theorem can be used to deduce z-domain representations for these systems, as will be shown in Chap. 5.

Theorem 3.8 Initial-Value Theorem ***The initial value of $x(nT)$ for a z transform of the form***

$$X(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{M} a_i z^{M-i}}{\sum_{i=0}^{N} b_i z^{N-i}} \tag{3.10}$$

occurs at

$$KT = (N - M)T \quad \blacktriangle$$

and the value of $x(nT)$ at $nT = KT$ is given by

$$x(KT) = \lim_{z\to\infty} [z^K X(z)] \quad \blacktriangle$$

Corollary ***If the degree of the numerator polynomial, $N(z)$, in the z transform of Eq. (3.10) is equal to or less than the degree of the denominator polynomial $D(z)$, then we have***

$$x(nT) = 0 \qquad \textit{for } n < 0$$

i.e., the signal is right sided. ▲

Proof From the definition of the z transform

$$X(z) = \sum_{n=-\infty}^{\infty} x(nT)z^{-n}$$

If the initial value of $x(nT)$ occurs at $nT = KT$, then

$$X(z) = \sum_{n=K}^{\infty} x(nT)z^{-n} = x(KT)z^{-K} + x(KT + T)z^{-(K+1)} + x(KT + 2T)z^{-(K+2)} + \cdots$$

On dividing both sides by the first term, we have

$$\begin{aligned}\frac{X(z)}{x(KT)z^{-K}} &= 1 + \frac{x(KT + T)z^{-(K+1)}}{x(KT)z^{-K}} + \frac{x(KT + 2T)z^{-(K+2)}}{x(KT)z^{-K}} + \cdots \\ &= 1 + \frac{x(KT + T)}{x(KT)z} + \frac{x(KT + 2T)}{x(KT)z^2} + \cdots\end{aligned}$$

If we take the limit as $z \to \infty$, we have

$$\lim_{z\to\infty} \frac{X(z)}{x(KT)z^{-K}} = 1$$

or

$$x(KT) = \lim_{z\to\infty} [z^K X(z)] \tag{3.11}$$

and from Eqs. (3.11) and (3.10), we can now write

$$x(KT) = \lim_{z\to\infty} X(z)z^K = \lim_{z\to\infty} \left[\frac{\sum_{i=0}^{M} a_i z^{M-i}}{\sum_{i=0}^{N} b_i z^{N-i}} \cdot z^K \right]$$
$$= \frac{a_0}{b_0} z^{M-N} \cdot z^K$$

and since the left-hand side of the equation is independent of z, we get

$$M - N + K = 0$$

Therefore,

$$K = N - M \quad \blacktriangle$$

i.e., *the initial value of $x(nT)$ occurs at $nT = KT$, where K is the difference between the denominator and numerator degrees in $X(z)$.*

With K known, $x(KT)$ can be obtained from Eq. (3.11) as

$$x(KT) = \lim_{z\to\infty} [z^K X(z)] \quad \blacktriangle$$

As has been demonstrated in the absolute-convergence theorem (Theorem 3.1), the z transform will not converge if $x(nT)$ is nonzero at $n = -\infty$. Consequently, a signal must start at some finite point in time in practice. The starting point of a signal as well as its value at the starting point are often of interest and Theorem 3.8 provides a means by which they can be determined.

If the denominator degree in $X(z)$ is equal to or exceeds the numerator degree, then the first nonzero value of $x(nT)$ will occur at $KT = (N-M)T$ and if the condition of the Corollary is satisfied, i.e., $N \geq M$, then $K \geq 0$, that is, $x(nT) = 0$ for $n < 0$. On the basis of this Corollary, one can determine by inspection whether a z transform represents a right-sided or two-sided signal. It is also very useful for checking whether a digital filter or discrete-time system is causal or noncausal (see Chap. 5).

Theorem 3.9 Final Value Theorem ***The value of $x(nT)$ as $n \longrightarrow \infty$ is given by***

$$\boldsymbol{x(\infty) = \lim_{z\to 1} [(z-1)X(z)]}$$

Proof From the time-shifting theorem (Theorem 3.4)

$$\mathcal{Z}[x(nT+T) - x(nT)] = zX(z) - X(z) = (z-1)X(z) \tag{3.12}$$

Alternatively, we can write

$$\mathcal{Z}[x(nT+T) - x(nT)] = \lim_{n\to\infty} \sum_{k=-n}^{n} [x(KT+T) - x(KT)]z^{-n}$$

and if $x(KT)$ is the first nonzero value of $x(nT)$, we have

$$\begin{aligned}
&\mathcal{Z}[x(nT+T)-x(nT)]\\
&= \lim_{n\to\infty}[x(KT)z^{-(K-1)}+x(KT+T)z^{-K}-x(KT)z^{-K}\\
&\quad +\cdots+x(nT)z^{-(n-1)}-x(nT-T)z^{-(n-1)}+x(nT+T)z^{-n}-x(nT)z^{-n}]\\
&= \lim_{n\to\infty}[(z^{-(K-1)}-z^{-K})x(KT)+(z^{-K}-z^{-(K+1)})x(KT+T)\\
&\quad +\cdots+(z^{-(n-1)}-z^{-n})x(nT)+x(nT+T)z^{-n}]\\
&= \lim_{n\to\infty}\left[\frac{(z-1)}{z^K}x(KT)+\frac{(z-1)}{z^{K+1}}x(KT+T)+\cdots+\frac{(z-1)}{z^n}x(nT)+\frac{1}{z^n}x(nT+T)\right]
\end{aligned} \tag{3.13}$$

Now from Eqs. (3.12) and (3.13), we can write

$$\begin{aligned}
&\lim_{z\to 1}(z-1)X(z)\\
&= \lim_{z\to 1}\lim_{n\to\infty}\left[\frac{(z-1)}{z^K}x(KT)+\cdots+\frac{(z-1)}{z^n}x(nT)+\frac{1}{z^n}x(nT+T)\right]\\
&= \lim_{n\to\infty}\lim_{z\to 1}\left[\frac{(z-1)}{z^K}x(KT)+\cdots+\frac{(z-1)}{z^n}x(nT)+\frac{1}{z^n}x(nT+T)\right]\\
&= \lim_{n\to\infty}x(nT+T)
\end{aligned}$$

Therefore,

$$x(\infty)=\lim_{z\to 1}[(z-1)X(z)] \quad \blacktriangle$$

The final-value theorem can be used to determine the steady-state value of a signal in the case where this is finite.

Theorem 3.10 Complex Convolution ***If the z transforms of two discrete-time signals $x_1(nT)$ and $x_2(nT)$ are available, then the z transform of their product, $X_3(z)$, can be obtained as***

$$X_3(z)=\mathcal{Z}[x_1(nT)x_2(nT)]=\frac{1}{2\pi j}\oint_{\Gamma_1}X_1(v)X_2\left(\frac{z}{v}\right)v^{-1}\,dv \tag{3.14a}$$

$$=\frac{1}{2\pi j}\oint_{\Gamma_2}X_1\left(\frac{z}{v}\right)X_2(v)v^{-1}\,dv \tag{3.14b}$$

where Γ_1 (or Γ_2) is a contour in the common region of convergence of $X_1(v)$ and $X_2(z/v)$ (or $X_1(z/v)$ and $X_2(v)$). The two contour integrals in the above equations are equivalent.

Proof From the definition of the z transform and Eq. (3.6), we can write

$$\begin{aligned}
X_3(z) &= \sum_{n=-\infty}^{\infty} [x_1(nT)x_2(nT)]z^{-n} \\
&= \sum_{n=-\infty}^{\infty} x_1(nT)\left[\frac{1}{2\pi j}\oint_{\Gamma_2} X_2(v)v^{n-1}\,dv\right]z^{-n} \\
&= \frac{1}{2\pi j}\oint_{\Gamma_2}\left[\sum_{n=-\infty}^{\infty} x_1(nT)\left(\frac{z}{v}\right)^{-n}\right]X_2(v)v^{-1}\,dv \\
&= \frac{1}{2\pi j}\oint_{\Gamma_2} X_1\left(\frac{z}{v}\right)X_2(v)v^{-1}\,dv
\end{aligned}$$

The order of integration and summation has been interchanged in the last but one line and this is, of course, permissible if contour Γ_2 satisfies the condition stated in the theorem. ▲

The obvious application of Theorem 3.10 is in obtaining the z transform of a product of discrete-time signals whose z transforms are available. The theorem is also vital in the design of nonrecursive digital filters, as will be shown in Chap. 9.

Like the contour integral for the inverse z transform, those in Eq. (3.14) appear quite challenging. However, the most difficult aspect in their evaluation relates to identifying the common region of convergence alluded to in the theorem. Once this is done, what remains is to find the residues of $X_1(z/v)X_2(v)v^{-1}$ or $X_1(z/v)X_2(v)v^{-1}$ at the poles that are encircled by contour Γ_1 or Γ_2, which can be added to give the complex convolution. The complex convolution can be evaluated through the following step-by-step technique:

1. Obtain the zero-pole plots of $X_1(z)$ and $X_2(z)$ and identify the region of convergence for each, as in Fig. 3.5*a* and *b*.
2. Identify which of the two z transforms has the larger radius of convergence. If that of $X_1(z)$ is larger, evaluate the contour integral in Eq. (3.14a); otherwise, evaluate the integral in Eq. (3.14b). In Fig. 3.5, $X_1(z)$ has a larger radius of convergence than $X_2(z)$ and hence the appropriate integral is the one in Eq. (3.14a).
3. Replace z by v in $X_1(z)$ and z by z/v in $X_2(z)$. Switch over from the z plane to the v plane at this point and plot the regions of convergence in the v plane. This can be accomplished as in Fig. 3.5*c* and *d*. The region of convergence in Fig. 3.5*c* is identical with that in Fig. 3.5*a* since the only change involved is a change in the name of the variable. In Fig. 3.5*d*, however, a so-called *conformal mapping* (or transformation) (see Sec. A.9) is involved. We note that if $v \to \infty$, then $z/v \to 0$ and if $v \to 0$, then $z/v \to \infty$; therefore, the region outside (inside) the radius of convergence in Fig. 3.5*b* *maps* onto the region inside (outside) the radius of convergence in Fig. 3.5*d*, as shown.
4. Since the radius of convergence in Fig. 3.5*b* has been assumed to be smaller than that in Fig. 3.5*a*, it follows that the radius of the shaded region in Fig. 3.5*d* is larger than that of the unshaded region in Fig. 3.5*c*. The area that appears shaded in both Fig. 3.5*c* and *d*, illustrated in Fig. 3.5*e*, is the common region of convergence of the product $X_1(z/v)X_2(v)v^{-1}$.

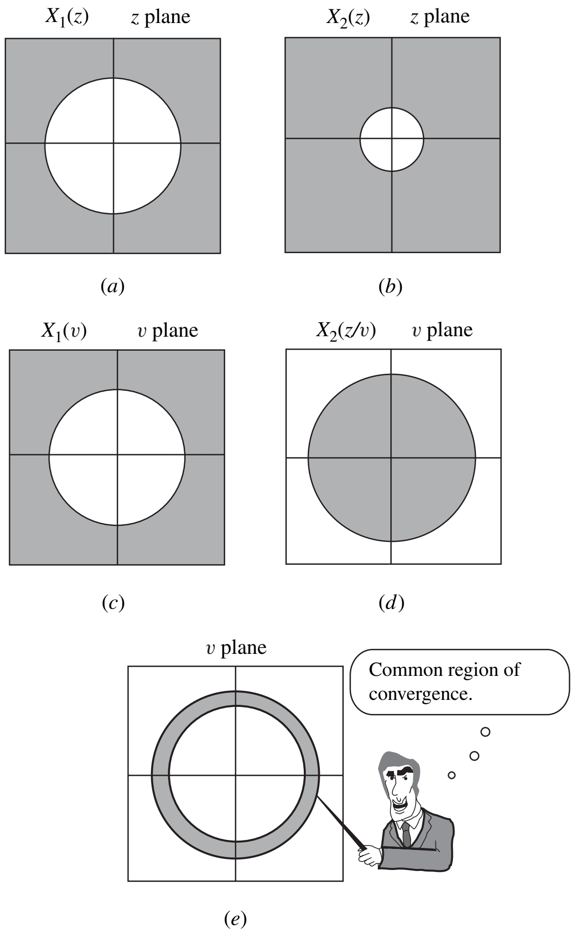


Figure 3.5 Complex convolution.

5. The integral is found by identifying the poles of $X_1(z/v)X_2(v)v^{-1}$ that are located inside the inner circle in Fig. 3.5*e*, finding the residues at these poles, and adding them up.

The technique is illustrated by Example 3.2 in Sec. 3.7.

In certain applications, contour Γ_1 or Γ_2 can be a circle in the common region of convergence and hence we can write $v = \rho e^{j\theta}$ and $z = re^{j\phi}$. In these applications, the above complex convolution integrals become real-convolution integrals. For example, Eq. (3.14b) gives

$$X_3(re^{j\phi}) = \frac{1}{2\pi}\int_0^{2\pi} X_1\left[\frac{r}{\rho}e^{j(\phi-\theta)}\right] X_2(\rho e^{j\theta})\, d\theta \tag{3.15}$$

(see Prob. 3.6, part (*a*)).

Theorem 3.11 Parseval's Discrete-Time Formula ***If $X(z)$ is the z transform of a discrete-time signal $x(nT)$, then***

$$\sum_{n=-\infty}^{\infty} |x(nT)|^2 = \frac{1}{\omega_s} \int_0^{\omega_s} |X(e^{j\omega T})|^2 \, d\omega \tag{3.16}$$

where $\omega_s = 2\pi/T$.

Proof Parseval's discrete-time formula can be derived from the complex-convolution theorem. Although the discrete-time signal $x(nT)$ has been implicitly assumed to be real so far, the z transform can be applied to a complex signal $x(nT)$ just as well as long as $X(z)$ converges. Consider a pair of complex-conjugate signals $x_1(nT)$ and $x_2(nT)$ such that

$$x_1(nT) = x(nT) \tag{3.17a}$$

and

$$x_2(nT) = x^*(nT) \tag{3.17b}$$

We can write

$$X_1(z) = X(z) \tag{3.18a}$$

and

$$\begin{aligned} X_2(z) &= \sum_{n=-\infty}^{\infty} x^*(nT) z^{-n} = \left[\sum_{n=-\infty}^{\infty} x(nT)(z^{-1})^{-n} \right]^* \\ &= X^*(z^{-1}) \end{aligned} \tag{3.18b}$$

From the complex-convolution theorem (Eq. (3.14a)) and the definition of the z transform, we get

$$\mathcal{Z}[x_1(nT)x_2(nT)] = \frac{1}{2\pi j} \oint_{\Gamma_1} X_1(v) X_2\left(\frac{z}{v}\right) v^{-1} \, dv \tag{3.19}$$

Equations (3.17)–(3.19) give

$$\sum_{n=-\infty}^{\infty} [x(nT)x^*(nT)] z^{-n} = \frac{1}{2\pi j} \oint_{\Gamma_1} X(v) X^*\left(\frac{v}{z}\right) v^{-1} \, dv \tag{3.20}$$

and if we let $z = 1$, we obtain

$$\sum_{n=-\infty}^{\infty} |x(nT)|^2 = \frac{1}{2\pi j} \oint_{\Gamma_1} |X(v)|^2 v^{-1} \, dv$$

Now if we let $v = e^{j\omega T}$, contour Γ_1 becomes the unit circle and the contour integral becomes a regular integral whose lower and upper limits of integration become 0 and $2\pi/T$, respectively. Simplifying, the real integral obtained yields Parseval's relation.

For a *normalized signal*, namely, for the case where $T = 1$, $\omega_s = 2\pi/T = 2\pi$ and hence Parseval's summation formula assumes the more familiar form

$$\sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi}\int_0^{2\pi} |X(e^{j\omega})|^2\, d\omega \quad \blacktriangle$$

Note, however, that this formula will give the wrong answer if applied to a signal which is not normalized. ▲

Parseval's formula is often used to solve a problem known as *scaling*, which is associated with the design of recursive digital filters in hardware form (see Chap. 14).

3.7 ELEMENTARY DISCRETE-TIME SIGNALS

The analysis of analog systems is facilitated by using several *elementary signals* such as the unit impulse and the unit step. Corresponding discrete-time signals can be used for the analysis of DSP systems. Some of the basic ones are defined in Table 3.1 and are illustrated in Fig. 3.6. The discrete-time *unit step, unit ramp, exponential*, and *sinusoid* are generated by letting $t = nT$ in the corresponding continuous-time signals. The discrete-time *unit impulse* $\delta(nT)$, however, is generated by letting $t = nT$ in the unit pulse function of Fig. 2.6*a*, which can be represented by the equation

$$p_\tau(t) = \begin{cases} 1 & \text{for } |t| \le \tau/2 < T \\ 0 & \text{otherwise} \end{cases}$$

Note that $\delta(nT)$ *cannot* be obtained from the continuous-time impulse $\delta(t)$ which is usually defined as an infinitely tall and infinitesimally thin pulse (see Sec. 6.2.1). Nevertheless, the discrete- and continuous-time impulse signals play more or less the same role in the analysis and representation of discrete- and continuous-time systems, respectively.

Table 3.1 Elementary discrete-time signals

Function	Definition
Unit impulse	$\delta(nT) = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{for } n \ne 0 \end{cases}$
Unit step	$u(nT) = \begin{cases} 1 & \text{for } n \ge 0 \\ 0 & \text{for } n < 0 \end{cases}$
Unit ramp	$r(nT) = \begin{cases} nT & \text{for } n \ge 0 \\ 0 & \text{for } n < 0 \end{cases}$
Exponential	$u(nT)e^{\alpha nT},\ (\alpha > 0)$
Exponential	$u(nT)e^{\alpha nT},\ (\alpha < 0)$
Sinusoid	$u(nT)\sin \omega nT$

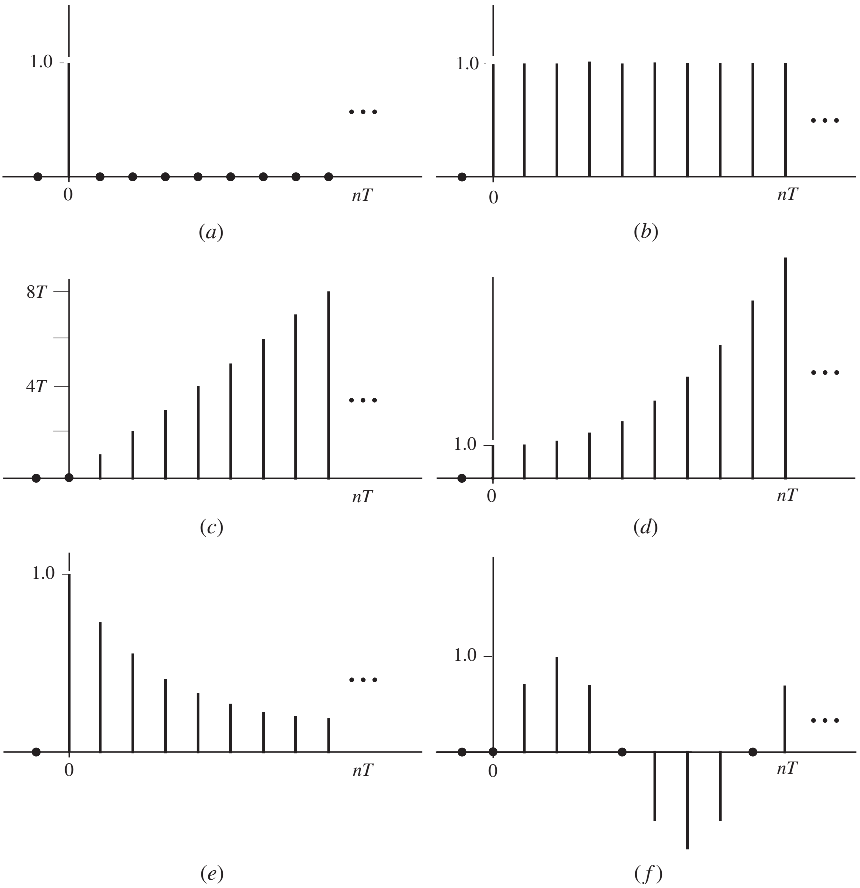


Figure 3.6 Elementary discrete-time functions: (*a*) Unit impulse, (*b*) unit step, (*c*) unit ramp, (*d*) increasing exponential, (*e*) decreasing exponential, (*c*) sinusoid.

The application of the *z* transform to the elementary functions as well as to some other discrete-time signals is illustrated by the following examples.

Example 3.1 Find the *z* transforms of (*a*) $\delta(nT)$, (*b*) $u(nT)$, (*c*) $u(nT-kT)K$, (*d*) $u(nT)Kw^n$, (*e*) $u(nT)e^{-\alpha nT}$, (*f*) $r(nT)$, and (*g*) $u(nT)\sin\omega nT$ (see Table 3.1).

■ **Solution**

(*a*) From the definitions of the *z* transform and $\delta(nT)$, we have

$$\mathcal{Z}\delta(nT) = \delta(0) + \delta(T)z^{-1} + \delta(2T)z^{-2} + \cdots = 1 \quad \blacksquare$$

(*b*) As in part (*a*)

$$\mathcal{Z}u(nT) = u(0) + u(T)z^{-1} + u(2T)z^{-2} + \cdots = 1 + z^{-1} + z^{-2} + \cdots$$

The series at the right-hand side is a binomial series of $(1 - z^{-1})^{-1}$, (see Eq. (A.47)). Hence, we have

$$\mathcal{Z}u(nT) = (1 - z^{-1})^{-1} = \frac{z}{z - 1} \quad \blacksquare$$

(c) From the time-shifting theorem (Theorem 3.4) and part (b), we have

$$\mathcal{Z}[u(nT - kT)K] = Kz^{-k}\mathcal{Z}u(nT) = \frac{Kz^{-(k-1)}}{z - 1} \quad \blacksquare$$

(d) From the complex-scale-change theorem (Theorem 3.5) and part (b), we get

$$\mathcal{Z}[u(nT)Kw^n] = K\mathcal{Z}\left[\left(\frac{1}{w}\right)^{-n} u(nT)\right]$$

$$= K\mathcal{Z}u(nT)|_{z\to z/w} = \frac{Kz}{z - w} \quad \blacksquare$$

(e) By letting $K = 1$ and $w = e^{-\alpha T}$ in part (d), we obtain

$$\mathcal{Z}[u(nT)e^{-\alpha nT}] = \frac{z}{z - e^{-\alpha T}} \quad \blacksquare$$

(f) From the complex-differentiation theorem (Theorem 3.6) and part (b), we have

$$\mathcal{Z}r(nT) = \mathcal{Z}[nTu(nT)] = -Tz\frac{d}{dz}[\mathcal{Z}u(nT)]$$

$$= -Tz\frac{d}{dz}\left[\frac{z}{(z - 1)}\right] = \frac{Tz}{(z - 1)^2} \quad \blacksquare$$

(g) From part (e), we deduce

$$\mathcal{Z}[u(nT)\sin\omega nT] = \mathcal{Z}\left[\frac{u(nT)}{2j}\left(e^{j\omega nT} - e^{-j\omega nT}\right)\right]$$

$$= \frac{1}{2j}\mathcal{Z}[u(nT)e^{j\omega nT}] - \frac{1}{2j}\mathcal{Z}\left[u(nT)e^{-j\omega nT}\right]$$

$$= \frac{1}{2j}\left(\frac{z}{z - e^{j\omega T}} - \frac{z}{z - e^{-j\omega T}}\right)$$

$$= \frac{z\sin\omega T}{z^2 - 2z\cos\omega T + 1} \quad \blacksquare$$

Example 3.2 Find the z transform of

$$x_3(nT) = u(nT)e^{-\alpha nT}\sin\omega nT$$

where $\alpha < 0$.

■ **Solution**

Evidently, we require the z transform of a product of signals and, therefore, this is a clear case for the complex convolution of Theorem 3.10. Let

$$x_1(nT) = u(nT)\sin\omega nT \qquad \text{and} \qquad x_2(nT) = u(nT)e^{-\alpha nT}$$

From Example 3.1, parts (g) and (e), we have

$$X_1(z) = \frac{z\sin\omega T}{(z-e^{j\omega T})(z-e^{-j\omega T})} \qquad \text{and} \qquad X_2(z) = \frac{z}{z-e^{-\alpha T}}$$

We note that $X_1(z)$ has a complex-conjugate pair of poles at $e^{\pm j\omega T}$ whereas $X_2(z)$ has a real pole at $z = e^{-\alpha T}$. Since $|e^{\pm j\omega T}| = 1$, the radius of convergence of $X_1(z)$ is unity and with α assumed to be negative, the radius of convergence of $X_2(z)$ is less than unity. Thus, according to the evaluation technique described earlier, the correct formula to use is that in Eq. (3.14a) and, by a lucky coincidence, the mappings in Fig. 3.5 apply.

Now

$$X_1(v) = \frac{v\sin\omega T}{(v-e^{j\omega T})(v-e^{-j\omega T})}$$

and thus it has poles at $v = e^{\pm j\omega T}$. On the other hand,

$$X_2(z/v) = \left.\frac{z}{z-e^{-\alpha T}}\right|_{z\to z/v} = \frac{z/v}{z/v-e^{-\alpha T}} = \frac{-ze^{\alpha T}}{v-ze^{\alpha T}}$$

and, as a result, it has a pole at $v = ze^{\alpha T}$. Hence the common region of convergence of $X_1(v)$ and $X_2(z/v)$ is the annulus given by

$$1 < |v| < ze^{\alpha T}$$

as depicted in Fig. 3.7. Therefore, the complex convolution assumes the form

$$\begin{aligned} X_3(z) &= \frac{1}{2\pi j}\oint_{\Gamma_1} X_1(v)X_2\left(\frac{z}{v}\right)v^{-1}dv \\ &= \frac{1}{2\pi j}\oint_{\Gamma_1}\frac{-ze^{\alpha T}\sin\omega T}{(v-ze^{\alpha T})(v-e^{j\omega T})(v-e^{-j\omega T})}dv \end{aligned}$$

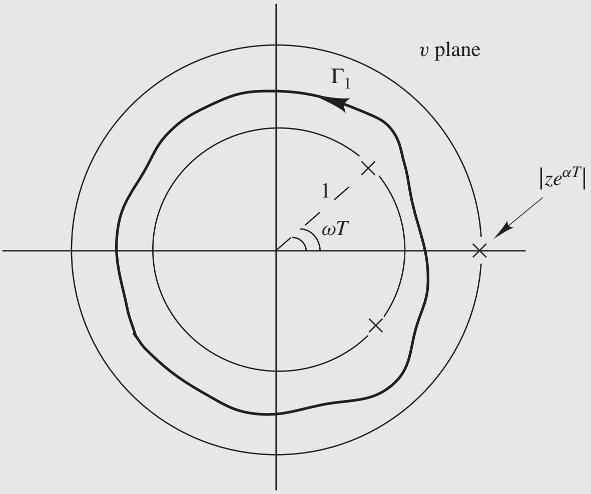


Figure 3.7 Complex convolution (Example 3.2).

where Γ_1 is a contour in the annulus of Fig. 3.7. By evaluating the residues of the integrand at $v = e^{+j\omega T}$ and $e^{-j\omega T}$, we obtain

$$\begin{aligned} X_3(z) &= \left.\frac{-ze^{\alpha T}\sin\omega T}{(v - ze^{\alpha T})(v - e^{-j\omega T})}\right|_{v=e^{j\omega T}} \\ &\quad + \left.\frac{-ze^{\alpha T}\sin\omega T}{(v - ze^{\alpha T})(v - e^{j\omega T})}\right|_{v=e^{-j\omega T}} \\ &= \frac{ze^{-\alpha T}\sin\omega T}{z^2 - 2ze^{-\alpha T}\cos\omega T + e^{-2\alpha T}} \qquad \blacksquare \end{aligned}$$

The radius of convergence of $X_3(z)$ is equal to the magnitude of the poles, which is given by the above equation as $\sqrt{e^{-2\alpha T}} = e^{-\alpha T}$. Alternatively, the annulus of convergence in Fig. 3.7 exists, if $|ze^{\alpha T}| > 1$, that is, $X_3(z)$ converges if $|z| > e^{-\alpha T}$.

The above approach was used primarily to illustrate the complex-convolution theorem which happens to be quite important in the design of nonrecursive filters (see Chap. 9). A simpler approach for the solution of the problem at hand would be to use the complex-scale-change theorem (Theorem 3.5), as will now be demonstrated. From Example 3.1, part (g), we have

$$\mathcal{Z}[u(nT)\sin\omega nT] = \frac{z\sin\omega T}{z^2 - 2z\cos\omega T + 1}$$

and from the complex-scale-change theorem, we can write

$$\mathcal{Z}[w^{-n}x(nT)] = X(wz)$$

Hence

$$\mathcal{Z}[u(nT)w^{-n}\sin\omega nT] = \frac{wz\sin\omega T}{(wz)^2 - 2(wz)\cos\omega T + 1}$$

$$= \frac{zw^{-1}\sin\omega T}{z^2 - 2zw^{-1}\cos\omega T + w^{-2}}$$

Now with $w = e^{\alpha T}$, we deduce

$$\mathcal{Z}[u(nT)e^{-\alpha nT}\sin\omega nT] = \frac{ze^{-\alpha T}\sin\omega T}{z^2 - 2ze^{-\alpha T}\cos\omega T + e^{-2\alpha T}} \quad \blacksquare$$

A list of the common z transforms is given in Table 3.2. A fairly extensive list can be found in the work of Jury [1].

Table 3.2 Standard z transforms

$x(nT)$	$X(z)$
$\delta(nT)$	1
$u(nT)$	$\dfrac{z}{z-1}$
$u(nT-kT)K$	$\dfrac{Kz^{-(k-1)}}{z-1}$
$u(nT)Kw^n$	$\dfrac{Kz}{z-w}$
$u(nT-kT)Kw^{n-1}$	$\dfrac{K(z/w)^{-(k-1)}}{z-w}$
$u(nT)e^{-\alpha nT}$	$\dfrac{z}{z-e^{-\alpha T}}$
$r(nT)$	$\dfrac{Tz}{(z-1)^2}$
$r(nT)e^{-\alpha nT}$	$\dfrac{Te^{-\alpha T}z}{(z-e^{-\alpha T})^2}$
$u(nT)\sin\omega nT$	$\dfrac{z\sin\omega T}{z^2-2z\cos\omega T+1}$
$u(nT)\cos\omega nT$	$\dfrac{z(z-\cos\omega T)}{z^2-2z\cos\omega T+1}$
$u(nT)e^{-\alpha nT}\sin\omega nT$	$\dfrac{ze^{-\alpha T}\sin\omega T}{z^2-2ze^{-\alpha T}\cos\omega T+e^{-2\alpha T}}$
$u(nT)e^{-\alpha nT}\cos\omega nT$	$\dfrac{z(z-e^{-\alpha T}\cos\omega T)}{z^2-2ze^{-\alpha T}\cos\omega T+e^{-2\alpha T}}$

3.8 *Z*-TRANSFORM INVERSION TECHNIQUES

The most fundamental method for the inversion of a z transform is of course the *general inversion method* described in Sec. 3.5 since this is part and parcel of the Laurent theorem (Theorem A.4).

If $X(z)z^{n-1}$ has only first- or second-order poles, the residues are relatively easy to evaluate. However, certain pitfalls can arise that could cause errors. To start with, if $X(z)$ does not have a zero at the origin, the presence of z^{n-1} in $X(z)z^{n-1}$ will introduce a first-order pole at the origin for $n = 0$, and this pole disappears for $n > 0$. This means that one would need to carry out two sets of calculations, one set to obtain $x(nT)$ for $n = 0$ and one set to obtain $x(nT)$ for $n > 0$. This problem is illustrated in the following example.

Example 3.3 Using the general inversion method, find the inverse z transforms of

(*a*)
$$X(z) = \frac{(2z-1)z}{2(z-1)\left(z+\frac{1}{2}\right)}$$

(*b*)
$$X(z) = \frac{1}{2(z-1)\left(z+\frac{1}{2}\right)}$$

■ **Solution**

(***a***) We can write

$$X(z)z^{n-1} = \frac{(2z-1)z\cdot z^{n-1}}{2(z-1)\left(z+\frac{1}{2}\right)} = \frac{(2z-1)z^{n}}{2(z-1)\left(z+\frac{1}{2}\right)}$$

We note that $X(z)z^{n-1}$ has simple poles at $z = 1$ and $-\frac{1}{2}$. Furthermore, the zero in $X(z)$ at the origin cancels the pole at the origin introduced by z^{n-1} for the case $n = 0$. Hence for any $n \geq 0$, Eq. (3.8) gives

$$\begin{aligned} x(nT) &= \underset{z=1}{\mathfrak{Res}}\left[X(z)z^{n-1}\right] + \underset{z=-\frac{1}{2}}{\mathfrak{Res}}\left[X(z)z^{n-1}\right] \\ &= \left.\frac{(2z-1)z^{n}}{2\left(z+\frac{1}{2}\right)}\right|_{z=1} + \left.\frac{(2z-1)z^{n}}{2(z-1)}\right|_{z=-\frac{1}{2}} \\ &= \frac{1}{3}+\frac{2}{3}\left(-\frac{1}{2}\right)^{n} \end{aligned}$$

Since the numerator degree in $X(z)$ does not exceed the denominator degree, $x(nT)$ is a one-sided signal, i.e., $x(nT) = 0$ for $n < 0$, according to the Corollary of Theorem 3.8. Therefore, for any value of n, we have

$$x(nT) = u(nT)\left[\tfrac{1}{3}+\tfrac{2}{3}\left(-\tfrac{1}{2}\right)^{n}\right] \quad ■$$

(*b*) In this z transform, $X(z)$ does not have a zero at the origin and, as a consequence, z^{n-1} introduces a pole in $X(z)z^{n-1}$ at the origin for the case $n = 0$, which must be taken into account in the evaluation of $x(0)$. Thus for $n = 0$, we have

$$X(z)z^{n-1}\Big|_{n=0} = \frac{z^{n-1}}{2(z-1)\left(z+\frac{1}{2}\right)}\Bigg|_{n=0} = \frac{1}{2z(z-1)\left(z+\frac{1}{2}\right)}$$

Hence

$$x(0) = \frac{1}{2(z-1)\left(z+\frac{1}{2}\right)}\Bigg|_{z=0} + \frac{1}{2z\left(z+\frac{1}{2}\right)}\Bigg|_{z=1} + \frac{1}{2z(z-1)}\Bigg|_{z=-\frac{1}{2}} = -1 + \tfrac{1}{3} + \tfrac{2}{3} = 0$$

Actually, this work is unnecessary. The initial-value theorem (Theorem 3.8), gives $x(0) = 0$ without any calculations.

On the other hand, for $n > 0$

$$x(nT) = \frac{z^{n-1}}{2\left(z+\frac{1}{2}\right)}\Bigg|_{z=1} + \frac{z^{n-1}}{2(z-1)}\Bigg|_{z=-\frac{1}{2}} = \tfrac{1}{3} - \tfrac{1}{3}\left(-\tfrac{1}{2}\right)^{n-1}$$

and as in part (a), $x(nT) = 0$ for $n < 0$. Thus, for any value of n, we have

$$x(nT) = u(nT - T)\left[\tfrac{1}{3} - \tfrac{1}{3}\left(-\tfrac{1}{2}\right)^{n-1}\right] \quad \blacksquare$$

The general inversion method tends to become somewhat impractical for z transforms of two-sided signals whereby $x(nT)$ is nonzero for negative values of n. For such z transforms, $X(z)z^{n-1}$ has a higher-order pole at the origin whose order is increased as n is made more negative. And the residue of such a pole is more difficult to evaluate since a higher-order derivative of a rational function in z needs to be calculated. However, the problem can be easily circumvented by using some other available inversion techniques, as will be shown next.

Owing to the uniqueness of the Laurent series in a given annulus of convergence, any technique that can be used to generate a power series for $X(z)$ that converges in the outermost annulus of convergence given by Eq. (3.7) can be used to obtain the inverse z transform. Several such techniques are available, for example, by

- using binomial series,
- using the convolution theorem,
- performing long division,
- using the initial-value theorem (Theorem 3.8), or
- expanding $X(z)$ into partial fractions.

3.8.1 Use of Binomial Series

A factor $(1+b)^r$, where r is a positive or negative integer, can be expressed in terms of the *binomial series* given by Eq. (A.47) and by letting $r=-1$ in Eq. (A.47), we obtain

$$(1+b)^{-1} = 1 + (-b) + b^2 + (-b)^3 + \cdots \tag{3.21a}$$

and if we replace b by $-b$ in Eq. (3.21a), we get

$$(1-b)^{-1} = [1+(-b)]^{-1} = 1 + b + b^2 + b^3 + \cdots \tag{3.21b}$$

By applying the ratio test of Theorem A.3, the series in Eqs. (3.21a) and (3.21b) are found to converge for all values of b such that $|b| < 1$. Thus if $b = w/z$, the series converges for all values of z such that $|z| > |w|$ and if $b = z/w$, then it converges for all values of z such that $|z| < |w|$.

By expressing $X(z)$ in terms of factors such as the above with either $b = w/z$ or $b = z/w$ as appropriate and then replacing the factors by their binomial series representations, all the possible Laurent series for $X(z)$ centered at the origin can be obtained. If we have $b = w/z$ in all the factors then the above series as well as the series obtained for $X(z)$ converge in the outermost annulus

$$|w| \le |z| \le R_\infty \qquad \text{for } R_\infty \to \infty$$

which makes the series a z transform by definition. If we have $b = z/w$ in all the factors, then their series and the series obtained for $X(z)$ converge in the *innermost annulus*, namely,

$$R_0 \le |z| \le |w| \qquad \text{for } R_0 \to 0$$

On the other hand, if we have $b = w/z$ in some factors and $b = z/w$ in others, then the series obtained for $X(z)$ will converge in one of the in-between annuli of convergence.

Example 3.4 Using binomial series, find the inverse z transform of

$$X(z) = \frac{Kz^m}{(z-w)^k}$$

where m and k are integers, and K and w are constants, possibly complex.

■ **Solution**

The inverse z transform can be obtained by finding the Laurent series that converges in the outermost annulus and then identifying the coefficient of z^n, which is $x(nT)$ by definition. Such a series can be obtained by expressing $X(z)$ as

$$\begin{aligned} X(z) &= Kz^{m-k}[1+(-wz^{-1})]^{-k} \\ &= Kz^{m-k}\left[1 + \binom{-k}{1}(-wz^{-1}) + \binom{-k}{2}(-wz^{-1})^2 \right. \\ &\qquad \left. + \cdots + \binom{-k}{n}(-wz^{-1})^n + \cdots \right] \end{aligned}$$

where

$$\binom{-k}{n} = \frac{-k(-k-1)\dots(-k-n+1)}{n!}$$

according to Eq. (A.48). Now if we let $n = n' + m - k$ and then replace n' by n, we have

$$X(z) = \sum_{n=-\infty}^{\infty} \left\{ Ku[(n+m-k)T] \right.$$
$$\left. \times \frac{(-k)(-k-1)\cdots(-n-m+1)(-w)^{n+m-k}}{(n+m-k)!} \right\} z^{-n}$$

Hence the inverse z transform, which is the coefficient of z^{-n}, is obtained as

$$\begin{aligned} x(nT) &= \mathcal{Z}^{-1}\left[\frac{Kz^m}{(z-w)^k}\right] \\ &= Ku[(n+m-k)T] \\ &\quad \times \frac{(-k)(-k-1)\cdots(-n-m+1)(-w)^{n+m-k}}{(n+m-k)!} \end{aligned}$$ ■

Incidentally, this is a fairly general inverse z transform since seven of the twelve inverse z transforms in Table 3.2 can be derived from it by choosing suitable values for the constants k, K, and m.

Example 3.5 (*a*) Using binomial series, find all the Laurent series of

$$X(z) = \frac{(z^2-4)}{z(z^2-1)(z^2+4)} \tag{3.22}$$

with center at the origin of the z plane. (*b*) Identify which Laurent series of $X(z)$ is a z transform.

■ **Solution**

The zero-pole plot of $X(z)$ depicted in Fig. 3.3*a* has three distinct annuli of convergence, namely, A_{I}, A_{II}, and A_{III} as illustrated in Fig. 3.3*d*. The radius of the inner circle of annulus A_{I} can be reduced to zero and that of the outer circle of annulus A_{III} can be increased to infinity. Thus three Laurent series can be obtained for this function, one for each annulus.

Annulus A_{I}: To obtain the Laurent series for the innermost annulus of convergence in Fig. 3.3*d*, that is, A_{I}, $X(z)$ must be expressed in terms of binomial series that converge for values of z in the annulus $R_0 < |z| < 1$ where $R_0 \to 0$. Equation (3.22) can be expressed as

$$\begin{aligned} X(z) &= \frac{(z^2-4)}{z(z^2-1)(z^2+4)} \\ &= \frac{(z^2-4)}{-4z(1-z^2)(1+z^2/4)} \\ &= \frac{(z^2-4)(1-z^2)^{-1}(1+z^2/4)^{-1}}{-4z} \end{aligned} \tag{3.23}$$

From Eqs. (3.21b) and (3.21a), we have

$$(1-z^2)^{-1} = 1 + z^2 + (z^2)^2 + \cdots + (z^2)^n + \cdots \tag{3.24a}$$

and

$$\begin{aligned} (1+z^2/4)^{-1} &= [1-(-z^2/4)]^{-1} \\ &= 1 + (-z^2/4) + (-z^2/4)^2 + \cdots + (-z^2/4)^k + \cdots \end{aligned} \tag{3.24b}$$

respectively. Since both of the above series converge and $(z^2 - 2)/(-4z)$ is finite for $0 < |z| < 1$, the substitution of Eqs. (3.24a) and (3.24b) into Eq. (3.23) will yield a series representation for $X(z)$ that converges in annulus A_{I}. We can write

$$\begin{aligned} X(z) &= \frac{(z^2-4)(1-z^2)^{-1}(1+z^2/4)^{-1}}{-4z} \\ &= \frac{(z^2-4)}{-4z}[1 + z^2 + \cdots + (z^2)^n + \cdots] \cdot [1 + (-z^2/4) + \cdots + (-z^2/4)^k + \cdots] \\ &= \frac{z^2-4}{-4z}\sum_{n=0}^{\infty}\sum_{k=0}^{\infty}(z^2)^n\left(-\frac{z^2}{4}\right)^k \end{aligned}$$

and after some routine algebraic manipulation, the series obtained can be expressed as

$$X(z) = z^{-1} + \sum_{n=1}^{\infty} C_n z^{2n-1} \quad \blacksquare \tag{3.25}$$

where

$$C_n = 1 + 2\sum_{k=1}^{n}\left(-\tfrac{1}{4}\right)^k$$

The sum in the formula for C_n is a geometric series with a common ratio of $-1/4$ and hence it can be readily evaluated as

$$\sum_{k=1}^{n} \left(-\tfrac{1}{4}\right)^k = -\tfrac{1}{5}\left[1-\left(-\tfrac{1}{4}\right)^n\right]$$

(see Eq. (A.46b)). Thus

$$C_n = \tfrac{1}{5}\left[3+2\left(-\tfrac{1}{4}\right)^n\right]$$

and on calculating the coefficients, the series in Eq. (3.25) assumes the form

$$X(z) = \cdots + \tfrac{19}{32}z^5 + \tfrac{5}{8}z^3 + \tfrac{1}{2}z + z^{-1} \tag{3.26}$$

Annulus A_{II} (see Fig. 3.3*d*): A series that converges in annulus A_{II}, that is, $1 < |z| < 2$, can be obtained in the same way. Equation (3.22) can be expressed as

$$\begin{aligned} X(z) &= \frac{(z^2-4)}{z(z^2-1)(z^2+4)} \\ &= \frac{(z^2-4)}{4z^3(1-1/z^2)(1+z^2/4)} \\ &= \frac{(z^2-4)(1-1/z^2)^{-1}(1+z^2/4)^{-1}}{4z^3} \end{aligned} \tag{3.27}$$

where

$$(1-1/z^2)^{-1} = 1+(1/z^2)+(1/z^2)^2+\cdots+(1/z^2)^n+\cdots \tag{3.28}$$

and $(1+z^2/4)^{-1}$ can be expressed in terms of the binomials series in Eq. (3.24b). The series in Eqs. (3.24b) and (3.28) converge in the region $1 < |z| < 2$, as can be easily shown by using the ratio test, and since $(z^2-4)/4z^3$ is finite for $|z| > 0$, and from Eq. (3.27) a series representation for $X(z)$ for annulus A_{II} can be obtained as

$$\begin{aligned} X(z) &= \frac{(z^2-4)(1-1/z^2)^{-1}(1+z^2/4)^{-1}}{4z^3} \\ &= \frac{(z^2-4)}{4z^3}[1+(1/z^2)+\cdots+(1/z^2)^n+\cdots] \\ &\quad \cdot[1+(-z^2/4)+\cdots+(-z^2/4)^k+\cdots] \\ &= \frac{z^2-4}{4z^3}\sum_{n=0}^{\infty}\sum_{k=0}^{\infty}\left(\frac{1}{z^2}\right)^n\left(-\frac{z^2}{4}\right)^k \end{aligned}$$

After some manipulation and some patience, the series obtained can be simplified to

$$X(z) = \sum_{n=1}^{\infty} \left(E_n z^{2n-3} - \tfrac{3}{5} z^{-(2n+1)} \right) \quad \blacksquare \tag{3.29}$$

where

$$E_n = \tfrac{2}{5} \left(-\tfrac{1}{4} \right)^{n-1}$$

If we calculate the numerical values of the coefficients in Eq. (3.29), we get

$$X(z) = \cdots + \tfrac{1}{40} z^3 - \tfrac{1}{10} z + \tfrac{2}{5} z^{-1} - \tfrac{3}{5} z^{-3} - \tfrac{3}{5} z^{-5} - \cdots \tag{3.30}$$

Annulus A_{III} (see Fig. 3.3*d*): A series that converges in annulus A_{III}, that is, $2 < |z| < R_\infty$, can be obtained by expressing $X(z)$ in Eq. (3.22) as

$$\begin{aligned} X(z) &= \frac{(z^2-4)}{z(z^2-1)(z^2+4)} \\ &= \frac{(z^2-4)}{z^5(1-1/z^2)(1+4/z^2)} \\ &= \frac{(z^2-4)(1-1/z^2)^{-1}(1+4/z^2)^{-1}}{z^5} \end{aligned} \tag{3.31}$$

where

$$\begin{aligned} (1+4/z^2)^{-1} &= [1-(-4/z^2)]^{-1} \\ &= 1 + (-4/z^2) + (-4/z^2)^2 + \cdots + (-4/z^2)^k + \cdots \end{aligned} \tag{3.32}$$

and $(1-1/z^2)^{-1}$ can be represented by the binomial series in Eq. (3.28). The series in Eqs. (3.28) and (3.32) converge in the region $2 < |z| < \infty$ and since $(z^2-4)/z^5$ is finite for $|z| < \infty$, a series representation for $X(z)$ for annulus A_{III} can be obtained from Eq. (3.31) as

$$\begin{aligned} X(z) &= \frac{(z^2-4)(1-1/z^2)^{-1}(1+4/z^2)^{-1}}{z^5} \\ &= \frac{(z^2-4)}{z^5}[1 + (1/z^2) + \cdots + (1/z^2)^n + \cdots] \\ &\quad \cdot [1 + (-4/z^2) + \cdots + (-4/z^2)^k + \cdots] \\ &= \frac{z^2-4}{z^5} \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} \left(\frac{1}{z^2} \right)^n \left(-\frac{4}{z^2} \right)^k \end{aligned}$$

After quite a bit of algebra, one can show that

$$X(z) = \sum_{n=0}^{\infty} G_n z^{-2n-3} \quad \blacksquare \tag{3.33}$$

where

$$G_n = F_n - 4F_{n-1}$$

with

$$F_n = \sum_{k=0}^{n} (-4)^k = \tfrac{1}{5}\left[1 - (-4)^{n+1}\right]$$

Hence

$$G_n = \tfrac{1}{5}\left[-3 + 8(-4)^n\right]$$

and on evaluating the coefficients in Eq. (3.33), we get

$$X(z) = z^{-3} - 7z^{-6} + 25z^{-7} - 103z^{-9} + \cdots \tag{3.34}$$

(*b*) A comparison of Eqs. (3.26), (3.30), and (3.34) shows that the three Laurent series obtained for $X(z)$ are all linear combinations of positive and/or negative powers of z and are, in fact, quite similar to each other. Yet only the last one is a z transform that satisfies the absolute-convergence theorem (Theorem 3.1) since this is the only Laurent series that converges in the outermost annulus. ■

3.8.2 Use of Convolution Theorem

From the real-convolution theorem (Theorem 3.7), we have

$$\mathcal{Z}^{-1}[X_1(z)X_2(z)] = \sum_{k=-\infty}^{\infty} x_1(kT)x_2(nT - kT)$$

Thus, if a z transform can be expressed as a product of two z transforms whose inverses are available, then performing the convolution summation will yield the desired inverse.

Example 3.6 Using the real-convolution theorem, find the inverse z transforms of

(*a*)
$$X_3(z) = \frac{z}{(z-1)^2}$$

(*b*)
$$X_4(z) = \frac{z}{(z-1)^3}$$

■ **Solution**

(*a*) Let

$$X_1(z) = \frac{z}{z-1} \qquad \text{and} \qquad X_2(z) = \frac{1}{z-1}$$

From Table 3.2, we can write

$$x_1(nT) = u(nT) \qquad \text{and} \qquad x_2(nT) = u(nT - T)$$

and hence for $n \geq 0$, the real convolution yields

$$\begin{aligned} x_3(nT) &= \sum_{k=-\infty}^{\infty} x_1(kT)x_2(nT - kT) = \sum_{k=-\infty}^{\infty} u(kT)u(nT - T - kT) \\ &= \cdots + \overbrace{u(-T)u(nT)}^{k=-1} + \overbrace{u(0)u(nT - T)}^{k=0} \\ &\quad + \overbrace{u(T)u(nT - 2T)}^{k=1} + \cdots + \overbrace{u(nT - T)u(0)}^{k=n-1} \\ &\quad + \overbrace{u(nT)u(-T)}^{k=n} + \cdots = 0 + 1 + 1 + \cdots + 1 + 0 = n \end{aligned}$$

For $n < 0$, we have

$$\begin{aligned} x_3(nT) &= \sum_{k=-\infty}^{\infty} u(kT)u(nT - T - kT) \\ &= \cdots + \overbrace{u(-T)u(nT)}^{k=-1} + \overbrace{u(0)u(nT - T)}^{k=0} + \overbrace{u(T)u(nT - 2T)}^{k=1} \\ &\quad + \cdots + \overbrace{u(nT - T)u(0)}^{k=n-1} + \overbrace{u(nT)u(-T)}^{k=n} + \cdots \end{aligned}$$

and since all the terms are zero, we get

$$x_3(nT) = 0$$

Alternatively, by virtue of the initial-value theorem, we have $x_3(nT) = 0$ since the numerator degree in $X_3(z)$ is less than the denominator degree.

Summarizing the results obtained, for any value of n, we have

$$x_3(nT) = u(nT)n \quad ■$$

(*b*) For this example, we can write

$$X_1(z) = \frac{z}{(z-1)^2} \qquad \text{and} \qquad X_2(z) = \frac{1}{z-1}$$

and from part (a), we have

$$x_1(nT) = u(nT)n \qquad \text{and} \qquad x_2(nT) = u(nT - T)$$

For $n \geq 0$, the convolution summation gives

$$\begin{aligned}
x_3(nT) &= \sum_{k=-\infty}^{\infty} ku(kT)u(nT - T - kT) \\
&= +\overbrace{0 \cdot [u(nT - T)]}^{k=0} + \overbrace{1 \cdot [u(nT - 2T)]}^{k=1} + \cdots + \overbrace{(n-1)u(0)}^{k=n-1} + \overbrace{nu(-T)}^{k=n} \\
&= +0 + 1 + 2 + \cdots + n - 1 + 0 \\
&= \sum_{k=1}^{n-1} k
\end{aligned}$$

Now by writing the series $1, 2, \ldots, n-1$ first in the forward and then in the reverse order and, subsequently, adding the two series number by number as follows a series of $n-1$ numbers, each of value n, is obtained:

$$\begin{array}{ccccc}
1 & 2 & 3 & \cdots & n-1 \\
n-1 & n-2 & n-3 & \cdots & 1 \\
\hline
n & n & n & n & n
\end{array}$$

Hence, twice the above sum is equal to[1] $(n-1) \times n$ and thus

$$x_3(nT) = \sum_{k=1}^{n-1} k = \tfrac{1}{2}n(n-1)$$

For $n < 0$, $x_3(nT) = 0$, as in part (a) and, therefore,

$$x_3(nT) = \sum_{k=1}^{n-1} k = \tfrac{1}{2}u(nT)n(n-1) \quad \blacksquare$$

3.8.3 Use of Long Division

Given a z transform $X(z) = N(z)/D(z)$, a series that converges in the outermost annulus of $X(z)$ can be readily obtained by arranging the numerator and denominator polynomials in descending powers of z and then performing polynomial division, also known as *long division*. The method turns out to

[1] *Gauss is reputed to have astonished his mathematics teacher by obtaining the sum of the numbers 1 to 100 in just a few seconds by using this technique.*

be rather convenient for finding the values of $x(nT)$ for negative values of n for the case where the z transform represents a two-sided signal. However, the method does not yield a closed-form solution for the inverse z transform but the problem can be easily eliminated by using long division along with one of the methods that yield closed-form solutions for right-sided signals. The method is best illustrated by an example.

Example 3.7 Using long division, find the inverse z transform of

$$X(z) = \frac{-\frac{1}{4} + \frac{1}{2}z - \frac{1}{2}z^2 - \frac{7}{4}z^3 + 2z^4 + z^5}{-\frac{1}{4} + \frac{1}{4}z - z^2 + z^3}$$

■ **Solution**

The numerator and denominator polynomials can be arranged in descending powers of z as

$$X(z) = \frac{z^5 + 2z^4 - \frac{7}{4}z^3 - \frac{1}{2}z^2 + \frac{1}{2}z - \frac{1}{4}}{z^3 - z^2 + \frac{1}{4}z - \frac{1}{4}}$$

Long division can now be carried out as follows:

$$\begin{array}{rl}
 & z^2 + 3z + 1 + z^{-2} + z^{-3} \\ \cline{2-2}
z^3 - z^2 + \frac{1}{4}z - \frac{1}{4} \;\Big| & z^5 + 2z^4 - \frac{7}{4}z^3 - \frac{1}{2}z^2 + \frac{1}{2}z - \frac{1}{4} \\
\Big| & \mp z^5 \pm z^4 \mp \frac{1}{4}z^3 \pm \frac{1}{4}z^2 \\ \cline{2-2}
 & 3z^4 - \frac{8}{4}z^3 - \frac{1}{4}z^2 + \frac{1}{2}z - \frac{1}{4} \\
 & \mp 3z^4 \pm 3z^3 \mp \frac{3}{4}z^2 \pm \frac{3}{4}z \\ \cline{2-2}
 & z^3 - z^2 + \frac{5}{4}z - \frac{1}{4} \\
 & \mp z^3 \pm z^2 \mp \frac{1}{4}z \pm \frac{1}{4} \\ \cline{2-2}
 & z \\
 & \mp z \pm 1 \mp \frac{1}{4}z^{-1} \pm \frac{1}{4}z^{-2} \\ \cline{2-2}
 & 1 - \frac{1}{4}z^{-1} + \frac{1}{4}z^{-2} \\
 & \mp 1 \pm z^{-1} \mp \frac{1}{4}z^{-2} \pm \frac{1}{4}z^{-3} \\ \cline{2-2}
 & \frac{3}{4}z^{-1} + \frac{1}{4}z^{-3} \\
 & \vdots
\end{array}$$

Hence

$$X(z) = z^2 + 3z + 1 + z^{-2} + z^{-3} + \cdots$$

and, therefore,

$$x(-2T) = 1 \qquad x(-T) = 3 \qquad x(0) = 1 \qquad x(T) = 0 \qquad x(2T) = 1, \ \ldots \quad \blacksquare$$

In Example 3.7, one could obtain any number of signal values by continuing the long division but open-ended solutions such as the one obtained are not very convenient in practice. A better strategy would be to continue the long division until $x(0)$ is obtained. At that point, $X(z)$ can be expressed in terms of the quotient plus the remainder as

$$X(z) = Q(z) + R(z)$$

where

$$R(z) = \frac{N'(z)}{D(z)}$$

The inverse z transform can then be obtained as

$$\begin{aligned} x(nT) &= \mathcal{Z}^{-1}[Q(z) + R(z)] \\ &= \mathcal{Z}^{-1}Q(z) + \mathcal{Z}^{-1}R(z) \end{aligned}$$

by virtue of the linearity of the inverse z transform. Since $R(z)$ represents a right-sided signal, its inverse $\mathcal{Z}^{-1}R(z)$ can be readily obtained by using any inversion method that yields a closed-form solution, for example, the general inversion method.

For the z transform in Example 3.7, we can write

$$Q(z) = z^2 + 3z + 1 \qquad \text{and} \qquad R(z) = \frac{z}{z^3 - z^2 + \frac{1}{4}z - \frac{1}{4}}$$

Thus

$$\begin{aligned} x(nT) &= \mathcal{Z}^{-1}Q(z) + \mathcal{Z}^{-1}R(z) \\ &= \mathcal{Z}^{-1}(z^2 + 3z + 1) + \mathcal{Z}^{-1}\left[\frac{z}{z^3 - z^2 + \frac{1}{4}z - \frac{1}{4}}\right] \end{aligned}$$

As may be recalled, the general inversion method is very convenient for finding $x(nT)$ for $n > 0$ but runs into certain complications for $n \le 0$. On the other hand, the long division method is quite straightforward for $n \le 0$ but does not give a closed-form solution for $n > 0$. A prudent strategy would, therefore, be to use the hybrid approach just described.

It is important to note that if long division is performed with the numerator and denominator polynomials of $X(z)$ arranged in ascending instead of descending powers of z, a Laurent series is obtained that converges in the innermost annulus about the origin, i.e., for

$$R_0 \leq r \leq R$$

where $R_0 \to 0$ and R is the radius of the circle passing through the pole nearest to the origin. Such a series is *not* considered to be a z transform in this textbook, as explained in Sec. 3.5.

3.8.4 Use of Initial-Value Theorem

Theorem 3.8 can be used to find the initial value of $x(nT)$, say, $x(K_0T)$. The term $x(K_0T)z^{-K_0}$ can then be subtracted from $X(z)$ to obtain

$$X'(z) = X(z) - x(K_0T)z^{-K_0}$$

Theorem 3.8 can then be used again to find the initial value of $x'(nT)$, say, $x'(K_1T)$. The term $x'(K_1T)z^{-K_1}$ can then be subtracted from $X'(z)$ to obtain

$$X''(z) = X'(z) - x'(K_1T)z^{-K_1}$$

and so on. This method, just like the long-division method, is useful for obtaining the values of $x(nT)$ for negative values of n but, like long division, it does not yield a closed-form solution.

Example 3.8 Find $x(nT)$ for $n \leq 0$ for

$$X(z) = \frac{3z^5 + 2z^4 - 2z^3 - 2z^2 - z + 4}{z^2 - 1}$$

■ **Solution**

Since the numerator degree in $X(z)$ exceeds the denominator degree, $x(nT)$ is nonzero for some negative values of n. From Theorem 3.8, the first nonzero value of $x(nT)$ occurs at

$$KT = (N - M)T = (2 - 5)T = -3T$$

i.e., $K = -3$, and the signal value is given by

$$\begin{aligned} x(-3T) &= \lim_{z\to\infty} \frac{X(z)}{z^3} = \frac{3z^5 + 2z^4 - 2z^3 - 2z^2 - z + 4}{(z^2 - 1)z^3} \\ &= \lim_{z\to\infty} \frac{3z^5}{z^5} = 3 \quad ■ \end{aligned}$$

Now if we subtract $3z^3$ from $X(z)$ and then apply Theorem 3.8 again, the second nonzero value of $x(nT)$ can be deduced. We can write

$$\begin{aligned}
X(z) - 3z^3 &= \frac{3z^5 + 2z^4 - 2z^3 - 2z^2 - z + 4}{z^2 - 1} - 3z^3 \\
&= \frac{3z^5 + 2z^4 - 2z^3 - 2z^2 - z + 4 - 3z^5 + 3z^3}{z^2 - 1} \\
&= \frac{2z^4 + z^3 - 2z^2 - z + 4}{z^2 - 1}
\end{aligned}$$

Hence

$$KT = (N - M)T = (2 - 4)T = -2T$$

and

$$\begin{aligned}
x(-2T) &= \lim_{z\to\infty} \frac{[X(z) - 3z^3]}{z^2} = \lim_{z\to\infty} \frac{2z^4 + z^3 - 2z^2 - z + 4}{(z^2 - 1)z^2} \\
&= \lim_{z\to\infty} \frac{2z^4}{z^4} = 2 \quad \blacksquare
\end{aligned}$$

Proceeding as before, we can obtain

$$\begin{aligned}
x(-T) &= \lim_{z\to\infty} \frac{[X(z) - 3z^3 - 2z^2]}{z} \\
&= \lim_{z\to\infty} \left(\frac{2z^4 + z^3 - 2z^2 - z + 4}{z^2 - 1} - 2z^2 \right) \frac{1}{z} \\
&= \lim_{z\to\infty} \frac{2z^4 + z^3 - 2z^2 - z + 4 - 2z^4 + 2z^2}{(z^2 - 1)z} \\
&= \lim_{z\to\infty} \frac{z^3 - z + 4}{(z^2 - 1)z} = 1 \quad \blacksquare
\end{aligned}$$

and

$$\begin{aligned}
x(0) &= \lim_{z\to\infty} \left[X(z) - 3z^3 - 2z^2 - z \right] \\
&= \lim_{z\to\infty} \left[\frac{z^3 - z + 4}{(z^2 - 1)} - z \right] \\
&= \lim_{z\to\infty} \left[\frac{z^3 - z + 4 - z^3 + z}{(z^2 - 1)} \right] \\
&= \lim_{z\to\infty} \left[\frac{4}{(z^2 - 1)} \right] = 0 \quad \blacksquare
\end{aligned}$$

3.8.5 Use of Partial Fractions

If the degree of the numerator polynomial in $X(z)$ is equal to or less than the degree of the denominator polynomial, the inverse of $X(z)$ can very quickly be obtained through the use of partial fractions. Two techniques are available, as detailed next.

Technique I: The function $X(z)/z$ can be expanded into partial fractions as

$$\frac{X(z)}{z} = \frac{R_0}{z} + \sum_{i=1}^{P} \frac{R_i}{z - p_i}$$

where P is the number of poles in $X(z)$ and

$$R_0 = \lim_{z\to 0} X(z) \qquad R_i = \underset{z=p_i}{\mathfrak{Res}} \left[\frac{X(z)}{z}\right]$$

Hence

$$X(z) = R_0 + \sum_{i=1}^{P} \frac{R_i z}{z - p_i} \tag{3.35}$$

and

$$x(nT) = \mathcal{Z}^{-1}\left(R_0 + \sum_{i=1}^{P} \frac{R_i z}{z - p_i}\right) = \mathcal{Z}^{-1} R_0 + \sum_{i=1}^{P} \mathcal{Z}^{-1} \frac{R_i z}{z - p_i}$$

Now from Table 3.2, we get

$$x(nT) = R_0 \delta(nT) + \sum_{i=1}^{P} u(nT) R_i p_i^n$$

Technique II: An alternative approach is to expand $X(z)$ into partial fractions as

$$X(z) = R_0 + \sum_{i=1}^{P} \frac{R_i}{z - p_i} \tag{3.36}$$

where

$$R_0 = \lim_{z\to\infty} X(z) \qquad R_i = \underset{z=p_i}{\mathfrak{Res}}\, X(z)$$

and P is the number of poles in $X(z)$ as before. Thus

$$\begin{aligned} x(nT) &= \mathcal{Z}^{-1}\left[R_0 + \sum_{i=1}^{P} \frac{R_i}{z - p_i}\right] \\ &= \mathcal{Z}^{-1} R_0 + \sum_{i=1}^{P} \mathcal{Z}^{-1} \frac{R_i}{z - p_i} \end{aligned}$$

and, therefore, Table 3.2 gives

$$X(nT) = R_0\delta(nT) + \sum_{i=1}^{P} u(nT - T)R_i p_i^{n-1}$$

Note that in a partial-fraction expansion, complex-conjugate poles give complex-conjugate residues. Consequently, one need only evaluate one residue for each pair of complex-conjugate poles. Note also that if the numerator degree in $X(z)$ is equal to the denominator degree, then the constant R_0 *must* be present in Eqs. (3.35) and (3.36). If the numerator degree exceeds the denominators degree, one could perform long division until a remainder is obtained in which the numerator degree is equal to or less than the numerator degree as was done in Sec. 3.8.3. The inversion can then be completed by expanding the remainder function into partial fractions.

It should be mentioned here that the partial-fraction method just described is very similar to the general inversion method of Sec. 3.8 in that both methods are actually techniques for obtaining Laurent series, the difference being that the general inversion method yields a Laurent series for $X(z)z^{n-1}$ whereas the partial-fraction method yields a Laurent series of $X(z)$. However, there is a subtle difference between the two: The general inversion method is complete in itself whereas in the partial-fraction method it is assumed that the inverse z transforms of

$$R_0 \qquad \frac{R_i}{z - p_i} \qquad \frac{zR_i}{z - p_i}$$

are known.

Example 3.9 Using the partial-fraction method, find the inverse z transforms of

(*a*)
$$X(z) = \frac{z}{z^2 + z + \frac{1}{2}}$$

(*b*)
$$X(z) = \frac{z}{\left(z - \frac{1}{2}\right)\left(z - \frac{1}{4}\right)}$$

■ **Solution**

(*a*) On expanding $X(z)/z$ into partial fractions as in Eq. (3.35), we get

$$\frac{X(z)}{z} = \frac{1}{z^2 + z + \frac{1}{2}} = \frac{1}{(z - p_1)(z - p_2)} = \frac{R_1}{z - p_1} + \frac{R_2}{z - p_2} \tag{3.37}$$

where

$$p_1 = \frac{e^{j3\pi/4}}{\sqrt{2}} \qquad \text{and} \qquad p_2 = \frac{e^{-j3\pi/4}}{\sqrt{2}}$$

Thus we obtain

$$R_1 = \underset{z=p_1}{\Re es}\left[\frac{X(z)}{z}\right] = -j \qquad \text{and} \qquad R_2 = \underset{z=p_2}{\Re es}\left[\frac{X(z)}{z}\right] = j$$

i.e., complex-conjugate poles give complex-conjugate residues, and so Eq. (3.37) gives

$$X(z) = \frac{-jz}{z-p_1} + \frac{jz}{z-p_2}$$

From Table 3.2, we now obtain

$$\begin{aligned} x(nT) &= u(nT)\left(-jp_1^n + jp_2^n\right) \\ &= \left(\tfrac{1}{2}\right)^{n/2} u(nT)\frac{1}{j}\left(e^{j3\pi n/4} - e^{-j3\pi n/4}\right) \\ &= 2\left(\tfrac{1}{2}\right)^{n/2} u(nT)\sin\frac{3\pi n}{4} \quad \blacksquare \end{aligned}$$

Alternatively, we can expand $X(z)$ into partial fractions using Eq. (3.36) as shown in part (*b*).

(*b*) $X(z)$ can be expressed as

$$X(z) = \frac{z}{\left(z-\frac{1}{2}\right)\left(z-\frac{1}{4}\right)} = R_0 + \frac{R_1}{z-\frac{1}{2}} + \frac{R_2}{z-\frac{1}{4}} \tag{3.38}$$

where

$$\begin{aligned} R_0 &= \lim_{z\to\infty} X(z) = \lim_{z\to\infty}\frac{z}{\left(z-\frac{1}{2}\right)\left(z-\frac{1}{4}\right)} \\ &= \lim_{z\to\infty}\frac{1}{z} = 0 \\ R_1 &= \underset{z=\frac{1}{2}}{\Re es}\, X(z) = \left.\frac{z}{\left(z-\frac{1}{4}\right)}\right|_{z=\frac{1}{2}} = 2 \\ R_2 &= \underset{z=\frac{1}{4}}{\Re es}\, X(z) = \left.\frac{z}{\left(z-\frac{1}{2}\right)}\right|_{z=\frac{1}{4}} = -1 \end{aligned}$$

Hence Eq. (3.38) gives

$$X(z) = \frac{2}{z-\frac{1}{2}} + \frac{-1}{z-\frac{1}{4}}$$

and from Table 3.2

$$x(nT) = 4u(nT - T)\left[\left(\tfrac{1}{2}\right)^n - \left(\tfrac{1}{4}\right)^n\right] \quad \blacksquare$$

And something to avoid. Given a z transform $X(z)$, one could represent the residues by variables, then generate a number of equations, and after that solve them for the residues. For example, given the z transform

$$X(z) = \frac{z^2 - 2}{(z-1)(z-2)} \tag{3.39}$$

one could write

$$\begin{aligned} X(z) &= R_0 + \frac{R_1}{z-1} + \frac{R_2}{z-2} \\ &= \frac{R_1 z - 2R_1 + R_2 z - R_2 + R_0(z-1)(z-2)}{(z-1)(z-2)} \\ &= \frac{R_1 z - 2R_1 + R_2 z - R_2 + R_0(z^2 - 3z + 2)}{(z-1)(z-2)} \\ &= \frac{R_1 z - 2R_1 + R_2 z - R_2 + R_0 z^2 - 3R_0 z + 2R_0}{(z-1)(z-2)} \\ &= \frac{R_0 z^2 + (R_1 + R_2 - 3R_0)z - 2R_1 - R_2 + 2R_0}{(z-1)(z-2)} \end{aligned} \tag{3.40}$$

One could then equate coefficients of equal powers of z in Eqs. (3.39) and (3.40) to obtain

$$\begin{aligned} z^2 &: \quad R_0 = 1 \\ z^1 &: -3R_0 + R_1 + R_2 = 0 \\ z^0 &: \ 2R_0 - 2R_1 - R_2 = -2 \end{aligned} \tag{3.41}$$

Solving this system of equations would give the correct solution as

$$R_0 = 1 \qquad R_1 = 1 \qquad R_2 = 2$$

For a z transform with six poles, a set of six simultaneous equations with six unknowns would need to be solved. Obviously, *this is a very inefficient method*, and it should definitely be avoided. The quick solution for this example is easily obtained by evaluating the residues individually, as follows:

$$R_0 = \left.\frac{z^2 - 2}{(z-1)(z-2)}\right|_{z=\infty} = 1 \qquad R_1 = \left.\frac{z^2 - 2}{(z-2)}\right|_{z=1} = 1 \qquad R_2 = \left.\frac{z^2 - 2}{(z-1)}\right|_{z=2} = 2$$

3.9 SPECTRAL REPRESENTATION OF DISCRETE-TIME SIGNALS

This section examines the application of the z transform as a tool for the spectral representation of discrete-time signals.

3.9.1 Frequency Spectrum

A spectral representation for a discrete-time signal $x(nT)$ can be obtained by evaluating its z transform $X(z)$ at $z = e^{j\omega T}$, that is, by letting

$$X(z)\big|_{z=e^{j\omega T}} = X(e^{j\omega T})$$

Evidently, this substitution will give a function of the frequency variable ω, which turns out to be complex. The magnitude and angle of $X(e^{j\omega T})$, that is,

$$A(\omega) = |X(e^{j\omega T})| \qquad \text{and} \qquad \phi(\omega) = \arg X(e^{j\omega T})$$

define the *amplitude spectrum* and *phase spectrum* of the discrete-time signal $x(nT)$, respectively, and the two together define the *frequency spectrum*.

The exponential function $e^{j\omega T}$ is a complex number of magnitude 1 and angle ωT and as ω is increased from zero to $2\pi/T$, $e^{j\omega T}$ will trace a circle of radius 1 in the z plane, which is referred to as the *unit circle*. Thus evaluating the frequency spectrum of a discrete-time signal at some frequency ω amounts to evaluating $X(z)$ at some point on the unit circle, say, point B, in Fig. 3.8.

Some geometrical features of the z plane are of significant practical interest. For example, zero frequency, that is, $\omega = 0$, corresponds to the point $z = e^{j\omega T}|_{\omega=0} = e^0 = 1$, that is, point A in

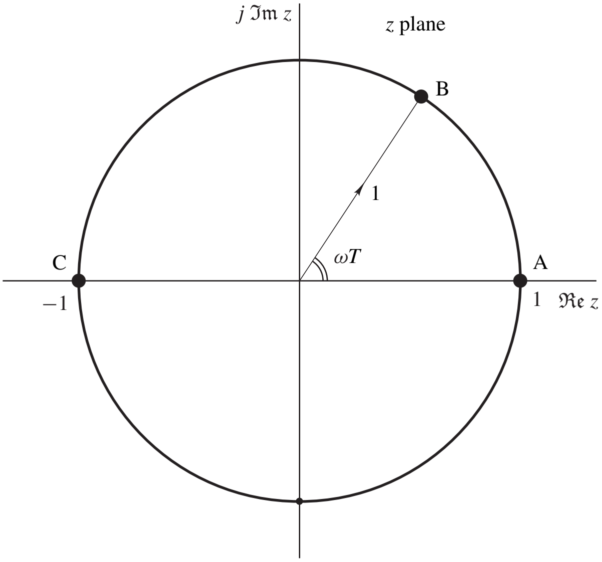


Figure 3.8 Evaluation of frequency spectrum of a discrete-time signal.

Fig. 3.8; half the sampling frequency, i.e., $\omega_s/2 = \pi/T$, which is known as the *Nyquist frequency*, corresponds to the point $z = e^{j\omega T}|_{\omega=\pi/T} = e^{j\pi} = -1$, that is, point C; and the sampling frequency corresponds to the point $z = e^{j\omega T}|_{\omega=2\pi/T} = e^{j2\pi} = 1$, that is, point A, which is also the location for zero frequency.

The frequency spectrum of a discrete-time signal can be determined very quickly through the use of MATLAB, the author's DSP software package D-Filter, or other similar software.

3.9.2 Periodicity of Frequency Spectrum

If frequency ω is changed to $\omega + k\omega_s$ where k is an integer, then

$$\begin{aligned} e^{j(\omega+k\omega_s)T} &= e^{j(\omega T+2k\pi)} = e^{j\omega T} \cdot e^{j2k\pi} \\ &= e^{j\omega T}(\cos 2k\pi + j \sin 2k\pi) \\ &= e^{j\omega T} \end{aligned}$$

Thus

$$X(z)\Big|_{z=e^{j(\omega+k\omega_s)T}} = X(z)\Big|_{z=e^{j\omega T}}$$

or

$$X(e^{j(\omega+k\omega_s)T}) = X(e^{j\omega T})$$

i.e., *the frequency spectrum of a discrete-time signal is a periodic function of frequency with period* ω_s. This actually explains why the sampling frequency corresponds to the same point as zero frequency in the z plane, namely, point A in Fig. 3.8. The frequency range between $-\omega_s/2$ and $\omega_s/2$ is often referred to as the *baseband*.

To consolidate these ideas, let us obtain spectral representations for the discrete-time signals that can be generated from the continuous-time signals of Examples 2.5 and 2.10 through the sampling process.

Example 3.10 The pulse signal of Example 2.5 (see Fig. 2.6a) is sampled using a sampling frequency of 100 rad/s to obtain a corresponding discrete-time signal $x(nT)$. Find the frequency spectrum of $x(nT)$ assuming that $\tau = 0.5$ s.

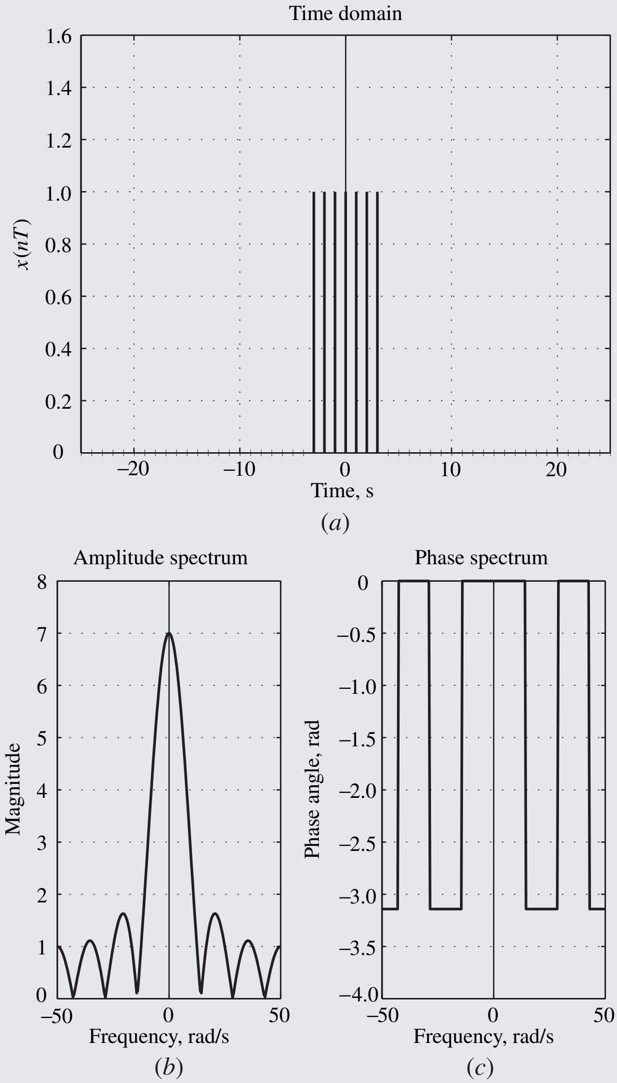
■ **Solution**

The sampling period is

$$T = \frac{2\pi}{\omega_s} = \frac{2\pi}{100} = 0.062832 \text{ s}$$

Hence from Fig. 2.6a, we note that there are

$$\text{int}\left(\frac{\tau}{T}\right) = \text{int}\left(\frac{0.5}{0.062832}\right) = 7$$



(*a*)

(*b*)

(*c*)

Figure 3.9 Frequency spectrum of discrete-time pulse signal (Example 3.10): (*a*) Discrete-time pulse, (*b*) amplitude spectrum, (*c*) phase spectrum.

samples in the range $-\tau/2$ to $\tau/2$, as illustrated in Fig. 3.9*a*. Thus the required discrete-time signal can be expressed as

$$x(nT) = \begin{cases} 1 & \text{for } -3 \leq n \leq 3 \\ 0 & \text{otherwise} \end{cases}$$

From the definition of the z transform, we get

$$X(z) = \sum_{n=-\infty}^{\infty} x(nT)z^{-n} = \sum_{n=-3}^{3} z^{-n}$$

The frequency spectrum of the signal is obtained as

$$\begin{aligned} X(e^{j\omega T}) &= 1 + (e^{j\omega T} + e^{-j\omega T}) + (e^{j2\omega T} + e^{-j2\omega T}) + (e^{j3\omega T} + e^{-j3\omega T}) \\ &= 1 + 2\cos\omega T + 2\cos 2\omega T + 2\cos 3\omega T \quad \blacksquare \end{aligned}$$

Hence the amplitude and phase spectrums of $x(nT)$ are given by

$$A(\omega) = |1 + 2\cos\omega T + 2\cos 2\omega T + 2\cos 3\omega T| \quad \blacksquare$$

and

$$\phi(\omega) = \begin{cases} 0 & \text{if } X(e^{j\omega T}) \geq 0 \\ -\pi & \text{otherwise} \end{cases} \quad \blacksquare$$

respectively. Their plots are depicted in Fig. 3.9b and c.

Example 3.11 The z transform of the discrete-time signal

$$x(nT) = u(nT)e^{-\alpha nT} \sin\omega_0 nT$$

(see Example 2.10) where α and ω_0 are positive constants and

$$u(nT) = \begin{cases} 1 & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases}$$

is the discrete-time *unit-step function* can be obtained as

$$X(z) = \frac{ze^{-\alpha T}\sin\omega_0 T}{z^2 - 2ze^{-\alpha T}\cos\omega_0 T + e^{-2\alpha T}}$$

(see Table 3.2). Deduce the frequency spectrum.

■ **Solution**

The given z transform can be expressed as

$$X(z) = \frac{a_1 z}{z^2 + b_1 z + b_0}$$

where

$$a_1 = e^{-\alpha T}\sin\omega_0 T$$
$$b_0 = e^{-2\alpha T}$$
$$b_1 = -2e^{-\alpha T}\cos\omega_0 T$$

The frequency spectrum of $x(nT)$ can be obtained by evaluating $X(z)$ at $z = e^{j\omega T}$, that is,

$$\begin{aligned}
X(e^{j\omega T}) &= \frac{a_1 e^{j\omega T}}{e^{j2\omega T} + b_1 e^{j\omega T} + b_0}\\
&= \frac{a_1 e^{j\omega T}}{\cos 2\omega T + j\sin 2\omega T + b_1\cos\omega T + jb_1\sin\omega T + b_0}\\
&= \frac{a_1 e^{j\omega T}}{b_0 + b_1\cos\omega T + \cos 2\omega T + j(b_1\sin\omega T + \sin 2\omega T)}\\
&= A(\omega)e^{j\phi(\omega)} \quad \blacksquare
\end{aligned}$$

where

$$\begin{aligned}
A(\omega) &= \frac{|a_1|\cdot|e^{j\omega T}|}{|(b_0 + b_1\cos\omega T + \cos 2\omega T) + j(b_1\sin\omega T + \sin 2\omega T)|}\\
&= \frac{|a_1|}{\sqrt{(b_0 + b_1\cos\omega T + \cos 2\omega T)^2 + (b_1\sin\omega T + \sin 2\omega T)^2}}\\
&= \frac{|a_1|}{\sqrt{1 + b_0^2 + b_1^2 + 2b_1(1+b_0)\cos\omega T + 2b_0\cos 2\omega T}} \quad \blacksquare
\end{aligned}$$

(See Eq. (A.32b).) Since $T > 0$, we have

$$\begin{aligned}
\phi(\omega) &= \arg(a_1) + \arg e^{j\omega T} - \arg[b_0 + b_1\cos\omega T + \cos 2\omega T + j(b_1\sin\omega T + \sin 2\omega T)]\\
&= \arg a_1 + \omega T - \tan^{-1}\frac{b_1\sin\omega T + \sin 2\omega T}{b_0 + b_1\cos\omega T + \cos 2\omega T} \quad \blacksquare
\end{aligned}$$

where

$$\arg a_1 = \begin{cases} 0 & \text{if} \geq 0\\ -\pi & \text{otherwise}\end{cases}$$

(See Eq. (A.32c).) The amplitude and phase spectrums of the discrete-time signal are illustrated in Fig. 3.10 for the case where $\alpha = 0.4$ and $\omega_0 = 2.0$ rad/s assuming a sampling frequency $\omega_s = 2\pi/T = 10$ rad/s.

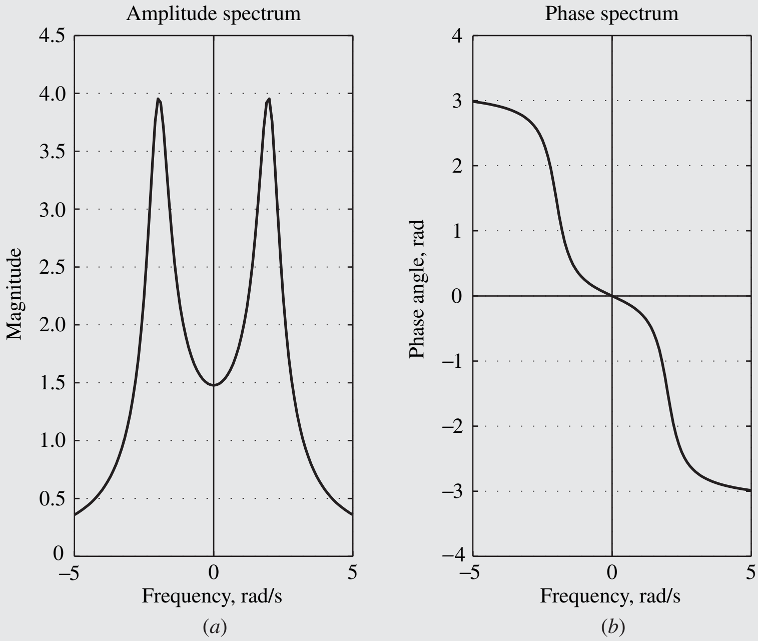


Figure 3.10 Frequency spectrum of discrete-time decaying sinusoidal signal (Example 3.11, $\alpha = 0.4$, $\omega_0 = 2.0$ rad/s, and $\omega_s = 10$ rad/s): (*a*) Amplitude spectrum, (*b*) phase spectrum.

The amplitude and phase spectrums of the discrete-time decaying sinusoidal signal of Example 3.11 over the frequency range $-3\omega_s/2$ to $3\omega_s/2$ with $\omega_s = 20$ rad/s are depicted in Fig. 3.11. As expected, the frequency spectrum is periodic with period $\omega_s = 20$ rad/s.

3.9.3 Interrelations

In the two examples presented in the preceding section, we have examined discrete-time signals that were obtained by sampling the continuous-time signals in Examples 2.5 and 2.10. If we compare the frequency spectrums of the discrete-time signals with those of the corresponding continuous-time signals (i.e., Fig. 2.7*a* and *b* with Fig. 3.9*b* and *c* and Fig. 2.11*b* and *c* with Fig. 3.10*a* and *b*), we note a strong resemblance between the two. Since the former are derived from the latter, it is reasonable to expect that some mathematical relation must exist between the two sets of spectrums. Such a relation does, indeed, exist but it depends critically on the frequency content of the continuous-time signal relative to the sampling frequency. If the highest frequency present in the signal is less than the Nyquist frequency (i.e., $\omega_s/2$), then the spectrum of the discrete-time signal over the baseband is exactly equal to that of the continuous-time signal times $1/T$, where T is the sampling period. Under these circumstances, the continuous-time signal can be recovered completely from the corresponding discrete-time signal by simply removing all frequency components outside the baseband and then multiplying by T.

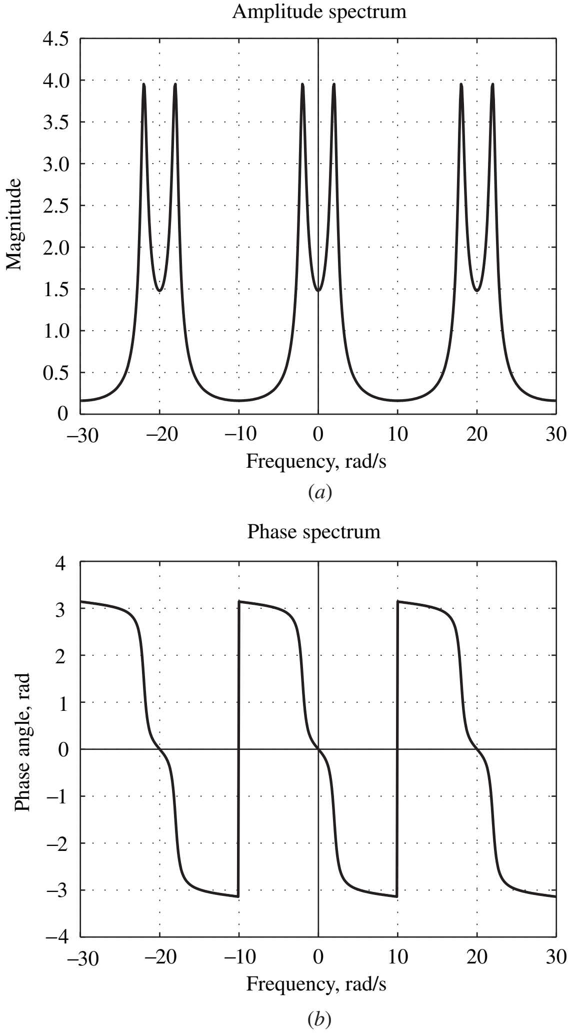


Figure 3.11 Frequency spectrum of discrete-time decaying sinusoidal signal (Example 3.11, $\alpha = 0.4$, $\omega_0 = 2.0$ rad/s, $\omega_s = 20$ rad/s): (*a*) Amplitude spectrum, (*b*) phase spectrum.

The above discussion can be encapsulated in a neat theorem, known as *the sampling theorem*, which states that *a continuous-time signal whose frequency spectrum comprises frequencies that are less than half the sampling frequency (or, alternatively, a continuous-time signal which is sampled at a rate that is higher than two times the highest frequency present in the signal) can be completely recovered from its sampled version*. The sampling theorem is obviously of crucial importance because if it is satisfied, then we can sample our signals to obtain corresponding discrete-time signals without incurring loss of information. The discrete-time signals can then be transmitted or archived using digital hardware. Since no loss of information is involved, the original continuous-signal can be recovered at any time. One can go one step further and process the discrete-time signal using a

DSP system. Converting such a processed discrete-time into a continuous-time signal will yield a processed continuous-time signal, and in this way the processing of continuous-time signals can be achieved by means of DSP systems.

If the sampling theorem is only approximately satisfied, for example, if there are some low-level components whose frequencies exceed the Nyquist frequency, then the relation between the spectrum of the continuous-time signal and that of the discrete-time signal becomes approximate and as more and more components have frequencies that exceed the Nyquist frequency, the relation becomes more and more tenuous and eventually it breaks down.

It follows from the above discussion that the sampling theorem and the spectral relationships that exist between continuous- and discrete-time signals are of considerable importance. They will, therefore, be examined in much more detail later on in Chap. 6.

REFERENCE

[1] E. I. Jury, *Theory and Application of the z-Transform Method*, New York: Wiley, 1964.

PROBLEMS

3.1. Construct the zero-pole plots of the following functions, showing the order of the zero or pole where applicable:

(*a*) $X(z) = \dfrac{z^2+1}{(z^2-3)^2}$

(*b*) $X(z) = \dfrac{(z^2+4)^2}{(z+1)^4}$

(*c*) $X(z) = \dfrac{z^2+2z+1}{z^2+\frac{3}{4}z+\frac{1}{8}}$

3.2. Construct the zero-pole plots of the following functions, showing the order of the zero or pole where applicable:

(*a*) $X(z) = z^2 + z^{-1}$

(*b*) $X(z) = \dfrac{z^7+1}{(z^2+1)^3}$

(*c*) $X(z) = \dfrac{1}{z^3+6z^2+11z+6}$

3.3. Construct the zero-pole plots of the following functions:

(*a*) $X(z) = \dfrac{\pi z^{-5}}{\pi z+1}$

(*b*) $X(z) = \dfrac{z(z+1)}{z^2-1.3z+0.42}$

(*c*) $X(z) = \dfrac{216z^2+162z+29}{(2z+1)(12z^2+7z+1)}$

3.4. Construct the zero-pole plots of the following functions:

(*a*) $X(z) = 4z^{-1}+3z^{-2}+2z^{-3}+z^{-4}+z^{-5}$

(*b*) $X(z) = \dfrac{z^6+2z^2}{(z^4+3z^2+1)}$

(*c*) $X(z) = \dfrac{3z+2-2z^{-1}-2z^{-2}-z^{-3}-4z^{-4}}{1-\frac{5}{4}z^{-}2+\frac{1}{4}z^{-4}}$

3.5. (*a*) Prove that the z transform is a linear operation.

(*b*) Repeat part (*a*) for the inverse z transform.

3.6. (*a*) Obtain the real-convolution integral of Eq. (3.15) from the complex convolution given in Eq. (3.14b).

(*b*) Derive Parseval's discrete-time formula in Eq. (3.16) starting with the complex-convolution formula given in Eq. (3.14b).

3.7. For each of the following functions, obtain all Laurent series with center $z = 0$. Show the region of convergence and identify which series is a z transform in each case:

(*a*) $X(z) = \dfrac{1}{1 - z^2}$

(*b*) $X(z) = \dfrac{1}{z(z - 1)^2}$

3.8. For each of the following functions, obtain all Laurent series with center $z = 0$. Show the region of convergence and identify which series is a z transform in each case:

(*a*) $X(z) = \dfrac{4z - 1}{z^4 - 1}$

(*b*) $X(z) = \dfrac{z^2 + 2z + 1}{(z - 1)\left(z + \frac{1}{2}\right)}$

3.9. For each of the following functions, obtain all Laurent series with center $z = 0$. Show the region of convergence and identify which series is a z transform in each case:

(*a*) $X(z) = \dfrac{4z^2 + 2z - 4}{(z^2 + 1)(z^2 - 4)}$

(*b*) $X(z) = \dfrac{7z^2 + 9z - 18}{z^3 - 9z}$

3.10. For each of the following functions, obtain all Laurent series with center $z = 0$. Show the region of convergence and identify which series is a z transform in each case:

(*a*) $X(z) = \dfrac{4z^2 + 1}{\left(z^2 - \frac{1}{4}\right)(z^2 - 1)}$

(*b*) $X(z) = \dfrac{z^5}{(z^2 - 1)(z^2 - 2)(z^2 + 3)}$

3.11. Find the z transforms of the following functions:

(*a*) $u(nT)(2 + 3e^{-2nT})$

(*b*) $u(nT) - \delta(nT)$

(*c*) $\frac{1}{2}u(nT)e^n r(nT - T)$

3.12. Find the z transforms of the following functions:

(*a*) $u(nT)[1 + (-1)^n]e^{-nT}$

(*b*) $u(nT)\sinh \alpha nT$

(*c*) $u(nT)\sin(\omega nT + \psi)$

(*d*) $u(nT)\cosh \alpha nT$

3.13. Find the z transforms of the following functions:

(*a*) $u(nT)nTw^n$

(*b*) $u(nT)(nT)^2$

(*c*) $u(nT)(nT)^3$

(*d*) $u(nT)nTe^{-4nT}$

3.14. Find the z transforms of the following discrete-time signals:

(*a*) $x(nT) = \begin{cases} 1 & \text{for } 0 \le n \le k \\ 0 & \text{otherwise} \end{cases}$

$$(b)\ \ x(nT) = \begin{cases} nT & \text{for } 0 \le n \le 5 \\ 0 & \text{otherwise} \end{cases}$$

3.15. Find the z transforms of the following discrete-time signals:

$$(a)\ \ x(nT) = \begin{cases} 0 & \text{for } n < 0 \\ 1 & \text{for } 0 \le n \le 5 \\ 2 & \text{for } 5 < n \le 10 \\ 3 & \text{for } n > 10 \end{cases}$$

$$(b)\ \ x(nT) = \begin{cases} 0 & \text{for } n < 0 \\ nT & \text{for } 0 \le n < 5 \\ (n-5)T & \text{for } 5 \le n < 10 \\ (n-10)T & \text{for } n < 10 \end{cases}$$

3.16. Find the z transforms of the following discrete-time signals:

$$(a)\ \ x(nT) = \begin{cases} 0 & \text{for } n < 0 \\ 1 & \text{for } 0 \le n \le 9 \\ 2 & \text{for } 10 < n \le 19 \\ -1 & \text{for } n \ge 20 \end{cases}$$

$$(b)\ \ x(nT) = \begin{cases} 0 & \text{for } n \le -3 \\ n & \text{for } -2 \le n < 1 \\ 1 & \text{for } 1 \le n \le 5 \\ 2 & \text{for } n \ge 6 \end{cases}$$

3.17. Find the z transforms of the following discrete-time signals:

$$(a)\ \ x(nT) = \begin{cases} 0 & \text{for } n < 0 \\ 2 + nT & \text{for } 2 + nT \end{cases}$$

$$(b)\ \ x(nT) = \begin{cases} 0 & \text{for } n < -2 \\ 2 & \text{for } -2 \le n \le -1 \\ 2 - nT & \text{for } n \ge 0 \end{cases}$$

3.18. Find the z transforms of the following discrete-time signals:

$(a)\ \ x(nT) = u(nT)nT(1 + e^{-\alpha nT})$

$$(b)\ \ x(nT) = \begin{cases} 0 & \text{for } n \le 0 \\ \frac{e^{-\alpha nT}}{nT} & \text{for } n > 0 \end{cases} \quad \left(\text{Note that } \ln\frac{1}{1-y} = \sum_{k=1}^{\infty} \frac{y^k}{k}\right)$$

3.19. By using the real-convolution theorem (Theorem 3.7), obtain the z transforms of the following:

$(a)\ \ x(nT) = \sum_{k=0}^{n} r(nT - kT)u(kT)$

$(b)\ \ x(nT) = \sum_{k=0}^{n} u(nT - kT)u(kT)e^{-\alpha kT}$

3.20. Prove that

$$\mathcal{Z} \sum_{k=0}^{n} x(kT) = \frac{z}{z-1} \mathcal{Z} x(nT)$$

3.21. Find $f(0)$ and $f(\infty)$ for the following z transforms:

$$(a)\ \ X(z) = \frac{2z - 1}{z - 1}$$

$$(b)\ \ X(z) = \frac{(e^{-\alpha T} - 1)z}{z^2 - (1 + e^{-\alpha T})z + e^{-\alpha T}}$$

$$(c)\ \ X(z) = \frac{Tze^{-4T}}{(z - e^{-4T})^2}$$

3.22. Find the z transforms of the following:

(a) $y(nT) = \sum_{i=0}^{N} a_i x(nT - iT)$

(b) $y(nT) = \sum_{i=0}^{N} a_i x(nT - iT) - \sum_{i=1}^{N} b_i y(nT - iT)$

3.23. Form the z transform of

$$x(nT) = [u(nT) - u(nT - NT)]W^{kn}$$

3.24. Find the z transforms of the following:

(a) $x(nT) = u(nT)\cos^2 \omega nT$

(a) $x(nT) = u(nT)\sin^4 \omega nT$

3.25. Find the z transforms of the following by using the complex-convolution theorem (Theorem 3.10):

(a) $x(nT) = u(nT)e^{-\alpha nT}\sin(\omega nT + \psi)$

(b) $x(nT) = r(nT)\sin(\omega nT + \psi)$

(c) $x(nT) = r(nT)e^{-\alpha nT}\cos(\omega nT + \psi)$

3.26. Find the inverse z transforms of the following:

(a) $X(z) = \dfrac{2}{2z - 1}$ (b) $X(z) = \dfrac{5}{z - e^{-T}}$

(c) $X(z) = \dfrac{3z}{3z + 2}$ (d) $X(z) = \dfrac{2z}{z^2 - 2z + 1}$

3.27. Find the inverse z transforms of the following:

(a) $X(z) = \dfrac{z + 2}{z^2 - \frac{1}{4}}$ (b) $X(z) = \dfrac{z^2}{\left(z - \frac{1}{2}\right)^5}$

(c) $X(z) = \dfrac{z(z + 1)}{(z - 1)(z^2 + 1)}$ (d) $X(z) = \dfrac{z^3 + 2z}{(z + 1)(z^2 + 1)}$

3.28. Find the inverse z transforms of the following by using the long-division method for $n \le 0$ and the general inversion method for $n > 0$ method:

(a) $X(z) = \dfrac{216z^3 + 96z^2 + 24z + 2}{12z^2 + 9z + 18}$

(b) $X(z) = \dfrac{3z^4 - z^3 - z^2}{z - 1}$

(c) $X(z) = \dfrac{3z^5 + 2z^4 - 2z^3 - 2z^2 - z + 4}{z^2 - 1}$

3.29. Find the inverse z transforms in Prob. 3.28 by using the initial-value theorem (Theorem 3.8) for $n \le 0$ and the partial-fraction method for $n > 0$ method.

3.30. Find the inverse z transforms of the following by using the general inversion method:

(a) $X(z) = \dfrac{z^2}{z^2 + 1}$ (b) $X(z) = \dfrac{2z^2}{2z^2 - 2z + 1}$

3.31. Find the inverse z transforms of the following by using the general inversion method:

(a) $X(z) = \dfrac{1}{\left(z - \frac{4}{5}\right)^4}$

(b) $X(z) = \dfrac{6z}{(2z^2 + 2z + 1)(3z - 1)}$

3.32. Find the inverse z transforms of the following by using the partial-fraction method:

(a) $X(z) = \dfrac{(z - 1)^2}{z^2 - 0.1z - 0.56}$ (b) $X(z) = \dfrac{4z^3}{(2z + 1)(2z^2 - 2z + 1)}$

3.33. Find the inverse z transforms of the following by using the real-convolution theorem (Theorem 3.7):

$$(a)\ X(z) = \frac{z^2}{z^2 - 2z + 1} \qquad (b)\ X(z) = \frac{z^2}{(z - e^{-T})(z - 1)}$$

3.34. Find the inverse z transform of

$$X(z) = \frac{z(z+1)}{(z-1)^3}$$

3.35. Find the inverse z transform of the following by means of long division:

$$X(z) = \frac{z(z^2 + 4z + 1)}{(z-1)^4}$$

3.36. (*a*) Derive expressions for the amplitude and phase spectrums of the signal represented by the z transform

$$X(z) = \frac{z^2 - z + \frac{1}{2}}{z^2 - \frac{1}{2}z + \frac{1}{4}}$$

(*b*) Calculate that amplitude and phase spectrums at $\omega = 0$, $\omega_S/4$, and $\omega_S/2$.

(*c*) Using MATLAB or a similar software package compute and plot the amplitude and phase spectrums.

3.37. Repeat parts (*a*) and (*b*) of Prob. 3.36, for the following z transform

$$X(z) = 0.086\frac{(z^2 - 1.58z + 1)}{z^2 - 1.77z + 0.81}$$

3.38. Repeat parts (*a*) and (*b*) of Prob. 3.36, for the following z transform

$$X(z) = \frac{(12z^3 + 6.4z^2 + 0.68z)}{(z + 0.1)(z^2 + 0.8z + 0.15)}$$

3.39. The z transform of a signal is given by

$$X(z) = \frac{z^2 + a_1 z + a_0}{z^2 + b_1 z + b_0}$$

(*a*) Show that the amplitude spectrum of the signal is given by

$$X(j\omega) = \frac{\sqrt{1 + a_0^2 + a_1^2 + 2a_1(1 + 2a_0)\cos\omega T + 2a_0\cos 2\omega T}}{\sqrt{1 + b_0^2 + b_1^2 + 2b_1(1 + 2b_0)\cos\omega T + 2b_0\cos 2\omega T}}$$

(*b*) Obtain an expression for the phase spectrum.

CHAPTER 4

DISCRETE-TIME SYSTEMS

4.1 INTRODUCTION

Digital-signal processing is carried out by using discrete-time systems. Various types of discrete-time systems have emerged since the invention of the digital computer such as digital control, robotic, and image-processing systems. Discrete-time systems that are designed to perform filtering are almost always referred to as digital filters, and a variety of digital filters have evolved over the years as detailed in Chap. 1.

A discrete-time system is characterized by a rule of correspondence that describes the relationship of the output signal produced with respect to the signal applied at the input of the system. Depending on the rule of correspondence, a discrete-time system can be linear or nonlinear, time invariant or time dependent, and causal or noncausal.

Discrete-time systems are built from a small set of basic constituent discrete-time elements that can perform certain elementary operations like addition and multiplication. By interconnecting a number of these basic elements, discrete-time networks can be formed that can be used to implement some fairly sophisticated discrete-time systems.

Two types of processes can be applied to discrete-time systems, analysis and design. *Analysis* can be used to deduce a mathematical representation for a discrete-time system or to find the output signal produced by a given input signal. *Design*, on the other hand, is the process of obtaining through the use of mathematical principles a discrete-time system that would produce a desired output signal when a specified signal is applied at the input.

This chapter deals with the analysis of discrete-time systems. First, the fundamental concepts of linearity, time invariance, and causality as applied to discrete-time systems are discussed and tests are provided that would enable one to ascertain the properties of a given system from its rule of correspondence. The representation of these systems in terms of networks and signal flow graphs is then examined and analysis methods are presented that can be used to derive mathematical representations for discrete-time systems in the form of *difference equations*. Next, an elementary analysis method based on a *mathematical induction technique* is presented that can be used to find the time-domain response of a discrete-time system to a given input signal. An alternative representation of discrete-time systems known as the *state-space representation* follows, which provides alternative methods of analysis and design. The chapter concludes with an introduction to the concept of stability and outlines a basic test that can be used to establish whether a discrete-time system is stable or not.

The design of discrete-time systems that can perform DSP, e.g., digital filters, will form the subject matter of several chapters, starting with Chap. 8.

4.2 BASIC SYSTEM PROPERTIES

A discrete-time system can be represented by the block diagram of Fig. 4.1. Input $x(nT)$ and output $y(nT)$ are the *excitation* and *response* of the system, respectively. The response is related to the excitation by some rule of correspondence. We can indicate this fact notationally as

$$y(nT) = \mathcal{R}x(nT)$$

where $\mathcal{R}$ is an operator.

Depending on its rule of correspondence, a discrete-time system can be classified as linear or nonlinear, time invariant or time dependent, and causal or noncausal [1].

4.2.1 Linearity

A discrete-time system is *linear* if and only if it satisfies the conditions

$$\mathcal{R}\alpha x(nT) = \alpha\mathcal{R}x(nT) \tag{4.1a}$$

$$\mathcal{R}[x_1(nT) + x_2(nT)] = \mathcal{R}x_1(nT) + \mathcal{R}x_2(nT) \tag{4.1b}$$

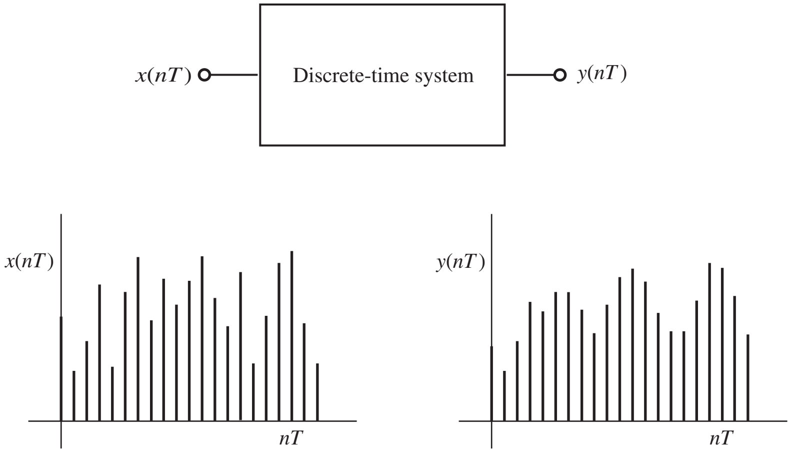


Figure 4.1 Discrete-time system.

for all possible values of α and all possible excitations $x_1(nT)$ and $x_2(nT)$. The condition in Eq. (4.1a) is referred to as the *proportionality* or *homogeneity* condition and that in Eq. (4.1b) as the *superposition* or *additivity* condition [1].

On applying first the superposition condition and then the proportionality condition, the response of a linear discrete-time system to an excitation $\alpha x_1(nT) + \beta x_2(nT)$, where α and β are arbitrary constants, can be expressed as

$$\begin{aligned} y(nT) &= \mathcal{R}[\alpha x_1(nT) + \beta x_2(nT)] = \mathcal{R}\alpha x_1(nT) + \mathcal{R}\beta x_2(nT) \\ &= \alpha\mathcal{R}x_1(nT) + \beta\mathcal{R}x_2(nT) \end{aligned}$$

Thus, the two conditions in Eqs. (4.1a) and (4.1b) can be combined into one, namely,

$$\mathcal{R}[\alpha x_1(nT) + \beta x_2(nT)] = \alpha\mathcal{R}x_1(nT) + \beta\mathcal{R}x_2(nT) \tag{4.1c}$$

If this condition is violated for any pair of excitations or any constant α or β, then the system is *nonlinear*.

The use of Eq. (4.1c) to check the linearity of a system tends to involve quite a bit of writing. A simpler approach that works well in a case where the system appears to be nonlinear is to first check whether the proportionality condition in Eq. (4.1a) is violated. If it is violated, then the work is done and the system can be classified as nonlinear. Otherwise, the superposition condition in Eq. (4.1b) must also be checked. Telltale signs of nonlinearity are terms like $|x(nT)|$ or $x^k(nT)$ in the rule of correspondence.

If the proportionality and superposition conditions hold for arbitrary excitations and arbitrary constants α and β, then the system is linear.

Example 4.1 (*a*) The response of a discrete-time system is of the form

$$y(nT) = \mathcal{R}x(nT) = 7x^2(nT - T)$$

Check the system for linearity. (*b*) Repeat part (*a*) if

$$y(nT) = \mathcal{R}x(nT) = (nT)^2 x(nT + 2T)$$

■ **Solution**

(*a*) A delayed version of the input signal appears squared in the characterization of the system and the proportionality condition is most likely violated. For an arbitrary constant α, we have

$$\mathcal{R}\alpha x(nT) = 7\alpha^2 x^2(nT - T)$$

On the other hand,

$$\alpha\mathcal{R}x(nT) = 7\alpha x^2(nT - T)$$

Clearly if $\alpha \neq 1$, then

$$\mathcal{R}\alpha x(nT) \neq \alpha \mathcal{R}x(nT)$$

that is, the proportionality condition is violated and, therefore, the system is nonlinear. ■

(*b*) For this case, the proportionality condition is not violated, as can be easily verified, and so we should use Eq. (4.1c), which combines both the proportionality and superposition rules. We can write

$$\begin{aligned}\mathcal{R}[\alpha x_1(nT) + \beta x_2(nT)] &= (nT)^2[\alpha x_1(nT + 2T) + \beta x_2(nT + 2T)]\\ &= \alpha(nT)^2 x_1(nT + 2T) + \beta(nT)^2 x_2(nT + 2T)\\ &= \alpha \mathcal{R}x_1(nT) + \beta \mathcal{R}x_2(nT)\end{aligned}$$

that is, the system is linear. ■

The squared term $(nT)^2$ may trick a few but it does not affect the linearity of the system since it is a time-dependent system parameter which is independent of the input signal.

4.2.2 Time Invariance

A discrete-time system is said to be *time invariant* if its response to an arbitrary excitation does not depend on the time of application of the excitation. The response of systems in general depends on a number of internal system parameters. In time-invariant systems, these parameters do not change with time.

Before we describe a test that can be used to check a discrete-time system for time invariance, the notion of a relaxed system needs to be explained. Systems in general have internal storage or memory elements that can store signal values. Such elements can serve as sources of internal signals and, consequently, a nonzero response may be produced even if the excitation is zero. If all the memory elements of a discrete-time system are empty or their contents are set to zero, the system is said to be *relaxed*. The response of such a system is zero for all n if the excitation is zero for all n.

Formally, an initially relaxed discrete-time system with excitation $x(nT)$ and response $y(nT)$, such that $x(nT) = y(nT) = 0$ for $n < 0$, is said to be time-invariant if and only if

$$\mathcal{R}x(nT - kT) = y(nT - kT) \tag{4.2}$$

for all possible excitations $x(nT)$ and all integers k. In other words, in a time-invariant discrete-time system, the response produced if the excitation $x(nT)$ is delayed by a period kT is numerically equal to the original response $y(nT)$ delayed by the same period kT. This must be the case, if the internal parameters of the system do not change with time. The behavior of a time-invariant discrete-time system is illustrated in Fig. 4.2. As can be seen, the response of the system to the delayed excitation shown in Fig. 4.2*b* is equal to the response shown in Fig. 4.2*a* delayed by kT.

A discrete-time system that does not satisfy the condition in Eq. (4.2) is said to be *time dependent*.

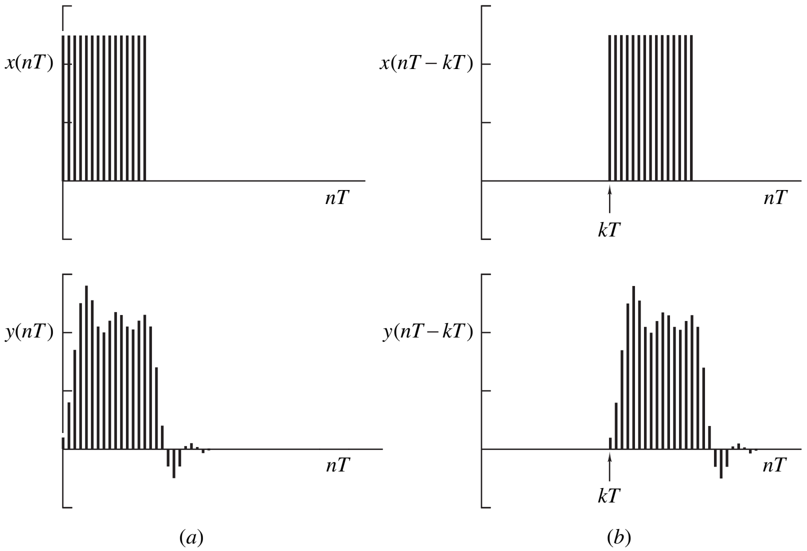


Figure 4.2 Time invariance: (*a*) Response to an excitation $x(nT)$, (*b*) response to a delayed excitation $x(nT - kT)$.

Example 4.2 (*a*) A discrete-time system is characterized by the equation

$$y(nT) = \mathcal{R}x(nT) = 2nTx(nT)$$

Check the system for time invariance. (*b*) Repeat part (*a*) if

$$y(nT) = \mathcal{R}x(nT) = 12x(nT - T) + 11x(nT - 2T)$$

■ **Solution**

(*a*) The response to a delayed excitation is

$$\mathcal{R}x(nT - kT) = 2nTx(nT - kT)$$

The delayed response is

$$y(nT - kT) = 2(nT - kT)x(nT - kT)$$

Clearly, for any $k \neq 0$

$$\mathcal{R}x(nT - kT) \neq y(nT - kT)$$

and, therefore, the system is time dependent. ■

(*b*) In this case

$$\begin{aligned}\mathcal{R}x(nT-kT) &= 12x[(n-k)T-T]+11x[(n-k)T-2T]\\ &= y(nT-kT)\end{aligned}$$

for all possible $x(nT)$ and all integers k, and so the system is time invariant. ■

In practical terms, one would first replace nT by $nT-kT$ in each and every occurrence of $x(nT)$ in the characterization of the system to obtain the response produced by a delayed excitation. Then one would replace each and every occurrence of nT by $nT-kT$ to obtain the delayed response. If the same expression is obtained in both cases, the system is time invariant. Otherwise, it is time dependent.

4.2.3 Causality

A discrete-time system is said to be *causal* if its response at a specific instant is independent of subsequent values of the excitation. More precisely, an initially relaxed discrete-time system in which $x(nT)=y(nT)=0$ for $n<0$ is said to be causal if and only if

$$\mathcal{R}x_1(nT)=\mathcal{R}x_2(nT) \qquad \text{for } n\le k \tag{4.3a}$$

for all possible distinct excitations $x_1(nT)$ and $x_2(nT)$, such that

$$x_1(nT)=x_2(nT) \qquad \text{for } n\le k \tag{4.3b}$$

Conversely, if

$$\mathcal{R}x_1(nT)\neq\mathcal{R}x_2(nT) \qquad \text{for } n\le k$$

for at least one pair of distinct excitations $x_1(nT)$ and $x_2(nT)$ such that

$$x_1(nT)=x_2(nT) \qquad \text{for } n\le k$$

for at least one value of k, then the system is *noncausal*.

The above causality test can be easily justified. If all possible pairs of excitations $x_1(nT)$ and $x_2(nT)$ that satisfy Eq. (4.3b) produce responses that are equal at instants $nT\le kT$, then the system response must depend only on values of the excitation at instants prior to nT, where $x_1(nT)$ and $x_2(nT)$ are specified to be equal, and the system is causal. This possibility is illustrated in Fig. 4.3. On the other hand, if at least two *distinct* excitations $x_1(nT)$ and $x_2(nT)$ that satisfy Eq. (4.3b) produce responses that are not equal at any instant $nT\le kT$, then the system response must depend on values of the excitation at instants subsequent to nT, since the differences between $x_1(nT)$ and $x_2(nT)$ occur after nT, and the system is noncausal.

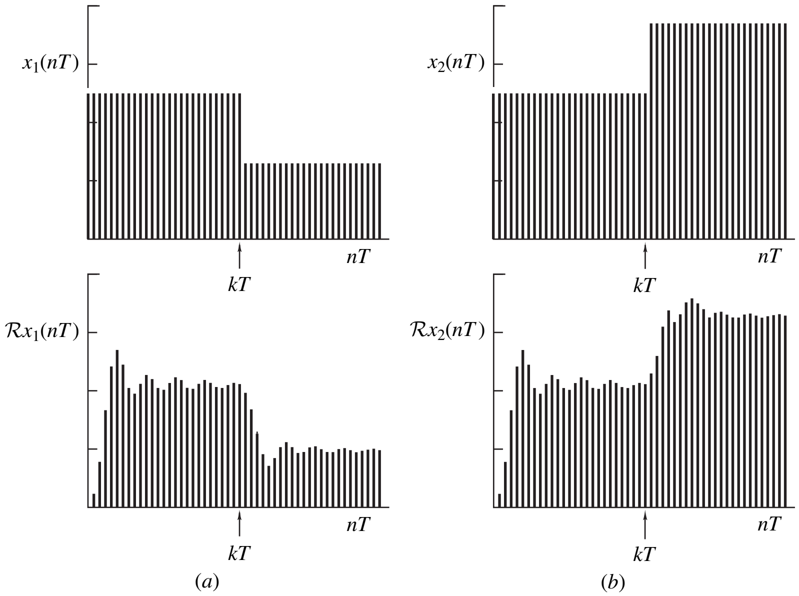


Figure 4.3 Causality: (*a*) Response to $x_1(nT)$, (*b*) response to $x_2(nT)$.

Example 4.3 (*a*) A discrete-time system is represented by

$$y(nT) = \mathcal{R}x(nT) = 3x(nT - 2T) + 3x(nT + 2T)$$

Check the system for causality. (*b*) Repeat part (*a*) if

$$y(nT) = \mathcal{R}x(nT) = 3x(nT - T) - 3x(nT - 2T)$$

■ **Solution**

(*a*) Let $x_1(nT)$ and $x_2(nT)$ be distinct excitations that satisfy Eq. (4.3b) and assume that

$$x_1(nT) \neq x_2(nT) \qquad \text{for } n > k$$

For $n = k$

$$\mathcal{R}x_1(nT)|_{n=k} = 3x_1(kT - 2T) + 3x_1(kT + 2T)$$
$$\mathcal{R}x_2(nT)|_{n=k} = 3x_2(kT - 2T) + 3x_2(kT + 2T)$$

and since we have assumed that $x_1(nT) \neq x_2(nT)$ for $n > k$, it follows that $x_1(kT + 2T) \neq x_2(kT + 2T)$ and thus

$$3x_1(kT + 2T) \neq 3x_2(kT + 2T)$$

Therefore,

$$\mathcal{R}x_1(nT) \neq \mathcal{R}x_2(nT) \qquad \text{for } n = k$$

that is, the system is noncausal. ■

(*b*) For this case

$$\mathcal{R}x_1(nT) = 3x_1(nT - T) - 3x_1(nT - 2T)$$
$$\mathcal{R}x_2(nT) = 3x_2(nT - T) - 3x_2(nT - 2T)$$

If $n \leq k$, then $n - 1, n - 2 < k$ and so

$$x_1(nT - T) = x_2(nT - T) \qquad \text{and} \qquad x_1(nT - 2T) = x_2(nT - 2T)$$

for $n \leq k$ or

$$\mathcal{R}x_1(nT) = \mathcal{R}x_2(nT) \qquad \text{for } n \leq k$$

that is, the system is causal. ■

Noncausality is often recognized by the appearance of one or more terms such as $x(nT + |k|T)$ in the characterization of the system. In such a case, all one would need to do is to find just one pair of distinct signals that satisfy Eq. (4.3b) but violate Eq. (4.3a) for just one value of n, as was done in Example 4.3(*a*). However, to demonstrate causality one would need to show that Eq. (4.3a) is satisfied for all possible distinct signals that satisfy Eq. (4.3b) for all possible values of $n \leq k$. Demonstrating that Eq. (4.3a) is satisfied for just one value of n, say, k, is not sufficient.

Note that the presence of one or more terms like $x(nT + |k|T)$ in the system equation is neither a necessary nor a sufficient condition for causality. This point is illustrated by the following example.

Example 4.4 A discrete-time system is characterized by the following equation

$$y(nT + 2T) = e^{nT} + 5x(nT + 2T)$$

Check the system for (a) linearity, (b) time invariance, and (c) causality.

■ **Solution**

By letting $n = n' - 2$ and then replacing n' by n, the system equation can be expressed as

$$y(nT) = \mathcal{R}x(nT) = e^{(n-2)T} + 5x(nT)$$

(*a*) We note that

$$\mathcal{R}[\alpha x(nT)] = e^{(n-2)T} + 5\alpha x(nT)$$

On the other hand,

$$\alpha \mathcal{R}x(nT) = \alpha[e^{(n-2)T} + 5x(nT)] = \alpha e^{(n-2)T} + 5\alpha x(nT + 2T)$$

For $\alpha \neq 1$, we have $e^{(n-2)T} \neq \alpha e^{(n-2)T}$ and hence

$$\mathcal{R}[\alpha x(nT)] \neq \alpha \mathcal{R}x(nT)$$

Therefore, the proportionality condition is violated and the system is nonlinear. ■

(*b*) The response to a delayed excitation is

$$\mathcal{R}x(nT - kT) = e^{(n-2)T} + 5x(nT - kT)$$

The delayed response is

$$y(nT - kT) = e^{nT-2T-kT} + 5x(nT - kT)$$

For any $k \neq 0$, we have $e^{(n-2)T} \neq e^{nT-2T-kT}$ and hence

$$y(nT - kT) \neq \mathcal{R}x(nT - kT)$$

Therefore, the system is time dependent. ■

(*c*) Let $x_1(nT)$ and $x_2(nT)$ be two arbitrary distinct excitations that satisfy Eq. (4.3b). The responses produced by the two signals are given by

$$\mathcal{R}x_1(nT) = e^{(n-2)T} + 5x_1(nT)$$

$$\mathcal{R}x_2(nT) = e^{(n-2)T} + 5x_2(nT)$$

and since

$$x_1(nT) = x_2(nT) \qquad \text{for } n \leq k$$

we have

$$\mathcal{R}x_1(nT) = \mathcal{R}x_2(nT) \qquad \text{for } n \leq k$$

that is, the condition for causality is satisfied and, therefore, the system is causal. ■

Discrete-time systems come in all shapes and forms. Systems that operate as digital filters are almost always linear although there are some highly specialized types of digital filters that are basically nonlinear. Most of the time, nonlinearity manifests itself as an imperfection that needs to be eliminated or circumvented.

In continuous-time systems, nine times out of ten, time dependence is an undesirable imperfection brought about by drifting component values that needs to be obviated. However, in discrete-time systems it turns out to be a most valuable property. Through the use of time dependence, adaptive systems such as adaptive filters can be built whose behavior can be changed or optimized online.

Causality is a prerequisite property for real-time systems because the present output cannot depend on future values of the input, which are not available. However, in nonreal-time applications no such problem is encountered as the numerical values of the signal to be processed are typically stored in a computer memory or mass storage device and are, therefore, readily accessible at any time during the processing. Knowledge of causality is important from another point of view. Certain design methods for digital filters, for example, those in Chaps. 9 and 15, yield noncausal designs and for a real-time application, the designer must know how to convert the noncausal filter obtained to a causal one.

4.3 CHARACTERIZATION OF DISCRETE-TIME SYSTEMS

Continuous-time systems are characterized in terms of differential equations. Discrete-time systems, on the other hand, are characterized in terms of *difference* equations. Two types of discrete-time systems can be identified: *nonrecursive* and *recursive*.

4.3.1 Nonrecursive Systems

In a nonrecursive discrete-time system, the output at any instant depends on a set of values of the input. In the most general case, the response of such a system at instant nT is a function of $x(nT - MT), \ldots, x(nT), \ldots, x(nT + KT)$, that is,

$$y(nT) = f\{x(nT - MT), \ldots, x(nT), \ldots, x(nT + KT)\}$$

where M and K are positive integers. If we assume linearity and time invariance, $y(nT)$ can be expressed as

$$y(nT) = \sum_{i=-K}^{M} a_i x(nT - iT) \tag{4.4}$$

where a_i for $i = -K, (-K + 1), \ldots, M$ are constants. If instant nT were taken to be the present, then the present response would depend on the past M values, the present value, and the future K values of the excitation. Equation (4.4) is a linear difference equation with constant coefficients of order $M + K$, and the system represented by this equation is said to be of the same order.

If $K > 0$ in Eq. (4.4), then $y(nT)$ would depend on $x(nT + T),\ x(nT + 2T), \ldots, x(nT + KT)$ and, obviously, the deference equation would represent a noncausal system but if $K = 0$, the representation of an Mth-order causal system would be obtained.

4.3.2 Recursive Systems

A recursive discrete-time system is a system whose output at any instant depends on a set of values of the input as well as a set of values of the output. The response of a fairly general recursive, linear, time-invariant, discrete-time system is given by

$$y(nT) = \sum_{i=-K}^{M} a_i x(nT - iT) - \sum_{i=1}^{N} b_i y(nT - iT) \tag{4.5}$$

that is, if instant nT were taken to be the present, then the present response would be a function of the past M values, the present value, and the future K values of the excitation as well as the past N values of the response. The dependence of the response on a number of past values of the response implies that a recursive discrete-time system must involve feedback from the output to the input. The order of a recursive discrete-time system is the same as the order of its difference equation, as in a nonrecursive system, and it is the larger of $M + K$ and $N + K$. The difference equation in Eq. (4.5) for the case where $K = M = N = 2$ is illustrated in Fig. 4.4.

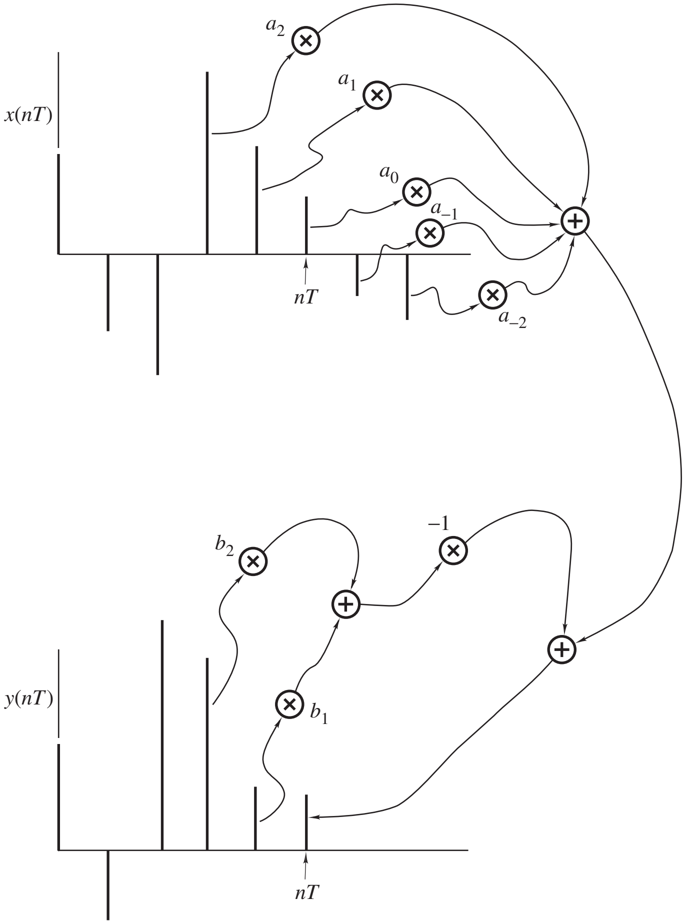


Figure 4.4 Graphical representation of recursive difference equation.

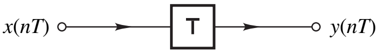
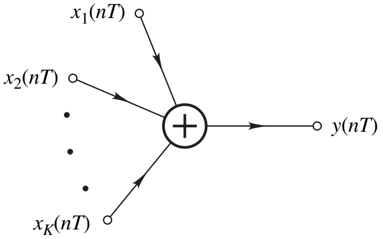
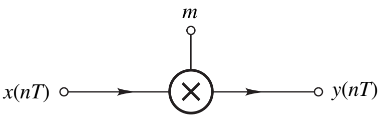
Note that Eq. (4.5) simplifies to Eq. (4.4) if $b_i = 0$ for $1, 2, \ldots, N$, and essentially the nonrecursive discrete-time system is a special case of the recursive one.

4.4 DISCRETE-TIME SYSTEM NETWORKS

The basic elements of discrete-time systems are the *adder*, the *multiplier*, and the *unit delay*. The characterizations and symbols for these elements are given in Table 4.1. Ideally, the adder produces the sum of its inputs and the multiplier multiplies its input by a constant instantaneously. The unit delay, on the other hand, is a memory element that can store just one number. At instant nT, in response to a synchronizing clock pulse, it delivers its content to the output and then updates its content with the present input. The device freezes in this state until the next clock pulse. In effect, on the clock pulse, the unit delay delivers its previous input to the output.

The basic discrete-time elements can be implemented in analog or digital form and many digital configurations are possible depending on the design methodology, the number system, and the type of arithmetic used. Although analog discrete-time elements may be used in certain specialized applications, for example, for the implementation of neural networks, discrete-time systems that are used for DSP are almost always digital and, therefore, the adder, multiplier, and unit delay are digital circuits. A practical approach would be to implement adders and multipliers through the use of parallel combinational circuits and unit delays through the use of delay flip-flops.[1] Under ideal

Table 4.1 Elements of discrete-time systems

Element	Symbol	Equation
Unit delay		$y(nT) = x(nT-T)$
Adder		$y(nT) = \sum_{i=1}^{K} x_i(nT)$
Multiplier		$y(nT) = mx(nT)$

[1] *Also known as D flip-flops.*

conditions, the various devices produce their outputs instantaneously, as mentioned above, but in practice the three types of devices introduce a small delay known as the *propagation delay* due to the fact that electrical signals take a certain amount of time to propagate from the input to the output of the device.

Collections of unit delays, adders, and multipliers can be interconnected to form *discrete-time networks*.

4.4.1 Network Analysis

The *analysis* of a discrete-time network, which is the process of deriving the difference equation characterizing the network, can be carried out by using the element equations given in Table 4.1. Network analysis can often be simplified by using the *shift operator* $\mathcal{E}^r$ which is defined by

$$\mathcal{E}^r x(nT) = x(nT + rT)$$

The shift operator is one of the basic operators of numerical analysis which will advance or delay a signal depending on whether r is positive or negative. Its main properties are as follows:

1. Since

$$\begin{aligned}\mathcal{E}^r[a_1x_1(nT) + a_2x_2(nT)] &= a_1x_1(nT + rT) + a_2x_2(nT + rT)\\ &= a_1\mathcal{E}^r x_1(nT) + a_2\mathcal{E}^r x_2(nT)\end{aligned}$$

we conclude that $\mathcal{E}^r$ is a linear operator which *distributes* with respect to a sum of functions of nT.

2. Since

$$\begin{aligned}\mathcal{E}^r\mathcal{E}^p x(nT) &= \mathcal{E}^r x(nT + pT) = x(nT + rT + pT)\\ &= \mathcal{E}^{r+p}x(nT)\end{aligned}$$

the shift operator obeys the usual *law of exponents*.

3. If $x_2(nT) = \mathcal{E}^r x_1(nT)$ then $\mathcal{E}^{-r}x_2(nT) = x_1(nT)$ for all $x_1(nT)$, and if $x_1(nT) = \mathcal{E}^{-r}x_2(nT)$ then $x_2(nT) = \mathcal{E}^r x_1(nT)$ for all $x_2(nT)$. Therefore, $\mathcal{E}^{-r}$ is the inverse of $\mathcal{E}^r$ and vice versa, that is,

$$\mathcal{E}^{-r}\mathcal{E}^r = \mathcal{E}^r\mathcal{E}^{-r} = 1$$

4. A linear combination of powers of $\mathcal{E}$ defines a meaningful operator, e.g., if

$$f(\mathcal{E}) = 1 + a_1\mathcal{E} + a_2\mathcal{E}^2$$

then

$$\begin{aligned}f(\mathcal{E})x(nT) &= (1 + a_1\mathcal{E} + a_2\mathcal{E}^2)x(nT)\\ &= x(nT) + a_1x(nT + T) + a_2x(nT + 2T)\end{aligned}$$

Further, given an operator $f(\mathcal{E})$ of the above type, an *inverse* operator $f(\mathcal{E})^{-1}$ may be defined such that

$$f(\mathcal{E})^{-1}x(\mathcal{E}) = x(\mathcal{E})f(\mathcal{E})^{-1} = 1$$

5. If $f_1(\mathcal{E})$, $f_2(\mathcal{E})$, and $f_3(\mathcal{E})$ are operators that comprise linear combinations of powers of $\mathcal{E}$, then they satisfy the *distributive, commutative*, and *associative* laws of algebra, that is,

$$f_1(\mathcal{E})[f_2(\mathcal{E}) + f_3(\mathcal{E})] = f_1(\mathcal{E})f_2(\mathcal{E}) + f_1(\mathcal{E})f_3(\mathcal{E})$$
$$f_1(\mathcal{E})f_2(\mathcal{E}) = f_2(\mathcal{E})f_1(\mathcal{E})$$
$$f_1(\mathcal{E})[f_2(\mathcal{E})f_3(\mathcal{E})] = [f_1(\mathcal{E})f_2(\mathcal{E})]f_3(\mathcal{E})$$

The above operators can be used to construct more complicated operators of the form

$$F(\mathcal{E}) = f_1(\mathcal{E})f_2(\mathcal{E})^{-1} = f_2(\mathcal{E})^{-1}f_1(\mathcal{E})$$

which may also be expressed as

$$F(\mathcal{E}) = \frac{f_1(\mathcal{E})}{f_2(\mathcal{E})}$$

without danger of ambiguity.

Owing to the above properties, the shift operator can be treated like an ordinary algebraic quantity [2], and operators that are linear combinations of powers of $\mathcal{E}$ can be treated as polynomials which can even be factorized. For example, the difference equation of a recursive system given in Eq. (4.5) can be expressed as

$$y(nT) = \left(\sum_{i=-K}^{M} a_i\mathcal{E}^{-i}\right)x(nT) - \left(\sum_{i=1}^{N} b_i\mathcal{E}^{-i}\right)y(nT)$$

and, therefore, the recursive system can represented in terms of operator notation as

$$y(nT) = \mathcal{R}x(nT) \tag{4.6}$$

where $\mathcal{R}$ is an operator given by

$$\mathcal{R} = \left(\frac{\sum_{i=-K}^{M} a_i\mathcal{E}^{-i}}{1 + \sum_{i=1}^{N} b_i\mathcal{E}^{-i}}\right)$$

The application of the above principles in the analysis of discrete-time networks is illustrated in Example 4.5(*b*) below.

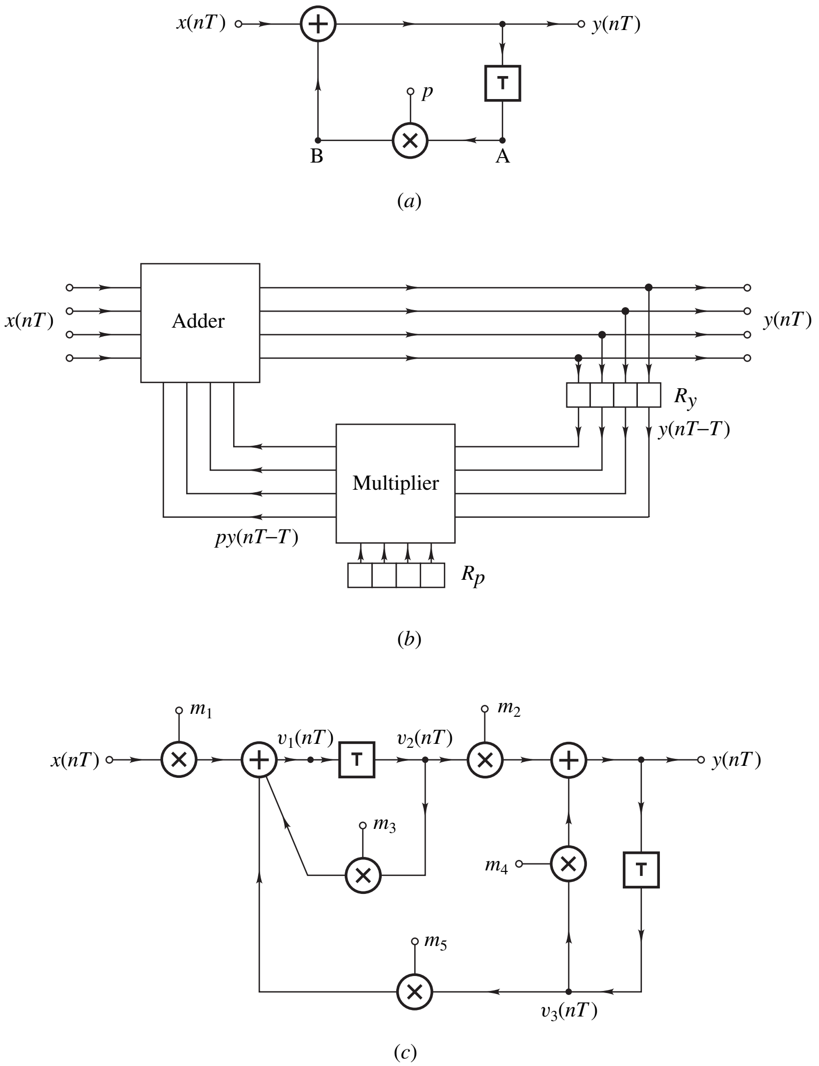


Figure 4.5 Discrete-time networks (Example 4.5): (*a*) First-order system, (*b*) implementation of first-order system, (*c*) second-order system.

Example 4.5 (*a*) Analyze the network of Fig. 4.5*a*. (*b*) Repeat part (*a*) for the network of Fig. 4.5*c*.

■ **Solution**

(*a*) From Fig. 4.5*a*, the signals at nodes A and B are $y(nT - T)$ and $py(nT - T)$, respectively. Thus,

$$y(nT) = x(nT) + py(nT - T) \quad ■ \tag{4.7}$$

(*b*) From Fig. 4.5*c*, we obtain

$$v_1(nT) = m_1x(nT) + m_3v_2(nT) + m_5v_3(nT)$$
$$v_2(nT) = \mathcal{E}^{-1}v_1(nT) \qquad v_3(nT) = \mathcal{E}^{-1}y(nT)$$
$$y(nT) = m_2v_2(nT) + m_4v_3(nT)$$

and on eliminating $v_2(nT)$ and $v_3(nT)$ in $v_1(nT)$ and $y(nT)$, we have

$$(1 - m_3\mathcal{E}^{-1})v_1(nT) = m_1x(nT) + m_5\mathcal{E}^{-1}y(nT) \tag{4.8}$$

and

$$(1 - m_4\mathcal{E}^{-1})y(nT) = m_2\mathcal{E}^{-1}v_1(nT) \tag{4.9}$$

On multiplying both sides of Eq. (4.9) by $(1 - m_3\mathcal{E}^{-1})$, we get

$$\begin{aligned}(1 - m_3\mathcal{E}^{-1})(1 - m_4\mathcal{E}^{-1})y(nT) &= (1 - m_3\mathcal{E}^{-1})m_2\mathcal{E}^{-1}v_1(nT)\\ &= m_2\mathcal{E}^{-1}(1 - m_3\mathcal{E}^{-1})v_1(nT)\end{aligned}$$

and on eliminating $(1 - m_3\mathcal{E}^{-1})v_1(nT)$ using Eq. (4.8), we have

$$[1 - (m_3 + m_4)\mathcal{E}^{-1} + m_3m_4\mathcal{E}^{-2}]y(nT) = m_1m_2\mathcal{E}^{-1}x(nT) + m_2m_5\mathcal{E}^{-2}y(nT)$$

Therefore,

$$y(nT) = a_1x(nT - T) + b_1y(nT - T) + b_2y(nT - 2T) \quad \blacksquare$$

where

$$a_1 = m_1m_2 \qquad b_1 = m_3 + m_4 \qquad \text{and} \qquad b_2 = m_2m_5 - m_3m_4$$

4.4.2 Implementation of Discrete-Time Systems

The mode of operation of discrete-time systems depends on the implementation of their constituent elements. A relatively simple paradigm to explain is the case whereby adders and multipliers are built as parallel combinational digital circuits, and the unit delays are constructed using delay flip-flops. If the discrete-time system in Fig. 4.5*a* were implemented according to this paradigm and the signals and coefficients were assumed to be represented, say, by 4-bit signed binary digits, then an implementation of the type shown in Fig. 4.5*b* would be obtained where the unit delay is an array of four clocked D flip-flops and R_P is a read-only register in which the digits of coefficient p are stored. Let us assume that the adder, multiplier, and unit delay have propagation delays τ_A, τ_M, and τ_{UD}, respectively.

At a sampling instant nT, a very brief clock pulse triggers the unit delay to deliver its content $y(nT - T)$ to its output and the content of the unit delay is replaced by the current value of $y(nT)$ in τ_{UD} s. The output of the unit delay will cause the correct product $py(nT - T)$ to appear at the output of the multiplier in τ_M s. This product will cause a new sum $x(nT) + py(nT - T)$ to appear at input of the unit delay and at the output of the system in τ_A s. By that time, the clock pulse would have disappeared and the unit delay would be in a dormant state with the previous system output recorded in its memory and the present output at the input of the unit delay. Obviously, this scheme will work out in practice only if the outputs of the unit delay, multiplier, and adder reach steady state before the next clock pulse which will occur at the next sampling instant. This implies that the sampling period T must be long enough to ensure that $T > \tau_M + \tau_A$. Otherwise, the unit delay will record an erroneous value for the output of the adder.

The system in Fig. 4.5*c* would operate in much the same way. Just before a sampling instant, signals are deemed to be in steady state throughout the system. When a clock pulse is received simultaneously by the two unit delays, the unit delays output their contents and their inputs overwrite their contents after a certain propagation delay. The outputs of the unit delays then propagate through the multipliers and adders and after as certain propagation delay, new numerical values appear at the inputs of the unit delays but by then, in the absence of a clock pulse, the unit delays will be dormant. We note that there are signal paths between the output of each unit delay and its input, between the output of the left and the input of the right unit delay, and between the output of the right and the input of the left unit delay. We note also that each of these signal paths involves a multiplier in series with an adder and thus the sampling period should be long enough to ensure that $T > \tau_m + \tau_a$, as in the first-order system of Fig. 4.5*a*.

This simple analysis has shown that the propagation delays of the multipliers and adders impose a lower limit on the sampling period T which translates into an upper limit on the sampling frequency $f_s = 1/T$; consequently, high sampling frequencies can be used only if fast hardware with short propagation delays is available, as may be expected.

4.4.3 Signal Flow-Graph Analysis

Given a discrete-time network, a corresponding topologically equivalent signal flow graph can be readily deduced by marking and labeling all the nodes of the network on a blank sheet of paper and then replacing

- Each adder by a node with one outgoing branch and as many incoming branches as there are inputs to the adder
- Each distribution node by a distribution node
- Each multiplier by a directed branch with transmittance equal to the constant of the multiplier
- Each direct transmission path by a directed branch with transmittance equal to unity
- Each unit delay by a directed branch with transmittance equal to the shift operator $\mathcal{E}^{-1}$.

For example, the signal flow graph of the network shown in Fig. 4.6*a* can be drawn by marking nodes A, C, F, G, H, D, and E on a sheet of paper and then replacing unit-delays, adders, multipliers, and signal paths by the appropriate nodes and branches, as depicted in Fig. 4.6*b*.

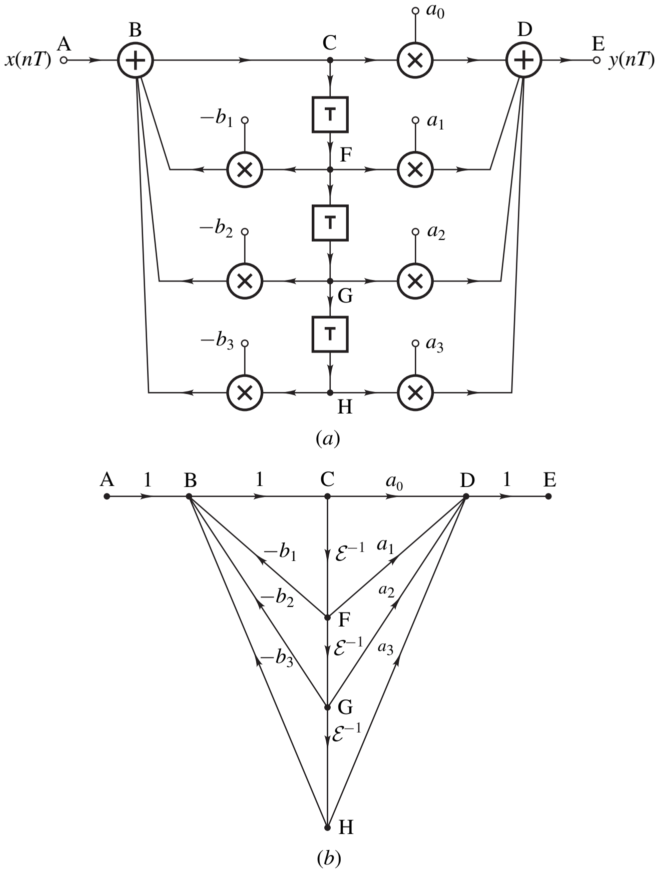


Figure 4.6 (*a*) Discrete-time network, (*b*) signal flow graph.

As can be seen in Fig. 4.6*b*, signal flow graphs provide a compact and easy-to-draw graphical representation for discrete-time networks and can, in addition, be used to analyze networks through the use of some well-established signal *flow-graph* methods [3–5]. Two signal flow-graph methods that are readily applicable for the analysis of discrete-time networks are the *node-elimination method* and *Mason's method*.

NODE ELIMINATION METHOD. In the node elimination method, the given signal flow graph is reduced down to a single branch between the input and output nodes through a sequence of node eliminations [5] and simplifications, and the transmittance of the last remaining branch is the operator of the network. From the functional relationship provided by this simplified signal flow graph, the difference equation of the network can be readily deduced.

Node elimination can be accomplished by applying a small set of rules, as follows:

Rule 1: K branches in series with transmittances $T_1, T_2, \ldots, T_K$ can be replaced by a single branch with transmittance $T_1 T_2 \ldots T_K$, as shown in Fig. 4.7*a*.

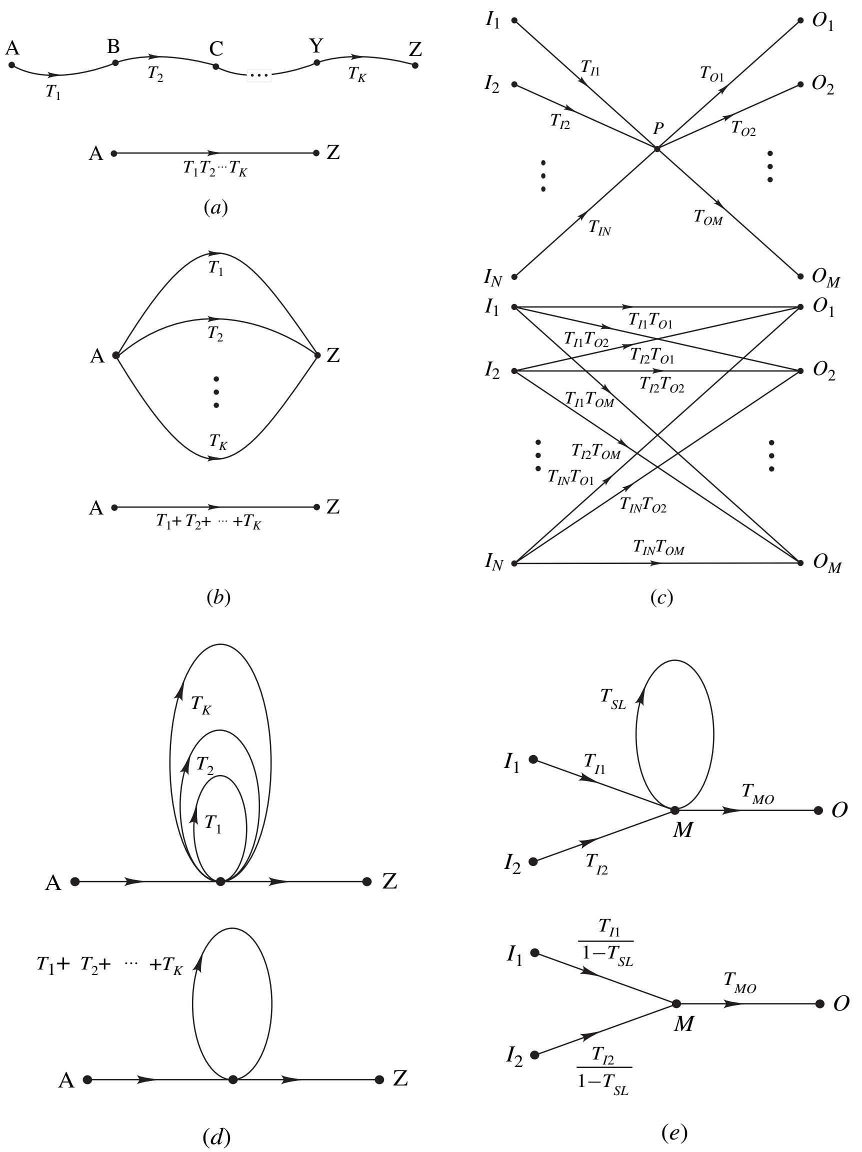


Figure 4.7 Node elimination rules: (*a*) Rule 1, (*b*) Rule 2, (*c*) Rule 3, (*d*) Rule 4*a*, (*e*) Rule 4*b*.

Rule 2: K branches in parallel with transmittances $T_1,\ T_2,\ldots,T_K$ can be replaced by a single branch with transmittance $T_1+T_2+\cdots+T_K$, as illustrated in Fig. 4.7*b*.

Rule 3: A node with N incoming branches with transmittances $T_{I1},\ T_{I2},\ldots,T_{IN}$ and M outgoing branches with transmittances $T_{O1},\ T_{O2},\ldots,T_{OM}$ can be replaced by $N \times M$ branches with transmittances $T_{I1}T_{O1},\ T_{I1}T_{O2},\ldots,T_{IN}T_{OM}$ as illustrated in Fig. 4.7*c*.

Rule 4*a*: K self-loops at a given node with transmittances $T_1,\ T_2, \dots, T_K$ can be replaced by a single self-loop with transmittance $T_1 + T_2 + \cdots + T_K$, as illustrated in Fig. 4.7*d*.

Rule 4*b*: A self-loop at a given node with transmittance T_{SL} can be eliminated by dividing the transmittance of each and every incoming branch by $1 - T_{SL}$ as shown in Fig. 4.7*e*.

Actually, Rule 4*a* is a special case of Rule 2 since a self-loop is, in effect, a branch that starts from and ends on one and the same node.

The above rules constitute a graphical way of doing algebra and, therefore, their validity can be readily demonstrated by showing that the equations of the simplified flow graph can be obtained from those of the original flow graph. For example, the equations of the bottom flow graph in Fig. 4.7*e* are given by

$$\begin{aligned} M &= \frac{T_{I1}}{1 - T_{SL}} I_1 + \frac{T_{I2}}{1 - T_{SL}} I_2 \\ O &= T_{MO} M \end{aligned}$$

and can be obtained from the equations of the top flow graph in Fig. 4.7*e*, that is,

$$\begin{aligned} M &= T_{I1} I_1 + T_{I2} I_2 + T_{SL} M \\ O &= T_{MO} M \end{aligned}$$

by moving the term $T_{SL} M$ in the first equation to the left-hand side and then dividing both sides by the factor $1 - T_{SL}$.

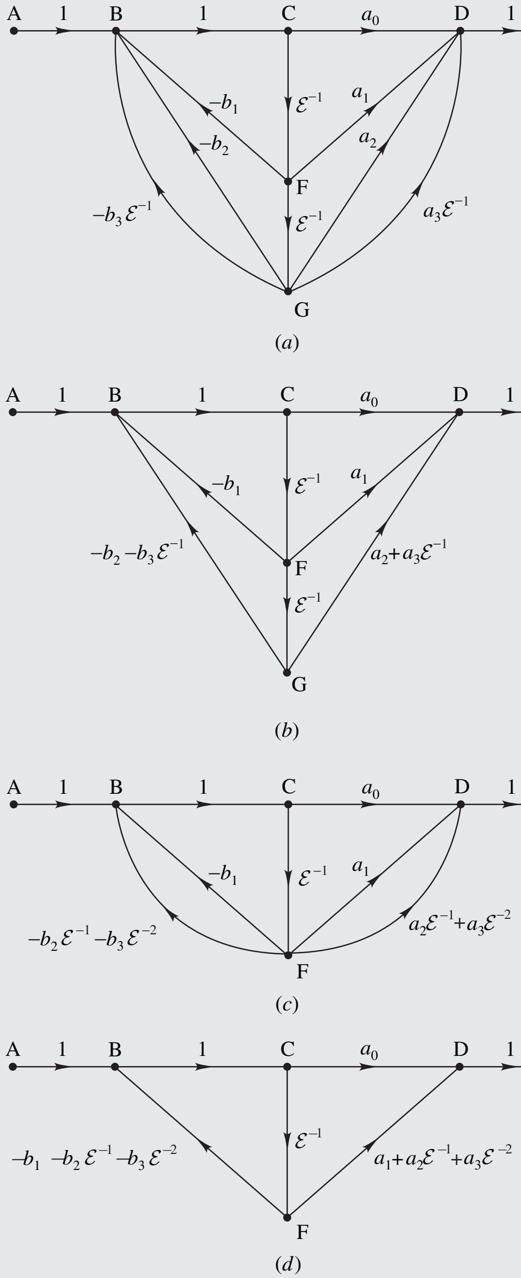
Example 4.6 Find the difference equation of the discrete-time network shown in Fig. 4.6*a* by using the node elimination method.

■ **Solution**

Eliminating node H in Fig. 4.6*b* using Rule 3 yields the signal flow graph of Fig. 4.8*a* and on combining parallel branches by using Rule 2, the graph of Fig. 4.8*b* can be deduced. Applying Rule 3 to node G in Fig. 4.8*b* yields the graph in Fig. 4.8*c* which can be simplified to the graph in Fig. 4.8*d* by combining the parallel branches.

Applying Rule 3 to node F in Fig. 4.8*d* yields the graph of Fig. 4.8*e* and on combining the parallel branches and then eliminating node C, the graph Fig. 4.8*f* can be obtained. In Fig. 4.8*f*, we note that there is a self-loop at node B and on using Rule 4*b* the graph of Fig. 4.8*g* is deduced, which can be simplified to the graph of Fig. 4.8*h* using Rule 1. Hence,

$$y(nT) = \left(\frac{T_1}{1 - T_2}\right) x(nT) \tag{4.10}$$



(*a*)

(*b*)

(*c*)

(*d*)

Figure 4.8 Signal flow graph reduction method (Example 4.6): (*a*) Elimination of node H, (*b*) combining of parallel branches, (*c*) Elimination of node G, (*d*) combining of parallel branches.

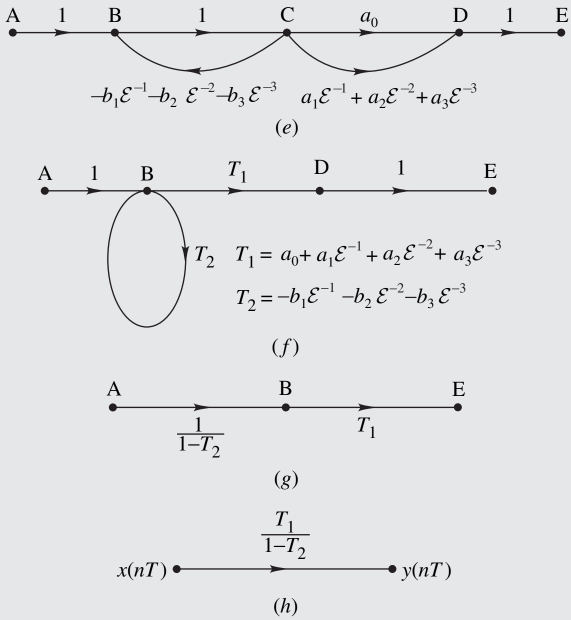

A
1
B
1
C
a_0
D
1
E
$-b_1\mathcal{E}^{-1}-b_2\,\mathcal{E}^{-2}-b_3\mathcal{E}^{-3}$
$a_1\mathcal{E}^{-1}+a_2\mathcal{E}^{-2}+a_3\mathcal{E}^{-3}$
(e)
A
1
B
T_1
D
1
E
T_2
$T_1 = a_0+a_1\mathcal{E}^{-1}+a_2\mathcal{E}^{-2}+a_3\mathcal{E}^{-3}$
$T_2 = -b_1\mathcal{E}^{-1}-b_2\,\mathcal{E}^{-2}-b_3\mathcal{E}^{-3}$
(f)
A
B
E
$\frac{1}{1-T_2}$
T_1
(g)
$\frac{T_1}{1-T_2}$
$x(nT)$
$y(nT)$
(h)

Figure 4.8 Cont'd (*e*) elimination of node F, (*f*) combining of parallel branches and elimination of node C, (*g*) elimination of self-loop and node D, (*h*) combining of series branches.

or

$$(1 - T_2)y(nT) = T_1x(nT) \tag{4.11}$$

Therefore,

$$y(nT) = T_1x(nT) + T_2y(nT) \tag{4.12}$$

and since

$$T_1 = a_0 + a_1\mathcal{E}^{-1} + a_2\mathcal{E}^{-2} + a_3\mathcal{E}^{-3} \tag{4.13}$$

and

$$T_2 = -[b_1\mathcal{E}^{-1} + b_2\mathcal{E}^{-2} + b_3\mathcal{E}^{-3}] \tag{4.14}$$

we obtain

$$\begin{aligned} y(nT) = a_0x(nT) + a_1x(nT - T) + a_2x(nT - 2T) + a_3x(nT - 3T) \\ - b_1y(nT - T) - b_2y(nT - 2T) - b_3y(nT - 3T) \quad \blacksquare \end{aligned}$$

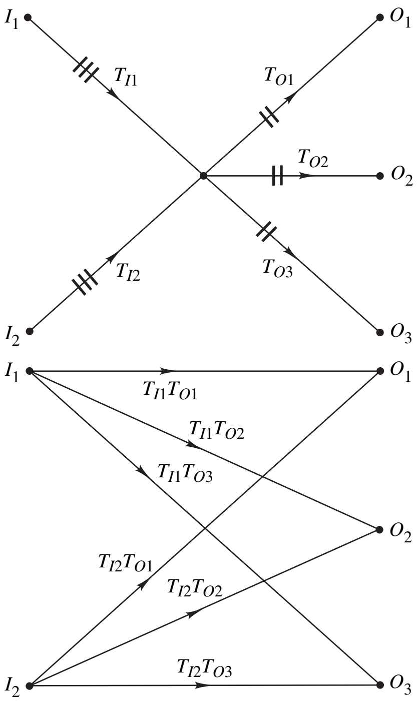


Figure 4.9 (*a*) Avoidance of node elimination errors.

The amount of work required to simplify a signal flow graph tends to depend on the order in which nodes are eliminated. It turns out that the required effort is reduced if at any one time one eliminates the node that would result in the smallest number of new paths. The number of new paths for a given node is equal to the number of incoming branches times the number of outgoing branches.

The most likely source of errors in signal flow graph simplification is the omission of one or more of the new paths generated by Rule 3. This problem can be circumvented to a large extent by drawing strokes on the branches involved as each new path is identified. At the end of the elimination process, each incoming branch should have as many strokes as there are outgoing branches and each outgoing branch should have as many strokes as there are incoming branches, as illustrated in Fig. 4.9. If the strokes do not tally, then the appropriate node elimination needs to be checked.

MASON'S METHOD. An alternative signal flow-graph analysis method is one based on the so-called *Mason's gain formula* [5, 6]. If i and j are arbitrary nodes in a signal flow graph representing a discrete-time network, then the response at node j produced by an excitation applied at node i is given by Mason's gain formula as

$$y_j(nT) = \left(\frac{1}{\Delta} \sum_k T_k \Delta_k \right) x_i(nT) \tag{4.15}$$

Parameter T_k is the transmittance of the kth direct path between nodes i and j, Δ is the determinant of the flow graph, and Δ_k is the determinant of the subgraph that does not touch (has no nodes or branches in common with) the kth direct path between nodes i and j.

The graph determinant Δ is given by

$$\Delta = 1 - \sum_u L_{u1} + \sum_v P_{v2} - \sum_w P_{w3} + \cdots$$

where L_{u1} is the loop transmittance of the uth loop, P_{v2} is the product of the loop transmittances of the vth pair of nontouching loops (loops that have neither nodes nor branches in common), P_{w3} is the product of loop transmittances of the wth triplet of nontouching loops, and so on.

The subgraph determinant Δ_k can be determined by applying the formula for Δ to the subgraph that does not touch the kth direct path between nodes i and j.

The derivation of Mason's formula can be found in [6]. Its application is illustrated by the following example.

Example 4.7 Analyze the discrete-time network of Fig. 4.6*a* using Mason's method.

■ **Solution**

From Fig. 4.6*b*, the direct paths of the flow graph are ABCDE, ABCFDE, ABCFGDE, and ABCFGHDE and hence

$$T_1 = a_0 \qquad T_2 = a_1\mathcal{E}^{-1} \qquad T_3 = a_2\mathcal{E}^{-2} \qquad T_4 = a_3\mathcal{E}^{-3}$$

The loops of the graph are BCFB, BCFGB, and BCFGHB and hence

$$L_{11} = -b_1\mathcal{E}^{-1} \qquad L_{21} = -b_2\mathcal{E}^{-2} \qquad L_{31} = -b_3\mathcal{E}^{-3}$$

All loops are touching since branch BC is common to all of them, and so

$$P_{v2} = P_{w3} = \cdots = 0$$

Hence

$$\Delta = 1 + b_1\mathcal{E}^{-1} + b_2\mathcal{E}^{-2} + b_3\mathcal{E}^{-3}$$

The determinants of the subgraphs Δ_k, $k = 1$, 2, 3, and 4, can similarly be determined by identifying each subgraph that does not touch the kth direct path. As can be seen in Fig. 4.6*b*, branch BC is common to all direct paths between input and output and, therefore, it does not appear in any of the subgraphs. Consequently, no loops are present in the k subgraphs and so

$$\Delta_1 = \Delta_2 = \Delta_3 = \Delta_4 = 1$$

Using Mason's formula given by Eq. (4.15), we obtain

$$y(nT) = \left(\frac{\sum_{i=0}^{3} a_i \mathcal{E}^{-i}}{1 + \sum_{i=1}^{3} b_i \mathcal{E}^{-i}} \right) x(nT)$$

or

$$y(nT) = \left(\sum_{i=0}^{3} a_i \mathcal{E}^{-i} \right) x(nT) - \left(\sum_{i=1}^{3} b_i \mathcal{E}^{-i} \right) y(nT) \quad \blacksquare$$

4.5 INTRODUCTION TO TIME-DOMAIN ANALYSIS

The time-domain response of simple discrete-time systems can be determined by solving the difference equation directly using mathematical induction. Although this approach is somewhat primitive, it demonstrates the mode by which discrete-time systems operate. The approach needs nothing more sophisticated than basic algebra as illustrated by the following examples.

Example 4.8 (*a*) Find the impulse response of the system in Fig. 4.5*a*. The system is initially relaxed, that is, $y(nT) = 0$ for $n < 0$, and p is a real constant. (*b*) Find the unit-step response of the system.

■ **Solution**

(*a*) From Example 4.5(*a*), the system is characterized by the difference equation

$$y(nT) = x(nT) + py(nT - T) \tag{4.16}$$

With $x(nT) = \delta(nT)$, we can write

$$\begin{aligned} y(0) &= 1 + py(-T) = 1 \\ y(T) &= 0 + py(0) = p \\ y(2T) &= 0 + py(T) = p^2 \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ y(nT) &= p^n \end{aligned}$$

and since $y(nT) = 0$ for $n \leq 0$, we have

$$y(nT) = u(nT)p^n \quad \blacksquare \tag{4.17}$$

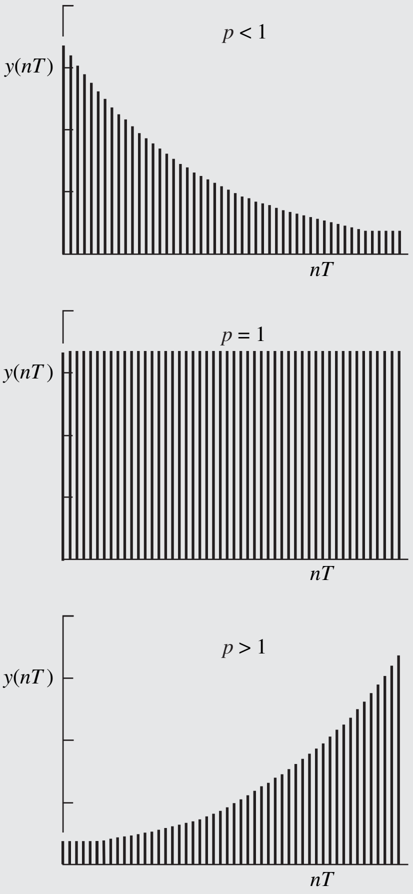


Figure 4.10 Impulse response of first-order system (Example 4.8(*a*)).

The impulse response is plotted in Fig. 4.10 for $p < 1$, $p = 1$, and $p > 1$. We note that the impulse response diverges if $p > 1$.
(*b*) With $x(nT) = u(nT)$, we get

$$\begin{aligned} y(0) &= 1 + py(-T) = 1 \\ y(T) &= 1 + py(0) = 1 + p \\ y(2T) &= 1 + py(T) = 1 + p + p^2 \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ y(nT) &= u(nT)\sum_{k=0}^{n} p^k \end{aligned}$$

This is a geometric series with common ratio p and hence we can write

$$y(nT) - py(nT) = u(nT)(1 - p^{(n+1)})$$

or

$$y(nT) = u(nT)\frac{1 - p^{(n+1)}}{1 - p} \quad \blacksquare \tag{4.18}$$

For $p < 1$, $\lim_{n\to\infty} p^{(n+1)} \to 0$ and hence the steady-state value of the response is obtained as

$$\lim_{n\to\infty} y(nT) = \frac{1}{1 - p}$$

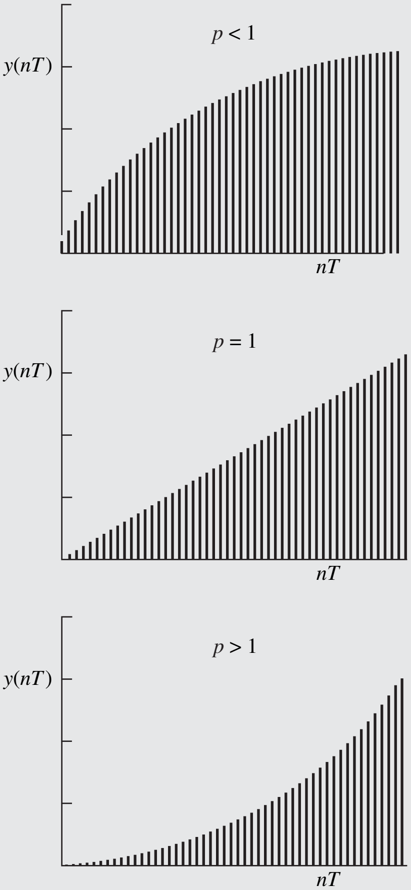


Figure 4.11 Unit-step response of first-order system (Example 4.8(*b*)).

For $p = 1$, Eq. (4.18) gives $y(nT) = 0/0$ but if we apply l'Hôpital's rule, we obtain

$$y(nT) = \lim_{p \to 1} \frac{d(1 - p^{(n+1)})/dp}{d(1 - p)/dp} = n + 1$$

Thus $y(nT) \to \infty$ as $n \to \infty$.

For $p > 1$, Eq. (4.18) gives

$$\lim_{n \to \infty} y(nT) \approx \frac{p^n}{p - 1} \to \infty$$

The unit-step response for the three values of p is illustrated in Fig. 4.11. Evidently, the response converges if $p < 1$ and diverges if $p \geq 1$.

Example 4.9 (*a*) Find the response of the system in Fig. 4.5*a* to the exponential excitation

$$x(nT) = u(nT)e^{j\omega nT}$$

(*b*) Repeat part (*a*) for the sinusoidal excitation

$$x(nT) = u(nT) \sin \omega nT$$

(*c*) Assuming that $p < 1$, find the response of the system to the sinusoidal excitation in part (*b*) as $n \to \infty$.

■ **Solution**

(*a*) With the system initially relaxed, the use of Eq. (4.16) gives

$$\begin{aligned} y(0) &= e^0 + py(-T) = 1 \\ y(T) &= e^{j\omega T} + py(0) = e^{j\omega T} + p \\ y(2T) &= e^{j2\omega T} + py(T) = e^{j2\omega T} + pe^{j\omega T} + p^2 \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \end{aligned}$$

$$\begin{aligned} y(nT) &= u(nT)(e^{j\omega nT} + pe^{j\omega(n-1)T} + \cdots + p^{(n-1)}e^{j\omega T} + p^n) \\ &= u(nT)e^{j\omega nT}(1 + pe^{-j\omega T} + \cdots + p^n e^{-jn\omega T)}) \\ &= u(nT)e^{j\omega nT} \sum_{k=0}^{n} p^k e^{-jk\omega T} \end{aligned}$$

This is a geometric series with a common ratio $pe^{-j\omega T}$ and, as in Example 4.8(b), the above sum can be obtained in closed form as

$$y(nT) = u(nT)\frac{e^{j\omega nT} - p^{(n+1)}e^{-j\omega T}}{1 - pe^{-j\omega T}} \tag{4.19}$$

Now consider the function

$$H(e^{j\omega T}) = \frac{1}{1 - pe^{-j\omega T}} = \frac{e^{j\omega T}}{e^{j\omega T} - p} \tag{4.20}$$

and let

$$H(e^{j\omega T}) = M(\omega)e^{j\theta(\omega)} \tag{4.21}$$

where

$$M(\omega) = |H(e^{j\omega T})| = \frac{1}{\sqrt{1 + p^2 - 2p\cos\omega T}} \tag{4.22a}$$

and

$$\theta(\omega) = \arg H(e^{j\omega T}) = \omega T - \tan^{-1}\frac{\sin\omega T}{\cos\omega T - p} \tag{4.22b}$$

as can be easily shown. On using Eqs. (4.19)–(4.21), $y(nT)$ can be expressed as

$$\begin{aligned} y(nT) &= u(nT)H(e^{j\omega T})(e^{j\omega nT} - p^{(n+1)}e^{-j\omega T}) \\ &= u(nT)M(\omega)(e^{j[\theta(\omega)+\omega nT]} - p^{(n+1)}e^{j[\theta(\omega)-\omega T]}) \quad \blacksquare \end{aligned} \tag{4.23}$$

(b) The system is linear and so

$$\begin{aligned} y(nT) &= \mathcal{R}u(nT)\sin\omega nT = \mathcal{R}u(nT)\frac{1}{2j}(e^{j\omega nT} - e^{-j\omega nT}) \\ &= \frac{1}{2j}[\mathcal{R}u(nT)e^{j\omega nT} - \mathcal{R}u(nT)e^{-j\omega nT}] \\ &= \frac{1}{2j}[y_1(nT) - y_2(nT)] \end{aligned} \tag{4.24}$$

where

$$y_1(nT) = \mathcal{R}u(nT)e^{j\omega nT} \qquad \text{and} \qquad y_2(nT) = \mathcal{R}u(nT)e^{-j\omega nT}$$

Partial response $y_1(nT)$ can be immediately obtained from Eq. (4.23) in part (a) as

$$y_1(nT) = u(nT)M(\omega)(e^{j[\theta(\omega)+\omega nT]} - p^{(n+1)}e^{j[\theta(\omega)-\omega T]}) \tag{4.25}$$

and since

$$y_2(nT) = \mathcal{R}u(nT)e^{j\omega nT}\Big|_{\omega\to-\omega} \tag{4.26}$$

partial response $y_2(nT)$ can be obtained by replacing ω by $-\omega$ in $y_1(nT)$, that is,

$$y_2(nT) = u(nT)M(-\omega)(e^{j[\theta(-\omega)-\omega nT]} - p^{(n+1)}e^{j[\theta(-\omega)+\omega T]}) \tag{4.27}$$

From Eqs. (4.22a) and (4.22b), we note that $M(\omega)$ is an even function and $\theta(\omega)$ is an odd function of ω, that is,

$$M(-\omega) = M(\omega) \qquad \text{and} \qquad \theta(-\omega) = -\theta(\omega)$$

Hence Eqs. (4.24), (4.25), and (4.27) yield

$$\begin{aligned} y(nT) &= u(nT)\frac{M(\omega)}{2j}(e^{j[\theta(\omega)+\omega nT]} - e^{-j[\theta(\omega)+\omega nT]}) \\ &\quad -u(nT)\frac{M(\omega)}{2j}p^{(n+1)}(e^{j[\theta(\omega)-\omega T]} - e^{-j[\theta(\omega)-\omega T]}) \\ &= u(nT)M(\omega)\sin[\omega nT + \theta(\omega)] \\ &\quad -u(nT)M(\omega)p^{(n+1)}\sin[\theta(\omega) - \omega T] \quad \blacksquare \end{aligned} \tag{4.28}$$

We note that the response of the system consists of two components. For a given frequency ω, the first term is a steady sinusoid of fixed amplitude and the second term is a transient component whose amplitude is proportional to $p^{(n+1)}$.

(c) If $p < 1$, then the transient term in Eq. (4.28) reduces to zero as $n \to \infty$ since $\lim_{n\to\infty} p^{(n+1)} \to 0$ and, therefore, we have

$$\tilde{y}(nT) = \lim_{n\to\infty} y(nT) = M(\omega)\sin[\omega nT + \theta(\omega)] \quad \blacksquare$$

This is called the *steady-state sinusoidal response* and, as can be seen, it is a sinusoid of amplitude $M(\omega)$ displaced by a phase angle $\theta(\omega)$. Since the input is a sinusoid whose amplitude and phase angle are unity and zero, respectively, the system has introduced a gain $M(\omega)$ and a phase shift $\theta(\omega)$, as illustrated in Fig. 4.12.

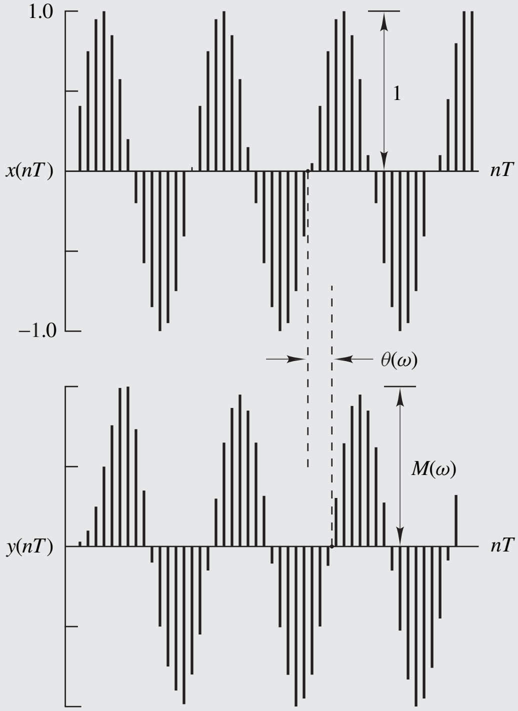


Figure 4.12 Steady-state sinusoidal response of first-order system (Example 4.9*b*).

The sinusoidal response in the above example turned out to comprise a steady-state and a transient component. This is a property of discrete-time systems in general, as will be demonstrated in Chap. 5. Functions $M(\omega)$ and $\theta(\omega)$, which will resurface in Sec. 5.5.1, facilitate one to find the steady-state sinusoidal response of a system for any specified frequency ω and by virtue of linearity one can also find the response produced by a signal that comprises an arbitrary linear combination of sinusoids of different frequencies. Obviously, these are very useful functions and are called the *amplitude response* and *phase response* of the system, respectively.

If p were greater than unity in the above example, then the amplitude of the transient component would, in principle, increase indefinitely since $\lim_{n\to\infty} p^{(n+1)} \to \infty$ in such a case. Lack of convergence in the time-domain response is undesirable in practice and when it can occur for at least one excitation, the system is said to be *unstable*. On the basis of the results obtained in Examples 4.8 and 4.9, the system in Fig. 4.5a is unstable if $p \geq 1$ since the unit-step response does not converge for $p \geq 1$. The system appears to be stable if $p < 1$ since the impulse, unit-step, and sinusoidal responses converge for this case but, at this point, we cannot be certain whether an excitation exists that would produce an unbounded time-domain response. The circumstances and conditions that must be satisfied to assure the stability of a discrete-time system will be examined in Sec. 4.7.

The above time-domain analysis method can be easily extended to higher-order systems. Consider, for example, the general system represented by Eq. (4.6). Assuming that $N \geq M$, then through some simple algebra one can express Eq. (4.6) in the form

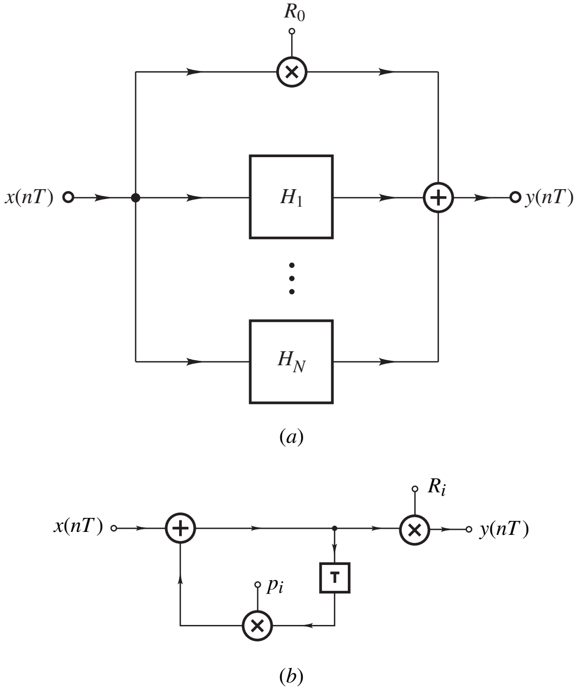
$$y(nT) = \left(R_0 + \sum_{i=1}^{N} \frac{R_i}{1 + p_i \mathcal{E}^{-1}} \right) x(nT) \tag{4.29}$$

where p_i and R_i are constants, possibly complex. This equation characterizes the equivalent parallel configuration of Fig. 4.13*a* where H_i is a first-order system characterized by

$$y(nT) = \frac{R_i}{1 + p_i \mathcal{E}^{-1}} x(nT)$$

The time-domain response of this first-order system can be obtained as in Examples 4.8 and 4.9 and the response of the multiplier in Fig. 4.13*a* is simply $R_0 x(nT)$. Therefore, by virtue of linearity, the response of the parallel configuration, and thus that of the original high-order system, can be deduced. For example, the impulse response of the first-order system in Fig. 4.13*b* can be obtained as

$$y(nT) = u(nT) R_i p_i^n \tag{4.30}$$



(*a*)

(*b*)

Figure 4.13 Representation of a high-order system in terms of a set of first-order systems.

as in Example 4.8(a) and the impulse response of the multiplier is $R_0\delta(nT)$. Thus Eqs. (4.29) and (4.30) give the impulse response of an N-order recursive system as

$$\begin{aligned} y(nT) &= \left(R_0 + \sum_{i=1}^{N} \frac{R_i}{1+p_i\mathcal{E}^{-1}}\right) x(nT) \\ &= R_0 x(nT) + \sum_{i=1}^{N} \frac{R_i}{1+p_i\mathcal{E}^{-1}} x(nT) \\ &= R_0\delta(nT) + u(nT)\sum_{i=1}^{N} R_i p_i^n \end{aligned}$$

The unit-step or sinusoidal response of the system can similarly be deduced.

Unfortunately, the induction method just described can easily run into serious complications and lacks both generality and potential. An alternative approach that overcomes some of these difficulties is the state-space method described in Sec. 4.8. The most frequently used method for time-domain analysis, however, involves the use of the z transform and is described in detail in Chap. 5.

4.6 CONVOLUTION SUMMATION

The response of a discrete-time system to an arbitrary excitation can be expressed in terms of the impulse response of the system.

An excitation $x(nT)$ can be expressed as a sum of signals as

$$x(nT) = \sum_{k=-\infty}^{\infty} x_k(nT) \tag{4.31}$$

where each signal $x_k(nT)$ has just one nonzero value equal to the value of $x(nT)$ at $n = k$, that is,

$$x_k(nT) = \begin{cases} x(kT) & \text{for } n = k \\ 0 & \text{otherwise} \end{cases}$$

as illustrated in Fig. 4.14. Each of the signals $x_k(nT)$ is actually an impulse signal and can be represented as

$$x_k(nT) = x(kT)\delta(nT - kT) \tag{4.32}$$

and hence Eqs. (4.31) and (4.32) give

$$x(nT) = \sum_{k=-\infty}^{\infty} x(kT)\delta(nT - kT) \tag{4.33}$$

Now consider a system characterized by the equation

$$y(nT) = \mathcal{R}x(nT) \tag{4.34}$$

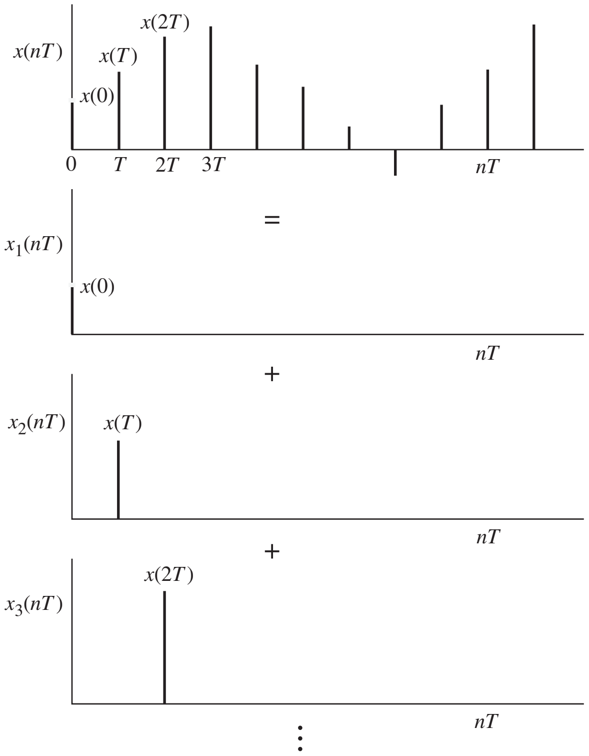


Figure 4.14 Convolution summation: decomposition of a discrete-time signal into a sum of impulses.

and let

$$h(nT) = \mathcal{R}\delta(nT) \tag{4.35}$$

be the impulse response of the system. Assuming that the system is linear and time invariant, Eqs. (4.33)–(4.35) give

$$\begin{aligned} y(nT) &= \mathcal{R}\sum_{k=-\infty}^{\infty} x(kT)\delta(nT-kT) = \sum_{k=-\infty}^{\infty} x(kT)\mathcal{R}\delta(nT-kT) \\ &= \sum_{k=-\infty}^{\infty} x(kT)h(nT-kT) \end{aligned} \tag{4.36a}$$

This relation is of considerable importance in the characterization as well as analysis of discrete-time systems and is known as the *convolution summation*.

Some special forms of the convolution summation are of particular interest. To start with, by letting $n' = n - k$ in Eq. (4.36a) and noting that the limits of the summation do not change, the alternative but equivalent form

$$y(nT) = \sum_{k=-\infty}^{\infty} h(kT)x(nT - kT) \tag{4.36b}$$

can be obtained. If the system is causal, $h(nT) = 0$ for $n < 0$ and thus Eqs. (4.36a) and (4.36b) give

$$y(nT) = \sum_{k=-\infty}^{n} x(kT)h(nT - kT) = \sum_{k=0}^{\infty} h(kT)x(nT - kT) \tag{4.36c}$$

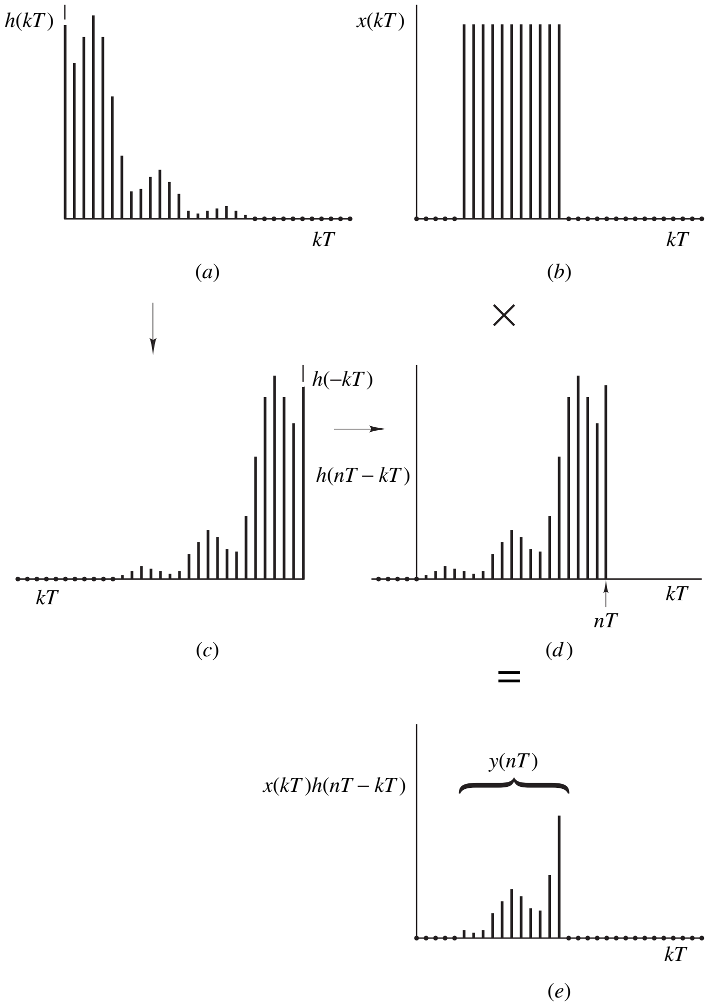


Figure 4.15 Convolution summation: graphical representation.

and if, in addition, $x(nT) = 0$ for $n < 0$, then

$$y(nT) = \sum_{k=0}^{n} x(kT)h(nT - kT) = \sum_{k=0}^{n} h(kT)x(nT - kT) \tag{4.36d}$$

The convolution summation plays a critical role in the application of the z transform to discrete-time systems, as will be demonstrated in Chap. 5, and the assumptions made here in deriving the convolution summation, namely, that the system is linear and time invariant, become inherited assumptions for the applicability of the z transform to discrete-time systems.

4.6.1 Graphical Interpretation

The fist convolution summation in Eq. (4.36d) is illustrated in Fig. 4.15. The impulse response $h(kT)$ is folded over with respect to the y axis, as in Fig. 4.15*c*, and is then shifted to the right by a time interval nT, as in Fig. 4.15*d*, to yield $h(nT - kT)$. Then $x(kT)$ is multiplied by $h(nT - kT)$, as in Fig. 4.15*e*. The sum of all values in Fig. 4.15*e* is the response of the system at instant nT.

Example 4.10 (*a*) Using the convolution summation, find the unit-step response of the system in Fig. 4.5*a*. (*b*) Hence find the response to the excitation

$$x(nT) = \begin{cases} 1 & \text{for } 0 \le n \le 4 \\ 0 & \text{otherwise} \end{cases}$$

■ **Solution**

(*a*) From Example 4.8(*a*), the impulse response of the system is given by

$$h(nT) = u(nT)p^n$$

(See Eq. (4.17).) Since the unit step is zero for $n < 0$, the convolution summation in Eq. (4.36a) gives

$$y(nT) = \mathcal{R}u(nT) = \sum_{k=-\infty}^{\infty} u(kT)p^k u(nT - kT)$$

$$= \cdots + \overbrace{u(-T)p^{-1}u(nT + T)}^{k=-1} + \overbrace{u(0)p^0 u(nT)}^{k=0} + \overbrace{u(T)p^1 u(nT - T)}^{k=1}$$

$$+ \cdots + \overbrace{u(nT)p^n u(0)}^{k=n} + \overbrace{u(nT + T)p^{n+1} u(-T)}^{k=n+1} + \cdots$$

For $n < 0$, we get $y(nT) = 0$ since all the terms are zero. For $n \geq 0$, we obtain

$$y(nT) = 1 + p^1 + p^2 + \cdots + p^n = 1 + \sum_{n=1}^{n} p^n$$

This is a geometric series and has a sum

$$S = \frac{1 - p^{(n+1)}}{1 - p}$$

(see Eq. (A.46b)). Hence, the response can be expressed in closed form as

$$y(nT) = u(nT)\frac{1 - p^{(n+1)}}{1 - p} \quad \blacksquare$$

(b) For this part, we observe that

$$x(nT) = u(nT) - u(nT - 5T) \tag{4.37}$$

and so

$$y(nT) = \mathcal{R}x(nT) = \mathcal{R}u(nT) - \mathcal{R}u(nT - 5T) \tag{4.38}$$

Thus

$$y(nT) = u(nT)\frac{1 - p^{(n+1)}}{1 - p} - u(nT - 5T)\frac{1 - p^{(n-4)}}{1 - p}$$

Alternatively, we can write

$$y(nT) = \begin{cases} u(nT)\dfrac{1 - p^{(n+1)}}{1 - p} & \text{for } n \leq 4 \\ \dfrac{p^{(n-4)} - p^{(n+1)}}{1 - p} & \text{for } n > 4 \end{cases} \quad \blacksquare$$

Example 4.11 An initially relaxed causal nonrecursive system was tested with an input

$$x(nT) = \begin{cases} 0 & \text{for } n < 0 \\ n & \text{for } n \geq 0 \end{cases}$$

and found to have the response given by the following table:

n	0	1	2	3	4	5	6	7
$y(nT)$	0	1	4	10	20	30	40	50

(*a*) Find the impulse response of the system for values of n over the range $0 \le n \le 5$.

(*b*) Using the result in part (a), find the unit-step response for $0 \le n \le 5$.

■ **Solution**

(a) Problems of this type can be easily solved by using the convolution summation. Since the system is causal and $x(nT) = 0$ for $n < 0$, the left-hand convolution summation in Eq. (4.36d) applies and hence

$$y(nT) = \mathcal{R}x(nT) = \sum_{k=0}^{n} x(kT)h(nT - kT)$$

or

$$y(nT) = x(0)h(nT) + x(T)h(nT - T) + \cdots + h(0)x(nT)$$

Evaluating $y(nT)$ for $n = 1,\ 2, \ldots,$ we get

$$y(T) = x(0)h(T) + x(T)h(0) = 0 \cdot h(T) + 1 \cdot h(0) = 1 \qquad \text{or} \qquad h(0) = 1$$

$$\begin{aligned} y(2T) &= x(0)h(2T) + x(T)h(T) + x(2T)h(0) \\ &= 0 \cdot h(2T) + 1 \cdot h(T) + 2 \cdot h(0) \\ &= 0 + h(T) + 2 = 4 \quad \text{or} \quad h(T) = 2 \end{aligned}$$

$$\begin{aligned} y(3T) &= x(0)h(3T) + x(T)h(2T) + x(2T)h(T) + x(3T)h(0) \\ &= 0 \cdot h(3T) + 1 \cdot h(2T) + 2 \cdot h(T) + 3 \cdot h(0) \\ &= h(2T) + 2 \cdot 2 + 3 \cdot 1 = 10 \quad \text{or} \quad h(2T) = 3 \end{aligned}$$

$$\begin{aligned} y(4T) &= x(0)h(4T) + x(T)h(3T) + x(2T)h(2T) + x(3T)h(T) + x(4T)h(0) \\ &= 0 \cdot h(4T) + 1 \cdot h(3T) + 2 \cdot h(2T) + 3 \cdot h(T) + 4 \cdot h(0) \\ &= h(3T) + 2 \cdot 3 + 3 \cdot 2 + 4 \cdot 1 = 20 \quad \text{or} \quad h(3T) = 4 \end{aligned}$$

$$\begin{aligned} y(5T) &= x(0)h(5T) + x(T)h(4T) + x(2T)h(3T) + x(3T)h(2T) + x(4T)h(T) \\ &\quad + x(5T)h(0) \\ &= 0 \cdot h(5T) + 1 \cdot h(4T) + 2 \cdot h(3T) + 3 \cdot h(2T) + 4 \cdot h(T) + 5 \cdot h(0) \\ &= 0 + h(4T) + 2 \cdot 4 + 3 \cdot 3 + 4 \cdot 2 + 5 \cdot 1 = 30 \end{aligned}$$

or $$h(4T) = 0$$

$$\begin{aligned} y(6T) &= x(0)h(6T) + x(T)h(5T) + x(2T)h(4T) + x(3T)h(3T) + x(4T)h(2T) \\ &\qquad + x(5T)h(T) + x(6T)h(0) \\ &= 0\cdot h(6T) + 1\cdot h(5T) + 2\cdot h(4T) + 3\cdot h(3T) + 4\cdot h(2T) + 5\cdot h(T) + 6\cdot h(0) \\ &= h(5T) + 2\cdot 0 + 3\cdot 4 + 4\cdot 3 + 5\cdot 2 + 6\cdot 1 \\ &= 40 \qquad \text{or} \qquad h(5T) = 0 \end{aligned}$$

Thus

$$h(0) = 1 \qquad h(T) = 2 \qquad h(2T) = 3 \qquad h(3T) = 4$$
$$h(4T) = 0 \qquad h(5T) = 0 \quad \blacksquare$$

(b) Using the convolution summation again, we obtain the unit-step response as follows:

$$y(nT) = \mathcal{R}x(nT) = \sum_{k=0}^{n} u(kT)h(nT - kT) = \sum_{k=0}^{n} h(nT - kT)$$

Hence

$$\begin{aligned} y(0) &= h(0) = 1 \\ y(T) &= h(T) + h(0) = 2 + 1 = 3 \\ y(2T) &= h(2T) + h(T) + h(0) = 3 + 2 + 1 = 6 \\ y(3T) &= h(3T) + h(2T) + h(T) + h(0) = 10 \\ y(4T) &= h(4T) + h(3T) + h(2T) + h(T) + h(0) = 15 \\ y(5T) &= h(5T) + h(4T) + h(3T) + h(2T) + h(T) + h(0) = 21 \end{aligned}$$

Thus

$$y(0) = 1 \qquad y(T) = 3 \qquad y(2T) = 6 \qquad y(3T) = 10$$
$$y(4T) = 15 \qquad y(5T) = 21 \quad \blacksquare$$

4.6.2 Alternative Classification

Discrete-time systems may also be classified on the basis of the duration of their impulse response either as *finite-duration impulse response* (FIR) systems or as *infinite-duration impulse response* (IIR) systems.[2]

[2] *Actually, the acronyms for these systems should be FDIR and IDIR filters since it is the duration that is infinite and not the response. However, the acronyms FIR and IIR are too entrenched to be changed.*

If the impulse response of a causal discrete-time system is of finite duration such that $h(nT) = 0$ for $n < -K$ and $n > M$, then the convolution summation in Eq. (4.36b) gives

$$y(nT) = \sum_{k=-K}^{M} h(kT)x(nT - kT)$$

This equation is of the same form as Eq. (4.4) with $a_{-K} = h(-K),\ a_{-K+1} = h(-K+1), \ldots, a_M = h(MT)$ and, in effect, such a system is nonrecursive. Conversely, if a nonrecursive system is characterized by Eq. (4.4), then its impulse response can be readily shown to be $h(-K) = a_{-K},\ h(-K+1) = a_{-K+1}, \ldots, h(MT) = a_M$ and, therefore, it is of finite duration.

In recursive systems, the impulse response is almost always of infinite duration but it can, in theory, be of finite duration, as will now be demonstrated. Consider a nonrecursive system characterized by the difference equation

$$y(nT) = x(nT) + 3x(nT - T) \tag{4.39a}$$

The impulse response of the system is obviously of finite duration since $h(0) = 1$, $h(T) = 3$, and $h(kT) = 0$ for all values of $k \neq 0$ or 1. If we premultiply both sides of Eq. (4.39a) by the operator $(1 + 4\mathcal{E}^{-1})$, we get

$$(1 + 4\mathcal{E}^{-1})y(nT) = (1 + 4\mathcal{E}^{-1})[x(nT) + 3x(nT - T)] \tag{4.39b}$$

and after simplification, we have

$$y(nT) = x(nT) + 7x(nT - T) + 12x(nT - 2T) - 4y(nT - T)$$

Thus, an FIR system can be represented by a recursive difference equation! Evidently, the manipulation has increased the order of the difference equation from one to two but the system response will not change in any way. Under these circumstances, there is no particular reason in applying such a manipulation. In fact, there is every reason to identify common factors and cancel them out since they tend to increase the order of the difference equation and, in turn, the complexity of the system. On the other hand, an IIR system cannot be nonrecursive and vice versa, as depicted in Fig. 4.16.

An arbitrary recursive system can be represented by an equation of the form

$$y(nT) = \frac{N(\mathcal{E}^{-1})}{D(\mathcal{E}^{-1})}x(nT)$$

and if operator polynomials $N(\mathcal{E}^{-1})$ and $D(\mathcal{E}^{-1})$ are free from common factors, then the recursive system is also an IIR system. Since common factors are a form of redundancy that must be removed, then, for all practical purposes, the terms recursive and IIR are exchangeable and so are the terms nonrecursive and FIR.

In this book, we shall be referring to systems as nonrecursive or recursive if the emphasis is on the difference equation or network and as FIR or IIR if the emphasis is on the duration of the impulse response.

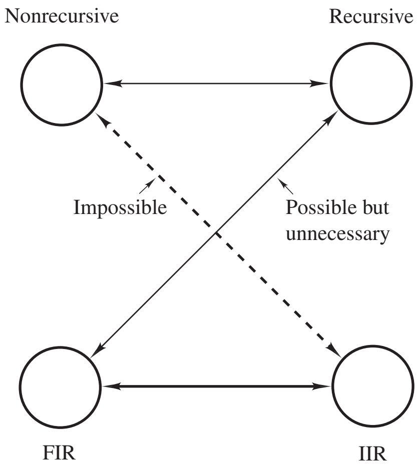


Figure 4.16 Nonrecursive versus FIR and recursive versus IIR systems.

4.7 STABILITY

A continuous- or discrete-time system is said to be *stable* if and only if any bounded excitation will result in a bounded response. In mathematical language, a discrete-time system is stable if and only if any input $x(nT)$ such that

$$|x(nT)| \leq P < \infty \qquad \text{for all } n \tag{4.40}$$

will produce an output $y(nT)$ that satisfies the condition

$$|y(nT)| \leq Q < \infty \qquad \text{for all } n \tag{4.41}$$

where P and Q are positive constants.

For a linear and time-invariant system, the convolution summation in Eq. (4.36b) gives

$$\begin{aligned}|y(nT)| &= \left|\sum_{k=-\infty}^{\infty} h(kT)x(nT-kT)\right|\\ &\leq \sum_{k=-\infty}^{\infty} |h(kT)x(nT-kT)|\\ &\leq \sum_{k=-\infty}^{\infty} |h(kT)|\cdot|x(nT-kT)|\end{aligned} \tag{4.42}$$

The equal sign in the first equation is replaced by the less than or equal sign in the second and third equations since some of the terms under the sum may be negative and, consequently, the magnitude of the sum may be smaller than the sum of the magnitudes, for example, $|2\cdot 2 + 3\cdot 3 + 7\cdot(-1)| < |2\cdot 2| + |3\cdot 3| + |7\cdot(-1)| = |2|\cdot|2| + |3|\cdot|3| + |7|\cdot|(-1)|$. The equal sign is retained to take care of the rare possibility where the terms are all positive.

If the input satisfies the condition in Eq. (4.42), then if we replace $|x(nT - kT)|$ by its largest possible value as specified in Eq. (4.40), we obtain

$$\begin{aligned} |y(nT)| &\le \sum_{k=-\infty}^{\infty} |h(kT)|P \\ &\le P \sum_{k=-\infty}^{\infty} |h(kT)| \end{aligned} \tag{4.43}$$

Now if the impulse response is absolutely summable, that is,

$$\sum_{k=-\infty}^{\infty} |h(kT)| \le R < \infty \tag{4.44}$$

then Eqs. (4.43) and (4.44) give

$$|y(nT)| \le Q < \infty \qquad \text{for all } n$$

where $Q = PR$. Therefore, Eq. (4.44) constitutes a sufficient condition for stability.

A system can be classified as stable only if its response is bounded for all possible bounded excitations. Consider the bounded excitation

$$x(nT - kT) = \begin{cases} P & \text{if } h(kT) \ge 0 \\ -P & \text{if } h(kT) < 0 \end{cases} \tag{4.45}$$

where P is a positive constant. From Eq. (4.36b)

$$\begin{aligned} |y(nT)| &= \left| \sum_{k=-\infty}^{\infty} h(kT)x(nT - kT) \right| \\ &= \sum_{k=-\infty}^{\infty} P|h(kT)| = P \sum_{k=-\infty}^{\infty} |h(kT)| \end{aligned}$$

since the product $h(kT)x(nT - kT)$ is always positive in this case by virtue of the definition of $x(nT - kT)$ in Eq. (4.45). Therefore, the condition in Eq. (4.41) will be satisfied if and only if the impulse response is absolutely summable and, therefore, Eq. (4.44) constitutes both a necessary and a sufficient condition for stability. Under these circumstances, the system is said to be *bounded-input, bounded-output* (or BIBO) stable.

Note that although stability is a crucial requirement for most systems, it should be mentioned here that there are certain inherently unstable systems which can be useful contrary to popular belief. Consider, for example, a continuous-time integrator which is a system that would integrate an input waveform. The response of such a system to a unit-step input would increase with time and would become unbounded as $t \to \infty$ since the area under the unit-step over an infinite period is infinite. Integrators would be classified as unstable in the above definition, yet they are useful in a number of DSP applications.[3] It should be mentioned, however, that such systems are problematic in practice because the level of their internal signals can easily become large enough to cause them to operate outside their linear range. Discrete-time systems also exist that are inherently unstable but which can be useful, for example, discrete-time integrators. These are systems that can perform numerical integration.

In nonrecursive systems, the impulse response is of finite duration and hence Eq. (4.44) is always satisfied. Consequently, these systems are always stable. This is a great advantage in certain applications, for example, in adaptive filters which are filters that change their characteristics on line. Recursive adaptive filters, would need certain recovery mechanisms to prevent them from becoming unstable.

The stability of a system can be checked by establishing whether the impulse response satisfies Eq. (4.44). This boils down to checking whether the series is absolutely convergent, and a number of tests are available at our disposal for this purpose such as the ratio test (see Theorem A.3 in Sec. A.5).

Example 4.12 (*a*) Check the system of Fig. 4.5*a* for stability. (*b*) A discrete-time system has an impulse response

$$h(nT) = u(nT)e^{0.1nT}\sin\frac{n\pi}{6}$$

Check the stability of the system.

■ **Solution**

(*a*) The impulse response of the system was obtained in Example 4.8(*a*) and is given by

$$h(nT) = u(nT)p^n$$

(See Eq. (4.17)). Hence

$$\sum_{k=-\infty}^{\infty} |h(kT)| = 1 + |p| + \cdots + |p^k| + \cdots \tag{4.46}$$

This is a geometric series and has a sum

$$\sum_{k=-\infty}^{\infty} |h(kT)| = \lim_{n\to\infty} \frac{1 - |p|^{(n+1)}}{1 - |p|}$$

[3] *They used to build analog computers with them during the 1950s and 1960s.*

(see Eq. (A.46b)). If $p > 1$,

$$\sum_{k=-\infty}^{\infty} |h(kT)| = \lim_{n\to\infty} \frac{1-|p|^{(n+1)}}{1-|p|} \to \infty$$

and if $p = 1$,

$$\sum_{k=-\infty}^{\infty} |h(kT)| = 1+1+1+\cdots = \infty$$

On the other hand, if $p < 1$,

$$\sum_{k=-\infty}^{\infty} |h(kT)| = \lim_{n\to\infty} \frac{1-|p|^{(n+1)}}{1-|p|} \to \frac{1}{1-|p|} = K < \infty$$

where K is a positive constant. Therefore, the system in is stable if and only if

$$|p| < 1 \quad \blacksquare$$

This result explains why the unit-step and sinusoidal responses of the system obtained in Examples 4.8(*b*) and 4.9(*b*) are bounded if $|p| < 1$ and unbounded if $|p| \geq 1$.
(*b*) We can write

$$\begin{aligned}\sum_{k=0}^{\infty} |h(nT)| &= \sum_{k=0}^{\infty} \left| u(kT)e^{0.1kT} \sin \frac{k\pi}{6} \right| \\ &= \sum_{k=3,9,15,\ldots}^{\infty} \left| u(kT)e^{0.1kT} \sin \frac{k\pi}{6} \right| + \sum_{k\neq 3,9,15,\ldots}^{\infty} \left| u(kT)e^{0.1kT} \sin \frac{k\pi}{6} \right| \\ &= \sum_{k=3,9,15,\ldots}^{\infty} \left| e^{0.1kT} \right| + \sum_{k\neq 3,9,15,\ldots}^{\infty} \left| e^{0.1kT} \sin \frac{k\pi}{6} \right| \to \infty\end{aligned}$$

that is, the impulse response is not absolutely summable and, therefore, the system is unstable. ■

4.8 STATE-SPACE REPRESENTATION

Systems in general can be represented in terms of equations that involve vectors and matrices. Characterizations of this type are usually based on so-called *state-space methods* and, as will be demonstrated in due course, they have certain inherent advantages. Below we develop a state-space representation for discrete-time systems.

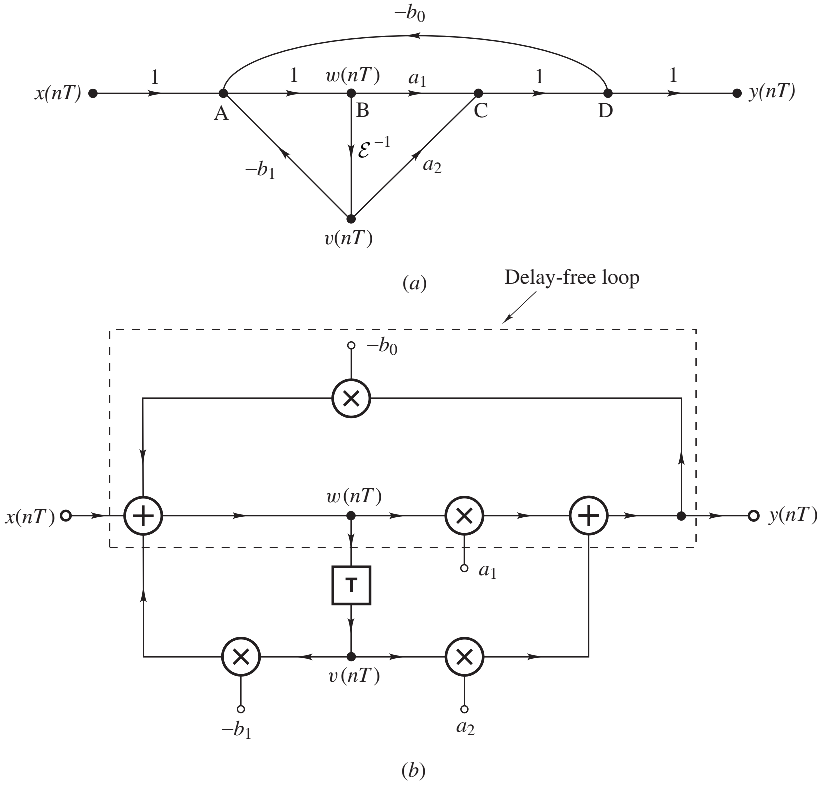
4.8.1 Computability

The state-space representation to be described is based on Mason's gain formula (see Sec. 4.4.3) but before we proceed with the representation itself, a prerequisite property of signal flow graphs known as *computability* needs to be explained.

Earlier on, we have shown that given a discrete-time network, a corresponding signal flow graph can be obtained. This process can obviously be reversed and given a signal flow graph, a discrete-time network can be derived from it. Consider the signal flow graph of Fig. 4.17*a* and its equivalent network shown in Fig. 4.17*b*. We observe that the signal flow graph has a loop that does not include the shift operator $\mathcal{E}^{-1}$, namely, loop ABCDA. In the network, this loop corresponds to the delay-free loop shown enclosed in dashed lines. If a signal $x(nT)$ is applied at the input of the network, the sum

$$w(nT) = x(nT) - b_1 v(nT) - b_0 y(nT)$$

will appear at the output of the left-hand adder in τ_A s, where τ_A is the propagation delay of the adder; the product $a_1 w(nT)$ will then be formed in τ_M s, where τ_M is the propagation delay of the



(*a*)

(*b*)

Figure 4.17 (*a*) Noncomputable signal flow graph, (*b*) equivalent network.

multiplier. Signals $a_1 w(nT)$ and $a_2 v(nT)$ will then be added in τ_A s to produce a new output

$$y'(nT) = a_1 w(nT) + a_2 v(nT)$$

which will be fed back to the left-hand adder; this will produce the signal

$$\begin{aligned} w'(nT) &= x(nT) - b_1 v(nT) - b_0 y'(nT) \\ &= x(nT) - b_1 v(nT) - b_0[a_1 w(nT) + a_2 v(nT)] \\ &= x(nT) - [b_1 + b_0 a_2] v(nT) - a_1 b_0 w(nT) \end{aligned}$$

at the output of the left-hand adder in τ_A s. This chain of events will continue indefinitely and the end result would be a runaway situation with the output of the network assuming some unpredictable value. Even though the signal flow graph may represent a valid set of algebraic equations, the presence of a delay-free loop renders the flow graph nonimplementable basically because a variable depends on itself, namely, $w(nT)$, in the above example. Obviously, flow graphs such as the above cannot be implemented in terms of practical networks and are, therefore, of no practical interest. They are said to be *noncomputable*.

If a unit delay were inserted in the feedback branch in Fig. 4.17*b*, then signals $w(nT)$ and $v(nT)$ would propagate as far as the input of that unit delay and in the absence of a clock pulse, action would stop, the signals would reach steady state, and the runaway situation would not take place.

The above problem cannot occur if each and every loop in a signal flow graph contains at least one shift operator $\mathcal{E}^{-1}$ in which case the network would have no delay-free loops. Such a signal flow graph is said to be *computable* for obvious reasons.

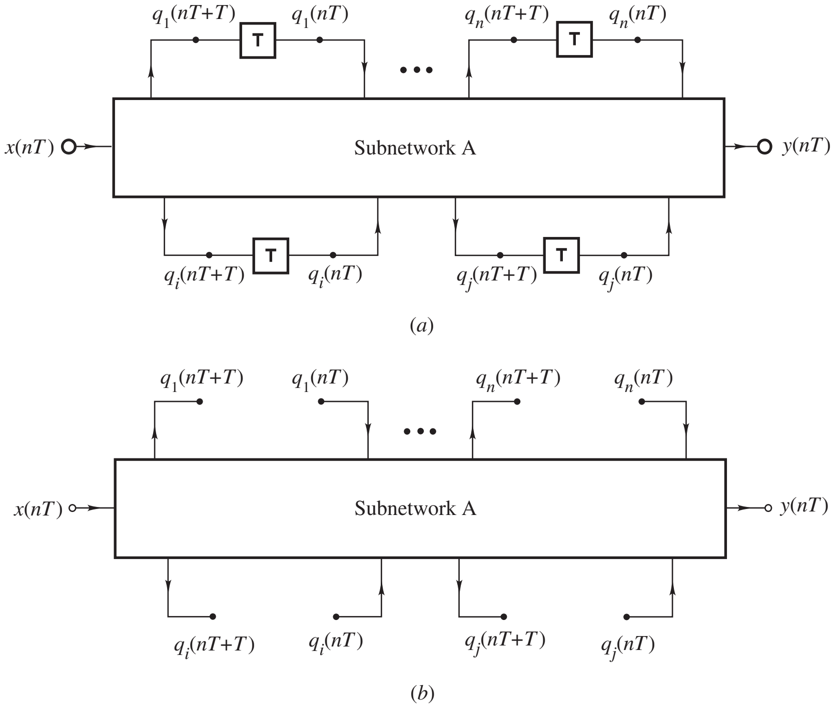
4.8.2 Characterization

Now consider an arbitrary discrete-time network containing N unit delays and assume that its signal flow graph is computable, i.e., it has no delay-free loops. The network can be drawn as shown in Fig. 4.18*a* where Subnetwork A consists exclusively of interconnected adders and multipliers. Subnetwork A cannot contain any loops since each loop of the network is completed by a unit delay which is not part of Subnetwork A, i.e., all loops would be broken if the unit delays in Fig. 4.18*a* were to be removed as depicted in Fig. 4.18*b*.

Now let $q_i(nT)$, for $i = 1, 2, \ldots, N$, be variables at the outputs of unit delays. These are stored quantities and can thus be referred to as *state variables*. The signals at the inputs of the unit delays can obviously be represented by corresponding variables $q_i(nT + T)$.

If all the state variables are assumed to be zero and the input $x(nT)$ is assumed to be nonzero, then from Mason's gain formula given in Eq. (4.15), the response at the input of the ith unit delay can be obtained as

$$q_i(nT + T) = \left(\frac{1}{\Delta} \sum_k T_k \Delta_k \right) x(nT)$$



(*a*)

(*b*)

Figure 4.18 Arbitrary discrete-time network.

where T_k is the transmittance of the kth direct path between input and node i, Δ is the determinant of the flow graph, and Δ_k is the determinant of the subgraph that does not touch the kth direct path between input and node i (see Sec. 4.4.3). Since there are no complete loops in the graph of Subnetwork A, we have

$$\Delta = \Delta_k = 1$$

Furthermore, since there are no unit delays in Subnetwork A, the transmittances T_k are independent of the shift operator $\mathcal{E}^{-1}$ and we conclude that

$$q_i(nT + T) = b_i x(nT) \tag{4.47}$$

for $i = 1, 2, \ldots, N$ where $b_1, b_2, \ldots, b_N$ are constants independent of nT for a time-invariant discrete-time system.

Similarly, if input $x(nT)$ and all the state variables except the jth state variable are assumed to be zero, we have

$$q_i(nT + T) = a_{ij} q_j(nT) \tag{4.48}$$

for $i = 1, 2, \ldots, N$ where $a_{1j}, a_{2j}, \ldots, a_{Nj}$ are again constants independent of nT for a time-invariant discrete-time system.

Now if the system is linear, then from the principle of superposition the response at the input of the ith unit delay is the sum of all the contributions originating from all the state variables plus the contribution from the input of the system. Hence, from Eqs. (4.47) and (4.48), we obtain

$$q_i(nT + T) = \sum_{j=1}^{N} a_{ij} q_j(nT) + b_i x(nT) \tag{4.49}$$

for $i = 1, 2, \ldots, N$. Similarly, the response of the system, $y(nT)$, is made of contributions from all the state variables plus the input of the system, and for the same reasons as before, we have

$$y(nT) = \sum_{j=1}^{N} c_j q_j(nT) + d_0 x(nT) \tag{4.50}$$

Therefore, from Eqs. (4.49) and (4.50), the discrete-time system can be characterized by the system of equations

$$\mathbf{q}(nT + T) = \mathbf{A}\mathbf{q}(nT) + \mathbf{b}x(nT) \tag{4.51a}$$

$$y(nT) = \mathbf{c}^T \mathbf{q}(nT) + dx(nT) \tag{4.51b}$$

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}$$

$$\mathbf{c}^T = \begin{bmatrix} c_1 \ c_2 \ \cdots \ c_N \end{bmatrix} \qquad d = d_0$$

and

$$\mathbf{q}(nT) = [q_1(nT) \ q_2(nT) \ \cdots \ q_N(nT)]^T \tag{4.52}$$

is a column vector whose elements are the state variables of the discrete-time network. This is referred to as a *state-space characterization*.

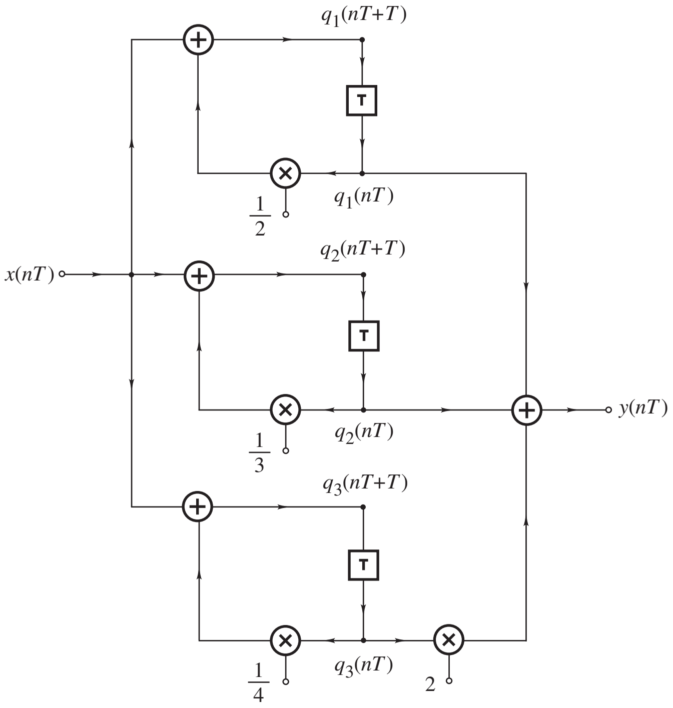


Figure 4.19 Parallel discrete-time system (Example 4.13).

Example 4.13 Obtain a state-space representation for the parallel system shown in Fig. 4.19.

■ **Solution**

State variables can be assigned to the system as shown in Fig. 4.19. By inspection, we obtain

$$q_1(nT+T) = \tfrac{1}{2}q_1(nT) + x(nT)$$
$$q_2(nT+T) = \tfrac{1}{3}q_2(nT) + x(nT)$$
$$q_3(nT+T) = \tfrac{1}{4}q_3(nT) + x(nT)$$

and

$$y(n) = q_1(nT) + q_2(nT) + 2q_3(nT)$$

Therefore,

$$\mathbf{q}(nT + T) = \mathbf{A}\mathbf{q}(nT) + \mathbf{b}x(nT)$$
$$y(n) = \mathbf{c}^T\mathbf{q}(nT) + dx(nT)$$

where $\mathbf{q}(nT)$ is the state-variable vector and

$$\mathbf{A} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \mathbf{c} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \qquad d = 0 \quad \blacksquare$$

In a state-space representation, state-variables can be assigned to nodes arbitrarily, e.g., in Example 4.13, $q_1(nT)$ could be assigned to the output of any one of the three unit delays, $q_2(nT)$ to the output of any one of the two remaining unit delays, and $q_3(nT)$ to the output of the remaining unit delay. In each possible assignment, a different representation would be obtained and, in effect, the state-space approach does not yield a unique representation.

A state-space characterization can also be obtained from a signal flow graph as illustrated by the following example.

Example 4.14 Obtain a state-space characterization for the system of Fig. 4.6.

■ **Solution**

State variables can be assigned to the flow graph of the system as depicted in Fig. 4.20, where

$$q_1(nT + T) = q_2(nT)$$
$$q_2(nT + T) = q_3(nT)$$
$$q_3(nT + T) = -b_3q_1(nT) - b_2q_2(nT) - b_1q_3(nT) + x(nT)$$

The output of the system can be expressed as

$$y(nT) = a_3q_1(nT) + a_2q_2(nT) + a_1q_3(nT) + a_0q_3(nT + T)$$

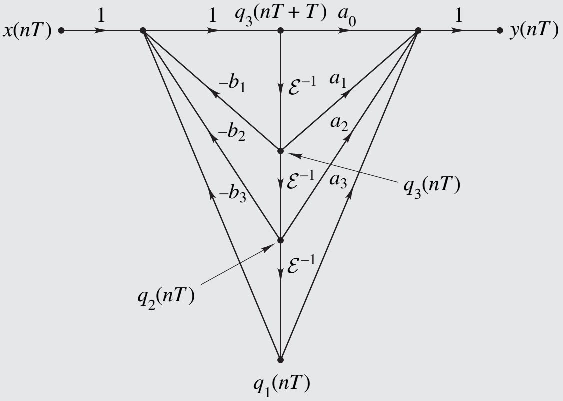


Figure 4.20 Assignment of state variables for the discrete-time system of Fig. 4.6.

and on eliminating $q_3(nT+T)$, we obtain

$$y(nT) = (a_3 - a_0b_3)q_1(nT) + (a_2 - a_0b_2)q_2(nT) + (a_1 - a_0b_1)q_3(nT) + a_0x(nT)$$

Hence, the system can be represented by

$$\begin{aligned}\mathbf{q}(nT+T) &= \mathbf{A}\mathbf{q}(nT) + \mathbf{b}x(nT)\\ y(nT) &= \mathbf{c}^T\mathbf{q}(nT) + dx(nT)\end{aligned} \quad ■$$

where

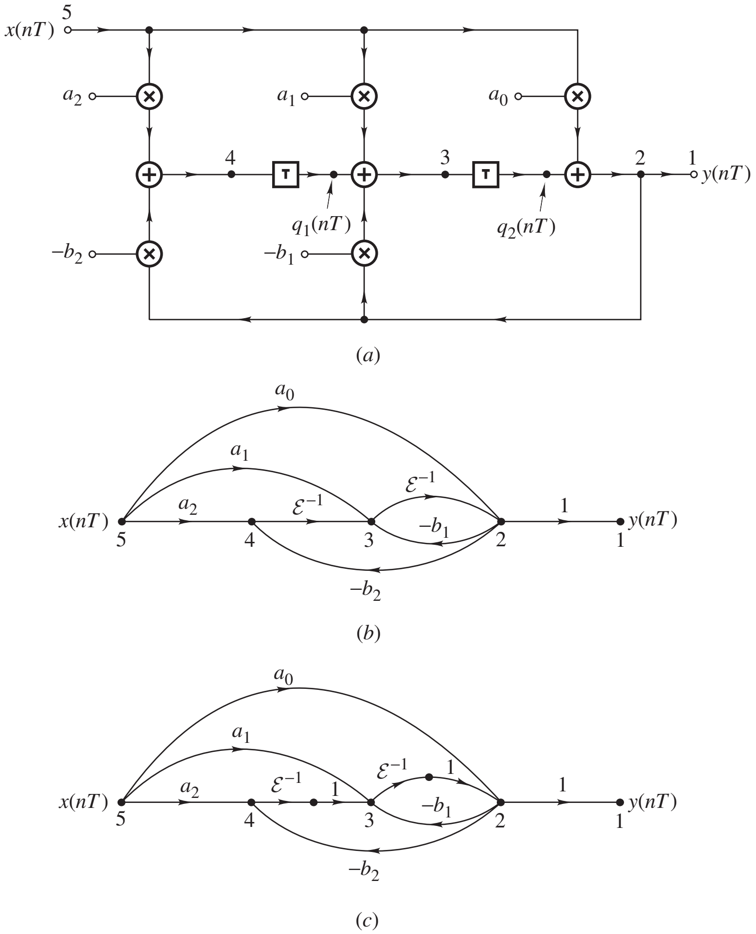
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0\\ 0 & 0 & 1\\ -b_3 & -b_2 & -b_1\end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$\mathbf{c}^T = [(a_3 - a_0b_3)\ (a_2 - a_0b_2)\ (a_1 - a_0b_1)] \qquad d = a_0$$

and

$$\mathbf{q}(nT) = [q_1(nT)\ q_2(nT)\ q_3(nT)]^T \tag{4.53}$$

Note that state variables must always be assigned at the outputs of unit delays but if a signal flow-graph representation is used, it is sometimes possible to inadvertently assign one or more state variables to nodes that are not outputs of unit delays. Consider, for example, the discrete-time network of Fig. 4.21*a* which can be represented by the signal flow graph of Fig. 4.21*b*. Assigning state-variables to nodes 2 and 3 will lead to erroneous results simply because these nodes represent the outputs of the adders, *not* of the unit delays.The problem can be avoided by working directly with the network or by inserting new nodes at the outputs of the unit delays as depicted in Fig. 4.21*c*.



(*a*)

(*b*)

(*c*)

Figure 4.21 Assignment of state variables.

Example 4.15 A system can be represented by the state-space equations in Eqs. (4.51a) and (4.51b) with

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{3} & -\frac{1}{4} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{c}^T = \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} & \frac{1}{12} \end{bmatrix} \qquad d = 2$$

Deduce the difference equation of the system.

■ **Solution**

From Eqs. (4.51a) and (4.51b), the state-space equations can be written out as

$$q_1(nT+T) = -\tfrac{1}{2}q_1(nT) - \tfrac{1}{3}q_2(nT) - \tfrac{1}{4}q_3(nT) + 2x(nT) \tag{4.54a}$$

$$q_2(nT+T) = q_1(nT) \tag{4.54b}$$

$$q_3(nT+T) = q_2(nT) \tag{4.54c}$$

and

$$y(nT) = -\tfrac{1}{4}q_1(nT) + \tfrac{1}{6}q_2(nT) + \tfrac{1}{12}q_3(nT) + 2x(nT) \tag{4.55}$$

From Eqs. (4.54b) and (4.54c), we can write

$$\mathcal{E}q_2(nT) = q_1(nT) \qquad \text{or} \qquad q_2(nT) = \mathcal{E}^{-1}q_1(nT) \tag{4.56a}$$

$$\begin{aligned}\mathcal{E}q_3(nT) = q_2(nT) \qquad &\text{or} \qquad q_3(nT) = \mathcal{E}^{-1}q_2(nT)\\ = \mathcal{E}^{-2}q_1(nT)&\end{aligned} \tag{4.56b}$$

Now if we eliminate $q_2(nT)$ and $q_3(nT)$ in Eq. (4.54a), we get

$$\mathcal{E}q_1(nT) = -\tfrac{1}{2}q_1(nT) - \tfrac{1}{3}\mathcal{E}^{-1}q_1(nT) - \tfrac{1}{4}\mathcal{E}^{-2}q_1(nT) + 2x(nT)$$

and hence

$$\left[\mathcal{E} + \tfrac{1}{2} + \tfrac{1}{3}\mathcal{E}^{-1} + \tfrac{1}{4}\mathcal{E}^{-2}\right] q_1(nT) = 2x(nT)$$

or

$$q_1(nT) = \frac{2}{\mathcal{E} + \frac{1}{2} + \frac{1}{3}\mathcal{E}^{-1} + \frac{1}{4}\mathcal{E}^{-2}}x(nT) \tag{4.57}$$

Similarly, if we eliminate $q_2(nT)$ and $q_3(nT)$ in Eq. (4.55), we get

$$\begin{aligned} y(nT) &= -\tfrac{1}{4}q_1(nT) + \tfrac{1}{6}\mathcal{E}^{-1}q_1(nT) + \tfrac{1}{12}\mathcal{E}^{-2}q_1(nT) + 2x(nT)\\ &= \left[-\tfrac{1}{4} + \tfrac{1}{6}\mathcal{E}^{-1} + \tfrac{1}{12}\mathcal{E}^{-2}\right] q_1(nT) + 2x(nT) \end{aligned} \tag{4.58}$$

If we now eliminate $q_1(nT)$ in Eq. (4.58) using Eq. (4.57), we have

$$\begin{aligned} y(nT) &= \frac{2\left[-\frac{1}{4} + \frac{1}{6}\mathcal{E}^{-1} + \frac{1}{12}\mathcal{E}^{-2}\right]}{\mathcal{E} + \frac{1}{2} + \frac{1}{3}\mathcal{E}^{-1} + \frac{1}{4}\mathcal{E}^{-2}}x(nT) + 2x(nT)\\ &= \frac{\left[2 + \frac{1}{2}\mathcal{E}^{-1} + \mathcal{E}^{-2} + \frac{2}{3}\mathcal{E}^{-3}\right]}{1 + \frac{1}{2}\mathcal{E}^{-1} + \frac{1}{3}\mathcal{E}^{-2} + \frac{1}{4}\mathcal{E}^{-3}}x(nT) \end{aligned}$$

Alternatively, we can write

$$\left[1+\tfrac{1}{2}\mathcal{E}^{-1}+\tfrac{1}{3}\mathcal{E}^{-2}+\tfrac{1}{4}\mathcal{E}^{-3}\right]y(nT)=\left[2+\tfrac{1}{2}\mathcal{E}^{-1}+\mathcal{E}^{-2}+\tfrac{2}{3}\mathcal{E}^{-3}\right]x(nT)$$

or

$$\begin{aligned} y(nT) = 2x(nT) + \tfrac{1}{2}x(nT-T) + x(nT-2T) + \tfrac{2}{3}x(nT-3T)\\ -\tfrac{1}{2}y(nT-T) - \tfrac{1}{3}y(nT-2T) - \tfrac{1}{4}y(nT-3T) \quad \blacksquare \end{aligned}$$

4.8.3 Time-Domain Analysis

The preceding state-space characterization leads directly to a relatively simple *time-domain* analysis.

For $n = 0, 1, \ldots,$ Eq. (4.51a) gives

$$\begin{aligned} \mathbf{q}(T) &= \mathbf{Aq}(0) + \mathbf{b}x(0)\\ \mathbf{q}(2T) &= \mathbf{Aq}(T) + \mathbf{b}x(T)\\ \mathbf{q}(3T) &= \mathbf{Aq}(2T) + \mathbf{b}x(2T)\\ &\cdots\cdots\cdots\cdots\cdots \end{aligned}$$

Hence

$$\begin{aligned} \mathbf{q}(2T) &= \mathbf{A}^2\mathbf{q}(0) + \mathbf{Ab}x(0) + \mathbf{b}x(T)\\ \mathbf{q}(3T) &= \mathbf{A}^3\mathbf{q}(0) + \mathbf{A}^2\mathbf{b}x(0) + \mathbf{Ab}x(T) + \mathbf{b}x(2T) \end{aligned}$$

and in general

$$\mathbf{q}(nT) = \mathbf{A}^n\mathbf{q}(0) + \sum_{k=0}^{n-1}\mathbf{A}^{(n-1-k)}\mathbf{b}x(kT)$$

where $\mathbf{A}^0$ is the $N \times N$ unity matrix. Therefore, from Eq. (4.51b), we obtain

$$y(nT) = \mathbf{c}^T\mathbf{A}^n\mathbf{q}(0) + \mathbf{c}^T\sum_{k=0}^{n-1}\mathbf{A}^{(n-1-k)}\mathbf{b}x(kT) + dx(nT)$$

If $x(nT) = 0$ for all $n < 0$, then for $n = -1$ Eq. (4.51a) yields

$$\mathbf{q}(0) = b_i x(-T) = 0$$

Thus for an initially relaxed system

$$y(nT) = \mathbf{c}^T\sum_{k=0}^{n-1}\mathbf{A}^{(n-1-k)}\mathbf{b}x(kT) + dx(nT)$$

The *impulse* response $h(nT)$ of the system is

$$h(nT) = \mathbf{c}^T \sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)} \mathbf{b}\delta(kT) + d\delta(nT)$$

For $n = 0$

$$h(0) = d\delta(nT) = d_0$$

and for $n > 0$

$$h(nT) = \mathbf{c}^T \mathbf{A}^{(n-1)} \mathbf{b}\delta(0) + \mathbf{c}^T \mathbf{A}^{(n-2)} \mathbf{b}\delta(T) + \cdots + d\delta(nT)$$

Therefore,

$$h(nT) = \begin{cases} d_0 & \text{for } n = 0 \\ \mathbf{c}^T \mathbf{A}^{(n-1)} \mathbf{b} & \text{for } n > 0 \end{cases} \tag{4.59}$$

Similarly, the *unit-step* response of the system is

$$y(nT) = \mathbf{c}^T \sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)} \mathbf{b} u(kT) + du(nT)$$

Hence, for $n \geq 0$

$$y(nT) = \mathbf{c}^T \sum_{k=0}^{n-1} \mathbf{A}^{(n-1-k)} \mathbf{b} + d \tag{4.60}$$

Example 4.16 An initially relaxed discrete-time system can be represented by the matrices

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ \frac{1}{4} & -\frac{1}{2} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \mathbf{c}^T = \begin{bmatrix} \frac{7}{8} & \frac{5}{4} \end{bmatrix} \qquad d = \frac{3}{2}$$

Find $h(17T)$.

■ **Solution**

From Eq. (4.59), we immediately get

$$h(17T) = \mathbf{c}^T \mathbf{A}^{16} \mathbf{b}$$

By forming the matrices $\mathbf{A}^2$, $\mathbf{A}^4$, $\mathbf{A}^8$, and then $\mathbf{A}^{16}$ through matrix multiplication, we get

$$h(17T) = \begin{bmatrix} \frac{7}{8} & \frac{5}{4} \end{bmatrix} \begin{bmatrix} \frac{610}{65,536} & -\frac{987}{32,768} \\ -\frac{987}{131,072} & \frac{1597}{65,536} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1076}{262,144}$$

4.8.4 Applications of State-Space Method

The state-space method offers the advantage that systems can be analyzed through the manipulation of matrices which can be carried out very efficiently using array or vector processors. Another important advantage of this method is that it can be used to characterize and analyze time-dependent systems, that is, systems in which one or more of the elements of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}^T$ and possibly constant d depend on nT. This advantage follows from the fact that only linearity is a prerequisite property for the derivation of the state-state representation.

Time-varying systems like adaptive filters are now used quite extensively in a variety of communications applications. The state-space method can also be used to realize digital filters that have certain important advantages, e.g., increased signal-to-noise ratio (see Sec. 14.7).

A negative aspect associated with state-space time-domain analysis is the fact that the solutions are not in closed form in general.

REFERENCES

[1] R. J. Schwarz and B. Friedland, *Linear Systems*, McGraw-Hill, New York, 1965.

[2] R. Butler and E. Kerr, *An Introduction to Numerical Methods*, Pitman, London, 1962.

[3] J. R. Abrahams and G. P. Coverley, *Signal Flow Analysis*, Pergamon, New York, 1965.

[4] B. C. Kuo, *Automatic Control Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1962.

[5] N. Balabanian and T. A. Bickart, *Electrical Network Theory*, Wiley, New York, 1969.

[6] S. J. Mason, "Feedback theory—Further properties of signal-flow graphs," *Proc. IRE*, Vol. 44, pp. 920–926, July 1956.

PROBLEMS

4.1. By using appropriate tests, check the systems characterized by the following equations for linearity, time invariance, and causality:

(*a*) $y(nT) = \mathcal{R}x(nT) = 1.25 + 2.5x(nT) + 5.0(nT + 2T)x(nT - T)$

(*b*) $y(nT) = \mathcal{R}x(nT) = \begin{cases} 6x(nT - 5T) & \text{for } x(nT) \le 6 \\ 7x(nT - 5T) & \text{for } x(nT) > 6 \end{cases}$

(*c*) $y(nT) = \mathcal{R}x(nT) = (nT + 3T)x(nT - 3T)$

4.2. Repeat Prob. 4.1 for the systems characterized by the following equations:

(*a*) $y(nT) = \mathcal{R}x(nT) = 5nTx^2(nT)$

(*b*) $y(nT) = \mathcal{R}x(nT) = 3x(nT + 3T)$

(*c*) $y(nT) = \mathcal{R}x(nT) = x(nT)\sin \omega nT$

4.3. Repeat Prob. 4.1 for the systems characterized by the following equations:

(*a*) $y(nT) = \mathcal{R}x(nT) = nT + K_1\Delta x(nT)$ where $\Delta x(nT) = x(nT + T) - x(nT)$

(*b*) $y(nT) = \mathcal{R}x(nT) = 1 + K_2\nabla x(nT)$ where $\nabla x(nT) = x(nT) - x(nT - T)$

(*c*) $y(nT) = \mathcal{R}x(nT) = x(nT + T)e^{-nT}$

4.4. Repeat Prob. 1.1 for the systems characterized by the following equations:

(*a*) $y(nT) = \mathcal{R}x(nT) = x^2(nT + T)e^{-nT}\sin \omega nT$

(*b*) $y(nT) = \mathcal{R}x(nT) = \frac{1}{3}e^{-0.01nT}\sum_{i=-1}^{1} x(nT - iT)$

(*c*) $y(nT + T) = \mathcal{R}x(nT) = \Delta x(nT) - \nabla x(nT)$

4.5. (*a*) Obtain the difference equation of the discrete-time network shown in Fig. P4.5*a*.

(*b*) Repeat part(*a*) for the network of Fig. P4.5*b*.

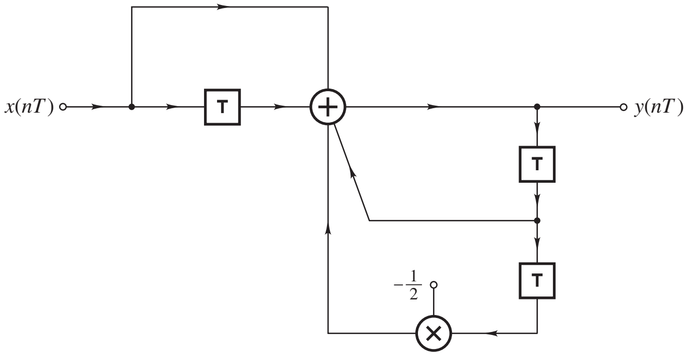


Figure P4.5*a*

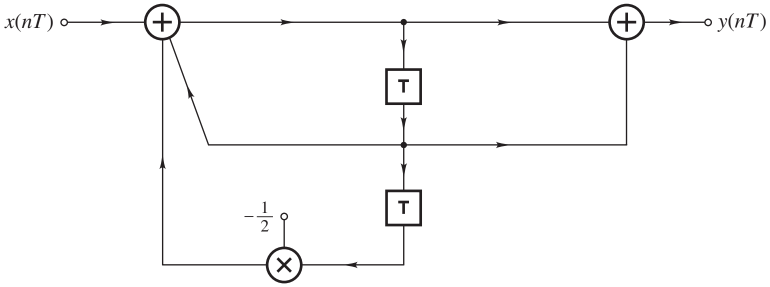


Figure P4.5*b*

4.6. (*a*) Obtain the difference equation of the network shown in Fig. P4.6*a*.

(*b*) Repeat part(*a*) for the network of Fig. P4.6*b*.

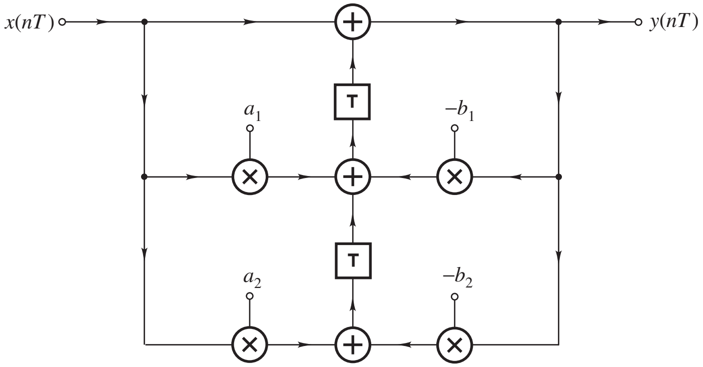


Figure P4.6*a*

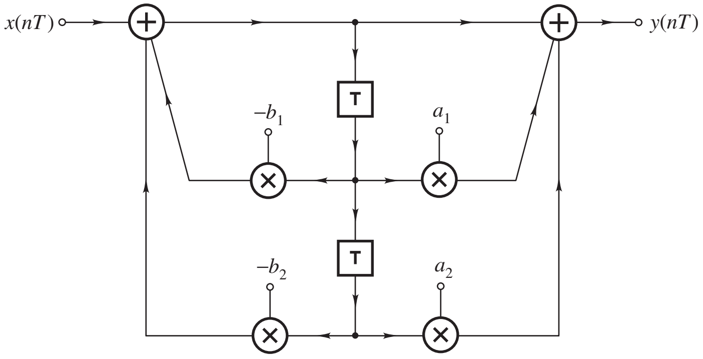


Figure P4.6*b*

4.7. Two second-order system sections of the type shown in Fig. P4.6*a* are connected in cascade as in Fig. P4.7. The parameters of the two sections are a_{11}, a_{21}, $-b_{11}$, $-b_{21}$ and a_{12}, a_{22}, $-b_{12}$, $-b_{22}$, respectively. Deduce the characterization of the combined system.

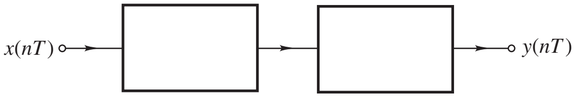


Figure P4.7

4.8. Two second-order systems of the type shown in Fig. P4.6*a* are connected in parallel as in Fig. P4.8. Obtain the difference equation of the combined system.

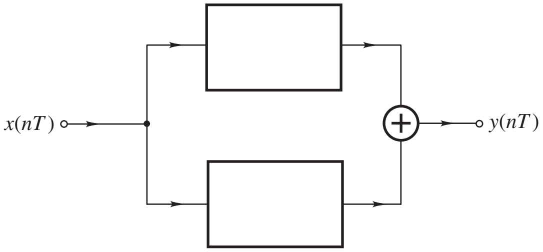


Figure P4.8

4.9. Fig. P4.9 shows a network with three inputs and three outputs.

(*a*) Derive a set of equations characterizing the network.

(*b*) Express the equations obtained in part (*a*) in the form

$$\mathbf{y} = \mathbf{M}\mathbf{x}$$

where **y** and **x** are column vectors given by $[y_1(nT)\ y_2(nT)\ y_3(nT)]^T$ and $[x_1(nT)\ x_2(nT)\ x_3(nT)]^T$, respectively, and **M** is a 3×3 matrix.

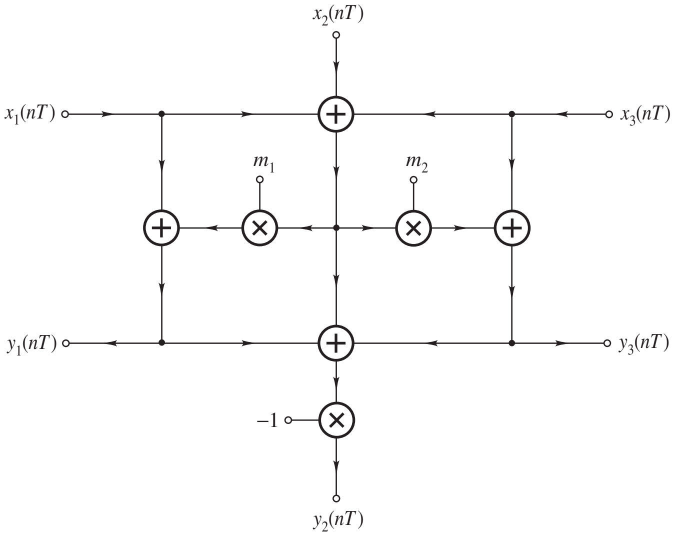


Figure P4.9

4.10. The network of Fig. P4.10 can be characterized by the equation

$$\mathbf{b} = \mathbf{Ca}$$

where $\mathbf{b} = [b_1\ b_2\ b_3]^T$ and $\mathbf{a} = [a_1\ a_2\ a_3]^T$ are column vectors and $\mathbf{C}$ is a 3×3 matrix. Obtain $\mathbf{C}$.

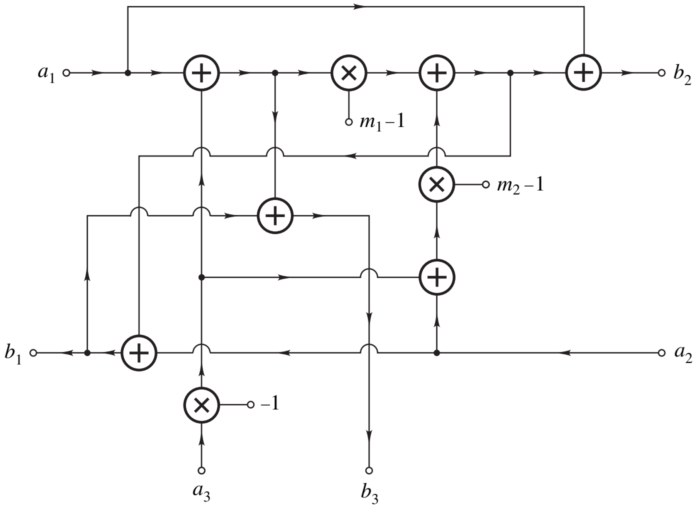


Figure P4.10

4.11. By using appropriate tests, check the systems of Fig. P4.11*a* to *c* for linearity, time invariance, and causality.

(*a*) The system of Fig. P4.11*a* uses a device N whose response is given by

$$\mathcal{R}x(nT) = |x(nT)|$$

(*b*) The system of Fig. P4.11*b* uses a multiplier M whose parameter is given by

$$m = 0.1x(nT)$$

(*c*) The system of Fig. P4.11*c* uses a multiplier M whose parameter is given by

$$m = 0.1v(nT)$$

where $v(nT)$ is an independent control signal.

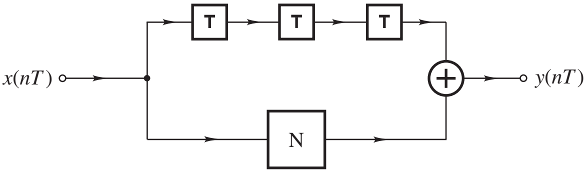


Figure P4.11*a*

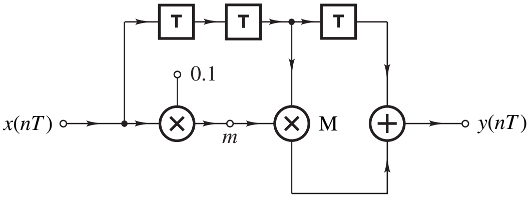


Figure P4.11*b*

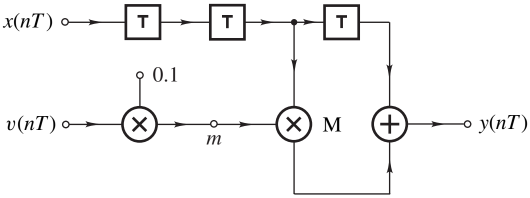


Figure P4.11*c*

4.12. An initially relaxed discrete-time system employs a device D, as shown in Fig. P4.12, which is characterized by the equation

$$w(nT) = 2(-1)^n|v(nT)|$$

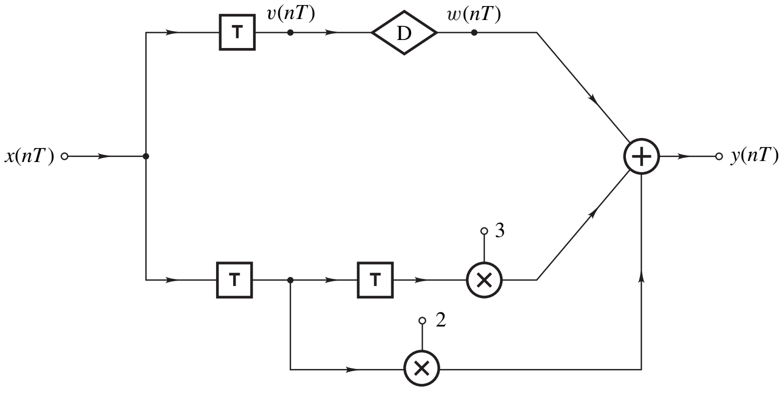


Figure P4.12

(a) Deduce the difference equation.

(b) By using appropriate tests, check the system for linearity, time invariance, and causality.

(c) Evaluate the time-domain response for the period 0 to $10T$ if the input signal is given by

$$x(nT) = u(nT) - 2u(nT - 4T)$$

where $u(nT)$ is the unit step.

(d) What is the order of the system?

4.13. The discrete-time system of Fig. P4.13 uses a device D whose response to an input $w(nT)$ is $d_0 + d_1 w(nT)$, where d_0 and d_1 are nonzero constants. By using appropriate tests, check the system for linearity, time invariance, and stability.

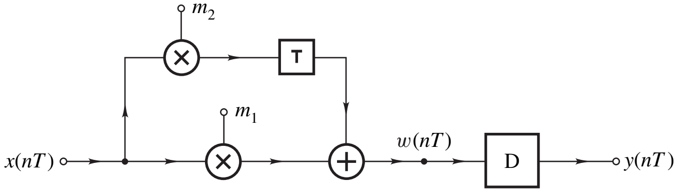


Figure P4.13

4.14. A discrete-time system is characterized by the equation

$$\begin{aligned} y(nT) &= \mathcal{R}x(nT) \\ &= a_0 x(nT) + a_1 x(nT - T) + nTx(nT)x(nT - T) + a_0 a_1 x(nT - 2T) \end{aligned}$$

(a) By using appropriate tests, check the system for linearity, time invariance, and stability.

(b) Find the unit-step response at $t = 5T$ if $a_0 = 2$, $a_1 = 3$, and $T = 1$ assuming that the system is initially relaxed.

4.15. The system of Fig. P4.15 is initially relaxed. Find the time-domain response for the period $nT = 0$ to $6T$, if

$$x(nT) = \begin{cases} \sin \omega nT & \text{for } n \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\omega = \pi/6T$ and $T = 1$.

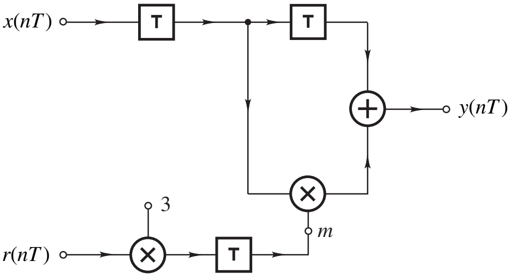


Figure P4.15

4.16. (*a*) Obtain the signal flow graph of the system shown in Fig. P4.16.

(*b*) Deduce the difference equation by using the node elimination method.

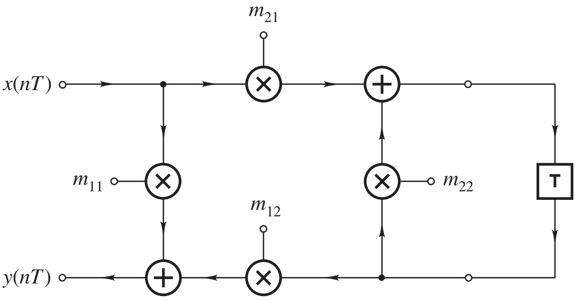


Figure P4.16

4.17. (*a*) Obtain the signal flow graph of the system shown in Fig. P4.17.

(*b*) Deduce the difference equation by using the node elimination method.

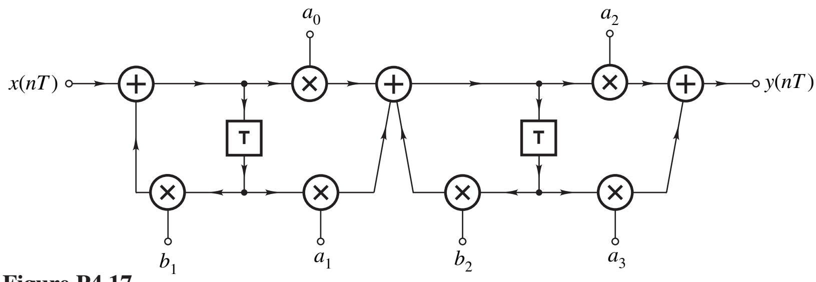


Figure P4.17

4.18. (*a*) Obtain the signal flow graph of the system shown in Fig. P4.6*a*.

(*b*) Deduce the difference equation by using the node elimination method.

4.19. Deduce the difference equation of the system shown in Fig. 4.19 by using the node elimination method.

4.20. Derive a *closed-form expression* for the response of the system in Fig. 4.5*a* to an excitation

$$x(nT) = \begin{cases} 1 & \text{for } 0 \le n \le 3 \\ 0 & \text{otherwise} \end{cases}$$

The system is initially relaxed and $p = \frac{1}{2}$.

4.21. (*a*) Show that

$$r(nT) = \begin{cases} 0 & \text{for } n \le 0 \\ T\sum_{k=1}^{n} u(nT - kT) & \text{otherwise} \end{cases}$$

(*b*) By using this relation obtain the unit-ramp response of the system shown in Fig. 4.5*a* in closed form. The system is initially relaxed.

(*c*) Sketch the response for $\alpha > 0$, $\alpha = 0$, and $\alpha < 0$.

4.22. The excitation in the first-order system of Fig. 4.5*a* is

$$x(nT) = \begin{cases} 1 & \text{for } 0 \le n \le 4 \\ 2 & \text{for } \quad n > 4 \\ 0 & \text{for } \quad n < 0 \end{cases}$$

Find the response in closed form.

4.23. Repeat Prob. 4.22 for an excitation

$$x(nT) = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{for } n < 0, n = 1, 2, 3, 4 \\ 1 & \text{for } n > 4 \end{cases}$$

4.24. Fig. P4.24 shows a second-order recursive system. Using MATLAB or similar software, compute and plot the unit-step response for $0 \le n \le 15$ if

(*a*) $\alpha = 1 \quad \beta = -\frac{1}{2}$

(*b*) $\alpha = \frac{1}{2} \quad \beta = -\frac{1}{8}$

(*c*) $\alpha = \frac{5}{4} \quad \beta = -\frac{25}{32}$

Compare the three responses and determine the frequency of the transient oscillation in terms of T where possible.

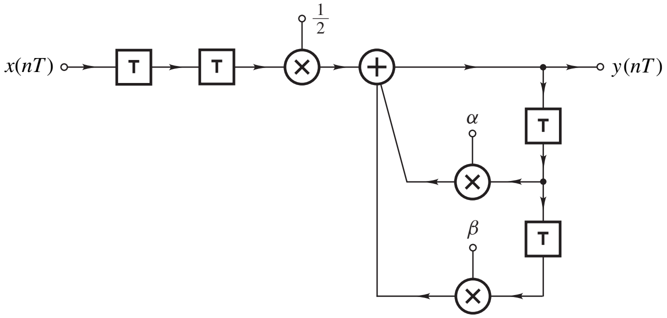


Figure P4.24

4.25. Fig. P4.25 shows a system comprising a cascade of two first-order sections. The input signal is

$$x(nT) = \begin{cases} \sin \omega nT & \text{for } n \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

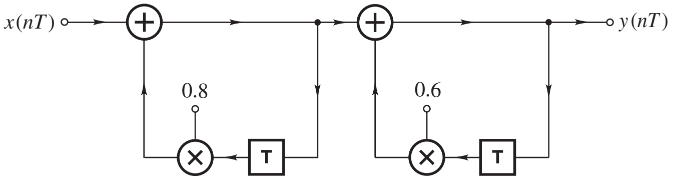


Figure P4.25

and $T = 1$ ms.

(*a*) Assuming that the two sections are linear, give an expression for the overall steady-state sinusoidal response.

(*b*) Compute the gain and phase shift of the system for a frequency $\omega = 20\pi$ rad/s. Repeat for $\omega = 200\pi$ rad/s.

4.26. Fig. P4.26 shows a linear first-order system.

(*a*) Assuming a sinusoidal excitation, derive an expression for the steady-state gain of the system.

(*b*) Using MATLAB, compute and plot the gain in decibels (dB), that is, $20 \log M(\omega)$, versus $\log \omega$ for $\omega = 0$ to 6 krad/s if $T = 1$ ms.

(*c*) Determine the lowest frequency at which the gain is reduced by 3 dB relative to the gain at zero frequency.

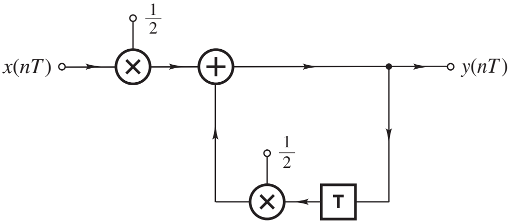


Figure P4.26

4.27. Two first-order systems of the type shown in Fig. 4.5*a* are connected in parallel as in Fig. P4.8. The multiplier constants for the two systems are $m_1 = e^{0.6}$ and $m_2 = e^{0.7}$. Find the unit-step response of the combined network in closed form.

4.28. The unit-step response of a system is

$$y(nT) = \begin{cases} nT & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases}$$

(*a*) Using the convolution summation, find the unit-ramp response.

(*b*) Check the system for stability.

4.29. A nonrecursive system has an impulse response

$$h(nT) = \begin{cases} nT & \text{for } 0 \le n \le 4 \\ (8-n)T & \text{for } 5 \le n \le 8 \\ 0 & \text{otherwise} \end{cases}$$

The sampling frequency is 2π rad/s.

(a) Deduce the network of the system.

(b) By using the convolution summation, determine the response $y(nT)$ at $nT = 4T$ if the input signal is given by

$$x(nT) = u(nT - T)e^{-nT}$$

(c) Illustrate the solution in part *(b)* by a graphical construction.

4.30. An initially relaxed nonrecursive causal system was tested with the input signal

$$x(nT) = u(nT) + u(nT - 2T)$$

and its response was found to be as shown in the following table:

n	0	1	2	3	4	5	$\cdots$	100	$\cdots$
$y(nT)$	3	5	9	11	12	12	$\cdots$	12	$\cdots$

(a) Find the impulse response for the period 0 to $5T$.

(b) Find the response for the period 0 to $5T$ if the input is changed to

$$x(nT) = u(nT) - u(nT - 2T)$$

4.31. The response of an initially relaxed fifth-order causal nonrecursive system to the excitation

$$x(nT) = u(nT)n$$

is given in the following table:

n	0	1	2	3	4	5	6	7	8	9	10	$\cdots$
$y(nT)$	0	1	3	7	14	25	41	57	73	89	105	$\cdots$

(a) Find the impulse response.

(b) Obtain the difference equation.

4.32. A discrete-time system has an impulse response

$$h(nT) = u(nT)nT$$

(a) Using the convolution summation, find the response $y(nT)$ for an excitation

$$x(nT) = u(nT)\sin 2nT$$

at $nT = 4T$. The sampling frequency is $\omega_s = 16$ rad/s.

(b) Illustrate graphically the steps involved in the solution of part (a).

4.33. An initially relaxed nonrecursive system was tested with the input signal

$$x(nT) = 2u(nT)$$

and found to have the response given in the following table:

n	0	1	2	3	4	5	$\cdots$	100	$\cdots$
$y(nT)$	2	6	12	20	30	30	$\cdots$	30	$\cdots$

(*a*) Deduce the difference equation.
(*b*) Construct a possible network for the system.

4.34. The unit-step response of an initially relaxed nonrecursive causal system is given in the following table:

n	0	1	2	3	4	5	$\cdots$
$y(nT)$	0	1	9	36	100	225	$\cdots$

(*a*) Find the impulse response for $0 \le nT \le 5T$ using the convolution summation.
(*b*) Assuming that the general pattern of the impulse response continues in subsequent values of nT, write a closed-form expression for the impulse response.
(*c*) Is the system stable or unstable? Justify your answer.

4.35. (*a*) A discrete-time system has an impulse response

$$h(nT) = u(nT - T)\frac{1}{n}$$

By using an appropriate test, check the system for stability.
(*b*) Repeat part (*a*) for the system characterized by

$$h(nT) = u(nT - T)\frac{1}{n!}$$

4.36. Check the systems represented by the following impulse responses for stability:
(*a*) $h(nT) = \dfrac{u(nT)n}{2^n}$
(*b*) $h(nT) = \dfrac{u(nT)n}{n+1}$
(*c*) $h(nT) = u(nT - T)\dfrac{(n+1)}{n^2}$

4.37. (*a*) Check the system of Fig. P4.37*a* for stability.
(*b*) Repeat part (*a*) for the system of Fig. P4.37*b*.

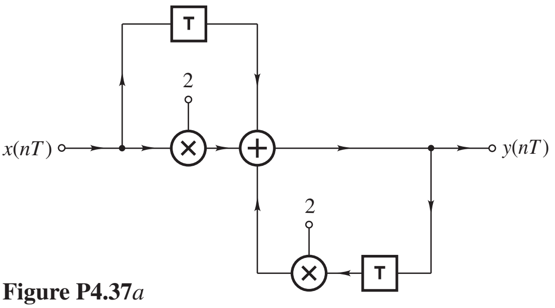


Figure P4.37*a*

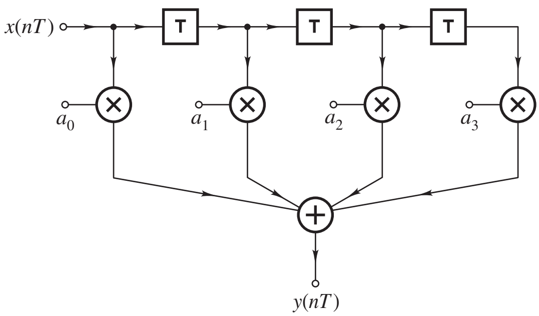


Figure P4.37*b*

4.38. (*a*) Derive a state-space representation for the system of Fig. P4.38.
(*b*) Calculate the response $y(nT)$ at $nT = 3T$ for an excitation

$$x(nT) = 2\delta(nT) + u(nT)$$

if $m_1 = \frac{1}{2}$ and $m_2 = \frac{1}{4}$.

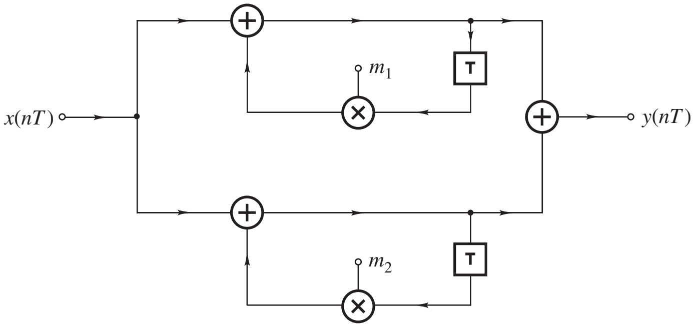


Figure P4.38

4.39. Derive a state-space representation for the system of Fig. P4.5*a*.

4.40. Derive a state-space representation for the system of Fig. P4.6*a*.

4.41. Derive a state-space representation for the system of Fig. P4.17.

4.42. Derive a state-space representation for the system of Fig. 4.21*a*.

4.43. Derive a state-space representation for the system of Fig. P4.43.

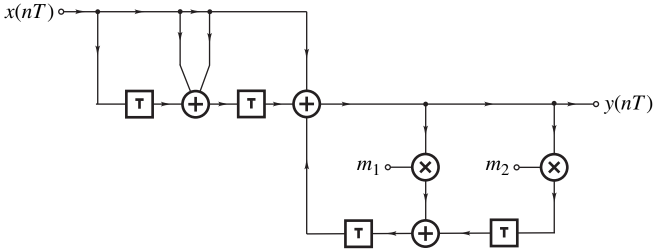


Figure P4.43

4.44. Derive a state-space representation for the system of Fig. P4.44.

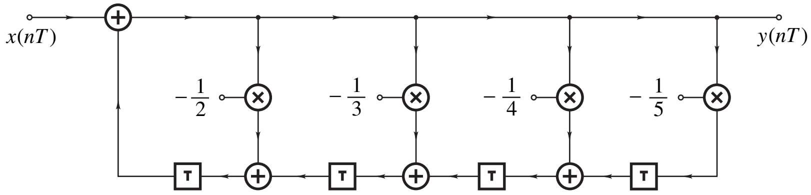


Figure P4.44

4.45. The system in Fig. 4.5*c* is initially relaxed.

(*a*) Derive a state-space representation.

(*b*) Give an expression for the response of the system at $nT = 5T$ if

$$x(nT) = u(nT)\sin\omega nT$$

4.46. Derive a state-space representation for the system of Fig. P4.46.

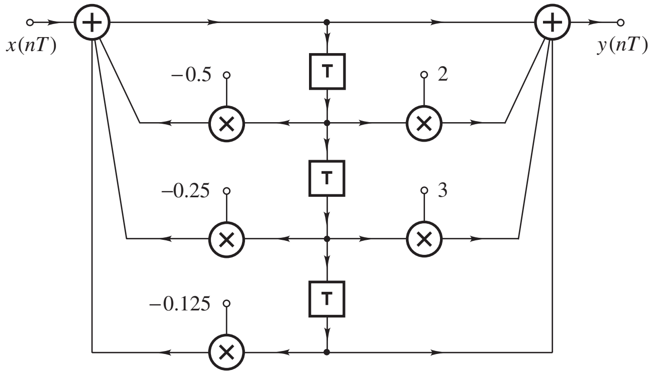


Figure P4.46

4.47. An initially relaxed discrete-time system is characterized by the state-space equations with

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{5}{16} & -1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \mathbf{c}^T = \begin{bmatrix} \frac{11}{8} & 2 \end{bmatrix} \qquad d = 2$$

(*a*) Calculate the impulse response for the period $nT = 0$ to $5T$ and for $nT = 17T$ using the state-space method.

(*b*) Calculate the unit-step response for $nT = 5T$.

4.48. (*a*) Deduce the difference equation of the system in Prob. 4.47.

(*b*) Calculate the impulse response for the period $nT = 0$ to $5T$ by using the difference equation.

(*c*) Calculate the unit-step response for $nT = 5T$ by using the difference equation.

4.49. A discrete-time system is characterized by the state-space equations with

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{4} & \frac{1}{2} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \mathbf{c}^T = \begin{bmatrix} -\frac{1}{4} & \frac{3}{2} \end{bmatrix} \qquad d = 1$$

(a) Assuming that $y(nT) = 0$ for $n < 0$, find $y(nT)$ for the period $nT = 0$ to $5T$ if $x(nT) = \delta(nT)$.

(b) Repeat part (a) if $x(nT) = u(nT)$

(c) Derive a network for the system.

4.50. A signal

$$x(nT) = 3u(nT)\cos\omega nT$$

is applied at the input of the system in Prob. 4.47. Find the response at instant 5T if $\omega = 1/10T$ by using the convolution summation.

4.51. Find the response of the system in Prob. 4.47 at $nT = 5T$ if the excitation is

$$x(nT) = u(nT - T)e^{-nT}$$

4.52. Find the response of the system in Prob. 4.49 at $nT = 5T$ if the excitation is

$$x(nT) = u(nT) + u(nT - 2T)$$

This page intentionally left blank

CHAPTER 5

THE APPLICATION OF THE *Z* TRANSFORM

5.1 INTRODUCTION

Through the use of the z transform, a discrete-time system can be characterized in terms of a so-called *discrete-time transfer function*, which is a complete representation of the system in the z domain. The transfer function can be used to find the response of a given system to an arbitrary time-domain excitation, to find its frequency response, and to ascertain whether the system is stable or unstable. Also, as will be shown in later chapters, the transfer function serves as the stepping stone between desired specifications and system design.

In this chapter, the discrete-time transfer function is defined and its properties are examined. It is then used as a tool for the stability, time-domain, and frequency-domain analysis of discrete-time systems. In Sec. 5.2, it is shown that the transfer function is a ratio of polynomials in complex variable z and, as a result, a discrete-time system can be represented by a set of zeros and poles. In Sec. 5.3, it is shown that the stability of a system is closely linked to the location of its poles. Several stability criteria are then presented, which are simple algorithms that enable one to determine with minimal computational effort whether a system is stable or unstable. Sections 5.4 and 5.5 deal with general time-domain and frequency-domain methods, respectively, that can be used to analyze systems of arbitrary order and complexity. The chapter concludes by introducing two types of system imperfection, known as *amplitude distortion* and *delay* (or *phase*) *distortion*, which can compromise the quality of the signal being processed.

5.2 THE DISCRETE-TIME TRANSFER FUNCTION

The *transfer function* of a discrete-time system is defined as the ratio of the z transform of the response to the z transform of the excitation.

Consider a linear, time-invariant, discrete-time system, and let $x(nT)$, $y(nT)$, and $h(nT)$ be the excitation, response, and impulse response, respectively. From the convolution summation in Eq. (4.36a), we have

$$y(nT) = \sum_{k=-\infty}^{\infty} x(kT)h(nT - kT)$$

and, therefore, from the real-convolution theorem (Theorem 3.7),

$$\mathcal{Z}y(nT) = \mathcal{Z}h(nT)\mathcal{Z}x(nT)$$

or

$$Y(z) = H(z)X(z)$$

In effect, *the transfer function of a discrete-time system is the z transform of the impulse response.*

Continuous-time systems can also be characterized in terms of transfer functions. In later chapters we shall be dealing with analog filters, which are continuous-time systems, and with digital filters, which are discrete-time systems, at the same time. To avoid possible confusion, we refer to the transfer functions of analog systems as *continuous-time* and those of digital systems as *discrete-time*.

The exact form of $H(z)$ can be derived (i) from the difference equation characterizing the system, (ii) from a network representation of the system, or (iii) from a state-space characterization, if one is available.

5.2.1 Derivation of $H(z)$ from Difference Equation

A noncausal, linear, time-invariant, recursive discrete-time system can be represented by the difference equation

$$y(nT) = \sum_{i=-M}^{N} a_i x(nT - iT) - \sum_{i=1}^{N} b_i y(nT - iT)$$

where M and N are positive integers. On applying the z transform to both sides of the difference equation, we get

$$\mathcal{Z}y(nT) = \mathcal{Z}\sum_{i=-M}^{N} a_i x(nT - iT) - \mathcal{Z}\sum_{i=1}^{N} b_i y(nT - iT)$$

If we use the linearity and time-shifting theorems of the z transform, we obtain

$$\begin{aligned} Y(z) = \mathcal{Z}y(nT) &= \sum_{i=-M}^{N} a_i z^{-i}\mathcal{Z}x(nT) - \sum_{i=1}^{N} b_i z^{-i}\mathcal{Z}y(nT) \\ &= \sum_{i=-M}^{N} a_i z^{-i}X(z) - \sum_{i=1}^{N} b_i z^{-i}Y(z) \end{aligned}$$

Now if we solve for $Y(z)/X(z)$ and then multiply the numerator and denominator polynomials by z^N, we get

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=-M}^{N} a_i z^{-i}}{1 + \sum_{i=1}^{N} b_i z^{-i}} = \frac{\sum_{i=-M}^{N} a_i z^{N-i}}{z^N + \sum_{i=1}^{N} b_i z^{N-i}}$$

$$= \frac{a_{(-M)} z^{M+N} + a_{(-M+1)} z^{M+N-1} + \cdots + a_N}{z^N + b_1 z^{N-1} + \cdots + b_N} \tag{5.1}$$

For example, if $M = N = 2$ we have

$$H(z) = \frac{N(z)}{D(z)} = \frac{a_{(-2)} z^4 + a_{(-1)} z^3 + a_0 z^2 + a_1 z + a_2}{z^2 + b_1 z + b_2}$$

For a causal, linear, time-invariant system, we have $M = 0$ and hence the transfer function assumes the form

$$H(z) = \frac{\sum_{i=0}^{N} a_i z^{N-i}}{z^N + \sum_{i=1}^{N} b_i z^{N-i}} = \frac{a_0 z^N + a_1 z^{N-1} + \cdots + a_N}{z^N + b_1 z^{N-1} + \cdots + b_N} \tag{5.2}$$

If we compare Eqs. (5.1) and (5.2), we note that *in a noncausal recursive system, the degree of the numerator polynomial is greater than that of the denominator polynomial.*

In a nonrecursive system, coefficients b_i are all zero and hence the above analysis gives

$$H(z) = a_{(-M)} z^M + a_{(-M+1)} z^{M-1} + \cdots + a_N z^{-N}$$

$$= \frac{a_{(-M)} z^{M+N} + a_{(-M+1)} z^{M+N-1} + \cdots + a_N}{z^N} \tag{5.3}$$

The order of a discrete-time transfer function, which is also the order of the system, is the order of $N(z)$ or $D(z)$, whichever is larger, i.e., $M + N$, if the system is noncausal or N if it is causal.

By factorizing the numerator and denominator polynomials, the transfer function of an arbitrary discrete-time system can be put in the form

$$H(z) = \frac{N(z)}{D(z)} = \frac{H_0 \prod_{i=1}^{Z} (z - z_i)^{m_i}}{\prod_{i=1}^{P} (z - p_i)^{n_i}} \tag{5.4}$$

where

$z_1, z_2, \ldots, z_Z$ are the zeros and $p_1, p_2, \ldots, p_N$ are the poles of $H(z)$,
m_i and n_i are the orders of zero z_i and pole p_i, respectively,
$M + N = \sum_{i=1}^{Z} m_i$ is the order of the numerator polynomial $N(z)$,
$N = \sum_{i=1}^{P} n_i$ is the order of the denominator polynomial $D(z)$, and
H_0 is a multiplier constant.

Thus a discrete-time system can be represented by a zero-pole plot such as the one in Fig. 5.1. From Eq. (5.3), we note that all the poles of a nonrecursive system are located at the origin of the z plane.

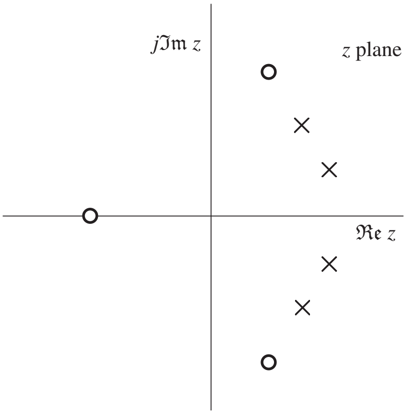


Figure 5.1 Typical zero-pole plot for $H(z)$.

5.2.2 Derivation of $H(z)$ from System Network

The z-domain characterizations of the unit delay, the adder, and the multiplier are obtained from Table 4.1 as

$$Y(z) = z^{-1}X(z) \qquad Y(z) = \sum_{i=1}^{K} X_i(z) \qquad \text{and} \qquad Y(z) = mX(z)$$

respectively. By using these relations, $H(z)$ can be derived directly from a network representation as illustrated in the following example.

Example 5.1 Find the transfer function of the system shown in Fig. 5.2.

■ **Solution**

From Fig. 5.2, we can write

$$W(z) = X(z) + \tfrac{1}{2}z^{-1}W(z) - \tfrac{1}{4}z^{-2}W(z)$$
$$Y(z) = W(z) + z^{-1}W(z)$$

Hence

$$W(z) = \frac{X(z)}{1 - \frac{1}{2}z^{-1} + \frac{1}{4}z^{-2}} \qquad \text{and} \qquad Y(z) = (1 + z^{-1})W(z)$$

Therefore,

$$\frac{Y(z)}{X(z)} = H(z) = \frac{z(z+1)}{z^2 - \frac{1}{2}z + \frac{1}{4}} \quad ■$$

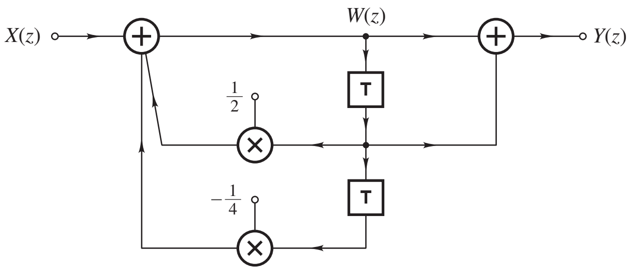


Figure 5.2 Second-order recursive system (Example 5.1).

5.2.3 Derivation of $H(z)$ from State-Space Characterization

Alternatively, $H(z)$ can be deduced from a state-space characterization. As was shown in Sec. 4.8.2, an arbitrary discrete-time system can be represented by the equations

$$\mathbf{q}(nT + T) = \mathbf{A}\mathbf{q}(nT) + \mathbf{b}x(nT) \tag{5.5a}$$

$$y(nT) = \mathbf{c}^T\mathbf{q}(nT) + dx(nT) \tag{5.5b}$$

(see Eqs. (4.51a) and (4.51b)). By applying the z transform to Eq. (5.5a), we obtain

$$\mathcal{Z}\mathbf{q}(nT + T) = \mathbf{A}\mathcal{Z}\mathbf{q}(nT) + \mathbf{b}\mathcal{Z}x(nT) = \mathbf{A}\mathbf{Q}(z) + \mathbf{b}X(z) \tag{5.6}$$

and since

$$\mathcal{Z}\mathbf{q}(nT + T) = z\mathcal{Z}\mathbf{q}(nT) = z\mathbf{Q}(z) \tag{5.7}$$

Equations (5.6) and (5.7) give

$$z\mathbf{Q}(z) = \mathbf{A}\mathbf{Q}(z) + \mathbf{b}X(z)$$

or

$$\mathbf{Q}(z) = (z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}X(z) \tag{5.8}$$

where $\mathbf{I}$ is the $N \times N$ identity matrix. Now on applying the z transform to Eq. (5.5b), we have

$$Y(z) = \mathbf{c}^T\mathbf{Q}(z) + dX(z)$$

and on eliminating $\mathbf{Q}(z)$ using Eq. (5.8), we get

$$\frac{Y(z)}{X(z)} = H(z) = \frac{N(z)}{D(z)} = \mathbf{c}^T(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d \tag{5.9}$$

Example 5.2 A discrete-time system can be represented by the state-space equations in Eq. (5.5) with

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{3} \\ 1 & 0 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \qquad \mathbf{c}^T = \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} \end{bmatrix} \qquad d = 2$$

Deduce the transfer function of the system.

■ **Solution**

The problem can be solved by evaluating the inverse of matrix

$$(z\mathbf{I} - \mathbf{A}) = \begin{bmatrix} z + \frac{1}{2} & \frac{1}{3} \\ -1 & z \end{bmatrix} \tag{5.10}$$

and then using Eq. (5.9).

The inverse of an $n \times n$ matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

is given by [1, 2]

$$\mathbf{A}^{-1} = \frac{1}{\det A} \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}^T \tag{5.11a}$$

where $\det \mathbf{A}$ is the *determinant* of $\mathbf{A}$,

$$A_{ij} = (-1)^{i+j} \det M_{ij}$$

and M_{ij} represents matrix $\mathbf{A}$ with its ith row and jth column deleted. A_{ij} and $\det M_{ij}$ are known as the *cofactor* and *minor determinant* of element a_{ij}, respectively. For a 2×2 matrix, we have

$$\begin{aligned} \mathbf{A}^{-1} &= \frac{1}{\det A} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^T = \frac{1}{\det A} \begin{bmatrix} a_{22} & -a_{21} \\ -a_{12} & a_{11} \end{bmatrix}^T \\ &= \frac{1}{\det A} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \end{aligned} \tag{5.11b}$$

Now from Eqs. (5.10) and (5.11b), we obtain

$$(z\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{(z + \frac{1}{2})z + \frac{1}{3}} \begin{bmatrix} z & -\frac{1}{3} \\ 1 & z + \frac{1}{2} \end{bmatrix} \tag{5.12}$$

and from Eqs. (5.9) and (5.12), we have

$$\begin{aligned} H(z) &= \mathbf{c}^T (z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d \\ &= \frac{1}{(z + \frac{1}{2})z + \frac{1}{3}} \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} z & -\frac{1}{3} \\ 1 & z + \frac{1}{2} \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} + 2 \\ &= \frac{1}{z^2 + \frac{1}{2}z + \frac{1}{3}} \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 2z \\ 2 \end{bmatrix} + 2 \\ &= \frac{-\frac{1}{2}z + \frac{1}{3} + 2z^2 + z + \frac{2}{3}}{z^2 + \frac{1}{2}z + \frac{1}{3}} = \frac{2z^2 + \frac{1}{2}z + 1}{z^2 + \frac{1}{2}z + \frac{1}{3}} \quad \blacksquare \end{aligned}$$

5.3 STABILITY

As can be seen in Eq. (5.1), the discrete-time transfer function is a rational function of z with real coefficients, and for causal systems the degree of the numerator polynomial is equal to or less than that of the denominator polynomial. We shall now show that the poles of the transfer function or, alternatively, the eigenvalues of matrix **A** in a state-space characterization, determine whether the system is stable or unstable.

5.3.1 Constraint on Poles

Consider a causal system with simple poles characterized by the transfer function

$$H(z) = \frac{N(z)}{D(z)} = \frac{H_0 \sum_{i=0}^{M} a_i z^{M-i}}{\prod_{i=1}^{N}(z - p_i)} \tag{5.13}$$

where $N \geq M$ and assume that the numerator and denominator polynomials $N(z)$ and $D(z)$ have no common factors that are not constants, i.e., they are *relatively prime.* Since such common factors can be canceled out at any time, they have no effect on the response of the system and, therefore, cannot affect its stability. The impulse response of such a system is given by

$$h(nT) = \mathcal{Z}^{-1}H(z) = \frac{1}{2\pi j} \oint_{\Gamma} H(z)z^{n-1}\,dz$$

and from Eq. (3.8), we get

$$h(0) = R_0 + \sum_{i=1}^{N} \mathfrak{Res}_{z=p_i}[z^{-1}H(z)] \tag{5.14a}$$

where

$$R_0 = \begin{cases} \mathfrak{Res}_{z=0}\left[\frac{H(z)}{z}\right] & \text{if } H(z)/z \text{ has a pole at the origin} \\ 0 & \text{otherwise} \end{cases}$$

and

$$h(nT) = \sum_{i=1}^{N} \mathop{\mathfrak{Res}}_{z=p_i}[H(z)z^{n-1}] \tag{5.14b}$$

for all $n > 0$.

Now if an arbitrary function $F(z)$ has a simple pole at $z = p_i$ and a function $G(z)$ is analytic at $z = p_i$, then it can be easily shown that

$$\mathop{\mathfrak{Res}}_{z=p_i}[F(z)G(z)] = G(p_i)\mathop{\mathfrak{Res}}_{z=p_i} F(z) \tag{5.14c}$$

(see Prob. 5.9). Thus Eqs. (5.14a)–(5.14c) give

$$h(nT) = \begin{cases} R_0 + \sum_{i=1}^{N} p_i^{-1}\,\mathfrak{Res}_{z=p_i}\, H(z) & \text{for } n = 0 \\ \sum_{i=1}^{N} p_i^{n-1}\,\mathfrak{Res}_{z=p_i}\, H(z) & \text{for } n > 0 \end{cases}$$

where the ith term in the summations is the contribution to the impulse response due to pole p_i. If we let

$$p_i = r_i e^{j\psi_i}$$

then the impulse response can be expressed as

$$h(nT) = \begin{cases} R_0 + \sum_{i=1}^{N} r_i^{-1}e^{-j\psi_i}\,\mathfrak{Res}_{z=p_i}\, H(z) & \text{for } n = 0 \\ \sum_{i=1}^{N} r_i^{n-1}e^{j(n-1)\psi_i}\,\mathfrak{Res}_{z=p_i}\, H(z) & \text{for } n > 0 \end{cases} \tag{5.15}$$

At this point, let us assume that all the poles are on or inside a circle of radius $r_{\max}$, that is,

$$r_i \le r_{\max} \qquad \text{for } i = 1, 2, \ldots, N \tag{5.16}$$

where $r_{\max}$ is the radius of the most distant pole from the origin. From Eq. (5.15), we can write

$$\sum_{n=0}^{\infty} |h(nT)| = \left| R_0 + \sum_{i=1}^{N} r_i^{-1}e^{-j\psi_i}\mathop{\mathfrak{Res}}_{z=p_i} H(z)\right| + \sum_{n=1}^{\infty}\left|\sum_{i=1}^{N} r_i^{n-1}e^{j(n-1)\psi_i}\mathop{\mathfrak{Res}}_{z=p_i} H(z)\right|$$

and since $|e^{j\theta}| = 1$ and the magnitude of a sum of complex numbers is always equal to or less than the sum of the magnitudes of the complex numbers (see Eq. (A.18)), we have

$$\sum_{n=0}^{\infty} |h(nT)| \le |R_0| + \sum_{i=1}^{N} r_i^{-1}|\mathop{\mathfrak{Res}}_{z=p_i} H(z)| + \sum_{n=1}^{\infty}\sum_{i=1}^{N} r_i^{n-1}|\mathop{\mathfrak{Res}}_{z=p_i} H(z)| \tag{5.17}$$

From the basics of complex analysis, if p_k is a simple pole of some function $F(z)$, then function $(z - p_k)F(z)$ is analytic at $z = p_k$ since the factor $(z - p_k)$ will cancel out the same factor in the denominator of $F(z)$ and will thereby remove pole p_k from $F(z)$. Hence, the residue of $F(z)$ at $z = p_k$ is a finite complex number in general. For this reason, R_0 as well as all the residues of $H(z)$ are finite and so

$$|\underset{z=p_i}{\mathfrak{Res}}\, H(z)| \le R_{\max} \qquad \text{for } i = 1, 2, \ldots, N$$

where $R_{\max}$ is the largest residue magnitude. If we replace the residue magnitudes by $R_{\max}$ and the radii of the poles by the largest pole radius $r_{\max}$ in Eq. (5.17), the inequality will continue to hold and thus

$$\sum_{n=0}^{\infty} |h(nT)| \le |R_0| + NR_{\max}r_{\max} + \frac{NR_{\max}}{r_{\max}} \sum_{n=1}^{\infty} r_{\max}^n$$

The sum at the right-hand side is a geometric series and if

$$r_{\max} < 1$$

the series converges and, therefore, we conclude that

$$\sum_{n=0}^{\infty} |h(nT)| \le K < \infty$$

where K is a finite constant. In effect, *if all the poles are inside the unit circle of the z plane, then the impulse response is absolutely summable.*

Let us now examine the situation where just a single pole of $H(z)$, let us say pole p_k, is located on or outside the unit circle. In such a case, as $n \to \infty$ the contributions to the impulse response due to all the poles other than pole p_k tend to zero since $r_i < 1$ and $r_i^{n-1} \to 0$ for $i \ne k$, whereas the contribution due to pole p_k either remains constant if $r_k = 1$ or tends to get larger and larger if $r_k > 1$ since r_k^{n-1} is increased as n is increased. Hence for a sufficiently large value of n, Eq. (5.15) can be approximated as

$$h(nT) \approx r_k^{n-1} e^{j(n-1)\psi_k} \underset{z=p_k}{\mathfrak{Res}}\, H(z)$$

and thus Eq. (5.17) gives

$$\sum_{n=0}^{\infty} |h(nT)| \approx |\underset{z=p_k}{\mathfrak{Res}}\, H(z)| \sum_{n=0}^{\infty} r_k^{n-1} \tag{5.18}$$

Since $r_k \ge 1$, the above geometric series diverges and as a consequence

$$\sum_{n=0}^{\infty} |h(nT)| \to \infty$$

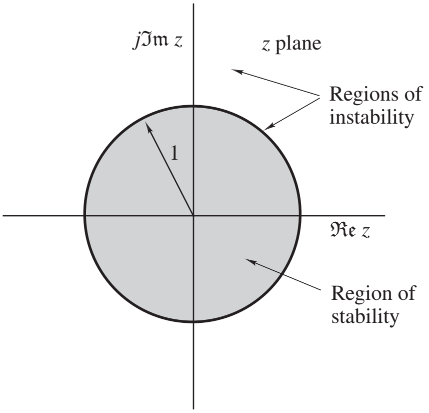


Figure 5.3 Permissible z-plane region for the location of the poles of $H(z)$.

That is, *if at least one pole is on or outside the unit circle, then the impulse response is not absolutely summable.*

From the above analysis, we conclude that *the impulse response is absolutely summable, if and only if all the poles are inside the unit circle.* Since the absolute summability of the impulse response is a necessary and sufficient condition for system stability, the inequality in Eq. (5.16) with $r_{\max} < 1$, that is,

$$|p_i| < 1 \qquad \text{for } i = 1, 2, \ldots, N$$

is also a necessary and sufficient condition for stability. The permissible region for the location of poles is illustrated in Fig. 5.3.

The above stability constraint has been deduced on the assumption that all the poles of the system are simple. However, the constraint applies equally well to the case where the system has one or more higher-order poles (see Prob. 5.10).

In Sec. 4.6.2, we found out that in nonrecursive systems the impulse response is always of finite duration and that assures its absolute summability and, in turn, the stability of these systems. This result is confirmed here by noting that the poles of these systems are always located at the origin of the z plane, right at the center of the region of stability, as can be seen in Eq. (5.3).

Example 5.3 Check the system of Fig. 5.4 for stability.

■ **Solution**

The transfer function of the system is

$$H(z) = \frac{z^2 - z + 1}{z^2 - z + \frac{1}{2}} = \frac{z^2 - z + 1}{(z - p_1)(z - p_2)}$$

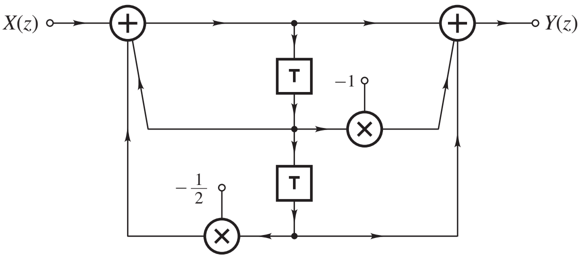


Figure 5.4 Second-order recursive system (Example 5.3).

where

$$p_1, p_2 = \tfrac{1}{2} \pm j\tfrac{1}{2}$$

since

$$|p_1|, |p_2| < 1$$

the system is stable. ■

5.3.2 Constraint on Eigenvalues

The poles of $H(z)$ are the values of z for which $D(z)$, the denominator polynomial of $H(z)$, becomes zero. The inverse of a matrix is given by the adjoint of the matrix divided by its determinant (see Eq. (5.11a)). Hence, $D(z)$ can be obtained from Eqs. (5.9) and (5.11a) as

$$D(z) = \det(z\mathbf{I} - \mathbf{A})$$

(see Example 5.4 below). Consequently, $D(z)$ is zero if and only if

$$\det(z\mathbf{I} - \mathbf{A}) = 0$$

Now the determinant of $(z\mathbf{I} - \mathbf{A})$ is the characteristic polynomial of matrix $\mathbf{A}$ [1, 2] and, consequently, the poles of an Nth-order transfer function $H(z)$ are numerically equal to the N *eigenvalues* $\lambda_1, \lambda_2, \ldots, \lambda_N$ of matrix $\mathbf{A}$. Therefore, a system characterized by the state-space equations in Eq. (5.5) is stable if and only if

$$|\lambda_i| < 1 \qquad \text{for } i = 1, 2, \ldots, N$$

Example 5.4 A discrete-time system is characterized by the state-space equations in Eq. (5.5) with

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{3} & -\frac{1}{4} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{c}^T = \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} & \frac{1}{12} \end{bmatrix} \qquad d = 2$$

Check the system for stability.

■ **Solution**

One approach to the problem would be to find the denominator of the transfer function $D(z)$ and then find the zeros of $D(z)$, which are the poles of the transfer function. We can write

$$z\mathbf{I} - \mathbf{A} = \begin{bmatrix} z + \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ -1 & z & 0 \\ 0 & -1 & z \end{bmatrix}$$

and from Eq. (5.11a), we obtain

$$(z\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{\det(z\mathbf{I} - \mathbf{A})} \begin{bmatrix} z^2 & -z & 1 \\ \frac{1}{3}z + \frac{1}{4} & (z + \frac{1}{2})z & -(z + \frac{1}{2}) \\ -\frac{1}{4}z & \frac{1}{4} & (z + \frac{1}{2})z + \frac{1}{3} \end{bmatrix}^T$$

Hence Eq. (5.9) yields

$$\frac{Y(z)}{X(z)} = H(z) = \frac{N(z)}{D(z)} = \mathbf{c}^T (z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d$$

$$= \frac{1}{\det(z\mathbf{I} - \mathbf{A})} \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} & \frac{1}{12} \end{bmatrix} \begin{bmatrix} z^2 & \frac{1}{3}z + \frac{1}{4} & -\frac{1}{4}z \\ -z & (z + \frac{1}{2})z & \frac{1}{4} \\ 1 & -(z + \frac{1}{2}) & (z + \frac{1}{2})z + \frac{1}{3} \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + 2$$

$$= \frac{1}{\det(z\mathbf{I} - \mathbf{A})} \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} & \frac{1}{12} \end{bmatrix} \begin{bmatrix} 2z^2 \\ -2z \\ 2 \end{bmatrix} + 2$$

Thus polynomials $N(z)$ and $D(z)$ can be deduced as

$$N(z) = \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} & \frac{1}{12} \end{bmatrix} \begin{bmatrix} 2z^2 \\ -2z \\ 2 \end{bmatrix} + 2 \det(z\mathbf{I} - \mathbf{A}) \tag{5.19a}$$

and

$$D(z) = \det(z\mathbf{I} - \mathbf{A}) \tag{5.19b}$$

respectively. Since $N(z)$ has nothing to do with stability, all we need to do is to find the determinant of matrix $z\mathbf{I} - \mathbf{A}$.

The determinant of a 3×3 matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

can be readily obtained by writing two copies of the matrix side by side as follows

$$\left| \begin{array}{ccccccccccc} a_{11} & & a_{12} & & a_{13} & | & a_{11} & & a_{12} & & a_{13} \\ & \searrow & & \searrow & & \searrow & & & & & \\ a_{21} & & a_{22} & & a_{23} & | & a_{21} & & a_{22} & & a_{23} \\ & & & \searrow & & \searrow & & \searrow & & & \\ a_{31} & & a_{32} & & a_{33} & | & a_{31} & & a_{32} & & a_{33} \end{array} \right|$$

The sum of element products along the south-east diagonals shown form the positive part of the determinant,

$$D^{+} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}$$

On the other hand, the sum of the products along the south-west diagonals shown below

$$\left| \begin{array}{ccccccccccc} a_{11} & & a_{12} & & a_{13} & | & a_{11} & & a_{12} & & a_{13} \\ & & & & & \swarrow & & \swarrow & & \swarrow & \\ a_{21} & & a_{22} & & a_{23} & | & a_{21} & & a_{22} & & a_{23} \\ & & & \swarrow & & \swarrow & & \swarrow & & & \\ a_{31} & & a_{32} & & a_{33} & | & a_{31} & & a_{32} & & a_{33} \end{array} \right|$$

form the negative part of the determinant,

$$D^{-} = a_{11}a_{23}a_{32} + a_{12}a_{21}a_{33} + a_{13}a_{22}a_{31}$$

The determinant of $\mathbf{A}$ is given by

$$\begin{aligned} \det \mathbf{A} &= D^{+} - D^{-} \\ &= (a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}) \\ &\quad -(a_{11}a_{23}a_{32} + a_{12}a_{21}a_{33} + a_{13}a_{22}a_{31}) \end{aligned} \tag{5.20}$$

Thus from Eq. (5.19b) and the above arrays (or from Eq. (5.20)), we obtain

$$\begin{aligned} D(z) = \det(z\mathbf{I} - \mathbf{A}) &= \left[(z + \tfrac{1}{2})z^2 + \tfrac{1}{4}\right] - \left(-\tfrac{1}{3}z\right) \\ &= z^3 + \tfrac{1}{2}z^2 + \tfrac{1}{3}z + \tfrac{1}{4} \end{aligned}$$

Using function *roots* of MATLAB, the poles of the system can be obtained as

$$p_0 = -0.6168 \qquad \text{or} \qquad |p_0| = 0.6168$$
$$p_1,\ p_2 = \quad 0.0584 \pm j0.6340 \qquad \text{or} \qquad |p_1| = |p_2| = 0.6367$$

and since $|p_i| < 1$ for $i = 0,\ 1$, and 2, the system is stable. ■

If the pole positions are not required, then the stability of the system can be easily ascertained by applying the Jury-Marden stability criterion (see Sec. 5.3.7).

5.3.3 Stability Criteria

The stability of a system can be checked by finding the roots of polynomial $D(z)$ or the eigenvalues of matrix $\mathbf{A}$ in a state-space representation. For a second- or third-order system, this is easily accomplished. For higher-order systems, however, the use of a computer program[1] is necessary. In certain applications, the designer may simply need to know whether a system is stable or unstable and the values of the poles of the transfer function may not be required. In such applications, the stability of the system can be checked quickly through the use of one of several available *stability tests* or *criteria* like the Schur-Cohn and Jury-Marden criteria [3]. Typically, these criteria are simple algorithms that involve an insignificant amount of computation relative to that required to find the roots of $D(z)$.

Some of the more important stability criteria will now be described. Derivations and proofs are omitted for the sake of brevity but the interested reader may consult the references at the end of the chapter.

Consider a system characterized by the transfer function

$$H(z) = \frac{N(z)}{D(z)} \tag{5.21}$$

where

$$N(z) = \sum_{i=0}^{M} a_i z^{M-i} \tag{5.22a}$$

and

$$D(z) = \sum_{i=0}^{N} b_i z^{N-i} \tag{5.22b}$$

and assume that $b_0 > 0$. This assumption simplifies the exposition of the stability criteria quite a bit. If b_0 happens to be negative, a positive b_0 can be obtained by simply replacing all the coefficients in $D(z)$ by their negatives. This modification amounts to multiplying the numerator and denominator of the transfer function by -1 and since such a manipulation does not change the response of the system, it does not affect its stability. Assume also that $N(z)$ and $D(z)$ have no common factors that are not constants. If there are such common factors in these polynomials, they must be identified and

[1] *For example, function roots of MATLAB.*

canceled out before the application of one of the stability criteria. Otherwise, a false result may be obtained, for example, if a common factor has a root inside the unit circle. In such a case, the transfer function will have a pole inside the unit circle that has nothing to do with the stability of the system.

5.3.4 Test for Common Factors

The presence of common factors in $N(z)$ and $D(z)$ can be checked by applying the following test. The coefficients of $N(z)$ and $D(z)$ are used to construct the $N \times (N+M)$ and $M \times (N+M)$ matrices

$$\mathbf{R}_N = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & & & a_M & 0 & \cdots & 0 & 0 \\ 0 & a_0 & a_1 & \cdots & & & a_{M-1} & a_M & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & & & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_0 & a_1 & \cdots & & & a_{M-1} & a_M \end{bmatrix}$$

and

$$\mathbf{R}_M = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 & b_0 & \cdots & & b_{N-1} & b_N \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & b_0 & b_1 & \cdots & & & b_N & & \cdots & 0 & 0 & 0 \\ b_0 & b_1 & b_2 & \cdots & & b_N & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}$$

respectively. Then the $(N+M) \times (N+M)$ matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_N \\ \mathbf{R}_M \end{bmatrix}$$

is formed and its determinant is computed. If

$$\det \mathbf{R} \neq 0$$

then $N(z)$ and $D(z)$ do not have a common factor that is not a constant, i.e., the two polynomials are relatively prime [4, 5]. Otherwise, if

$$\det \mathbf{R} = 0$$

the two polynomials are not relatively prime.

In most practical situations, for example, in the transfer functions obtained through the design processes to be described in later chapters, polynomials $N(z)$ and $D(z)$ are almost always relatively prime but the possibility that they might not be should not be totally ignored.

Example 5.5 Check the numerator and denominator polynomials of the transfer function

$$H(z) = \frac{N(z)}{D(z)} = \frac{z^2 + 3z + 2}{3z^3 + 5z^2 + 3z + 1}$$

for common factors.

■ **Solution**

Matrix **R** can be formed as

$$\mathbf{R} = \begin{bmatrix} 1\,3\,2\,0\,0 \\ 0\,1\,3\,2\,0 \\ 0\,0\,1\,3\,2 \\ 0\,3\,5\,3\,1 \\ 3\,5\,3\,1\,0 \end{bmatrix}$$

Through the use of MATLAB, we find that $\det \mathbf{R} = 0$. Therefore, $N(z)$ and $D(z)$ have a common factor that is not a constant. In actual fact

$$H(z) = \frac{(z+1)(z+2)}{(z+1)(3z^2+2z+1)} \quad ■$$

5.3.5 Schur-Cohn Stability Criterion

The *Schur-Cohn* stability criterion was established during the early twenties [3], long before the era of digital systems, and its main application at that time was as a mathematical tool for the purpose of establishing whether or not a general polynomial of z has zeros inside the unit circle of the z plane. This criterion has been superseded in recent years by other more efficient criteria and is rarely used nowadays. Nevertheless, it is of interest as it is the basis of some of the modern criteria.

The Schur-Cohn criterion states that a polynomial $D(z)$ of the type given in Eq. (5.22b), whose coefficients may be complex, has roots inside the unit circle of the z plane if and only if

$$\det \mathbf{S} \begin{cases} < 0 & \text{if } k \text{ is odd} \\ > 0 & \text{if } k \text{ is even} \end{cases}$$

for $k = 1, 2, \ldots, N$ where $\mathbf{S}_k$ is a $2k \times 2k$ matrix given by

$$\mathbf{S}_k = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{B}_k^T & \mathbf{A}_k^T \end{bmatrix}$$

with

$$\mathbf{A}_k = \begin{bmatrix} b_N & 0 & 0 & \cdots & 0 \\ b_{N-1} & b_N & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{N-k+1} & b_{N-k+2} & b_{N-k+3} & \cdots & b_N \end{bmatrix}$$

and

$$\mathbf{B}_k = \begin{bmatrix} b_0 & b_1 & b_2 & \cdots & b_{k-1} \\ 0 & b_0 & b_1 & \cdots & b_{k-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & b_0 \end{bmatrix}$$

The polynomial coefficients $b_0, b_1, \ldots, b_N$ can, in general, be complex. Polynomials whose roots are inside the unit circle are sometimes referred to as *Schur polynomials* [5].

The Schur-Cohn criterion involves the evaluation of the determinants of N matrices of dimensions ranging from 2×2 to $2N \times 2N$, which would require a large amount of computation.

5.3.6 Schur-Cohn-Fujiwara Stability Criterion

A more efficient stability criterion was developed by Fujiwara during the mid-twenties [3]. This is actually a modified version of the Schur-Cohn criterion and for this reason it is usually referred to as the *Schur-Cohn-Fujiwara* criterion. In this criterion, the coefficients of $D(z)$, which can be complex, are used to construct the $N \times N$ matrix

$$\mathbf{F} = \begin{bmatrix} f_{11} & \cdots & f_{1N} \\ \vdots & \vdots & \vdots \\ f_{N1} & \cdots & f_{NN} \end{bmatrix}$$

where

$$f_{ij} = \sum_{k=1}^{\min(i,j)} (b_{i-k}\, b_{j-k} - b_{N-i+k}\, b_{N-j+k}) \tag{5.23}$$

The Schur-Cohn-Fujiwara criterion states that the zeros of $D(z)$ are located inside the unit circle if and only if $\mathbf{F}$ is a positive definite matrix.

An $N \times N$ matrix $\mathbf{F}$ is said to be *positive definite* if the quadratic form $\mathbf{x}^T\mathbf{F}\mathbf{x}$ is a positive quantity for every nonzero column vector $\mathbf{x}^T$ of dimension N. Matrix $\mathbf{F}$ is positive definite if and only if its principal minor determinants (or simply minors) are positive [1, 2], that is,

$$|f_{11}| > 0 \qquad \begin{vmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{vmatrix} > 0$$

$$\begin{vmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{vmatrix} > 0 \qquad \ldots \qquad \begin{vmatrix} f_{11} & \cdots & f_{1N} \\ \vdots & \vdots & \vdots \\ f_{N1} & \cdots & f_{NN} \end{vmatrix} > 0$$

Evidently, like the original Schur-Cohn criterion, this criterion involves the evaluation of N determinants. However, the dimensions of the matrices involved now range from 1×1 to $N \times N$ and, therefore, the amount of computation is significantly reduced.

It should be mentioned that matrix $\mathbf{F}$ is symmetrical with respect to both the main and cross diagonals, i.e.,

$$f_{ij} = f_{ji} = f_{(N+1-i)(N+1-j)} = f_{(N+1-j)(N+1-i)}$$

As a result, only the elements with subscripts $i = 1$ to K and $j = i, i+1, \ldots, N+1-i$ need to be computed where

$$K = \begin{cases} (N+1)/2 & \text{for } N \text{ odd} \\ N/2 & \text{for } N \text{ even} \end{cases}$$

These are the elements covered by the triangle formed by three lines drawn through the first row, the main diagonal, and the cross diagonal.

Example 5.6 (*a*) A digital system is characterized by the transfer function

$$H(z) = \frac{z^4}{4z^4 + 3z^3 + 2z^2 + z + 1}$$

Check the system for stability using the Schur-Cohn-Fujiwara criterion. (*b*) Repeat part (*a*) if

$$H(z) = \frac{z^2 + 2z + 1}{z^4 + 6z^3 + 3z^2 + 4z + 5}$$

■ **Solution**

(*a*) The denominator polynomial of the transfer function is given by

$$D(z) = 4z^4 + 3z^3 + 2z^2 + z + 1$$

Using Eq. (5.23), the Fujiwara matrix can be constructed as

$$\mathbf{F} = \begin{bmatrix} 15 & 11 & 6 & 1 \\ 11 & 23 & 15 & 6 \\ 6 & 15 & 23 & 11 \\ 1 & 6 & 11 & 15 \end{bmatrix}$$

The principal minors can be obtained as

$$|15| = 15 \qquad \begin{vmatrix} 15 & 11 \\ 11 & 23 \end{vmatrix} = 224$$

$$\begin{vmatrix} 15 & 11 & 6 \\ 11 & 23 & 15 \\ 6 & 15 & 23 \end{vmatrix} = 2929 \qquad \begin{vmatrix} 15 & 11 & 6 & 1 \\ 11 & 23 & 15 & 6 \\ 6 & 15 & 23 & 11 \\ 1 & 6 & 11 & 15 \end{vmatrix} = 27{,}753$$

and since they are all positive, the system is stable. ■

(*b*) In this case

$$D(z) = z^4 + 6z^3 + 3z^2 + 4z + 5$$

and hence Eq. (5.23) gives

$$|f_{11}| = b_0^2 - b_4^2 = -24$$

i.e., the principal minor of order 1 is negative, and the system can be classified as unstable. There is no need to compute the remaining principal minors because a matrix cannot be positive definite if any one of its principal minors is zero or negative. ■

A simplified version of the Schur-Cohn stability criterion was described by Jury in 1962 [6] (see also Chap. 3 of Ref. [3]) and a simplified version of the Schur-Cohn-Fujiwara criterion was described by Anderson and Jury in 1973 [7].

5.3.7 Jury-Marden Stability Criterion

A stability criterion that has been applied widely through the years is one developed by Jury during the early sixties [3] using a relation due to Marden [8] that gives the Schur-Cohn determinants in terms of second-order determinants. This criterion is often referred to as the *Jury-Marden criterion* and, as is demonstrated below, it is both very efficient and easy to apply. In this criterion, the coefficients of $D(z)$, which are assumed to be real, are used to construct an array of numbers known as the *Jury-Marden array*, as in Table 5.1. The first two rows of the array are formed by entering the coefficients of $D(z)$ directly in ascending order for the first row and in descending order for the second. The elements of the third and fourth rows are computed as

$$c_i = \begin{vmatrix} b_i & b_N \\ b_{N-i} & b_0 \end{vmatrix} = b_i b_0 - b_{N-i} b_N \qquad \text{for } i = 0,\ 1,\ \ldots,\ N-1$$

Table 5.1 The Jury-Marden array

Row	Coefficients					
1	b_0	b_1	b_2	b_3	$\cdots$	b_N
2	b_N	b_{N-1}	b_{N-2}	b_{N-3}	$\cdots$	b_0
3	c_0	c_1	c_2	$\cdots$	c_{N-1}	
4	c_{N-1}	c_{N-2}	c_{N-3}	$\cdots$	c_0	
5	d_0	d_1	$\cdots$	d_{N-2}		
6	d_{N-2}	d_{N-3}	$\cdots$	d_0		
	$\vdots$	$\vdots$	$\cdots$			
2N − 3	r_0	r_1	r_2			

those of the fifth and sixth rows as

$$d_i = \begin{vmatrix} c_i & c_{N-1} \\ c_{N-1-i} & c_0 \end{vmatrix} = c_i c_0 - c_{N-1-i} c_{N-1} \qquad \text{for } i = 0,\ 1,\ \ldots,\ N-2$$

and so on until $2N - 3$ rows are obtained. The last row comprises three elements, say, r_0, r_1, and r_2.

The Jury-Marden criterion states that polynomial $D(z)$ has roots inside the unit circle of the z plane if and only if the following conditions are satisfied:

(i) $D(1) > 0$

(ii) $(-1)^N D(-1) > 0$

(iii) $b_0 > |b_N|$
$|c_0| > |c_{N-1}|$
$|d_0| > |d_{N-2}|$
$\ldots\ldots\ldots\ldots$
$|r_0| > |r_2|$

As can be seen, the Jury-Marden criterion involves determinants of 2 × 2 matrices and is easy to apply even without the use of a computer. Note that *all three* of the preceding three conditions must be satisfied for the system to be stable. Therefore, the Jury-Marden array need not be constructed if either of conditions (i) or (ii) is violated. If these conditions are satisfied, then one can begin evaluating the elements of the Jury-Marden array. If a row is encountered where the magnitude of the first coefficient is equal to or less than the magnitude of the last coefficient, then the construction of the array can be terminated and the system declared unstable.

Thus to save unnecessary effort, conditions (i) and (ii) should be checked first. If they are satisfied, then one can proceed with the Jury-Marden array.

Example 5.7 Check the systems of Example 5.6, parts (*a*) and (*b*), for stability using the Jury-Marden criterion.

■ **Solution**

(*a*) We have

$$D(1) = 11 \qquad (-1)^4 D(-1) = 3$$

and thus conditions (i) and (ii) are satisfied. The Jury-Marden array can be constructed as shown in Table 5.2 and since $b_0 > |b_4|$, $|c_0| > |c_3|$, $|d_0| > |d_2|$, condition (iii) is also satisfied and the system is stable. ■

(*b*) In this case

$$(-1)^4 D(-1) = -1$$

i.e., condition (ii) is violated and the system is unstable. ■

Table 5.2 Jury-Marden array for Example 5.7

Row	Coefficients				
1	4	3	2	1	1
2	1	1	2	3	4
3	15	11	6	1	
4	1	6	11	15	
5	224	159	79		

Example 5.8 A discrete-time system is characterized by the transfer function

$$H(z) = \frac{z^4}{7z^4 + 3z^3 + mz^2 + 2z + 1}$$

Find the range of m that will result in a stable system.

■ **Solution**

The transfer function can be expressed as

$$H(z) = \frac{N(z)}{D(z)}$$

where

$$D(z) = 7z^4 + 3z^3 + mz^2 + 2z + 1$$

The stability problem can be solved by finding the range of m that satisfies all the conditions imposed by the Jury-Marden stability criterion.
Condition (i) gives

$$D(1) = 7 + 3 + m + 2 + 1 > 0$$

or

$$m > -13 \tag{5.24}$$

From condition (ii), we have

$$(-1)^4 D(-1) = 7 - 3 + m - 2 + 1 > 0$$

Table 5.3 Jury-Marden array for Example 5.8

Row	Coefficients				
1	7	3	m	2	1
2	1	2	m	3	7
3	48	19	$6m$	11	
4	11	$6m$	19	48	
5	2183	$912 - 66m$	$288m - 209$		

or

$$m > -3 \tag{5.25}$$

The Jury-Marden array can be constructed as shown in Table 5.3.

Hence for stability, the conditions

$$7 > 1 \qquad |48| > |11| \qquad |2183| > |288m - 209|$$

must be satisfied. The third condition is satisfied if

$$2183 > \pm(288m - 209)$$

i.e.,

$$m > -\frac{2183 - 209}{288} \qquad \text{or} \qquad m < \frac{2183 + 209}{288}$$

which implies that

$$m > -6.8542 \qquad \text{or} \qquad m < 8.3056 \tag{5.26}$$

Now for stability all the Jury-Marden conditions must be satisfied and thus from Eqs. (5.24)–(5.26), the allowable range of m is obtained as

$$-3 < m < 8.3056 \quad \blacksquare$$

5.3.8 Lyapunov Stability Criterion

Another stability criterion states that a discrete-time systems characterized by a state-space representation is stable if and only if for any positive definite matrix **Q**, there exists a unique positive definite matrix **P** that satisfies the *Lyapunov equation* [9]

$$\mathbf{A}^T\mathbf{P}\mathbf{A} - \mathbf{P} = -\mathbf{Q}$$

In this criterion, a positive definite matrix **Q** is assumed, say **Q** = **I**, and the Lyapunov equation is solved for **P** [10]. If **P** is found to be positive definite, the system is classified as stable. This criterion is less practical to apply than the Jury-Marden criterion and, as a consequence, it is not used for routine analysis. Nevertheless, it has some special features that make it suitable for the study of certain parasitic oscillations that can occur in digital filters (see Sec. 14.9).

5.4 TIME-DOMAIN ANALYSIS

The *time-domain* response of a discrete-time system to any excitation $x(nT)$ can be readily obtained from Eq. (5.1) as

$$y(nT) = \mathcal{Z}^{-1}[H(z)X(z)]$$

Any one of the inversion techniques described in Sec. 3.8 can be used.

Example 5.9 Find the unit-step response of the system shown in Fig. 5.4.

■ **Solution**

From Example 5.3

$$H(z) = \frac{z^2 - z + 1}{(z - p_1)(z - p_2)}$$

where

$$p_1 = \tfrac{1}{2} - j\tfrac{1}{2} = \frac{e^{-j\pi/4}}{\sqrt{2}} \qquad \text{and} \qquad p_2 = \tfrac{1}{2} + j\tfrac{1}{2} = \frac{e^{j\pi/4}}{\sqrt{2}}$$

and from Table 3.2

$$X(z) = \frac{z}{z - 1}$$

On expanding $H(z)X(z)/z$ into partial fractions, we have

$$H(z)X(z) = \frac{R_0 z}{z - 1} + \frac{R_1 z}{z - p_1} + \frac{R_2 z}{z - p_2}$$

where

$$R_0 = 2 \qquad R_1 = \frac{e^{j5\pi/4}}{\sqrt{2}} \qquad \text{and} \qquad R_2 = R_1^* = \frac{e^{-j5\pi/4}}{\sqrt{2}}$$

Hence

$$\begin{aligned} y(nT) &= \mathcal{Z}^{-1}[H(z)X(z)] \\ &= 2u(nT) + \frac{1}{(\sqrt{2})^{n+1}}u(nT)(e^{j(n-5)\pi/4} + e^{-j(n-5)\pi/4}) \\ &= 2u(nT) + \frac{1}{(\sqrt{2})^{n-1}}u(nT)\cos\left[(n-5)\frac{\pi}{4}\right] \quad \blacksquare \end{aligned}$$

The unit-step response of the system is plotted in Fig. 5.5.

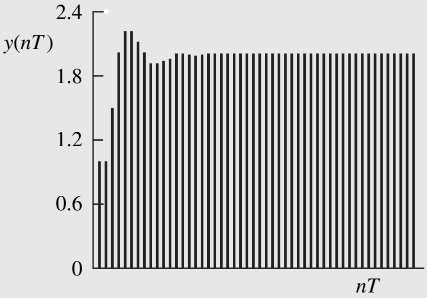


Figure 5.5 Unit-step response (Example 5.9).

5.5 FREQUENCY-DOMAIN ANALYSIS

The response of a first-order discrete-time system to a sinusoidal excitation was examined in Sec. 4.5 and it was found to comprise two components, a transient and a steady-state sinusoidal component. We will now show that the same is also true for the response of a system of arbitrary order. If the system is stable, the transient component tends to diminish rapidly to zero as time advances and in due course only the sinusoidal component prevails. The amplitude and phase angle of the sinusoidal output waveform produced by a sinusoidal waveform of unit amplitude and zero phase angle turn out to be functions of frequency. Together they enable one to determine the steady-state response of a system to a sinusoidal waveform of arbitrary frequency or the response produced by arbitrary linear combinations of sinusoidal waveforms, and can be used, in addition, to find the responses produced by complex waveforms.

5.5.1 Steady-State Sinusoidal Response

Let us consider a causal system characterized by the transfer function of Eq. (5.13). The sinusoidal response of such a system is

$$y(nT) = \mathcal{Z}^{-1}[H(z)X(z)]$$

where

$$X(z) = \mathcal{Z}[u(nT)\sin\omega nT] = \frac{z\sin\omega T}{(z-e^{j\omega T})(z-e^{-j\omega T})} \tag{5.27}$$

or

$$y(nT) = \frac{1}{2\pi j}\oint_\Gamma H(z)X(z)z^{n-1}\,dz$$
$$= \sum_{\text{All poles}} \Re es[H(z)X(z)z^{n-1}] \tag{5.28a}$$

Assuming that the poles of the system are simple, then for $n > 0$ Eqs. (5.27) and (5.28a) yield

$$y(nT) = \underset{z=e^{j\omega T}}{\Re es}[H(z)X(z)z^{n-1}] + \underset{z=e^{-j\omega T}}{\Re es}[H(z)X(z)z^{n-1}] + \sum_{i=1}^{N}\underset{z=p_i}{\Re es}[H(z)X(z)z^{n-1}]$$
$$= \frac{1}{2j}[H(e^{j\omega T})e^{j\omega nT} - H(e^{-j\omega T})e^{-j\omega nT}] + \sum_{i=1}^{N}\underset{z=p_i}{\Re es}[H(z)X(z)z^{n-1}] \tag{5.28b}$$

and if we let $p_i = r_i e^{j\psi_i}$, the summation part in the above equation can be expressed as

$$\sum_{i=1}^{N}\underset{z=p_i}{\Re es}[H(z)X(z)z^{n-1}] = \sum_{i=1}^{N} X(p_i)p_i^{n-1}\underset{z=p_i}{\Re es}\,H(z)$$

(see Prob. 5.9). Now if the system is stable, then $|p_i| = r_i < 1$ for $i = 1, 2, \ldots, N$ and hence as $n \to \infty$, we have $r_i^{n-1} \to 0$. Thus $p_i^{n-1} = r_i^{n-1}e^{j(n-1)\psi_i} \to 0$ and, therefore,

$$\lim_{n\to\infty}\sum_{i=1}^{N}\underset{z=p_i}{\Re es}[H(z)X(z)z^{n-1}] = \lim_{n\to\infty}\sum_{i=1}^{N} X(p_i)p_i^{n-1}\underset{z=p_i}{\Re es}\,H(z) \to 0 \tag{5.28c}$$

Hence, Eqs. (5.28b) and (5.28c) give the steady-state sinusoidal response of the system as

$$\tilde{y}(nT) = \lim_{n\to\infty} y(nT) = \frac{1}{2j}\left[H(e^{j\omega T})e^{j\omega nT} - H(e^{-j\omega T})e^{-j\omega nT}\right] \tag{5.28d}$$

This result also holds true for systems that have one or more higher-order poles (see Prob. 5.28) as well as for noncausal systems as can be easily demonstrated.

From the linearity of complex conjugation, *the sum of a number of complex conjugates is equal to the complex conjugate of the sum*, and if we use the z transform, we obtain

$$H(e^{-j\omega T}) = \sum_{n=-\infty}^{\infty} h(nT)e^{j\omega nT} = \left[\sum_{n=-\infty}^{\infty} h(nT)e^{-j\omega nT}\right]^* = H^*(e^{j\omega T}) \tag{5.28e}$$

If we let

$$H(e^{j\omega T}) = M(\omega)e^{j\theta(\omega)}$$

where $$M(\omega) = |H(e^{j\omega T})| \qquad \text{and} \qquad \theta(\omega) = \arg H(e^{j\omega T}) \tag{5.29}$$

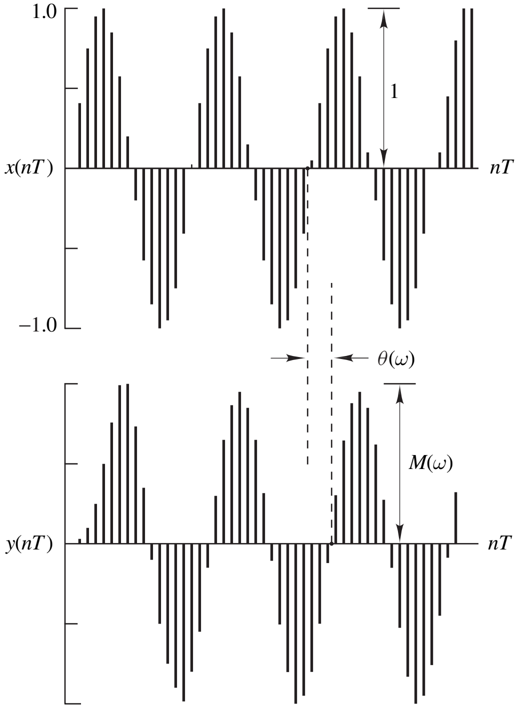


Figure 5.6 Sinusoidal response of an arbitrary system.

then from Eqs. (5.28d) and (5.28e), the steady-state response of the system can be expressed as

$$\begin{aligned}\tilde{y}(nT) &= \frac{1}{2j}\left[H(e^{j\omega T})e^{j\omega nT} - H^*(e^{j\omega T})e^{-j\omega nT}\right]\\ &= \frac{1}{2j}\left[M(\omega)e^{j[\omega nT+\theta(\omega)]} - M(\omega)e^{-j[\omega nT+\theta(\omega)]}\right]\\ &= M(\omega)\sin[\omega nT + \theta(\omega)] \qquad (5.30)\end{aligned}$$

Clearly, the effect of a system on a sinusoidal excitation is to introduce a *gain* $M(\omega)$ and a *phase shift* $\theta(\omega)$, as illustrated in Fig. 5.6. As functions of frequency, $M(\omega)$ and $\theta(\omega)$ are known as the *amplitude and phase responses* and function $H(e^{j\omega T})$ from which they are derived is referred to as the *frequency response* of the system.[2]

As may be recalled from Sec. 3.9.1, the frequency spectrum of a discrete-time signal is the z transform of the signal evaluated on the unit circle of the z plane. Since $H(z)$ is the z transform of the impulse response, it follows that $H(e^{j\omega T})$ is also the frequency spectrum of the impulse response.

[2] *Some people refer to $M(\omega)$ as the magnitude response for obvious reasons.*

That is, function $H(e^{j\omega T})$ has a dual physical interpretation, namely, it is the frequency response of the system or the frequency spectrum of the impulse response.

In digital filters, the gain often varies over several orders of magnitude as the frequency is varied and to facilitate the plotting of the amplitude response, the gain is usually measured in decibels (dB) as

$$\text{Gain} = 20\log_{10} M(\omega)$$

The gain in filters is typically equal to or less than unity and it is usually convenient to work with the reciprocal of the gain, which is known as *attenuation*. Like the gain, the attenuation can be expressed in dB as

$$\text{Attenuation} = 20\log_{10}\frac{1}{M(\omega)} = -20\log_{10} M(\omega)$$

The phase shift is measured either in degrees or in radians.

5.5.2 Evaluation of Frequency Response

The above analysis has shown that the amplitude and phase responses of a system can be obtained by evaluating the transfer function $H(z)$ on the unit circle $|z| = 1$ of the z plane, which is very much what we do to find the amplitude and phase spectrums of a discrete-time signal. This can be done very efficiently by using MATLAB or other similar digital signal processing (DSP) software. It can also be done by using a graphical method as will be demonstrated below. The method is inefficient and is unlikely to be used in practice, yet it merits consideration because it reveals some of the basic properties of discrete-time systems and provides, in addition, intuitive appreciation of the influence of the zero and pole locations on the amplitude response of a system.

Let us consider a general transfer function expressed in terms of its zeros and poles as in Eq. (5.4). The frequency response of the system at some frequency ω, can be obtained as

$$H(z)\,|_{z\to e^{j\omega T}} = H(e^{j\omega T}) = M(\omega)e^{j\theta(\omega)} \tag{5.31}$$

$$= \frac{H_0\prod_{i=1}^{Z}(e^{j\omega T}-z_i)^{m_i}}{\prod_{i=1}^{P}(e^{j\omega T}-p_i)^{n_i}} \tag{5.32}$$

and by letting

$$e^{j\omega T} - z_i = M_{z_i}e^{j\psi_{z_i}} \tag{5.33a}$$

$$e^{j\omega T} - p_i = M_{p_i}e^{j\psi_{p_i}} \tag{5.33b}$$

we obtain

$$M(\omega) = \frac{|H_0|\prod_{i=1}^{Z}M_{z_i}^{m_i}}{\prod_{i=1}^{P}M_{p_i}^{n_i}} \tag{5.34}$$

$$\theta(\omega) = \arg H_0 + \sum_{i=1}^{Z} m_i\psi_{z_i} - \sum_{i=1}^{P} n_i\psi_{p_i} \tag{5.35}$$

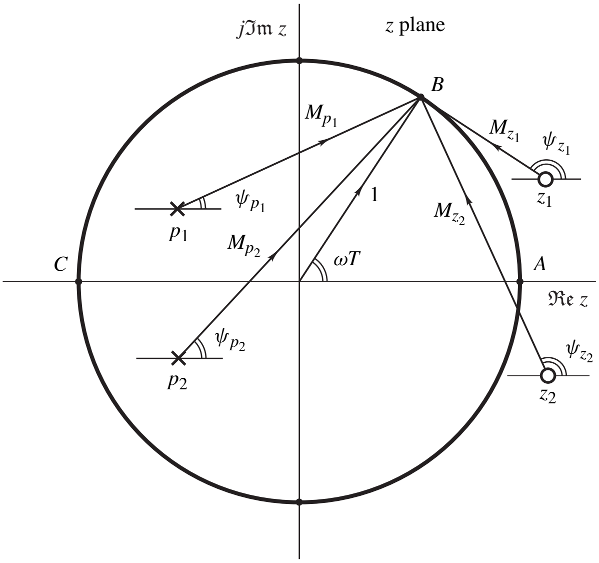


Figure 5.7 Graphical evaluation of frequency response of a discrete-time system.

where arg $H_0 = \pi$ if H_0 is negative. Thus $M(\omega)$ and $\theta(\omega)$ can be determined graphically through the following procedure:

1. Mark the zeros and poles of the system in the z plane.
2. Draw the unit circle $z = 1$.
3. Draw complex number (or vector) $e^{j\omega T}$ where ω is the frequency of interest.
4. Draw m_i complex numbers of the type given by Eq. (5.33a) for each zero of $H(z)$ of order m_i.
5. Draw n_i complex numbers of the type given by Eq. (5.33b) for each pole of order n_i.
6. Measure the magnitudes and angles of the complex numbers in Steps 4 and 5 and use Eqs. (5.34) and (5.35) to calculate the gain $M(\omega)$ and phase shift $\theta(\omega)$, respectively.

The procedure is illustrated in Fig. 5.7 for the case of a second-order discrete-time system with simple zeros and poles. The amplitude and phase responses of a system can be obtained by repeating the above procedure for a number of frequencies in the range of interest.

5.5.3 Periodicity of Frequency Response

Point A in Fig. 5.7 corresponds to zero frequency, point C corresponds to half the sampling frequency, i.e., $\omega_s/2 = \pi/T$, which is often referred to as the *Nyquist frequency*, and one complete revolution of vector $e^{j\omega T}$ about the origin corresponds to an increase in frequency equal to the sampling frequency $\omega_s = 2\pi/T$ rad/s.

If vector $e^{j\omega T}$ in Fig. 5.7 is rotated k complete revolutions, the vector will return to its original position and the values of $M(\omega)$ and $\theta(\omega)$ will obviously remain the same as before. As a result

$$H(e^{j(\omega+k\omega_s)T}) = H(e^{j\omega T})$$

We conclude, therefore, that the *frequency response* is a *periodic function* of frequency with a period ω_s.

5.5.4 Aliasing

The periodicity of the frequency response can be viewed from a different perspective by examining the discrete-time sinusoidal signal given by

$$x(nT) = \sin[(\omega + k\omega_s)nT]$$

Using the appropriate trigonometric identity, we can write

$$\begin{aligned} x(nT) &= \sin \omega nT \cos k\omega_s nT + \cos \omega nT \sin k\omega_s nT \\ &= \sin \omega nT \cos \left(k \cdot \frac{2\pi}{T} \cdot nT \right) + \cos \omega nT \sin \left(k \cdot \frac{2\pi}{T} \cdot nT \right) \\ &= \sin \omega nT \cos 2kn\pi + \cos \omega nT \sin 2kn\pi \\ &= \sin \omega nT \end{aligned}$$

We conclude that discrete-time signals $\sin(\omega + k\omega_s)nT$ and $\sin \omega nT$ are numerically identical for any value of k as illustrated in Fig. 5.8. Consequently, if signal $\sin(\omega + k\omega_s)t$ is sampled at a sampling rate of ω_s, the sampled version of $\sin \omega t$ will be obtained and the frequency of the signal will appear to have changed from $\omega + k\omega_s$ to ω. This effect is known as *aliasing* since frequency $\omega + k\omega_s$ is impersonating frequency ω. Now if the frequency of the sinusoidal input of a discrete-time system is increased from ω to $\omega + k\omega_s$, the system will obviously produce the same output as before since the two input signals will, after all, be numerically identical.

Another facet of aliasing can be explored by considering a sinusoidal signal whose frequency is in the range $(k - \frac{1}{2})\omega_s$ to $k\omega_s$ where k is an integer, say, frequency $k\omega_s - \omega$, where $0 < \omega \leq \omega_s/2$.

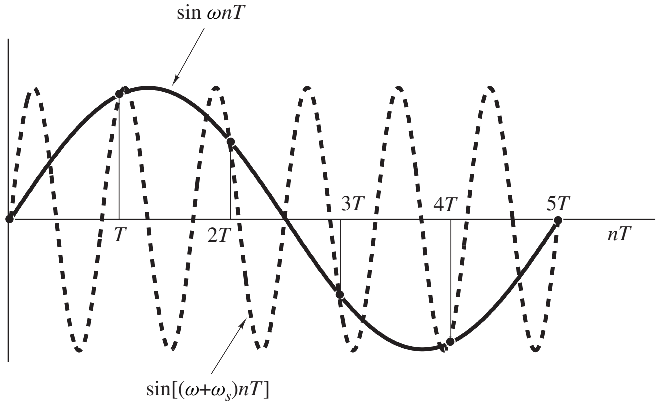


Figure 5.8 Plots of $\sin(\omega nT)$ and $\sin[(\omega + \omega_s)nT]$ versus nT.

In this case, the signal can be expressed as

$$\begin{aligned}
x(nT) &= \sin(k\omega_s - \omega)nT \\
&= \sin k\omega_s nT \cos \omega nT - \cos k\omega_s nT \sin \omega nT \\
&= \sin\left(k \cdot \frac{2\pi}{T} \cdot nT\right) \cos \omega nT - \cos\left(k \cdot \frac{2\pi}{T} \cdot nT\right) \sin \omega nT \\
&= \sin 2kn\pi \cos \omega nT - \cos 2kn\pi \sin \omega nT \\
&= -\sin \omega nT = \sin(-\omega nT)
\end{aligned}$$

Consequently, a positive frequency $k\omega_s - \omega$ in the range $(k - \frac{1}{2})\omega_s$ to $k\omega_s$ will be aliased to the negative frequency $-\omega$.

The above analysis demonstrates that the highest frequency that can be present in a discrete-time sinusoidal signal is $\omega_s/2$. If a continuous-time signal has sinusoidal components whose frequencies exceed $\omega_s/2$, then the frequencies of any such components will be aliased. This can cause some serious problems as will be demonstrated in Chap. 6.

The effects of aliasing can be demonstrated in a different setting that is very familiar to movie fans. As the cowboy wagon accelerates into the sunset, the wheels of the wagon appear to accelerate in the forward direction, then reverse, slow down, stop momentarily, and after that they accelerate again in the forward direction. This series of events happen in the reverse order if the wagon decelerates. Actually, this is exactly what we should see, it is not an illusion, and it has to do with the fact that the image we see on the screen is a series of still photographs which constitute a sampled signal. The phenomenon is easily explained by the illustrations in Fig. 5.9. In this context, the sampling frequency ω_s is the number of film frames per second as taken by the movie camera and the number of wheel revolutions per second defines the frequency of a signal component. Let us examine what happens as the number of wheel revolutions is increased from 0 to $5\omega_s/4$. In Fig. 5.9*a*, the wheel revolves at a speed $\omega_s/4$ and the marker will thus move a quarter revolution before the next frame. The wheel appears to be rotating in the clockwise direction. In Fig. 5.9*b*, the wheel revolves at a speed $\omega_s/2$ and the marker will thus move half a revolution before the next frame. If this speed were maintained, the viewer would have difficulty discerning the direction of rotation since the marker on the wheel would alternate between the top and bottom. In Fig. 5.9*c*, the wheel revolves at a speed $3\omega_s/4$ and the marker will thus move three-quarters of a revolution before the next frame. Miraculously, the wheel will appear to turn in the counterclockwise direction at $\omega_s/4$ revolutions per second. This is analogous to the situation where a frequency of a sinusoidal signal in the range $\omega_s/2$ to ω_s is aliased to a negative frequency. Increasing the rotation speed to, say, $7\omega_s/8$ as in Fig. 5.9*d*, the wheel will appear to rotate slowly in the reverse direction and if the rotation speed is exactly ω_s, the wheel will appear to stop as can be seen in Fig. 5.9*e*, that is, the sampling frequency will appear to behave very much like zero frequency.[3] If the speed of the wheel is increased a bit more, say, to $9\omega_s/8$, then the wheel will appear to move slowly in the forward direction, as depicted in Fig. 5.9*f*,

[3] *This is actually the basis of the stroboscope which is an instrument that can be used to measure the speed in motors and other machinery.*

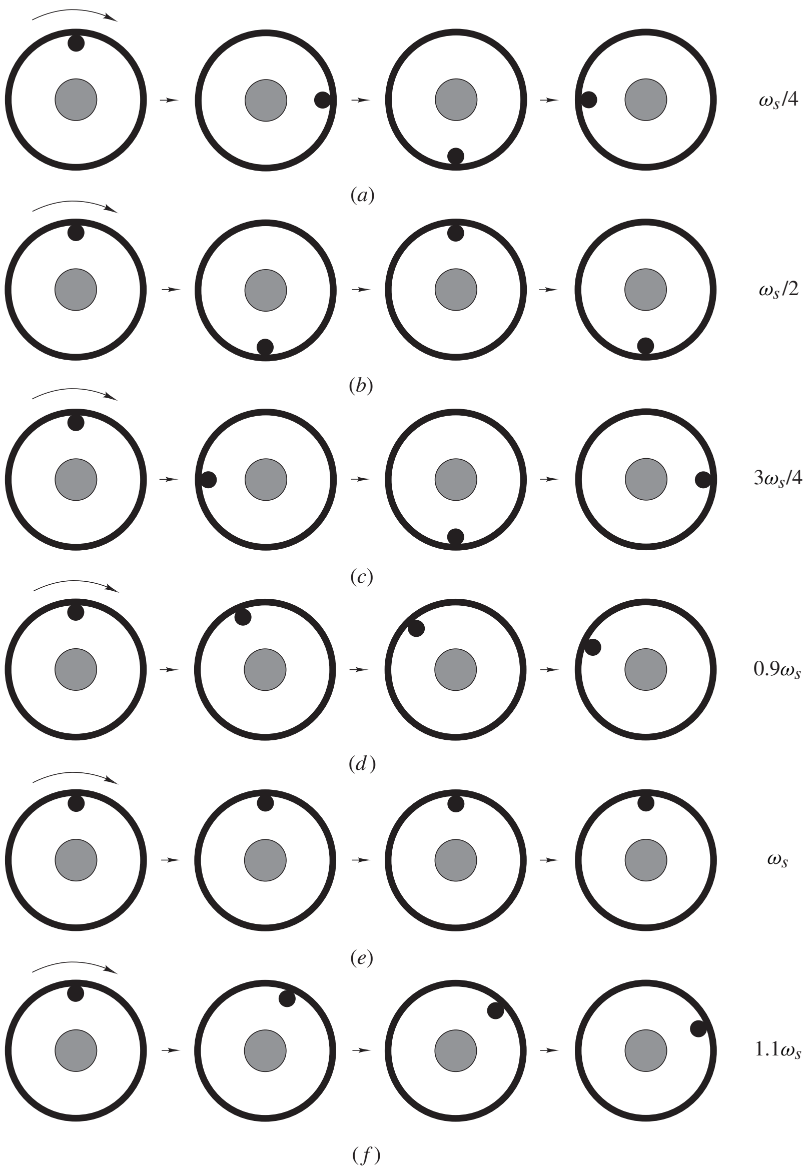


Figure 5.9 Aliasing at the movies.

and at a speed of $5\omega_s/4$ the wheel will appear to rotate at the rate of $\omega_s/4$ revolutions per second as depicted in Fig. 5.9*a*, that is, back to square one. This analogy provides a visual demonstration as to why the signals $\sin(\omega + k\omega_s)nT$ and $\sin \omega nT$ cannot be distinguished, on the one hand, and why the highest frequency in a discrete-time signal cannot exceed $\omega_s/2$.

5.5.5 Frequency Response of Digital Filters

In view of the periodicity of the frequency response, a discrete-time system is completely specified in the frequency domain by its frequency response over the frequency range $-\omega_s/2 \leq \omega \leq \omega_s/2$ which is known as the baseband. In Chap. 1, four types of filters were described, namely, lowpass, highpass, bandpass, and bandstop, depending on the range of frequencies selected or rejected. In discrete-time systems such as digital filters, these terms are applied with respect to the positive half of the baseband, e.g., a highpass filter is one that will select frequencies is some range $\omega_p \leq \omega \leq \omega_s/2$ and reject frequencies in some range $0 \leq \omega \leq \omega_a$, where $\omega_a < \omega_p$.

The magnitude of $H(z)$ is a surface over the z plane. From Eq. (5.4), since $z \to z_i$, $|H(z)| \to 0$ since $(z - z_i) \to 0$. On the other hand, as $z \to p_i$, $|H(z)| \to \infty$ since $(z - p_i) \to 0$. After all, z_i is a zero and p_i is a pole. So if a zero $z_i = r_{z_i} e^{j\phi_{z_i}}$ is located close to the unit circle, then the gain of the system at frequencies close to ϕ_{z_i}/T will be very small. On the other hand, if a pole $p_i = r_{p_i} e^{j\phi_{p_i}}$ is located close to the unit circle, then the gain of the system will be large at frequencies close to ϕ_{p_i}/T. On the basis of these observations, one can easily visualize the amplitude response of a system by simply inspecting its zero-pole plot. If the poles are clustered in the region near the point (1, 0) and the zeros are clustered near the point (−1, 0), then the system is a lowpass filter. More precisely, a system is a lowpass filter if the poles and zeros are enclosed in the sectors $-\Phi_p \leq \phi_{p_i} \leq \Phi_p$ and $-\Phi_z \geq \phi_{z_i} \geq \Phi_z$, respectively, where Φ_z and Φ_p are positive angles such that $\Phi_p < \Phi_z < \pi$. On the basis of these principles, the system represented by the zero-pole plot of Fig. 5.10*a* should be a lowpass filter and this, indeed, is the case as can be seen in the 3-D plot of Fig. 5.10*b*.

The angle of $H(z)$ is also a surface over the z plane but, unfortunately, this is usually far too convoluted to be correlated to the zero-pole plot. See, for example, the 3-D plot of Fig. 5.10*c*, which represents the angle of $H(z)$ for the lowpass filter under consideration.

The amplitude and phase responses could be displayed in terms of 3-D plots, as depicted in Fig. 5.10*d* and *e*, by evaluating the magnitude and angle of $H(z)$ on the unit circle, i.e., by letting $z = e^{j\omega T}$. If the surfaces in Fig. 5.10*b* and *c* were deemed to represent solid objects, say, made of wax, then the amplitude and phase responses would be the profiles of the cores punched through these objects by a cylindrical corer tool of radius 1. Three-dimensional plots such as these are both difficult to plot as well as visualize, particularly, the one for the phase response. For these reasons, the amplitude and phase responses are usually plotted in terms of 2-D plots of $20 \log M(\omega)$ and $\theta(\omega)$, respectively, as illustrated in Fig. 5.10*f* and *g*. To continue the geometrical interpretation, these 2-D plots can be obtained by spreading ink over the surfaces of the wax cores obtained before and then rolling them over a white sheet of paper.

It should be mentioned here that ambiguities can arise in the evaluation of the phase response owing to the fact that $\theta = \tan^{-1} \mu$ is a multivalued function of μ (see Sec. A.3.7). Typically, one would evaluate the phase response of a system by finding the real and imaginary parts of the frequency response, i.e.,

$$H(e^{j\omega T}) = \Re e\, H(e^{j\omega T}) + j\, \Im m\, H(e^{j\omega T})$$

and then compute the phase response as

$$\theta(\omega) = \tan^{-1} \frac{\Im m\, H(e^{j\omega T})}{\Re e\, H(e^{j\omega T})}$$

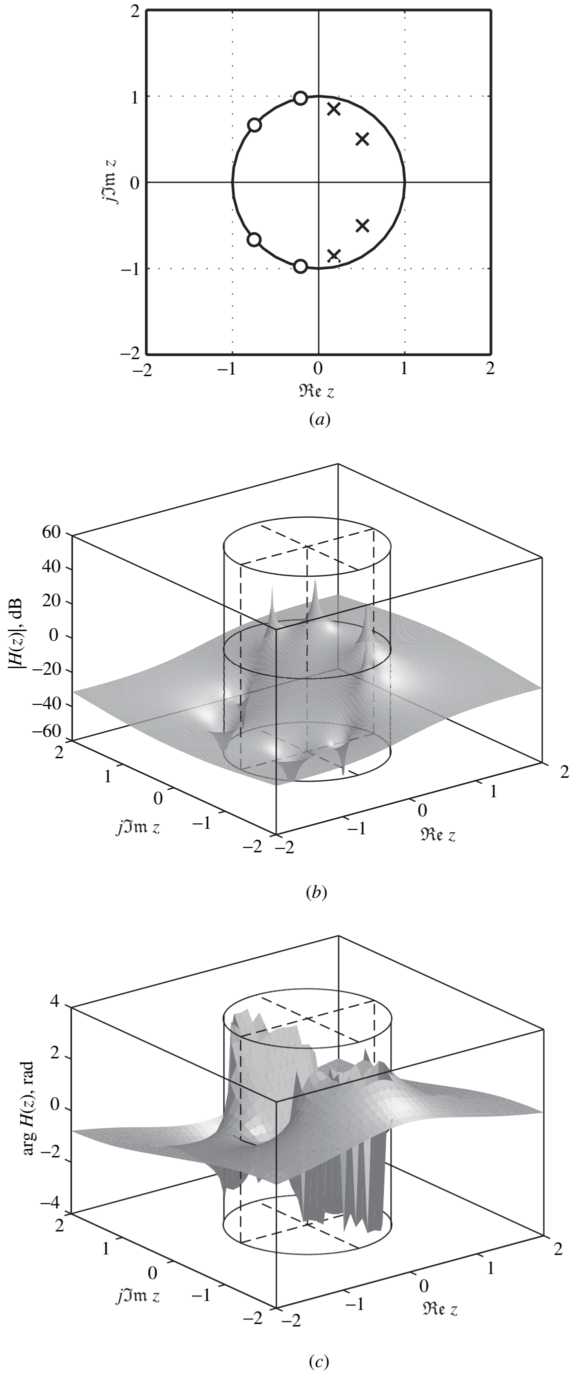
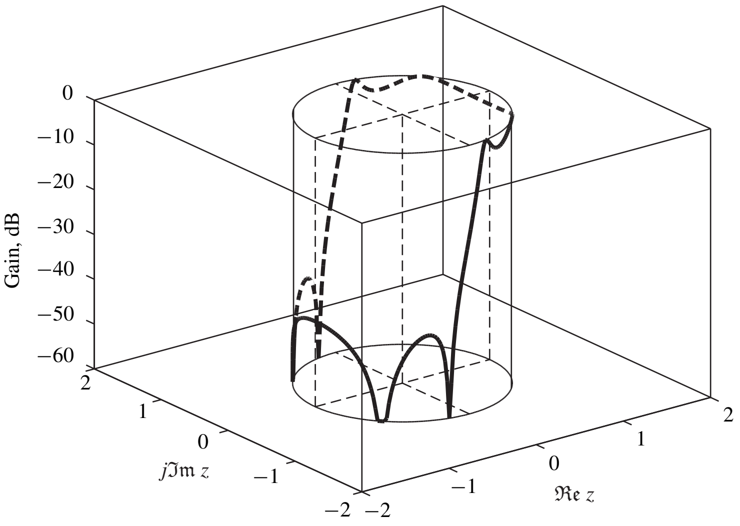
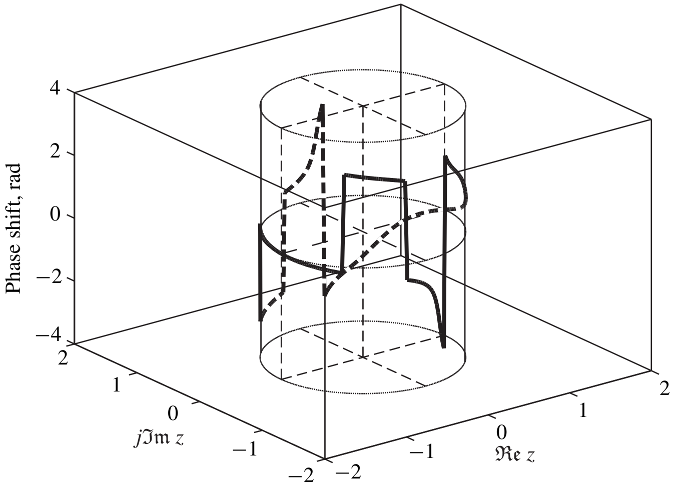


Figure 5.10 Frequency response of lowpass filter: (*a*) Zero-pole plot, (*b*) plot of $20 \log |H(z)|$ versus $z = \Re e\, z + j\, \Im m\, z$, (*c*) plot of arg $H(z)$ versus z.



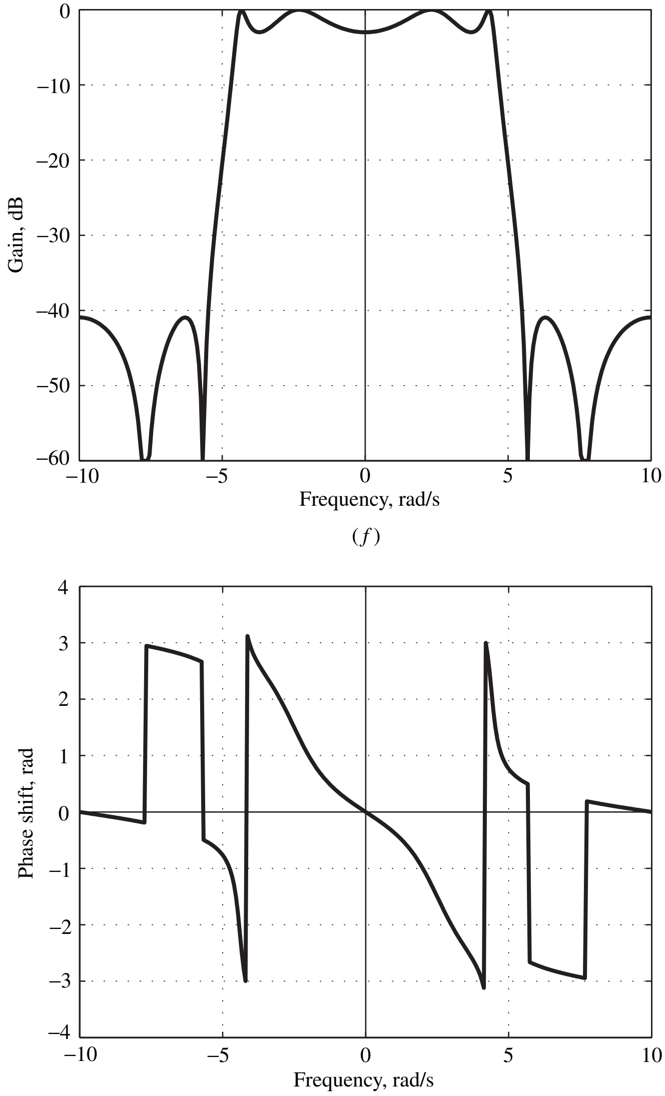
(*d*)



(*e*)

Figure 5.10 Cont'd Frequency response of lowpass filter: (*d*) Plot of $20 \log |H(e^{j\omega T})|$ versus z, (*e*) plot of $\arg H(e^{j\omega T})$ versus z.

Things would work out perfectly if $-\pi < \theta(\omega) < \pi$. However, if the value of $\theta(\omega)$ is outside this range, the phase response computed by the typical DSP software, including MATLAB, would be wrong. The phase response of causal systems is a decreasing function of frequency because of certain physical reasons to be explained shortly and at some frequency it will decrease below $-\pi$. When this happens, the typical DSP software will yield a positive angle in the range 0 to π instead of the correct negative value, i.e., an angle $\pi - \epsilon$ will be computed instead of the correct angle of $-\pi - \epsilon$; thus an abrupt discontinuity of $+2\pi$ will be introduced as an artifact. This problem can be corrected by

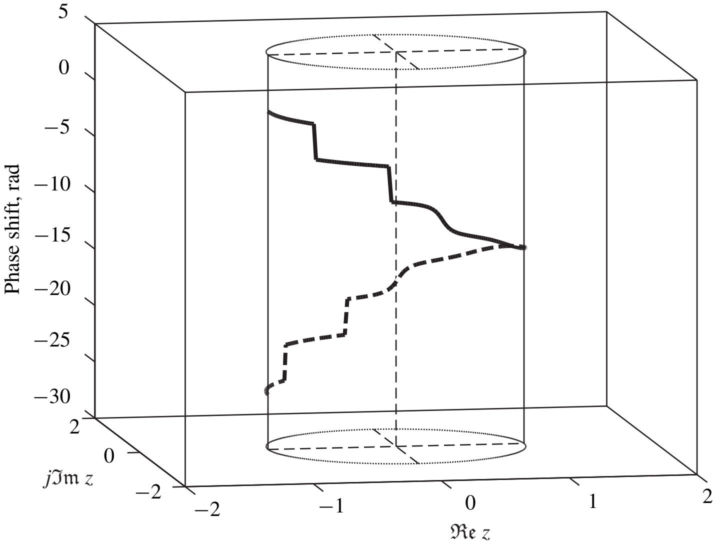


(*f*)

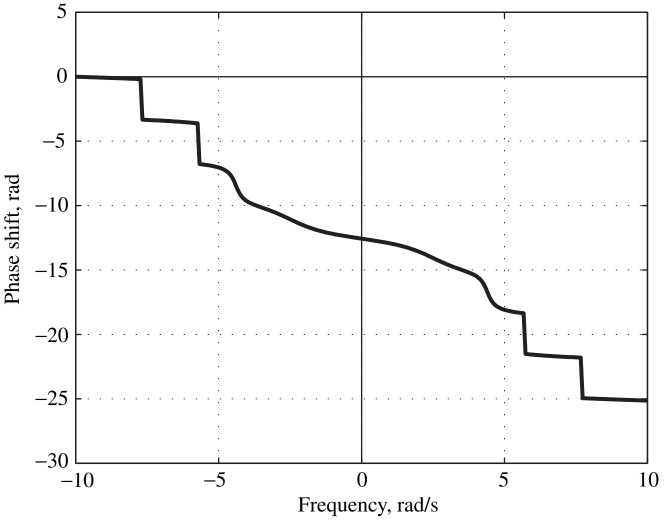
(*g*)

Figure 5.10 Cont'd Frequency response of lowpass filter: (*f*) Plot of $20 \log M(\omega)$ versus ω, (*g*) plot of $\theta(\omega)$ versus ω.

monitoring the change in the phase response as the frequency is increased and whenever a sign change is observed in the phase response from a negative to a positive value, which corresponds to a crossing of the negative real axis, to subtract an angle of 2π from the phase response at that frequency as well as all the subsequent frequencies (see Sec. A.3.7). This problem is quite apparent in the 3-D and 2-D plots of Fig. 5.10*e* and *g*, which were computed with MATLAB using function atan2. The corrected phase responses are depicted in Fig. 5.10*h* and *i*. Incidentally, the phase response continues to have discontinuities after correction but these are legitimate. They are caused by the zeros in Fig. 5.10*a*.



(*h*)



(*i*)

Figure 5.10 Cont'd Frequency response of lowpass filter: (h) Corrected plot of arg $H(e^{j\omega T})$ versus z, (i) corrected plot of $\theta(\omega)$ versus ω.

Example 5.10 The discrete-time system shown in Fig. 5.11 is a nonrecursive filter. The multiplier constants are

$$A_0 = 0.3352 \qquad A_1 = 0.2540 \qquad A_2 = 0.0784$$

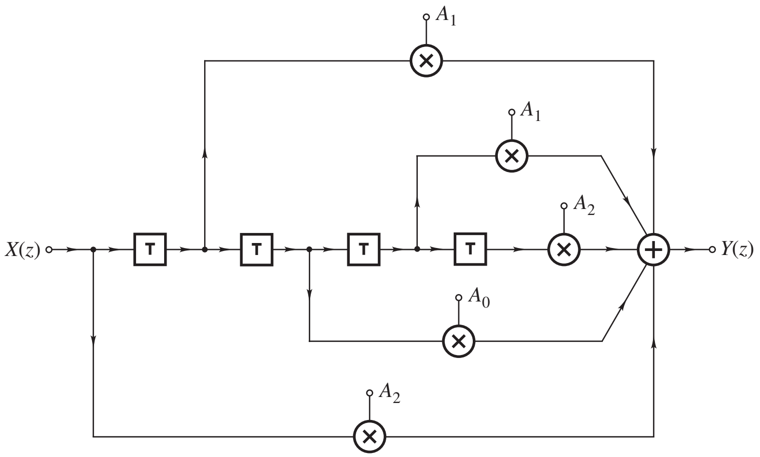


Figure 5.11 Fourth-order, nonrecursive filter (Example 5.10).

and the sampling frequency is $\omega_s = 20$ rad/s. (*a*) Construct the zero-pole plot of the filter. (*b*) Plot the surface $|H(z)|$ as a function of $z = \Re e\, z + j\, \Im m\, z$. (*c*) Obtain expressions for the amplitude and phase responses. (*d*) Plot the amplitude and phase responses first in terms of 3-D plots and then in terms of 2-D plots.

■ **Solution**

(*a*) The transfer function of the filter can be readily obtained by inspection as

$$H(z) = A_2 + A_1 z^{-1} + A_0 z^{-2} + A_1 z^{-3} + A_2 z^{-4} \tag{5.36a}$$

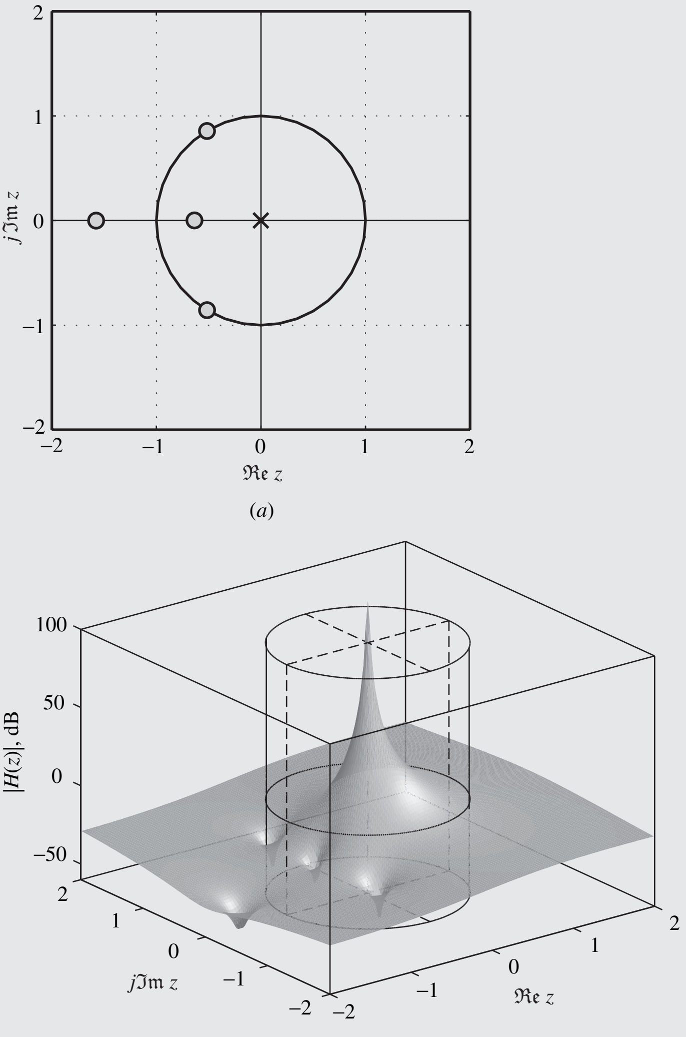
$$= \frac{A_2 z^2 + A_1 z + A_0 + A_1 z^{-1} + A_2 z^{-2}}{z^2} \tag{5.36b}$$

$$= \frac{A_2 z^4 + A_1 z^3 + A_0 z^2 + A_1 z + A_2}{z^4} \tag{5.36c}$$

From Eq. (5.36c), we note that the filter has four zeros and a fourth-order pole at the origin. Using MATLAB, the zeros can be obtained as

$$z_1 = -1.5756 \qquad z_2 = -0.6347 \qquad z_3, z_4 = -0.5148 \pm j0.8573$$

Hence the zero-pole plot of Fig. 5.12*a* can be obtained. We note that the high-order pole at the origin tends to create high gain at low frequencies, whereas the zeros tend to produce low gain at high frequencies. Thus, the system must be a lowpass filter. ■



(*a*)

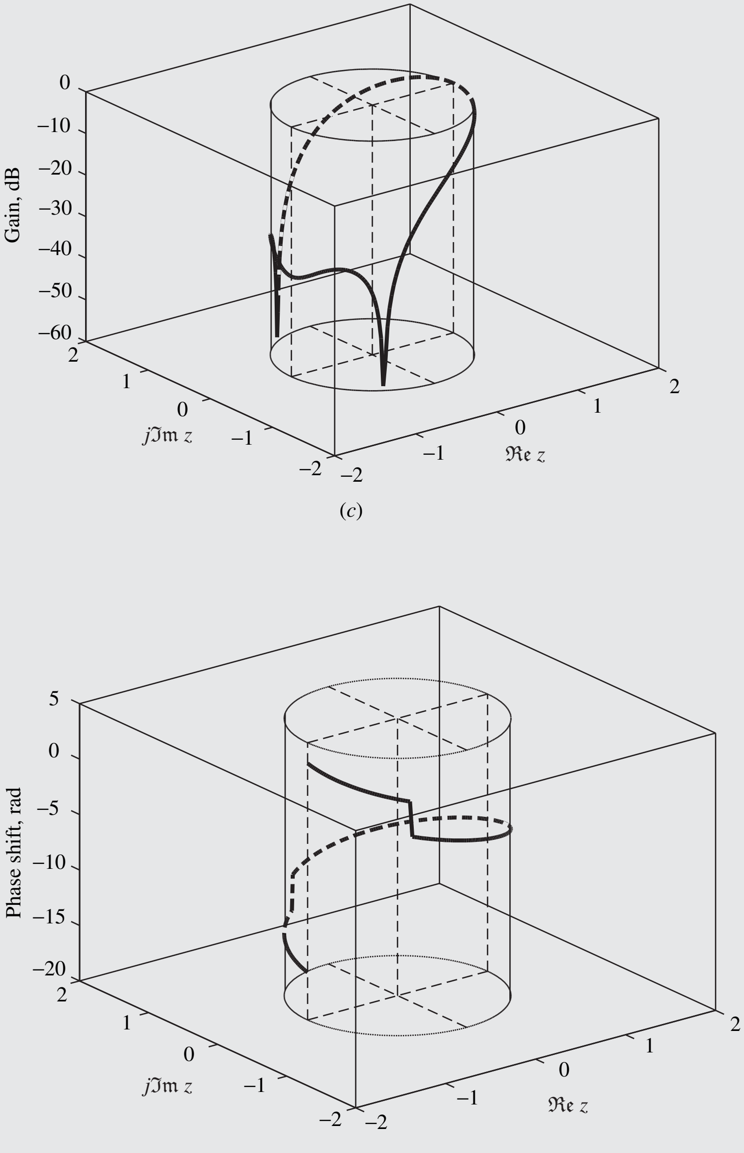
(*b*)

Figure 5.12 Frequency response of lowpass filter (Example 5.10): (*a*) Zero-pole plot, (*b*) plot of $20 \log |H(z)|$ versus $z = \Re e\, z + j\, \Im m\, z$.

(*b*) The 3-D plot of $20 \log |H(z)|$ versus z is shown in Fig. 5.12*b*.

(*c*) From Eq. (5.36b), we have

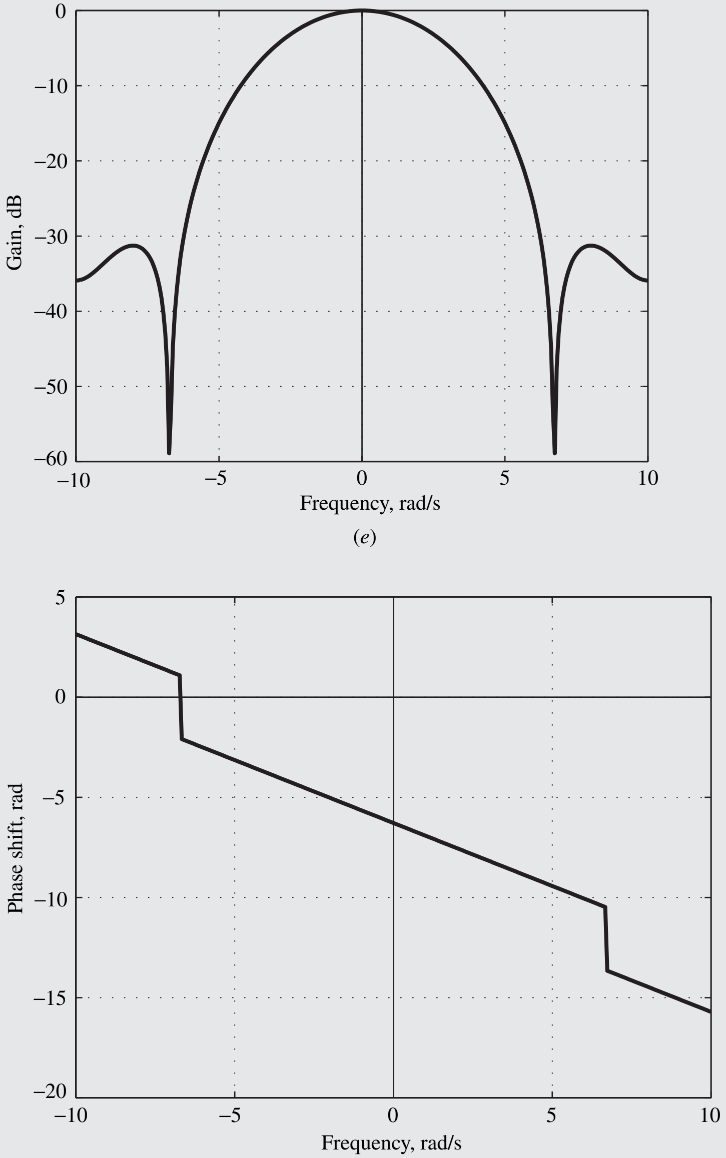
$$\begin{aligned} H(e^{j\omega T}) &= \frac{A_2(e^{j2\omega T} + e^{-j2\omega T}) + A_1(e^{j\omega T} + e^{-j\omega T}) + A_0}{e^{j2\omega T}} \\ &= \frac{2A_2 \cos 2\omega T + 2A_1 \cos \omega T + A_0}{e^{j2\omega T}} \end{aligned}$$



(*c*)

(*d*)

Figure 5.12 Cont'd Frequency response of lowpass filter (Example 5.10): (*c*) Plot of $20 \log |H(e^{j\omega T})|$ versus z, (*d*) corrected plot of $\arg H(e^{j\omega T})$ versus z .



(*e*)

(*f*)

Figure 5.12 Cont'd Frequency response of lowpass filter (Example 5.10): (*e*) Plot of $20 \log M(\omega)$ versus ω, (*f*) corrected plot of $\theta(\omega)$ versus ω.

and so

$$M(\omega) = |2A_2 \cos 2\omega T + 2A_1 \cos \omega T + A_0| \quad \blacksquare$$

$$\theta(\omega) = \theta_N - 2\omega T \quad \blacksquare$$

where

$$\theta_N = \begin{cases} 0 & \text{if } 2A_2 \cos 2\omega T + 2A_1 \cos \omega T + A_0 \geq 0 \\ \pi & \text{otherwise} \end{cases}$$

(*d*) The amplitude and phase responses are depicted in Fig. 5.12*c* and *d* as 3-D plots and in Fig. 5.12*e* and *f* as 2-D plots. ■

An interesting property of nonrecursive filters is that they can have a linear phase response, as can be seen in Fig. 5.12*f*. This is an important feature that makes nonrecursive filters attractive in a number of applications.

Example 5.11 A recursive digital filter is characterized by the transfer function

$$H(z) = H_0 \prod_{i=1}^{3} H_i(z)$$

where

$$H_i(z) = \frac{a_{0i} + a_{1i}z + z^2}{b_{0i} + b_{1i}z + z^2}$$

and the numerical values of the coefficients are given in Table 5.4. The sampling frequency is 20 rad/s. (*a*) Construct the zero-pole plot of the filter. (*b*) Plot the surface $|H(z)|$ as a function of $z = \Re e\, z + j\, \Im m\, z$. (*c*) Obtain expressions for the amplitude and phase responses. (*d*) Plot the amplitude and phase responses first in terms of 3-D plots and then in terms of 2-D plots.

Table 5.4 Transfer-function coefficients for Example 5.11

i	a_{0i}	a_{1i}	b_{0i}	b_{1i}
1	−1.0	0.0	8.131800E−1	7.870090E−8
2	1.0	−1.275258	9.211099E−1	5.484026E−1
3	1.0	1.275258	9.211097E−1	−5.484024E−1
	$H_0 = 1.763161E - 2$			

■ **Solution**

(a) The zeros and poles of the transfer function can be readily obtained as

$$z_1, z_2 = \pm 1 \qquad z_3, z_4 = 0.6376 \pm j0.7703$$
$$z_5, z_6 = -0.6376 \pm j0.7703$$

and

$$p_1, p_2 = \pm j0.9018 \qquad p_3, p_4 = 0.2742 \pm j0.7703$$
$$p_5, p_6 = -0.2742 \pm j0.7703$$

respectively. Hence the zero-pole plot depicted in Fig. 5.13a can be readily constructed. Since there is a cluster of poles close to the unit circle at $\omega T \approx \pi/2$ and zeros at $(1, 0)$ and $(-1, 0)$, the recursive filter must be a bandpass filter which will select frequencies closed to $\omega = \pi/2T$. ■

(b) The 3-D plot of $20 \log |H(z)|$ versus z depicted in Fig. 5.13b demonstrates clearly that this is a bandpass filter. ■

(c) The frequency response of the filter can be obtained as

$$H(z)\,|_{z\rightarrow e^{j\omega T}} = H(e^{j\omega T}) = M(\omega)e^{j\theta(\omega)}$$

with

$$M(\omega) = |H_0| \prod_{i=1}^{3} \left|H_i(e^{j\omega T})\right| = |H_0| \prod_{i=1}^{3} M_i(\omega)$$

and

$$\theta(\omega) = \arg H_0 + \sum_{i=1}^{3} \arg H_i(e^{j\omega T}) = \sum_{i=1}^{3} \theta_i(\omega)$$

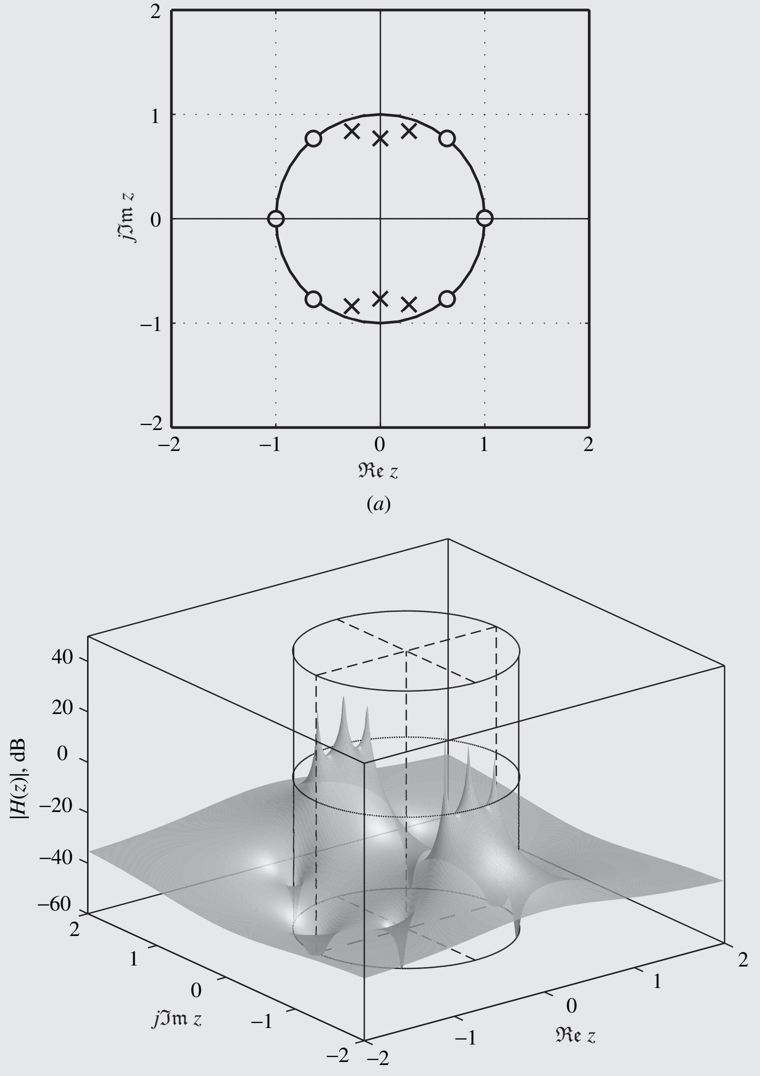
where

$$M_i(\omega) = \left|H_i(e^{j\omega T})\right| = \left|\frac{a_{0i} + a_{1i}e^{j\omega T} + e^{j2\omega T}}{b_{0i} + b_{1i}e^{j\omega T} + e^{j2\omega T}}\right|$$

$$= \left|\frac{(a_{0i} + a_{1i}\cos\omega T + \cos 2\omega T) + j(a_{1i}\sin\omega T + \sin 2\omega T)}{(b_{0i} + b_{1i}\cos\omega T + \cos 2\omega T) + j(b_{1i}\sin\omega T + \sin 2\omega T)}\right|$$

$$= \left[\frac{(a_{0i} + a_{1i}\cos\omega T + \cos 2\omega T)^2 + (a_{1i}\sin\omega T + \sin 2\omega T)^2}{(b_{0i} + b_{1i}\cos\omega T + \cos 2\omega T)^2 + (b_{1i}\sin\omega T + \sin 2\omega T)^2}\right]^{\frac{1}{2}}$$

$$= \left[\frac{1 + a_{0i}^2 + a_{1i}^2 + 2(1 + a_{0i})a_{1i}\cos\omega T + 2a_{0i}\cos 2\omega T}{1 + b_{0i}^2 + b_{1i}^2 + 2(1 + b_{0i})b_{1i}\cos\omega T + 2b_{0i}\cos 2\omega T}\right]^{\frac{1}{2}}$$ ■



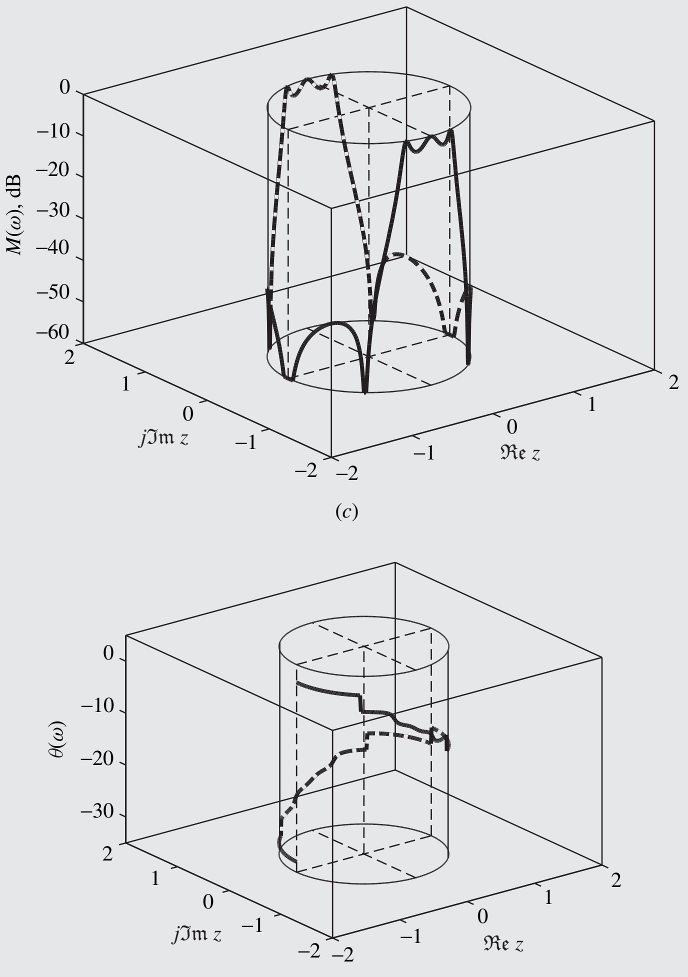
(*a*)

(*b*)

Figure 5.13 Frequency response of bandpass filter (Example 5.11): (*a*) Zero-pole plot, (*b*) plot of $20 \log |H(z)|$ versus $z = \mathfrak{Re}\, z + j\, \mathfrak{Im}\, z$.

and

$$\begin{aligned}\theta_i(\omega) &= \arg H_i(e^{j\omega T}) \\ &= \arg \frac{a_{0i} + a_{1i}e^{j\omega T} + e^{j2\omega T}}{b_{0i} + b_{1i}e^{j\omega T} + e^{j2\omega T}} \\ &= \arg \frac{(a_{0i} + a_{1i}\cos\omega T + \cos 2\omega T) + j(a_{1i}\sin\omega T + \sin 2\omega T)}{(b_{0i} + b_{1i}\cos\omega T + \cos 2\omega T) + j(b_{1i}\sin\omega T + \sin 2\omega T)}\end{aligned}$$



(*c*)

(*d*)

Figure 5.13 Cont'd Frequency response of bandpass filter (Example 5.11): (*c*) Plot of $20 \log |H(e^{j\omega T})|$ versus z, (*d*) corrected plot of arg $|H(e^{j\omega T})|$ versus z.

$$= \tan^{-1} \frac{a_{1i} \sin \omega T + \sin 2\omega T}{a_{0i} + a_{1i} \cos \omega T + \cos 2\omega T}$$

$$- \tan^{-1} \frac{b_{1i} \sin \omega T + \sin 2\omega T}{b_{0i} + b_{1i} \cos \omega T + \cos 2\omega T} \quad \blacksquare$$

The 3-D plots for the amplitude and phase responses are depicted in Fig. 5.13*c* and *d* and the corresponding 2-D plots can be readily obtained from the above expressions as shown in Fig. 5.13*e* and *f*. As can be seen from these plots, the system being analyzed is definitely a bandpass filter. ■

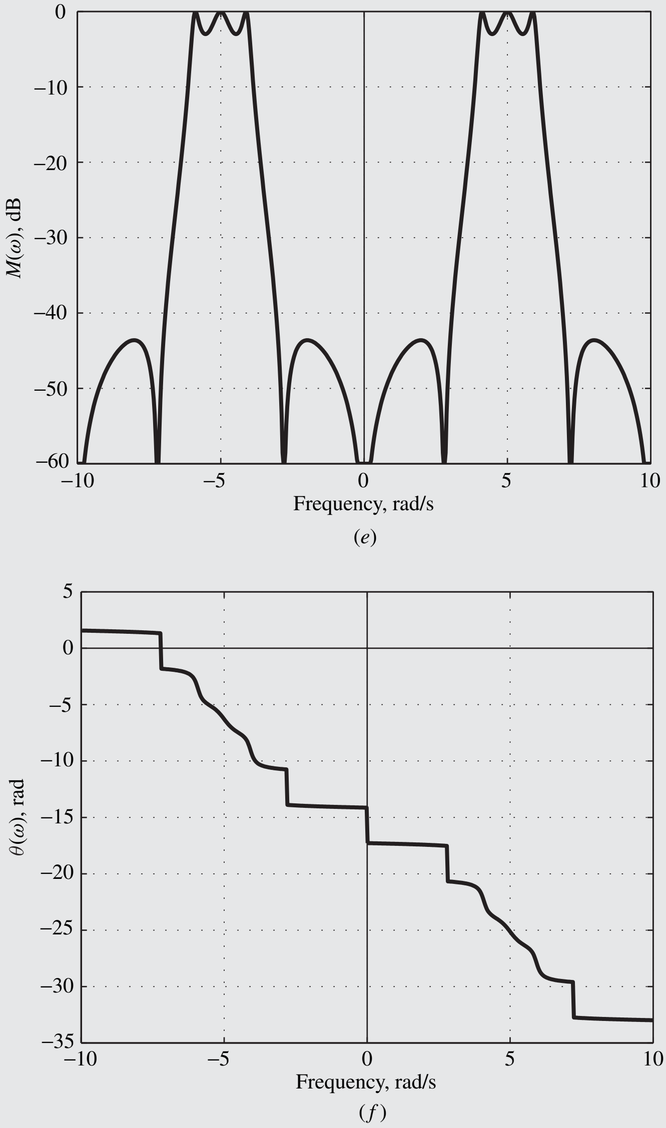


Figure 5.13 Cont'd Frequency response of bandpass filter (Example 5.11): (*e*) Plot of $20 \log M(\omega)$ versus ω, (*f*) corrected plot of $\theta(\omega)$ versus ω.

5.6 TRANSFER FUNCTIONS FOR DIGITAL FILTERS

In the previous section, we have demonstrated that the filtering action of a discrete-time system depends critically on the patterns formed by the zeros and poles of the transfer function in the *z* plane. In this section, we show that a set of standard low-order transfer functions can be derived through the judicious choice of the zero/pole locations.

5.6.1 First-Order Transfer Function

A first-order transfer function can only have a real zero and a real pole, i.e. it must be of the form

$$H(z) = \frac{z - z_0}{z - p_0}$$

and to ensure that the system is stable, the pole must satisfy the condition $-1 < p_0 < 1$. The zero can be anywhere on the real axis of the z plane. If the pole is close to point $(1, 0)$ and the zero is close to or at point $(-1, 0)$, then we have a lowpass filter; if the zero and pole positions are interchanged, then we get a highpass filter.

Certain applications call for discrete-time systems that have a constant amplitude response and a varying phase response. Such systems can be constructed by using *allpass transfer functions*. A first-order allpass transfer function is of the form

$$H(z) = \frac{p_0 z - 1}{z - p_0} = p_0 \frac{z - 1/p_0}{z - p_0}$$

where the zero is the reciprocal of the pole. The frequency response of a system characterized by $H(z)$ is given by

$$H(e^{j\omega T}) = \frac{p_0 e^{j\omega T} - 1}{e^{j\omega T} - p_0} = \frac{p_0 \cos\omega T + jp_0 \sin\omega T - 1}{\cos\omega T + j\sin\omega T - p_0}$$

and hence the amplitude and phase responses can be obtained as

$$\begin{aligned} M(\omega) &= \left| \frac{p_0 \cos\omega T - 1 + jp_0 \sin\omega T}{\cos\omega T - p_0 + j\sin\omega T} \right| \\ &= \left[\frac{(p_0 \cos\omega T - 1)^2 + (p_0 \sin\omega T)^2}{(\cos\omega T - p_0)^2 + (\sin\omega T)^2} \right]^{\frac{1}{2}} = 1 \end{aligned}$$

and

$$\theta(\omega) = \tan^{-1} \frac{p_0 \sin\omega T}{p_0 \cos\omega T - 1} - \tan^{-1} \frac{\sin\omega T}{\cos\omega T - p_0}$$

respectively.

5.6.2 Second-Order Transfer Functions

LOWPASS TRANSFER FUNCTION. As was shown earlier, a system whose poles and zeros are located in the sectors $-\Phi_p \le \phi_{p_i} \le \Phi_p$ and $-\Phi_z \ge \phi_{z_i} \ge \Phi_z$, respectively, where Φ_p and Φ_z are positive angles such that $\Phi_z > \Phi_p$ is a lowpass filter. Hence a lowpass second-order transfer function can be constructed by placing a complex-conjugate pair of poles anywhere inside the unit circle and a pair of zeros at the Nyquist point, as shown in Fig. 5.14*a*. Such a transfer function can

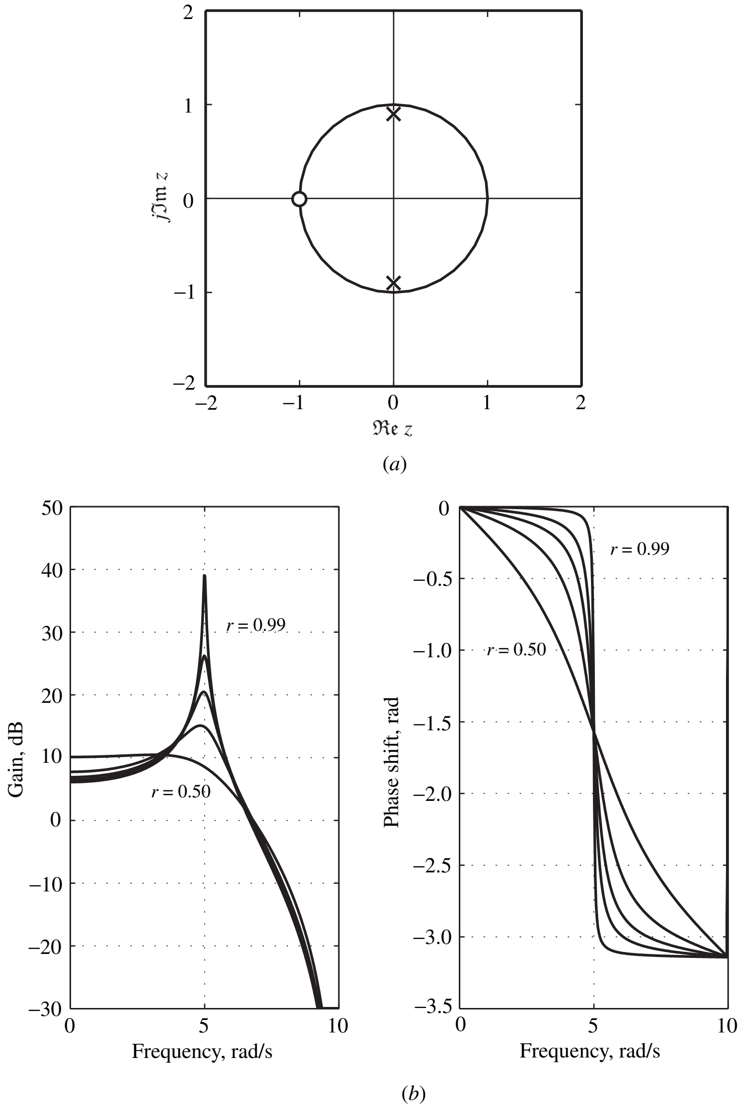


Figure 5.14 Frequency response of second-order lowpass filter: (*a*) Zero-pole plot, (*b*) amplitude and phase responses.

be constructed as

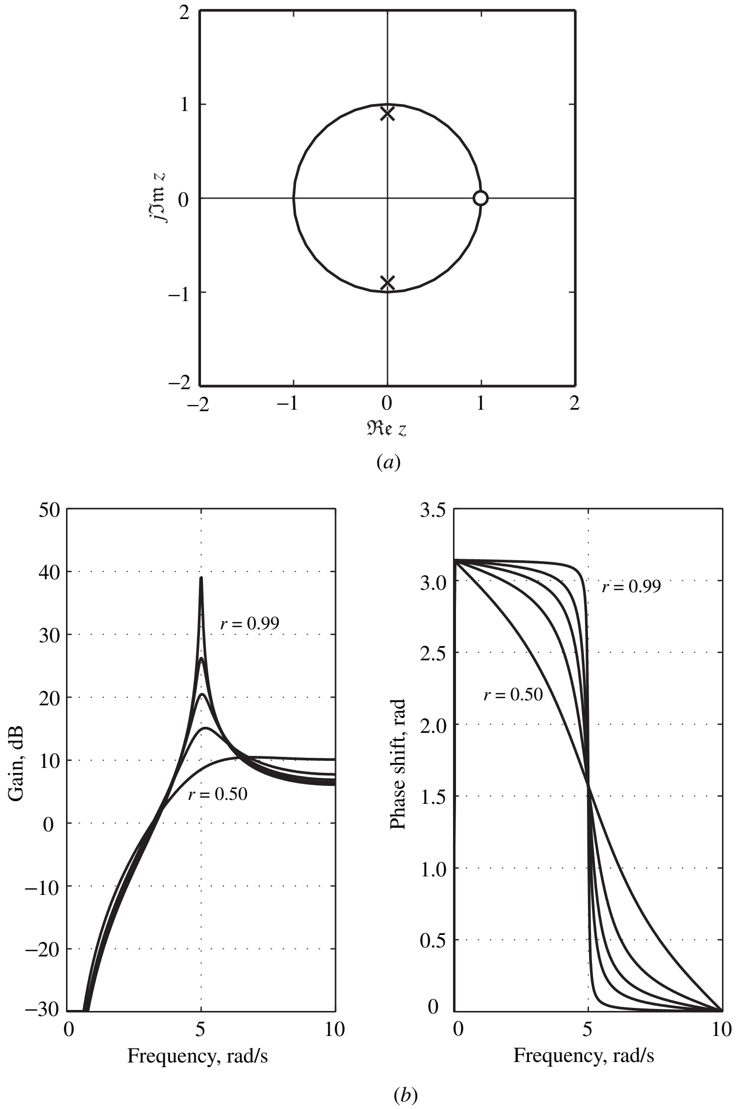
$$H_{LP}(z) = \frac{(z+1)^2}{(z - re^{j\phi})(z - re^{-j\phi})} = \frac{z^2 + 2z + 1}{z^2 - 2r(\cos\phi)z + r^2} \tag{5.37}$$

where $0 < r < 1$. As the poles move closer to the unit circle, the amplitude response develops a peak at frequency $\omega = \phi/T$ while the slope of the phase response tends to become steeper and steeper at that frequency, as illustrated in Fig. 5.14*b*.

HIGHPASS TRANSFER FUNCTION. If the zeros and poles of a system are located in the sectors $-\Phi_z \leq \phi_{z_i} \leq \Phi_z$ and $-\Phi_p \geq \phi_{p_i} \geq \Phi_p$, where Φ_z and Φ_p are positive angles such that $\Phi_p > \Phi_z$, then the system is a highpass filter. A highpass transfer function can be readily obtained form Eq. (5.37) by simply moving the zeros from point $(-1, 0)$ to $(1, 0)$ as in Fig. 5.15*a*, that is,

$$H_{HP}(z) = \frac{(z-1)^2}{z^2 - 2r(\cos\phi)z + r^2} = \frac{(z^2 - 2z + 1)}{z^2 - 2r(\cos\phi)z + r^2} \tag{5.38}$$

The amplitude and phase responses obtained are shown in Fig. 5.15*b*.



(*a*)

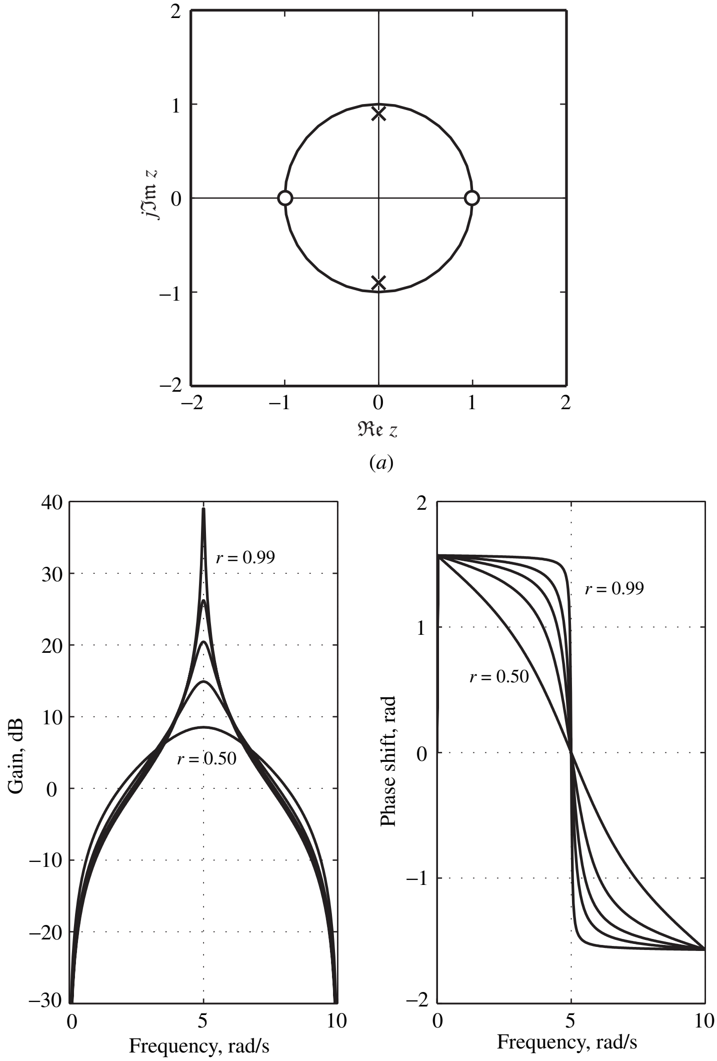
(*b*)

Figure 5.15 Frequency response of second-order highpass filter: (*a*) Zero-pole plot, (*b*) amplitude and phase responses.

BANDPASS TRANSFER FUNCTION. In a bandpass system, a cluster of poles is sandwiched between clusters of zeros in the neighborhoods of points (1, 0) and (−1, 0). A second-order bandpass transfer function can be obtained from the lowpass transfer function of Eq. (5.37) by moving one zero from point (−1, 0) to (1, 0), as shown in Fig. 5.16*a*. The transfer function assumes the form

$$H_{BP}(z) = \frac{z^2 - 1}{z^2 - 2r(\cos\phi)z + r^2} \tag{5.39}$$

and some typical amplitude and phase responses are shown in Fig. 5.16*b*.



(*a*)

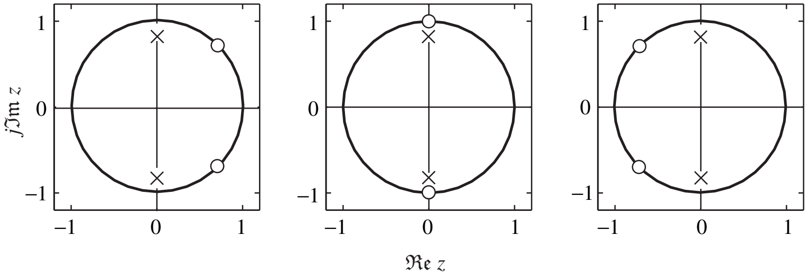
(*b*)

Figure 5.16 Frequency response of second-order bandpass filter: (*a*) Zero-pole plot, (*b*) amplitude and phase responses.

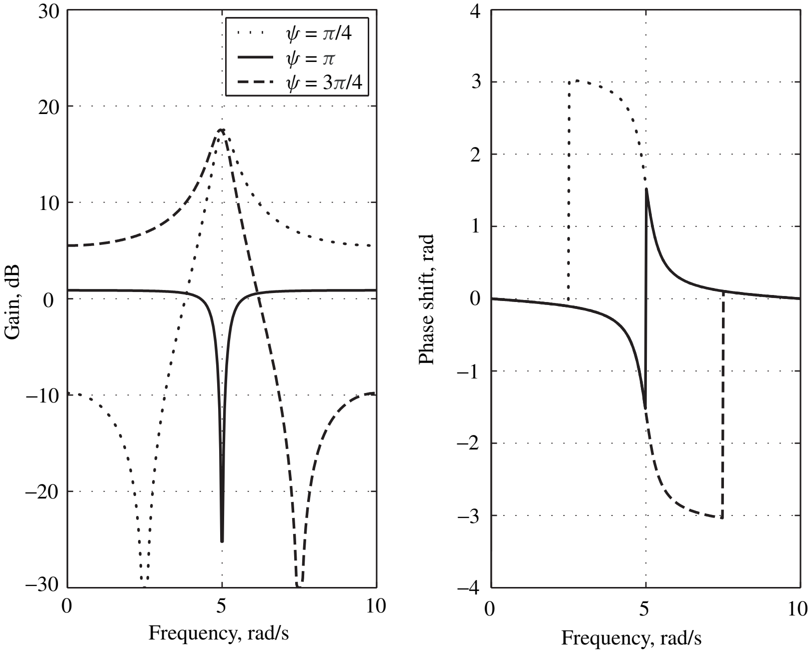
NOTCH TRANSFER FUNCTION. A notch system is one that has a notch in its amplitude response, as may be expected, and such a response can be achieved by placing a complex-conjugate pair of zeros on the unit circle, as illustrated in Fig. 5.17*a*. The transfer function of such a system assumes the form

$$H_N(z) = \frac{z^2 - 2(\cos\psi)z + 1}{z^2 - 2r(\cos\phi)z + r^2} \tag{5.40}$$

and as can be seen in Fig. 5.17*b* three types of behavior can be achieved depending on the relative location of the zeros in relation to the poles. If $\psi > \phi$, then a lowpass notch filter is obtained and if $\phi > \psi$, then a highpass notch is the outcome. The case, $\phi = \psi$ will yield a filter that will reject frequencies in the neighborhood of $\omega = \phi/T$, and such a filter is usually referred to as a *bandstop filter*.



(*a*)



(*b*)

Figure 5.17 Frequency response of second-order notch filter ($\phi = \pi/2$): (*a*) Zero-pole plots, (*b*) amplitude and phase responses.

ALLPASS TRANSFER FUNCTION. An Nth-order allpass transfer function with a denominator polynomial $b_0 + b_1 z + \cdots + b_{N-1} z^{N-1} + b_N z^N$ can be obtained by constructing a corresponding numerator polynomial $b_N + b_{N-1} z + \cdots + b_1 z^{N-1} + b_0 z^N$ by simply reversing the order of the coefficients. Hence a second-order allpass transfer function can be obtained as

$$H_{AP}(z) = \frac{r^2 z^2 - 2(\cos\phi)z + 1}{z^2 - 2r(\cos\phi)z + r^2} \tag{5.41}$$

As in the first-order allpass transfer function, the zeros of the second-order allpass transfer function are the reciprocals of the poles (see Prob. 5.31). To demonstrate that this is indeed an allpass transfer function, we note that

$$\begin{aligned} M_{AP}(\omega) = |H_{AP}(e^{j\omega T})| &= \left[H_{AP}(e^{j\omega T}) \cdot H_{AP}^*(e^{j\omega T})\right]^{\frac{1}{2}} \\ &= \left[H_{AP}(e^{j\omega T}) \cdot H_{AP}(e^{-j\omega T})\right]^{\frac{1}{2}} \end{aligned}$$

and hence

$$\begin{aligned} M_{AP}(\omega) &= \left\{ \left[H_{AP}(z) \cdot H_{AP}(z^{-1})\right]_{z=e^{j\omega T}} \right\}^{\frac{1}{2}} \\ &= \left\{ \left[\frac{r^2 z^2 + 2(\cos\phi)z + 1}{z^2 + 2r(\cos\phi)z + r^2} \cdot \frac{r^2 z^{-2} + 2(\cos\phi)z^{-1} + 1}{z^{-2} + 2r(\cos\phi)z^{-1} + r^2}\right]_{z=e^{j\omega T}} \right\}^{\frac{1}{2}} \\ &= \left\{ \left[\frac{r^2 z^2 + 2(\cos\phi)z + 1}{z^2 + 2r(\cos\phi)z + r^2} \cdot \frac{r^2 + 2(\cos\phi)z + z^2}{1 + 2r(\cos\phi)z + z^2 r^2}\right]_{z=e^{j\omega T}} \right\}^{\frac{1}{2}} = 1 \end{aligned}$$

As in the first-order allpass transfer function, the zeros of a second-order (also an Nth-order) transfer function are the reciprocals of corresponding poles.

As will be shown in the next section, a nonlinear phase response in a filter would lead to phase distortion which is undesirable in certain applications. Some of the design methods for recursive filters to be explored later on in Chap. 11 tend to yield filters with nonlinear phase responses. The phase responses of these filters can be linearized through the use of allpass systems known as *delay equalizers* (see Sec. 16.8).

5.6.3 Higher-Order Transfer Functions

Higher-order transfer functions can be obtained by forming products or sums of first- and/or second-order transfer functions. Methods for obtaining transfer functions that will yield specified frequency responses will be explored in later chapters.

5.7 AMPLITUDE AND DELAY DISTORTION

In practice, a discrete-time system can distort the information content of a signal to be processed as will now be demonstrated.

Consider an application where a digital filter characterized by a transfer function $H(z)$ is to be used to select a specific signal $x_k(nT)$ from a sum of signals

$$x(nT) = \sum_{i=1}^{m} x_i(nT)$$

Let the amplitude and phase responses of the filter be $M(\omega)$ and $\theta(\omega)$, respectively. Two parameters associated with the phase response are the *absolute delay* $\tau_a(\omega)$ and the *group delay* $\tau_g(\omega)$ which are defined as

$$\tau_a(\omega) = \frac{\theta(\omega)}{\omega} \tag{5.42a}$$

$$\tau_g(\omega) = \frac{d\theta(\omega)}{d\omega} \tag{5.42b}$$

As functions of frequency, $\tau_a(\omega)$ and $\tau_g(\omega)$ are known as the absolute-delay and group-delay characteristics.

Now assume that the amplitude spectrum of signal $x_k(nT)$ is concentrated in frequency band B given by

$$B = \{\omega : \omega_L \le \omega \le \omega_H\}$$

as illustrated in Fig. 5.18. Also assume that the filter has amplitude and phase responses

$$M(\omega) = \begin{cases} G_0 & \text{for } \omega \in B \\ 0 & \text{otherwise} \end{cases} \tag{5.43}$$

and

$$\theta(\omega) = -\tau_g\omega + \theta_0 \qquad \text{for } \omega \in B \tag{5.44}$$

respectively, where G_0 and τ_g are constants. The z transform of the output of the filter is given by

$$Y(z) = H(z)X(z) = H(z)\sum_{i=1}^{m} X_i(z) = \sum_{i=1}^{m} H(z)X_i(z)$$

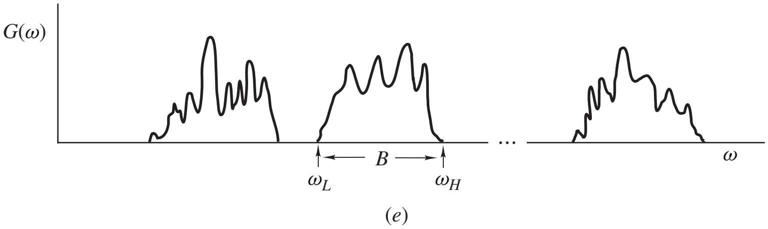


Figure 5.18 Amplitude spectrum of a sum of signals.

and thus the frequency spectrum of the output signal is obtained as

$$\begin{aligned} Y(e^{j\omega T}) &= \sum_{i=1}^{m} H(e^{j\omega T}) X_i(e^{j\omega T}) \\ &= \sum_{i=1}^{m} M(\omega) e^{j\theta(\omega)} X_i(e^{j\omega T}) \end{aligned} \tag{5.45}$$

Hence from Eqs. (5.43)–(5.45), we have

$$Y(e^{j\omega T}) = G_0 e^{-j\omega\tau_g + j\theta_0} X_k(e^{j\omega T})$$

since all signal spectrums except $X_k(e^{j\omega T})$ will be multiplied by zero. If we let $\tau_g = mT$ where m is a constant, we can write

$$Y(z) = G_0 e^{j\theta_0} z^{-m} X_k(z)$$

Therefore, from the time-shifting theorem of the z transform (Theorem 3.4), we deduce the output of the filter as

$$y(nT) = G_0 e^{j\theta_0} x_k(nT - mT)$$

That is, *if the amplitude response of the filter is constant with respect to frequency band B and zero elsewhere and its phase response is a linear function of ω, that is, the group delay is constant in frequency band B, then the output signal is a delayed replica of signal $x_k(nT)$ except that a constant multiplier $G_0 e^{j\theta_0}$ is introduced.*

If the amplitude response of the system is not constant in frequency band B, then the so-called *amplitude distortion* will be introduced since different frequency components of the signal will be amplified by different amounts. On the other hand, if the group delay is not constant in band B, different frequency components will be delayed by different amounts, and *delay (or phase) distortion* will be introduced. Amplitude distortion can be quite objectionable in practice. Consequently, the amplitude response is required to be flat to within a prescribed tolerance in each frequency band that carries information. If the ultimate receiver of the signal is the human ear, e.g., when a speech or music signal is to be processed, delay distortion turns out to be quite tolerable. However, in other applications it can be as objectionable as amplitude distortion, and the delay characteristic is required to be fairly flat. Applications of this type include data transmission where the signal is to be interpreted by digital hardware and image processing where the signal is used to reconstruct an image which is to be interpreted eventually by the human eye.

From Eq. (5.42a), we note that the absolute delay $\tau_a(\omega)$ is constant if the phase response is linear at all frequencies. In such a case, the group delay is also constant and, therefore, delay distortion can also be avoided by ensuring that the absolute delay is constant. However, a constant absolute delay is far more difficult to achieve in practice since the phase response would need to be linear at all frequencies.[4]

[4] *This is why the absolute delay is hardly ever mentioned in DSP and communications textbooks.*

REFERENCES

[1] G. Strang, *Introduction to Linear Algebra*, 3rd ed., MA: Wellesley-Cambridge Press, 2003.

[2] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed., New York: Academic, 1985.

[3] E. I. Jury, *Theory and Application of the Z-Transform Method*, New York: Wiley, 1964.

[4] E. I. Jury, *Inners and Stability of Dynamical Systems*, New York: Wiley-Interscience, 1974.

[5] N. K. Bose, *Digital Filters*, New York: North-Holland, 1985.

[6] E. I. Jury, "A simplified stability criterion for linear discrete systems," *Proc. IRE*, vol. 50, pp. 1493–1500, June 1962.

[7] B. D. O. Anderson and E. I. Jury, "A simplified Schur-Cohn test," *IEEE Trans. Automatic Control*, vol. 18, pp. 157–163, Apr. 1973.

[8] M. Marden, *The Geometry of the Zeros of a Polynomial in a Complex Variable*, New York: Amer. Math. Soc., pp. 152–157, 1949.

[9] H. Freeman, *Discrete-Time Systems,* New York: Wiley, 1965.

[10] S. J. Hammarling, "Numerical solution of the stable, non-negative definite Lyapunov equation," *IMA J. Numer. Anal.*, vol. 2, pp. 303–323, 1982.

PROBLEMS

5.1. Derive the transfer functions of the systems in Fig. P4.5*a* and *b*.

5.2. Derive the transfer functions of the systems in Fig. P4.6*a* and *b*.

5.3. Derive the transfer functions of the systems in Fig. P4.7 and P4.8.

5.4. A recursive system is characterized by the equations

$$\begin{aligned} y(nT) &= y_1(nT) + \tfrac{7}{4}y(nT-T) - \tfrac{49}{32}y(nT-2T) \\ y_1(nT) &= x(nT) + \tfrac{1}{2}y_1(nT-T) \end{aligned}$$

Obtain its transfer function.

5.5. A system is represented by the state-space equations

$$\begin{aligned} \mathbf{q}(nT+T) &= \mathbf{A}\mathbf{q}(nT) + \mathbf{b}x(nT) \\ y(nT) &= \mathbf{c}^T\mathbf{q}(nT) + dx(nT) \end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \mathbf{c}^T = \begin{bmatrix} \frac{7}{2} & \frac{5}{2} & \frac{5}{2} \end{bmatrix} \qquad d = 1$$

Deduce its transfer function.

5.6. Show that

$$H(z) = \frac{\sum_{i=0}^{M} a_i z^{M-i}}{z^N + \sum_{i=1}^{N} b_i z^{N-i}}$$

represents a causal system only if $M \leq N$.

5.7. (*a*) Find the impulse response of the system shown in Fig. 5.2.

(*b*) Repeat part (*a*) for the system of Fig. 5.4.

5.8. Obtain the impulse response of the system in Prob. 5.4. Sketch the response.

5.9. At $z = p_i$, $F(z)$ is analytic and $G(z)$ has a simple pole. Show that

$$\operatorname*{\mathfrak{R}es}_{z=p_i}[F(z)G(z)] = F(p_i)\operatorname*{\mathfrak{R}es}_{z=p_i} G(z)$$

5.10. (*a*) In Sec. 5.3.1, it was shown that a system with simple poles is stable if and only if its poles are inside the unit circle of the z plane. Show that this constraint applies equally well to a system that has one or more second-order poles.

(*b*) Indicate how you would proceed to confirm the validity of the stability constraint in part (*a*) for the case where the system has one or more poles of order higher than two.

5.11. Starting from first principles, show that

$$H(z) = \frac{z}{\left(z - \frac{1}{4}\right)^4}$$

represents a stable system.

5.12. (*a*) A recursive system is represented by

$$H(z) = \frac{z^6}{6z^6 + 5z^5 + 4z^4 + 3z^3 + 2z^2 + z + 1}$$

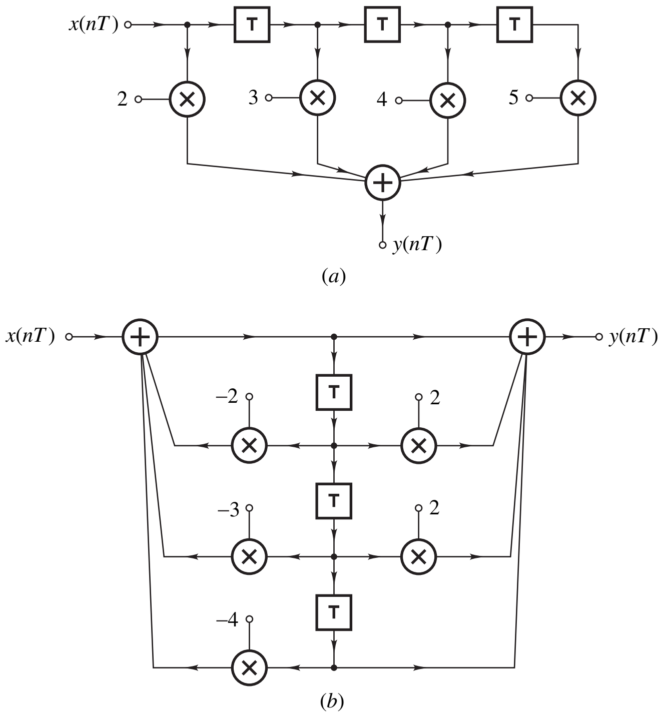
Check the system for stability.

(*b*) Repeat part (*a*) if

$$H(z) = \frac{(z+2)^2}{6z^6 + 5z^5 - 4z^4 + 3z^3 + 2z^2 + z + 1}$$

5.13. (*a*) Check the system of Fig. P5.13*a* for stability.

(*b*) Check the system of Fig. P5.13*b* for stability.



(*a*)

(*b*)

Figure P5.13*a* and *b*

5.14. (*a*) A system is characterized by the difference equation

$$y(nT) = x(nT) - \tfrac{1}{2}x(nT - T) - \tfrac{1}{3}x(nT - 2T) - \tfrac{1}{4}x(nT - 3T) - \tfrac{1}{5}x(nT - 4T)$$

By using appropriate tests, check the stability of the system.

(*b*) Repeat part (*a*) for the system represented by the equation

$$y(nT) = x(nT) - \tfrac{1}{2}y(nT - T) - \tfrac{1}{3}y(nT - 2T) - \tfrac{1}{4}y(nT - 3T) - \tfrac{1}{5}y(nT - 4T)$$

5.15. Obtain (*a*) the transfer function, (*b*) the impulse response, and (*c*) the necessary condition for stability for the system of Fig. P5.15. The constants m_1 and m_2 are given by

$$m_1 = 2r\cos\theta \qquad \text{and} \qquad m_2 = -r^2$$

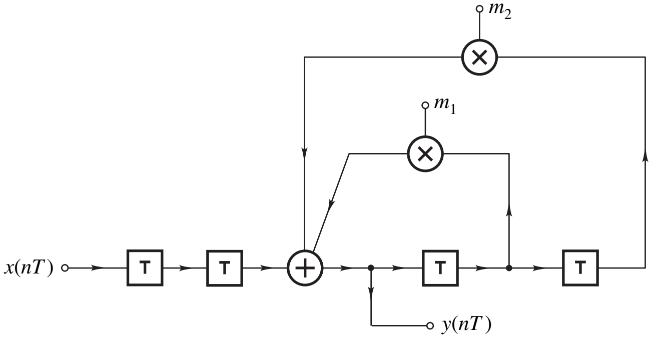


Figure P5.15

5.16. A system is characterized by the transfer function

$$H(z) = \frac{z^4}{4z^4 + 3z^3 + mz^2 + z + 1}$$

Find the range of m that will result in a stable system.

5.17. Find the permissible range for m in Fig. P5.17 if the system is to be stable.

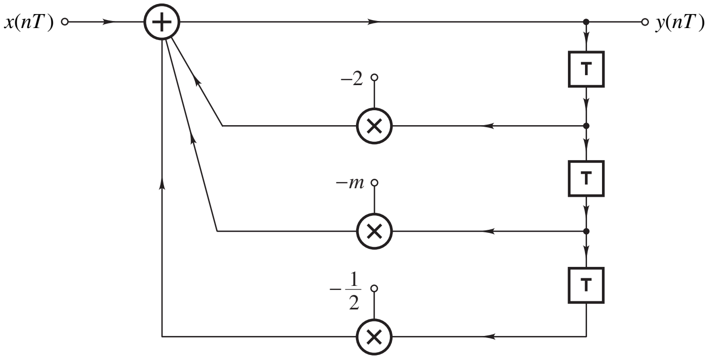


Figure P5.17

5.18. A system is characterized by the transfer function

$$H(z) = \frac{1}{z^2 + \frac{1}{4}}$$

Derive an expression for its unit-step response.

5.19. A system is characterized by the transfer function

$$H(z) = \frac{32z}{z - \frac{1}{2}}$$

(a) Find the response of the system at $t = 4T$ using the convolution summation if the excitation is

$$x(nT) = (5 + n)u(nT)$$

(b) Give a graphical construction for the convolution in part (a) indicating relevant quantities.

5.20. Repeat part (a) of Prob. 5.19 using the general inversion formula in Eq. (3.8).

5.21. A system is characterized by the transfer function

$$H(z) = \frac{z^2 - z + 1}{z^2 - z + 0.5}$$

Obtain its unit-step response.

5.22. Find the unit-step response of the system shown in Fig. 5.2.

5.23. Find the unit-ramp response of the system shown in Fig. 5.4 if $T = 1$ s.

5.24. The input excitation in Fig. 5.2 is

$$x(nT) = \begin{cases} n & \text{for } 0 \le n \le 2 \\ 4 - n & \text{for } 2 < n \le 4 \\ 0 & \text{for } \quad n > 4. \end{cases}$$

Determine the response for $0 \le n \le 5$ by using the z transform.

5.25. Repeat Prob. 4.24 by using the z transform. For each of the three cases deduce the exact frequency of the transient oscillation and also the steady-state value of the response if $T = 1$ s.

5.26. A system has a transfer function

$$H(z) = \frac{1}{z^2 + \frac{1}{4}}$$

(a) Find the response if

$$x(nT) = u(nT) \sin \omega nT$$

(b) Deduce the steady-state sinusoidal response.

5.27. A system is characterized by

$$H(z) = \frac{1}{(z - r)^2}$$

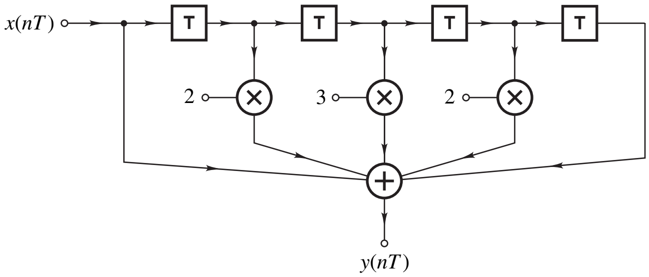


Figure P5.29

where $|r| < 1$. Show that the steady-state sinusoidal response is given by

$$y(nT) = M(\omega)\sin[\omega nT + \theta(\omega)]$$

where

$$M(\omega) = |H(e^{j\omega T})| \qquad \text{and} \qquad \theta(\omega) = \arg H(e^{j\omega T})$$

5.28. (*a*) In Sec. 5.5.1 it was shown that the steady-state sinusoidal response of a system with simple poles is given by Eq. (5.30). Show that this equation is also valid for a system with one or more second-order poles.

(*b*) Indicate how you would proceed to show that Eq. (5.30) applies equally well to a system with one or more poles of order higher than two.

5.29. Figure P5.29 depicts a nonrecursive system.

(*a*) Derive an expression for its amplitude response.

(*b*) Derive an expression for its phase response.

(*c*) Calculate the gain in dB at $\omega = 0$, $\omega_s/4$, and $\omega_s/2$ (ω_s is the sampling frequency in rad/s).

(*d*) Calculate the phase-shift in degrees at $\omega = 0$, $\omega_s/4$, and $\omega_s/2$.

5.30. The discrete-time signal

$$x(nT) = u(nT)\sin\omega nT$$

is applied to the input of the system in Fig. P5.30.

(*a*) Give the steady-state time-domain response of the system.

(*b*) Derive an expression for the amplitude response.

(*c*) Derive an expression for the phase response.

(*d*) Calculate the gain and phase shift for $\omega = \pi/4T$ rad/s.

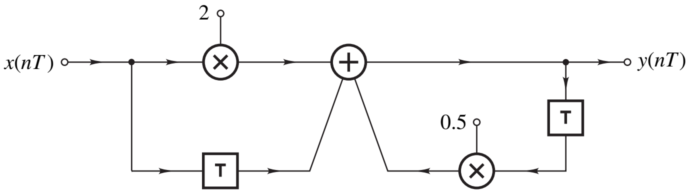


Figure P5.30

5.31. Show that the poles of the allpass transfer function in Eq. (5.41) are the reciprocals of the zeros.

5.32. Figure P5.32 shows a nonrecursive system.

(*a*) Derive expressions for the amplitude and phase responses.

(*b*) Determine the transmission zeros of the system, i.e., zero-gain frequencies.

(*c*) Sketch the amplitude and phase responses.

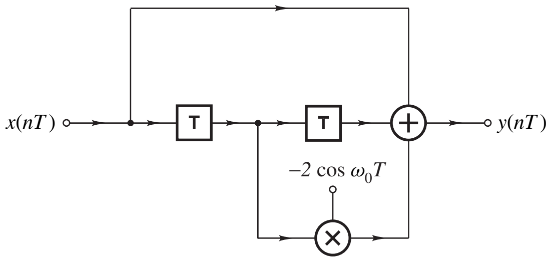


Figure P5.32

5.33. Show that the equation

$$y(nT) = x(nT) + 2x(nT - T) + 3x(nT - 2T) + 4x(nT - 3T) + 3x(nT - 4T)$$
$$+ 2x(nT - 5T) + x(nT - 6T)$$

represents a constant-delay system.

5.34. Derive expressions for the amplitude and phase responses of the system shown in Fig. 5.4.

5.35. Table P5.35 gives the transfer function coefficients of four digital filters labeled A to D. Using MATLAB, compute and plot $20 \log M(\omega)$ versus ω in the range 0 to 5.0 rad/s. On the basis of the plots obtained, identify a lowpass, a highpass, bandpass, and a bandstop filter. Each filter has a transfer function of the form

$$H(z) = H_0 \prod_{i=1}^{2} \frac{a_{0i} + a_{1i}z + a_{0i}z^2}{b_{0i} + b_{1i}z + z^2}$$

and the sampling frequency is 10 rad/s in each case.

Table P5.35 Transfer-function coefficients for Prob. 5.35

Filter	i	a_{0i}	a_{1i}	b_{0i}	b_{1i}
A	1	$2.222545E - 1$	$-4.445091E - 1$	$4.520149E - 2$	$1.561833E - 1$
	2	$3.085386E - 1$	$-6.170772E - 1$	$4.509715E - 1$	$2.168171E - 1$
		$H_0 = 1.0$			
B	1	5.490566	9.752955	$7.226400E - 1$	$4.944635E - 1$
	2	$5.871082E - 1$	-1.042887	$7.226400E - 1$	$-4.944634E - 1$
		$H_0 = 2.816456E - 2$			
C	1	$1.747744E - 1$	$1.517270E - 8$	$5.741567E - 1$	1.224608
	2	1.399382	$1.214846E - 7$	$5.741567E - 1$	-1.224608
		$H_0 = 8.912509E - 1$			
D	1	9.208915	$1.561801E + 1$	$5.087094E - 1$	-1.291110
	2	2.300089	1.721670	$8.092186E - 1$	-1.069291
		$H_0 = 6.669086E - 4$			

5.36. Show that the gain and phase shift in a digital filter satisfy the relations

$$M(\omega_s - \omega) = M(\omega) \qquad \text{and} \qquad \theta(\omega_s - \omega) = -\theta(\omega)$$

This page intentionally left blank

CHAPTER 6

THE SAMPLING PROCESS

6.1 INTRODUCTION

The sampling process was briefly reviewed in Chap. 1 and there was reason to refer to it in Sec. 3.9.3. In this chapter, it is treated in some detail both from a theoretical as well as practical point of view. The sampling process involves several aspects that need to be addressed in detail, as follows:

- The constituent components of a sampling system
- The underlying principles that make the sampling process possible
- The applications of the sampling process
- The imperfections introduced through the use of practical components

The sampling process requires several components. Converting a continuous- to a discrete-time signal would require some sort of a switch. However, a sampling system that uses just a simple switch would introduce a certain kind of signal distortion known as *aliasing* if the signal is not bandlimited. Continuous-time signals, man-made or otherwise, are only approximately bandlimited, at best, and almost always they must be preprocessed by suitable analog lowpass filters to render them bandlimited so as to prevent aliasing. At some point, a discrete-time signal would need to be converted back to a continuous-time signal and this conversion requires some sort of a sample-and-hold device. In practice, devices of this type tend to produce a noisy version of the required continuous-time signal and once again a suitable analog lowpass filter would be required to remove the noise introduced.

The one mathematical principle that makes the sampling process workable is the sampling theorem. The validity of this theorem can be demonstrated by examining the relationships that exist between the spectrums of continuous- and discrete-time signals. A most important relation in this respect is the so-called *Poisson's summation formula* which gives the frequency spectrum of a discrete-time signal in terms of the spectrum of the underlying continuous-time signal.

The connection between the spectral properties of discrete- and continuous-time signals is made by examining a class of signals referred to here as *impulse-modulated signals*. These are both sampled as well as continuous in time and, therefore, they share common characteristics with discrete-time signals on the one hand and continuous-time signals on the other. Consequently, they serve as a bridge between the discrete- and continuous-time worlds. The dual personality of impulse-modulated signals allows them to possess both a Fourier and a z transform and by examining the properties of these signals, some fundamental relations can be established between these transforms. From this link follow the spectral relationships between discrete- and continuous-time signals.

The study of impulse-modulated signals requires a clear understanding of what are impulse functions and what are their spectral characteristics. This subject has received considerable attention throughout most of the twentieth century and some very rigorous theories have been proposed, for example, the treatment of impulse functions as *generalized functions* [1]. What these theories offer in rigor, they lack in practicality and, in consequence, they have not received the attention they deserve. At the other extreme, authors often define impulse functions in terms of thin tall pulses which are easy to reproduce in the lab but which lack the mathematical sophistication of generalized functions. In order to obtain a true impulse function, the duration of the pulse must be made infinitesimally small and its amplitude must be made infinitely large but this limiting operation is fraught with perils and pitfalls. In this chapter, a somewhat new way of looking at impulse functions is proposed which provides practical solutions to the classical DSP problems without compromising mathematical principles.

Through the sampling process, digital filters can be used to process continuous-time signals. The continuous-time signal is first converted to a discrete-time signal, which is then processed by a digital filter. Subsequently, the processed discrete-time signal is converted back to a continuous-time signal. Once we establish a relation between analog and digital filters, in addition to our being able to use digital filters to perform analog-filter functions we can also design digital filters by using analog filter methodologies. In fact, some of the better infinite-duration impulse response (IIR) digital filters are designed by transforming analog into digital filters.

In addition to analog filters, switches, and sample-and-hold devices, a sampling system also uses quantizers and encoders. All these components have imperfections that need to be examined carefully.

In this chapter, the Fourier transform theory of Chap. 2 is first extended to impulse functions and then to periodic and impulse-modulated signals. On the basis of these principles, Poisson's summation formula is derived in a rather practical way. From this formula, the crucial interrelations that exist between the spectrums of continuous- and discrete-time signals are established. From these interrelations, the conditions that must be satisfied for a discrete-time signal to be a true representation of the underlying continuous-time signal become immediately obvious and the validity of the sampling theorem can be easily established. The chapter concludes by examining the imperfections introduced by the various components of the sampling system.

6.2 FOURIER TRANSFORM REVISITED

6.2.1 Impulse Functions

The properties and theorems of the Fourier transform described in Sec. 2.3.3 apply to the extent that the convergence theorem (Theorem 2.5) is satisfied. In practice, a number of important signals are not absolutely integrable and, therefore, two situations can arise: Either the integral in Eq. (2.27) or that in Eq. (2.29) does not converge. Signals of this category include impulse signals and the entire class of periodic signals. We will show in this section that many of the mathematical difficulties associated with these signals can be circumvented by paying particular attention to the definition of impulse functions.

Impulse signals are used in many applications and are part and parcel of the sampling process, as will be demonstrated later on in this chapter; consequently, their properties, spectral or otherwise, must be clearly understood by the DSP practitioner. Such signals can be modeled in terms of impulse functions. A *unit impulse function* that has been used for many years can be generated by scaling the amplitude of the pulse signal in Example 2.5*a* from unity to $1/\tau$, that is,

$$\bar{p}_\tau(t) = \frac{1}{\tau} p_\tau(t) = \begin{cases} \frac{1}{\tau} & \text{for } |t| \le \tau/2 \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

The Fourier transform of this pulse is obtained from Example 2.5*a* as

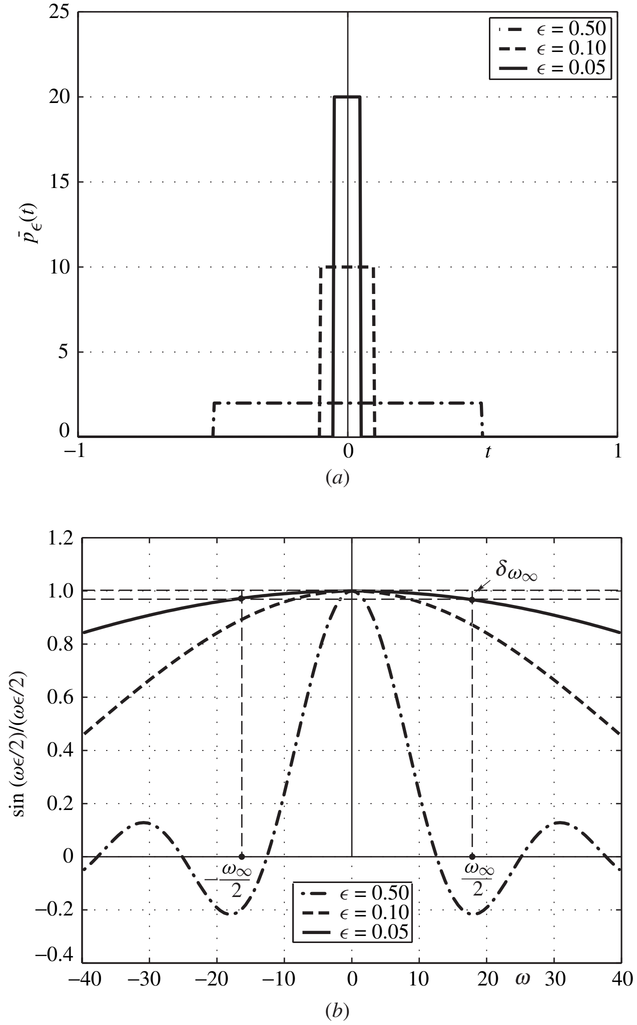
$$\mathcal{F}\bar{p}_\tau(t) = \frac{1}{\tau}\mathcal{F}p_\tau(t) = \frac{2\sin\omega\tau/2}{\omega\tau} \tag{6.2}$$

Evidently, as τ approaches zero, the pulse in Eq. (6.1) becomes very thin and very tall, as can be seen in Fig. 6.1*a* but the area of the pulse remains constant and equal to unity. As long as τ is finite, the absolute integrability of the signal is assured and, therefore, it would satisfy Theorem 2.5. If we now attempt to find the Fourier transform of the pulse as $\tau \to 0$, we get

$$\mathcal{F}\lim_{\tau\to 0}\bar{p}_\tau(t) = \int_{-\infty}^{\infty}\lim_{\tau\to 0}\bar{p}_\tau(t)e^{-j\omega t}\,dt = \int_{-\tau/2}^{\tau/2}\lim_{\tau\to 0}\bar{p}_\tau(t)e^{-j\omega t}\,dt$$

If we now attempt to evaluate the limit $\lim_{\tau\to 0}[\bar{p}_\tau(t)e^{-j\omega t}]$, we find that it becomes unbounded at $\tau = 0$ and, therefore, the above integral cannot be evaluated. More formally, the integral does not exist in the Riemann sense of a definite integral (see pp. 217–221 of Kaplan [2]). However, since the definite integral of a real function of t would give the area bounded by the graph of the function and the t axis, we might be tempted to write

$$\begin{aligned}\mathcal{F}\lim_{\tau\to 0}\bar{p}_\tau(t) &= \int_{-\tau/2}^{\tau/2}\lim_{\tau\to 0}\bar{p}_\tau(t)e^{-j\omega t}\,dt \\ &\approx \int_{-\tau/2}^{\tau/2}\lim_{\tau\to 0}\bar{p}_\tau(t)\,dt = 1\end{aligned}$$



(*a*)

(*b*)

Figure 6.1 Impulse function: (*a*) Pulse function for three values of ϵ, (*b*) corresponding Fourier transform.

since $e^{-j\omega t} \to 1$ for $-\tau/2 \le t \le \tau/2$ with $\tau \to 0$ and the area of the pulse $\bar{p}_\tau(t)$ is equal to unity and remains unity as $\tau \to 0$. Many authors have taken this approach in the past [3]. Interestingly, the Fourier transform obtained in the above analysis is consistent with the limit of the Fourier transform of the pulse given by Eq. (6.2), that is,

$$\lim_{\tau\to 0} \mathcal{F}\bar{p}_\tau(t) = \lim_{\tau\to 0} \frac{2\sin\omega\tau/2}{\omega\tau} = 1$$

Now, if we attempt to find the inverse Fourier transform of 1, we run into certain mathematical difficulties. From Eq. (2.29), we have

$$\begin{aligned}\mathcal{F}^{-1}1 &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{j\omega t}\,d\omega \\ &= \frac{1}{2\pi}\left[\int_{-\infty}^{\infty}\cos\omega t\,d\omega + j\int_{-\infty}^{\infty}\sin\omega t\,d\omega\right]\end{aligned}$$

Mathematicians will tell us that these integrals do not converge or do not exist[1] and, therefore, we conclude that the inverse Fourier transform of 1 does not satisfy Eq. (2.29).

Defining a unit impulse function in terms of an infinitesimally thin, infinitely tall pulse is obviously problematic. However, there are certain important practical advantages as well in using a thin, tall pulse. Specifically, pulses are very easy to create in terms of electrical voltage or current waveforms and, in fact, we will use them later on in this chapter in the implementation of sampling systems. For this reason, we would like to define the unit impulse function in terms of a thin, tall pulse but at the same time we would like to find a way to avoid the above mathematical pitfalls.

The above difficulties can be circumvented in a practical way by defining impulse functions in terms of the way they interact with other functions under integration while adopting a somewhat practical interpretation of the limit of a function. In this approach, a function $\gamma(t)$ is said to be a *unit impulse function* if for a function $x(t)$ which is continuous for $|t| < \epsilon$, we have

$$\int_{-\infty}^{\infty}\gamma(t)x(t)\,dt \doteq x(0) \tag{6.3}$$

where the symbol $\doteq$ is used to indicate that the relation is approximate in the very special sense that the integral at the left-hand side can be made to approach the value of $x(0)$ to any desired degree of precision. Now consider the pulse function

$$\bar{p}_\epsilon(t) = \lim_{\tau\to\epsilon}\bar{p}_\tau(t) = \begin{cases}\frac{1}{\epsilon} & \text{for } |t| \le \epsilon/2 \\ 0 & \text{otherwise}\end{cases} \qquad \text{with } \epsilon \ne 0 \tag{6.4}$$

where ϵ is a small but finite constant. If we let

$$\gamma(t) = \bar{p}_\epsilon(t)$$

then Eqs. (6.3) and (6.4) yield

$$\begin{aligned}\int_{-\infty}^{\infty}\lim_{\tau\to\epsilon}\bar{p}_\tau(t)x(t)\,dt &= \int_{-\epsilon/2}^{\epsilon/2}\frac{1}{\epsilon}x(t)\,dt \doteq \frac{1}{\epsilon}x(0)\int_{-\epsilon/2}^{\epsilon/2}dt \\ &\doteq x(0)\end{aligned}$$

[1] *Presumably the values of these integrals would depend on the limiting behavior of the sine and cosine functions at infinity but nobody seems to have come up with a reasonable answer for that so far!*

and by making ϵ smaller and smaller the integral at the left-hand can be made to approach the value $x(0)$ as closely as desired. In other words, the pulse function of Eq. (6.4) satisfies Eq. (6.3) and it is, therefore, an impulse function that can be represented, say, by $\delta(t)$.

From Eq. (6.4) and Table 2.1, we have

$$\lim_{\tau\to\epsilon} \bar{p}_\tau(t) \leftrightarrow \lim_{\tau\to\epsilon} \frac{2\sin\omega\tau/2}{\omega\tau} \tag{6.5}$$

As τ is reduced, the pulse at the left-hand side tends to become thinner and taller whereas the so-called *sinc function* at the right-hand side tends to be flattened out as depicted in Fig. 6.1*b*. For some small value of ϵ, the sinc function will be equal to unity to within an error δ_{ω_∞} over a bandwidth $-\omega_\infty/2 \le \omega \le \omega_\infty/2$ as shown in Fig. 6.1*b* where ω_∞ is inversely related to ϵ, i.e., the smaller the ϵ the larger the ω_∞. Evidently, for some sufficiently small but finite ϵ, the sinc function would be approximately equal to unity over the frequency range $-\omega_\infty/2$ to $\omega_\infty/2$ which could include all the frequencies of practical interest. Therefore, from Eq. (6.5), we can write

$$\delta(t) = \bar{p}_\epsilon(t) \leftrightarrow \frac{2\sin\omega\epsilon/2}{\omega\epsilon} = i(\omega) \tag{6.6}$$

and since

$$i(\omega) = \frac{2\sin\omega\epsilon/2}{\omega\epsilon} \doteq 1 \qquad \text{for } |\omega| < \omega_\infty/2$$

function $i(\omega)$ may be referred to as a *frequency-domain unity function*.

Let us now examine the sinc function

$$\text{sinc}_\varpi(t) = \frac{\sin\varpi t/2}{\pi t}$$

This is, of course, a pulse-like function that tends to become thinner and taller as ϖ is increased, very much like the pulse in Fig. 6.1*a*, as can be seen in Fig. 6.2*a*. Now let us consider the function

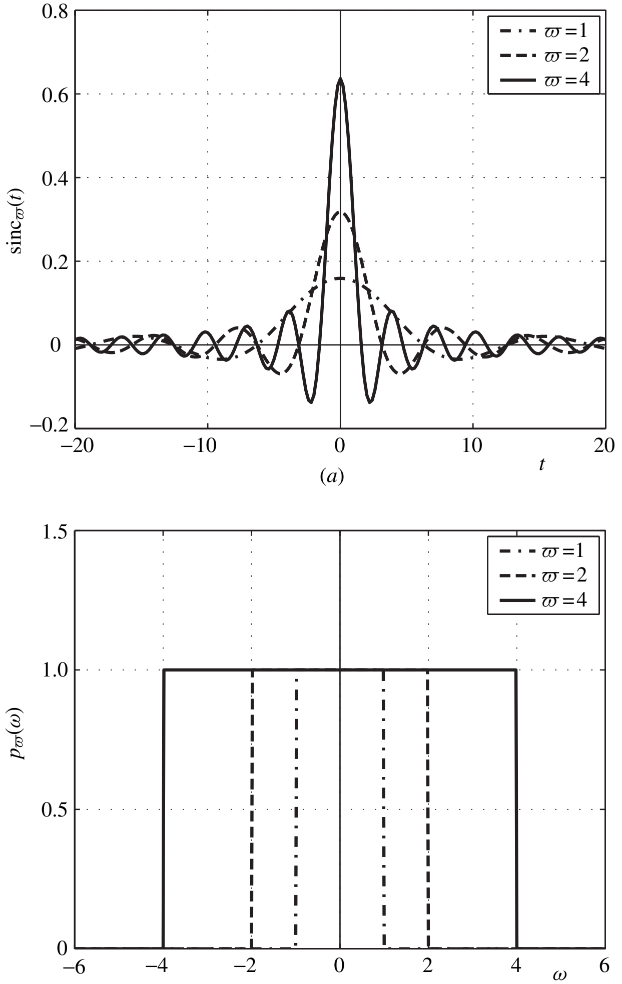
$$\text{sinc}_{\omega_\infty/2}(t) = \lim_{\varpi\to\omega_\infty/2} \frac{\sin\varpi t/2}{\pi t} = \frac{\sin\omega_\infty t/4}{\pi t} \tag{6.7}$$

where ω_∞ is a large but finite constant. If we let

$$\gamma(t) = \text{sinc}_{\omega_\infty/2}(t)$$

then Eqs. (6.3) and (6.7) yield

$$\begin{aligned}\int_{-\infty}^{\infty} \text{sinc}_{\omega_\infty/2}(t)x(t)\,dt &= \int_{-\infty}^{\infty} \frac{\sin\omega_\infty t/4}{\pi t}x(t)\,dt \\ &= \int_{-\infty}^{-\epsilon/2} \frac{\sin\omega_\infty t/4}{\pi t}x(t)\,dt + \int_{-\epsilon/2}^{\epsilon/2} \frac{\sin\omega_\infty t/4}{\pi t}x(t)\,dt \\ &\quad + \int_{\epsilon/2}^{\infty} \frac{\sin\omega_\infty t/4}{\pi t}x(t)\,dt\end{aligned} \tag{6.8a}$$



(a)

(b)

Figure 6.2 Impulse function: (a) Sinc function for three values of ϖ, (b) corresponding Fourier transforms.

If $\phi(x)$ is an absolutely integrable function and a and b are finite or infinite constants, then

$$\lim_{\omega\to\infty}\int_a^b \sin(\omega t)\phi(t)\,dt = 0 \tag{6.8b}$$

according to the Riemann-Lebesque lemma [4]. Thus if $x(t)/t$ is absolutely integrable, the first and the last integrals at the right-hand side in Eq. (6.8a) approach zero. Since

$$x(t) \simeq x(0)|t| \leq \epsilon/2 \tag{6.8c}$$

Eqs. (6.8a)–(6.8c) give give[2]

$$\begin{aligned}\int_{-\infty}^{\infty} \operatorname{sinc}_{\omega_\infty/2}(t)x(t)\,dt &\doteq \int_{-\epsilon/2}^{\epsilon/2} \frac{\sin \omega_\infty t/4}{\pi t} x(t)\,dt \\ &\doteq x(0)\int_{-\epsilon/2}^{\epsilon/2} \frac{\sin \omega_\infty t/4}{\pi t}\,dt\end{aligned} \tag{6.8d}$$

Now for a large ω_∞, it is known that

$$\int_{-\epsilon/2}^{\epsilon/2} \frac{\sin \omega_\infty t/4}{\pi t} \doteq \int_{-\infty}^{\infty} \frac{\sin \omega_\infty t/4}{\pi t}\,dt = 1 \tag{6.8e}$$

(see pp. 280–281 of Ref. [4]) and, therefore, Eqs. (6.8d) and (6.8e) give

$$\int_{-\infty}^{\infty} \operatorname{sinc}_{\omega_\infty/2}(t)x(t)\,dt \doteq x(0)$$

In effect, the sinc function of Eq. (6.7) satisfies Eq. (6.3) and we conclude, therefore, that $\operatorname{sinc}_{\omega_\infty/2}(t)$ is another impulse function that could be represented, say, by $\delta'(t)$.

From Example 2.7, we have

$$\frac{\sin \varpi t/2}{\pi t} \leftrightarrow p_{\varpi}(\omega) \tag{6.9}$$

where

$$p_{\varpi}(\omega) = \begin{cases} 1 & \text{for } |\omega| \le \varpi/2 \\ 0 & \text{otherwise} \end{cases}$$

and hence from Eq. (6.9), we can write

$$\delta'(t) = \frac{\sin \omega_\infty t/2}{\pi t} \leftrightarrow p_{\omega_\infty}(\omega) = i'(\omega) \tag{6.10}$$

where

$$i'(\omega) = 1 \qquad \text{for } |\omega| \le \omega_\infty/2$$

Like function $i(\omega)$, function $i'(\omega)$ behaves as a frequency-domain unity function as can be seen in Fig. 6.2*b*.

In the above analysis, we have identified two distinct impulse functions, namely, $\delta(t)$ and $\delta'(t)$ and we have demonstrated that their Fourier transforms are unity functions, as shown in Table 6.1. In fact, there are other Fourier transform pairs with these properties but the above two are entirely sufficient for the purposes of this textbook. Since the two impulse functions have the same properties, they are alternative but *equivalent* forms and each of the two transform pairs in Eqs. (6.6) and (6.10) can be represented by

$$\delta(t) \leftrightarrow i(\omega)$$

[2] *See 278281 of Ref. [4] for a relevant discussion.*

Table 6.1 Impulse and unity functions

$\boldsymbol{\delta(t)}$	$\boldsymbol{i(\omega)}$
$\bar{p}_\epsilon(t)$ $\dfrac{\sin \omega_\infty t/2}{\pi t}$	$\dfrac{2\sin \omega\epsilon/2}{\omega\epsilon}$ $p_{\omega_\infty}(\omega)$

or even by the symbolic notation

$$\delta(t) \leftrightsquigarrow 1 \tag{6.11}$$

where the wavy, two-way arrow $\leftrightsquigarrow$ signifies that the relation is approximate with the understanding that it can be made as exact as desired by making ϵ in Eq. (6.6) small enough or ω_∞ in Eq. (6.7) large enough. Symbolic graphs for the impulse and unity functions are shown in Fig. 6.3a.

Some important properties of impulse functions, which will be found very useful in establishing the relationships between continuous- and discrete-time signals, can be stated in terms of the following theorem:

Theorem 6.1A Properties of Time-Domain Impulse Functions ***Assuming that $x(t)$ is a continuous function of t for $|t| < \epsilon$, the following relations hold:***

(a)
$$\int_{-\infty}^{\infty} \delta(t-\tau)x(t)\,dt = \int_{-\infty}^{\infty} \delta(-t+\tau)x(t)\,dt \doteq x(\tau)$$

(b)
$$\delta(t-\tau)x(t) = \delta(-t+\tau)x(t) \doteq \delta(t-\tau)x(\tau)$$

(c)
$$\delta(t)x(t) = \delta(-t)x(t) \doteq \delta(t)x(\mathbf{0})$$

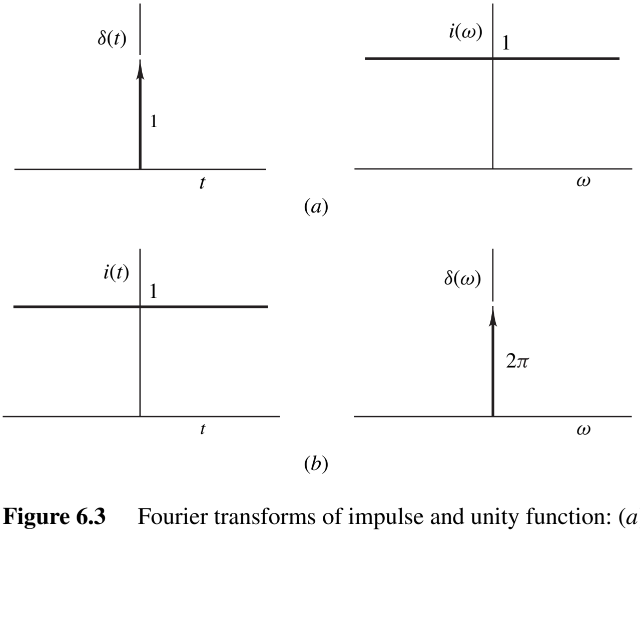


Figure 6.3 Fourier transforms of impulse and unity function: (a) $\delta(t) \leftrightarrow i(\omega)$, ($b$) $i(t) \leftrightarrow 2\pi\delta(\omega)$.

Proof

(*a*) From Eq. (6.6), we can write

$$\begin{aligned}\int_{-\infty}^{\infty} \delta(t-\tau)x(t)\,dt &= \int_{-\infty}^{\infty} \bar{p}_\epsilon(t-\tau)x(t)\,dt\\ &\bumpeq \frac{1}{\epsilon}x(\tau)\int_{\tau-\epsilon/2}^{\tau+\epsilon/2} dt \bumpeq x(\tau) \qquad \blacktriangle\end{aligned}$$

(*b*) Let $x(t)$ and $\xi(t)$ be continuous functions of t for $|t| < \epsilon$. We can write

$$\begin{aligned}\int_{-\infty}^{\infty} \delta(t-\tau)x(t)\xi(t)\,dt &= \int_{-\infty}^{\infty} \bar{p}_\epsilon(t-\tau)x(t)\xi(t)\,dt\\ &\bumpeq \frac{1}{\epsilon}x(\tau)\xi(\tau)\int_{\tau-\epsilon/2}^{\tau+\epsilon/2} dt\\ &\bumpeq x(\tau)\xi(\tau) \end{aligned}\tag{6.13a}$$

On the other hand,

$$\begin{aligned}\int_{-\infty}^{\infty} \delta(t-\tau)x(\tau)\xi(t)\,dt &= \int_{-\infty}^{\infty} \hat{p}_\epsilon(t-\tau)x(\tau)\xi(t)\,dt\\ &\bumpeq \frac{1}{\epsilon}x(\tau)\xi(\tau)\int_{\tau-\epsilon/2}^{\tau+\epsilon/2} dt\\ &\bumpeq x(\tau)\xi(\tau) \end{aligned}\tag{6.13b}$$

From Eqs. (6.13a) and (6.13b), we have

$$\int_{-\infty}^{\infty} \delta(t-\tau)x(t)\xi(t)\,dt \bumpeq \int_{-\infty}^{\infty} \delta(t-\tau)x(\tau)\xi(t)\,dt$$

and, therefore,

$$\delta(t-\tau)x(t) \bumpeq \delta(t-\tau)x(\tau) \tag{6.14a}$$

Since impulse functions, as defined above, are even functions of t, we have

$$\delta(-t) = \delta(t) \qquad \text{and} \qquad \delta(-t+\tau) = \delta(t-\tau) \tag{6.14b}$$

and hence Eqs. (6.14a) and (6.14b) yield

$$\delta(t-\tau)x(t) = \delta(-t+\tau)x(t) \bumpeq \delta(t-\tau)x(\tau) \qquad \blacktriangle$$

(*c*) Part (*c*) follows readily from part (*b*) by letting $\tau = 0$. ▲

In words, part (a) of the theorem is saying that *integrating an impulse function times a continuous function causes the integral to assume the value of the continuous function at the location of the impulse function.* Similarly, parts (b) and (c) are saying that *multiplying a continuous function by an impulse function yields a product of the impulse function times the value of the continuous function at the location of the impulse.*

The above theorem applies also to the impulse function in Eq. (6.10). The theorem is essentially a generalization of the definition of impulse functions and, in fact, any distinct functions that satisfy it may be deemed to be *equivalent* impulse functions. The theorem is of considerable practical importance as will be found out later on in this chapter.

In the above analysis, time-domain impulse functions have been examined whose Fourier transforms are unity functions in the frequency domain. Occasionally, frequency-domain impulse functions are required whose inverse Fourier transforms are unity functions in the time domain. Such functions can be readily obtained from the impulse and unity functions examined already and, as will be shown below, they are required for the spectral representation of periodic signals. Consider the Fourier transform pair in Eq. (6.6), namely,

$$\delta(t) = \bar{p}_\epsilon(t) \leftrightarrow \frac{2\sin\omega\epsilon/2}{\omega\epsilon} = i(\omega)$$

By applying the symmetry theorem (Theorem 2.7), we can write

$$i(t) \leftrightarrow 2\pi\delta(-\omega) \tag{6.15a}$$

where

$$i(t) = \frac{2\sin t\epsilon/2}{t\epsilon} \simeq 1 \qquad \text{for } |t| < t_\infty \tag{6.15b}$$

and

$$\delta(-\omega) = \lim_{\varpi\to\epsilon} \bar{p}_\varpi(-\omega) = \lim_{\varpi\to\epsilon} \bar{p}_\varpi(\omega) = \delta(\omega) \tag{6.15c}$$

where t_∞ is a positive constant that defines the range of t over which $i(t) \simeq 1$ and is inversely related to ϵ. Therefore, from Eqs. (6.15a)–(6.15c), we can write

$$i(t) \leftrightarrow 2\pi\delta(\omega) \tag{6.16a}$$

or

$$1 \leftrightsquigarrow 2\pi\delta(\omega) \tag{6.16b}$$

where $i(t)$ and $\delta(\omega)$ may be referred to as *time-domain unity function* and *frequency-domain unit impulse function*, respectively, by analogy with the frequency-domain unity function and time-domain impulse function, respectively. These functions can be represented by the symbolic graphs of Fig. 6.3b.

The properties of time-domain impulse functions apply equally well to frequency-domain impulse functions as summarized by Theorem 6.1B below.

Theorem 6.1B Properties of Frequency-Domain Impulse Functions *Assuming that $X(j\omega)$ is a continuous function of ω for $|\omega| < \epsilon$, the following relations hold:*

(a)
$$\int_{-\infty}^{\infty} \delta(\omega - \varpi)X(j\omega)\,d\omega = \int_{-\infty}^{\infty} \delta(-\omega + \varpi)X(j\omega)\,d\omega \bumpeq X(j\varpi)$$

(b)
$$\delta(\omega - \varpi)X(j\omega) = \delta(-\omega + \varpi)X(j\omega) \bumpeq \delta(\omega - \varpi)X(j\varpi)$$

(c)
$$\delta(\omega)X(j\omega) = \delta(-\omega)X(j\omega) \bumpeq \delta(\omega)X(\mathbf{0}) \quad \blacktriangle$$

6.2.2 Periodic Signals

The above approach circumvents the problem of impulse functions in a practical way. However, a similar problem arises if we attempt to find the Fourier transform of a periodic signal. Consider, for example, $x(t) = \cos\omega_0 t$. We can write

$$\begin{aligned}
\mathcal{F}x(t) &= \int_{-\infty}^{\infty} (\cos\omega_0 t)e^{-j\omega t}\,dt = \int_{-\infty}^{\infty} \tfrac{1}{2}\left[e^{j\omega_0 t} + e^{-j\omega_0 t}\right]e^{-j\omega t}\,dt \\
&= \int_{-\infty}^{\infty} \tfrac{1}{2}\left[e^{j(\omega_0-\omega)t} + e^{-j(\omega_0+\omega)t}\right]dt \\
&= \int_{-\infty}^{\infty} \tfrac{1}{2}\{\cos[(\omega_0 - \omega)t] + j\sin[(\omega_0 - \omega)t] \\
&\qquad + \cos[(\omega_0 + \omega)t] - j\sin[(\omega_0 + \omega)t]\}\,dt
\end{aligned}$$

As can be seen, we have run into the same difficulty as before, that is, we are attempting to evaluate integrals of sines and cosines over the infinite range $-\infty \le t \le \infty$ and, therefore, $\mathcal{F}x(t)$ does not exist. However, this problem can also be circumvented in a practical way by simply using the transform pair in Eq. (6.16a). On applying the frequency shifting theorem (Theorem 2.10), we can write

$$i(t)e^{j\omega_0 t} \leftrightarrow 2\pi\delta(\omega - \omega_0) \tag{6.18a}$$

and

$$i(t)e^{-j\omega_0 t} \leftrightarrow 2\pi\delta(\omega + \omega_0) \tag{6.18b}$$

and since $i(t) \bumpeq 1$, we have

$$e^{j\pm\omega_0 t} \leftrightsquigarrow 2\pi\delta(\omega \mp \omega_0) \tag{6.18c}$$

If we add Eqs. (6.18a) and (6.18b), we deduce

$$i(t)[e^{j\omega_0 t} + e^{-j\omega_0 t}] \leftrightarrow 2\pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]$$

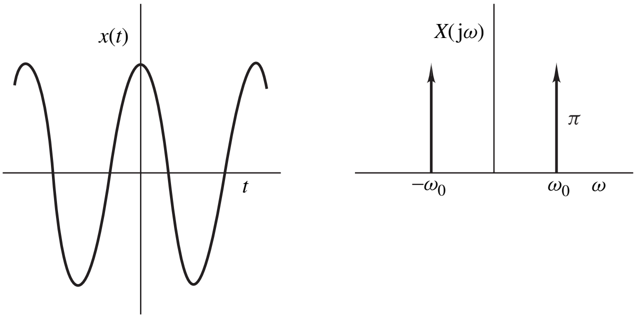


Figure 6.4 Fourier transform of cosine function: $x(t) \leftrightarrow X(j\omega)$ where $x(t) = \cos \omega_0 t$ and $X(j\omega) = \pi[\delta(\omega - \omega_0) + \delta(\omega - \omega_0)]$.

and hence

$$i(t)\cos \omega_0 t \leftrightarrow \pi[\delta(\omega + \omega_0) + \delta(\omega - \omega_0)]$$

Since $i(t) \bumpeq 1$ for $|t| < t_\infty$ (see Eq. (6.15b)), we may write

$$\cos \omega_0 t \leftrightsquigarrow \pi[\delta(\omega + \omega_0) + \delta(\omega - \omega_0)] \tag{6.19}$$

The Fourier transform of a cosine function can thus be represented by the symbolic graph of Fig. 6.4.

If we now subtract Eq. (6.18b) from Eq. (6.18a), we obtain

$$i(t)[e^{j\omega_0 t} - e^{-j\omega_0 t}] \leftrightarrow 2\pi[\delta(\omega - \omega_0) - \delta(\omega + \omega_0)]$$

and hence

$$i(t)\sin \omega_0 t \leftrightarrow j\pi[\delta(\omega + \omega_0) - \delta(\omega - \omega_0)]$$

or

$$\sin \omega_0 t \leftrightsquigarrow j\pi[\delta(\omega + \omega_0) - \delta(\omega - \omega_0)] \tag{6.20}$$

With Fourier transforms available for exponentials, sines, and cosines, Fourier transforms of arbitrary periodic signals that satisfy the Dirichlet conditions in Theorem 2.1 can be readily obtained. From Eq. (6.18a), we can write

$$i(t)X_k e^{jk\omega_0 t} \leftrightarrow 2\pi X_k \delta(\omega - k\omega_0) \tag{6.21}$$

Therefore, Eq. (2.3) gives

$$i(t)\sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t} \leftrightarrow 2\pi \sum_{k=-\infty}^{\infty} X_k \delta(\omega - k\omega_0)$$

or

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t} \leftrightsquigarrow 2\pi \sum_{k=-\infty}^{\infty} X_k \delta(\omega - k\omega_0) \tag{6.22}$$

In effect, *the frequency spectrum obtained by applying the Fourier transform to a periodic signal comprises a sequence of frequency-domain impulses whose strengths are equal to* 2π *times the Fourier series coefficients* $\{X_k\}$.

6.2.3 Unit-Step Function

Another time function that poses difficulties is the unit step $u(t)$ as can be easily shown. However, by defining the unit step in terms of a function that is absolutely integrable, the problem can be circumvented in the same way as before. We can define

$$u(t) = \lim_{\alpha \to \epsilon} \begin{cases} e^{-\alpha t} & \text{for } t > 0 \\ 0 & \text{for } t < 0 \end{cases}$$

where ϵ is a very small but finite constant. The Fourier transform for the unit step can be obtained from Example 2.6*a*, as

$$U(j\omega) = \lim_{\alpha \to \epsilon} \frac{1}{j\omega + \alpha}$$

or

$$u(t) \leftrightsquigarrow \frac{1}{j\omega} \tag{6.23}$$

The Fourier transform pairs obtained in this chapter along with those obtained in Chap. 2 are summarized in Table 6.2 for the sake of easy reference. For impulses and periodic signals, the transforms are approximate, as has been pointed out earlier, but can be made to approach any desired degree of precision by making ϵ in Eq. (6.6) smaller and ω_∞ in Eq. (6.10) or t_∞ in Eq. (6.15*b*) larger. Note that the impulse functions in Eqs. (6.6) and (6.10) would break down if we were to make ϵ zero in the first case and ω_∞ infinite in the second case but in practice there is very little to be gained in doing so. After all pulses of infinite amplitude cannot be created in the laboratory.

6.2.4 Generalized Functions

Analogous but exact Fourier transform pairs to those given by Eqs. (6.11), (6.16b), (6.18c), (6.19), (6.22), and (6.23) can be obtained but a more sophisticated definition of impulse functions is required in terms of *generalized functions*[3] as detailed by Lighthill [1]. In that approach, impulse and unity functions are defined in terms of well-behaved functions that can be differentiated any number of

[3] *See Ref. [5] for a brief introduction to generalized functions.*

Table 6.2 Standard Fourier transforms

$x(t)$	$X(j\omega)$
$\delta(t)$	1
1	$2\pi\delta(\omega)$
$\delta(t-t_0)$	$e^{-j\omega t_0}$
$e^{j\omega_0 t}$	$2\pi\delta(\omega-\omega_0)$
$\cos\omega_0 t$	$\pi[\delta(\omega+\omega_0)+\delta(\omega-\omega_0)]$
$\sin\omega_0 t$	$j\pi[\delta(\omega+\omega_0)-\delta(\omega-\omega_0)]$
$p_\tau(t)=\begin{cases}1 & \text{for } \lvert t\rvert \le \tau/2\\ 0 & \text{for } \lvert t\rvert > \tau/2\end{cases}$	$\dfrac{2\sin\omega\tau/2}{\omega}$
$\dfrac{\sin\varpi t/2}{\pi t}$	$p_\varpi(\omega)=\begin{cases}1 & \text{for } \lvert\omega\rvert \le \varpi/2\\ 0 & \text{for } \lvert\omega\rvert > \varpi/2\end{cases}$
$q_\tau(t)=\begin{cases}1-\dfrac{2\lvert t\rvert}{\tau} & \text{for } \lvert t\rvert \le \tau/2\\ 0 & \text{for } \lvert t\rvert > \tau/2\end{cases}$	$\dfrac{8\sin^2\omega\tau/4}{\tau\omega^2}$
$\dfrac{4\sin^2\varpi t/4}{\pi\varpi t^2}$	$q_\varpi(\omega)=\begin{cases}1-\dfrac{2\lvert\omega\rvert}{\varpi} & \text{for } \lvert\omega\rvert \le \varpi/2\\ 0 & \text{for } \lvert\omega\rvert > \varpi/2\end{cases}$
$e^{-\alpha t^2}$	$\sqrt{\frac{\pi}{\alpha}}e^{-\omega^2/4\alpha}$
$\dfrac{1}{\sqrt{4\alpha\pi}}e^{-t^2/4\alpha}$	$e^{-\alpha\omega^2}$
$u(t)$	$\dfrac{1}{j\omega}$
$u(t)e^{-\alpha t}$	$\dfrac{1}{a+j\omega}$
$u(t)e^{-\alpha t}\sin\omega_0 t$	$\dfrac{\omega_0}{(a+j\omega)^2+\omega_0^2}$

times, for example, in terms of exponential functions such as $\sqrt{(n/\pi)}e^{-nt^2}$ and $e^{-t^2/4n}$, respectively (see Example 2.11). It turns out that generalized functions solve one problem, namely, the limiting behavior of impulse functions, but create another: Apart from being of a somewhat abstract nature, generalized functions are also difficult, if not impossible, to realize in terms of voltage or current waveforms in the laboratory. In the practical definitions of impulse and unity functions defined in Sec. 6.2.1, the transform pairs are approximate but as parameter ϵ in Eq. (6.6) is reduced and parameter ω_∞ in Eq. (6.15c) is increased, the inexact transform pairs tend to approach their exact counterparts. In effect, *the approximate transform pairs are for all practical purposes equivalent to their exact counterparts*. In subsequent sections of this chapter and later on in the book the special symbols $\leftrightsquigarrow$ and $\triangleq$ will sometimes be replaced by the standard two-way arrow and equal to sign, respectively, for the sake of consistency with the literature but with the clear understanding that an approximation is involved as to what constitutes an impulse function.

Example 6.1 (a) Find the Fourier transform of the periodic signal

$$x(t) = \cos^4 \omega_0 t$$

(b) Repeat part (a) for the periodic signal

$$\tilde{x}(t) = \sum_{n=-\infty}^{\infty} x(t + nT)$$

where

$$x(t) = \begin{cases} \sin \omega_0 t & \text{for } 0 \le t \le \tau_0/2 \\ 0 & \text{for } -\tau_0/2 \le t \le 0 \end{cases}$$

where $\omega_0 = 2\pi/\tau_0$.

■ **Solution**

(a) We can write

$$\begin{aligned} x(t) &= (\cos^2 \omega_0 t)(\cos^2 \omega_0 t) \\ &= \tfrac{1}{4}(\cos 2\omega_0 t + 1)(\cos 2\omega_0 t + 1) \\ &= \tfrac{1}{4}(\cos^2 2\omega_0 t + 2\cos 2\omega_0 t + 1) \\ &= \tfrac{1}{4}\left[\tfrac{1}{2}(\cos 4\omega_0 t + 1) + 2\cos 2\omega_0 t + 1\right] \\ &= \tfrac{1}{8}\cos 4\omega_0 t + \tfrac{1}{2}\cos 2\omega_0 t + \tfrac{3}{8} \end{aligned}$$

Now from Table 6.2, we get

$$X(j\omega) = \pi\left\{\tfrac{1}{8}[\delta(\omega + 4\omega_0) + \delta(\omega - 4\omega_0)] + \tfrac{1}{2}[\delta(\omega + 2\omega_0) + \delta(\omega - 2\omega_0)] + \tfrac{3}{4}\delta(\omega)\right\}$$ ■

(b) The Fourier series of periodic signal $\tilde{x}(t)$ is given by Eqs. (2.3) and (2.5) where

$$\begin{aligned} X_n &= \frac{1}{\tau_0}\int_{-\tau_0/2}^{\tau_0/2} [u(t)\sin \omega_0 t]e^{-jn\omega_0 t}\,dt \\ &= \frac{1}{\tau_0}\int_{0}^{\tau_0/2} \sin \omega_0 t\,[\cos n\omega_0 t - j\sin n\omega_0 t]\,dt \\ &= \frac{1}{\tau_0}\int_{0}^{\tau_0/2} [\cos n\omega_0 t \sin \omega_0 t - j\sin n\omega_0 t \sin \omega_0 t]\,dt \end{aligned}$$

$$= \frac{1}{2\tau_0}\int_0^{\tau_0/2}\Big\{[\sin(n+1)\omega_0 t - \sin(n-1)\omega_0 t] - j[\cos(n-1)\omega_0 t + \cos(n+1)\omega_0 t]\Big\}\,dt$$

$$= \frac{1}{2\tau_0}\left[\frac{-\cos(n+1)\omega_0 t}{(n+1)\omega_0} + \frac{\cos(n-1)\omega_0 t}{(n-1)\omega_0} - j\frac{\sin(n-1)\omega_0 t}{(n-1)\omega_0} + j\frac{\sin(n+1)\omega_0 t}{(n+1)\omega_0}\right]_0^{\tau_0/2}$$

$$= \frac{1}{4\pi}\left[\frac{\cos(n-1)\pi - 1}{n-1} - \frac{\cos(n+1)\pi - 1}{n+1} - j\frac{\sin(n-1)\pi}{n-1} + j\frac{\sin(n+1)\pi}{n+1}\right]$$

$$= \frac{1}{4\pi}\left[\frac{-\cos n\pi - 1}{n-1} - \frac{-\cos n\pi - 1}{n+1} + j\frac{\sin n\pi}{n-1} - j\frac{\sin n\pi}{n+1}\right]$$

$$= -\left[\frac{\cos n\pi + 1 - j\sin n\pi}{2\pi(n^2-1)}\right]$$

Evaluating X_n and noting that l'Hôpital's rule is required for the cases $n = \pm 1$, the following values of X_n can be obtained:

$$X_0 = \frac{1}{\pi} \qquad X_1 = -X_{-1} = -\frac{j}{4} \qquad X_2 = X_{-2} = -\frac{1}{3\pi}$$

$$X_3 = X_{-3} = 0 \qquad X_4 = X_{-4} = -\frac{1}{15\pi} \qquad X_5 = X_{-5} = 0$$

$$X_6 = X_{-6} = -\frac{1}{35\pi}, \qquad \ldots$$

On using Eqs. (2.9) and (2.10), the Fourier series of $\tilde{x}(t)$ can be deduced as

$$\tilde{x}(t) = \tfrac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos n\omega_0 t + \sum_{n=1}^{\infty} b_n \sin n\omega_0 t \tag{6.24}$$

where

$$a_0 = 2X_0 \qquad a_n = X_n + X_{-n} \qquad b_n = j(X_n - X_{-n})$$

or

$$a_0 = \frac{2}{\pi} \qquad a_1 = 0 \qquad a_2 = -\frac{2}{3\pi} \qquad a_3 = 0 \qquad a_4 = -\frac{2}{15\pi}$$

$$a_5 = 0 \qquad a_6 = -\frac{2}{35\pi} \qquad a_7 = 0, \;\cdots$$

$$b_1 = \frac{1}{2} \qquad b_2 = 0 \qquad b_3 = 0 \qquad b_4 = 0 \qquad b_5 = 0 \qquad b_6 = 0, \;\cdots$$

Now from Table 6.2, we get

$$\mathcal{F}\tilde{x}(t) = a_0\pi\delta(\omega) + \sum_{k=1}^{\infty} a_n\pi[\delta(\omega + n\omega_0) + \delta(\omega - n\omega_0)]$$

$$+ \sum_{k=1}^{\infty} jb_n\pi[\delta(\omega + n\omega_0) - \delta(\omega - n\omega_0)] \quad \blacksquare \tag{6.25}$$

6.3 INTERRELATION BETWEEN THE FOURIER SERIES AND THE FOURIER TRANSFORM

Discrete-time signals are usually sampled versions of continuous-time signals and, therefore, it stands to reason that they inherit their spectral characteristics from the continuous-time signals from which they are derived. Specifically, if the frequency spectrum of the underlying continuous-time signal is known, then that of the discrete-time signal can be deduced by using Poisson's summation formula. The following theorem is prerequisite for the derivation of this most important formula.

Theorem 6.2 Fourier-Series Kernel Theorem

$$\sum_{n=-\infty}^{\infty} \delta(t - nT) \leftrightarrow \omega_s \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_s) \tag{6.26}$$

where $\omega_s = 2\pi/T$.

The relation in Eq. (6.26) can be demonstrated to be valid on the basis of the principles developed in Sec. 6.2. To start with, on applying the inverse Fourier transform to the right-hand side of Eq. (6.26), we get

$$\begin{aligned}\mathcal{F}^{-1}\left[\omega_s \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_s)\right] &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\omega_s \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_s)\right] e^{j\omega t}\,d\omega \\ &= \frac{1}{T}\sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(\omega - n\omega_s)e^{j\omega t}\,d\omega\end{aligned}$$

and from Theorem 6.1B, part (*a*), we have

$$\mathcal{F}^{-1}\left[\omega_s \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_s)\right] \doteq \frac{1}{T}\sum_{n=-\infty}^{\infty} e^{jn\omega_s t} \tag{6.27}$$

Consider the so-called *Fourier-series kernel* (see Papoulis [4], pp. 42–45), which is defined as

$$k_{N_\infty}(t) = \frac{1}{T}\sum_{n=-N_\infty}^{N_\infty} e^{jn\omega_s t} \tag{6.28}$$

where N_∞ is a finite integer. Since this is a geometric series with common ratio $e^{j\omega_s t}$, its sum can be obtained as

$$\begin{aligned}
k_{N_\infty}(t) &= \frac{1}{T}\sum_{n=-N_\infty}^{N_\infty} e^{jn\omega_s t} = \frac{1}{T}\frac{e^{j(N_\infty+1)\omega_s t} - e^{-jN_\infty\omega_s t}}{e^{j\omega_s t} - 1}\\
&= \frac{1}{T}\frac{e^{j(2N_\infty+1)\omega_s t/2} - e^{-j(2N_\infty+1)\omega_s t/2}}{e^{j\omega_s t/2} - e^{-j\omega_s t/2}}\\
&= \frac{1}{T}\frac{\sin[(2N_\infty+1)\omega_s t/2]}{\sin(\omega_s t/2)}
\end{aligned}$$

(see Eq. (A.46b)). We can write

$$k_{N_\infty}(t) = \frac{1}{T}\frac{\pi t}{\sin(\omega_s t/2)} \cdot \frac{\sin[(2N_\infty+1)\omega_s t/2]}{\pi t}$$

and if we let $(2N_\infty + 1)\omega_s = \omega_\infty$, then

$$k_{N_\infty}(t) = \frac{1}{T}\frac{\pi t}{\sin(\omega_s t/2)} \cdot \frac{\sin \omega_\infty t/2}{\pi t}$$

If $N_\infty \ggg 1$, $(\sin \omega_\infty t/2)/\pi t$ behaves as a time-domain impulse function (see Table 6.1) and hence for $-T/2 < t < T/2$, $k_{N_\infty}(t)$ can be expressed as

$$k_{N_\infty}(t) = \xi(t)\delta(t)$$

where function

$$\xi(t) = \frac{1}{T}\frac{\pi t}{\sin(\omega_s t/2)}$$

is continuous and assumes the value of unity at $t = 0$. Now from Theorem 6.1A, part (*c*), we get

$$k_{N_\infty}(t) = \xi(t)\delta(t) \doteq \xi(0)\delta(t) = \delta(t)$$

At this point, if we let $t = t + nT$ in $k_{N_\infty(t)}$, we can easily verify that the Fourier-series kernel is periodic with period T, as illustrated in Fig. 6.5 (see Prob. 6.14, part (*a*)). Therefore, for $N_\infty \ggg 1$, it behaves as an infinite series of impulse functions located at $t = 0, \pm T, \pm 2T, \ldots, \pm nT, \ldots$ and from Eq. (6.28), we can write

$$k_{N_\infty}(t) = \frac{1}{T}\sum_{n=-N_\infty}^{N_\infty} e^{jn\omega_s t} \doteq \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{6.29}$$

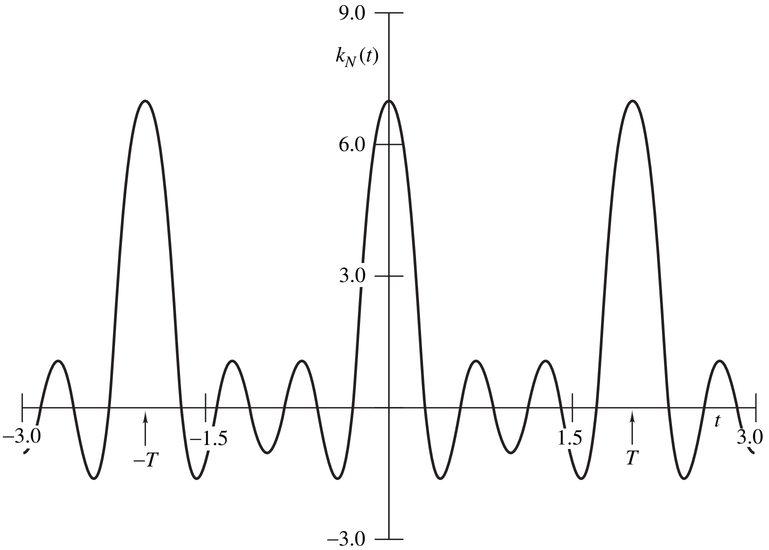


Figure 6.5 Fourier-series kernel.

Since $N_\infty \ggg 1$, Eqs. (6.27)–(6.29) yield

$$\mathcal{F}^{-1}\left[\omega_s \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_s)\right] \doteq \frac{1}{T}\sum_{n=-\infty}^{\infty} e^{jn\omega_s t} \doteq \frac{1}{T}\sum_{n=-N_\infty}^{N_\infty} e^{jn\omega_s t}$$

$$\doteq \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

Impulse functions as defined in Sec. 6.2.1 are absolutely integrable and hence they satisfy the convergence theorem of the Fourier transform (Theorem 2.5). We thus conclude that

$$\sum_{n=-\infty}^{\infty} \delta(t - nT) \leftrightsquigarrow \omega_s \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_s) \quad \blacktriangle$$

An exact version of the above result can be obtained through the use of generalized functions [5].

The Fourier-series kernel theorem (Theorem 6.2) leads to a direct relationship between the Fourier series and the Fourier transform. This relationship is stated in the following theorem:

Theorem 6.3 Interrelation Between the Fourier Series and the Fourier Transform ***Given a nonperiodic signal $x(t)$ with a Fourier transform $X(j\omega)$, a periodic signal with period T can be constructed as***

$$\tilde{x}(t) = \sum_{n=-\infty}^{\infty} x(t + nT) \tag{6.30}$$

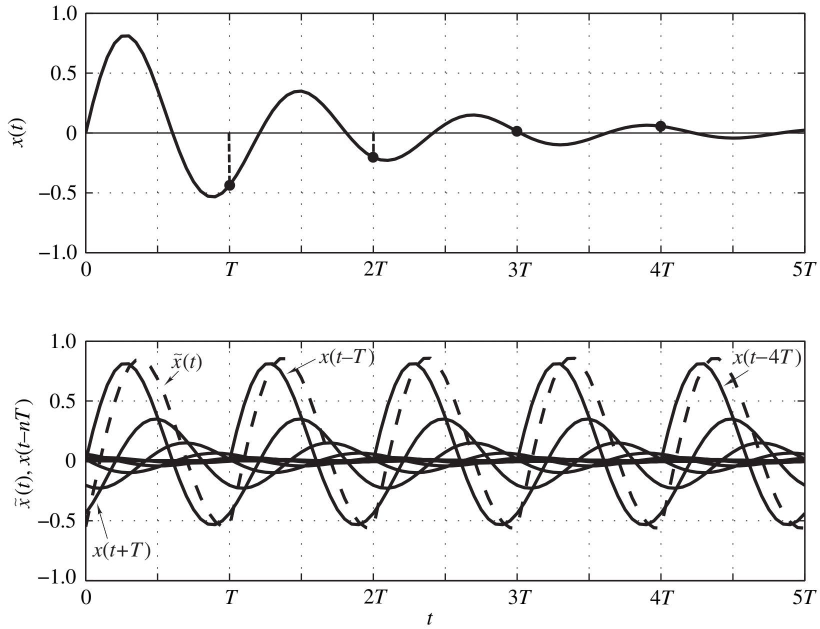


Figure 6.6 Generation of periodic signal $\tilde{x}(t)$ through the addition of an infinite number of shifted copies of $x(t)$ over the range $-\infty < t < \infty$.

(see Fig. 6.6 and Prob. 6.14, part (b)). The Fourier series coefficients of $\tilde{x}(t)$ are given by

$$X_n = \frac{X(jn\omega_s)}{T} \quad \blacktriangle \tag{6.31}$$

where

$$X(j\omega) = \mathcal{F}x(t)$$

The above theorem states, in effect, that *the Fourier series coefficient of the nth harmonic of periodic signal $\tilde{x}(t)$ is numerically equal to the Fourier transform of $x(t)$ evaluated at the frequency of the harmonic divided by T*.

The validity of the relationship in Eq. (6.31) can be demonstrated by using our practical approach to impulse and unity functions as described in Sec. 6.2.1. From Eq. (6.22), the Fourier transform of a periodic signal $\tilde{x}(t)$ is given by

$$\tilde{X}(j\omega) \bumpeq 2\pi \sum_{n=-\infty}^{\infty} X_n \delta(\omega - n\omega_s) \tag{6.32}$$

From Theorem 6.1A, part (*a*), Eq. (6.30) can be expressed as

$$\begin{aligned}
\tilde{x}(t) &= \sum_{n=-\infty}^{\infty} x(t+nT) \\
&\doteq \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau+nT)\,d\tau \\
&\doteq \int_{-\infty}^{\infty} x(\tau) \sum_{n=-\infty}^{\infty} \delta(t-\tau+nT)\,d\tau \\
&\doteq x(t) \otimes \sum_{n=-\infty}^{\infty} \delta(t-nT) \qquad (6.33)
\end{aligned}$$

where the last two lines represent time convolution (see Theorem 2.14), and on using Theorem 6.2 and Eq. (6.33), we obtain

$$\begin{aligned}
\tilde{X}(j\omega) &\doteq \mathcal{F}x(t) \cdot \mathcal{F} \sum_{n=-\infty}^{\infty} \delta(t-nT) \\
&\doteq X(j\omega) \cdot \omega_s \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_s) \\
&\doteq 2\pi \sum_{n=-\infty}^{\infty} \frac{X(j\omega)}{T} \delta(\omega - n\omega_s) \qquad (6.34)
\end{aligned}$$

If we now use Theorem 6.1B, part (*b*), Eq. (6.34) yields

$$\tilde{X}(j\omega) \doteq 2\pi \sum_{n=-\infty}^{\infty} \frac{X(jn\omega_s)}{T} \delta(\omega - n\omega_s) \qquad (6.35)$$

and on comparing Eqs. (6.32) and (6.35), we deduce

$$X_n \doteq \frac{X(jn\omega_s)}{T} \qquad \blacktriangle$$

It should be mentioned at this point that Eq. (6.31) holds independently of the values of $x(t)$ for $|t| > T/2$. If $x(t) = 0$ for $|t| > T/2$, the shifted copies of $x(t)$ do not overlap and

$$\tilde{x}(t) = x(t) \qquad \text{for } |t| < T/2$$

whereas if $x(t) \neq 0$ for $|t| > T/2$, they do overlap and so

$$\tilde{x}(t) \neq x(t) \qquad \text{for } |t| < T/2$$

In the latter case, $\tilde{x}(t)$ is said to be an aliased version of $x(t)$. For the nonaliased case, the Fourier series coefficients give one spectral representation for a periodic signal and the Fourier transform gives another, which are interrelated through Eq. (6.31).

Example 6.2 Given the nonperiodic signal

$$x(t) = p_{\tau/2}\left(t + \tfrac{1}{4}\tau\right) - p_{\tau/2}\left(t - \tfrac{1}{4}\tau\right)$$

where

$$p_{\tau/2} = \begin{cases} 1 & \text{for } |t| < \tau/4 \\ 0 & \text{otherwise} \end{cases}$$

a periodic signal $\tilde{x}(t)$ with period T such as that in Eq. (6.30) can be constructed. Show that the Fourier series coefficients of $\tilde{x}(t)$ are related to the Fourier transform of $x(t)$ through the relation in Eq. (6.31).

■ **Solution**

The Fourier series coefficients of $\tilde{x}(t)$ can be obtained from Example 2.3 as

$$X_n = \begin{cases} 0 & \text{for } n = 0 \\ j\dfrac{4\sin^2 n\omega_s\tau/4}{n\omega_s T} & \text{for } n = 1,\ 2,\ \ldots \end{cases} \tag{6.36}$$

by noting that $k = n$, $\tau_0 = T$, and $\omega_0 = \omega_s = 2\pi/T$ in the present context. From the definition of the Fourier transform, we can write

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt$$

and since $x(t)$ is an odd function of t, Eqs. (2.37a)–(2.37b) give

$$X(j\omega) = \Re e\, X(j\omega) + j\,\Im m\, X(j\omega) \tag{6.37a}$$

where

$$\Re e\, X(j\omega) = 0 \tag{6.37b}$$

and

$$\begin{aligned}\Im m\, X(j\omega) &= -2\int_0^{\infty} x(t)\sin\omega t\, dt\\ &= -2\int_0^{\tau/2} -\sin\omega t\, dt\\ &= 2\left[\frac{\cos\omega t}{\omega}\right]_0^{\tau/2}\\ &= 2\left[\frac{-\cos\omega\tau/2+1}{\omega}\right]_0^{\tau/2}\\ &= \frac{4\sin^2\omega\tau/4}{\omega} \qquad (6.37c)\end{aligned}$$

Hence Eqs. (6.37a) and (6.37c) give

$$X(j\omega) = j\frac{4\sin^2\omega\tau/4}{\omega}$$

where $X(0) = 0$ as can be readily verified. If we let $\omega = n\omega_s$, we get

$$X(jn\omega_s) = \begin{cases} 0 & \text{for } n = 0\\ j\dfrac{4\sin^2 n\omega_s\tau/4}{n\omega_s} & \text{for } n = 1,\ 2,\ \ldots\end{cases} \qquad (6.38)$$

Now on comparing Eqs. (6.36) and (6.38), we note that Theorem 6.3 is satisfied. ■

6.4 POISSON'S SUMMATION FORMULA

Given an arbitrary nonperiodic signal $x(t)$ that has a Fourier transform, the periodic signal in Eq. (6.30) can be immediately constructed. Such a signal has a Fourier series of the form

$$\tilde{x}(t) = \sum_{n=-\infty}^{\infty} x(t+nT) = \sum_{n=-\infty}^{\infty} X_n e^{jn\omega_s t} \qquad (6.39)$$

Now from Eqs. (6.31) and (6.39), we obtain

$$\sum_{n=-\infty}^{\infty} x(t+nT) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(jn\omega_s)e^{j\omega_s t} \qquad (6.40)$$

This relationship is known as *Poisson's summation formula* and as will be shown below, it provides a crucial link between the frequency spectrum of a discrete-time signal and that of the underlying continuous-time signal.

Two special cases of Poisson's formula are of interest. If $x(t)$ assumes nonzero values for $t < 0$, then if we let $t = 0$ in Eq. (6.40), we obtain

$$\sum_{n=-\infty}^{\infty} x(nT) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(jn\omega_s) \tag{6.41a}$$

On the other hand, if $x(t) = 0$ for $t < 0$, then

$$\lim_{t\to 0} x(t) + \sum_{n=1}^{\infty} x(nT) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(jn\omega_s) \tag{6.41b}$$

Now the Fourier series also holds at a discontinuity provided that the value of the periodic signal at the discontinuity is deemed to be

$$\lim_{t\to 0} x(t) = \frac{x(0-) + x(0+)}{2}$$

(see Theorem 2.1) and since $x(0-) = 0$ in the present case, Eq. (6.41b) assumes the form

$$\frac{x(0+)}{2} + \sum_{n=1}^{\infty} x(nT) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(jn\omega_s)$$

or

$$\sum_{n=0}^{\infty} x(nT) = \frac{x(0+)}{2} + \frac{1}{T} \sum_{n=-\infty}^{\infty} X(jn\omega_s) \tag{6.41c}$$

where $x(0) \equiv x(0+)$.

Poisson's summation formula is illustrated in Fig. 6.7 for the signal

$$x(t) = u(t)e^{-at} \sin \omega t$$

with $a = 0.35$ and $\omega = 2.6$. This important formula states, in effect, that *the sum of the signal values of $x(t)$ at $t = nT$ in Fig. 6.7a over the range $-\infty < t < \infty$ is equal to the sum of the complex values*

$$X(jn\omega_s) = |X(jn\omega_s)|e^{j \arg X(jn\omega_s)}$$

in Fig. 6.7b for $-\infty < n < \infty$ divided by the sampling period T.

As an aside, note that there is only one term in the time-domain summations in Eqs. (6.41a) and (6.41c) if $x(t) = 0$ for $|t| > T/2$, and hence we have

$$x(0) = \begin{cases} \frac{1}{T} \sum_{n=-\infty}^{\infty} X(jn\omega_s) & \text{if } x(t) = 0 \text{ for } t < -T/2 \text{ and } t > T/2 \\ \frac{x(0+)}{2} + \frac{1}{T} \sum_{n=-\infty}^{\infty} X(jn\omega_s) & \text{if } x(t) = 0 \text{ for } t < 0 \text{ and } t > T/2 \end{cases}$$

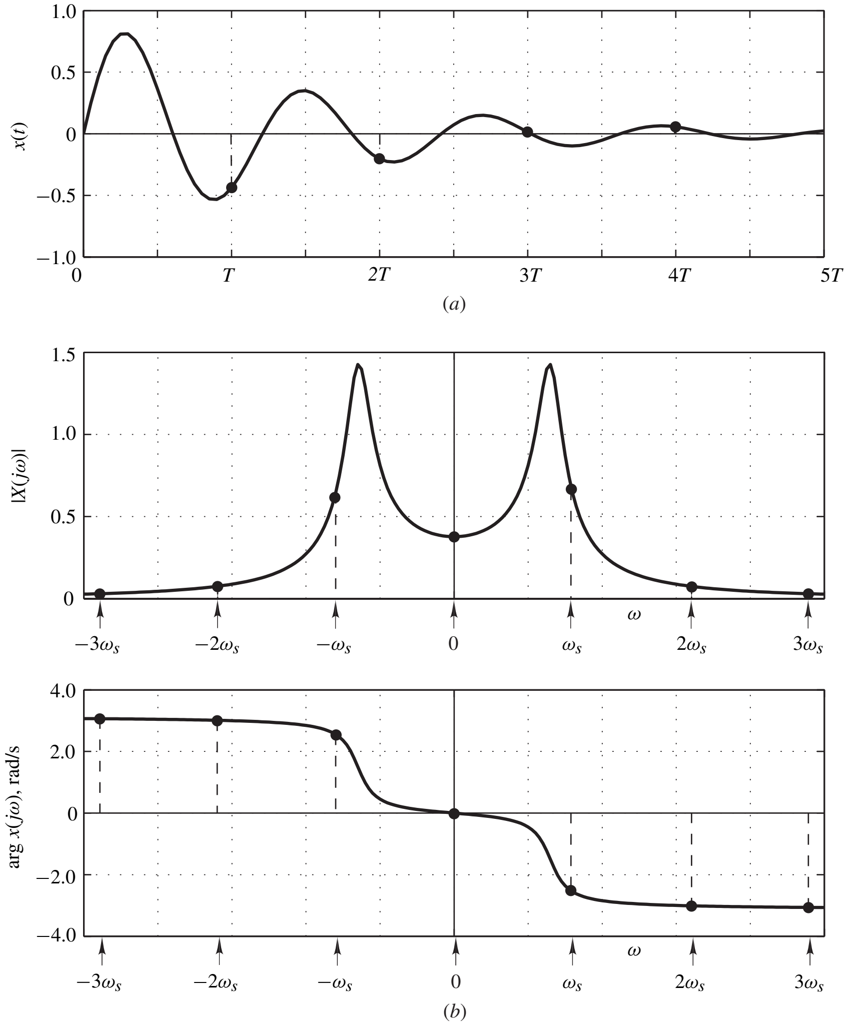


Figure 6.7 Poisson summation formula for the case where $x(t)$ is defined over the range $-\infty < t < \infty$: (*a*) Time domain, (*b*) frequency domain.

6.5 IMPULSE-MODULATED SIGNALS

An *impulse-modulated signal*, denoted as $\hat{x}(t)$, can be generated by sampling a continuous-time signal $x(t)$ using an impulse modulator as illustrated in Fig. 6.8*a*. An *impulse modulator* is essentially a subsystem whose response to an input $x(t)$ is given by

$$\hat{x}(t) = c(t)x(t) \tag{6.42a}$$

where $c(t)$ is a carrier signal of the form

$$c(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{6.42b}$$

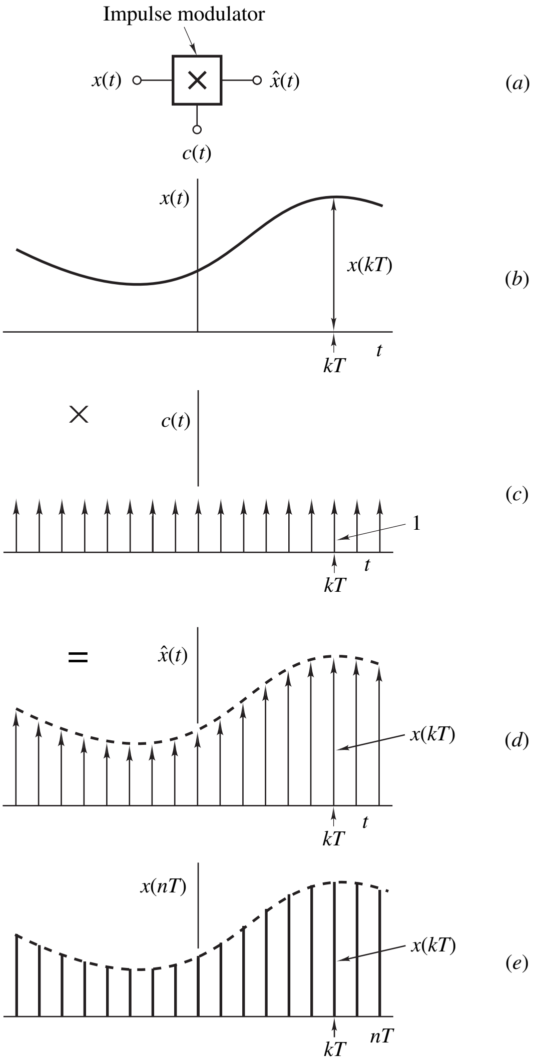


Figure 6.8 Generation of an impulse-modulated signal: (*a*) Ideal impulse modulator, (*b*) continuous-time signal, (*c*) impulse-modulated carrier, (*d*) impulse-modulated signal $\hat{x}(t)$, (*d*) discrete-time signal $x(nT)$.

(see Fig. 6.8c). From Eqs. (6.42a) and (6.42b), we have

$$\begin{aligned}\hat{x}(t) &= x(t)\sum_{n=-\infty}^{\infty}\delta(t-nT)\\ &= \sum_{n=-\infty}^{\infty}x(t)\delta(t-nT)\end{aligned}\tag{6.42c}$$

and if we apply Theorem 6.1A, part (*b*), to Eq. (6.42c), we obtain

$$\hat{x}(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT) \tag{6.42d}$$

Often $x(t) = 0$ for $t \leq 0-$. In such a case Eq. (6.42d) assumes the form

$$\hat{x}(t) = \sum_{n=0}^{\infty} x(nT)\delta(t - nT) \tag{6.42e}$$

In effect, an impulse-modulated signal is a sequence of continuous-time impulses, like that illustrated in Fig. 6.8*d*. A signal of this type can be converted into a discrete-time signal by simply replacing each impulse of strength $x(nT)$ by a number $x(nT)$ as shown in Fig. 6.8*e*.

6.5.1 Interrelation Between Fourier and z Transforms

Observe that an impulse-modulated signal is both a sampled as well as a continuous-time signal and this dual personality will immediately prove very useful. To start with, since it is continuous in time, it has a Fourier transform, that is,

$$\hat{X}(j\omega) = \mathcal{F} \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT) = \sum_{n=-\infty}^{\infty} x(nT)\mathcal{F}\delta(t - nT) \tag{6.43a}$$

Clearly

$$\hat{X}(j\omega) = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT} = X_D(z)\Big|_{z=e^{j\omega T}} \tag{6.43b}$$

where

$$X_D(z) = \mathcal{Z}x(nT)$$

For a right-sided signal, Eq. (6.43b) assumes the form

$$\hat{X}(j\omega) = \sum_{n=0}^{\infty} x(nT)e^{-j\omega nT} = X_D(z)\Big|_{z=e^{j\omega T}} \tag{6.43c}$$

The above analysis has shown that *the Fourier transform of an impulse-modulated signal $\hat{x}(t)$ is numerically equal to the z transform of the corresponding discrete-time signal $x(nT)$ evaluated on the unit circle $|z| = 1$.* In other words, the frequency spectrum of $\hat{x}(t)$ is equal to that of $x(nT)$.

Example 6.3 (*a*) The continuous-time signal

$$x(t) = \begin{cases} 0 & \text{for } t < 3.5 \text{ s} \\ 1 & \text{for } -3.5 \le t < -2.5 \\ 2 & \text{for } -2.5 \le t < 2.5 \\ 1 & \text{for } 2.5 \le t \le 3.5 \\ 0 & \text{for } t > 3.5 \end{cases}$$

is subjected to impulse modulation. Find the frequency spectrum of $\hat{x}(t)$ in closed form assuming a sampling frequency of 2π rad/s. (*b*) Repeat part (*a*) for the signal

$$x(t) = u(t)e^{-t}\sin 2t$$

assuming a sampling frequency of 2π rad/s.

■ **Solution**

(*a*) The frequency spectrum of an impulse-modulated signal, $\hat{x}(t)$, can be readily obtained by evaluating the z transform of $x(nT)$ on the unit circle of the z plane. The impulse-modulated version of $x(t)$ can be expressed as

$$\begin{aligned} \hat{x}(t) &= \delta(t+3T) + 2\delta(t+2T) + 2\delta(t+T) + 2\delta(0) \\ &\quad +2\delta(t-T) + 2\delta(t-2T) + \delta(t-3T) \end{aligned}$$

where $T = 1$ s. A corresponding discrete-time signal can be obtained by replacing impulses by numbers as

$$\begin{aligned} x(nT) &= \delta(nT+3T) + 2\delta(nT+2T) + 2\delta(nT+T) + 2\delta(0) \\ &\quad +2\delta(nT-T) + 2\delta(nT-2T) + \delta(nT-3T) \end{aligned}$$

Hence

$$X_D(z) = \mathcal{Z}x(t) = z^3 + 2z^2 + 2z^1 + 2 + 2z^{-1} + 2z^{-2} + z^{-3}$$

and, therefore, from Eq. (6.43b)

$$\begin{aligned} \hat{X}(j\omega) = X_D(e^{j\omega T}) &= (e^{j3\omega T} + e^{-j3\omega T}) + 2(e^{j2\omega T} + e^{-j2\omega T}) \\ &\quad + 2(e^{j\omega T} + e^{-j\omega T}) + 2 \\ &= 2\cos 3\omega T + 4\cos 2\omega T + 4\cos \omega T + 2 \quad \blacksquare \end{aligned}$$

(*b*) A discrete-time signal can be readily derived from $x(t)$ by replacing t by nT as

$$x(nT) = u(nT)e^{-nT}\sin 2nT = u(nT)e^{-nT} \times \frac{1}{2j}\left(e^{j2nT} - e^{-j2nT}\right)$$

$$= u(nT)\frac{1}{2j}\left(e^{nT(-1+j2)} - e^{nT(-1-j2)}\right)$$

Since $T = 2\pi/\omega_s = 1$ s, Table 3.2 gives

$$X_D(z) = \frac{1}{2j}\left(\frac{z}{z - e^{-1+j2}} - \frac{z}{z - e^{-1-j2}}\right)$$

and after some manipulation

$$X_D(z) = \frac{ze^{-1}\sin 2}{z^2 - 2ze^{-1}\cos 2 + e^{-2}}$$

Therefore, the frequency spectrum of the impulse-modulated signal is given by

$$\hat{X}(j\omega) = X_D(e^{j\omega T}) = \frac{e^{j\omega - 1}\sin 2}{e^{2j\omega} - 2e^{j\omega-1}\cos 2 + e^{-2}} \qquad \blacksquare$$

6.5.2 Spectral Interrelation Between Discrete- and Continuous-Time Signals

Let $X(j\omega)$ be the Fourier transform of $x(t)$. From the frequency-shifting theorem of the Fourier transform (Theorem 2.10), the transform pair

$$x(t)e^{-j\omega_0 t} \leftrightarrow X(j\omega_0 + j\omega)$$

can be formed. On using Poisson's summation formula given by Eq. (6.41a), we get

$$\sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega_0 nT} = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(j\omega_0 + jn\omega_s)$$

where $\omega_s = 2\pi/T$ and if we now replace ω_0 by ω, we obtain

$$\sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT} = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(j\omega + jn\omega_s) \tag{6.44}$$

Therefore, from Eqs. (6.43b) and (6.44), we deduce

$$\hat{X}(j\omega) = X_D(e^{j\omega T}) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(j\omega + jn\omega_s) \tag{6.45a}$$

Similarly, for a right-sided signal, the use of Eq. (6.41c) in the above analysis along with Eq. (6.43c) gives

$$\hat{X}(j\omega) = X_D(e^{j\omega T}) = \frac{x(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} X(j\omega + jn\omega_s) \tag{6.45b}$$

that is, *the frequency spectrum of the impulse-modulated signal $\hat{x}(t)$ is equal to the frequency spectrum of discrete-time signal $x(nT)$ and the two can be uniquely determined from the frequency spectrum of the continuous-time signal $x(t)$, namely, $X(j\omega)$.*

As is to be expected, $\hat{X}(j\omega)$ is a *periodic function* of ω with period ω_s since the frequency spectrum of discrete-time signals is periodic as shown in Sec. 3.9.2. Indeed, if we replace $j\omega$ by $j\omega + jm\omega_s$ in Eq. (6.45a), we get

$$\begin{aligned}\hat{X}(j\omega + jm\omega_s) &= \frac{1}{T}\sum_{n=-\infty}^{\infty} X[j\omega + j(m+n)\omega_s]\\ &= \frac{1}{T}\sum_{n'=-\infty}^{\infty} X(j\omega + jn'\omega_s)\\ &= \hat{X}(j\omega)\end{aligned}$$

The above relationships can be extended to the s domain. By virtue of a principle of complex analysis known as *analytic continuation* (see Sec. A.8), given a Fourier transform $F(j\omega)$, the *Laplace transform* $F(s)$ can be obtained by replacing $j\omega$ by s in $F(j\omega)$, that is,

$$F(s) = F(j\omega)\Big|_{j\omega=s}$$

(See Sec. 10.2.2 for a more detailed description of the Laplace transform.) Thus if we let $j\omega = s$ and $e^{sT} = z$, Eqs. (6.45a) and (6.45b) assume the forms

$$\hat{X}(s) = X_D(z) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(s + jn\omega_s) \tag{6.46a}$$

and

$$\hat{X}(s) = X_D(z) = \frac{x(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} X(s + jn\omega_s) \tag{6.46b}$$

where $X(s)$ and $\hat{X}(s)$ are the Laplace transforms of $x(t)$ and $\hat{x}(t)$, respectively. If the value of $x(0+)$ is not available, it can be deduced from $X(s)$ as

$$x(0+) = \lim_{s\to\infty} [sX(s)]$$

by using the initial-value theorem of the one-sided Laplace transform [3] (see Sec. 10.2.4).

The relationship in Eq. (6.46b) turns out to be of significant practical importance. It will be used in Sec. 6.9 to establish a relationship between analog and digital filters. This relationship is the basis of the so-called *invariant impulse-response method* for the design of IIR filters described in Chap. 11.

Example 6.4 (*a*) Using Poisson's summation formula, obtain $\hat{X}(j\omega)$ if $x(t) = \cos\omega_0 t$. (*b*) Repeat part (*a*) for $x(t) = u(t)e^{-t}$.

■ **Solution**

(*a*) From Table 6.2

$$X(j\omega) = \mathcal{F}\cos\omega_0 t = \pi[\delta(\omega + \omega_0) + \delta(\omega - \omega_0)]$$

Hence Eq. (6.45a) gives

$$\hat{X}(j\omega) = \frac{\pi}{T}\sum_{n=-\infty}^{\infty} [\delta(\omega + n\omega_s + \omega_0) + \delta(\omega + n\omega_s - \omega_0)] \quad ■$$

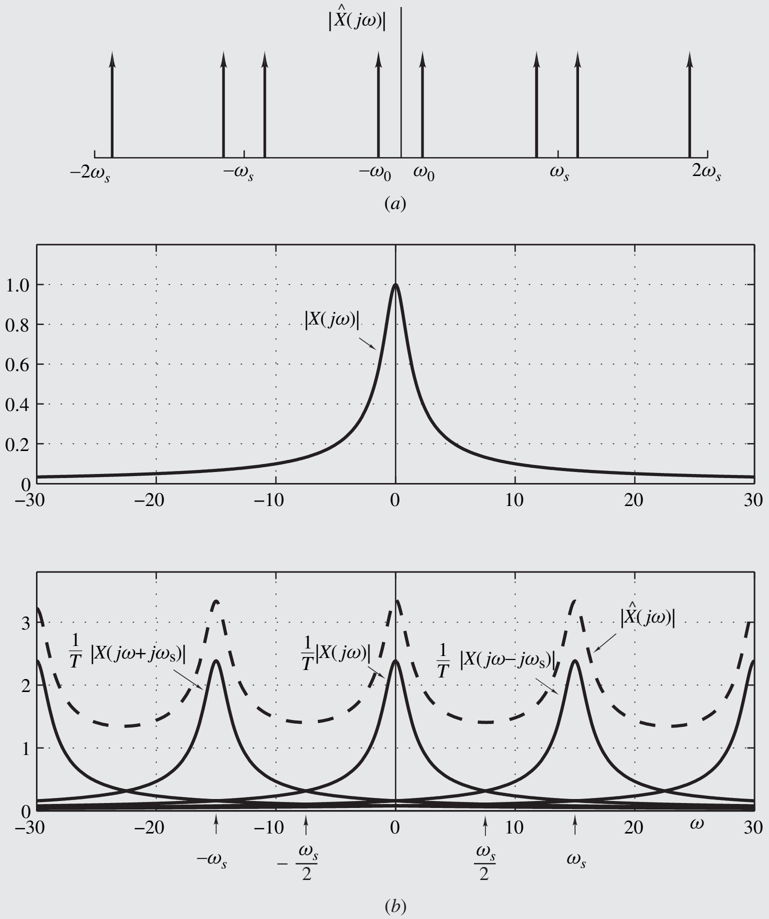
The amplitude spectrum of $\hat{x}(t)$ is illustrated in Fig. 6.9*a*.

(*b*) From Table 6.2, we have

$$X(j\omega) = \mathcal{F}[u(t)e^{-t}] = \frac{1}{1 + j\omega}$$

Since

$$x(0+) = \lim_{t\to 0}[u(t)e^{-t}] = 1$$



(*a*)

(*b*)

Figure 6.9 Amplitude spectrum of $\hat{x}(t)$: (*a*) Example 6.4*a*, (*b*) Example 6.4*b*.

Eq. (6.45b) gives

$$\hat{X}(j\omega) = \frac{1}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} \frac{1}{1 + j(\omega + n\omega_s)} \quad \blacksquare$$

The amplitude spectrum of $\hat{x}(t)$ is plotted in Fig. 6.9*b* for a sampling frequency $\omega_s =$ 15 rad/s.

6.6 THE SAMPLING THEOREM

The application of digital filters for the processing of continuous-time signals is made possible by the *sampling theorem*[4] which is as follows:

Theorem 6.4 Sampling Theorem ***A bandlimited signal $x(t)$ for which***

$$X(j\omega) = 0 \qquad \textit{for } |\omega| \geq \frac{\omega_s}{2} \tag{6.47}$$

where $\omega_s = 2\pi/T$, can be uniquely determined from its values $x(nT)$. ▲

The validity of the sampling theorem can be demonstrated by showing that a bandlimited signal $x(t)$ can be recovered from an impulse-modulated version of the signal, $\hat{x}(t)$, by using an ideal lowpass filter as depicted in Fig. 6.10.[5]

Assume that $x(t)$ is bandlimited and that the sampling frequency ω_s is high enough to ensure that the condition in Eq. (6.47) is satisfied. The frequency spectrum of such a signal could assume the form depicted in Fig. 6.11*a*. Poisson's summation formula in Eq. (6.45*a*) gives the frequency spectrum of the impulse modulated signal $\hat{x}(t)$ as

$$\hat{X}(j\omega) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(j\omega + jn\omega_s)$$

Evidently, the spectrum of $\hat{x}(t)$ can be derived from that of $x(t)$ through a process of periodic continuation whereby exact copies of the spectrum of $x(t)/T$ are shifted by frequencies $\{\cdots, -2\omega_s, -\omega_s, \omega_s, 2\omega_s, \cdots\}$ and are then added. If $x(t)$ satisfies the condition in Eq. (6.47), then the shifted copies of the spectrum, often referred to as *sidebands*, would not overlap and, consequently, the spectrum of $\hat{x}(t)$ would assume the form depicted in Fig. 6.11*b*. If the impulse-modulated signal is now passed through an ideal lowpass filter with cutoff frequencies at $\pm\omega_s/2$ as illustrated in Fig. 6.11c, all the sidebands would be rejected and the spectrum of the filter output would be an

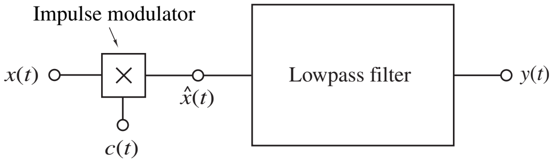


Figure 6.10 Sampling theorem: Derivation of $x(t)$ from $\hat{x}(t)$ by using a lowpass filter.

[4] *The sampling theorem is attributed to Nyquist, Shannon, or both, depending on what one reads. In actual fact, the historical record shows that both of these individuals made a significant contribution to the sampling theorem. Nyquist provided an intuitive derivation of the sampling theorem as early as 1928 in Ref. [6] whereas Shannon provided a rigorous proof for it in Ref. [7].*

[5] *See Sec. 10.2 for a brief summary of the basics of analog filters.*

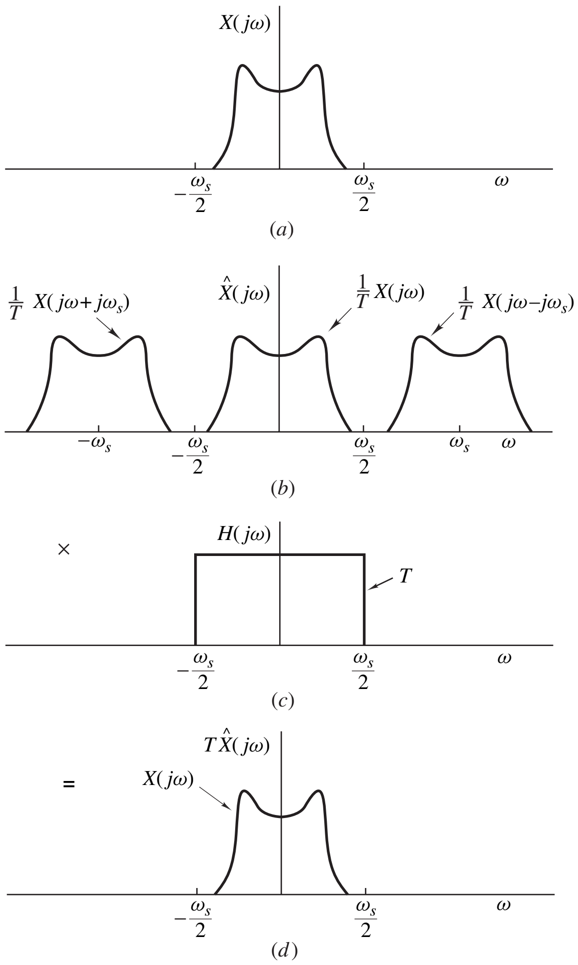


Figure 6.11 Sampling theorem—derivation of $x(t)$ from $\hat{x}(t)$ by using a lowpass filter: (a) $X(j\omega)$, (b) $\hat{X}(j\omega)$, (c) frequency response of ideal lowpass filter, (d) lowpass-filtered version of $\hat{X}(j\omega)$.

exact copy of the spectrum of the continuous-time signal, that is, the continuous-time signal will be recovered, as shown in Fig. 6.11d.

The above thought experiment can be repeated through analysis. Consider a lowpass filter with a frequency response

$$H(j\omega) = \begin{cases} T & \text{for } |\omega| < \omega_s/2 \\ 0 & \text{for } |\omega| \geq \omega_s/2 \end{cases}$$

such as that illustrated in Fig. 6.11c. The frequency spectrum of the filter output is given by

$$X(j\omega) = H(j\omega)\hat{X}(j\omega) \tag{6.48}$$

(see Eq. (10.6a)). Thus from Eqs. (6.43b) and (6.48), we can write

$$X(j\omega) = H(j\omega) \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}$$

and hence

$$\begin{aligned} x(t) &= \mathcal{F}^{-1}\left[H(j\omega) \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}\right] \\ &= \sum_{n=-\infty}^{\infty} x(nT)\mathcal{F}^{-1}[H(j\omega)e^{-j\omega nT}] \end{aligned} \tag{6.49}$$

The frequency response of the lowpass filter is actually a frequency-domain pulse of height T and base ω_s, that is, $H(j\omega) = Tp_{\omega_s}(\omega)$ as shown in Fig. 6.11*c* and hence from Table 6.2, we have

$$\frac{T\,\sin(\omega_s t/2)}{\pi t} \leftrightarrow H(j\omega)$$

and from the time-shifting theorem of the Fourier transform (Theorem 2.9), we obtain

$$\frac{T\,\sin[\omega_s(t-nT)/2]}{\pi(t-nT)} \leftrightarrow H(j\omega)e^{-j\omega nT} \tag{6.50}$$

Therefore, from Eqs. (6.49) and (6.50), we conclude that

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT)\frac{\sin[\omega_s(t-nT)/2]}{\omega_s(t-nT)/2} \tag{6.51}$$

For an ideal lowpass filter, the frequency spectrum in Fig. 6.11*d* is exactly the same as that in Fig. 6.11*a* and thus the output of the ideal filter in Fig. 6.10 must be $x(t)$. In effect, Eq. (6.51) is an interpolation formula that can be used to determine signal $x(t)$ from its values $x(nT)$. That this, indeed, is the case, we note that the right-hand side in Eq. (6.51) assumes the values of $x(t)$ for $-\infty \le n \le \infty$ if $t = nT$ since $\lim_{x\to 0} \sin x/x = 1$.

Note that the above analysis provides the standard method for the reconstruction of the original signal from an impulse-modulated version of the signal and, as will be shown below, it can also be used to reconstruct the continuous-time signal from a discrete-time version.

6.7 ALIASING

If

$$X(j\omega) \neq 0 \qquad \text{for } |\omega| \ge \frac{\omega_s}{2}$$

as in Fig. 6.12*a*, for example, frequencies pertaining to the shifted copies will move into the baseband of $X(j\omega)$ as depicted in Fig. 6.12*b*. As a result, $\hat{X}(j\omega)$ (dashed curve in Fig. 6.12*b*) will no longer be

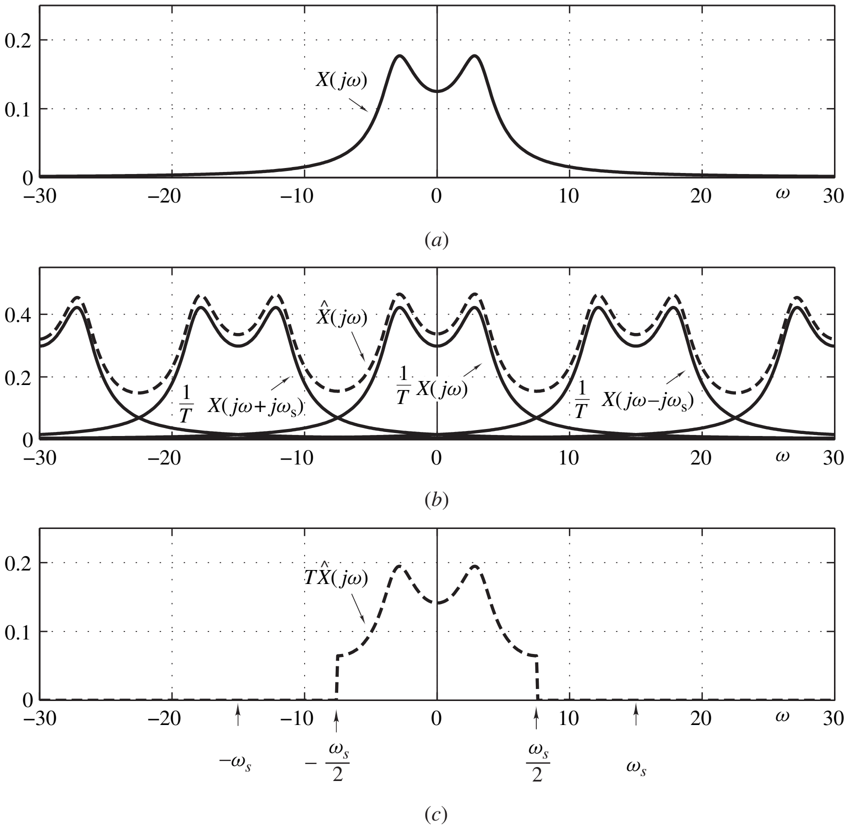


Figure 6.12 Aliasing of an impulse-modulated signal: (*a*) $X(j\omega)$, (*b*) shifted copies of $X(j\omega)/T$ and $\hat{X}(j\omega)$, (*c*) lowpass-filtered version of $\hat{X}(j\omega)$.

equal to $X(j\omega)$ over the baseband, and the use of an ideal lowpass filter will at best yield a distorted version of $x(t)$, as illustrated in Fig. 6.12*c*. The cause of the problem is *aliasing*, which was explained in some detail in Sec. 5.5.4.

6.8 GRAPHICAL REPRESENTATION OF INTERRELATIONS

Various important interrelations have been established in the preceding sections among continuous-time, impulse-modulated, and discrete-time signals. These are illustrated pictorially in Fig. 6.13. The two-directional paths between $\hat{x}(t)$ and $x(nT)$ and between $\hat{X}(j\omega)$ and $X_D(z)$ render the Fourier transform applicable to DSP. The two-directional paths between $x(t)$ and $x(nT)$ and between $X(j\omega)$ and $X_D(z)$ will allow us to use digital filters for the processing of continuous-time signals. And the path between $X(s)$ and $X_D(z)$ will allow us to design digital filters by using analog-filter methodologies.

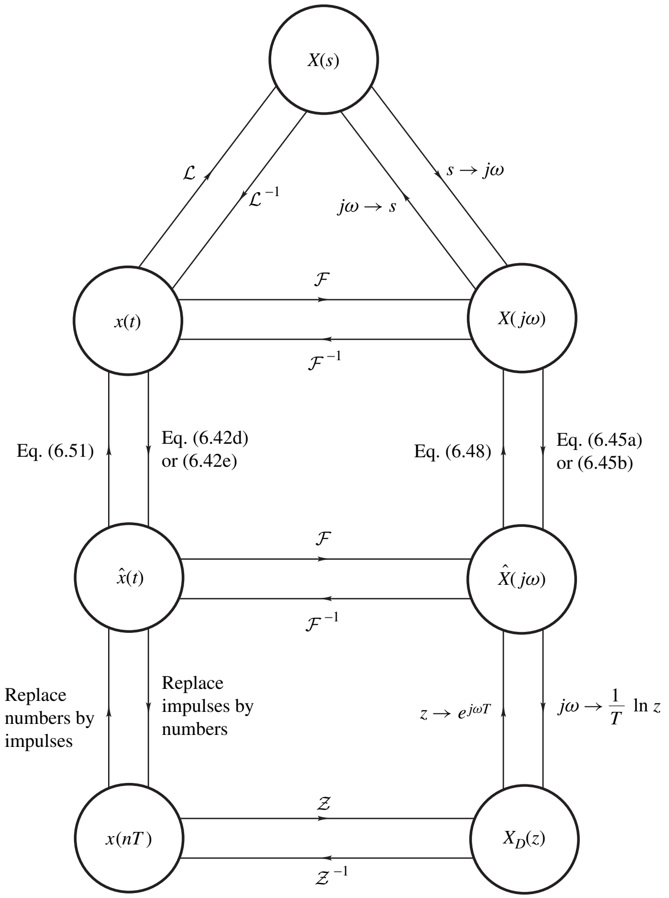


Figure 6.13 Interrelations between continuous-time, impulse-modulated, and discrete-time signals.

6.9 PROCESSING OF CONTINUOUS-TIME SIGNALS USING DIGITAL FILTERS

Consider the filtering scheme of Fig. 6.14*a* where S_1 and S_2 are impulse modulators and F_A and F_{LP} are analog filters characterized by transfer functions $H_A(s)$ and $H_{\text{LP}}(s)$, respectively, and assume that F_{LP} is an ideal lowpass filter with a frequency response

$$H_{\text{LP}}(j\omega) = \begin{cases} T^2 & \text{for } |\omega| < \omega_s/2 \\ 0 & \text{otherwise} \end{cases} \tag{6.52}$$

Filter F_A in cascade with impulse modulator S_2 constitute a so-called *impulse-modulated filter* $\hat{F}_A$.

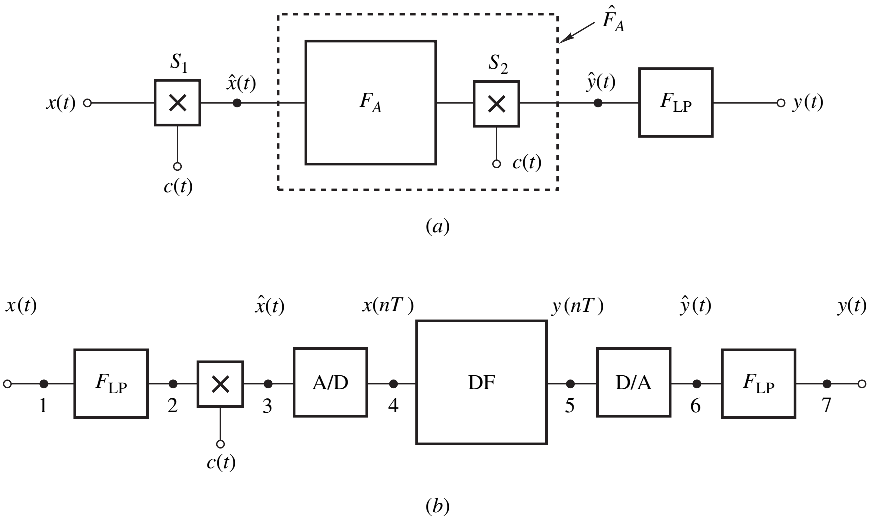


Figure 6.14 The processing of continuous-time signals: (*a*) Using an impulse-modulated filter, (*b*) using a digital filter.

By analogy with Eqs. (6.42e) and (6.46b), the impulse response and transfer function of filter $\hat{F}_A$ can be expressed as

$$\hat{h}_A(t) = \sum_{n=0}^{\infty} h_A(nT)\delta(t - nT) \tag{6.53a}$$

and

$$\hat{H}_A(s) = H_D(z) = \frac{h_A(0+)}{2} + \frac{1}{T}\sum_{n=-\infty}^{\infty} H_A(s + jn\omega_s) \tag{6.53b}$$

respectively, where

$$h_A(t) = \mathcal{L}^{-1}H_A(s) \qquad h_A(0+) = \lim_{s\to\infty}[sH_A(s)]$$

$$H_D(z) = \mathcal{Z}h_A(nT) \qquad z = e^{sT}$$

The transfer function of the cascade arrangement of the impulse-modulated filter and the lowpass filter is simply the product of their individual transfer functions, that is, $\hat{H}_A(s)H_{LP}(s)$, and hence the Laplace transform of $y(t)$ can be obtained as

$$Y(s) = \hat{H}_A(s)H_{LP}(s)\hat{X}(s)$$

Therefore, the Fourier transform of $y(t)$ in Fig. 6.14*a* is

$$Y(j\omega) = \hat{H}_A(j\omega)H_{LP}(j\omega)\hat{X}(j\omega) \tag{6.54}$$

and if

$$x(0+) = h_A(0+) = 0 \tag{6.55a}$$

and

$$X(j\omega) = H_A(j\omega) = 0 \qquad \text{for } |\omega| \geq \omega_s/2 \tag{6.55b}$$

then $\hat{X}(j\omega)$ and $\hat{H}_A(j\omega)$ are *periodic continuations* of $X(j\omega)/T$ and $H_A(j\omega)/T$, respectively, and thus Eqs. (6.45a), (6.53b), (6.55a), and (6.55b) give

$$\hat{X}(j\omega) = \frac{1}{T}X(j\omega) \qquad \text{and} \qquad \hat{H}_A(j\omega) = \frac{1}{T}H_A(j\omega) \qquad \text{for } |\omega| < \frac{\omega_s}{2} \tag{6.56}$$

Hence Eqs. (6.54), (6.52), and (6.56) give

$$Y(j\omega) = \frac{1}{T}H_A(j\omega) \cdot T^2 \cdot \frac{1}{T}X(j\omega) = H_A(j\omega)X(j\omega) \qquad \text{for } |\omega| < \omega_s/2 \tag{6.57a}$$

and

$$Y(j\omega) = 0 \qquad \text{for } |\omega| \geq \omega_s/2 \tag{6.57b}$$

Now from the assumptions made in Eqs. (6.55a) and (6.55b), $H_A(j\omega)X(j\omega) = 0$ for $|\omega| \geq \omega_s/2$ and thus Eq. (6.57b) is redundant, i.e., Eq. (6.57a) holds for all frequencies. On replacing $j\omega$ by s in Eq. (6.57a), the s-domain representation of the impulse-modulated filter can be obtained as

$$Y(s) = H_A(s)X(s)$$

After much science and some extravagant use of components, we have managed to construct an impulse-modulated filter that operates exactly like analog filter F_A in Fig. 6.14a. At first sight, this appears to be an unnecessarily complicated and highly redundant design of an analog filter that would certainly not excite the market place if offered as a commercial product! However, an important feat has been achieved in the process as will now be demonstrated. From Eq. (6.53b), we note that, just like a digital filter, an impulse-modulated filter can be characterized by a *discrete-time transfer function* and by replacing the impulse-modulated filter in Fig. 6.14*a* by an equivalent digital filter, the filtering of continuous-time signals by means of digital filters can be achieved. On the other hand, given an analog filter whose transfer function, $H_A(s)$, satisfies the conditions

$$h_A(0+) = 0 \qquad \text{and} \qquad H_A(j\omega) \approx 0 \qquad \text{for } |\omega| \geq \omega_s/2$$

a corresponding impulse-modulated filter can be obtained and from that an equivalent digital filter can be deduced that has approximately the same frequency response as the analog filter. Therefore, digital filters can be designed by using analog-filter methodologies. This possibility will be explored further in Chap. 11.

A digital-filter implementation of Fig. 6.14*a* can be obtained by replacing the impulse-modulated filter by a digital filter together with suitable interfacing devices, as shown in Fig. 6.14*b*.

The *analog-to-digital* and *digital-to-analog* converters are required to convert impulses into numbers and numbers into impulses. The input lowpass filter is used to bandlimit $x(t)$ (if it is not already bandlimited) to prevent aliasing errors. The detailed operation of such a filtering scheme is illustrated by the following example.

Example 6.5 The configuration of Fig. 6.14*b* is used to filter a periodic signal given by

$$\tilde{x}(t) = \sum_{n=-\infty}^{\infty} x(t + n\tau_0) \qquad where \qquad x(t) = \begin{cases} \sin \omega_0 t & \text{for } 0 \le t \le \tau_0/2 \\ 0 & \text{for } -\tau_0/2 \le t \le 0 \end{cases}$$

where $\omega_0 = 2\pi/\tau_0$. The lowpass filters are characterized by

$$H_{\text{LP}}(j\omega) = \begin{cases} 1 & \text{for } 0 \le |\omega| < 6\omega_0 \\ 0 & \text{otherwise} \end{cases}$$

and the digital filter has a baseband response

$$H_D(e^{j\omega T}) = \begin{cases} T & \text{for } 0.95\omega_0 < |\omega| < 1.05\omega_0 \\ 0 & \text{otherwise} \end{cases}$$

Assuming that $\omega_s = 12\omega_0$, find the time- and frequency-domain representations of the signals at nodes 1, 2, . . . , 7.

■ **Solution**

Node 1 From Example 6.1, part (*b*), the Fourier series of the signal can be expressed as

$$x_1(t) = \tfrac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos n\omega_0 t + \sum_{n=1}^{\infty} b_n \sin n\omega_0 t \quad ■ \tag{6.58}$$

where

$$a_0 = \frac{2}{\pi} \qquad a_1 = 0 \qquad a_2 = -\frac{2}{3\pi} \qquad a_3 = 0 \qquad a_4 = -\frac{2}{15\pi}$$

$$a_5 = 0 \qquad a_6 = -\frac{2}{35\pi} \qquad a_7 = 0 \cdots$$

$$b_1 = \frac{1}{2} \qquad b_2 = 0 \qquad b_3 = 0 \qquad b_4 = 0 \qquad b_5 = 0 \qquad b_6 = 0, \ \cdots$$

and from Table 6.2, the Fourier transform of $x_1(t)$ can be obtained as

$$X_1(j\omega) = \mathcal{F}x_1(t) = a_0\pi\delta(\omega) + \sum_{k=1}^{\infty} a_n\pi[\delta(\omega + n\omega_0) + \delta(\omega - n\omega_0)]$$

$$+ \sum_{k=1}^{\infty} jb_n\pi[\delta(\omega + n\omega_0) - \delta(\omega - n\omega_0)] \quad ■ \tag{6.59}$$

Node 2 The bandlimiting filter will remove all frequencies equal to or greater than $6\omega_0$ and hence Eq. (6.58) gives

$$\begin{aligned} x_2(t) &= \tfrac{1}{2}a_0 + \sum_{n=1}^{4} a_n \cos n\omega_0 t + \sum_{n=1}^{4} b_n \sin n\omega_0 t \\ &= \frac{1}{\pi} + \frac{1}{2}\sin\omega_0 t - \frac{2}{3\pi}\cos 2\omega_0 t - \frac{2}{15\pi}\cos 4\omega_0 t \quad \blacksquare \end{aligned}$$

and

$$\begin{aligned} X_2(j\omega) &= \mathcal{F}x_2(t) \\ &= 2\delta(\omega) + j\frac{\pi}{2}[\delta(\omega+\omega_0) - \delta(\omega-\omega_0)] - \frac{2}{3}[\delta(\omega+2\omega_0) \\ &\quad +\delta(\omega-2\omega_0)] - \frac{2}{15}[\delta(\omega+4\omega_0) + \delta(\omega-4\omega_0)] \quad \blacksquare \end{aligned}$$

Nodes 3 and 4 The output of the impulse modulator is obtained from Eq. (6.42d) as

$$\begin{aligned} \hat{x}_3(t) &= \sum_{n=-\infty}^{\infty} x(nT)\delta(t-nT) \\ &= \sum_{n=-\infty}^{\infty} \left(\frac{1}{\pi} + \frac{1}{2}\sin\omega_0 nT - \frac{2}{3\pi}\cos 2\omega_0 nT - \frac{2}{15\pi}\cos 4\omega_0 nT\right)\delta(t-nT) \end{aligned}$$

Thus

$$x_4(nT) = \frac{1}{\pi} + \frac{1}{2}\sin\omega_0 nT - \frac{2}{3\pi}\cos 2\omega_0 nT - \frac{2}{15\pi}\cos 4\omega_0 nT$$

and from Eq. (6.45a), we have

$$\begin{aligned} \hat{X}_3(j\omega) &= X_4(e^{j\omega T}) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X_2(j\omega + jn\omega_0) \\ &= \frac{1}{T}\sum_{n=-\infty}^{\infty} \left\{ 2\delta(\omega+n\omega_s) + j\frac{\pi}{2}[\delta(\omega+n\omega_s+\omega_0) - \delta(\omega+n\omega_s-\omega_0)] \right. \\ &\quad -\frac{2}{3}[\delta(\omega+n\omega_s+2\omega_0) + \delta(\omega+n\omega_s-2\omega_0)] \\ &\quad \left. -\frac{2}{15}[\delta(\omega+n\omega_s+4\omega_0) + \delta(\omega+n\omega_s-4\omega_0)] \right\} \quad \blacksquare \end{aligned}$$

Nodes 5 and 6 The bandpass digital filter will reject all components except those with frequencies $\pm\omega_0 \pm n\omega_s$ and it will also provide a gain of T, and so

$$\begin{aligned} X_5(e^{j\omega T}) &= \hat{X}_6(j\omega) \\ &= \sum_{n=-\infty}^{\infty} j\frac{\pi}{2}[\delta(\omega + n\omega_s + \omega_0) - \delta(\omega + n\omega_s - \omega_0)] \quad \blacksquare \end{aligned}$$

Thus, from Table 6.2, we get

$$x_5(nT) = \tfrac{1}{2}\sin\omega_0 nT \quad \blacksquare$$

and

$$\hat{x}_6(t) = \frac{1}{2}\sum_{n=-\infty}^{\infty}(\sin\omega_0 nT)\delta(t - nT) \quad \blacksquare$$

Node 7 Finally, the lowpass filter at the output will reject all components with frequencies outside the baseband, and as a result

$$X_7(j\omega) = \left(\frac{j\pi}{2}\right)[\delta(\omega + \omega_0) - \delta(\omega - \omega_0)] \quad \blacksquare$$

and

$$x_7(t) = \tfrac{1}{2}\sin\omega_0 t \quad \blacksquare$$

The various signal waveforms and amplitude spectrums are illustrated in Fig. 6.15*a* and *b*, respectively.

6.10 PRACTICAL A/D AND D/A CONVERTERS

A practical implementation of the *analog-to-digital* interface is shown in Fig. 6.16*a*. The function of the *sample-and-hold* device is to generate a signal of the form

$$\tilde{x}(t) = \sum_{n=-\infty}^{\infty} x(nT)p_T(t - nT)$$

such as that illustrated in Fig. 6.16*b*. The function of the *encoder*, on the other hand, is to convert each signal level $x(nT)$ into a corresponding binary number. Since the number of bits in the binary representation must be finite, the response of the encoder denoted by $x_q(nT)$ can assume only a finite number of discrete levels; that is, $x_q(nT)$ will be a *quantized signal*. Assuming that the encoder is designed such that each value of $x(nT)$ is rounded to the nearest discrete level, the response of the encoder will be of the form depicted in Fig. 6.16*c*. We can write

$$x_q(nT) = x(nT) - e(nT)$$

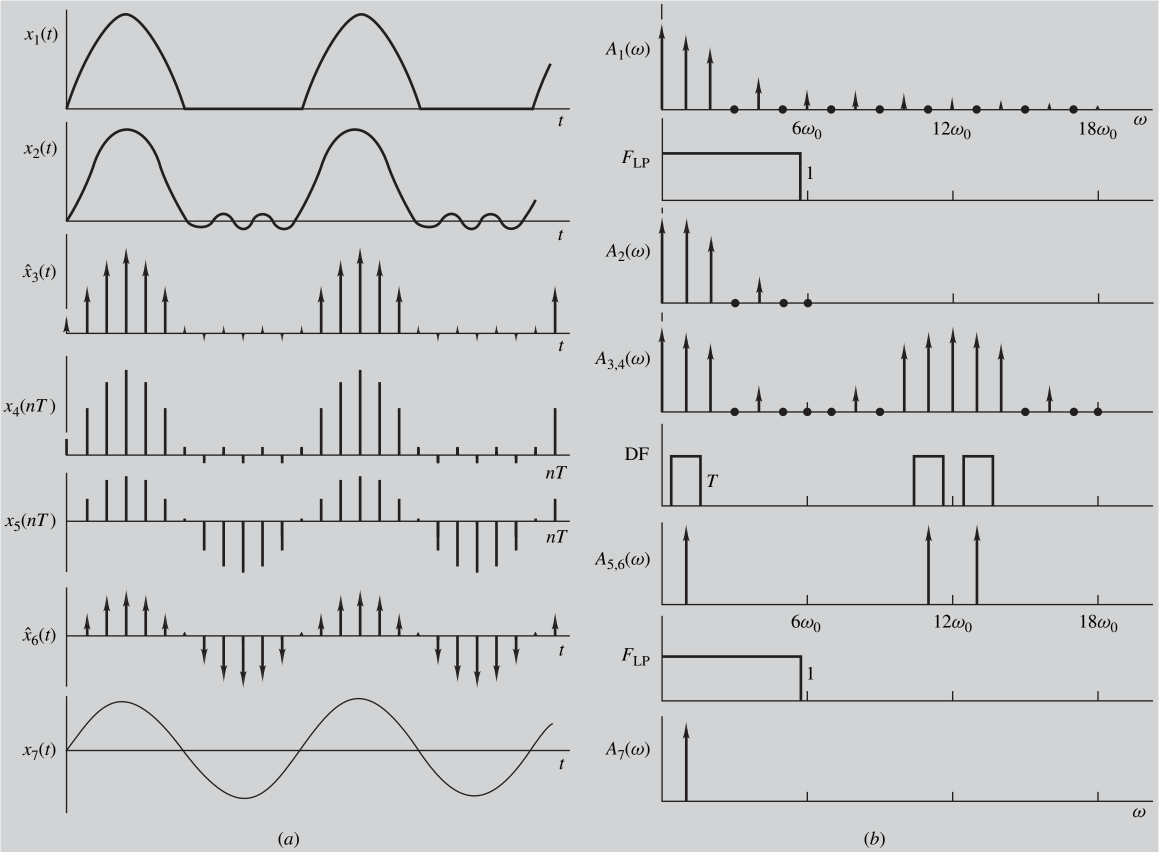


Figure 6.15 Example 6.5: (*a*) Time-domain representations of signals at nodes 1, 2, . . . , 7, (*b*) amplitude spectrums of signals at nodes 1, 2, . . . , 7.

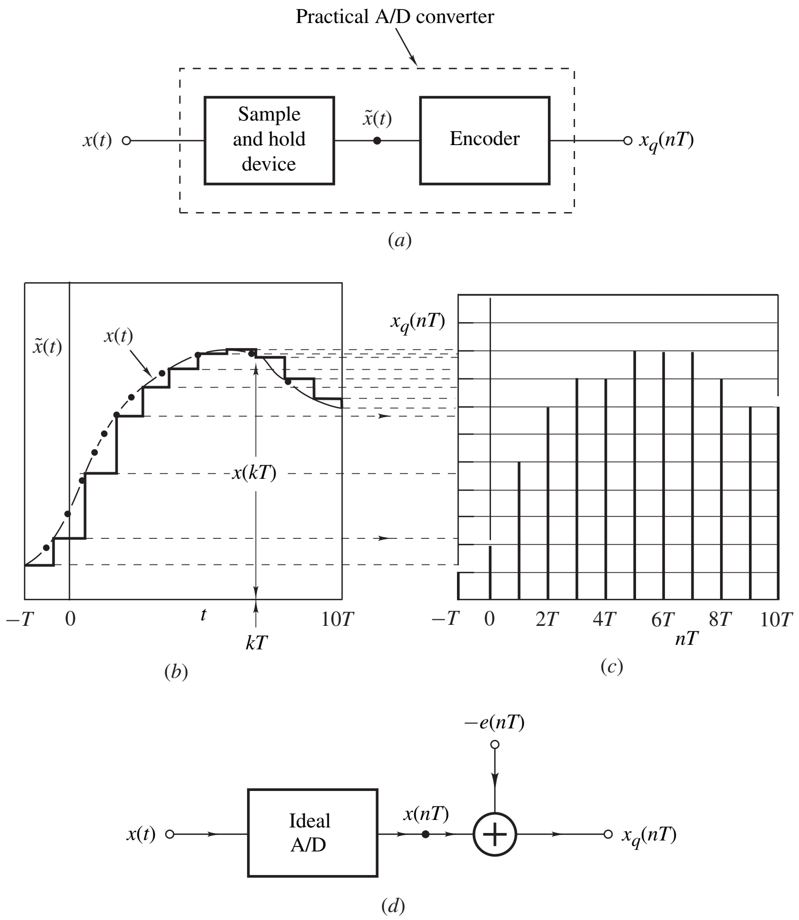


Figure 6.16 Analog-to-digital interface: (*a*) Practical A/D converter, (*b*) response of a practical A/D converter, (*c*) output of encoder, (*d*) model for a practical A/D converter.

where $e(nT)$ is the *quantization error*. Hence a practical A/D converter can be represented by the model of Fig. 6.16*d*, where $-e(nT)$ can be regarded as a noise source. The effect of this noise source on the filter response will be considered in Sec. 14.5.

The purpose of the *D/A converter* in Fig. 6.14*b* is to generate an impulse-modulated signal $\hat{y}(t)$ like the one shown in Fig. 6.17*a* where the impulse functions are very thin and very tall pulses as defined in Sec. 6.2.1. However, a practical D/A converter will produce a waveform of the form illustrated in Fig. 6.17*b* where the pulses may not be very thin or very tall. Such a waveform can be represented by the equation

$$\tilde{y}(t) = \sum_{n=-\infty}^{\infty} y(nT)p_\tau(t - nT)$$

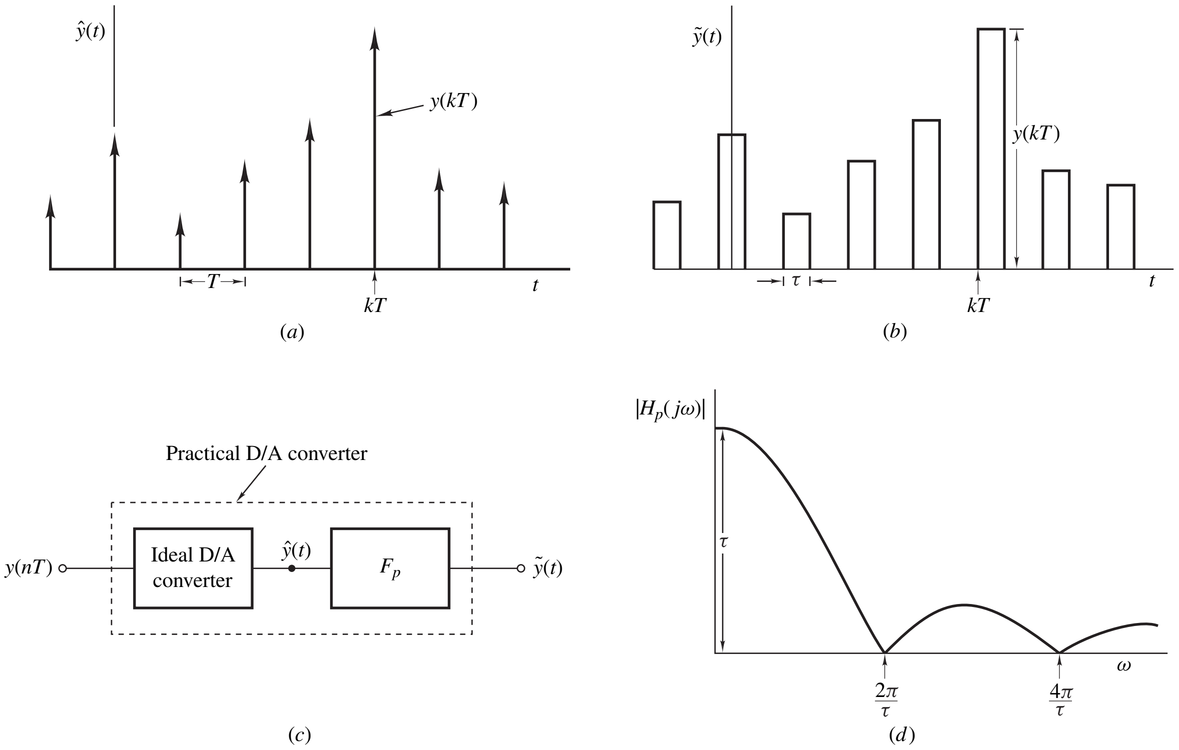


Figure 6.17 Digital-to-analog interface: (*a*) Response of an ideal D/A converter, (*b*) response of a practical D/A converter, (*c*) model for a practical D/A converter, (*d*) amplitude response of fictitious filter F_p.

From Table 6.2

$$\mathcal{F}p_\tau(t) = \frac{2\sin(\omega\tau/2)}{\omega}$$

and by using the time-shifting theorem (Theorem 2.9), we obtain

$$\mathcal{F}p_\tau(t - nT) = \frac{2\sin(\omega\tau/2)}{\omega}e^{-j\omega nT}$$

Hence the Fourier transform of $\tilde{y}(t)$ can be obtained as

$$\begin{aligned}\tilde{Y}(j\omega) &= \sum_{n=-\infty}^{\infty} y(nT)\mathcal{F}p_\tau(t - nT)\\ &= \frac{2\sin(\omega\tau/2)}{\omega}\sum_{n=-\infty}^{\infty} y(nT)e^{-j\omega nT}\end{aligned}$$

The above equation can be expressed as

$$\tilde{Y}(j\omega) = H_p(j\omega)\hat{Y}(j\omega)$$

where $$H_p(j\omega) = \frac{\tau\,\sin(\omega\tau/2)}{\omega\tau/2} \qquad \text{and} \qquad \hat{Y}(j\omega) = \sum_{n=-\infty}^{\infty} y(nT)e^{-j\omega nT}$$

By analogy with Eq. (6.43a), we can write

$$\hat{Y}(j\omega) = \sum_{n=-\infty}^{\infty} y(nT)e^{-j\omega nT} = \mathcal{F}\sum_{n=-\infty}^{\infty} y(nT)\delta(t - nT) = \mathcal{F}\hat{y}(t)$$

where $\hat{y}(t)$ is an impulse-modulated signal of the type that would be produced by an ideal D/A converted. Therefore, a practical D/A converter can be modeled in terms of an ideal D/A converter followed by a fictitious filter F_p, as depicted in Fig. 6.17*c*, with a frequency response $H_p(j\omega)$. The amplitude response of this filter is given by

$$|H_p(j\omega)| = \tau\left|\frac{\sin(\omega\tau/2)}{\omega\tau/2}\right| \tag{6.60}$$

and is sketched in Fig. 6.17*d*. Clearly, a practical D/A converter will introduce distortion in the overall amplitude response often referred to as *sinc distortion* which is actually a form of amplitude distortion (see Sec. 5.7).

The effect of sinc distortion can be easily illustrated if we assume that the digital filter in Fig. 6.14*b* is a bandpass filter with an ideal amplitude response such as that in Fig. 6.18*a*. Due to sinc distortion, the amplitude response of the D/A converter will assume the form in Fig. 6.18*b* and thus the overall amplitude response of the bandpass filter in cascade with the D/A converter will

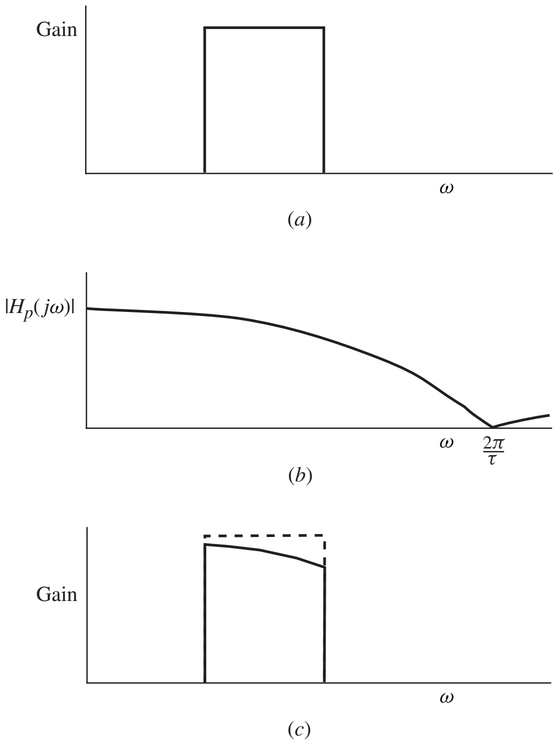


Figure 6.18 Effect of sinc distortion: (*a*) Amplitude response of an ideal bandpass filter, (*b*) response of a practical D/A converter, (*c*) overall response of bandpass filter in cascade with D/A converter.

be the product of the amplitude responses in Fig. 6.18*a* and *b*, as depicted in Fig. 6.18*c*. Evidently, the D/A converter will introduce a reduction in the overall gain, which would tend to increase with frequency.

Sinc distortion can be reduced in a number of ways. One possibility would be to make the pulse width τ as small as possible which would move the notch at $\omega = 2\pi/\tau$ to a higher frequency. Another possibility would be to design the output lowpass filter in Fig. 6.14*b* such that

$$|H_p(j\omega)H_{\text{LP}}(j\omega)| \approx \begin{cases} 1 & \text{for } |\omega| \leq \omega_s/2 \\ 0 & \text{otherwise} \end{cases}$$

On the other hand, in an application like that in Example 6.5 it is sometimes possible to design the digital filter such that the digital filter in cascade with the D/A converter satisfies the desired filtering specifications. For example, the amplitude response of the bandpass filter can be designed to increase as the frequency is increased so as to compensate for the reduction in the amplitude response of the D/A converter, as illustrated in Fig. 6.19. Such a design would necessitate the use of optimization and an actual design of this type can be found in Sec, 16.7 (see Example 16.3).

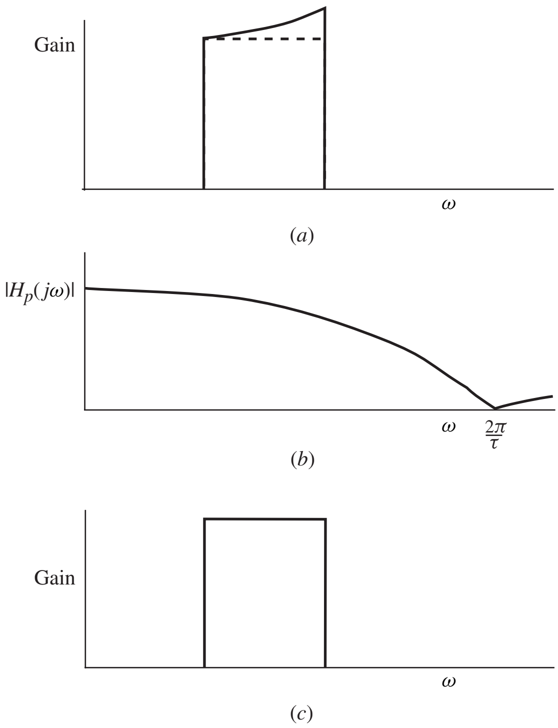


Figure 6.19 Compensation for sinc distortion: (*a*) Amplitude response an ideal bandpass filter, (*b*) response of a practical D/A converter, (*c*) overall response of bandpass filter and D/A converter.

Example 6.6 A bandpass digital filter with an idealized amplitude response

$$M_{BP}(\omega)\begin{cases}\le 10^{-3} & \text{for } 0 \le \omega \le 1200 \text{ rad/s}\\ = 1 & \text{for } 1500 \le \omega \le 3000 \text{ rad/s}\\ \le 10^{-3} & \text{for } 3300 \le \omega \le 5000 \text{ rad/s}\end{cases}$$

is used in the DSP scheme shown in Fig. 6.14*b*. The output lowpass filter is ideal with a constant passband gain of $1/\tau$ and the D/A converter will produce an output waveform of the type shown in Fig. 6.17*b* with a pulse width $\tau = 0.3$ ms. (*a*) Find the maximum passband loss introduced by the D/A converter in dB if the sampling frequency is $\omega_s = 10^4$ rad/s. (*b*) Find the minimum sampling frequency that would keep the maximum passband loss to a value equal to or less than 0.1 dB.

■ **Solution**

(*a*) The overall amplitude response of the DSP system is given by

$$M(\omega) = M_{BP}(\omega) \cdot |H_p(j\omega)| \cdot |H_{LP}(j\omega)|$$

and from Eq. (6.60), we have

$$M(\omega) = M_{BP}(\omega) \cdot \tau \left| \frac{\sin(\omega\tau/2)}{\omega\tau/2} \right| \cdot \frac{1}{\tau}$$

$$= M_{BP}(\omega) \cdot \left| \frac{\sin(\omega\tau/2)}{\omega\tau/2} \right|$$

Thus the D/A converter will introduce a loss which increases with frequency. The maximum passband loss will occur at the upper edge of the passband, i.e., at $\omega = 3000$ rad/s, and hence

$$\text{Gain} = M(3000) = \left| \frac{\sin(3000 \times 0.3 \times 10^{-3}/2)}{3000 \times 0.3 \times 10^{-3}/2} \right| = 0.9666$$

or

$$\text{Loss} = 20 \log \frac{1}{0.9666} = 0.2951 \text{ dB} \quad ■$$

(*b*) The loss will be equal to or less than 0.1 dB at $\omega = 3000$ if

$$\text{Loss} = 20 \log \frac{\omega\tau/2}{\sin(\omega\tau/2)} \le 0.1$$

$$= 20 \log \frac{3000\tau/2}{\sin(3000\tau/2)} \le 0.1$$

Hence we require

$$\frac{3000\tau/2}{\sin(3000\tau/2)} \le 10^{0.1/20} = 1.011579$$

A short MATLAB program will reveal that

$$3000\tau/2 \le 1.011579 \sin(3000\tau/2)$$

if $\tau \le 1.7525 \times 10^{-4}$ and, therefore, $\omega_s \ge 2\pi/(1.7525 \times 10^{-4}) = 3.5853 \times 10^4$ rad/s. ■

REFERENCES

[1] M. J. Lighthill, *Introduction to Fourier Analysis and Generalised Functions*, Cambridge: Cambridge University Press, 1958.

[2] W. Kaplan, *Advanced Calculus*, 3rd ed., Reading, MA: Addison-Wesley, 1962.

[3] R. J. Schwarz and B. Friedland, *Linear Systems*, New York: McGraw-Hill, 1965.

[4] A. Papoulis, *The Fourier Integral and Its Applications*, New York: McGraw-Hill, 1962.

[5] A. Antoniou, *Digital Filters: Analysis, Design, and Applications*, 2nd ed., New York: McGraw-Hill, 1993.

[6] H. Nyquist, “Certain topics in telegraph transmission theory,” *Trans. A.I.E.E*, pp. 617–644, Feb. 1928. (See also *Proc. IEEE*, vol. 90, pp. 280–305, Feb. 2002.)

[7] C. E. Shannon, “Communication in the presence of noise,” *Proc. IRE*, vol. 37, pp. 10–21, Jan. 1949. (See also *Proc. IEEE*, vol. 86, pp. 447–457, Feb. 1998.)

PROBLEMS

6.1. (*a*) Assuming that

$$\delta(\omega) = \lim_{\varpi \to \epsilon} \bar{p}_{\varpi}(\omega)$$

where

$$p_{\varpi}(\omega) = \begin{cases} 1 & \text{for } |\omega| \le \varpi/2 \\ 0 & \text{otherwise} \end{cases}$$

and ϵ is a very small but finite constant, show that $\delta(\omega)$ is an even function of ω, that is,

$$\delta(-\omega) = \delta(\omega)$$

(*b*) Using the above definition, prove part (*a*) of Theorem 6.1B.

(*c*) Prove part (*b*) of Theorem 6.1B.

(*d*) Prove part (*c*) of Theorem 6.1B.

6.2. (*a*) A periodic signal $\tilde{x}(t)$ can be represented by Eq. (6.30) with

$$x(t) \equiv p_\tau(t) = \begin{cases} 1 & \text{for } -\tau/2 \le t \le \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

Obtain the Fourier transform of $\tilde{x}(t)$.
(*b*) Repeat part (*a*) if

$$x(t) = p_{\tau/2}\left(t + \tfrac{1}{4}\tau\right) - p_{\tau/2}\left(t - \tfrac{1}{4}\tau\right)$$

6.3. (*a*) A periodic signal $\tilde{x}(t)$ can be represented by Eq. (6.30) with

$$x(t) = \begin{cases} 1 & \text{for } -\tau/2 \le t < -\tau/4 \\ 2 & \text{for } -\tau/4 \le t < \tau/4 \\ 1 & \text{for } \tau/4 \le t < \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

Obtain the Fourier transform of $\tilde{x}(t)$.

(*b*) Repeat part (*a*) if

$$x(t) = \begin{cases} 1 & \text{for } -3\tau/8 \le t < -\tau/4 \\ -2 & \text{for } -\tau/4 \le t < \tau/4 \\ 1 & \text{for } \tau/4 \le t \le 3\tau/8 \\ 0 & \text{otherwise} \end{cases}$$

6.4. (*a*) A periodic signal $\tilde{x}(t)$ can be represented by Eq. (6.30) with

$$x(t) = \begin{cases} 1 & \text{for } -\tau/2 \le t \le -\tau_1/2 \\ 1 & \text{for } \tau_1/2 \le t \le \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

where $\tau > \tau_1$. Obtain the Fourier transform of $\tilde{x}(t)$.

(*b*) Repeat part (*a*) if

$$x(t) = \begin{cases} 1 & \text{for } -\tau/2 \le t \le -\tau_1/2 \\ -1 & \text{for } \tau_1/2 \le t \le \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

where $\tau > \tau_1$.

6.5. (*a*) A periodic signal $\tilde{x}(t)$ can be represented by Eq. (6.30) with

$$x(t) = \begin{cases} 1 & \text{for } -\tau/2 \le t \le -\tau_2/2 \\ 1 & \text{for } -\tau_1/2 \le t \le \tau_1/2 \\ 1 & \text{for } \tau_2/2 \le t \le \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

where $\tau > \tau_2 > \tau_1$. Obtain the Fourier transform of $\tilde{x}(t)$.

(*b*) Repeat part (*a*) if

$$x(t) = \begin{cases} 1 & \text{for } -\tau/2 \le t \le -\tau_2/2 \\ -1 & \text{for } -\tau_1/2 < t < \tau_1/2 \\ 1 & \text{for } \tau_2/2 \le t \le \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

where $\tau > \tau_2 > \tau_1$.

6.6. (*a*) A periodic signal $\tilde{x}(t)$ can be represented by Eq. (6.30) with

$$x(t) = \begin{cases} \sin \omega_0 t & \text{for } 0 \le t \le \tau_0/4 \\ 0 & \text{otherwise} \end{cases}$$

where $\omega_0 = 2\pi/\tau_0$. Obtain the Fourier transform of $\tilde{x}(t)$.

(*b*) Repeat part (*a*) if

$$x(t) = \begin{cases} \cos \omega_0 t & \text{for } 0 \le t \le \tau_0/4 \\ 0 & \text{otherwise} \end{cases}$$

where $\omega_0 = 2\pi/\tau_0$.

6.7. (*a*) A periodic signal $\tilde{x}(t)$ can be represented by Eq. (6.30) with

$$x(t) = \begin{cases} \sinh \alpha t & \text{for } -\tau/2 \le t \le \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

Obtain the Fourier transform of $\tilde{x}(t)$.

(*b*) Repeat part (*a*) if

$$x(t) = \begin{cases} \cosh \alpha t & \text{for } -\tau/2 \le t \le \tau/2 \\ 0 & \text{otherwise} \end{cases}$$

6.8. (*a*) Find the Fourier transform of the periodic signal shown in Fig. P6.8*a* where $\omega_0 = 2\pi/\tau_0$. Sketch the amplitude spectrum of the signal.

(*b*) Repeat part(*a*) for the signal shown in Fig. P6.8*b*.

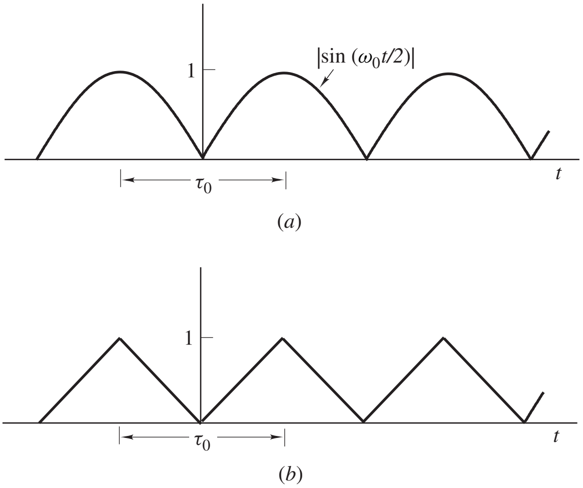


Figure P6.8*a* **and** *b*

6.9. (*a*) Find the Fourier transform of the periodic signal shown in Fig. P6.9*a*. Sketch the amplitude spectrum.

(*b*) Repeat part (*a*) for the signal shown in Fig. P6.9*b*.

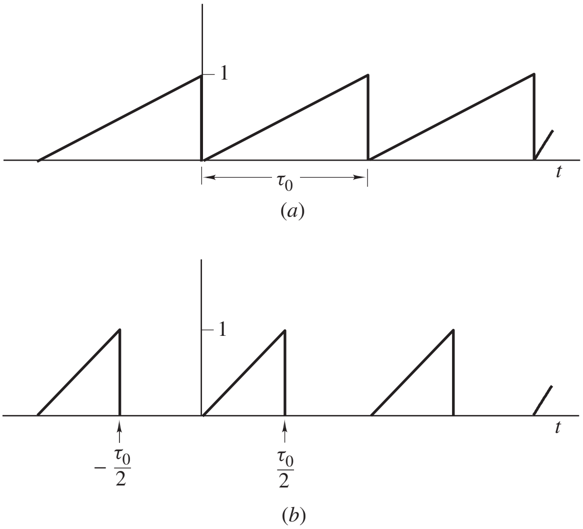
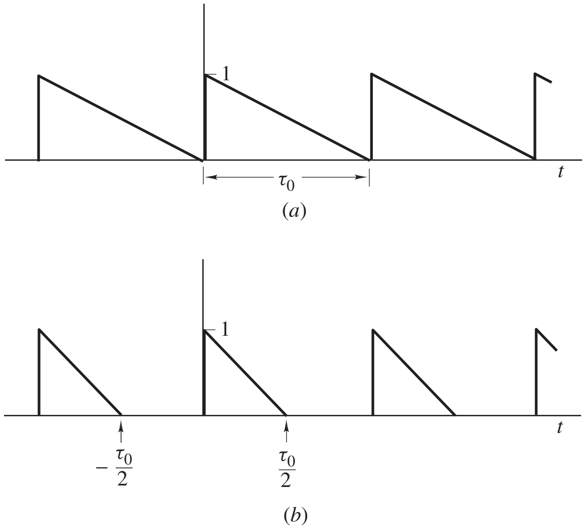


Figure P6.9*a* **and** *b*

6.10. (*a*) Find the Fourier transform of the periodic signal shown in Fig. P6.10*a*. Sketch the amplitude spectrum.
(*b*) Repeat part (*a*) for the signal shown in Fig. P6.10*b*.

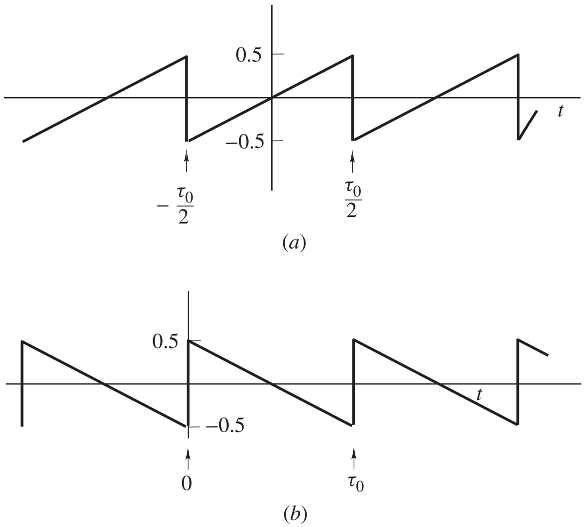


(*a*)

(*b*)

Figure P6.10*a* and *b*

6.11. (*a*) Find the Fourier transform of the periodic signals shown in Fig. P6.11*a*. Sketch the amplitude spectrum.
(*b*) Repeat part (*a*) for the signal shown in Fig. P6.11*b*.



(*a*)

(*b*)

Figure P6.11*a* and *b*

6.12. Find the Fourier transforms of the periodic signals

(*a*)

$$\tilde{x}(t) = \cos^2 \omega_0 t + \cos^4 \omega_0 t$$

(*b*)

$$\tilde{x}(t) = \tfrac{1}{2} + \sin \omega_0 t + \tfrac{1}{4} \sin^2 \omega_0 t + \cos^4 \omega_0 t$$

6.13. Find the Fourier transforms of the periodic signals

(*a*)

$$\tilde{x}(t) = (\sin 5\omega_0 t \cos \omega_0 t)^2$$

(*b*)

$$\tilde{x}(t) = (\cos 3\omega_0 t \cos 2\omega_0 t)^2$$

(*c*)

$$\tilde{x}(t) = (\cos \omega_0 t + j \sin \omega_0 t)^n$$

6.14. (*a*) Show that the Fourier series kernel in Eq. (6.28) is periodic with period T.

(*b*) Show that the signal in Eq. (6.30) is periodic with period T.

6.15. (*a*) Show that the periodic signal in Prob. 6.2, part (*a*), satisfies Theorem 6.3.

(*b*) Repeat part (*a*) for the periodic signal in Prob. 6.2, part (*b*).

6.16. (*a*) Show that the periodic signal in Prob. 6.3, part (*a*), satisfies Theorem 6.3.

(*b*) Repeat part (*a*) for the periodic signal in Prob. 6.4, part (*a*).

6.17. (*a*) Show that the periodic signal in Prob. 6.7, part (*a*), satisfies Theorem 6.3.

(*b*) Repeat part (*a*) for the periodic signal in Prob. 6.8, part (*b*).

6.18. (*a*) Signal $\hat{x}(t)$ is obtained by applying impulse modulation to the nonperiodic signal in Prob. 6.3, part (*a*). Obtain the Fourier transform of $\hat{x}(t)$ in closed form if $\tau = 5T$.

(*b*) Repeat part (*a*) if $\tau = 6T$.

6.19. (*a*) Signal $\hat{x}(t)$ is obtained by applying impulse modulation to the nonperiodic signal in Prob. 6.4, part (*a*). Obtain the Fourier transform of $\hat{x}(t)$ in closed form if $\tau = 6T$ and $\tau_1 = T$.

(*b*) Repeat part (*a*) if $\tau = 7T$ and $\tau_1 = 1.5T$.

6.20. (*a*) Signal $\hat{x}(t)$ is obtained by applying impulse modulation to the nonperiodic signal in Prob. 6.7, part (*a*). Obtain the Fourier transform of $\hat{x}(t)$ in closed form if $\omega_s = 2\pi/T = 18$ rad/s and $\tau = 1.0$ s.

(*b*) Repeat part (*a*) if $\omega_s = 2\pi/T = 20$ rad/s.

6.21. (*a*) Find the Fourier transform of

$$x(t) = p_\tau(t - 2T)$$

where $\tau = (N - 1)T/2$ and N is odd. The sampling frequency is $\omega_s = 2\pi/T$.

(*b*) Find the Fourier transform of the impulse-modulated signal $\hat{x}(t)$ in closed form.

(*c*) Find the Fourier transform of $\hat{x}(t)$ using Poisson's summation formula.

6.22. Repeat parts (*a*), (*b*), and (*c*) of Prob. 6.21 if

$$x(t) = \begin{cases} \alpha + (1 - \alpha)\cos \dfrac{\pi t}{\tau} & \text{for } |t| \le \tau \\ 0 & \text{otherwise} \end{cases}$$

assuming that $\omega_s = 2\pi/T$.

6.23. (*a*) Find the Fourier transform of

$$x(t) = u(t)2e^{-0.5t+0.1}$$

The sampling frequency is $\omega_s = 2\pi/T$.

(*b*) Find the Fourier transform of $\hat{x}(t)$ in closed form.

(*c*) Find the Fourier transform of $\hat{x}(t)$ using Poisson's summation formula.

6.24. The signal

$$x(t) = u(t)e^{-t}\cos 2t$$

is sampled at a rate of 2π rad/s.

(*a*) Find the Fourier transform of $x(t)$.

(*b*) Find the Fourier transform of $\hat{x}(t)$ in closed form.

(*c*) Show that

$$\hat{X}(j\omega) = X_D(e^{j\omega T}) = \frac{1}{2} + \sum_{k=-\infty}^{\infty} \frac{1 + j(\omega + 2\pi k)}{[1 + j(\omega + 2\pi k)]^2 + 4}$$

(*d*) By evaluating the left- and right-hand sides for a number of frequencies in the range $0 \le \omega \le \omega_s/2$, demonstrate that the relation in part (*c*) holds true. (**Hint:** The left-hand side is the *z* transform of $x(nT)$ evaluated on the unit circle $|z| = 1$. The right-hand side is, as can be seen, an infinite series but the magnitudes of its terms tend to diminish rapidly and eventually become negligible as $|k|$ increases.)

6.25. (*a*) Find the Fourier transform of

$$x(t) = u(t)e^{-0.01t}\sin 2\pi t$$

(*b*) Find the Fourier transform of $\hat{x}(t)$ in closed form assuming a sampling frequency $\omega_s = 10\pi$ rad/s.

(*c*) Repeat part (*b*) using Poisson's summation formula.

6.26. A nonperiodic pulse signal $x(t)$ assumes the form depicted in Fig. P6.26.

(*a*) Obtain a representation for $x(t)$ in the form of a summation.

(*b*) Find the Fourier transform of $x(t)$.

(*c*) Obtain the Fourier transform of impulse-modulated signal $\hat{x}(t)$ in the form of an infinite summation.

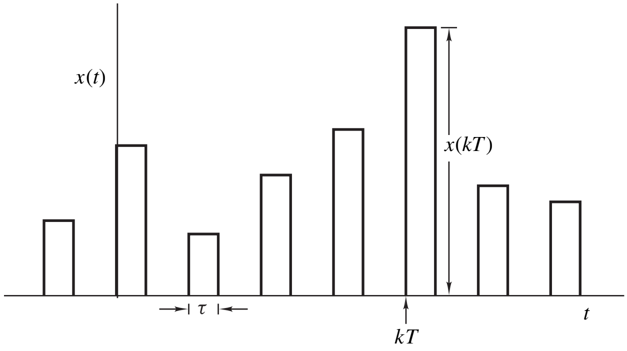


Figure P6.26

6.27. (*a*) Obtain the Fourier of

$$x(t) = \begin{cases} 1 - \dfrac{|t|}{\tau} & \text{for } |t| \le \tau \\ 0 & \text{otherwise} \end{cases}$$

where $\tau = (N-1)T/2$ and $T = 2\pi/\omega_s$.

(*b*) Using Poisson's summation formula, show that

$$\hat{X}(j\omega) \approx \frac{8}{\omega^2 (N-1)T^2} \sin^2 \frac{\omega(N-1)T}{4} \qquad \text{for } |\omega| < \frac{\omega_s}{2}$$

if $\omega_s \gg 16/\pi N$. (**Hint:** Note that $X(j\omega) \to 0$ if $\omega_s \gg 16/\pi N$.)

6.28. The filtering scheme of Example 6.5 is used to process the signal of Fig. P6.8*b*.

(*a*) Assuming that $\omega_0 = 2\pi/\tau_0$, find the time- and frequency-domain representations of the signals at nodes $1, 2, \ldots, 7$.

(*b*) Sketch the various waveforms and amplitude spectrums.

6.29. The DSP system of Fig. 6.14*b* employs a bandpass digital filter with an amplitude response like that depicted in Fig. P6.29. The output of the D/A converter is of the form shown in Fig. 6.17*b*, where $\tau = 3.0$ ms and the sampling frequency is 1000 rad/s.

(*a*) Assuming an ideal output lowpass filter with a constant passband gain of $1/\tau$, sketch the overall amplitude response of the configuration indicating relevant quantities.

(*b*) The gain at $\omega = 300$ rad/s is required to be equal to or greater than 0.99 times the gain at $\omega = 200$ rad/s. Find the maximum permissible value of τ. Hence, calculate the lowest sampling frequency.

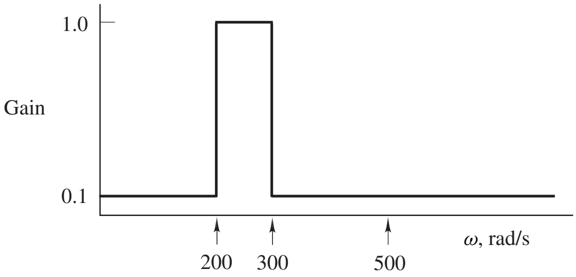


Figure P6.29

6.30. A digital filter is connected in cascade with a practical D/A converter as shown in Fig. 6.14*b*. The amplitude response of the filter is illustrated in Fig. P6.30*a* and the time-domain response of the D/A converter is a staircase waveform of the type shown in Fig. P6.30*b*. The sampling frequency is $\omega_s = 8$ rad/s.

(*a*) Calculate the overall gain of the cascade arrangement for $\omega = 0, 2-\epsilon, 2+\epsilon, 3-\epsilon, 3+\epsilon$, and 4 rad/s ($\epsilon$ is a small positive constant, say less than 0.0001).

(*b*) Sketch the amplitude response of the arrangement, indicating relevant quantities.

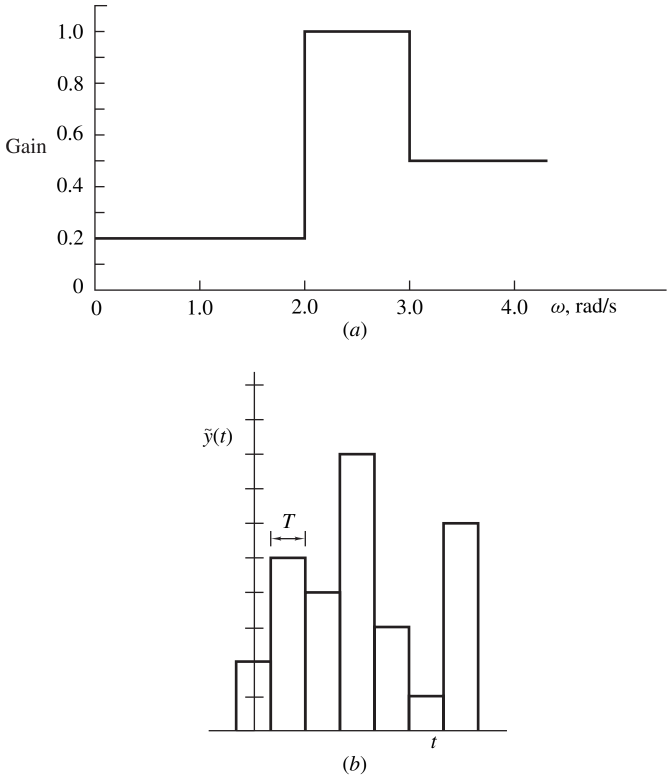


Figure P6.30

6.31. The DSP system of Fig. 6.14*b* uses a bandpass filter characterized by the transfer function

$$H(z) = H_0 \prod_{j=1}^{3} \frac{a_{0j} + a_{1j}z + z^2}{b_{0j} + b_{1j}z + z^2}$$

and has a passband ripple[6] of 1 dB. The sampling frequency is 20 rad/s. The output lowpass filter is ideal with a constant passband gain of $1/\tau$ and the D/A converter will produce an output waveform of the type shown in Fig. 6.17*b*.

(*a*) Find the amplitude response of the system, including the effect of the D/A converter, assuming a pulse duration $\tau = T/100$ in the waveform of Fig. 6.17*b* .

(*b*) Repeat part (*a*) if $\tau = T$.

(*c*) Find the lowest sampling frequency that would assure a maximum passband ripple that does not exceed 1.2 dB over the passband range 4.0 to 6.0 rad/s.

[6] *This is the difference between the maximum and minimum passband gain in dB.*

Table P6.31 Coefficients of $H(z)$ (Prob. 6.31)

j	a_{0j}	a_{1j}	b_{0j}	b_{1j}
1	−1.0	0.0	$7.190253E-1$	$7.461412E-8$
2	1.0	−1.691081	$8.670819E-1$	$5.623427E-1$
3	1.0	1.691082	$8.670819E-1$	$-5.623425E-1$
$H_0 = 1.640287E-2$				

This page intentionally left blank

CHAPTER 7

THE DISCRETE FOURIER TRANSFORM

7.1 INTRODUCTION

Given a finite-duration discrete-time signal, a corresponding periodic discrete-time signal can be generated which has a *discrete Fourier transform* (DFT) that happens to be a discrete-frequency spectrum. Thus given a signal that can be represented by a sequence of numbers a spectral characterization of the signal can be obtained, which can also be represented by a sequence of numbers. Consequently, the DFT is highly amenable to processing by computers and digital signal processors. The DFT turns out to be a sampled version of the frequency spectrum of the original finite-duration nonperiodic signal and, therefore, it is a very important tool for digital signal processing (DSP).

In a landmark paper published in 1965, Cooley and Tukey demonstrated that the direct evaluation of the DFT entails a considerable amount of redundancy and through an ingenious method that has come to be known as the *fast Fourier transform* (FFT), these authors and others [1–4] have shown that a huge amount of computation can be eliminated without degrading the precision of the DFT in any way. The FFT approach renders a highly amenable tool for computation also a very efficient one and for these reasons the DFT has found widespread applications over the years.

By analogy with the convolution summation described in Sec. 4.6, periodic convolutions can be constructed, which can also benefit greatly through the application of the FFT [5]. Through the periodic convolutions, some highly efficient implementations of finite-duration impulse response (FIR) filters can be achieved.

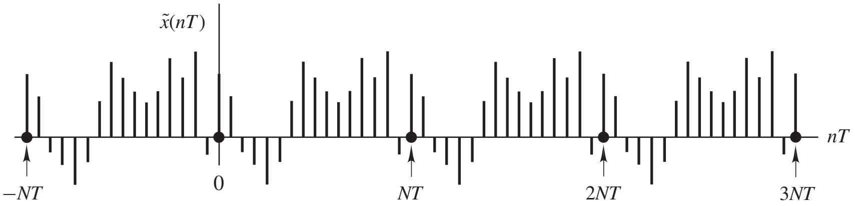


Figure 7.1 A periodic discrete-time signal.

This chapter deals with the definition of the DFT and inverse-DFT and their properties. The relations between the DFT and the z transform, the continuous-time Fourier transform (CFT), and Fourier series are then established. The chapter also deals with a technique known as the *window technique* which enables the processing of long- or infinite-duration signals through the use of the FFT method. Periodic convolutions and their efficient computation through the use of FFTs are then discussed.

7.2 DEFINITION

A discrete-time signal $\tilde{x}(nT)$ is periodic if it satisfies the condition

$$\tilde{x}(nT + rNT) = \tilde{x}(nT) \qquad \text{for } 1, 2, \dots, \infty$$

where N is an integer and NT is said to be the *period* of the signal. A periodic discrete-time signal is illustrated in Fig. 7.1.

The DFT of a periodic signal $\tilde{x}(nT)$ is defined as

$$\tilde{X}(jk\Omega) = \sum_{n=0}^{N-1} \tilde{x}(nT)W^{-kn} = \mathcal{D}\tilde{x}(nT) \tag{7.1}$$

where $$W = e^{j2\pi/N} \qquad \Omega = \frac{\omega_s}{N} \qquad \text{and} \qquad \omega_s = \frac{2\pi}{T}$$

In general, $\tilde{X}(jk\Omega)$ is complex and can be put in the form

$$\tilde{X}(jk\Omega) = A(k\Omega)e^{j\phi(k\Omega)}$$

where $$A(k\Omega) = |\tilde{X}(jk\Omega)| \qquad \text{and} \qquad \phi(k\Omega) = \arg \tilde{X}(jk\Omega)$$

are discrete-frequency functions. The functions $A(k\Omega)$ and $\phi(k\Omega)$ are referred to as the *amplitude spectrum* and *phase spectrum* of $\tilde{x}(nT)$, respectively. They are entirely analogous to the corresponding spectrums of continuous-time and nonperiodic discrete-time signals.

7.3 INVERSE DFT

The function $\tilde{x}(nT)$ is said to be the *inverse* DFT (IDFT) of $\tilde{X}(jk\Omega)$ and is given by

$$\tilde{x}(nT) = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}(jk\Omega)W^{kn} = \mathcal{D}^{-1}\tilde{X}(jk\Omega) \tag{7.2}$$

The validity of the inverse DFT can be demonstrated by eliminating the DFT in Eq. (7.2) using Eq. (7.1). We can write

$$\begin{aligned}\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(jk\Omega)W^{kn} &= \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{m=0}^{N-1}\tilde{x}(mT)W^{-km}\right]W^{kn}\\ &= \frac{1}{N}\sum_{m=0}^{N-1}\tilde{x}(mT)\sum_{k=0}^{N-1}W^{k(n-m)}\end{aligned}$$

where one can show that

$$\sum_{k=0}^{N-1}W^{k(n-m)} = \begin{cases} N & \text{for } m = n\\ 0 & \text{otherwise}\end{cases}$$

(see Prob. 7.1). Therefore

$$\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(jk\Omega)W^{kn} = \tilde{x}(nT)$$

7.4 PROPERTIES

7.4.1 Linearity

The DFT obeys the law of *linearity*; that is, for any two constants a and b,

$$\mathcal{D}[a\tilde{x}(nT) + b\tilde{y}(nT)] = a\tilde{X}(jk\Omega) + b\tilde{Y}(jk\Omega)$$

7.4.2 Periodicity

From Eq. (7.2)

$$\begin{aligned}\tilde{X}[j(k+rN)\Omega] &= \sum_{n=0}^{N-1}\tilde{x}(nT)W^{-(k+rN)n} = \sum_{n=0}^{N-1}\tilde{x}(nT)W^{-kn}\\ &= \tilde{X}(jk\Omega)\end{aligned}$$

since $W^{-rnN} = 1$. In effect, $\tilde{X}(jk\Omega)$ is a *periodic* function of $k\Omega$ with period $N\Omega$ $(= \omega_s)$.

7.4.3 Symmetry

The DFT has certain *symmetry* properties that are often useful. For example, assuming that $\tilde{x}(nT)$ is real, we have

$$\begin{aligned}\tilde{X}[j(N-k)\Omega] &= \sum_{n=0}^{N-1}\tilde{x}(nT)W^{-(N-k)n} = \sum_{n=0}^{N-1}\tilde{x}(nT)W^{kn}\\ &= \left[\sum_{n=0}^{N-1}\tilde{x}(nT)W^{-kn}\right]^{*} = \tilde{X}^{*}(jk\Omega)\end{aligned}$$

and as a result

$$\Re e\,\tilde{X}[j(N-k)\Omega] = \Re e\,\tilde{X}(jk\Omega) \qquad \Im m\,\tilde{X}[j(N-k)\Omega] = -\,\Im m\,\tilde{X}(jk\Omega)$$

$$A[(N-k)\Omega] = A(k\Omega) \qquad \phi[(N-k)\Omega] = -\phi(k\Omega) + 2\pi r$$

where r is any integer. If $\tilde{x}(nT)$ is real and, in addition,

$$\tilde{x}(nT) = \pm\tilde{x}[(N-n)T]$$

we obtain

$$\tilde{X}(jk\Omega) = \pm\sum_{n=0}^{N-1}\tilde{x}[(N-n)T]W^{-kn} = \pm\sum_{m=0}^{N-1}\tilde{x}(mT)W^{-k(N-m)}$$

$$= \pm\left[\sum_{n=0}^{N-1}\tilde{x}(nT)W^{-kn}\right]^* = \pm\tilde{X}^*(jk\Omega)$$

Thus if

$$\tilde{x}(nT) = \tilde{x}[(N-n)T]$$

we have

$$\Im m\,\tilde{X}(jk\Omega) = 0$$

and if

$$\tilde{x}(nT) = -\tilde{x}[(N-n)T]$$

then

$$\Re e\,\tilde{X}(jk\Omega) = 0$$

Example 7.1 Find the DFT of $\tilde{x}(nT)$ if

$$\tilde{x}(nT) = \begin{cases} 1 & \text{for } 2 \le n \le 6 \\ 0 & \text{for } n = 0, 1, 7, 8, 9 \end{cases}$$

assuming that $N = 10$.

■ **Solution**

By noting that the DFT is a geometric series (see Sec. A.5) with a common ratio W^{-k}, Eq. (7.1) gives

$$\tilde{X}(jk\Omega) = \sum_{n=2}^{6} W^{-kn} = \frac{W^{-2k} - W^{-7k}}{1 - W^{-k}}$$

$$= e^{-j4\pi k/5}\frac{\sin(\pi k/2)}{\sin(\pi k/10)} \quad ■$$

The amplitude and phase spectrums of $\tilde{x}(nT)$ can be obtained as

$$A(k\Omega) = \left|\frac{\sin(\pi k/2)}{\sin(\pi k/10)}\right|$$

and

$$\phi(k\Omega) = C\pi - \frac{4\pi k}{5} \qquad \text{where } C = \begin{cases} 1 & \text{if } \dfrac{\sin(\pi k/2)}{\sin(\pi k/10)} < 0 \\ 0 & \text{otherwise} \end{cases}$$

and are plotted in Fig. 7.2.

7.5 INTERRELATION BETWEEN THE DFT AND THE *Z* TRANSFORM

The DFT is defined in Sec. 7.2 in terms of periodic discrete-time signals. Nevertheless, it can also be applied for the analysis of nonperiodic discrete-time signals as long as they are of finite duration. Given an arbitrary finite-duration discrete-time nonperiodic signal $x(nT)$ such that

$$x(nT) = 0 \qquad \text{for } n < 0 \qquad \text{and} \qquad n > M$$

where $M < N$, then a corresponding *periodic signal* $\tilde{x}(nT)$ with period NT can be readily formed as

$$\tilde{x}(nT) = \sum_{r=-\infty}^{\infty} x(nT + rNT) \tag{7.3}$$

by analogy with Eq. (2.1). Since $\tilde{x}(nT)$ can be derived from $x(nT)$, the DFT of $\tilde{x}(nT)$ must be related to the frequency spectrum and, in turn, to the z transform of $x(nT)$ as will now be demonstrated.

From Eqs. (7.1) and (7.3), we can write

$$\tilde{X}(jk\Omega) = \sum_{n=0}^{N-1}\sum_{r=-\infty}^{\infty} x(nT + rNT)W^{-kn} = \sum_{r=-\infty}^{\infty}\sum_{n=0}^{N-1} x(nT + rNT)W^{-kn}$$

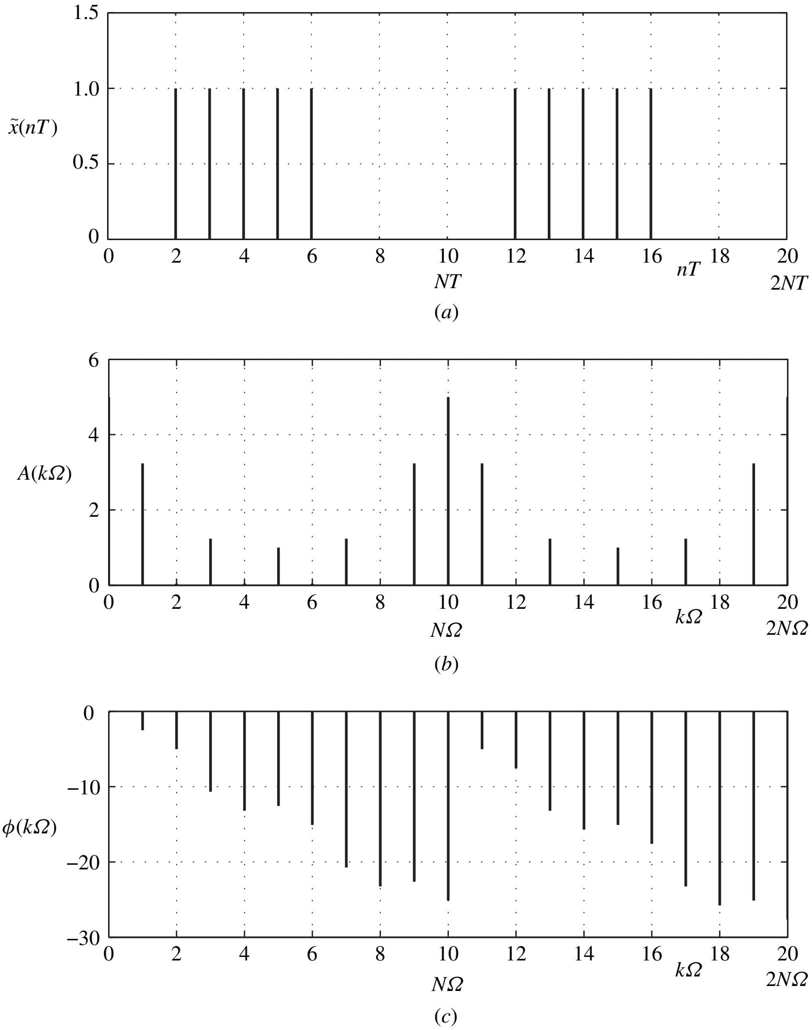


Figure 7.2 DFT of $\tilde{x}(nT)$ (Example 7.1): (*a*) Periodic discrete-time signal, (*b*) amplitude spectrum, (*c*) phase spectrum.

and by letting $n = m - rN$, we have

$$\begin{aligned}\tilde{X}(jk\Omega) &= \sum_{r=-\infty}^{\infty} \sum_{m=rN}^{rN+N-1} x(mT)W^{-k(m-rN)} \\ &= \cdots + \sum_{m=-N}^{-1} x(mT)W^{-km} + \sum_{m=0}^{N-1} x(mT)W^{-km} + \sum_{m=N}^{2N-1} x(mT)W^{-km} + \cdots \\ &= \sum_{m=-\infty}^{\infty} x(mT)W^{-km}\end{aligned}$$

Alternatively, on replacing W by $e^{j2\pi/N}$ and m by n we have

$$\tilde{X}(jk\Omega) = \sum_{n=-\infty}^{\infty} x(nT)e^{-jk\Omega nT}$$

and, therefore,
$$\tilde{X}(jk\Omega) = X_D(e^{jk\Omega T}) \tag{7.4}$$

where
$$X_D(z) = \mathcal{Z}x(nT)$$

In effect, *the DFT of a periodic signal $\tilde{x}(nT)$ is a sampled version of the frequency spectrum of the nonperiodic signal $x(nT)$*, which is obtained by evaluating the z transform of $x(nT)$ on the unit circle $|z| = 1$, as shown in Fig. 7.3.

Through a technique known as *zero padding*, the above interrelationship can be used to facilitate the computation of the frequency spectrum of a finite-duration discrete-time signal to any desired resolution, as will now be demonstrated.

Consider a nonperiodic signal $x(nT)$ which is zero outside the interval 0 to $(L-1)T$. Such a signal has a frequency spectrum

$$X_D(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} x(nT)z^{-n}\Big|_{z=e^{j\omega T}} = \sum_{n=0}^{L-1} x(nT)e^{-jn\omega T}$$

If we let $\omega = k\omega_s/N = 2\pi k/NT$, we get

$$X_D(e^{jk\omega_s/N}) = \sum_{n=0}^{L-1} x(nT)e^{-j2\pi kn/N} \tag{7.5}$$

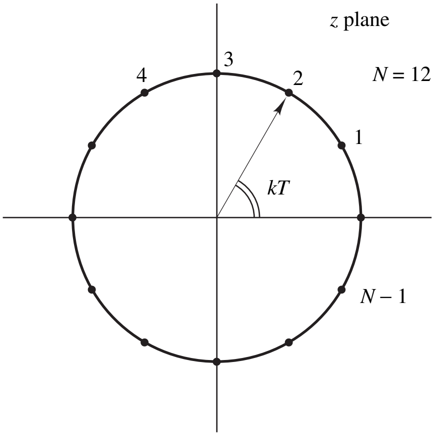


Figure 7.3 Relation between $\tilde{X}(jk\Omega)$ and $X_D(e^{jk\Omega T})$.

Let us construct a periodic signal $\tilde{x}(nT)$ with period NT such that

$$\tilde{x}(nT) = \begin{cases} x(nT) & \text{for } 0 \le n \le L-1 \\ 0 & \text{for } L \le n \le N-1 \end{cases} \tag{7.6}$$

through periodic continuation (see Eq. (7.3)). From the definition of the DFT, we have

$$\begin{aligned} \tilde{X}(jk\Omega) &= \sum_{n=0}^{N-1} \tilde{x}(nT)W^{-kn} \\ &= \sum_{n=0}^{L-1} x(nT)e^{-j2\pi kn/N} \end{aligned} \tag{7.7}$$

If we now compare Eqs. (7.5) and (7.7), we conclude that

$$X_D(e^{jk\omega_s/N}) = \tilde{X}(jk\Omega) \tag{7.8}$$

As was shown earlier, the DFT gives a sampled version of the frequency spectrum of a discrete-time signal and by increasing the number of zeros in Eq. (7.6), $\Omega = \omega_s/N$ is reduced and, consequently, a higher density of sample points can be achieved.

The zero padding technique is illustrated in Fig. 7.4 which shows the DFT of Example 7.1 with the value of N increased from 10 to 20. As can be seen by comparing Fig. 7.4*b* and *c* with Fig. 7.2*b* and *c*, the density of samples in the amplitude and phase spectrums is doubled.

7.5.1 Frequency-Domain Sampling Theorem

The application of the DFT to DSP is made possible by the *frequency-domain sampling theorem* which is analogous to the time-domain sampling theorem considered in Sec. 6.6 except that the roles of time and frequency are interchanged.

Theorem 7.1 Frequency-Domain Sampling Theorem ***A z transform $X_D(z)$ whose inverse satisfies the finite-duration condition***

$$\boldsymbol{x(nT) = \mathcal{Z}^{-1}X_D(z) = 0 \qquad \textit{for } n < 0 \qquad \textit{and} \quad n \ge N} \tag{7.9}$$

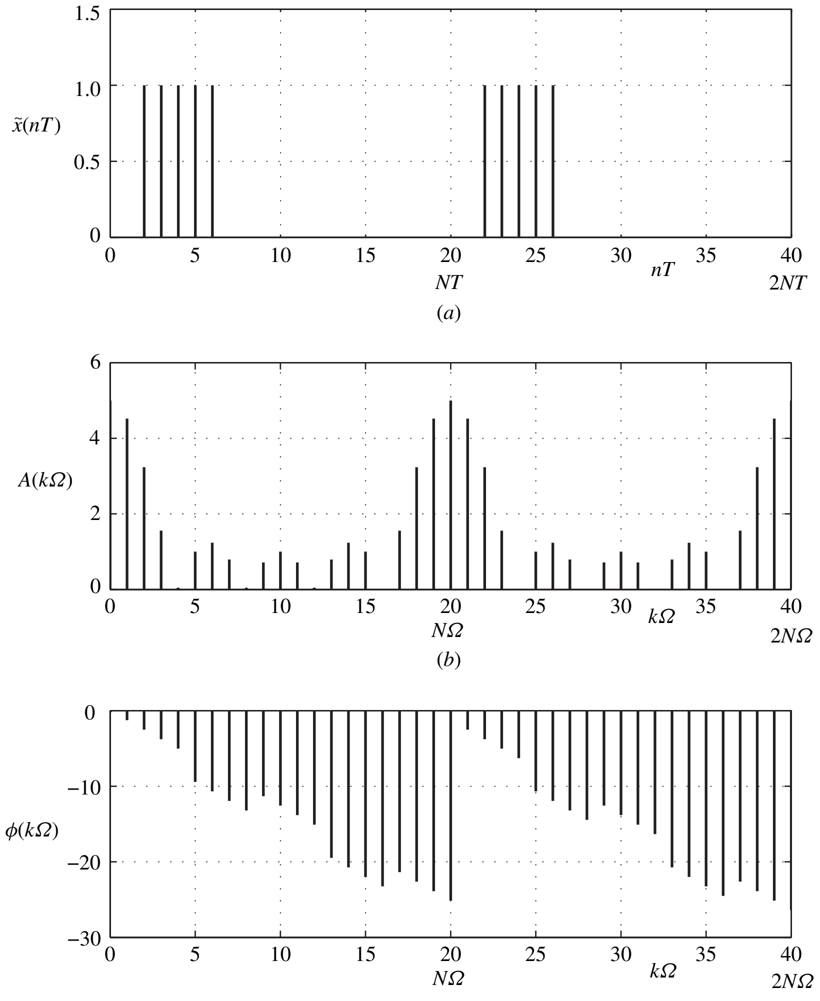
can be uniquely determined from its values $X_D(e^{jk\Omega T})$ or, equivalently, from the values of the DFT of $\tilde{x}(nT)$, $\tilde{X}(jk\Omega)$, by virtue of Eq. (7.4). ▲

The validity of Theorem 7.1 can be easily demonstrated. If the condition in Eq. (7.9) is satisfied, then $\tilde{x}(nT)$ as given by Eq. (7.3) is a periodic continuation of $x(nT)$ and $x(nT)$ can be isolated by multiplying $\tilde{x}(nT)$ by a discrete-time pulse as illustrated in Fig. 7.5. We can write

$$x(nT) = [u(nT) - u(nT - NT)]\tilde{x}(nT) \tag{7.10}$$

and so

$$X_D(z) = \mathcal{Z}\{[u(nT) - u(nT - NT)]\tilde{x}(nT)\}$$



(*a*)

(*b*)

(*c*)

Figure 7.4 Effect of zero padding: (*a*) Periodic discrete-time signal of Example 7.1 with zero padding, (*b*) amplitude spectrum, (*c*) phase spectrum.

Now

$$\tilde{x}(nT) = \mathcal{D}^{-1}\tilde{X}(jk\Omega) \qquad \text{where } \tilde{X}(jk\Omega) = X_D(e^{jk\Omega T})$$

and from Eq. (7.2)

$$\begin{aligned} X_D(z) &= \mathcal{Z}\left\{[u(nT) - u(nT - NT)]\frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(jk\Omega)W^{kn}\right\} \\ &= \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}(jk\Omega)\mathcal{Z}\{[u(nT) - u(nT - NT)]W^{kn}\} \end{aligned}$$

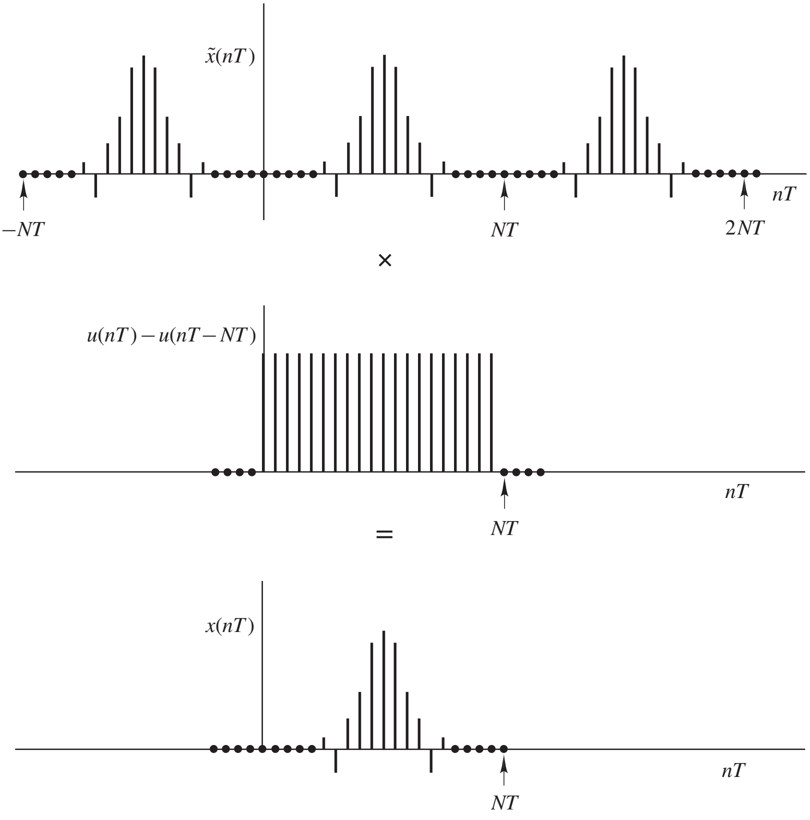


Figure 7.5 Derivation of $x(nT)$ from $\tilde{x}(nT)$.

The z transform of the unit-step is given by

$$\mathcal{Z}u(nT) = \frac{z}{z-1}$$

Therefore, from the time-shifting and complex-scale-change theorems (Theorems 3.4 and 3.5), we have

$$X_D(z) = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}(jk\Omega)\frac{1-z^{-N}}{1-W^k z^{-1}} \tag{7.11}$$

since $W^{-kN} = 1$.

In summary, if the nonperiodic signal $x(nT)$ is zero outside the range $0 \le nT \le (N-1)T$, then the periodic signal $\tilde{x}(nT)$ and its DFT $\tilde{X}(jk\Omega)$ can be obtained from $x(nT)$ and $X_D(z)$ by using Eqs. (7.3) and (7.4), respectively. Conversely, $x(nT)$ and $X_D(z)$ can be obtained from $\tilde{x}(nT)$ and $\tilde{X}(jk\Omega)$ by using Eqs. (7.10) and (7.11), respectively, as illustrated in Fig. 7.6. Therefore, $x(nT)$ can be represented by the DFT of $\tilde{x}(nT)$. As a result, any finite-duration discrete-time signal can be

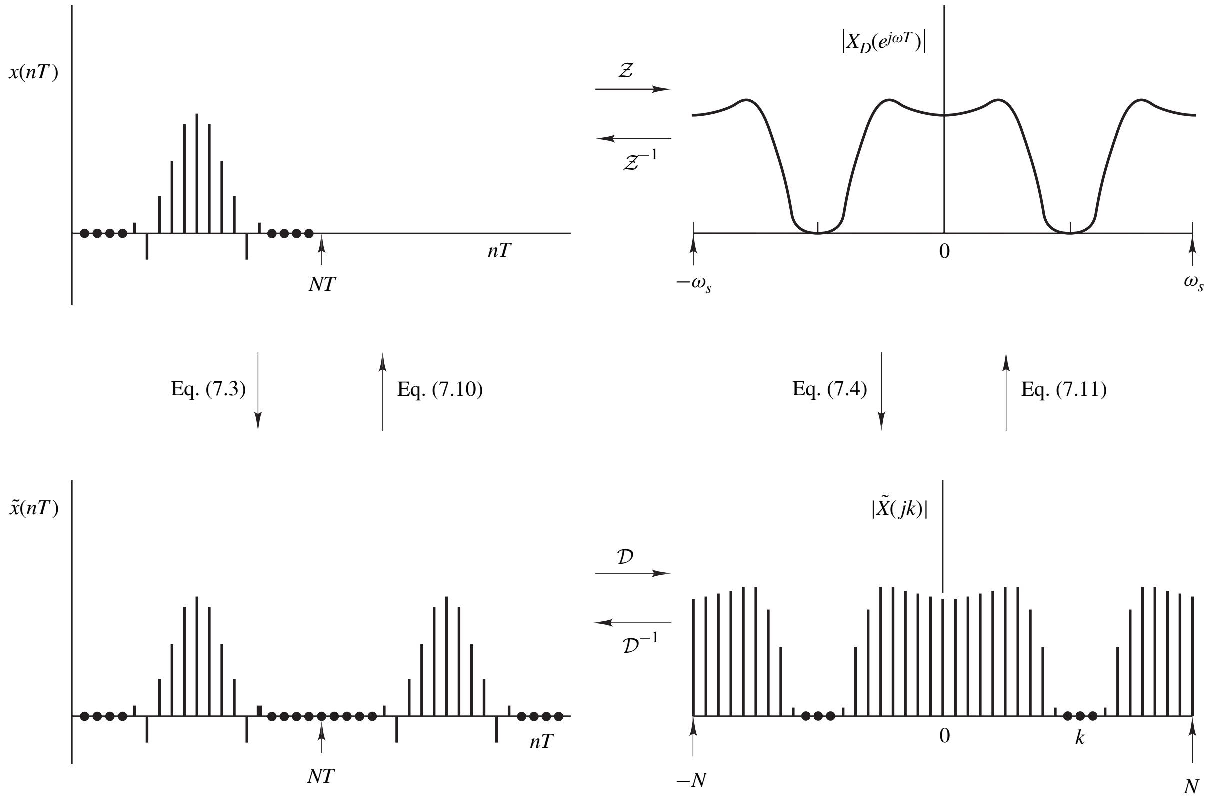


Figure 7.6 Interrelations between the DFT and the z transform.

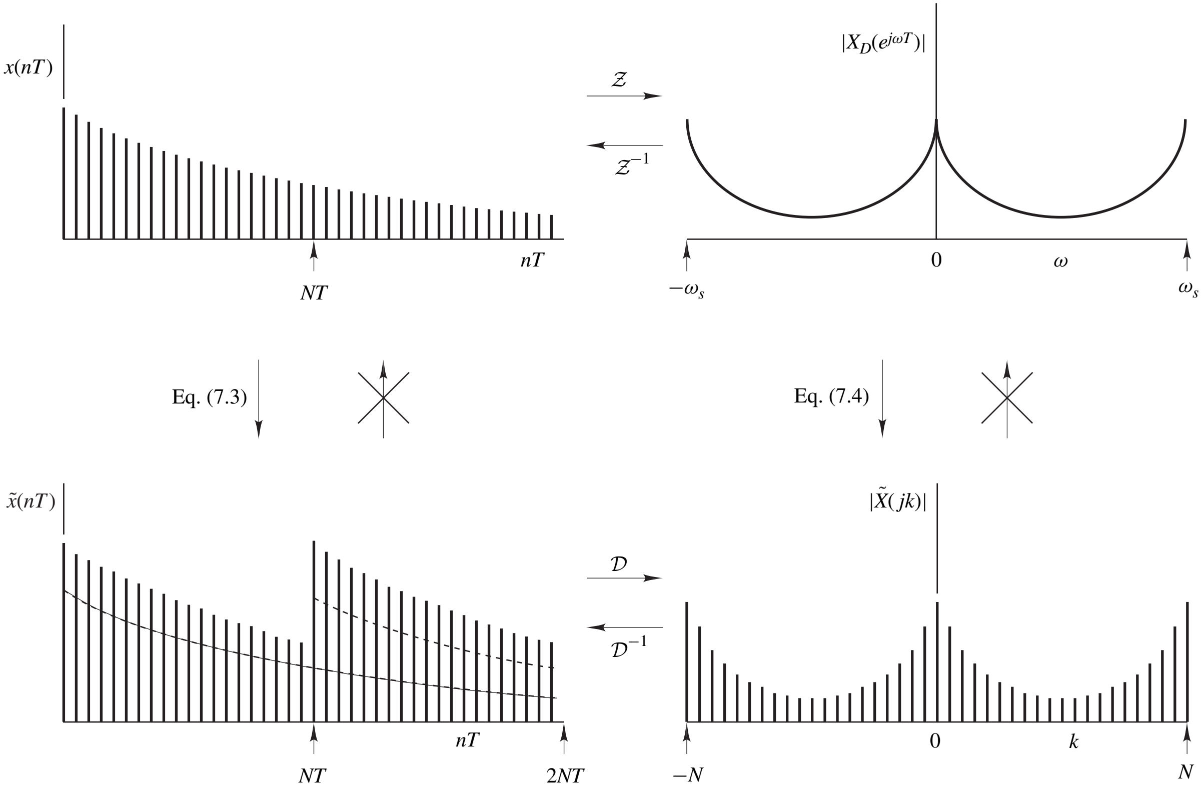


Figure 7.7 Time-domain aliasing.

processed by employing the FFT algorithms to be described in Sec. 7.11 provided that a sufficiently large value of N is chosen.

7.5.2 Time-Domain Aliasing

If

$$x(nT) \neq 0 \qquad \text{for } n < 0 \qquad \text{or} \qquad n \geq N$$

then parts of one or more of the shifted signal copies $x(nT \pm NT), x(nT \pm 2NT), \ldots$ in Eq. (7.3) will contribute to the value of $\tilde{x}(nT)$ in the range $0 \leq t < NT$ thereby causing so-called *time-domain aliasing* which is analogous to the frequency-domain aliasing encountered in Sec. 6.7. Consequently, $\tilde{x}(nT)$ will not be a periodic continuation of $x(nT)$, as illustrated in Fig. 7.7. Nevertheless, the periodic signal $\tilde{x}(nT)$ and its DFT $\tilde{X}(jk\Omega)$ can again be obtained from the nonperiodic signal $x(nT)$ and its z transform $X_D(z)$, respectively, by using Eqs. (7.3) and (7.4), as illustrated in Fig. 7.7. However, in such a case, $x(nT)$ *cannot* be recovered from $\tilde{x}(nT)$ by using Eq. (7.10), because of the signal distortion introduced by time-domain aliasing, and so Eq. (7.11) will not yield the z transform of $x(nT)$. Under these circumstances, the DFT of $\tilde{x}(nT)$ is at best a distorted representation for $x(nT)$.

7.6 INTERRELATION BETWEEN THE DFT AND THE CFT

The frequency spectrum of a nonperiodic discrete-time signal $x(nT)$ is related to that of the underlying continuous-time signal $x(t)$, as was shown in Sec. 6.5.2. On the other hand, given a finite-duration discrete-time signal $x(nT)$, a corresponding periodic discrete-time signal $\tilde{x}(nT)$ can be generated by using Eq. (7.3), which has a DFT. Consequently, a direct interrelation exists between the DFT of $\tilde{x}(nT)$ and the CFT [5] of $x(t)$. This can be readily established by using the results of Secs. 6.5.2 and 7.5.

Let $X(j\omega)$ and $\hat{X}(j\omega)$ be the CFTs of $x(t)$ and $\hat{x}(t)$, respectively, where $\hat{x}(t)$ is the impulse-modulated version of $x(t)$. From Eqs. (7.4) and (6.45a), we have

$$\hat{X}(jk\Omega) = X_D(e^{jk\Omega T}) = \tilde{X}(jk\Omega) = \frac{1}{T}\sum_{r=-\infty}^{\infty} X(jk\Omega + jr\omega_s) \tag{7.12a}$$

and, therefore, from Eqs. (7.3) and (7.12a)

$$\mathcal{D}\sum_{r=-\infty}^{\infty} x(nT + rNT) = \frac{1}{T}\sum_{r=-\infty}^{\infty} X(jk\Omega + jr\omega_s) \tag{7.12b}$$

Now if

$$x(t) = 0 \qquad \text{for } t < 0 \qquad \text{and} \qquad t \geq NT \tag{7.13}$$

and

$$X(j\omega) = 0 \qquad \text{for } |\omega| \geq \frac{\omega_s}{2} \tag{7.14}$$

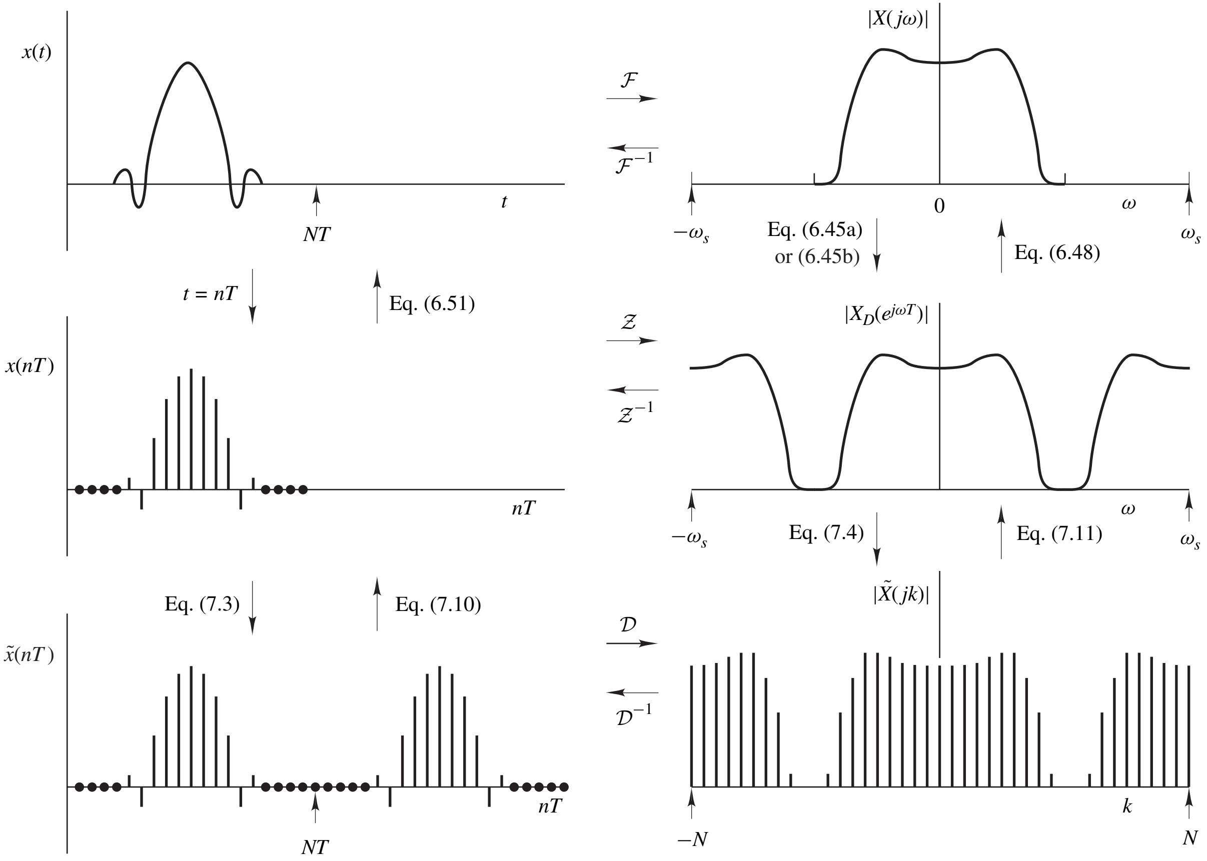


Figure 7.8 Interrelations between the DFT and the CFT.

the left- and right-hand summations in Eq. (7.12b) become *periodic continuations* of $x(nT)$ and $X(j\omega)$, respectively, and as a result

$$\tilde{x}(nT) = x(nT) \qquad \text{for } 0 \le nT \le (N-1)T$$
$$\tilde{X}(jk\Omega) = \frac{1}{T}X(jk\Omega) \qquad \text{for } |k\Omega| < \frac{\omega_s}{2}$$

Hence the periodic signal $\tilde{x}(nT)$ and its DFT $\tilde{X}(jk\Omega)$ can be obtained from the continuous-time signal $x(t)$ and its CFT $X(j\omega)$, respectively, and conversely, as depicted in Fig. 7.8. That is, a continuous-time signal $x(t)$ can be represented by a DFT and, accordingly, it can be processed by using the FFT method.

7.6.1 Time-Domain Aliasing

If Eq. (7.13) is violated, then parts of one or more of the shifted signal copies $x(t \pm NT)$, $x(t \pm 2NT)$, ... will contribute to the value of $x(t)$ in the range $0 \le t < NT$ thereby causing time-domain aliasing and, as in the case of discrete-time signals described in Sec. 7.5.2, $x(t)$ cannot be recovered from $\tilde{x}(t)$. On the other hand, if the spectrum of $x(t)$ does not satisfy the bandlimiting condition in Eq. (7.14), then $X(j\omega)$ cannot be recovered from the DFT of $\tilde{x}(t)$.

7.7 INTERRELATION BETWEEN THE DFT AND THE FOURIER SERIES

The results of the previous section lead directly to a relationship between the DFT and the Fourier series [5].

A periodic signal $\tilde{x}(t)$ with a period τ_0 can be expressed as

$$\tilde{x}(t) = \sum_{r=-\infty}^{\infty} x(t + r\tau_0) \tag{7.15}$$

where $x(t) = 0$ for $t < 0$ and $t \ge \tau_0$[1]. Alternatively, by using the Fourier series (see Eqs. (2.3) and (2.5)), we have

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t}$$

where $\omega_0 = 2\pi/\tau_0$ and

$$X_k = \frac{1}{\tau_0}\int_0^{\tau_0} x(t)e^{-jk\omega_0 t}dt$$

[1] *Note that $x(t)$ is defined to be zero for $t \le -\tau_0/2$ and $t > \tau_0/2$ in Chap. 2 but the two definitions are equivalent.*

Now with $t = nT$ and $\tau_0 = NT$, Eq. (7.15) becomes

$$\tilde{x}(nT) = \sum_{r=-\infty}^{\infty} x(nT + rNT)$$

and, consequently, Eq. (7.12a) yields

$$\tilde{X}(jk\Omega) = \frac{1}{T} \sum_{r=-\infty}^{\infty} X(jk\Omega + jr\omega_s) \tag{7.16}$$

where
$$X(jk\Omega) = \mathcal{F}x(t)\Big|_{\omega=k\Omega} = \int_0^{\tau_0} x(t)e^{-jk\Omega t}\,dt$$

or
$$X(jk\Omega) = \int_0^{\tau_0} x(t)e^{-jk\omega_0 t}\,dt$$

since $\Omega = \omega_s/N = 2\pi/NT = 2\pi/\tau_0 = \omega_0$. Evidently

$$X(jk\Omega) = \tau_0 X_k$$

and since $\tau_0 = NT$, Eq. (7.16) can be put in the form

$$\tilde{X}(jk\Omega) = \frac{1}{T} \sum_{r=-\infty}^{\infty} X[j(k + rN)\Omega] = N \sum_{r=-\infty}^{\infty} X_{k+rN} \tag{7.17}$$

In effect, *the DFT of $\tilde{x}(nT)$ can be expressed in terms of the Fourier-series coefficients of $\tilde{x}(t)$.*

Now with

$$X_k \approx 0 \qquad \text{for } |k| \geq \frac{N}{2}$$

Eq. (7.17) gives

$$\tilde{X}(jk\Omega) \approx NX_k \qquad \text{for } |k| < \frac{N}{2}$$

or
$$X_k \approx \frac{1}{N}\tilde{X}(jk\Omega) \qquad \text{for } |k| < \frac{N}{2}$$

Thus the Fourier-series coefficients of $\tilde{x}(t)$ can be efficiently computed by using the FFT method.

7.8 WINDOW TECHNIQUE

As was demonstrated in Secs. 7.5 and 7.6, in order to process a discrete- or continuous-time signal, say, $x(nT)$ or $x(t)$, through the FFT method, a corresponding periodic discrete-time signal

$$\tilde{x}(nT) = \sum_{r=-\infty}^{\infty} x(nT + rT)$$

must first be generated through periodic continuation. If $x(nT) \neq 0$ or $x(t) \neq 0$ outside the time range 0 to $(N-1)T$, then time-domain aliasing will be introduced, as was shown in Secs. 7.5.2 and 7.6.1, which will distort the signal over the period 0 to $(N-1)T$. The effects of time-domain aliasing can be reduced by forcing the signal to become zero outside the time range 0 to $(N-1)T$ in such a way so as to as far as possible preserve the signal's frequency spectrum. This can be done through the use of the so-called *window technique*.

The window technique entails the use of a class of functions known collectively as *window functions* (or simply as *windows*) [6, 7]. The underlying windows are continuous-time functions but they are usually applied in their nonperiodic or periodic discrete-time form. Discrete-time windows inherit their characteristics from their continuous-time counterparts. Consequently, it is worthwhile to study the basic principles associated with continuous-time windows before we proceed to the properties and application of discrete-time windows. For the sake of simplicity, we consider the case of two-sided windows that are defined over the range $-\infty \leq t \leq \infty$ but the principles involved are also valid for right-sided windows that are defined over the range $0 \leq t \leq \infty$.

7.8.1 Continuous-Time Windows

Two-sided window functions share two fundamental properties:

1. They are defined to be zero *outside* some interval $-\tau \leq t \leq \tau$, that is,

$$w(t) = \begin{cases} w_0(t) & \text{for } -\tau \leq t \leq \tau \\ 0 & \text{otherwise} \end{cases}$$

where $w_0(t)$ is normally an even function of t.

2. Their spectral energy[2] is concentrated in a narrow frequency range $-\omega_m \leq \omega \leq \omega_m$ and is zero outside this range.

The most basic window is the rectangular window which is defined as

$$w_R(t) = \begin{cases} 1 & \text{for } -\tau \leq t \leq \tau \\ 0 & \text{otherwise} \end{cases} \tag{7.18}$$

[2] *The spectral energy of a signal with respect to a range of frequencies ω_1 to ω_2 is proportional to the integral of the square of the amplitude spectrum of the signal over the given frequency range.*

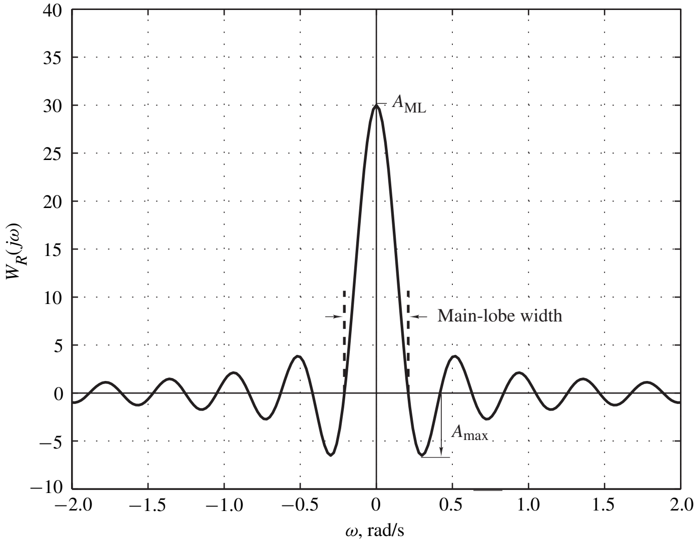


Figure 7.9 Spectrum of rectangular window ($\tau = 15.0$ s).

The frequency spectrum of the rectangular window is given by the Fourier transform of the pulse function $p_\tau(t)$ (see Table 6.2) as

$$W_R(j\omega) = \frac{2\sin\omega\tau}{\omega} \tag{7.19}$$

and is illustrated in Fig. 7.9.

Windows are characterized by their *main-lobe width* B_{ML}, which is the bandwidth between the first negative and the first positive zero crossings, and by their *ripple ratio r* which is defined as

$$r = 100\frac{A_{\text{max}}}{A_{\text{ML}}}\ \% \qquad \text{or} \qquad R = 20\log\frac{A_{\text{max}}}{A_{\text{ML}}}\ \text{dB}$$

where A_{max} and A_{ML} are the maximum side-lobe and main-lobe amplitudes, respectively (see Fig. 7.9).

From Eq. (7.19), we note that the frequency spectrum of the rectangular window has zero crossings when $\omega\tau = \pm m\pi$ for $m = 1,\ 2, \ldots$ and hence $B_{\text{ML}} = 2\pi/\tau$. On the other hand, the ripple ratio can be plotted as a function of τ as shown in Fig. 7.10. Evidently, the ripple ratio assumes a value of about -13.26 dB which is practically independent of the window length τ. This is a fairly large ripple ratio which tends to limit the usefulness of the rectangular window.

The window technique entails multiplying the signal to be processed by the window function, i.e.,

$$x_w(t) = w(t)x(t)$$

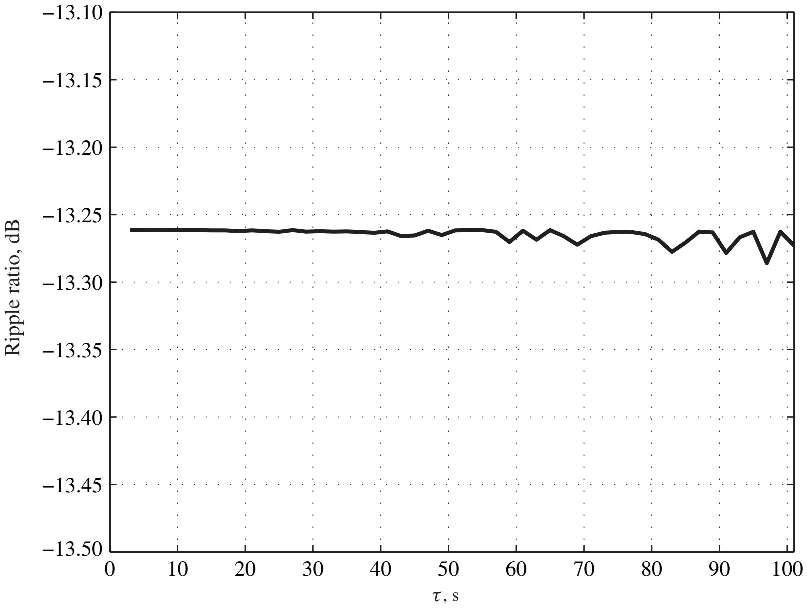


Figure 7.10 Ripple ratio of rectangular window versus window length τ.

In order to examine the effect of a window on the frequency spectrum of a signal, consider a signal that has a uniform real spectrum over the frequency range $-\omega_c < \omega < -\omega_c$, that is,

$$X(j\omega) = \begin{cases} 1 & \text{for } |\omega| < \omega_c \\ 0 & \text{otherwise} \end{cases}$$

as shown in Fig. 7.11*a* and assume that the spectrum of $w(t)$ is concentrated in the range $-\omega_m \leq \omega \leq \omega_m$ and is zero outside this range, i.e.,

$$W(j\omega) = \begin{cases} W_0(\omega) & \text{for } |\omega| \leq \omega_m \\ 0 & \text{otherwise} \end{cases}$$

where $W_0(\omega)$ is an even real function of ω as depicted in Fig. 7.11*b*. For the sake of convenience and without loss of generality, assume that the area under the graph of $W(j\omega)$ is equal to 2π. The reason for this will become evident below.

Since $x_w(t)$ is the product of two time-domain functions, the spectrum of the modified signal can be obtained by using the frequency convolution of the Fourier transform (Theorem 2.15) as

$$X_w(j\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} W(j\varpi)X(j\omega - j\varpi)\,d\varpi \tag{7.20a}$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\varpi)W(j\omega - j\varpi)\,d\varpi \tag{7.20b}$$

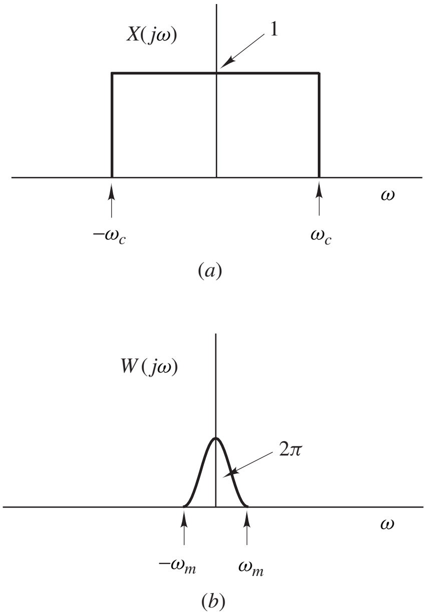


Figure 7.11 Frequency spectrums of $x(t)$ and $w(t)$.

Using Eq. (7.20b), the spectrum of $x_w(t)$ at some arbitrary frequency ω can be obtained through the following graphical construction:

1. Replace the frequency variable ω in $X(j\omega)$ and $W(j\omega)$ by variable ϖ to obtain $X(j\varpi)$ and $W(j\varpi)$, respectively, as shown in Fig. 7.12*a* and *b*.
2. Replace ϖ in $W(j\varpi)$ by $-\varpi$ in Fig. 7.12*b*, that is, turn over the spectrum of $w(t)$ end-to-end with respect to the ϖ axis, to obtain $W(-j\varpi)$. This step would have no effect in the present application because the spectrum of the window has been assumed to be an even function of ω and it is, therefore, symmetrical about the y axis.
3. Shift $W(-j\varpi)$ to the right by ω to obtain $W(j\omega - j\varpi)$ as shown in Fig. 7.12*c*.
4. Form the product $X(j\varpi)W(j\omega - j\varpi)$ as shown in Fig. 7.12*d*.
5. Area $A(j\omega)$ in Fig. 7.12*d* is given by

$$A(j\omega) = \int_{-\infty}^{\infty} X(j\varpi)W(j\omega - j\varpi)\, d\varpi$$

and from Eq. (7.20b) we note that $X_w(j\omega) = A(j\omega)/2\pi$. Since the area under the graph $W(j\varpi)$ has been assumed to be equal to 2π, $A(j\omega)$ would be equal to 2π in the frequency range 0 to $\omega_c - \omega_m$ and a fraction of that amount in the range $\omega_c - \omega_m$ to $\omega_c + \omega_m$. In effect, $X_w(j\omega)$ would assume the value of unity for frequencies in the range 0 to $\omega_c - \omega_m$ and a fraction between 1 and 0 in the range $\omega_c - \omega_m$ to $\omega_c + \omega_m$.

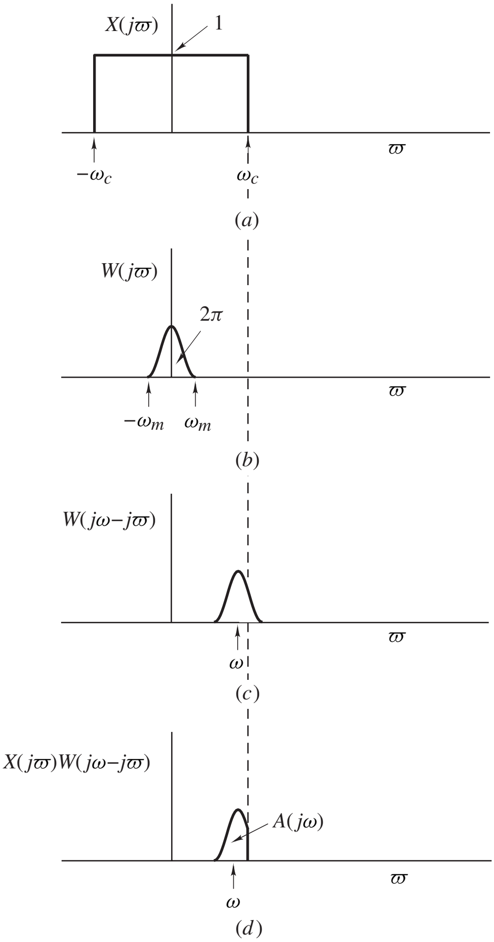


Figure 7.12 Evaluation of convolution integral.

If ω is varied in the range ω_1 to ω_5 through the point of discontinuity $\omega = \omega_c$, the above graphical construction will give Fig. 7.13*c* to *f* and by multiplying each of the areas $A(j\omega_1)$, $A(j\omega_2)$,... by the factor $1/2\pi$, the frequency spectrum of the modified signal can be obtained as illustrated in Fig. 7.13*g*. Since the areas $A(j\omega_1)$, $A(j\omega_2)$, ... assume values in the range 2π to 0, it follows that $X_w(j\omega)$ will assume values in the range 1 to 0. As can be seen in Fig. 7.13g, the spectrum of the modified signal is the same as that of the original signal except that a transition band has been introduced at the point of discontinuity $\omega = \omega_c$. The same would also happen at any other discontinuity.

From Fig. 7.13*g*, we observe that the width of the transition band introduced is equal to $2\omega_m$, which is the main-lobe width of the window. Therefore, the main-lobe width should, in practice, be as small as possible.

The window considered in the above demonstration is actually an idealization. In practice, the window has side lobes and, therefore, the spectral energy outside the main lobe is never zero. If an

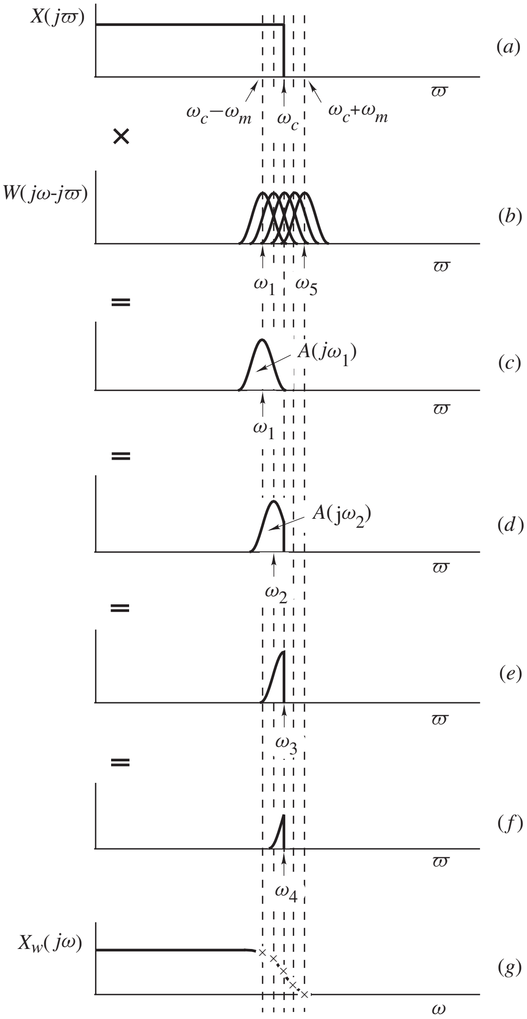


Figure 7.13 Effect of an idealized window on the spectrum of a signal.

actual window were to be used in the graphical construction of Fig. 7.13, as illustrated in Fig. 7.14, then the areas under the graphs in Fig. 7.14*d* and *e*, namely, $A(j\omega_0)$ and $A(j\omega_6)$, respectively, would be slightly different from 2π in the first case and from zero in the second case due to the area of the side lobes. Similarly, the spectrum of $x_w(t)$ would be slightly different from unity and 0 at frequencies ω_0 and ω_6, respectively, as shown in Fig. 7.14*f*. Now as ω is varied, the area of the side ripples that happen to fall in the range $0 < \omega < \omega_c$ will vary and, consequently, the spectrum of the modified signal will exhibit oscillations whose amplitude would be directly related to the amplitude of the window side ripples. We conclude, therefore, that the amplitude of side lobes should, in practice, be

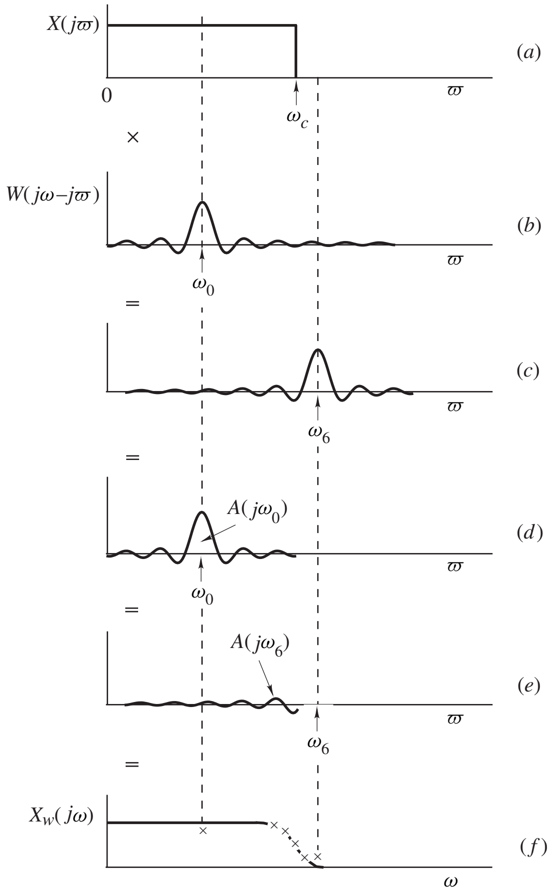


Figure 7.14 Effect of window side lobes.

as small as possible in relation to the amplitude of the main lobe, i.e., the ripple ratio should be as small as possible.

Example 7.2 The infinite-duration signal

$$x(t) = \frac{\sin \omega_c t}{\pi t}$$

where $\omega_c = 1$ rad/s, is to be converted into a corresponding finite-duration signal $x_w(t)$ by using the rectangular window defined in Eq. (7.18) with $\tau = 15.0$ s. (*a*) Obtain $X(j\omega)$, the frequency spectrum of $x(t)$. (*b*) Repeat part (*a*) for the modified signal $x_w(t)$. (*c*) Plot the difference between $X(j\omega)$ and $X_w(j\omega)$.

■ **Solution**

(a) From Table 6.2, we have

$$X(j\omega) = p_{2\omega_c}(\omega) \quad ■ \tag{7.21}$$

where

$$p_{2\omega_c}(\omega) = \begin{cases} 1 & \text{for } |\omega| < \omega_c \\ 0 & \text{otherwise} \end{cases}$$

(b) From Eqs. (7.19), (7.20b), and (7.21), the frequency spectrum of the modified signal is obtained as

$$\begin{aligned} X_w(j\omega) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\varpi)W_R(j\omega - j\varpi)\,d\varpi \\ &= \frac{1}{\pi}\int_{-\infty}^{\infty} p_{2\omega_c}(\varpi)\frac{\sin(\omega-\varpi)\tau}{\omega-\varpi}\,d\varpi \\ &= \frac{1}{\pi}\int_{-\omega_c}^{\omega_c} \frac{\sin(\omega-\varpi)\tau}{\omega-\varpi}\,d\varpi \quad ■ \qquad (7.22) \end{aligned}$$

The real integral in Eq. (7.22) can be evaluated by means of some numerical formula, for example, Simpson's one-third rule (see p. 381 of Ref. [11]) given by

$$\int_{x_1}^{x_{m+1}} f(x)\,dx = \frac{\Delta x}{3}\left(y_1 + 4\sum_{i \text{ even}}^{m} y_i + 2\sum_{i \text{ odd}}^{m-1} y_i + y_{m+1}\right) \tag{7.23}$$

where m is assumed to be an even integer.

The graphical construction for the convolution integral of Eq. (7.22) for $\tau = 15.0$ s and $\omega_c = 1$ rad/s is illustrated in Fig. 7.15. The spectrum of $x_w(t)$, that is, $X_m(j\omega)$ is illustrated in Fig. 7.15e.

(c) The difference between $X_w(\omega)$ and $X(\omega)$, which is plotted in Fig. 7.15f, is seen to oscillate about zero and becomes significantly larger at frequencies close to the band edges $\pm\omega_c$. The oscillations in $X_w(\omega)$ are caused by the side lobes in the spectrum of the window, as was anticipated earlier. The larger difference near the band edges is due to the finite main-lobe width and it is caused by the fact that convolution tends to smooth the value of the function in the region of a discontinuity. If the ripple ratio and main-lobe width of the window function were decreased, the spectrum of the finite-duration signal $x_w(t)$ would tend to approach that of the infinite-duration signal $x(t)$. ■

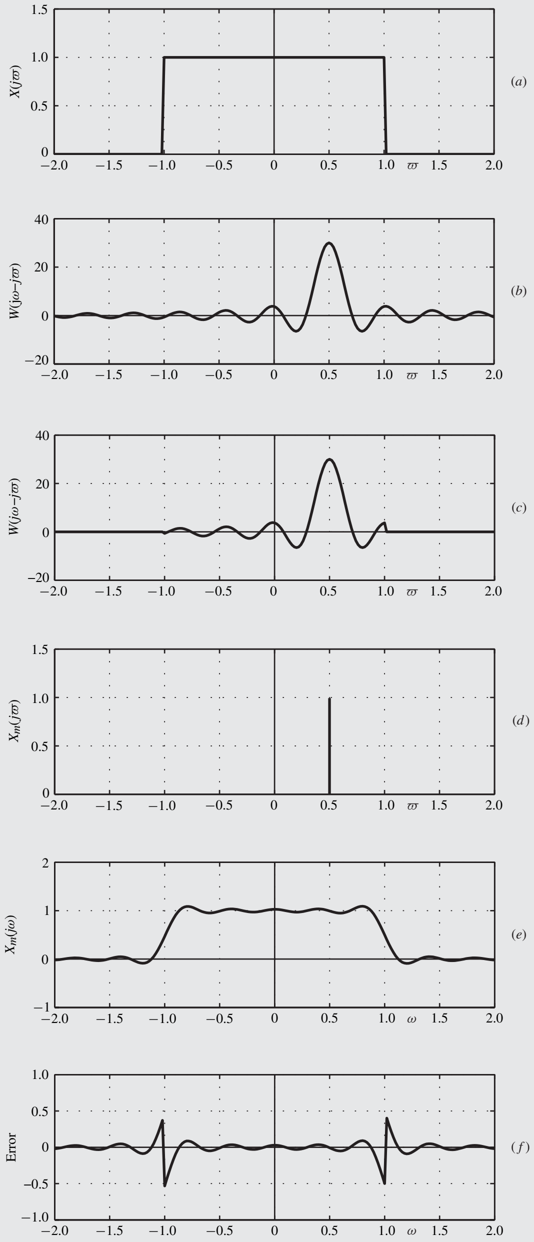


Figure 7.15 Use of rectangular window (Example 7.2): (*a*) Spectrum of original signal, (*b*) shifted spectrum of window, (*c*) truncated spectrum of window, (*d*) area under the curve in plot (*c*), (*e*) modified spectrum of signal, (*f*) difference between the spectrums of the original and modified signals.

KAISER WINDOW. For a fixed value of τ, the rectangular window has a fairly narrow main lobe relative to that in other windows, which is an advantage because it tends to introduce narrow transition bands at signal discontinuities. Its main disadvantage is that it has a high ripple ratio which causes a large in-band error (see Fig. 7.15*e*). Since the main-lobe width of the rectangular window is given by $B_w = 2\pi/\tau$, it can be controlled by choosing parameter τ but as was found out, the ripple ratio of the rectangular window is practically independent of τ and cannot be controlled (see Fig. 7.10). The alternative is to use more sophisticated windows, such as the Kaiser and Dolph-Chebyshev windows [6, 8] which offer two degrees of freedom, namely, τ and some other parameter. An even more advanced window is the so-called *ultraspherical window* [9, 10] which offers three degrees of freedom, namely, τ and two other parameters. In these parametric windows the available degrees of freedom can be used to adjust the main-lobe width and/or the ripple ratio. An important feature of known windows is that the main-lobe width is always inversely proportional to the window length τ.

The Kaiser window is a fairly flexible and easy to compute window and it is, therefore, used widely. It is given by

$$w_K(t) = \begin{cases} \dfrac{I_0(\beta)}{I_0(\alpha)} & \text{for } -\tau \le t \le \tau \\ 0 & \text{otherwise} \end{cases} \tag{7.24}$$

where α is an independent parameter and

$$\beta = \alpha\sqrt{1 - \left(\frac{t}{\tau}\right)^2} \tag{7.25}$$

Function $I_0(x)$ is the *zeroth-order modified Bessel function of the first kind.*[3] It can be computed to any degree of precision by using the rapidly converging series

$$I_0(x) = 1 + \sum_{k=1}^{\infty} \left[\frac{1}{k!}\left(\frac{x}{2}\right)^k\right]^2 \tag{7.26}$$

The frequency spectrum of the Kaiser window is given by [6]

$$W_K(j\omega) = \frac{2}{I_0(\alpha)} \frac{\sin\left(\tau\sqrt{\omega^2 - \omega_a^2}\right)}{\sqrt{\omega^2 - \omega_a^2}} \tag{7.27}$$

where $\omega_a = \alpha/\tau$. The Kaiser window and its frequency spectrum are plotted in Fig. 7.16*a* and *b* for $\tau = 15.0$ s.

Using Eq. (7.27), the ripple ratio and main-lobe width of the Kaiser window can be plotted as functions of α as depicted in Fig. 7.17*a* and *b*. The ripple ratio decreases rapidly as α is increased but it is practically independent of the window length. On the other hand, the main-lobe width increases

[3] *Function besseli(0,x) in MATLAB.*

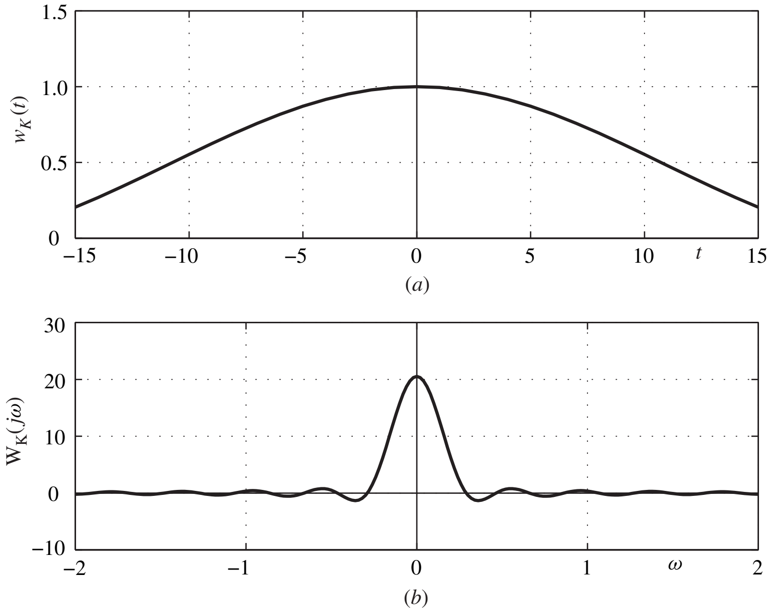


Figure 7.16 Kaiser window ($\alpha = 3.0$, $\tau = 15.0$ s): (*a*) Time domain, (*b*) frequency domain.

with increasing values of α but, as in other windows, it is inversely proportional to the window length.

Example 7.3 (*a*) Solve the problem of Example 7.2 using the Kaiser window with $\alpha = 3.0$ and $\tau = 15.0$ s and compare the results with those obtained with the rectangular window in Example 7.2. (*b*) Repeat for $\tau = 63.0$ s.

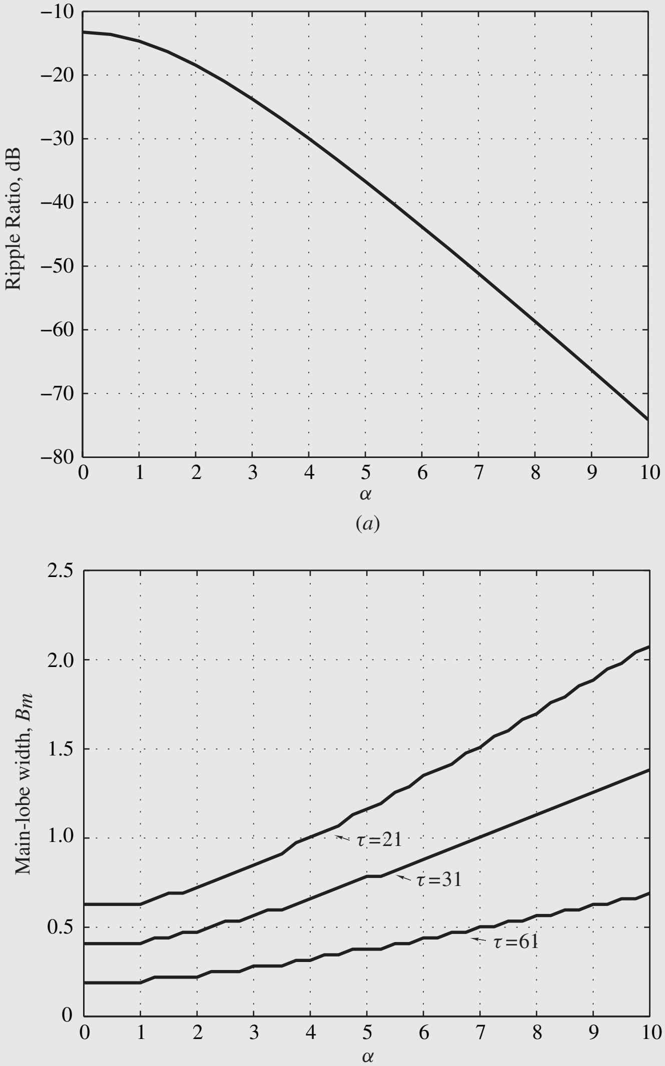
■ **Solution**

(*a*) For this example,

$$X_w(j\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\varpi)W_K(j\omega - j\varpi)\,d\varpi$$

where $X(j\omega)$ and $W_K(j\omega)$ are given by Eqs. (7.21) and (7.27), respectively. Evaluating the above integral with $\alpha = 3.0$ and $\tau = 15.0$ s using the formula in Eq. (7.23), as in Example 7.2, the frequency spectrum of $x_w(t)$ shown in Fig. 7.18*a* can be obtained. The difference between the spectrums of $x(t)$ and $x_w(t)$ is plotted in Fig. 7.18*b*. ■

(*b*) The spectrum of $x_w(t)$ for $\alpha = 3.0$ and $\tau = 63.0$ s is obtained, as shown in Fig. 7.18*c*, as in part (*a*). The difference between the spectrums of $x(t)$ and $x_w(t)$ is plotted in Fig. 7.18*d*.



(*a*)

(*b*)

Figure 7.17 Characteristics of Kaiser window : (*a*) Ripple ratio in dB versus α for $\tau = 21$ s, (*b*) main-lobe width versus α for different window lengths.

A comparison of Fig. 7.15*e* and 7.18*a* shows that the rectangular window offers narrower transition bands whereas the Kaiser window offers a reduced ripple amplitude. By increasing τ in the Kaiser window, which amounts to reducing the main-lobe width, the width of the transition bands can be reduced, as can be seen by comparing Fig. 7.18*a* and *c*, while maintaining a reduced ripple amplitude. ■

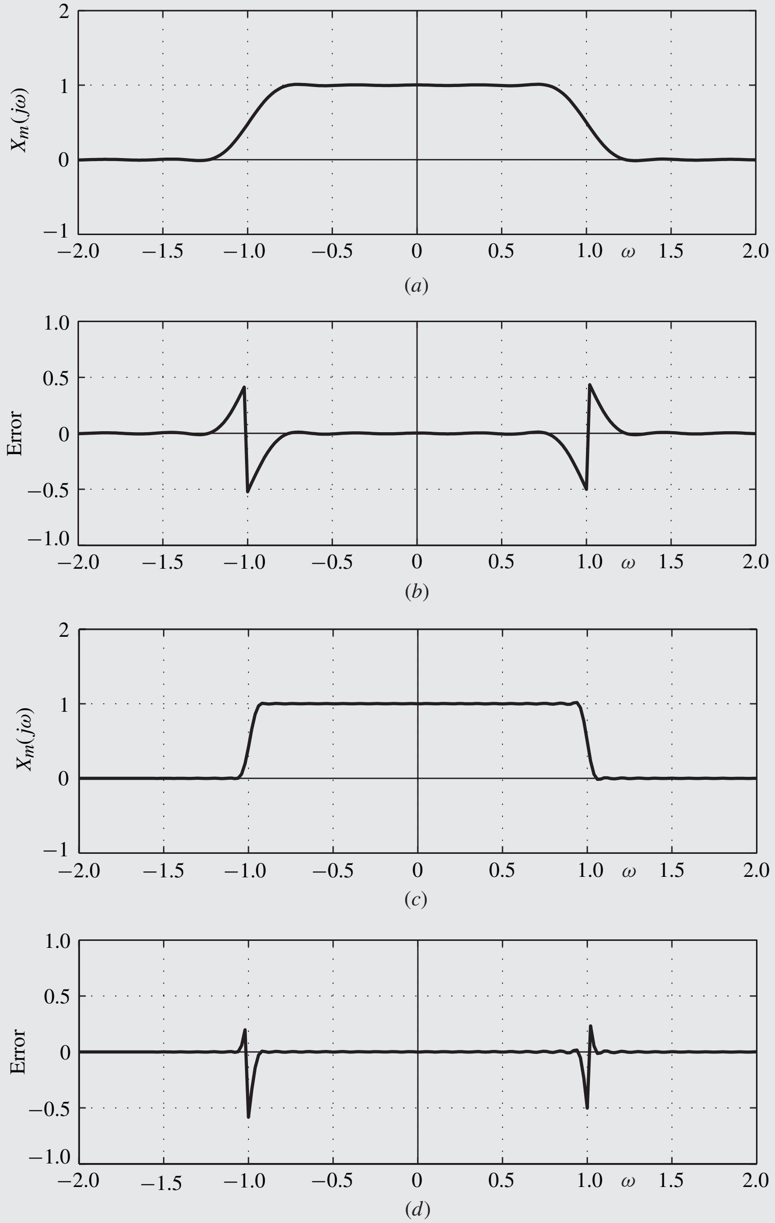


Figure 7.18 Use of Kaiser window (Example 7.3): (*a*) Spectrum of modified signal for $\alpha = 3.0$ and $\tau = 15.0$ s, (*b*) difference between the spectrums of the original and modified signals, (*c*) spectrum of modified signal for $\alpha = 3.0$ and $\tau = 63.0$ s, (*d*) difference between the spectrums of the original and modified signals.

7.8.2 Discrete-Time Windows

Given a continuous-time window, two corresponding discrete-time windows can be derived, namely, odd- and even-length windows. These are obtained by sampling the continuous-time window function at $nT = 0, \pm T, \pm 2T, \ldots, \pm(N_w - 1)T/2$ for an odd-length window or at $nT = \pm\frac{1}{2}T, \pm\frac{3}{2}T, \ldots, \pm(N_w - 1)T/2$ for an even-length window. Typically an odd window length is preferred although even-length windows are sometimes required for certain specialized applications (see, for example, Sec. 18.4.1).

A rectangular discrete-time window of odd length can be readily obtained by letting $t = nT$ and $\tau = [(N_w - 1)T]/2$ in Eq. (7.18), that is,

$$w_R(nT) = \begin{cases} 1 & \text{for } -[(N_w - 1)T]/2 \le nT \le [(N_w - 1)T]/2 \\ 0 & \text{otherwise} \end{cases}$$

Its z transform is given by

$$W_R(z) = \sum_{n=-(N_w-1)/2}^{(N_w-1)/2} z^{-n}$$

and since the above sum is a geometric series, a closed-form expression can be obtained as

$$\begin{aligned} W_R(z) &= \frac{z^{(N_w-1)/2} - z^{-(N_w+1)/2}}{1 - z^{-1}} \\ &= \frac{z^{N_w/2} - z^{-N_w/2}}{z^{1/2} - z^{-1/2}} \end{aligned}$$

By evaluating the z transform on the unit circle, the frequency spectrum of the rectangular window can be obtained as

$$W_R(e^{j\omega T}) = \frac{e^{j\omega N_w T/2} - e^{-j\omega N_w T/2}}{e^{j\omega T/2} - e^{-j\omega T/2}} = \frac{\sin(\omega N_w T/2)}{\sin(\omega T/2)}$$

Similarly, a Kaiser discrete-time window of odd length can be derived from Eqs. (7.24) and (7.25) as

$$w_K(nT) = \begin{cases} \dfrac{I_0(\beta)}{I_0(\alpha)} & \text{for } -(N_w - 1)T/2 \le nT \le (N_w - 1)T/2 \\ 0 & \text{otherwise} \end{cases} \tag{7.28}$$

where α is an independent parameter and

$$\beta = \alpha\sqrt{1 - \left(\frac{2n}{N_w - 1}\right)^2} \tag{7.29}$$

(see Prob. 7.13, part (*a*)). The z transform of $w_K(nT)$ is given by

$$W_K(z) = \sum_{n=-(N_w-1)/2}^{(N_w-1)/2} w_K(nT)z^{-n}$$

and hence the frequency spectrum of the Kaiser window can be obtained as

$$W_K(e^{j\omega T}) = \sum_{n=-(N_w-1)/2}^{(N_w-1)/2} w_K(nT)e^{-j\omega T}$$

An approximate but closed-form expression for the spectrum of the discrete-time Kaiser window can be deduced by noting that the spectral energy of the continuous-time Kaiser window is concentrated at low frequencies (see Fig. 7.16*b*) and, therefore, for a sufficiently large sampling frequency ω_s, we have

$$W_K(j\omega) \approx 0 \qquad \text{for } |\omega| \geq \frac{\omega_s}{2} \tag{7.30}$$

Consequently, the spectrum of impulse-modulated signal $\hat{w}_K(t)$ and, equivalently, the spectrum of $w_K(nT)$ can be expressed as

$$\hat{W}_K(j\omega) = W_K(e^{j\omega T}) \approx \frac{1}{T}W_K(j\omega) \qquad \text{for } 0 \leq |\omega| < \frac{\omega_s}{2} \tag{7.31}$$

according to Eq. (6.45a). Hence Eqs. (7.27), (7.30), and (7.31) give

$$W_K(e^{j\omega T}) \approx \frac{N_w - 1}{\alpha I_0(\alpha)} \cdot \frac{\sin\left[\alpha\sqrt{(\omega/\omega_a)^2 - 1}\right]}{\sqrt{(\omega/\omega_a)^2 - 1}} \tag{7.32}$$

where

$$\omega_a = \frac{2\alpha}{(N_w - 1)T}$$

(see Prob. 7.13, part (*b*)). The application of the window technique to discrete-time signals involves the same steps as in the continuous-time case although some subtle differences should be noted for its successful application. The modified discrete-time signal is obtained by multiplying the signal by a suitable discrete-time window, i.e.,

$$x_w(nT) = w(nT)x(nT)$$

The frequency spectrum of the modified signal $x_w(nT)$ can be obtained by finding the z transform of $x_w(nT)$ and then evaluating it on the unit circle. Since the z transform of a product of discrete-time functions can be obtained by using the complex-convolution theorem (Theorem 3.10), we can write

$$\mathcal{Z}x_w(nT) = X_w(z) = \frac{1}{2\pi j}\oint_\Gamma X(v)W\left(\frac{z}{v}\right)v^{-1}\,dv \tag{7.33a}$$

$$= \frac{1}{2\pi j}\oint_\Gamma X\left(\frac{z}{v}\right)W(v)v^{-1}\,dv \tag{7.33b}$$

where $X(z)$ is the z transform of $x(nT)$ and Γ represents a contour in the common region of convergence of $X(v)$ and $W(z/v)$. With

$$v = e^{j\varpi T} \qquad \text{and} \qquad z = e^{j\omega T}$$

and $X(v)$ as well as $W(z/v)$ convergent on the unit circle of the v plane, Eqs. (7.33a) and (7.33b) give the frequency spectrum of $x_w(t)$ as

$$X_w(e^{j\omega T}) = \frac{T}{2\pi}\int_0^{2\pi/T} X(e^{j\varpi T})W(e^{j(\omega-\varpi)T})\,d\varpi \tag{7.34a}$$

$$= \frac{T}{2\pi}\int_0^{2\pi/T} X(e^{j(\omega-\varpi)T})W(e^{j\varpi T})\,d\varpi \tag{7.34b}$$

These formulas are essentially convolution integrals of the type given by Eqs. (7.20a) and (7.20b) except that the integration is carried out over the finite frequency range $0 \le \varpi \le 2\pi/T$. Thus the effects of a discrete-time window on a discrete-time signal are entirely analogous to those encountered in the continuous-time case. The side lobes of the window function tend to introduce oscillations in the amplitude spectrum of the signal whose amplitude tends to decrease as the ripple ratio of the window is decreased. The main-lobe width tends to even out abrupt changes in the amplitude spectrum and, as a consequence, it tends to introduce transition bands at discontinuities. To minimize this effect, the main-lobe width should be as small as possible. As in the continuous-time case, the main-lobe width can be reduced by increasing the length of the window.

7.8.3 Periodic Discrete-Time Windows

The application of the window technique to periodic discrete-time signals involves one more wrinkle. Since the values of a periodic signal $\tilde{x}(nT)$ are normally defined at $0, T, \ldots, (N-1)T$ and N is usually a power of 2, to facilitate the application of the FFT method N must be even. As may be recalled from Sec. 7.8.2, the values of a window of odd-length N_w occur at $0, \pm T, \ldots, \pm(N_w-1)/2$ whereas those of an even-length window occur at $\pm T/2, \pm 3T/2, \ldots, \pm(N_w-1)/2$. Consequently, a window of odd length is easier to apply. Therefore, the window length N_w should not exceed $N-1$.

A periodic window $\tilde{w}(nT)$ with a period NT can be readily obtained from a nonperiodic window $w(nT)$ through periodic continuation whereby shifted copies of $w(nT)$, namely, $w(nT+rNT)$ for $\ldots, -2, -1, 0, 1, 2, \ldots$ are added up according to Eq. (7.3). Now the DFT is normally computed with respect to the period 0 to $(N-1)T$ whereas nonperiodic windows of odd length N_w are defined with respect to the period $-[(N_w-1)T]/2$ to $[(N_w-1)T]/2$. This discrepancy can be easily circumvented by constructing a right-sided discrete-time window before converting it into a periodic window through periodic continuation. For example, a right-side discrete-time Kaiser window can be readily obtained as

$$\vec{w}_K(nT) = w_K[nT - (N_w - 1)T/2] \tag{7.35a}$$

and a corresponding periodic version can be constructed as

$$\tilde{w}_K(nT) = \sum_{r=-\infty}^{\infty} \vec{w}_K(nT + rNT) \tag{7.35b}$$

as illustrated in Fig. 7.19*b*. An alternative but equivalent possibility would be to apply periodic continuation to the two-sided window as

$$\tilde{w}(nT) = \begin{cases} w_K(0) & \text{for } n = 0 \\ w_K(nT) & \text{for } 1 \le n \le (N_w - 1)/2 \\ 0 & \text{for } [(N_w - 1)/2 + 1] \le n \le [N - (N_w + 1)/2] \\ w_K[(n - N)T] & \text{for } [N - (N_w - 1)/2] \le n \le (N - 1) \end{cases} \tag{7.35c}$$

This possibility is illustrated in Fig. 7.19*c*.

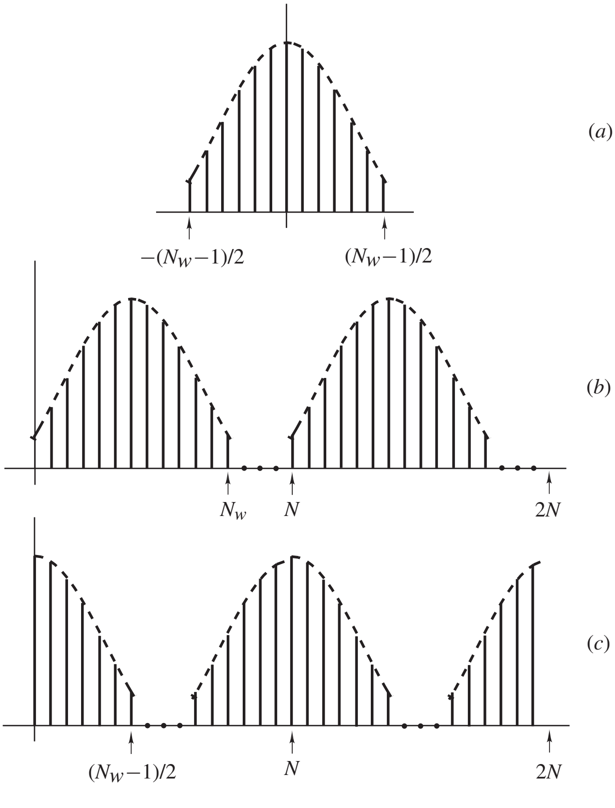


Figure 7.19 Generation of a periodic window function: (*a*) Nonperiodic window, (*b*) periodic window of Eq. (7.35b), (*c*) periodic window of Eq. (7.35c).

Example 7.4 (*a*) Obtain a periodic Kaiser window of length 13 for a 16-point DFT using Eq. (7.35b). (*b*) Repeat part (*a*) using Eq. (7.35c).

- **Solution**

From Eqs. (7.35b) and (7.35c), the values shown in columns 2 and 3 of Table 7.1 can be obtained. The numerical values of the Kaiser window can be calculated using Eqs. (7.28) and (7.29) where $I_0(x)$ is given by Eq. (7.26). ■

Table 7.1 Values of periodic Kaiser window of length 13 (Example 7.4)

$\tilde{w}_K(nT)$	Eq. (7.35b)	Eq. (7.35c)
$\tilde{w}_K(0)$	$w_K(-6T)$	$w_K(0)$
$\tilde{w}_K(T)$	$w_K(-5T)$	$w_K(T)$
$\tilde{w}_K(2T)$	$w_K(-4T)$	$w_K(2T)$
$\tilde{w}_K(3T)$	$w_K(-3T)$	$w_K(3T)$
$\tilde{w}_K(4T)$	$w_K(-2T)$	$w_K(4T)$
$\tilde{w}_K(5T)$	$w_K(-T)$	$w_K(5T)$
$\tilde{w}_K(6T)$	$w_K(0)$	$w_K(6T)$
$\tilde{w}_K(7T)$	$w_K(T)$	0
$\tilde{w}_K(8T)$	$w_K(2T)$	0
$\tilde{w}_K(9T)$	$w_K(3T)$	0
$\tilde{w}_K(10T)$	$w_K(4T)$	$w_K(-6T)$
$\tilde{w}_K(11T)$	$w_K(5T)$	$w_K(-5T)$
$\tilde{w}_K(12T)$	$w_K(6T)$	$w_K(-4T)$
$\tilde{w}_K(13T)$	0	$w_K(-3T)$
$\tilde{w}_K(14T)$	0	$w_K(-2T)$
$\tilde{w}_K(15T)$	0	$w_K(-T)$

7.8.4 Application of Window Technique

The window technique can be applied to continuous- and discrete-time nonperiodic or periodic signals. If the signal to be processed is a nonperiodic continuous-time signal, then it must be subjected to three processes before it can be processed, namely, sampling, windowing, and periodic continuation. Depending on the order of these operations, three options are available to us, namely, to apply windowing to the continuous-time signal, to the discrete-time signal, or to the periodic discrete-time signal. The three possibilities are, of course, equivalent but the first is the easiest to explain.

A given continuous-time signal $x(t)$ of arbitrary length with a frequency spectrum $X(j\omega)$ can be forced to assume zero values outside the range $0 \leq t < N_w T$ through windowing such that

$$x_w(t) = 0 \qquad \text{for } t < 0 \qquad \text{and} \qquad t \geq N_w T \tag{7.36}$$

as illustrated in Fig. 7.20*c*. By choosing a window with sufficiently small main-lobe width and ripple ratio, the spectrum of the modified signal can be made to approach the spectrum of the original signal so that $X_w(j\omega) \approx X(j\omega)$ as shown in Fig. 7.20*c*.

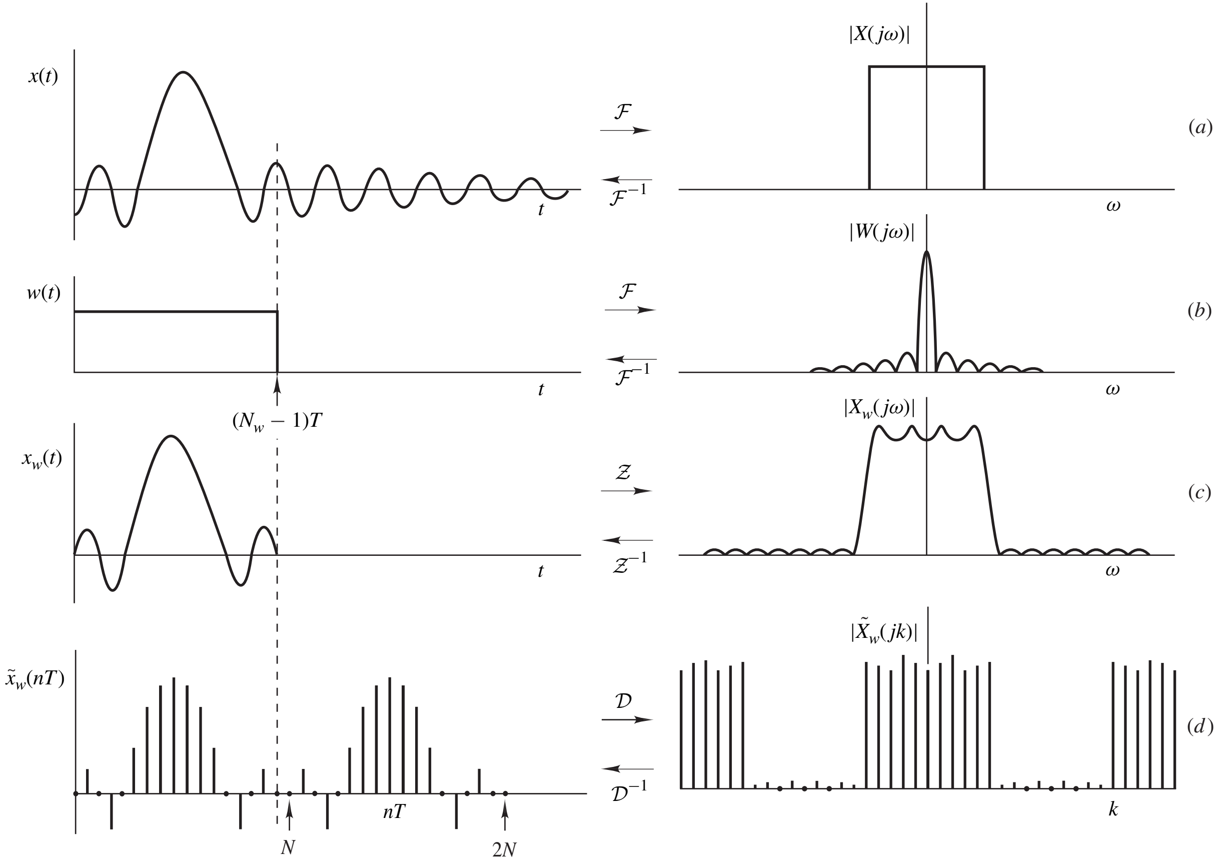


Figure 7.20 The use of window functions.

If the frequency spectrum of $x_w(t)$, is bandlimited and a sufficiently high sampling frequency ω_s is used, we have

$$X_w(j\Omega) \approx 0 \qquad \text{for } |\Omega| > \frac{\omega_s}{2} \tag{7.37}$$

in which case Eq. (7.12b) would assume the form

$$\mathcal{D} \sum_{r=-\infty}^{\infty} x_w(nT + rNT) \approx \frac{1}{T} \sum_{r=-\infty}^{\infty} X_w(jk\Omega + jr\omega_s)$$

as illustrated in Fig. 7.20*d*. Unfortunately, the frequency spectrum of the signal is usually not known *a priori* and, therefore, the sampling frequency that would satisfy Eq. (7.37) is also not known. One way of estimating the correct sampling frequency would be to compute the DFT of $x_w(nT)$ for progressively larger sampling frequencies (smaller sampling periods) until two successive DFT evaluations yield approximately the same DFT values. When that happens, the reduction in aliasing brought about by the last sampling frequency can be deemed to be negligible and the last but one sampling frequency can be taken to be the lowest acceptable sampling frequency for the application at hand. One possibility would be to start with some low sampling frequency, say ω_s, and then carry out DFT evaluations for sampling frequencies $2\omega_s, 4\omega_s, \ldots$. Since the sampling frequency for each new DFT evaluation is twice the previous value, the number of points N must also be doubled to maintain a constant resolution for the DFT, i.e., $\Omega = \omega_s/N$. This technique is illustrated by the following example.

Example 7.5 An infinite-duration right-sided discrete-time signal $x(nT)$ is obtained by sampling the continuous-time signal

$$x(t) = u(t)\left[A_0 e^{p_0 t} + 2M_1 e^{\sigma_1 t} \cos(\omega_1 t + \theta_1)\right]$$

where $A_0 = 4.532$, $M_1 = 2.350$, $\theta_1 = -2.873$ rad, $p_0 = -2.322$, $\sigma_1 = -1.839$, and $\omega_1 = 1.754$ rad/s. A finite duration signal can be obtained by applying the discrete-time Kaiser window with $\alpha = 1.0$. Using the above technique, find the lowest sampling frequency (largest sampling period) that would result in negligible aliasing.

■ **Solution**

A problem of this type can be solved by obtaining the DFT of the signal for several increasing sampling frequencies until two successive DFTs differ from one another by an error that is considered acceptable for the application at hand. The last DFT can be deemed to be free of aliasing and the sampling frequency used in the last evaluation can be taken to be the required one.

Table 7.2 Values of N, T, and ω_s for Example 7.5

N	T, s	ω_s, rad/s
16	1.6	3.9264
32	0.8	7.8528
64	0.4	15.7056
128	0.2	31.4112

In order to be able to compare successive DFTs of the same signal, it is important to keep the discrete-frequency parameter Ω fixed. From the definition of the DFT in Sec. 7.2, we have

$$\Omega = \frac{\omega_s}{N} = \frac{2\pi}{NT}$$

and by selecting the values of N and T shown in Table 7.2, a constant value of Ω equal to 0.2454 rad/s can be achieved. The corresponding values of ω_s for $N = 16,\ 32,\ 64,$ and 128 are given by the product $N\Omega$ and are listed in column 3 of Table 7.2.

A periodic Kaiser window function can be constructed as

$$\tilde{w}(nT) = \sum_{r=-\infty}^{\infty} w(nT + NT)$$

where

$$w(nT) = w_K(nT) \qquad \text{for} \quad -(N-2)/2 \leq n < (N-2)/2$$

is a nonperiodic window function of odd length. Assuming a window length $N_w = N-1$, Eq. (7.35b) gives

$$\tilde{w}(nT) = \begin{cases} w_K(nT) & \text{for } 0 \leq n \leq (N-2)/2 \\ 0 & \text{for } n = N/2 \\ w_K[(n-N)] & \text{for } (N+2)/2 \leq n \leq (N-1) \end{cases}$$

The values of $w_K(nT)$ can be determined by using Eqs. (7.28) and (7.29) where $I_0(x)$ is given by Eq. (7.26).

On obtaining the DFT of $\tilde{x}_w(nT)$ for the four sets of parameters shown in Table 7.2, the results plotted in Fig. 7.21 can be obtained. As can be seen, the difference between the DFTs for the last and last-but-one cases is barely noticeable and we conclude, therefore, that the use of the sampling frequency 15.7056 rad/s would entail negligible aliasing. ■

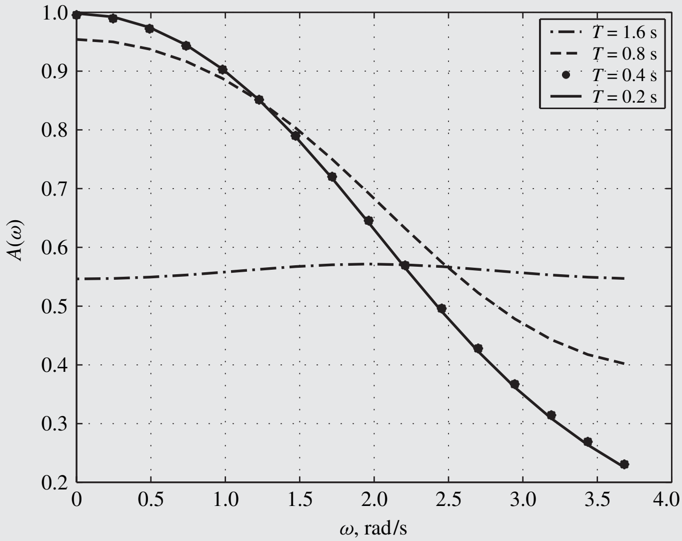


Figure 7.21 Amplitude spectrum of $\tilde{x}(t)$ for different sampling frequencies (Example 7.5).

Discrete-time windows are also used in the design of FIR digital filters and a good collection of such functions, including the Dolph-Chebyshev window, can be found in Sec. 9.4.

7.9 SIMPLIFIED NOTATION

The preceding somewhat complicated notation for the DFT was adopted in order to eliminate possible confusion between the various transforms. As we shall be dealing exclusively with the DFT for the rest of this chapter, we can write

$$X(k) = \sum_{n=0}^{N-1} x(n)W^{-kn} \qquad x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)W^{kn}$$

where

$$x(n) \equiv \tilde{x}(nT) \qquad X(k) \equiv \tilde{X}(jk\Omega)$$

i.e., $x(n)$ and $X(k)$ are implicitly assumed to be periodic. Furthermore, signals and their frequency spectrums are deemed to be normalized with respect to time and frequency whereby the discrete time and frequency variables T and Ω are assumed to be 1 s and $2\pi/N$ rad/s, respectively (see Sec. 1.4).

7.10 PERIODIC CONVOLUTIONS

The convolutions of the CFT and the z transform find numerous applications in signal processing. Analogous and equally useful convolutions exist for the DFT. These are usually referred to as *periodic* or *cyclic* convolutions.

7.10.1 Time-Domain Periodic Convolution

The *time-domain convolution* of two periodic signals $x(n)$ and $h(n)$, each with period N, is defined as

$$y(n) = \sum_{m=0}^{N-1} x(m)h(n-m) \tag{7.38a}$$

$$= \sum_{m=0}^{N-1} x(n-m)h(m) \tag{7.38b}$$

Like $x(n)$ or $h(n)$, $y(n)$ is a periodic function of n with period N.

The periodic time-domain convolution can be illustrated by a graphical construction which is very similar to that used for the convolution summation described in Sec. 4.6.1; the main difference being the periodicity of the signals involved in the present convolution. The time-domain convolution is illustrated by the following example.

Example 7.6 Each of the signals

$$x(n) = e^{-\alpha n} \qquad \text{for } 0 \le n \le 9$$

and

$$h(n) = \begin{cases} 1 & \text{for } 3 \le n \le 6 \\ 0 & \text{for } 0 \le n \le 2 \\ 0 & \text{for } 7 \le n \le 9 \end{cases}$$

is periodic with period $N = 10$. Find the periodic convolution of the two signals at $n = 5$.

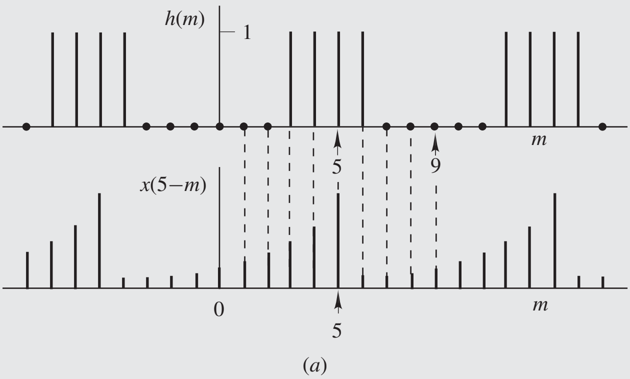
■ **Solution**

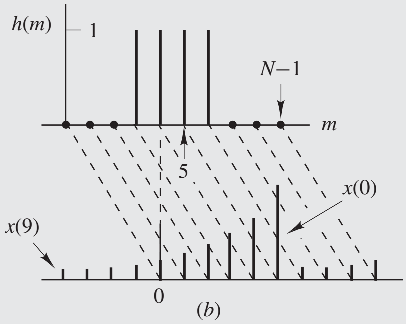
A graphical construction for the time-domain convolution can be obtained as follows:

1. Replace variable n by m in both $h(n)$ and $x(n)$ to obtain $x(m)$ and $h(m)$.
2. Turn signal $x(m)$ over end-to-end to obtain $x(-m)$.
3. Shift $x(-m)$ to the right by 5 s as shown in Fig. 7.22*a* which can be drawn as shown in Fig. 7.22*b*.
4. Noting the periodicity of $x(m)$, construct the equivalent graph shown in Fig. 7.22*c*.
5. Form all the possible products $h(m)x(5-m)$ in Fig. 7.22*c*.
6. Add all the nonzero products in Fig. 7.22*c* to obtain the time-convolution.

From Fig. 7.22*c*, we have

$$y(5) = 1 + e^{-\alpha} + e^{-2\alpha} + e^{-9\alpha}$$





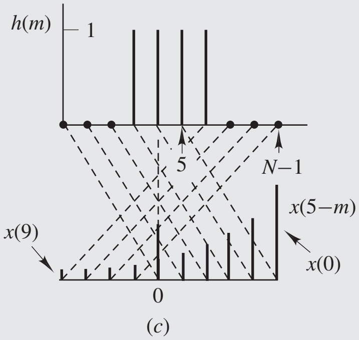


Figure 7.22 Time-domain, periodic convolution (Example 7.6).

Alternatively, from the time-domain convolution of Eq. (7.38b), we have

$$y(k) = x(k)h(0) + x(k-1)h(1) + \cdots + x(k)h(0) + x(k+1)h(-1)$$
$$+ \cdots + x(k - N + 1)h(N - 1)$$

Hence

$$\begin{aligned}
y(5) &= x(5)h(0) + x(4)h(1) + x(3)h(2) + x(2)h(3) + x(1)h(4)\\
&\quad +h(5)x(0) + h(6)x(-1) + h(7)x(-2) + h(8)x(-3)\\
&\quad +h(9)x(-4)\\
&= x(5)h(0) + x(4)h(1) + x(3)h(2) + x(2)h(3) + x(1)h(4)\\
&\quad +x(0)h(5) + x(9)h(6) + x(8)h(7) + x(7)h(8) + x(6)h(9)\\
&= x(2)h(3) + x(1)h(4) + x(0)h(5) + x(9)h(6)\\
&= x(0) + x(1) + x(2) + x(9)\\
&= 1 + e^{-\alpha} + e^{-2\alpha} + e^{-9\alpha} \quad \blacksquare
\end{aligned}$$

Theorem 7.2*A* Time-Domain Periodic Convolution ***If h(n) and x(n) are two periodic discrete-time functions, each with period N, with DFTs H(k) and X(k), respectively, then the DFT of the time-domain convolution of h(n) and x(n) is given by***

$$\boldsymbol{Y(k) = H(k)X(k)}$$

Proof The DFT of $y(n)$ can be obtained from the definition in Sec. 7.2 as

$$\begin{aligned}
Y(k) &= \sum_{n=0}^{N-1}\left[\sum_{m=0}^{N-1} h(m)x(n-m)\right]W^{-kn}\\
&= \sum_{m=0}^{N-1} h(m)W^{-km}\sum_{n=0}^{N-1} x(n-m)W^{-k(n-m)}\\
&= \sum_{n=0}^{N-1} h(n)W^{-kn}\sum_{n=0}^{N-1} x(n)W^{-kn}
\end{aligned}$$

and, therefore,

$$Y(k) = H(k)X(k) \quad \blacktriangle \tag{7.39}$$

7.10.2 Frequency-Domain Periodic Convolution

The *frequency-domain convolution* of two DFTs $H(k)$ and $X(k)$ is defined as

$$Y(k) = \frac{1}{N}\sum_{m=0}^{N-1} X(m)H(k-m) \tag{7.40a}$$

$$= \frac{1}{N}\sum_{m=0}^{N-1} X(k-m)H(m) \tag{7.40b}$$

Theorem 7.2*B* Frequency-Domain Periodic Convolution ***The IDFT of the frequency-domain periodic convolution of H(k) and X(k) is given by***

$$\boldsymbol{y(n) = h(n)x(n)} \quad \blacktriangle \tag{7.41}$$

The proof is left as an exercise (see Prob. 7.23).

7.11 FAST FOURIER-TRANSFORM ALGORITHMS

The direct evaluation of the DFT involves N complex multiplications and $N-1$ complex additions for each value of $X(k)$, and since there are N values to determine, N^2 multiplications and $N(N-1)$ additions are necessary. Consequently, for large values of N, say in excess of 1000, direct evaluation involves a considerable amount of computation.

It turns out that the direct evaluation of the DFT on the basis of its definition entails a large amount of redundancy and through some clever strategies, a huge reduction in the amount of computation can be achieved. These strategies have come to be known collectively as *fast Fourier transforms* (FFTs) although, to be precise, these methods have nothing to do with transforms in the true sense of the word.

We describe here two FFTs, the so-called *decimation-in-time* and *decimation-in-frequency algorithms*.

7.11.1 Decimation-in-Time Algorithm

Let the desired DFT be

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{-kn} \qquad \text{where } W_N = e^{j2\pi/N}$$

and assume that

$$N = 2^r$$

where r is an integer. The above summation can be split into two parts as

$$X(k) = \sum_{\substack{n=0 \\ n \text{ even}}}^{N-1} x(n)W_N^{-kn} + \sum_{\substack{n=0 \\ n \text{ odd}}}^{N-1} x(n)W_N^{-kn}$$

Alternatively

$$X(k) = \sum_{n=0}^{N/2-1} x_{10}(n)W_N^{-2kn} + W_N^{-k} \sum_{n=0}^{N/2-1} x_{11}(n)W_N^{-2kn} \tag{7.42}$$

where

$$x_{10}(n) = x(2n) \tag{7.43a}$$

$$x_{11}(n) = x(2n+1) \tag{7.43b}$$

for $0 \le n \le N/2 - 1$. Since

$$W_N^{-2kn} = e^{-j4kn\pi/N} = e^{-j2kn\pi/(N/2)} = W_{N/2}^{-kn}$$

Eq. (7.42) can be expressed as

$$X(k) = \sum_{n=0}^{N/2-1} x_{10}(n) W_{N/2}^{-kn} + W_N^{-k} \sum_{n=0}^{N/2-1} x_{11}(n) W_{N/2}^{-kn}$$

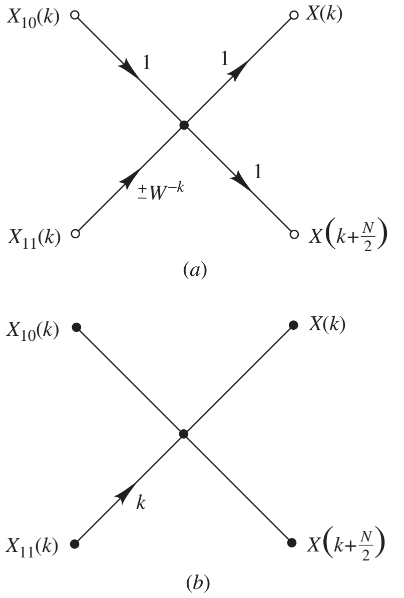
Clearly

$$X(k) = X_{10}(k) + W_N^{-k} X_{11}(k) \tag{7.44}$$

and since $X_{10}(k)$ and $X_{11}(k)$ are periodic, each with period $N/2$, we have

$$\begin{aligned} X\left(k + \frac{N}{2}\right) &= X_{10}\left(k + \frac{N}{2}\right) + W_N^{-(k+N/2)} X_{11}\left(k + \frac{N}{2}\right) \\ &= X_{10}(k) - W_N^{-k} X_{11}(k) \end{aligned} \tag{7.45}$$

Equations (7.44) and (7.45) can be represented by the signal flow graph of Fig. 7.23*a*, where the minus sign in $\pm W_N^{-k}$ is pertinent in the computation of $X(k + N/2)$. Since the graph looks very much like a butterfly, it has come to be known as a *butterfly* and it is often represented by the simplified diagram of Fig. 7.23*b* for convenience.



(*a*)

(*b*)

Figure 7.23 (*a*) Butterfly flow graph, (*b*) simplified diagram.

What we have accomplished so far is to express the desired N-element DFT as a function of two $(N/2)$-element DFTs. Assuming that the values of $X_{10}(k)$ and $X_{11}(k)$ are available in corresponding arrays, the values of $X(k)$ can be readily computed as depicted in Fig. 7.24*a*.

$X_{10}(k)$ and $X_{11}(k)$ can now be expressed in terms of $(N/4)$-element DFTs by repeating the above cycle of activities. For $X_{10}(k)$, we can write

$$\begin{aligned}
X_{10}(k) &= \sum_{n=0}^{N/2-1} x_{10}(n)W_{N/2}^{-kn} \\
&= \sum_{n=0}^{N/4-1} x_{10}(2n)W_{N/2}^{-2kn} + \sum_{n=0}^{N/4-1} x_{10}(2n+1)W_{N/2}^{-k(2n+1)} \\
&= \sum_{n=0}^{N/4-1} x_{20}(n)W_{N/4}^{-kn} + W_N^{-2k} \sum_{n=0}^{N/4-1} x_{21}(n)W_{N/4}^{-kn} \qquad (7.46\text{a})
\end{aligned}$$

and, similarly, for $X_{11}(k)$

$$\begin{aligned}
X_{11}(k) &= \sum_{n=0}^{N/2-1} x_{11}(n)W_{N/2}^{-kn} \\
&= \sum_{n=0}^{N/4-1} x_{22}(n)W_{N/4}^{-kn} + W_N^{-2k} \sum_{n=0}^{N/4-1} x_{23}(n)W_{N/4}^{-kn} \qquad (7.46\text{b})
\end{aligned}$$

where

$$x_{20}(n) = x_{10}(2n) \qquad (7.47\text{a})$$

$$x_{21}(n) = x_{10}(2n+1) \qquad (7.47\text{b})$$

$$x_{22}(n) = x_{11}(2n) \qquad (7.47\text{c})$$

$$x_{23}(n) = x_{11}(2n+1) \qquad (7.47\text{d})$$

for $0 \le n \le N/4 - 1$. Consequently, from Eqs. (7.46a) and (7.46b), we have

$$\begin{aligned}
X_{10}(k) &= X_{20}(k) + W_N^{-2k} X_{21}(k) \\
X_{10}\left(k + \frac{N}{4}\right) &= X_{20}(k) - W_N^{-2k} X_{21}(k) \\
X_{11}(k) &= X_{22}(k) + W_N^{-2k} X_{23}(k) \\
X_{11}\left(k + \frac{N}{4}\right) &= X_{22}(k) - W_N^{-2k} X_{23}(k)
\end{aligned}$$

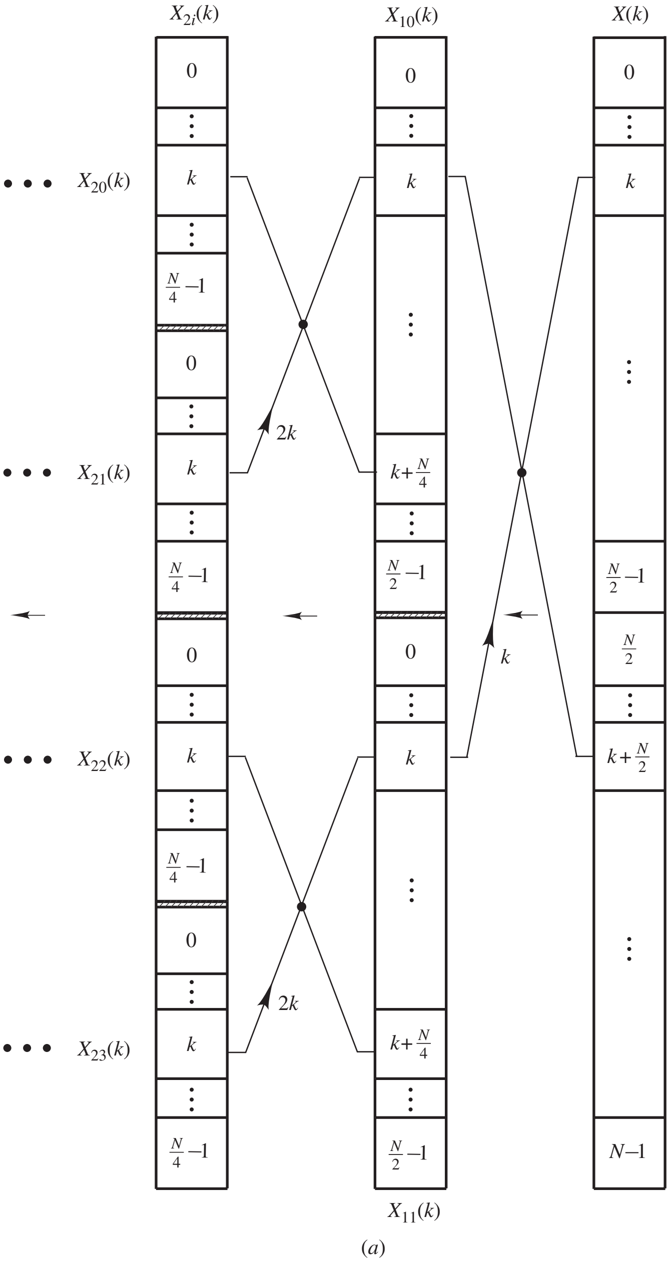


Figure 7.24 Decimation-in-time FFT algorithm: (*a*) First and second cycles.

Thus if the values of $X_{20}(k)$, $X_{21}(k)$, $X_{22}(k)$, and $X_{23}(k)$ are available, those of $X_{10}(k)$ and $X_{11}(k)$ and in turn those of $X(k)$ can be computed, as illustrated in Fig. 7.24*a*.

In exactly the same way, the *mth cycle* of the above procedure would yield

$$\begin{aligned}
X_{(m-1)0}(k) &= X_{m0}(k) + W_N^{-2^{m-1}k} X_{m1}(k) \\
X_{(m-1)0}\left(k + \frac{N}{2^m}\right) &= X_{m0}(k) - W_N^{-2^{m-1}k} X_{m1}(k) \\
X_{(m-1)1}(k) &= X_{m2}(k) + W_N^{-2^{m-1}k} X_{m3}(k) \\
X_{(m-1)1}\left(k + \frac{N}{2^m}\right) &= X_{m2}(k) - W_N^{-2^{m-1}k} X_{m3}(k) \\
&\cdots\cdots\cdots\cdots\cdots\cdots
\end{aligned}$$

where

$$x_{m0}(n) = x_{(m-1)0}(2n) \tag{7.48a}$$

$$x_{m1}(n) = x_{(m-1)0}(2n+1) \tag{7.48b}$$

$$x_{m2}(n) = x_{(m-1)1}(2n) \tag{7.48c}$$

$$x_{m3}(n) = x_{(m-1)1}(2n+1) \tag{7.48d}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

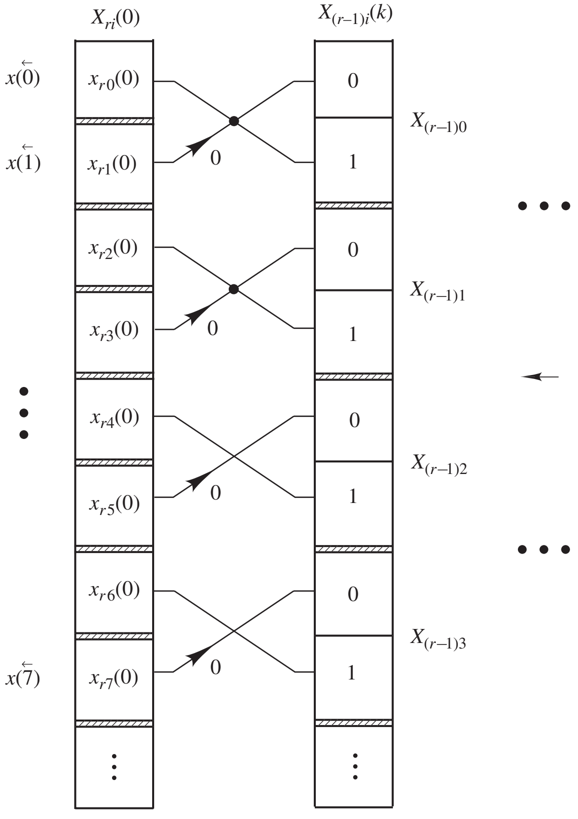
for $0 \le n \le N/2^m - 1$. Clearly, the procedure would terminate with the rth cycle ($N = 2^r$) since $x_{r0}(n)$, $x_{r1}(n)$, ... reduce the one-element sequences, in which case

$$X_{ri}(0) = x_{ri}(0) \qquad \text{for } i = 0, 1, \ldots, N-1$$

The values of the penultimate DFTs can be obtained from the above equations as

$$\begin{aligned}
X_{(r-1)0}(0) &= x_{r0}(0) + W_N^0 x_{r1}(0) \\
X_{(r-1)0}(1) &= x_{r0}(0) - W_N^0 x_{r1}(0) \\
X_{(r-1)1}(0) &= x_{r2}(0) + W_N^0 x_{r3}(0) \\
X_{(r-1)1}(1) &= x_{r2}(0) - W_N^0 x_{r3}(0) \\
&\cdots\cdots\cdots\cdots\cdots\cdots
\end{aligned}$$

Assuming that the sequence $\{x_{r0}(0),\ x_{r1}(0),\ \ldots\}$ is available in an array, the values of $X_{(r-1)i}(k)$ for $i = 0, 1, \ldots$ can be computed as in Fig. 7.24*b*. Then the values of $X_{(r-2)i}(k)$, $X_{(r-3)i}(k)$, ... can be computed in sequence, and ultimately the values of $X(k)$ can be obtained.



(*b*)

Figure 7.24 Cont'd Decimation-in-time FFT algorithm: (*b*) *r*th cycle.

The only remaining task at this point is to identify elements $x_{r0}(0), x_{r1}(0), \ldots$. Fortunately, this turns out to be easy. It can be shown that $x_{rp}(0)$ is given by

$$x_{rp}(0) = x(\overleftarrow{p}) \tag{7.49}$$

where $\overleftarrow{p}$ is the r-bit binary representation of p reversed in decimal form. For example, if $N = 16$, $r = 4$ and hence we have

$$\begin{array}{rcccc}
x_{40}(0) = x(\overleftarrow{0}) & \rightarrow 0000 & \rightarrow 0000 & \rightarrow & x(0) \\
x_{41}(0) = x(\overleftarrow{1}) & \rightarrow 0001 & \rightarrow 1000 & \rightarrow & x(8) \\
x_{42}(0) = x(\overleftarrow{2}) & \rightarrow 0010 & \rightarrow 0100 & \rightarrow & x(4) \\
\vdots & \vdots & \vdots & & \vdots \\
x_{4(15)}(0) = x(\overleftarrow{15}) & \rightarrow 1111 & \rightarrow 1111 & \rightarrow & x(15)
\end{array}$$

In effect, *sequence* $\{x_{r0}(0), x_{r1}(0), \ldots\}$ *is a reordered version of sequence* $\{x(0), x(1), \ldots\}$, namely, $\{x(\overleftarrow{0}),\ x(\overleftarrow{1}),\ \ldots\}$ or $\{x(0),\ x(8),\ \ldots\}$. Entering the time sequence, properly reordered in the left-hand array in Fig. 7.24*b* and performing the computations involved in the signal flow graphs of Fig. 7.24*b* and *a* will produce the DFT of the given time sequence in the right-hand array of Fig. 7.24*a* properly ordered as illustrated.

In the above discussion, N has been assumed to be a power of 2. Nevertheless, the algorithm can be applied to any other finite-duration time sequence by including a number of trailing zero elements in the given sequence. This would amount to zero padding and, as may be recalled from Sec. 7.5, including trailing zeros improves the resolution of the frequency spectrum of the discrete-time signal.

Example 7.7 Construct the decimation-in-time algorithm for $N = 8$.

■ **Solution**

From Eqs. (7.43a) and (7.43b), we have

$$x_{10}(n) = x(2n) \qquad x_{11}(n) = x(2n+1)$$

for $n = 0,\ 1,\ 2,\ 3$. Hence

$$x_{10}(n) = \{x(0), x(2), x(4), x(6)\} \tag{7.50a}$$

$$x_{11}(n) = \{x(1), x(3), x(5), x(7)\} \tag{7.50b}$$

Now from Eqs. (7.47a)–(7.47d), we obtain

$$x_{20}(n) = x_{10}(2n) \qquad x_{21}(n) = x_{10}(2n+1)$$

$$x_{22}(n) = x_{11}(2n) \qquad x_{23}(n) = x_{11}(2n+1)$$

for $n = 0,\ 1$. Thus Eqs. (7.50a) and (7.50b), give

$$x_{20}(n) = \{x(0), x(4)\} \qquad x_{21}(n) = \{x(2), x(6)\} \tag{7.51a}$$

$$x_{22}(n) = \{x(1), x(5)\} \qquad x_{23}(n) = \{x(3), x(7)\} \tag{7.51b}$$

Finally, from Eqs. (7.48a)–(7.48d), the elements of the third cycle can be generated by replacing k in the second cycle first by $2k$ and then by $2k+1$,

$$x_{30}(n) = x_{20}(2n) \qquad x_{31}(n) = x_{20}(2n+1)$$

$$x_{32}(n) = x_{21}(2n) \qquad x_{33}(n) = x_{21}(2n+1)$$

$$x_{34}(n) = x_{22}(2n) \qquad x_{35}(n) = x_{22}(2n+1)$$

$$x_{36}(n) = x_{23}(2n) \qquad x_{37}(n) = x_{23}(2n+1)]$$

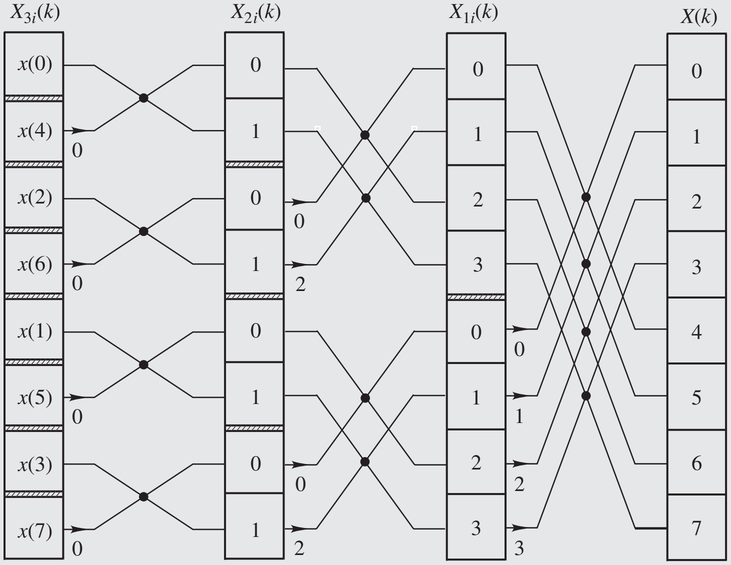


Figure 7.25 Decimation-in-time FFT algorithm for $N = 8$ (Example 7.7).

for $k = 0$. Hence from Eqs. (7.51a) and (7.51b), we obtain

$$
\begin{aligned}
x_{30}(0) &= x(0) = X(0) & \qquad x_{31}(0) &= x(4) = X(4)\\
x_{32}(0) &= x(2) = X(2) & \qquad x_{33}(0) &= x(6) = X(6)\\
x_{34}(0) &= x(1) = X(1) & \qquad x_{35}(0) &= x(5) = X(5)\\
x_{36}(0) &= x(3) = X(3) & \qquad x_{37}(0) &= x(7) = X(7)
\end{aligned}
$$

The complete algorithm is illustrated in Fig. 7.25. ■

The algorithm can be easily programmed as computations can be carried out in place in a single array. As can be observed in Fig. 7.25, once the outputs of each input butterfly are computed, the input elements are no longer needed for further processing and can be replaced by the corresponding outputs. When we proceed in the same way from left to right, at the end of computation the input array will contain the elements of the desired DFT properly ordered. The input elements can be entered in the appropriate array locations by using a simple reordering subroutine.

In general, each cycle of the algorithm involves $N/2$ butterflies, as can be seen in Fig. 7.25, and each butterfly requires one (complex) multiplication. Since there are r cycles of computation and $r = \log_2 N$, the total number of multiplications is $(N/2)\log_2 N$ as opposed to N^2 in the case of direct evaluation. This constitutes a huge saving in computation. For example, if $N \geq 512$, then the number of multiplications is reduced to *a fraction of 1 percent of that required by direct evaluation*.

7.11.2 Decimation-in-Frequency Algorithm

In the decimation-in-time algorithm, the given time sequence is split in two by separating the even- and odd-index elements. The same procedure is then applied repeatedly on each new sequence until one-element sequences are obtained. The *decimation-in-frequency* algorithm can be developed by splitting the given sequence about its midpoint and then repeating the same for each resulting sequence until one-element sequences are obtained. We can write

$$X(k) = \sum_{n=0}^{N/2-1} x(n)W_N^{-kn} + \sum_{n=N/2}^{N-1} x(n)W_N^{-kn}$$

$$= \sum_{n=0}^{N/2-1} \left[x(n) + W_N^{-kN/2}x\left(n+\frac{N}{2}\right)\right] W_N^{-kn}$$

and on replacing k first by $2k$ and then by $2k+1$, we obtain

$$X(2k) = \sum_{n=0}^{N/2-1} x_{10}(n)W_{N/2}^{-kn} = X_{10}(k) \qquad (7.52a)$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} x_{11}(n)W_{N/2}^{-kn} = X_{11}(k) \qquad (7.52b)$$

where

$$x_{10}(n) = x(n) + x\left(n+\frac{N}{2}\right) \qquad (7.53a)$$

$$x_{11}(n) = \left[x(n) - x\left(n+\frac{N}{2}\right)\right] W_N^{-n} \qquad (7.53b)$$

for $0 \le n \le N/2-1$. Thus the even- and odd-index values of $X(k)$ are given by the DFTs of $x_{10}(n)$ and $x_{11}(n)$, respectively. Assuming that the values of $x(n)$ are stored sequentially in an array, the values of $x_{10}(n)$ and $x_{11}(n)$ can be computed as illustrated in Fig. 7.26*a*, where the left-hand butterfly represents Eqs. (7.53a) and (7.53b).

The same cycle of activities can now be applied to $x_{10}(n)$ and $x_{11}(n)$. For $x_{10}(n)$, we can write

$$X(2k) = \sum_{n=0}^{N/4-1} \left[x_{10}(n) + W_N^{-kN/2}x_{10}\left(n+\frac{N}{4}\right)\right] W_{N/2}^{-kn}$$

and, similarly, for $x_{11}(n)$

$$X(2k+1) = \sum_{n=0}^{N/4-1} \left[x_{11}(n) + W_N^{-kN/2}x_{11}\left(n+\frac{N}{4}\right)\right] W_{N/2}^{-kn}$$

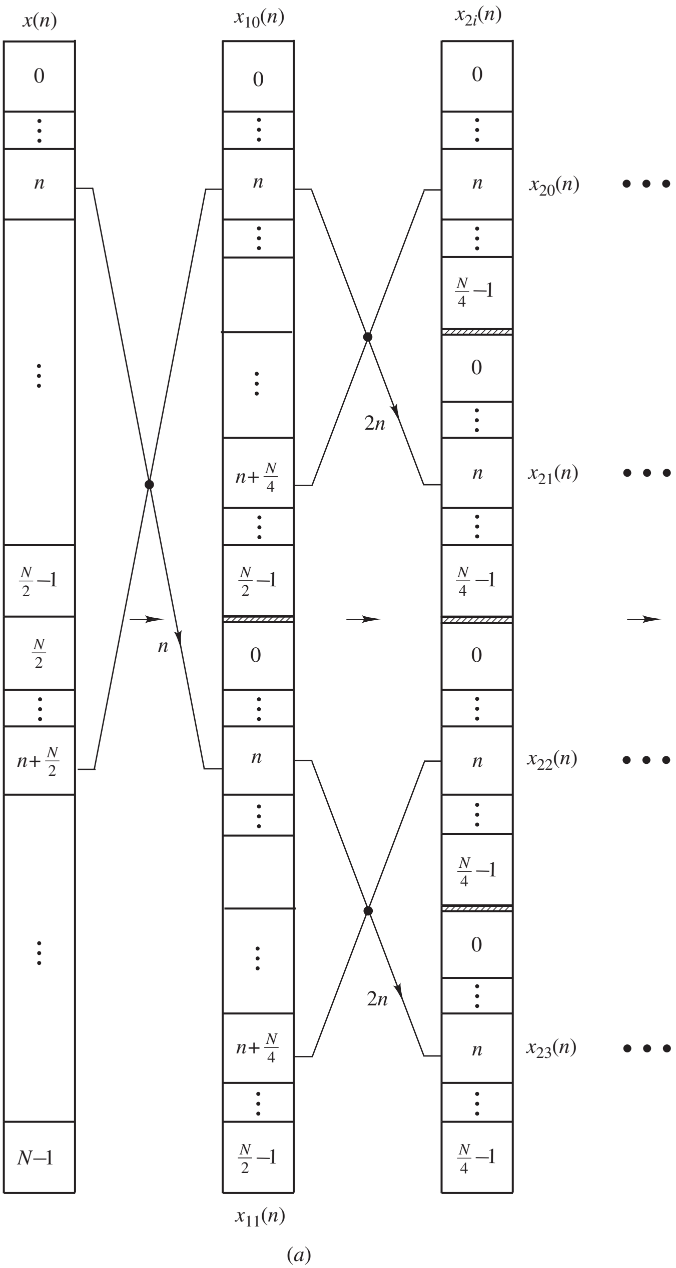


Figure 7.26 Decimation-in-frequency FFT algorithm: (*a*) First and second cycles.

Hence with k replaced first by $2k$ and then by $2k+1$, we have

$$X(4k) = \sum_{n=0}^{N/4-1} x_{20}(n)W_{N/4}^{-kn} = X_{20}(k) \tag{7.54a}$$

$$X(4k+2) = \sum_{n=0}^{N/4-1} x_{21}(n)W_{N/4}^{-kn} = X_{21}(k) \tag{7.54b}$$

$$X(4k+1) = \sum_{n=0}^{N/4-1} x_{22}(n)W_{N/4}^{-kn} = X_{22}(k) \tag{7.54c}$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} x_{23}(n)W_{N/4}^{-kn} = X_{23}(k) \tag{7.54d}$$

where

$$x_{20}(n) = x_{10}(n) + x_{10}\left(n + \frac{N}{4}\right) \tag{7.55a}$$

$$x_{21}(n) = \left[x_{10}(n) - x_{10}\left(n + \frac{N}{4}\right)\right] W^{-2n} \tag{7.55b}$$

$$x_{22}(n) = x_{11}(n) + x_{11}\left(n + \frac{N}{4}\right) \tag{7.55c}$$

$$x_{23}(n) = \left[x_{11}(n) - x_{11}\left(n + \frac{N}{4}\right)\right] W^{-2n} \tag{7.55d}$$

for $0 \le n \le (N/4-1)$. The values of $x_{20}(n)$, $x_{21}(n)$, $\ldots$ can be computed as in Fig. 7.26*a*. The DFT of each of these sequences gives one-quarter of the values of $X(k)$.

Similarly, the *mth cycle* of the above procedures would yield

$$X(2^m k) = \sum_{n=0}^{N_1-1} x_{m0}(n)W_{N_1}^{-kn} = X_{m0}(k) \tag{7.56a}$$

$$X(2^m k + 2^{m-1}) = \sum_{n=0}^{N_1-1} x_{m1}(n)W_{N_1}^{-kn} = X_{m1}(k) \tag{7.56b}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

where $N_1 = N/2^m$ and

$$x_{m0}(n) = x_{(m-1)0}(n) + x_{(m-1)0}\left(n + \frac{N}{2^m}\right)$$

$$x_{m1}(n) = \left[x_{(m-1)0}(n) - x_{(m-1)0}\left(n + \frac{N}{2^m}\right)\right] W_N^{-2^{m-1}n}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

for $0 \le n \le N/2^m - 1$.

As for the decimation-in-time algorithm, the procedure terminates with the rth cycle at which point $x_{r0}(n)$, $x_{r1}(n)$, ... reduce the one-element sequences each giving one value of the desired DFT except that the elements of the DFT are not properly ordered. What we get instead is the sequence

$$X(\overleftarrow{0}) = x_{r0}(0) \tag{7.57a}$$

$$X(\overleftarrow{1}) = x_{r1}(0) \tag{7.57b}$$

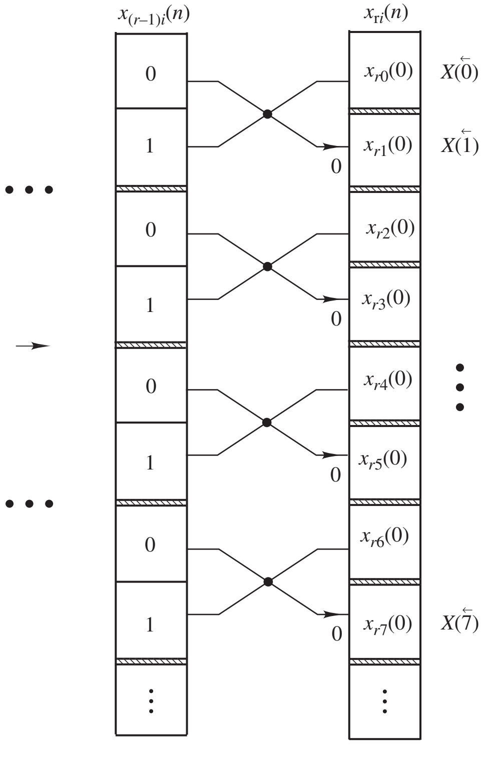
$$\vdots$$

$$X(\overleftarrow{p}) = x_{rp}(0) \tag{7.57c}$$

$$\vdots$$

$$X(\overleftarrow{N-1}) = x_{r(N-1)}(0) \tag{7.57d}$$

where $\overleftarrow{p}$ is the r-bit binary representation of p reversed in decimal form as in the decimation-in-time algorithm. The last two cycles of the algorithm are illustrated in Fig. 7.26*b*. Note that the elements of the DFT can be put in the right order by simply reversing the r-bit binary representation of the location index at the end of the computation, as is done for the input sequence in the case of the decimation-in-time algorithm. The advantage of this algorithm is that the values of $x(n)$ are entered in the input array sequentially.



(*b*)

Figure 7.26 Cont'd Decimation-in-frequency FFT algorithm: (*b*) rth cycle.

Example 7.8 Construct the decimation-in-frequency algorithm for $N = 8$.

■ **Solution**

From Eqs. (7.52a) and (7.52b), we have

$$X_{10}(k) = X(2k) \qquad \text{and} \qquad X_{11}(k) = X(2k+1)$$

for $k = 0,\ 1,\ 2,\ 3$. Hence

$$X_{10}(k) = \{X(0),\ X(2),\ X(4),\ X(6)\} \tag{7.58a}$$

$$X_{11}(k) = \{X(1),\ X(3),\ X(5),\ X(7)\} \tag{7.58b}$$

From Eqs. (7.54a)–(7.54d), we have

$$X_{20}(k) = X(4k) \qquad X_{21}(k) = X(4k+2)$$

$$X_{22}(k) = X(4k+1) \qquad X_{23}(k) = X(4k+3)$$

for $k = 0,\ 1$. Thus Eqs. (7.58a) and (7.58b), give

$$X_{20}(k) = \{X(0),\ X(4)\} \qquad X_{21}(k) = \{X(2),\ X(6)\} \tag{7.59a}$$

$$X_{22}(k) = \{X(1),\ X(5)\} \qquad X_{23}(k) = \{X(3),\ X(7)\} \tag{7.59b}$$

Finally, from Eqs. (7.56a) and (7.56b) or Eqs. (7.57a)–(7.57c), the elements of the 3rd cycle can be generated by replacing k in the 2nd cycle first by $2k$ and then by $2k + 1$, that is,

$$X_{30}(k) = X[4(2k)] \qquad X_{31}(k) = X[4(2k+1)]$$

$$X_{32}(k) = X[4(2k)+2)] \qquad X_{33}(k) = X[4(2k+1)+2]$$

$$X_{34}(k) = X[4(2k)+1] \qquad X_{35}(k) = X[4(2k+1)+1]$$

$$X_{36}(k) = X[4(2k)+3] \qquad X_{37}(k) = X[4(2k+1)+3]$$

for $k = 0$. Hence from Eqs. (7.59a) and (7.59b), we obtain

$$X_{30}(0) = X(0) = x(0) \qquad X_{31}(0) = X(4) = x(4)$$

$$X_{32}(0) = X(2) = x(2) \qquad X_{33}(0) = X(6) = x(6)$$

$$X_{34}(0) = X(1) = x(1) \qquad X_{35}(0) = X(5) = x(5)$$

$$X_{36}(0) = X(3) = x(3) \qquad X_{37}(0) = X(7) = x(7)$$

The complete algorithm is illustrated in Fig. 7.27. ■

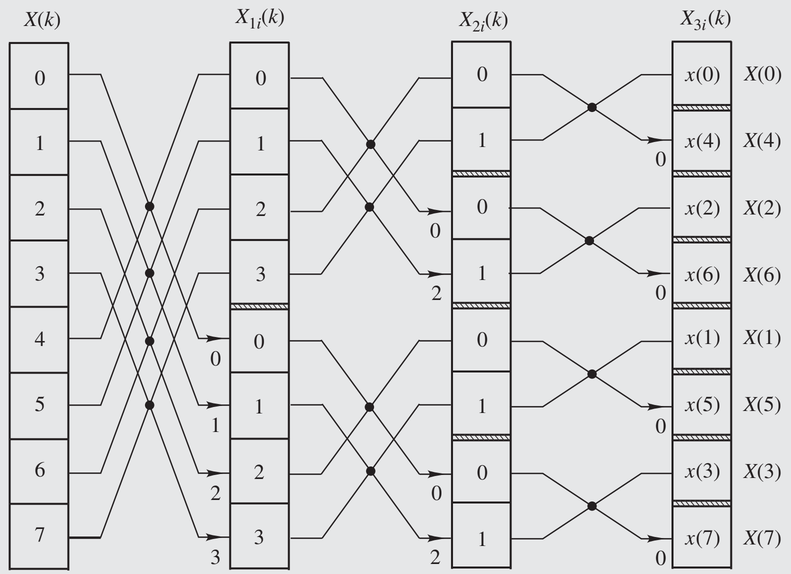


Figure 7.27 Decimation-in-frequency FFT algorithm for $N = 8$ (Example 7.8).

7.11.3 Inverse DFT

Owing to the similarity between Eqs. (7.1) and (7.2), the preceding two algorithms can be readily employed for the computation of the IDFT. Equation (7.2) can be put in the form

$$x^*(n) = \left[\frac{1}{N}\sum_{k=0}^{N-1} X(k)W^{kn}\right]^* = \frac{1}{N}\sum_{k=0}^{N-1} X^*(k)W^{-kn}$$

or

$$x^*(n) = \mathcal{D}\left[\frac{1}{N}X^*(k)\right]$$

Therefore,

$$x(n) = \left\{\mathcal{D}\left[\frac{1}{N}X^*(k)\right]\right\}^*$$

Thus if a computer program is available that can be used to compute the DFT of a complex signal $x(n)$, then exactly the same program can be used to compute the inverse DFT, i.e., $x(n)$, by entering the complex conjugate of $X(k)/N$ as input and then taking the complex conjugate of the output.

7.12 APPLICATION OF THE FFT APPROACH TO SIGNAL PROCESSING

The processing of discrete-time signals can be carried out through the use of a digital filters in the form of software or hardware. Alternatively, one can simulate the action of a digital filter through the use of the convolution summation.

The response of an FIR filter to an excitation $x(n)$ is given by

$$y(n) = \sum_{m=-\infty}^{\infty} x(n-m)h(m)$$

and if

$$\begin{aligned} h(n) &= 0 \qquad \text{for } n < 0 \qquad \text{and} \qquad n > N-1 \\ x(n) &= 0 \qquad \text{for } n < 0 \qquad \text{and} \qquad n > L-1 \end{aligned}$$

then we have

$$y(n) = \sum_{m=0}^{N-1} x(n-m)h(m) \qquad \text{for } 0 \le n \le N+L-2 \tag{7.60}$$

A software implementation for the filter can be readily obtained by programming Eq. (7.60) directly. However, this approach can involve a large amount of computation since N multiplications are necessary for each sample of the response. The alternative is to use the FFT method [12].

Let us define $(L+N-1)$-element DFTs for $h(n)$, $x(n)$, and $y(n)$, as in Sec. 7.2, which we can designate as $H(k)$, $X(k)$, and $Y(k)$, respectively. From Eqs. (7.39) and (7.60), we have

$$Y(k) = H(k)X(k)$$

and hence

$$y(n) = \mathcal{D}^{-1}[H(k)X(k)]$$

Therefore, an arbitrary finite-duration signal can be processed through the following procedure:

1. Compute the DFTs of $h(n)$ and $x(n)$ using an FFT algorithm.
2. Compute the product $H(k)X(k)$ for $k = 0, 1, \ldots$.
3. Compute the IDFT of $Y(k)$ using an FFT algorithm.

The evaluation of $H(k)$, $X(k)$, or $y(n)$ requires $[(L+N-1)/2]\log_2(L+N-1)$ complex multiplications, and step 2 above entails $L+N-1$ of the same. Since one complex multiplication corresponds to four real ones, the total number of real multiplications per output sample is $6\log_2(L+N-1)+4$, as opposed to N in the case of direct evaluation using Eq. (7.60). Clearly, for large values of N, the FFT approach is much more efficient. For example, if $N = L = 512$, the number of multiplications would be reduced to 12.5 percent of that required by direct evaluation.

The above convolution method of implementing digital filters can also be applied to IIR digital filters but only if the frequency response of the filter is bandlimited. In such a case, an impulse response of finite duration can be obtained through the use of a suitable window function.

In the convolution method for the implementation of digital filters, the entire input sequence must be available before the processing can start. Consequently, if the input sequence is long, a long delay known as *latency* will be introduced, which is usually objectionable in *real-time* or even *quasi-real-time* applications. For such applications, the input sequence is usually broken down into small blocks or segments that can be processed individually. In this way, the processing can begin as soon as the first segment is received and the processed signal begins to become available soon after. Simultaneously, new segments of the input continue to be received while the processing continues.

Two segmentation techniques have evolved for the processing of signals, as follows:

1. Overlap-and-add method
2. Overlap-and-save method

These are two somewhat different schemes of dealing with the fact that the periodic convolution produces a longer sequence than the length of either the signal $x(n)$ or the impulse response $h(n)$ of the filter being simulated.

7.12.1 Overlap-and-Add Method

In the overlap-and-add method, successive convolution summations produce consecutive processed segments of the signal that are overlapped to give the overall processed signal as will be shown below.

The input signal can be expressed as a sum of signal segments $x_i(n)$ for $i = 1, 2, \ldots, q$, each comprising L samples, such that

$$x(n) = \sum_{i=0}^{q} x_i(n)$$

for $0 \le n \le qL - 1$, where

$$x_i(n) = \begin{cases} x(n) & \text{for } iL \le n \le (i+1)L - 1 \\ 0 & \text{otherwise} \end{cases} \tag{7.61}$$

as illustrated in Fig. 7.28. With this manipulation, Eq. (7.60) assumes the form

$$y(n) = \sum_{m=0}^{N-1} \sum_{i=0}^{q} x_i(n-m)h(m)$$

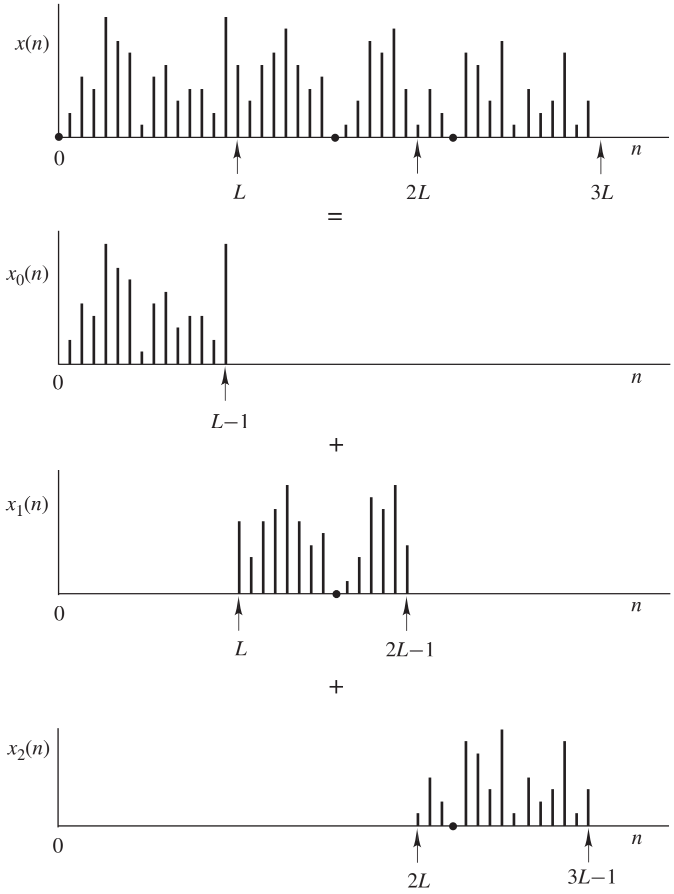


Figure 7.28 Segmentation of input sequence.

and on interchanging the order of summation, we get

$$y(n) = \sum_{i=0}^{q} c_i \tag{7.62}$$

where

$$c_i(n) = \sum_{m=0}^{N-1} x_i(n-m)h(m) \tag{7.63}$$

In this way, $y(n)$ can be computed by evaluating a number of partial convolutions.

For $iL - 1 \le n \le (i+1)L + N - 1$, Eqs. (7.63) and (7.61) give

$$\begin{aligned}
c_i(iL-1) &= 0 \\
c_i(iL) &= x(iL)h(0) \\
c_i(iL+1) &= x(iL+1)h(0) + x(iL)h(1) \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
c_i[(i+1)L+N-2] &= x[(i+1)L-1]h(N-1) \\
c_i[(i+1)L+N-1] &= 0
\end{aligned}$$

Evidently, the ith partial-convolution sequence has $L+N-1$ nonzero elements which can be stored in an array C_i, as demonstrated in Fig. 7.29. From Eq. (7.63), the elements of C_i can be computed as

$$c_i(n) = \mathcal{D}^{-1}[H(k)X_i(k)]$$

Now from Eq. (7.62), an array Y containing the values of $y(n)$ can be readily formed, as illustrated in Fig. 7.29, by *entering the elements of nonoverlapping segments in* C_0, $C_1, \ldots$ *and then adding*

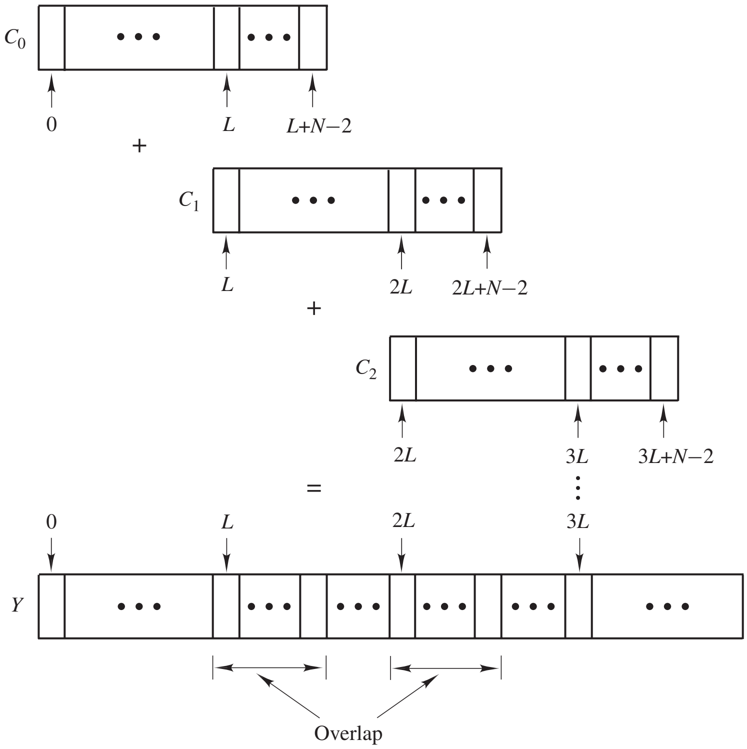


Figure 7.29 Overlap-and-add implementation.

the elements in overlapping adjacent segments. As can be seen, processing can start as soon as L input samples are received, and the first batch of L output samples is available as soon as the first input segment is processed. Evidently, a certain amount of latency is still present but through the overlap-and-add method, this is reduced from $(qL - 1)T$ to $(L - 1)T$ s where T is the sampling period.

7.12.2 Overlap-and-Save Method

If $x(n) = 0$ for $n < 0$ as before, then the first L elements of convolution summation $c_0(n)$, namely, elements 0 to $L - 1$, are equal to the corresponding L elements of $y(n)$. However, this does not apply to the last $N - 1$ elements of $c_0(n)$, i.e., elements L to $L + N - 2$, owing to the overlap between convolutions $c_0(n)$ and $c_1(n)$ as can be seen in Fig. 7.29. This problem can be avoided through the following scheme.

If we define $\bar{x}_1(n)$ such that

$$\bar{x}_1(n) = \begin{cases} x(n) & \text{for } L - (N-1) \le n \le 2L - (N-1) - 1 \\ 0 & \text{otherwise} \end{cases}$$

as illustrated in Fig. 7.30, then the convolution of $\bar{x}_1(n)$ with $h(n)$ would assume the form

$$\bar{c}_1(n) = \sum_{m=0}^{N-1} \bar{x}_1(n-m)h(m) \qquad \text{for } L - (N-1) \le n \le 2L - 1$$

Straightforward evaluation of $\bar{c}_1(n)$ for $n = L,\ L + N - 2$, and $2L - (N - 1) - 1$ gives

$$\begin{aligned}
\bar{c}_1(L) &= \bar{x}_1(L)h(0) + \bar{x}_1(L-1)h(1) + \cdots + \bar{x}_1(L-N+1)h(N-1) \\
&= c_0(L) + c_1(L) = y(L) \\
\bar{c}_1(L+N-2) &= \bar{x}_1(L+N-2)h(0) + \bar{x}_1(L+N-3)h(1) + \cdots \\
&\quad + \bar{x}_1(L-1)h(N-1) \\
&= c_0(L+N-2) + c_1(L+N-2) = y(L+N-2) \\
\bar{c}_1[2L-(N-1)-1] &= \bar{x}_1(2L-N)h(0) + \bar{x}_1(2L-N-1)h(1) + \cdots \\
&\quad + \bar{x}_1(2L-2N+1)h(N-1) \\
&= c_1[2L-(N-1)-1] = y[2L-(N-1)-1]
\end{aligned}$$

where $c_i(n)$ for $i = 0, 1$ are given by Eq. (7.63) and L is assumed to be greater than $2(N - 1)$ for the sake of convenience. Evidently,

$$\bar{c}_1(n) = \sum_{m=0}^{N-1} x(n-m)h(m) = y(n) \qquad \text{for } L \le n \le 2L - (N-1) - 1$$

that is, $\bar{c}_1(n)$ gives elements L to $2L - (N - 1) - 1$ of the required output, which can be stored in the unshaded part of array C_1 in Fig. 7.31.

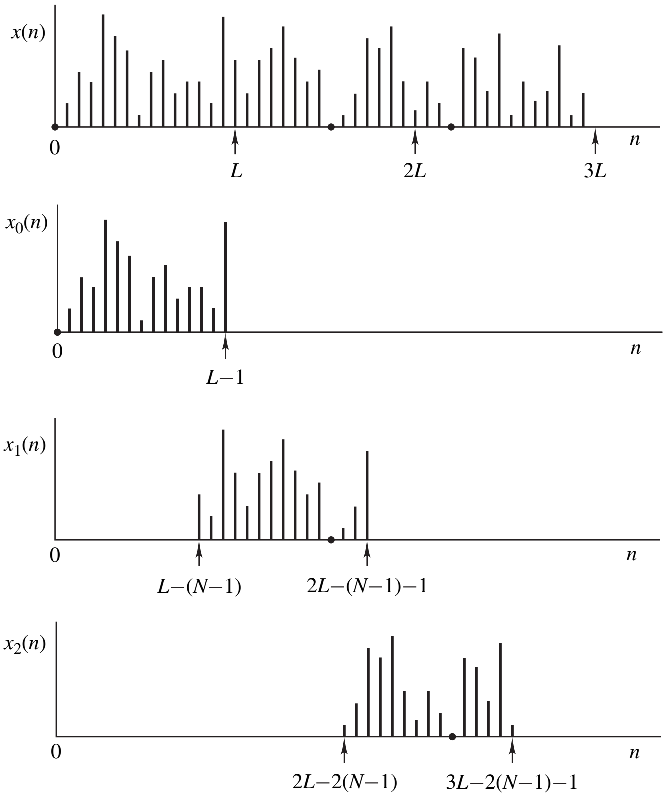


Figure 7.30 Alternative segmentation of input sequence.

Similarly, by letting

$$\bar{x}_i(n) = \begin{cases} x(n) & \text{for } iL - (i-1)(N-1) \le n \le (i+1)L - i(N-1) - 1 \\ 0 & \text{otherwise} \end{cases}$$

one can easily show that for $iL - (i-1)(N-1) \le n \le (i+1)L - i(N-1) - 1$

$$\bar{c}_i(n) = y(n) \tag{7.64}$$

for $i = 2, 3, \ldots$ (see Prob. 7.34). In effect, *the processed signal can be evaluated by computing the first L elements of $c_0(n)$ and elements $iL - (i-1)(N-1)$ to $(i+1)L - i(N-1) - 1$ of the partial convolutions $\bar{c}_i(n)$ for $i = 1, 2, \ldots$, and then concatenating the sequences obtained* as shown in Fig. 7.31.

In the scheme just described, the input sequences rather than the output sequences are overlapped, as can be seen in Fig. 7.30, and the last $N - 1$ elements of each input sequence are saved

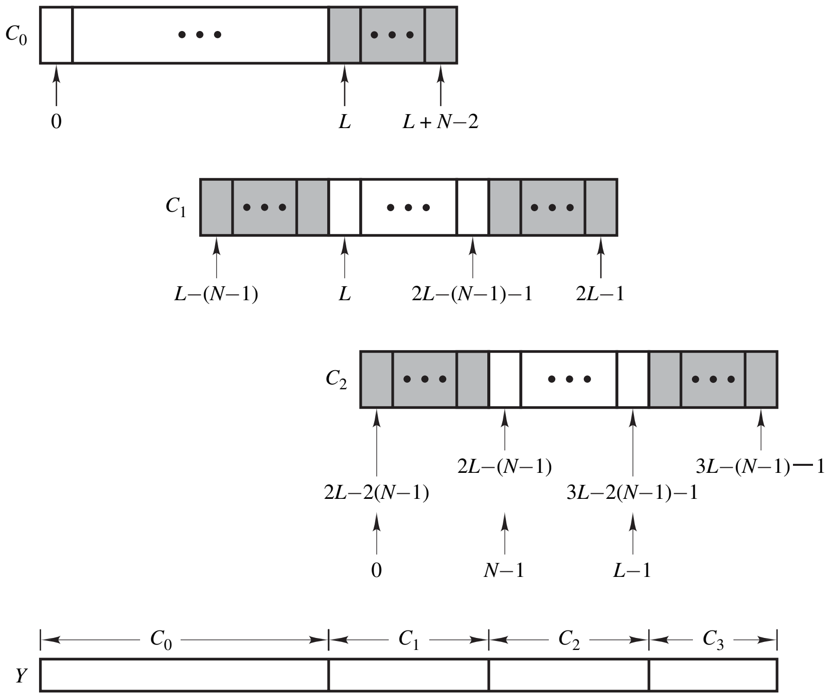


Figure 7.31 Overlap-and-save implementation.

to be re-used for the computation of the next partial convolution. For these reasons, the scheme is known as the *overlap-and-save method*.

REFERENCES

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math Comp.,* vol. 19, pp. 297–301, Apr. 1965.

[2] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, D. E. Nelson, C. M. Rader, and P. D. Welch, "What is the fast Fourier transform?" *IEEE Trans. Audio Electroacoust.*, vol. 15, pp. 45–55, June 1967.

[3] G. D. Bergland, "A guided tour of the fast Fourier transform," *IEEE Spectrum*, vol. 6, pp. 41–52, July 1969.

[4] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "Historical notes on the fast Fourier transform," *IEEE Trans. Audio Electroacoust.*, vol. 15, pp. 76–79, June 1967.

[5] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "Application of the Fast Fourier transform to computation of Fourier integrals, Fourier series and convolution integrals," *IEEE Trans. Audio Electroacoust.*, vol. 15, pp. 79–84, June 1967.

[6] J. F. Kaiser, "Nonrecursive digital filter design using the I_0-sinh window function," *IEEE Int. Symp. Circuit Theory*, pp. 20–23, 1974.

[7] H. Babic and G. C. Temes, "Optimum low-order windows for discrete Fourier transform systems," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, pp. 512–517, Dec. 1976.

[8] C. L. Dolph, "A current distribution for broadside arrays which optimizes the relationship between beamwidth and side-lobe level," *Proc. IRE*, vol. 34, pp. 335–348, June 1946.

[9] R. L. Streit, "A two-parameter family of weights for nonrecursive digital filters and antennas," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, pp. 108–118, Feb. 1984.

[10] S. W. A. Bergen and A. Antoniou, "Design of ultraspherical window functions with prescribed spectral characteristics," *Applied Journal of Signal Processing*, vol. 13, pp. 2053–2065, 2004.

[11] M. L. James, G. M. Smith, and J. C. Wolford *Applied Numerical Methods for Digital Computation*, 3rd ed., New York: Harper & Row, 1985.

[12] H. D. Helms, "Fast Fourier transform method of computing difference equations and simulating filters," *IEEE Trans. Audio Electroacoust.*, vol. 15, pp. 85–90, June 1967.

PROBLEMS

7.1. Show that

$$\sum_{k=0}^{N-1} W^{k(n-m)} = \begin{cases} N & \text{for } m = n \\ 0 & \text{otherwise} \end{cases}$$

7.2. Show that

(a) $\mathcal{D}\tilde{x}(nT + mT) = W^{km}\tilde{X}(jk\Omega)$

(b) $\mathcal{D}^{-1}\tilde{X}(jk\Omega + jl\Omega) = W^{-nl}\tilde{x}(nT)$

7.3. The definition of the DFT can be extended to include complex discrete-time signals. Show that

(a) $\mathcal{D}\tilde{x}^*(nT) = \tilde{X}^*(-jk\Omega)$

(b) $\mathcal{D}^{-1}\tilde{X}^*(jk\Omega) = \tilde{x}^*(-nT)$

7.4. (a) A complex discrete-time signal is given by

$$\tilde{x}(nT) = \tilde{x}_1(nT) + j\tilde{x}_2(nT)$$

where $\tilde{x}_1(nT)$ and $\tilde{x}_2(nT)$ are real. Show that

$$\begin{aligned}
\Re e\,\tilde{X}_1(jk\Omega) &= \tfrac{1}{2}\{\Re e\,\tilde{X}(jk\Omega) + \Re e\,\tilde{X}[j(N-k)\Omega]\} \\
\Im m\,\tilde{X}_1(jk\Omega) &= \tfrac{1}{2}\{\Im m\,\tilde{X}(jk\Omega) - \Im m\,\tilde{X}[j(N-k)\Omega]\} \\
\Re e\,\tilde{X}_2(jk\Omega) &= \tfrac{1}{2}\{\Im m\,\tilde{X}(jk\Omega) + \Im m\,\tilde{X}[j(N-k)\Omega]\} \\
\Im m\,\tilde{X}_2(jk\Omega) &= -\tfrac{1}{2}\{\Re e\,\tilde{X}(jk\Omega) - \Re e\,\tilde{X}[j(N-k)\Omega]\}
\end{aligned}$$

(b) A DFT is given by

$$\tilde{X}(jk\Omega) = \tilde{X}_1(jk\Omega) + j\tilde{X}_2(jk\Omega)$$

where $\tilde{X}_1(jk\Omega)$ and $\tilde{X}_2(jk\Omega)$ are real DFTs. Show that

$$\begin{aligned}
\Re e\,\tilde{x}_1(nT) &= \tfrac{1}{2}\{\Re e\,\tilde{x}(nT) + \Re e\,\tilde{x}[(N-n)T]\} \\
\Im m\,\tilde{x}_1(nT) &= \tfrac{1}{2}\{\Im m\,\tilde{x}(nT) - \Im m\,\tilde{x}[(N-n)T]\} \\
\Re e\,\tilde{x}_2(nT) &= \tfrac{1}{2}\{\Im m\,\tilde{x}(nT) + \Im m\,\tilde{x}[(N-n)T]\} \\
\Im m\,\tilde{x}_2(nT) &= -\tfrac{1}{2}\{\Re e\,\tilde{x}(nT) - \Re e\,\tilde{x}[(N-n)T]\}
\end{aligned}$$

7.5. Figure P7.5 shows four real discrete-time signals. Classify their DFTs as real, imaginary, or complex. Assume that $N = 10$ in each case.

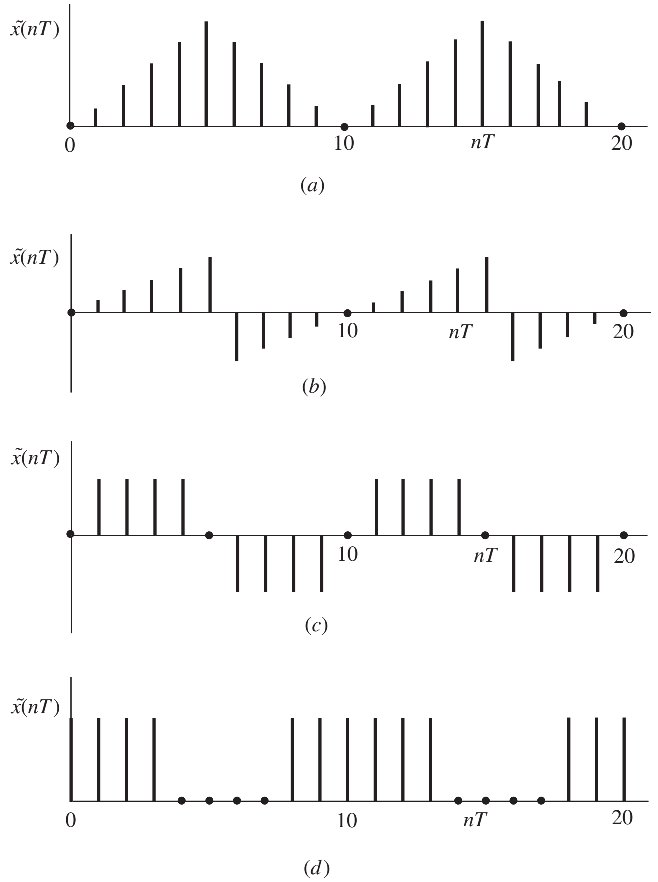


Figure P7.5

7.6. Find the DFTs of the following periodic signals:

$$(a)\ \tilde{x}(nT) = \begin{cases} 1 & \text{for } n = 3, 7 \\ 0 & \text{for } n = 0, 1, 2, 4, 5, 6, 8, 9 \end{cases}$$

$$(b)\ \tilde{x}(nT) = \begin{cases} 1 & \text{for } 0 \le n \le 5 \\ 2 & \text{for } 6 \le n \le 9 \end{cases}$$

7.7. Find the DFTs of the following periodic signals:

$$(a)\ \tilde{x}(nT) = \begin{cases} 2e^{-an} & \text{for } 0 \le n \le 5 \\ 0 & \text{for } 6 \le n \le 9 \end{cases}$$

The period is 10 in each case.

$$(b)\ \tilde{x}(nT) = \begin{cases} n & \text{for } 0 \le n \le 2 \\ 0 & \text{for } 3 \le n \le 7 \\ -(10-n) & \text{for } n = 8, 9 \end{cases}$$

The period is 10 in each case.

7.8. Find the DFTs of the following periodic signals in closed form:

(*a*) $x(n) = e^{-\beta n}$ for $0 \le n \le 31$ if $N = 32$.

(*b*) Repeat part (*a*) for $x(n) = e^{-\gamma n}/2^{\epsilon}$ for $0 \le n \le 31$ if $N = 32$.

7.9. A periodic signal is given by

$$\tilde{x}(nT) = \sum_{r=-\infty}^{\infty} w_{\text{H}}(nT + rNT)$$

where

$$w_{\text{H}}(nT) = \begin{cases} \alpha + (1-\alpha)\cos\dfrac{2\pi n}{N-1} & \text{for } |n| \le \dfrac{N-1}{2} \\ 0 & \text{otherwise} \end{cases}$$

Find $\tilde{X}(jk\Omega)$.

7.10. Obtain the IDFTs of the following:

(*a*) $\tilde{X}(jk\Omega) = (-1)^k \left(1 + 2\cos\dfrac{2\pi k}{10}\right)$

(*b*) $\tilde{X}(jk\Omega) = 1 + 2j(-1)^k \left(\sin\dfrac{3k\pi}{5} + \sin\dfrac{4k\pi}{5}\right)$

The value of N is 10.

7.11. (*a*) Find the z transform of $x(nT)$ for the DFTs of Prob. 7.10. Assume that $x(nT) = 0$ outside the range $0 \le n \le 9$ in each case.

7.12. (*a*) Working from first principles, derive an expression for the frequency spectrum of the rectangular window of length 31 in closed form.

(*b*) Repeat part (*a*) for a window length of 32.

7.13. (*a*) Starting with Eq. (7.25), derive Eq. (7.29).

(*b*) Starting with Eq. (7.27), derive Eq. (7.32).

7.14. Show that the Kaiser window includes the rectangular window as a special case.

7.15. Compute the values of the Kaiser window of length $N_w = 7$ and $\alpha = 3.0$.

7.16. Construct Table 7.1 for a Kaiser length of length 31.

7.17. Function $w_{\text{H}}(nT)$ in Prob. 7.9 with $\alpha = 0.54$ is known as the *Hamming* window. Obtain a closed-form expression for the frequency spectrum of the window.

7.18. Using MATLAB or similar software, plot the ripple ratio and main-lobe width of the Hamming window described in Prob. 7.17 as a function of the window length.

7.19. The triangular window [4] is given by

$$w_{\text{TR}}(nT) = \begin{cases} 1 - \dfrac{2|n|}{N-1} & \text{for } |n| \le \dfrac{N-1}{2} \\ 0 & \text{otherwise} \end{cases}$$

[4] *This is also known as the Bartlett window.*

(*a*) Assuming that $w_{\mathrm{TR}}(t)$ is bandlimited, obtain an approximate expression for $W_{\mathrm{TR}}(e^{j\omega T})$.
(*b*) Estimate the main-lobe width if $N \gg 1$.
(*c*) Estimate the ripple ratio if $N \gg 1$.
(*Hint*: See Prob. 6.27.)

7.20. An infinite-duration discrete-time signal is described by

$$x(nT) = u(nT)\left[A_0 e^{p_0 nT} + 2M_1 e^{\sigma_1 t}\cos(\omega_1 nT + \theta_1)\right]$$

where $A_0 = 4.532$, $M_1 = 2.350$, $\theta_1 = -2.873$ rad, $p_0 = -2.322$, $\sigma_1 = -1.839$, and $\omega_1 = 1.754$ rad/s.

(*a*) Obtain an expression for the frequency spectrum of the signal.

(*b*) Plot the frequency spectrum over the range $0 \le \omega \le \omega_s/2$ assuming a sampling frequency $\omega_s = 10$ rad/s.

(*c*) Repeat part (*b*) if the signal is modified through the use of a rectangular window of length 21.

(*d*) Repeat part (*b*) if the signal is modified through the use of a Kaiser window of length 21 and $\alpha = 1.0$.

(*e*) Compare the results obtained in parts (*c*) and (*d*).

7.21. An infinite-duration right-sided discrete-time signal $x(nT)$ is obtained by sampling the continuous-time signal

$$x(t) = u(t)[A_0 e^{p_0 t} + 2M_1 e^{\sigma_1 t}\cos(\omega_1 t + \theta_1)]$$

where $A_0 = 5.0$, $M_1 = 2.0$, $\theta_1 = -3.0$ rad, $p_0 = -2.0$, $\sigma_1 = -1.5$, and $\omega_1 = 2.5$ rad/s. A finite duration signal can be obtained by applying the discrete-time Kaiser window with $\alpha = 2.0$. Following the approach in Example 7.5, find the lowest sampling frequency that would result in negligible aliasing error.

7.22. Repeat Prob. 7.21 if $A_0 = 4.0$, $M_1 = 3.0$, $\theta_1 = -2.0$ rad, $p_0 = -3.0$, $\sigma_1 = -2.0$, $\omega_1 = 1.5$ rad/s, and $\alpha = 1.5$.

7.23. Prove Theorem 6.2*B*.

7.24. (*a*) Periodic signals $x(n)$ and $h(n)$ are given by

$$x(n) = \begin{cases} 1 & \text{for } 0 \le n \le 4 \\ 2 & \text{for } 5 \le n \le 9 \end{cases}$$

$$h(n) = n \qquad \text{for } 0 \le n \le 9$$

Find the time-domain convolution

$$y(n) = \sum_{m=0}^{9} x(m)h(n-m)$$

at $n = 4$ assuming a period $N = 10$.

(*b*) Repeat part (*a*) if

$$x(n) = u(n-4)e^{-\alpha n} \qquad \text{for } 0 \le n \le 9 \quad h(n) = \begin{cases} 1 & \text{n} = 0, 1, 8, 9 \\ 0 & \text{otherwise} \end{cases}$$

7.25. Two periodic signals are given by

$$x(n) = \cos n\pi/9 \qquad \text{and} \qquad h(n) = u(n-4) \qquad \text{for } 0 \le n \le 9$$

Find the time-domain convolution $y(n)$ at $n = 5$ assuming that $N = 10$.

(*b*) Repeat part (*a*) if

$$x(n) = \cos n\pi/9 \qquad \text{and} \qquad h(n) = \begin{cases} e^{-\beta n} & \text{for } 0 \le n \le 3 \\ 0 & \text{otherwise} \end{cases}$$

7.26. Show that

$$\mathcal{D}\,[x(n)h(n)] = \frac{1}{N}\sum_{m=0}^{N-1} X(m)H(k-m)$$

where $X(k) = \mathcal{D}x(n)$ and $H(k) = \mathcal{D}h(n)$.

7.27. Construct the flow graph for a 16-element decimation-in-time FFT algorithm.

7.28. Construct the flow graph for a 16-element decimation-in-frequency FFT algorithm.

7.29. (*a*) Compute the Fourier-series coefficients for the periodic signal depicted in Fig. P7.29 by using a 32-element FFT algorithm.

(*b*) Repeat part (*a*) using a 64-element FFT algorithm.

(*c*) Repeat part (*a*) using an analytical method.

(*d*) Compare the results obtained.

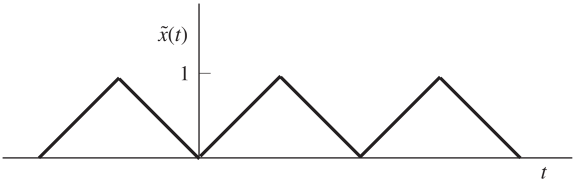


Figure P7.29

7.30. Repeat Prob. 7.29 for the signal of Fig. P7.30.

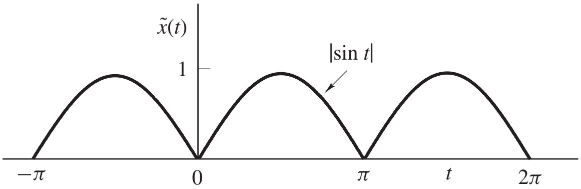


Figure P7.30

7.31. (*a*) Compute the Fourier transform of

$$x(t) = \begin{cases} \frac{1}{2}(1+\cos t) & \text{for } 0 \le |t| \le \pi \\ 0 & \text{otherwise} \end{cases}$$

by using a 64-element FFT algorithm. The desired resolution in the frequency domain is 0.5 rad/s.

(*b*) Repeat part (*a*) for a frequency domain resolution of 0.25 rad/s.

(*c*) Repeat part (*a*) by using an analytical method.

(*d*) Compare the results in parts (*a*) to (*c*).

7.32. Repeat Prob. 7.31 for the signal

$$x(t) = \begin{cases} 1 - |t| & \text{for } |t| < 1 \\ 0 & \text{otherwise} \end{cases}$$

The desired frequency-domain resolutions for parts (*a*) and (*b*) are $\pi/4$ and $\pi/8$ rad/s, respectively.

7.33. An FFT program is available which allows for a maximum of 64 complex input elements. Show that this program can be used to process a real 128-element sequence.

7.34. Demonstrate the validity of Eq. (7.64).

CHAPTER 8

REALIZATION OF DIGITAL FILTERS

8.1 INTRODUCTION

The previous chapters considered the basics of signal analysis and the characterization and analysis of discrete-time systems. From this chapter onward, the design of discrete-time systems that can be used in DSP will be examined in great detail. Discrete-time systems come in all shapes and forms. However, this textbook is concerned with discrete-time systems that can be used to reshape the spectral characteristics of discrete-time signals, and such systems are of course digital filters be it nonrecursive or recursive, FIR or IIR filters, one- or two-dimensional, single-rate or multirate, adaptive or fixed.

In broad terms, the design of digital filters encompasses all the activities that need to be undertaken from the point where a need for a specific type of digital filter is identified to the point where a prototype is constructed, tested, and approved. The compendium of activities that need to be undertaken to obtain a design can be packaged into four basic steps, as follows:

1. Approximation
2. Realization
3. Study of arithmetic errors
4. Implementation

When performed successfully, these steps would lead to the implementation of a digital filter that would satisfy a set of prescribed specifications which would depend on the application at hand.

The *approximation step* is the process of generating a transfer function that would satisfy the desired specifications, which may concern the amplitude or phase response or even the time-domain response of the filter. The available methods for the solution of the approximation problem can be classified as *direct* or *indirect*. In direct methods, the problem is solved directly in the z domain. In indirect methods, a continuous-time transfer function is first obtained and then converted into a corresponding discrete-time transfer function. Nonrecursive filters are always designed through direct methods whereas recursive filters can be designed either through direct or indirect methods. Approximation methods can also be classified as *closed-form* or *iterative*. In closed-form methods, the problem is solved through a small number of design steps using a set of closed-form formulas. In iterative methods, an initial solution is assumed and, through the application of optimization methods, a series of progressively improved solutions are obtained until some design criterion is satisfied. In general, the designer is interested in approximation methods that

- are simple,
- are reliable,
- yield precise designs,
- require minimal computation effort, and so on.

The *realization* or *synthesis* of a digital filter is the process of generating a digital-filter network or structure from the transfer function or some other characterization of the filter. The network obtained is said to be the *realization* of the transfer function. As for approximation methods, realization methods can be classified as *direct* or *indirect*. In direct methods the realization is obtained directly from a given discrete-time transfer function whereas in indirect realizations, the filter structure is obtained indirectly from an equivalent prototype analog filter. Many realization methods have been proposed in the past that lead to digital-filter structures of varying complexity and properties. The designer is usually interested in realizations that

- are easy to implement in very-large-scale integrated (VLSI) circuit form,
- require the minimum number of unit delays, adders, and multipliers,
- are not seriously affected by the use of finite-precision arithmetic in the implementation, and so on.

Designs of all types from that of a refrigerator or an electrical drill to that of a microwave communications channel entail imperfections of various sorts brought about by modeling inaccuracies, component tolerances, unusual or unexpected nonlinear effects, and so on. A design will be approved to the extent that design imperfections do not violate the desired specifications. In digital filters and digital systems in general, most imperfections are caused by numerical imprecision of some form and studying the ways in which numerical imprecision will manifest itself needs to be undertaken. During the approximation step, the coefficients of the transfer function are determined to a high degree of precision. In practice, however, digital hardware have finite precision that depends on the length of registers used to store numbers; the type of number system used (e.g., signed-magnitude, two's complement); the type of arithmetic used (e.g., fixed-point or floating-point), and so on. Consequently, filter coefficients must be quantized (e.g., rounded or truncated) before they can be stored in registers. When the transfer function coefficients are quantized, errors are introduced in the amplitude and phase responses of the filter, which are commonly referred to as *quantization errors*. Such errors can cause the digital filter to violate the required specifications and in extreme cases even to become unstable. Similarly, the signals to be processed as well as the internal signals of a digital filter (e.g., the products generated by multipliers) must be quantized. Since errors introduced by the quantization of signals are actually sources of noise (see Sec. 14.5), they can have a dramatic effect on the performance of the filter. Under these circumstances, the design process cannot be deemed to be complete until the *effects of arithmetic errors* on the performance of the filter are investigated and ways are found to mitigate any problems associated with numerical imprecision.

The *implementation* of a digital filter can assume two forms, namely, *software* or *hardware*, as detailed in Sec. 1.8. In the first case, implementation involves the simulation of the filter network on a general-purpose digital computer, workstation, or DSP chip. In the second case, it involves the conversion of the filter network into a dedicated piece of hardware. The choice of implementation is usually critically dependent on the application at hand. In *nonreal-time* applications where a record of the data to be processed is available, a software implementation may be entirely satisfactory. In *real-time* applications, however, where data must be processed at a very high rate (e.g., in communication systems), a hardware implementation is mandatory. Often the best engineering solution might be partially in terms of software and partially in terms of hardware since software and hardware are highly exchangeable nowadays.

The design of digital filters may often involve other steps that do not appear explicitly in the above list. For example, if a digital filter is required to process continuous-time signals, the *effects of the interfacing devices* (e.g., analog-to-digital and digital-to-analog converters) on the accuracy of processing must be investigated.

The natural order of the four basic design steps is as stated in the preceding discussion, namely, approximation, realization, study of imperfections, and implementation. However, realization is that much easier to learn than the approximation process and for this reason it will be treated first, in this chapter, along with some implementation aspects. The approximation step is a multifaceted activity that involves a diverse range of principles since there are many types of digital filters and many methodologies to choose from. It even necessitates on occasion the design of analog filters since some of the best IIR filters can be derived only from analog filters. The approximation step for FIR filters is considered in Chap. 9, for analog filters in Chap. 10, for IIR filters in Chaps. 11 and 12. Chapters 13 and 14 have to do with the study of numerical errors associated with the use of finite word length in digital hardware. Some more advanced, optimization-based, approximation methods for FIR and IIR filters can be found in Chaps. 15 and 16. Chapter 17 deals with a fairly advanced class of digital filters, namely, the class of *wave digital filters* which are known to possess certain highly desirable properties, and Chap. 18 which concludes the book deals with a variety of digital-filter applications.

8.2 REALIZATION

As stated in the introduction, two types of realization methods have evolved over the past 30 to 40 years, namely, direct and indirect. In direct methods, the transfer function is put in some form that enables the identification of an interconnection of elemental digital-filter subnetworks. The most frequently used direct realization methods of this class are [1–4], as follows:

1. Direct
2. Direct canonic
3. State-space
4. Lattice
5. Parallel
6. Cascade

In indirect methods, on the other hand, a given analog-filter network is represented by the so-called *wave characterization*, which is normally used to represent microwave circuits and systems, and through the use of a certain transformation the analog-filter network is converted into a topologically related digital-filter network [5–8].

8.2.1 Direct Realization

A filter characterized by the Nth-order transfer function

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{N} a_i z^{-i}}{1 + \sum_{i=1}^{N} b_i z^{-i}} \tag{8.1a}$$

can be represented by the equation

$$\frac{Y(z)}{X(z)} = H(z) = \frac{N(z)}{D(z)} = \frac{N(z)}{1 + D'(z)} \tag{8.1b}$$

where
$$N(z) = \sum_{i=0}^{N} a_i z^{-i} \tag{8.2a}$$

and
$$D'(z) = \sum_{i=1}^{N} b_i z^{-i} \tag{8.2b}$$

From Eq. (8.1b), we can write

$$Y(z) = N(z)X(z) - D'(z)Y(z)$$

or
$$Y(z) = U_1(z) + U_2(z)$$

where
$$U_1(z) = N(z)X(z) \tag{8.3a}$$

and
$$U_2(z) = -D'(z)Y(z) \tag{8.3b}$$

and hence the realization of $H(z)$ can be broken down into the realization of two simpler transfer functions, $N(z)$ and $-D'(z)$, as illustrated in Fig. 8.1.

Consider the realization of $N(z)$. From Eqs. (8.2a) and (8.3a)

$$U_1(z) = [a_0 + z^{-1} N_1(z)]X(z)$$

where
$$N_1(z) = \sum_{i=1}^{N} a_i z^{-i+1}$$

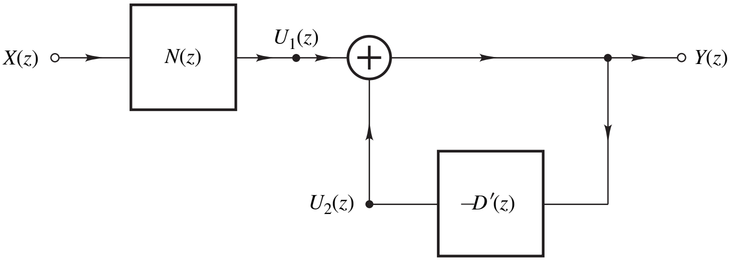


Figure 8.1 Decomposition of $H(z)$ into two simpler transfer functions.

and thus $N(z)$ can be realized by using a multiplier with a constant a_0 in parallel with a network characterized by $z^{-1}N_1(z)$. In turn, $z^{-1}N_1(z)$ can be realized by using a unit delay in cascade with a network characterized by $N_1(z)$. Since the unit delay can precede or follow the realization of $N_1(z)$, two possibilities exist for $N(z)$, as depicted in Fig. 8.2.

The above procedure can now be applied to $N_1(z)$. That is, $N_1(z)$ can be expressed as

$$N_1(z) = a_1 + z^{-1}N_2(z) \qquad \text{where } N_2(z) = \sum_{i=2}^{N} a_i z^{-i+2}$$

and as before two networks can be obtained for $N_1(z)$. Clearly, there are four networks for $N(z)$. Two of them are shown in Fig. 8.3.

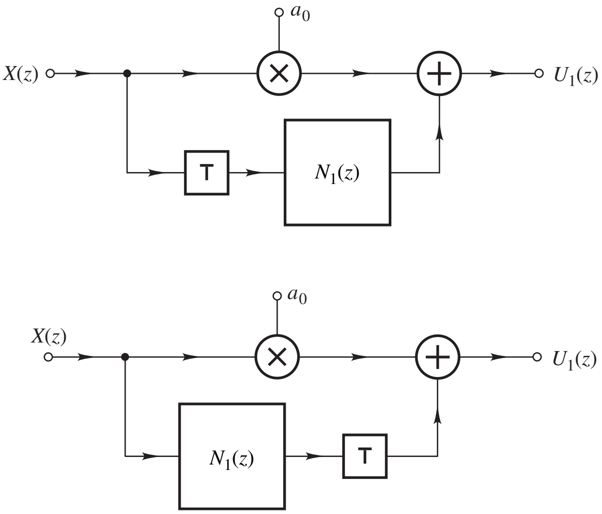


Figure 8.2 Two realizations of $N(z)$.

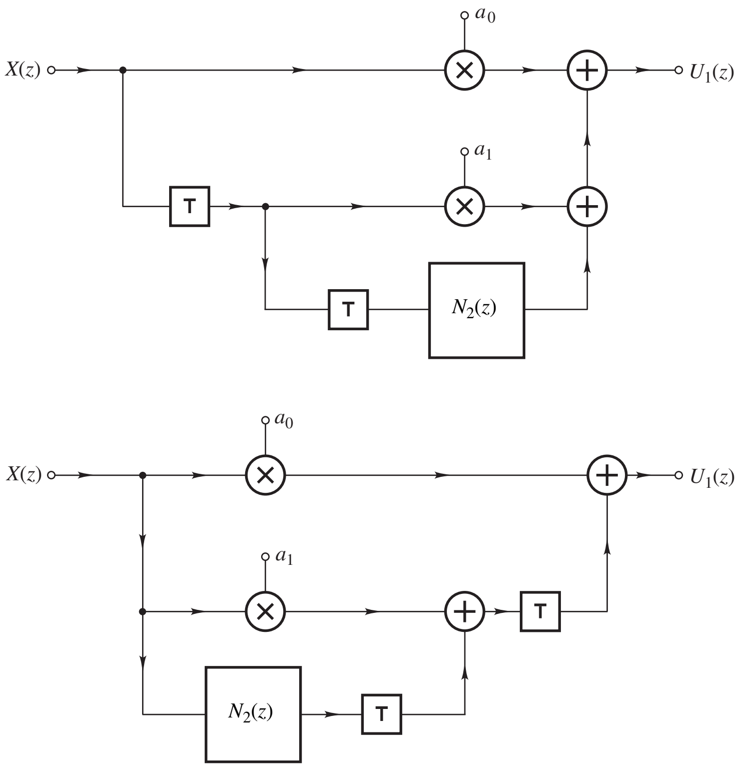
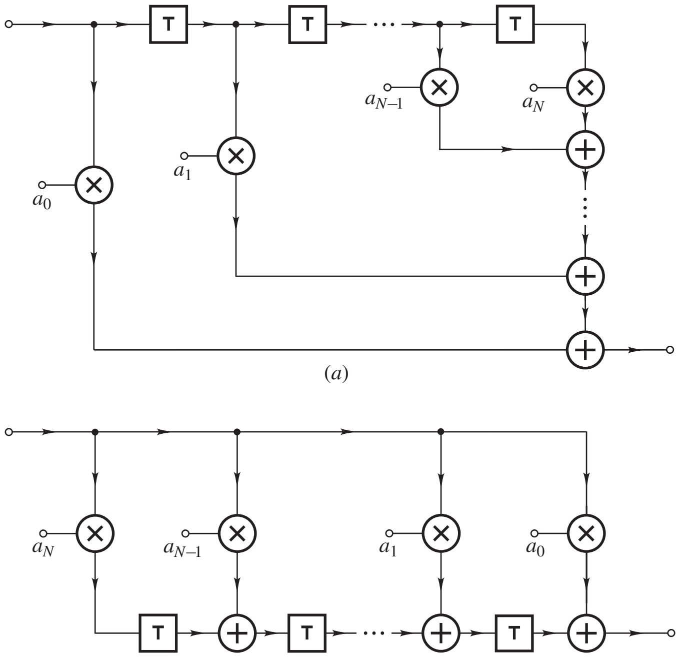


Figure 8.3 Two of four possible realizations of $N(z)$.

The above cycle of activities can be repeated N times whereupon $N_N(z)$ will reduce to a single multiplier. In each cycle of the procedure there are two possibilities, and since there are N cycles, a total of 2^N distinct networks can be deduced for $N(z)$. Three of the possibilities are depicted in Fig. 8.4*a* to *c*. These structures are obtained by placing the unit delays consistently at the left in the first case, consistently at the right in the second case, and alternately at the left and right in the third case. Note that in the realization of Fig. 8.4*a*, the adders accumulate the products generated by the multipliers from the top to the bottom of the realization. If they are added from the bottom to the top, the structure of Fig. 8.4*d* is obtained, which can form the basis of systolic structures (see Sec. 8.3.2).

$-D'(z)$ can be realized in exactly the same way by using Eqs. (8.2b) and (8.3b) instead of Eqs. (8.2a) and (8.3a) the only differences being the negative sign in $-D'(z)$ and the fact that the first term in $D'(z)$ is b_1 not b_0. Thus a network for $-D'(z)$ can be readily obtained by replacing $a_0, a_1, a_2, \ldots$ in Fig. 8.4a by 0, $-b_1, -b_2, \ldots$.

Finally, the realization of $H(z)$ can be accomplished by interconnecting the realizations of $N(z)$ and $-D'(z)$ as in Fig. 8.1.



(*a*)

(*b*)

Figure 8.4 Four possible realizations of $N(z)$.

Example 8.1 Realize the transfer function

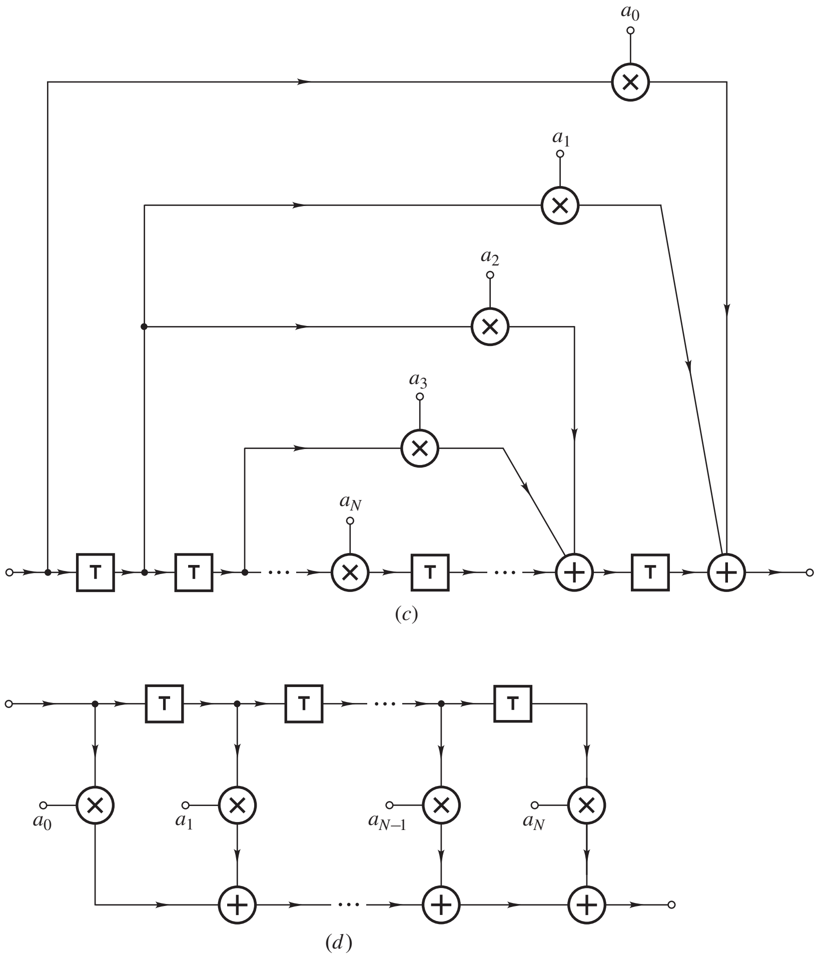
$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

■ **Solution**

Two realizations of $H(z)$ can be readily obtained from Fig. 8.4*a* and *b*, as shown in Fig. 8.5*a* and *b*. ■

8.2.2 Direct Canonic Realization

The smallest number of unit delays required to realize an Nth-order transfer function is N. An Nth-order discrete-time network that employs just N unit delays is said to be *canonic* with respect to the number of unit delays. The direct realization of the previous section does not yield canonic structures but through the use of a specific nonrecursive realization from those obtained through the direct realization it is possible to eliminate half of the unit delays, as will now be shown.



(*c*)

(*d*)

Figure 8.4 Cont'd Four possible realizations of $N(z)$.

Equation (8.1b) can be expressed as

$$Y(z) = N(z)Y'(z)$$

where

$$Y'(z) = \frac{X(z)}{1 + D'(z)} \qquad \text{or} \qquad Y'(z) = X(z) - D'(z)Y'(z)$$

With this manipulation, $H(z)$ can be realized as shown by the block diagram in Fig. 8.6*a*. On using the nonrecursive network of Fig. 8.4*a* for both $N(z)$ and $-D'(z)$ in Fig. 8.6*a*, the realization of Fig. 8.6*b* can be obtained after replacing 2-input by multiinput adders. As can be observed in Fig. 8.6*b*, the signals at nodes A', B', ... are equal to the corresponding signals at nodes A, B, ... Therefore, nodes A', B', ... can be merged with nodes A, B, ..., respectively, and one set of unit delays can be eliminated to yield a more economical canonic realization.

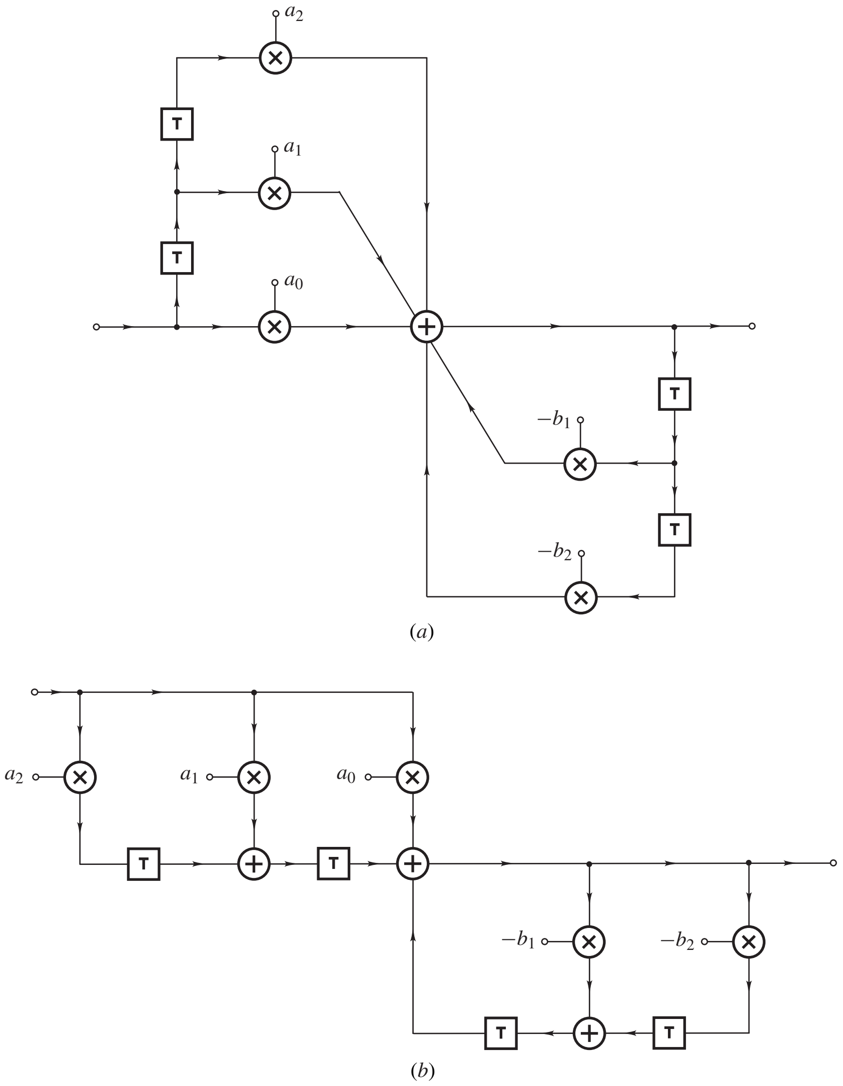


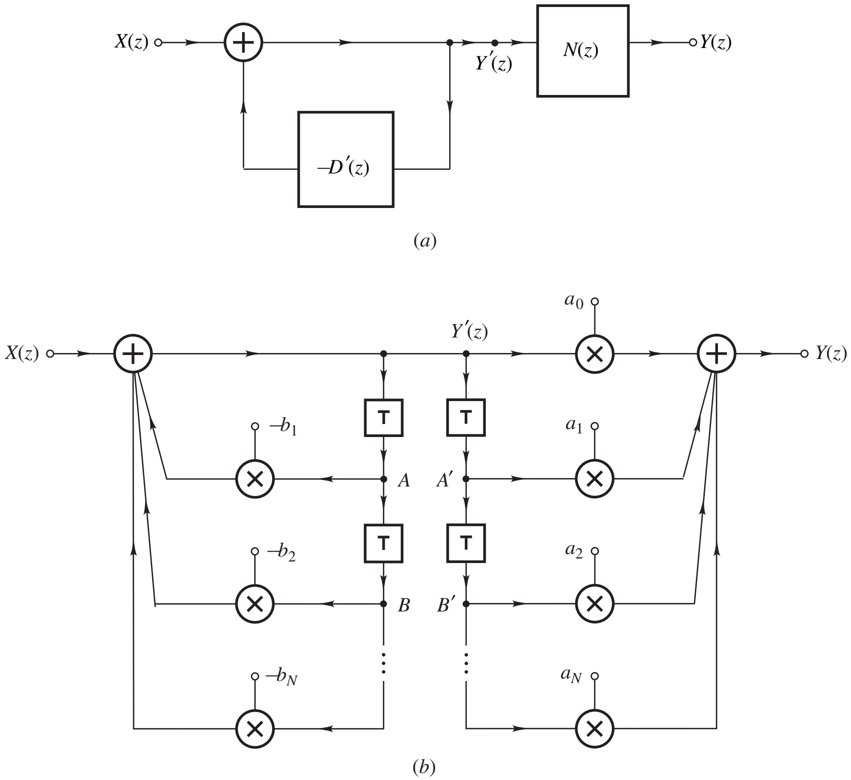
Figure 8.5 Two possible realizations of $H(z)$ (Example 8.1): (*a*) Using the structure in Fig. 8.4*a*, (*b*) using the structure in Fig. 8.4*b*.

8.2.3 State-Space Realization

Another approach to the realization of digital filters is to start with the *state-space* characterization

$$\mathbf{q}(nT+T) = \mathbf{A}\mathbf{q}(nT) + \mathbf{b}x(nT) \tag{8.4a}$$

$$y(nT) = \mathbf{c}^T\mathbf{q}(nT) + dx(nT) \tag{8.4b}$$



(a)

(b)

Figure 8.6 Derivation of the canonic realization of $H(z)$: (a) Block diagram, (b) possible realization.

For an Nth-order filter, Eqs. (8.4a) and (8.4b) give

$$q_i(nT+T)=\sum_{j=1}^{N} a_{ij}q_j(nT)+b_i x(nT) \qquad \text{for } i=1,\ 2,\dots,\ N \tag{8.5}$$

and

$$y(nT)=\sum_{j=1}^{N} c_j q_j(nT)+d_0 x(nT) \tag{8.6}$$

respectively. By assigning nodes to $x(nT)$, $y(nT)$, $q_i(nT)$, and $q_i(nT+T)$ for $i=1,2,\dots,N$, the state-space signal flow graph of Fig. 8.7 can be obtained, which can be readily converted into a network.

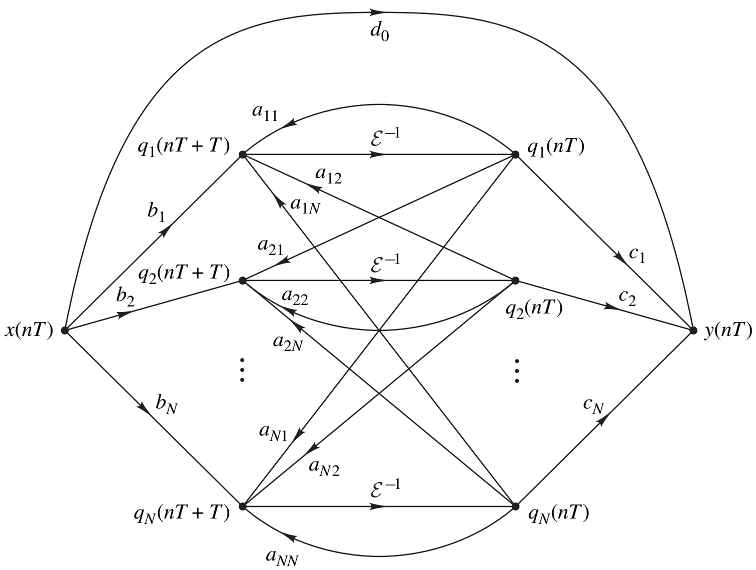


Figure 8.7 State-space signal flow graph.

Example 8.2 A digital filter is characterized by the state-space equations in Eqs. (8.4a) and (8.4b) with

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{3} & -\frac{1}{4} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{c}^T = \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} & \frac{1}{12} \end{bmatrix} \qquad d = 2$$

Obtain a direct *canonic* realization.

■ **Solution**

In order to obtain a direct canonic realization, we need to deduce the transfer function of the filter. From Eq. (5.9) and Example 5.4, we have

$$\begin{aligned} \frac{Y(z)}{X(z)} &= H(z) = \frac{N(z)}{D(z)} = \mathbf{c}^T(z\mathbf{I}-\mathbf{A})^{-1}\mathbf{b} + d \\ &= \frac{1}{\det(z\mathbf{I}-\mathbf{A})} \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} & \frac{1}{12} \end{bmatrix} \begin{bmatrix} z^2 & \frac{1}{3}z + \frac{1}{4} & -\frac{1}{4}z \\ -z & \left(z+\frac{1}{2}\right)z & \frac{1}{4} \\ 1 & -\left(z+\frac{1}{2}\right) & \left(z+\frac{1}{2}\right)z + \frac{1}{3} \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + 2 \\ &= \frac{1}{\det(z\mathbf{I}-\mathbf{A})} \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} & \frac{1}{12} \end{bmatrix} \begin{bmatrix} 2z^2 \\ -2z \\ 2 \end{bmatrix} + 2 \end{aligned}$$

where

$$\det(z\mathbf{I} - \mathbf{A}) = z^3 + \tfrac{1}{2}z^2 + \tfrac{1}{3}z + \tfrac{1}{4}$$

Thus polynomials $N(z)$ and $D(z)$ can be deduced as

$$N(z) = \left[-\tfrac{1}{4}\ \tfrac{1}{6}\ \tfrac{1}{12}\right] \begin{bmatrix} 2z^2 \\ -2z \\ 2 \end{bmatrix} + 2\det(z\mathbf{I} - \mathbf{A})$$

$$= 2z^3 + \tfrac{1}{2}z^2 + \tfrac{1}{3}z + \tfrac{2}{3}$$

and

$$D(z) = \det(z\mathbf{I} - \mathbf{A}) = z^3 + \tfrac{1}{2}z^2 + \tfrac{1}{3}z + \tfrac{1}{4}$$

respectively. Therefore,

$$H(z) = \frac{2z^3 + \frac{1}{2}z^2 + \frac{1}{3}z + \frac{2}{3}}{z^3 + \frac{1}{2}z^2 + \frac{1}{3}z + \frac{1}{4}} = \frac{2 + \frac{1}{2}z^{-1} + \frac{1}{3}z^{-2} + \frac{2}{3}z^{-3}}{1 + \frac{1}{2}z^{-1} + \frac{1}{3}z^{-2} + \frac{1}{4}z^{-3}}$$

The required realization is shown in Fig. 8.8 where

$$a_0 = 2 \qquad a_1 = \tfrac{1}{2} \qquad a_2 = \tfrac{1}{3} \qquad a_3 = \tfrac{2}{3}$$
$$b_1 = \tfrac{1}{2} \qquad b_2 = \tfrac{1}{3} \qquad b_3 = \tfrac{1}{4} \quad \blacksquare$$

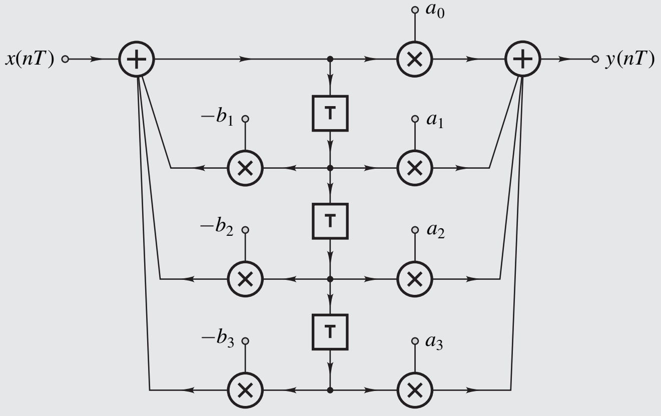


Figure 8.8 Canonic realization (Example 8.2).

8.2.4 Lattice Realization

Yet another method is the so-called *lattice realization* method of Gray and Markel [4]. This is based on the configuration depicted in Fig. 8.9*a*. The networks represented by the blocks in Fig. 8.9*a* can assume a number of distinct forms. The most basic section is the 2-multiplier first-order lattice section depicted in Fig. 8.9*b*.

A transfer function of the type given by Eq. (8.1a) can be realized by obtaining values for the multiplier constants $\nu_0, \nu_1, \ldots, \nu_N$ and $\mu_1, \mu_2, \ldots, \mu_N$ in Fig. 8.9*a* using the transfer function coefficients $a_0, a_1, \ldots, a_N$ and $1, b_1, \ldots, b_N$. The realization can be accomplished by using a recursive algorithm comprising N iterations whereby polynomials of the form

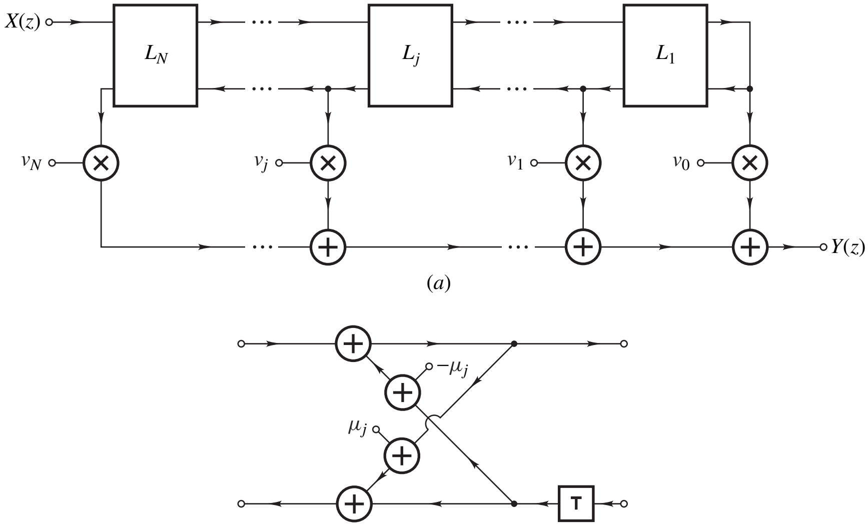
$$N_j(z) = \sum_{i=0}^{j} \alpha_{ji} z^{-i} \qquad D_j(z) = \sum_{i=0}^{j} \beta_{ji} z^{-i}$$

are generated for $j = N, N-1, \ldots, 0$, and for each value of j the multiplier constants ν_j and μ_j are evaluated using coefficients α_{jj} and β_{jj} in the above polynomials. The steps involved are detailed below.

Step 1: Let $N_j(z) = N(z)$ and $D_j(z) = D(z)$ and assume that $j = N$, that is

$$N_N(z) = \sum_{i=0}^{j} \alpha_{ji} z^{-i} = \sum_{i=0}^{N} a_i z^{-i} \tag{8.7a}$$

$$D_N(z) = \sum_{i=0}^{j} \beta_{ji} z^{-i} = \sum_{i=0}^{N} b_i z^{-i} \qquad \text{with } b_0 = 1 \tag{8.7b}$$



(*a*)

(*b*)

Figure 8.9 (*a*) General lattice configuration, (*b*) *j*th lattice section.

Step 2: Obtain ν_j, μ_j, $N_{j-1}(z)$, and $D_{j-1}(z)$ for $j = N, N-1, \ldots, 2$ using the following recursive relations:

$$\nu_j = \alpha_{jj} \qquad \mu_j = \beta_{jj} \tag{8.8a}$$

$$P_j(z) = D_j\left(\frac{1}{z}\right)z^{-j} = \sum_{i=0}^{j} \beta_{ji} z^{i-j} \tag{8.8b}$$

$$N_{j-1}(z) = N_j(z) - \nu_j P_j(z) = \sum_{i=0}^{j-1} \alpha_{ji} z^{-i} \tag{8.8c}$$

$$D_{j-1}(z) = \frac{D_j(z) - \mu_j P_j(z)}{1-\mu_j^2} = \sum_{i=0}^{j-1} \beta_{ji} z^{-i} \tag{8.8d}$$

Step 3: Let $j = 1$ in Eqs. (8.8a)–(8.8d) and obtain ν_1, μ_1, and $N_0(z)$ as follows:

$$\nu_1 = \alpha_{11} \qquad \mu_1 = \beta_{11} \tag{8.9a}$$

$$P_1(z) = D_1\left(\frac{1}{z}\right)z^{-1} = \beta_{10}z^{-1} + \beta_{11} \tag{8.9b}$$

$$N_0(z) = N_1(z) - \nu_1 P_1(z) = \alpha_{00} \tag{8.9c}$$

Step 4: Complete the realization by letting

$$\nu_0 = \alpha_{00}$$

The above lattice realization procedure is illustrated in the following example by obtaining a general second-order lattice structure.

Example 8.3 Realize the transfer function of Example 8.1 using the lattice method.

■ **Solution**

From Eqs. (8.7a) and (8.7b), we can write

$$N_2(z) = \alpha_{20} + \alpha_{21}z^{-1} + \alpha_{22}z^{-2} = a_0 + a_1z^{-1} + a_2z^{-2}$$
$$D_2(z) = \beta_{20} + \beta_{21}z^{-1} + \beta_{22}z^{-2} = 1 + b_1z^{-1} + b_2z^{-2}$$

For $j = 2$, Eqs. (8.8a)–(8.8d) yield

$$\nu_2 = \alpha_{22} = a_2 \qquad \mu_2 = \beta_{22} = b_2$$

$$P_2(z) = D_2\left(\frac{1}{z}\right)z^{-2} = z^{-2} + b_1z^{-1} + b_2 = \beta_{20}z^{-2} + \beta_{21}z^{-1} + \beta_{22}$$

$$N_1(z) = N_2(z) - \nu_2 P_2(z) = a_0 + a_1z^{-1} + a_2z^{-2} - \nu_2(z^{-2} + b_1z^{-1} + b_2)$$
$$= \alpha_{10} + \alpha_{11}z^{-1}$$

$$D_1(z) = \frac{D_2(z) - \mu_2 P_2(z)}{1-\mu_2^2} = \frac{1 + b_1z^{-1} + b_2z^{-2} - \mu_2(z^{-2} + b_1z^{-1} + b_2)}{1-\mu_2^2}$$
$$= \beta_{10} + \beta_{11}z^{-1}$$

where

$$\alpha_{10} = a_0 - a_2 b_2 \qquad \alpha_{11} = a_1 - a_2 b_1$$
$$\beta_{10} = 1 \qquad \beta_{11} = \frac{b_1}{1 + b_2}$$

Similarly, from Eqs. (8.9a)–(8.9c), we have

$$\nu_1 = \alpha_{11} = a_1 - a_2 b_1 \qquad \mu_1 = \beta_{11} = \frac{b_1}{1 + b_2}$$
$$P_1(z) = D_1\left(\frac{1}{z}\right) z^{-1} = \beta_{10} z^{-1} + \beta_{11}$$
$$N_0(z) = N_1(z) - \nu_1 P_1(z) = \alpha_{10} + \alpha_{11} z^{-1} - \nu_1(\beta_{10} z^{-1} + \beta_{11}) = \alpha_{00}$$

where

$$\alpha_{00} = (a_0 - a_2 b_2) - \frac{(a_1 - a_2 b_1) b_1}{1 + b_2}$$

and from step 4, we have

$$\nu_0 = \alpha_{00}$$

Summarizing, the multiplier constants for a general second-order lattice realization are as follows:

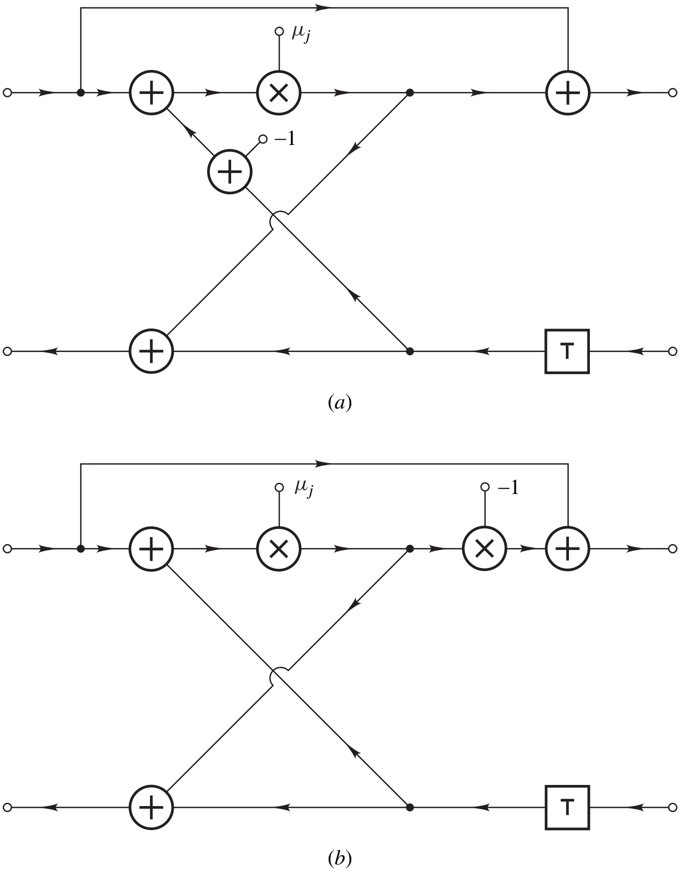
$$\nu_0 = (a_0 - a_2 b_2) - \frac{(a_1 - a_2 b_1) b_1}{1 + b_2}$$
$$\nu_1 = a_1 - a_2 b_1 \qquad \nu_2 = a_2$$
$$\mu_1 = \frac{b_1}{1 + b_2} \qquad \mu_2 = b_2 \quad \blacksquare$$

The 2-multiplier section of Fig. 8.9*b* yields structures that are canonic with respect to the number of unit delays. However, the number of multipliers can be quite large, as can be seen in Example 8.3. More economical realizations can be obtained by using *1-multiplier* first-order sections of the type shown in Fig. 8.10. Such realizations can be obtained by first realizing the transfer function in terms of 2-multiplier sections as described above and then replacing each of the 2-multiplier sections by either of the 1-multiplier sections of Fig. 8.10. The denominator multiplier constants $\mu_1, \mu_2, \ldots, \mu_N$ remain the same as before. However, the numerator multiplier constants $\nu_0, \nu_1, \ldots, \nu_N$ must be modified as

$$\tilde{\nu}_j = \frac{\nu_j}{\xi_j}$$

where

$$\xi_j = \begin{cases} 1 & \text{for } j = N \\ \prod_{i=j}^{N-1} (1 + \varepsilon_i \mu_{i+1}) & \text{for } j = 0, 1, \ldots, N-1 \end{cases}$$



(*a*)

(*b*)

Figure 8.10 1-multiplier section: (*a*) For case where $\varepsilon_i = +1$, (*b*) for case where $\varepsilon_i = -1$.

Each parameter ε_i is a constant which is equal to $+1$ or -1 depending on whether the ith 2-multiplier section is replaced by the 1-multiplier section of Fig. 8.10*a* or that of Fig. 8.10*b*. The choice between the two types of sections is, in theory, arbitrary; however, in practice, it can be used to improve the performance of the structure in some respect. For example, by choosing the types of sections such that the signal levels at the internal nodes of the filter are maximized, an improved signal-to-noise ratio can be achieved (see Ref. [4] and Chap. 14).

8.2.5 Cascade Realization

When the transfer function coefficients are quantized, errors are introduced in the amplitude and phase responses of the filter. It turns out that when a transfer function is realized directly in terms of a single Nth-order network using any one of the methods described so far, the sensitivity of the structure to coefficient quantization increases rapidly with N. Consequently, small errors introduced by coefficient quantization give rise to large errors in the amplitude and phase responses. This problem can to some extent be overcome by realizing high-order filters as interconnections of first- and second-order networks. In this and the next section, it is shown that an arbitrary transfer

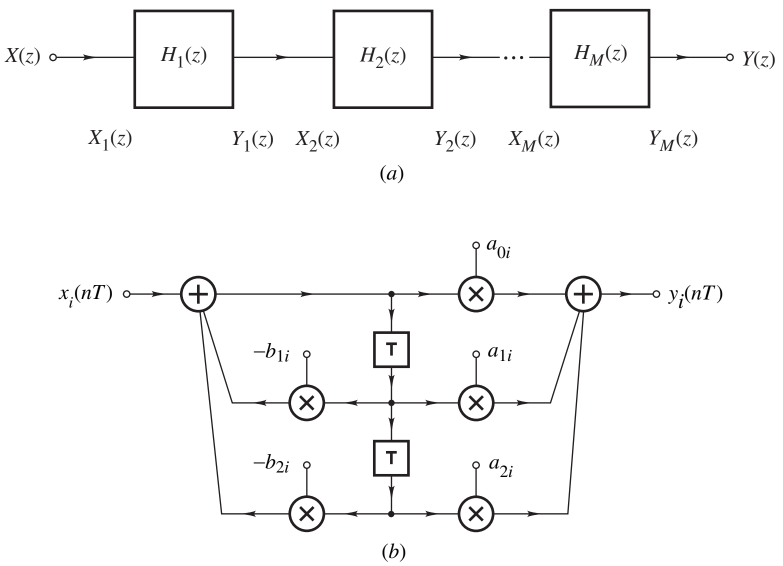


Figure 8.11 (a) Cascade realization of $H(z)$, (b) canonic second-order section.

function can be realized by connecting a number of first- and second-order structures in *cascade* or in *parallel*. Another approach to the reduction of coefficient quantization effects is to use the *wave* realization method, which is known to yield low-sensitivity structures. This possibility will be examined in Chap. 17.

Consider an arbitrary number of filter sections connected in cascade as shown in Fig. 8.11a and assume that the ith section is characterized by

$$Y_i(z) = H_i(z)X_i(z) \tag{8.10}$$

From Fig. 8.11a, we note that

$$\begin{aligned}
Y_1(z) &= H_1(z)X_1(z) = H_1(z)X(z) \\
Y_2(z) &= H_2(z)X_2(z) = H_2(z)Y_1(z) = H_1(z)H_2(z)X(z) \\
Y_3(z) &= H_3(z)X_3(z) = H_3(z)Y_2(z) = H_1(z)H_2(z)H_3(z)X(z) \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
Y(z) &= Y_M(z) = H_M(z)Y_{M-1}(z) = H_1(z)H_2(z)\cdots H_M(z)X(z)
\end{aligned}$$

Therefore, the overall transfer function of a cascade arrangement of filter sections is equal to the product of the individual transfer functions, that is,

$$H(z) = \prod_{i=1}^{M} H_i(z)$$

An Nth-order transfer function can be factorized into a product of first- and second-order transfer functions of the form

$$H_i(z) = \frac{a_{0i} + a_{1i}z^{-1}}{1 + b_{1i}z^{-1}} \tag{8.11a}$$

and

$$H_i(z) = \frac{a_{0i} + a_{1i}z^{-1} + a_{2i}z^{-2}}{1 + b_{1i}z^{-1} + b_{2i}z^{-2}} \tag{8.11b}$$

respectively. Now the individual first- and second-order transfer functions can be realized using any one of the methods described so far. Connecting the filter sections obtained in cascade would realize the required transfer function. For example, one could use the canonic section of Fig. 8.11*b* with $a_{2i} = b_{2i} = 0$ for a first-order transfer function to obtain a cascade canonic realization.

Example 8.4 Obtain a cascade realization of the transfer function

$$H(z) = \frac{216z^3 + 96z^2 + 24z}{(2z+1)(12z^2 + 7z + 1)}$$

using canonic sections.

■ **Solution**

The transfer function can be expressed as

$$H(z) = 9 \times \frac{z}{z + \frac{1}{2}} \times \frac{z^2 + \frac{4}{9}z + \frac{1}{9}}{z^2 + \frac{7}{12}z + \frac{1}{12}}$$

$$= 9 \times \frac{1}{1 + \frac{1}{2}z^{-1}} \times \frac{1 + \frac{4}{9}z^{-1} + \frac{1}{9}z^{-2}}{1 + \frac{7}{12}z^{-1} + \frac{1}{12}^{-2}}$$

Hence, the cascade canonic realization shown in Fig. 8.12 can be readily obtained. ■

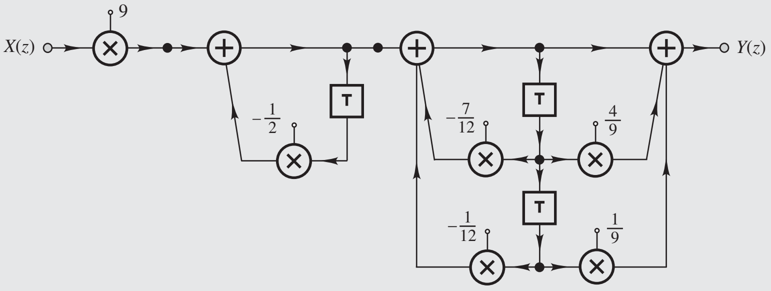


Figure 8.12 Cascade realization of $H(z)$ (Example 8.4).

8.2.6 Parallel Realization

Another realization comprising first- and second-order filter sections is based on the parallel configuration of Fig. 8.13. Assuming that the ith section in Fig. 8.13 can be represented by Eq. (8.10) and noting that $X_1(z) = X_2(z) = \cdots = X_M(z) = X(z)$, we can write

$$\begin{aligned}
Y(z) &= Y_1(z) + Y_2(z) + \cdots + Y_M(z) \\
&= H_1(z)X_1(z) + H_2(z)X_2(z) + \cdots + H_M(z)X_M(z) \\
&= H_1(z)X(z) + H_2(z)X(z) + \cdots + H_M(z)X(z) \\
&= [H_1(z) + H_2(z) + \cdots + H_M(z)]X(z) \\
&= H(z)X(z)
\end{aligned}$$

where

$$H(z) = \sum_{i=1}^{M} H_i(z)$$

Through the use of partial fractions, an N-order transfer function $H(z)$ can be expressed as a sum of first- and second-order transfer functions just like those in Eqs. (8.11a) and (8.11b). Connecting the sections obtained in parallel as in Fig. 8.13 would result in a parallel realization.

An alternative parallel realization can be readily obtained by expanding $H(z)/z$ instead of $H(z)$ into partial fractions.

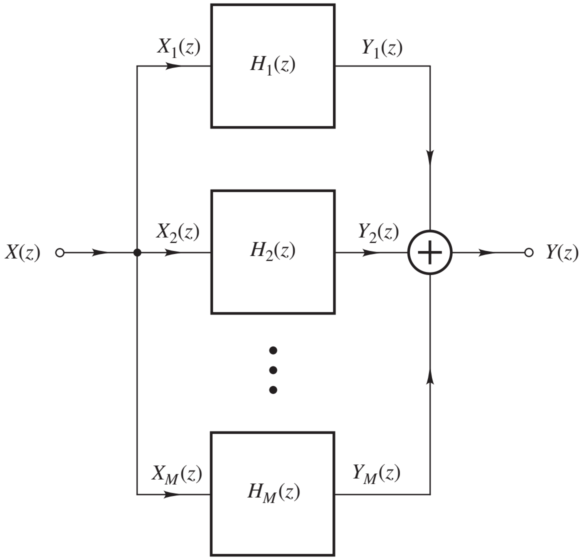


Figure 8.13 Parallel realization of $H(z)$.

Example 8.5 Obtain a parallel realization of the transfer function

$$H(z) = \frac{10z^4 - 3.7z^3 - 1.28z^2 + 0.99z}{(z^2 + z + 0.34)(z^2 + 0.9z + 0.2)}$$

using canonic sections.

■ **Solution**

We first need to find the poles of the transfer function. We have

$$H(z) = \frac{10z^4 - 3.7z^3 - 1.28z^2 + 0.99z}{(z - p_1)(z - p_2)(z - p_3)(z - p_4)}$$

where

$$p_1, p_2 = 0.5 \mp j0.3$$
$$p_3 = -0.4$$
$$p_4 = -0.5$$

If we expand $H(z)/z$ into partial fractions, we get

$$\frac{H(z)}{z} = \frac{R_1}{z - 0.5 + j0.3} + \frac{R_2}{z - 0.5 - j0.3} + \frac{R_3}{z + 0.4} + \frac{R_4}{z + 0.5}$$

The ith residue of $H(z)/z$ is given by

$$R_i = \left.\frac{(z - p_i)H(z)}{z}\right|_{z=p_i}$$

and through routine arithmetic or through the use of MATLAB, we get

$$\begin{aligned} R_1 &= \left.\frac{(10z^4 - 3.7z^3 - 1.28z^2 + 0.99z)}{z(z - p_2)(z - p_3)(z - p_4)}\right|_{z=p_1} \\ &= \left.\frac{(10z^3 - 3.7z^2 - 1.28z + 0.99)}{(z - p_2)(z - p_3)(z - p_4)}\right|_{z=p_1} \\ &= 1.0 \end{aligned}$$

Similarly,

$$R_2 = 1 \qquad R_3 = 3 \qquad R_4 = 5$$

and thus

$$H(z) = \frac{z}{z - 0.5 + j0.3} + \frac{z}{z - 0.5 - j0.3} + \frac{3z}{z + 0.4} + \frac{5z}{z + 0.5}$$

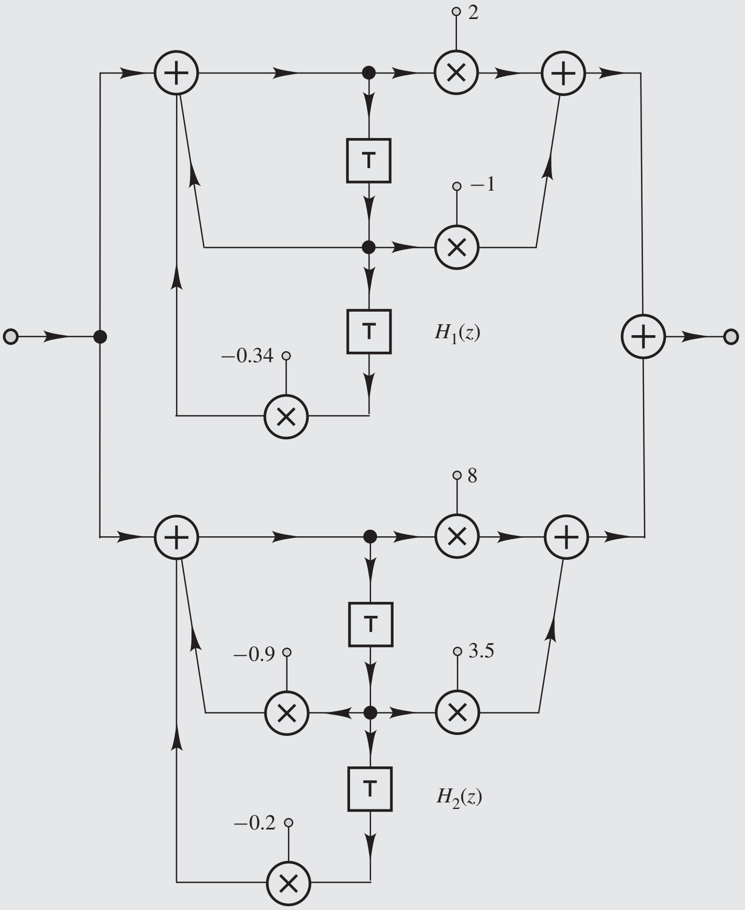


Figure 8.14 Parallel realization of $H(z)$ (Example 8.5).

Now, if we combine the first two and the last two partial fractions into second-order transfer functions, we get

$$H(z) = \frac{2z^2 - z}{z^2 - z + 0.34} + \frac{8z^2 + 3.5z}{z^2 + 0.9z + 0.2}$$

or

$$H(z) = H_1(z) + H_2(z)$$

where

$$H_1(z) = \frac{2 - z^{-1}}{1 - z^{-1} + 0.34z^{-2}} \qquad \text{and} \qquad H_2(z) = \frac{8 + 3.5z^{-1}}{1 + 0.9z^{-1} + 0.2z^{-2}}$$

Using canonic sections, the parallel realization shown in Fig. 8.14 can be obtained. ■

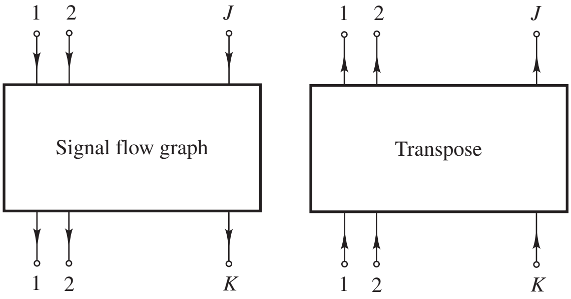


Figure 8.15 Transposition.

8.2.7 Transposition

Given a signal flow graph with inputs $j = 1, 2, \ldots, J$ and outputs $k = 1, 2, \ldots, K$, a corresponding signal flow graph can be derived by reversing the direction in each and every branch such that the J input nodes become output nodes and the K output nodes become input nodes, as illustrated in Fig. 8.15. The signal flow graph so derived, is said to be the *transpose* (or *adjoint*) of the original signal flow graph [9] (see also Chap. 4 of Ref. [10]). An interesting property of transposition is summarized in terms of the following theorem.

Theorem 8.1 Transposition ***If a signal flow graph and its transpose are characterized by transfer functions $H_{jk}(z)$ and $H_{kj}(z)$, respectively, then***

$$H_{jk}(z) = H_{kj}(z)$$

Proof See Ref. [9] or [10] for the proof. ▲

The transposition property can be used as a tool in the realization process since given an arbitrary digital network obtain through anyone of the realization procedures described in this chapter, an alternative realization can be derived through transposition.

Example 8.6 Obtain the transpose of the canonic network of Fig. 8.16*a*.

■ **Solution**

The signal flow graph of the canonic section of Fig. 8.16*a* can be readily obtained as shown in Fig. 8.16*b*. The transpose of the signal flow graph is shown in Fig. 8.16*c* and the transpose network is shown in Fig. 8.16*d*. ■

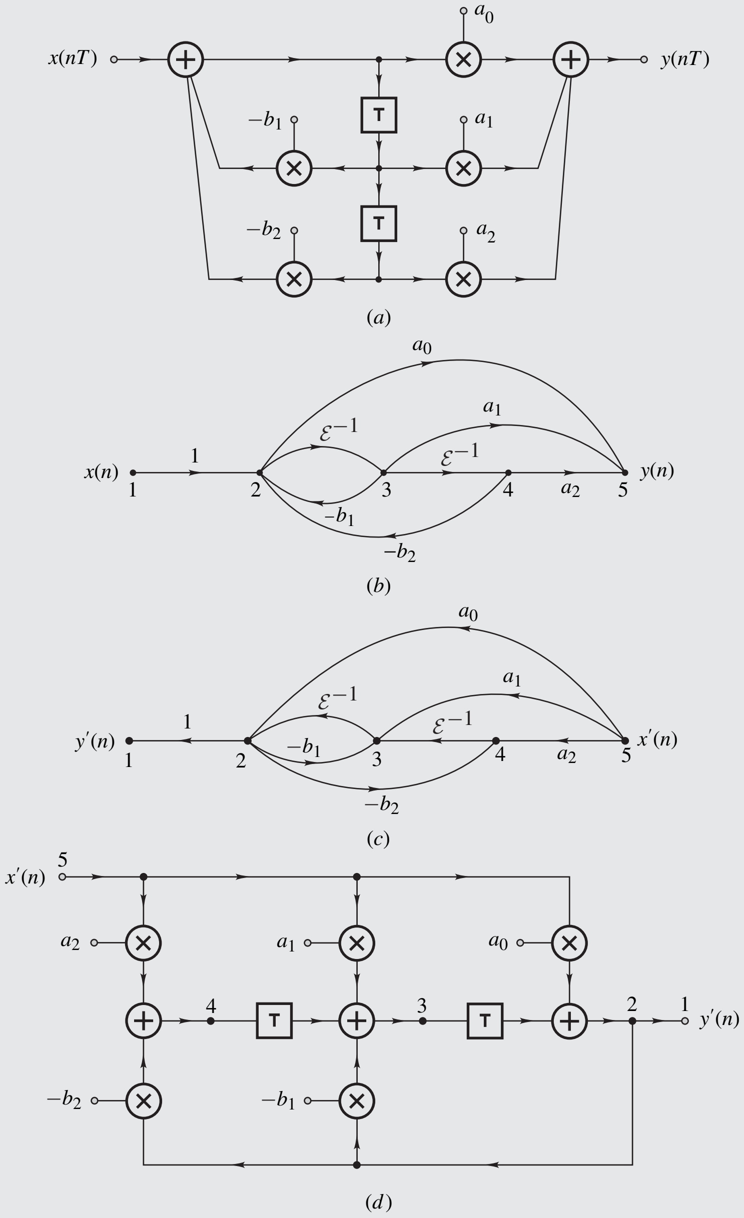


Figure 8.16 Transpose realization (Example 8.6): (*a*) Original realization, (*b*) signal flow graph of original realization, (*c*) transpose signal flow graph, (*d*) transpose realization.

8.3 IMPLEMENTATION

As was stated in Sec. 1.8, the *implementation* of digital filters can assume two forms, namely, software and hardware. This classification is somewhat artificial, however, since software and hardware are highly interchangeable nowadays. In nonreal-time applications, usually speed is not of considerable importance and the implementation might assume the form of a computer program on a general-purpose computer or DSP chip, which will emulate the operation of the digital filter. Such an implementation would be based on the difference equations characterizing one of the digital-filter structures described in the previous sections. On the other hand, if a digital filter is to be used in some communications system, speed is of the essence and the implementation would assume the form of a dedicated, highly specialized, piece of hardware. Depending on the application, a hardware implementation may comprise one or several interconnected VLSI circuit chips depending on the complexity of the required digital filter. Progress continues to be made in this technology in accordance with Moore's Law and as more and more functions can be accommodated on a VLSI chip, on the one hand, more complicated digital filters can be accommodated on a single chip and, on the other, fewer chips are needed to implement digital filters of high complexity.

8.3.1 Design Considerations

In practice, fabrication costs may be classified as recurring, e.g., the cost of parts, and nonrecurring, e.g., the design costs. For special-purpose systems like digital filters, demand is usually relatively small. Consequently, the design costs predominate over other costs and should be kept as low as possible. If the realization of the digital filter can be decomposed into a few types of basic building blocks that can be simply interconnected repetitively in a highly regular fashion, considerable savings in the design costs can be achieved. The reason is that the few types of building blocks need to be designed only once. A modular design of this type offers another advantage which can lead to cost reductions. By simply varying the number of modules used in a chip, a large selection of different digital filters can be easily designed that meet a variety of performance criteria or specifications. In this way, the nonrecurring design costs can be spread over a larger number of units fabricated and, therefore, the cost per unit can be reduced.

In certain real-time applications, high-order filters are required to operate at very high sampling rates. In such applications, a very large amount of computation needs to be carried out during each sampling period and the implementation must be very fast. While progress continues to be made in increasing the speed of gates and reducing the propagation delays by reducing the lengths of interconnection wires, progress is slowing down in these areas and the returns are slowly diminishing. Therefore, any major improvement in the speed of computation must of necessity be achieved through the concurrent use of many *processing elements*. It turns out that the degree of concurrency is an underlying property of the digital-filter realization. For example, realizations that comprise parallel substructures allow a high degree of concurrency and, therefore, lead to fast implementations. When a large number of processing elements must operate simultaneously, communication among processing elements becomes critical. Since the cost, performance, and speed of the chip depend heavily on the delay and area of the interconnection network, a high degree of concurrency should be achieved in conjunction with simple, short, and regular communication paths among processing elements.

8.3.2 Systolic Implementations

VLSI chip designers have been well aware of the merits of simplicity of form, regularity, and concurrency for a number of years and have developed special VLSI structures that offer many of these advantages. A family of such structures is the family of *systolic arrays* which are highly regular VLSI networks of simply connected processing elements that rhythmically process and pass data from one element to the next [11, 12]. The operation of these arrays is analogous to the rhythmical systolic operation of the heart and arteries by which blood is pumped forward from one artery to the next. Evidently, systolic realizations satisfy the design requirements alluded to earlier and are, as a consequence, highly suitable for the implementation of digital filters.

Close examination of the types of structures considered so far reveals that most of them are not suitable for systolic implementation. However, some of them can be made suitable by simple modifications, as will be demonstrated below. A useful technique in this process is known as *pipelining*. In this technique, the computation is partitioned into smaller parcels that can be assigned to a series of different concurrent processing elements in such a way as to achieve a speed advantage. A pipeline in the present context is, in a way, analogous to a modern assembly line of cars whereby the task of building a car is partitioned into a set of small subtasks carried out by concurrent workers (or robots) working at different stations along the assembly line. Pipelining will introduce some delay in the system, but once the pipeline is filled, a car will roll off the assembly line every few minutes. This sort of efficiency cannot be achieved by having all the workers working concurrently on one car for obvious reasons.

Consider the realization of

$$y(nT) = \sum_{i=0}^{N} a_i x(nT - iT)$$

shown in Fig. 8.17*a*, and assume that each addition and multiplication can be performed in τ_a and τ_m seconds, respectively. This structure can be readily obtained from Fig. 8.4*d*. Processing elements can be readily identified, as illustrated by the dashed lines. The additional unit delay at the right and the adder at the left with zero input are used as place holders in order to improve the regularity of the structure; they serve no other purpose. A basic disadvantage associated with this implementation is that the *processing rate*, which is the maximum sampling rate allowed by the structure, is limited. The processing rate of an implementation is the reciprocal of the time taken to perform all the required arithmetic operations between two successive samples. While the multiplications in Fig. 8.17*a* can be carried out concurrently, the $N + 1$ additions must be carried out sequentially from left to right. Therefore, a processing time of $\tau_m + (N + 1)\tau_a$ seconds is required, which can be large in practice since N can be large.

The processing rate in the structure of Fig. 8.17*a* can be increased by using faster adders. A more efficient approach, however, is to increase the degree of concurrency through the application of pipelining. Consider the possibility of adding unit delays between processing elements, as depicted in Fig. 8.17*b*. Since the top and bottom outputs of each processing element are delayed by the same amount by the additional unit delays, the two signals are not shifted relative to each other, and the operation of the structure is not destroyed. The only effect is that the overall output will be delayed by NT seconds, since there are N additional delays between processing elements. Indeed, straightforward analysis gives the output of the modified structure as

$$y_p(nT) = \sum_{i=0}^{N} a_i x(nT - iT - NT)$$

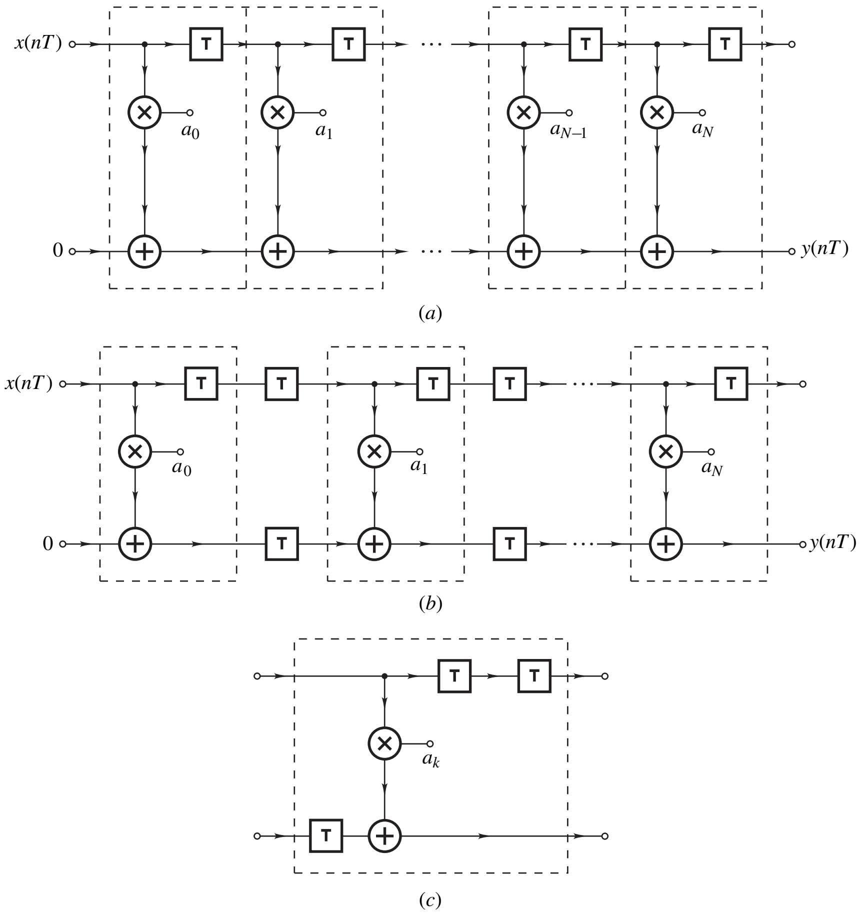


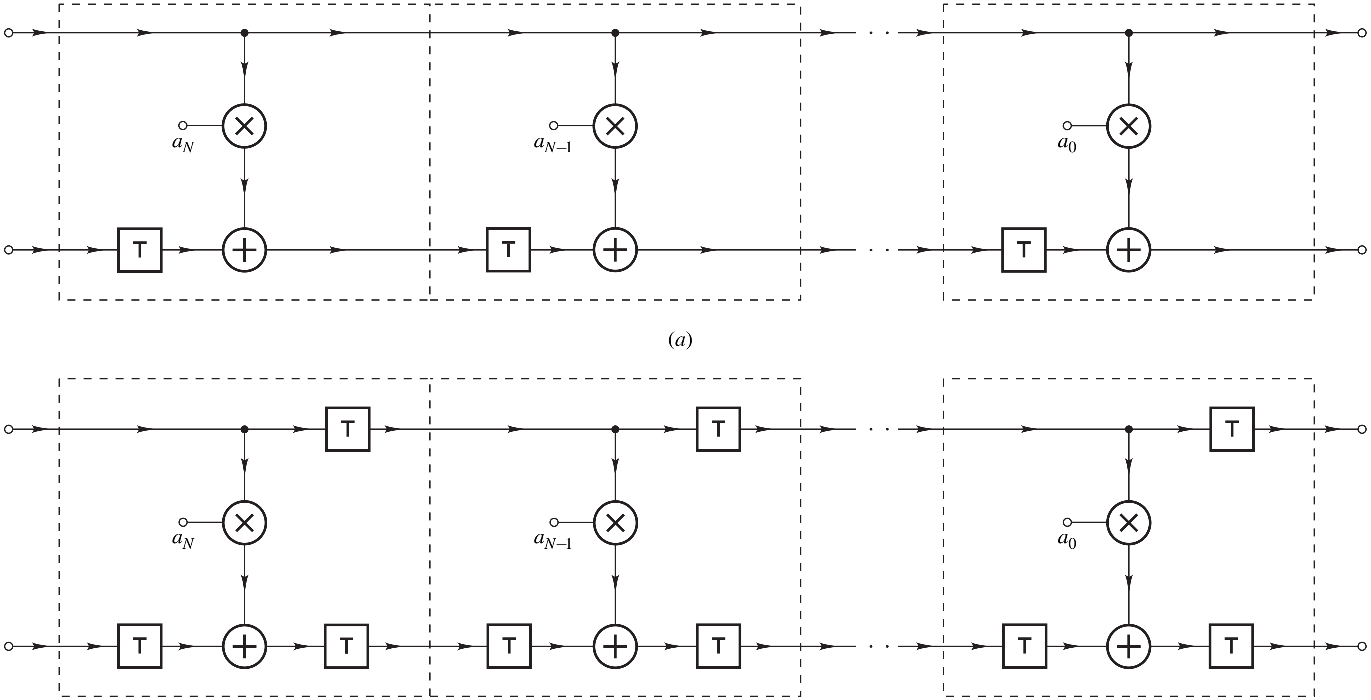
Figure 8.17 (*a*) Realization of *N*th-order nonrecursive filter, (*b*) corresponding systolic realization, (*c*) typical processing element.

that is,

$$y_p(nT) = y(nT - NT)$$

where $y(nT)$ is the output of the original structure. The delay NT is said to be the *latency* of the structure. In the modified structure, only one multiplication and one addition is required per digital-filter cycle and, therefore, the processing rate is $1/(\tau_m + \tau_a)$. In effect, the processing rate does not, in this case, decrease as the value of N is increased. The additional unit delays in Fig. 8.17*b* may be absorbed into the processing elements, as depicted in Fig. 8.17*c*.

An alternative structure that is amenable to a systolic implementation is depicted in Fig. 8.18*a*. This is obtained from the structure of Fig. 8.4*b*. As can be seen, only one multiplication and one



(*a*)

(*b*)

Figure 8.18 (*a*) Alternative realization of *N*th-order nonrecursive filter, (*b*) corresponding systolic realization.

addition are required per digital-filter cycle, and so the processing rate is $1/(\tau_m + \tau_a)$. The basic disadvantage of this structure is that the input signal has to be communicated directly to all the processing elements simultaneously. Consequently, for large values of N, wires become long and the associated propagation delays are large, thereby imposing an upper limit on the sampling rate. The problem can be easily overcome by using padding delays, as in Fig. 8.18*b*.

Example 8.7 A DSP chip that realizes the nonrecursive filter shown in Fig. 8.19 is readily available as an off-the-shelf component. The chip is fitted with registers for coefficients m_0 to m_3, which can accommodate arbitrary multiplier constants. Realize the transfer function

$$H(z) = \frac{216z^3 + 96z^2 + 24z + 2}{(2z + 1)(12z^2 + 7z + 1)}$$

using two of these DSP chips along with any necessary interfacing devices.

■ **Solution**

The transfer function can be expressed as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\frac{216}{24} + \frac{96}{24}z^{-1} + z^{-2} + \frac{2}{24}z^{-3}}{1 + \frac{26}{24}z^{-1} + \frac{9}{24}z^{-2} + \frac{1}{24}z^{-3}}$$

or as

$$Y(z) = \frac{N(z)}{1 + D'(z)}X(z)$$

where

$$N(z) = 9 + 4z^{-1} + z^{-2} + \tfrac{1}{12}z^{-3}$$

$$D'(z) = \tfrac{26}{24}z^{-1} + \tfrac{9}{24}z^{-2} + \tfrac{1}{24}z^{-3}$$

Hence

$$Y(z) = N(z)X(z) - Y(z)D'(z)$$

This equation can be realized using two nonrecursive filters with transfer functions $N(z)$ and $-D'(z)$ as shown in Fig. 8.1. $N(z)$ can be realized by the structure in Fig. 8.19 if $m_0 = 9$, $m_1 = 4$, $m_2 = 1$, $m_3 = 1/12$. On the other hand, $-D'(z)$ can be realized by the structure in Fig. 8.19 if $m_0 = 0$, $m_1 = -26/24$, $m_2 = -9/24$, and $m_3 = -1/24$. ■

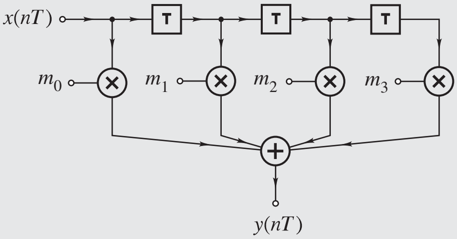


Figure 8.19 (*a*) Nonrecursive filter (Example 8.7).

REFERENCES

[1] B. Gold and C. M. Rader, *Digital Processing of Signals*, New York: McGraw-Hill, 1969.

[2] A. Antoniou, "Realization of digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 95–97, Mar. 1972.

[3] L. B. Jackson, A. G. Lindgren, and Y. Kim, "Synthesis of state-space digital filters with low roundoff noise and coefficient sensitivity," *in Proc. IEEE Int. Symp. Circuits and Systems, 1977*, pp. 41–44.

[4] A. H. Gray, Jr. and J. D. Markel, "Digital lattice and ladder filter synthesis," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 491–500, Dec. 1973.

[5] A. Fettweis, "Digital filter structures related to classical filter networks," *Arch. Elektron. Übertrag.*, vol. 25, pp. 79–89, 1971.

[6] A. Sedlmeyer and A. Fettweis, "Digital filters with true ladder configuration," *Int. J. Circuit Theory Appl.*, vol. 1, pp. 5–10, Mar. 1973.

[7] L. T. Bruton, "Low-sensitivity digital ladder filters," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 168–176, Mar. 1975.

[8] A. Antoniou and M. G. Rezk, "Digital-filter synthesis using concept of generalized-immittance convertor," *IEE J. Electron. Circuits Syst.*, vol. 1, pp. 207–216, Nov. 1977.

[9] A. Fettweis, "A general theorem for signal-flow networks with applications," *Arch. Elektron. Übertrag.*, vol. 25, pp. 557–561, 1971.

[10] A. Antoniou, *Digital Filters: Analysis, Design, and Applications*, New York: McGraw-Hill, 1993.

[11] H. T. Kung, "Why systolic architectures," *IEEE Computer*, vol. 15, pp. 37–46, Jan. 1982.

[12] S. Y. Kung, "VLSI array processors," *IEEE ASSP Magazine*, vol. 2, pp. 4–22, July 1985.

PROBLEMS

8.1. (*a*) Obtain the signal flow graph of the digital filter shown in Fig. P8.1.

(*b*) Deduce the transfer function of the filter using the node elimination method.

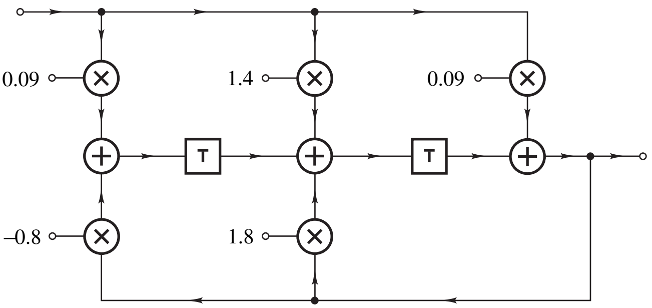


Figure P8.1

8.2. (*a*) The flow graph of Fig. P8.2*a* represents a recursive filter. Deduce the transfer function.
(*b*) Repeat part (*a*) for the flow graph of Fig. P8.2*b*.

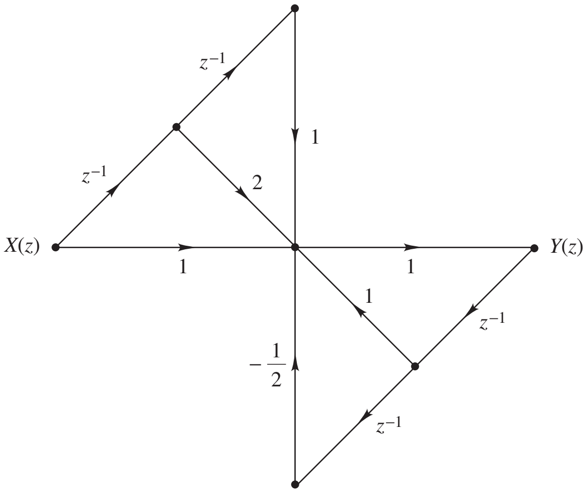


Figure P8.2*a*

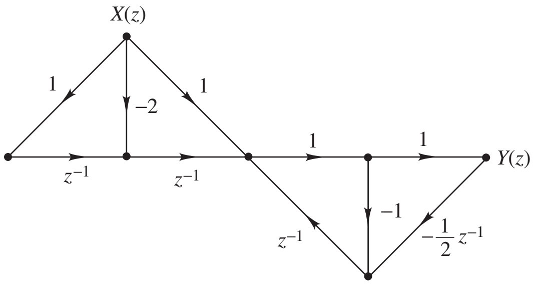


Figure P8.2*b*

8.3. (*a*) Convert the flow graph of Fig. P8.3 into a topologically equivalent network.
(*b*) Obtain an alternative realization by using the direct canonic method.

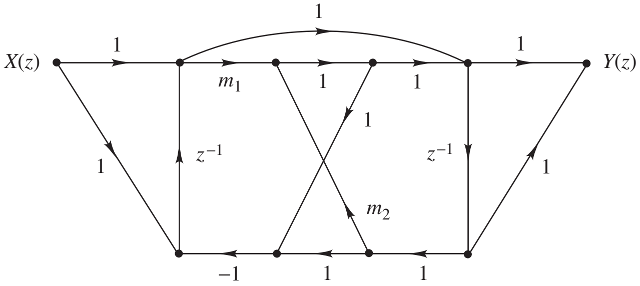


Figure P8.3

8.4. (*a*) Derive flow-graph representations for the filter of Fig. 8.5*b*.
(*b*) Repeat part (*a*) for the filter of Fig. 8.14.

8.5. A flow graph is said to be *computable* if there are no closed delay-free loops (see Sec. 4.8.1).
(*a*) Check the flow graphs of Fig. P8.5*a* for computability.
(*b*) Repeat part (*a*) for the filter of Fig. P8.5*b*.

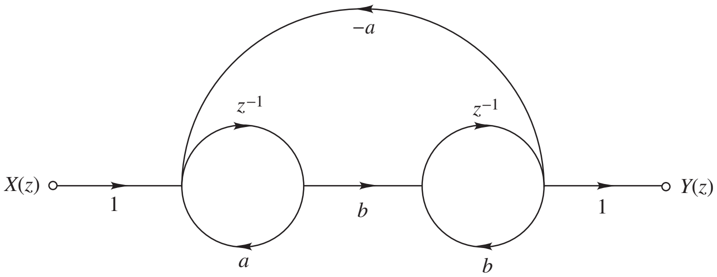


Figure P8.5*a*

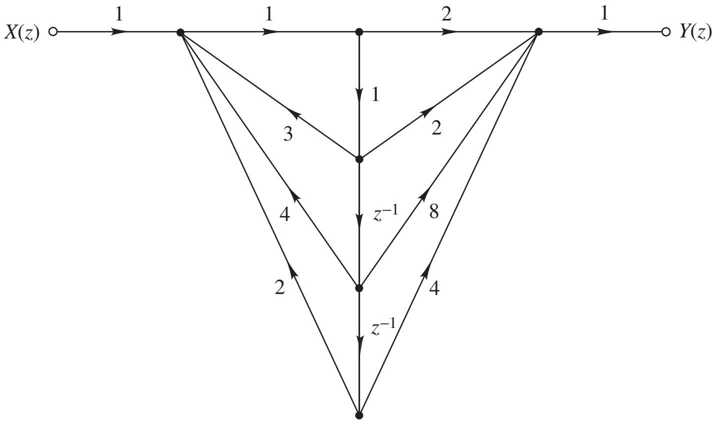


Figure P8.5*b*

8.6. By using first the direct and then the direct canonic method, realize the following transfer functions:

$$(a)\ H(z)=\frac{4(z-1)^4}{4z^4+3z^3+2z^2+z+1} \qquad (b)\ H(z)=\frac{(z+1)^2}{4z^3-2z^2+1}$$

8.7. A digital filter is characterized by the state-space equations

$$\begin{aligned}\mathbf{q}(nT+T) &= \mathbf{A}\mathbf{q}(nT)+\mathbf{b}x(nT)\\ y(nT) &= \mathbf{c}^T\mathbf{q}(nT)+dx(nT)\end{aligned}$$

where

$$\mathbf{A}=\begin{bmatrix}0 & 1\\ -\frac{5}{16} & -1\end{bmatrix} \qquad \mathbf{b}=\begin{bmatrix}0\\1\end{bmatrix} \qquad \mathbf{c}^T=\begin{bmatrix}-\frac{11}{8} & 2\end{bmatrix} \qquad d=2$$

(*a*) Obtain a state-space realization.
(*b*) Obtain a corresponding direct canonic realization.
(*c*) Compare the realizations in parts (*a*) and (*b*).

8.8. Repeat Prob. 8.7 if

$$\mathbf{A}=\begin{bmatrix}-0.1 & -0.5\\ 1.1 & -0.2\end{bmatrix} \qquad \mathbf{b}=\begin{bmatrix}0.7\\2.0\end{bmatrix} \qquad \mathbf{c}^T=\begin{bmatrix}8.8\\-0.6\end{bmatrix} \qquad d=8.0$$

8.9. Repeat Prob. 8.7 if

$$\mathbf{A}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ \frac{25}{64} & -\frac{29}{32} & \frac{3}{4}\end{bmatrix} \qquad \mathbf{b}=\begin{bmatrix}0\\0\\1\end{bmatrix} \qquad \mathbf{c}^T=\begin{bmatrix}\frac{25}{64} & \frac{3}{32} & \frac{11}{4}\end{bmatrix} \qquad d=1$$

8.10. (*a*) Realize the transfer function

$$H(z)=\frac{z(z+1)}{z^2-\frac{1}{2}z+\frac{1}{4}}$$

using a lattice structure.

(*b*) Repeat part (*a*) for the transfer function

$$H(z)=\frac{z^2+2z+1}{z^2+0.5z+0.3}$$

8.11. Realize the transfer function

$$H(z)=\frac{0.0154z^3+0.0462z^2+0.0462z+0.0154}{z^3-1.990z^2+1.572z-0.4582}$$

using the lattice method.

8.12. A recursive digital filter is characterized by the state-space equations in Prob. 8.7 with

$$\mathbf{A}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -\frac{1}{2} & -m & -2\end{bmatrix} \qquad \mathbf{b}=\begin{bmatrix}0\\0\\1\end{bmatrix} \qquad \mathbf{c}^T=\begin{bmatrix}1 & 2 & -1\end{bmatrix} \qquad d=1$$

(*a*) Determine the range of *m* for which the filter is stable.
(*b*) Obtain a state-space realization for the filter.

(*c*) Obtain a lattice realization.

(*d*) Compare the realizations in parts (*b*) and (*c*).

8.13. (*a*) Realize the transfer function

$$H(z) = \frac{6z}{(6z^3 + 6z^2 + 3z)(3z - 1)}$$

using direct canonic sections in cascade.

(*b*) Repeat part (*a*) using direct canonic sections in parallel.

8.14. (*a*) Realize the transfer function

$$H(z) = \frac{216z^3 + 168z^2 + 48z}{24(z + \frac{1}{2})(z^2 + \frac{1}{16})}$$

using low-order direct canonic sections in cascade.

(*b*) Repeat part (*a*) using direct canonic sections in parallel.

8.15. (*a*) Obtain a cascade realization for the transfer function

$$H(z) = \frac{4(z - 1)(z + 1)^2}{(2z + 1)(2z^2 - 2z + 1)}$$

using canonic sections.

(*b*) Obtain a parallel realization of the transfer function in part (*a*) using canonic sections.

8.16. (*a*) Obtain a cascade realization for the transfer function

$$H(z) = \frac{12z^3 + 6.4z^2 + 0.68z}{(z + 0.1)(z^2 + 0.8z + 0.15)}$$

using canonic sections.

(*b*) Obtain a parallel realization of the transfer function in part (*a*) using canonic sections.

8.17. (*a*) Obtain a realization of the transfer function

$$H(z) = \frac{96z^2 - 72z + 13}{24(z - \frac{1}{2})(z - \frac{1}{3})(z - \frac{1}{4})}$$

using a canonic first-order and a canonic second-order section in cascade.

(*b*) Obtain a parallel realization of the transfer function in part (*a*) using canonic first-order sections.

8.18. (*a*) Realize the transfer function

$$H(z) = \frac{16(z + 1)z^2}{(4z + 3)(4z^2 - 2z + 1)}$$

using canonic sections in cascade.

(*b*) Repeat part (*a*) using canonic sections in parallel.

8.19. First-order filter sections of the type depicted in Fig. P8.19 are available. Using sections of this type, obtain a parallel realization of the transfer function

$$H(z) = \frac{216z^2 + 162z + 29}{(2z + 1)(3z + 1)(4z + 1)}$$

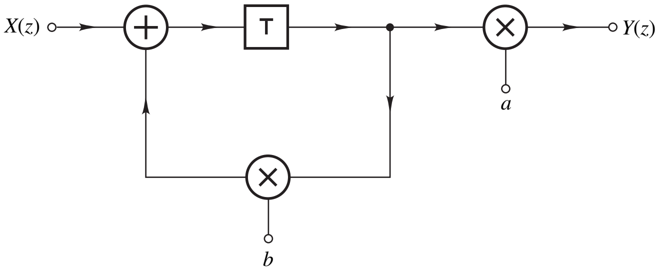


Figure P8.19

8.20. (*a*) Construct a flow chart for the software implementation of an N-section cascade filter assuming second-order filter sections.

(*b*) Write a computer program that will emulate the cascade filter in part (*a*).

8.21. (*a*) Construct a flow chart for the software implementation of an N-section parallel filter assuming second-order filter sections.

(*b*) Write a computer program that will emulate the parallel filter in part (*a*).

8.22. (*a*) Construct a flow chart for the software implementation of an Nth-order state-space filter.

(*b*) Write a computer program that will emulate the state-space filter of part (*a*).

8.23. (*a*) Construct a flow chart for the software implementation of an 2nd-order lattice filter.

(*b*) Write a computer program that will emulate the lattice filter of part (*a*).

8.24. (*a*) Obtain the transpose of the network of Fig. 5.2.

(*b*) Repeat part (*a*) for the network of Fig. 5.11.

8.25. (*a*) Obtain the transpose of the network shown in Fig. 8.5*b*.

(*b*) Repeat part (*a*) for the network of Fig. 8.14.

8.26. A digital-filter network that has a constant gain at all frequencies is said to be an *allpass network*.

(*a*) Show that the network depicted in Fig. P8.26 is an allpass network.

(*b*) Obtain an alternative allpass network using transposition.

(*c*) Show that the transpose network has the same transfer function as the original network.

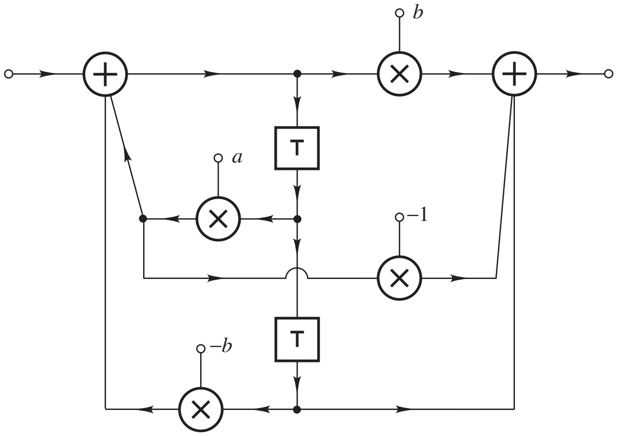


Figure P8.26

8.27. DSP VLSI chips that realize the module shown in Fig. P8.27*a* and the adder shown in Fig. P8.27*b* are readily available as off-the-shelf components. The chip in Fig. P8.27*a* is fitted with a register for coefficient a_k, which can accommodate an arbitrary multiplier constant. Using as many chips as necessary of the type shown in Fig. P8.27*a* plus an adder of the type shown in Fig. P8.27*b*, realize the transfer function

$$H(z) = \frac{1.1z^2 - 2.2z + 1.1}{z^2 - 0.4z + 0.3}$$

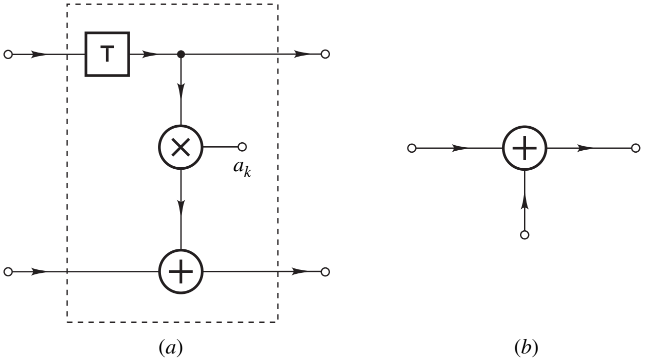


Figure P8.27

8.28. Realize the transfer function

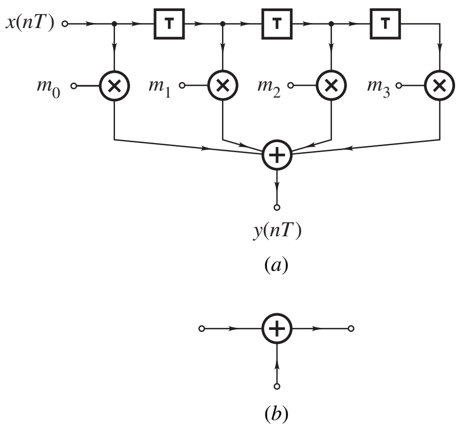
$$H(z) = \frac{z^2 - \frac{1}{2}z + \frac{1}{3}}{z^3 - \frac{1}{2}z^2 + \frac{1}{4}z + \frac{1}{8}}$$

using the VLSI chip of Prob. 8.27.

8.29. A DSP chip that realizes the nonrecursive filter shown in Fig. P8.29*a* is readily available as an off-the-shelf component. The chip is fitted with registers for coefficients m_0 to m_3, which can accommodate arbitrary multiplier constants. Realize the transfer function

$$H(z) = \frac{216z^3 + 96z^2 + 24z + 2}{(2z + 1)(12z^2 + 7z + 1)}$$

using two of these chips along with a 2-input adder such as that in Fig. P8.29*b*.



(a)

(b)

Figure P8.29

8.30. A DSP chip that realizes the recursive filter shown in Fig. 8.16*d* is readily available as an off-the-shelf component. The chip is fitted with registers that can accommodate the coefficients a_0, a_1, a_2, b_1, and b_2. Realize the transfer function

$$H(z) = \frac{48(z + 0.138)(z^2 + 0.312z + 0.0694)}{(2z + 1)(12z^2 + 7z + 1)}$$

using exactly two of these DSP chips, i.e., no other types of components are available. Show the configuration chosen and give suitable values to the various coefficients.

CHAPTER 9

DESIGN OF NONRECURSIVE (FIR) FILTERS

9.1 INTRODUCTION

The preceding chapter has dealt with the realization of digital filters whereby given an arbitrary transfer function or state-space characterization, a digital-filter network or structure is deduced. This and several of the subsequent chapters will deal with the approximation process whereby given some desirable filter characteristics or specifications, a suitable transfer function is derived.

As was mentioned in the introduction of Chap. 8, approximation methods can be classified as *direct* or *indirect*. In direct methods the discrete-time transfer function is generated directly in the z domain whereas in indirect methods it is derived from a continuous-time transfer function. Approximations can also be classified as *noniterative* or *iterative*. The former usually entail a set of formulas and transformations that yield designs of high precision with minimal computational effort. Iterative methods, on the other hand, are based on optimization algorithms. In these methods an initial design is assumed and is progressively improved until a discrete-time transfer function is obtained that satisfies the prerequisite specifications. These methods are very versatile and can, therefore, be used to obtain solutions to problems that are intractable with noniterative methods although they usually require a large amount of computation.

Approximation methods for the design of nonrecursive filters differ quite significantly from those used for the design of recursive filters. The basic reason for this is that in nonrecursive filters the transfer function is a polynomial in z^{-1} whereas in recursive filters it is ratio of polynomials in z.

Nonrecursive filters are designed by using direct noniterative or iterative methods whereas recursive filters are designed by using indirect noniterative methods or direct iterative methods.

The approximation problem for nonrecursive filters can be solved by applying the Fourier series or through the use of numerical analysis formulas. These methods provide *closed-form* solutions and, as a result, they are easy to apply and involve only a minimal amount of computation. Unfortunately, the designs obtained are suboptimal with respect to filter complexity whereby *a filter design is said to be optimal if the filter order is the lowest that can be achieved for the required specifications.* Another approach to the design of nonrecursive filters is to use a powerful multivariable optimization algorithm known as the *Remez exchange algorithm* as will be shown in Chap. 15. The Remez approach yields optimal designs but, unfortunately, a huge amount of computation is required to complete a design, which renders the approach unsuitable for applications where nonrecursive filters have to be designed in real or quasi-real time.

This chapter begins by examining the basic properties of nonrecursive filters. Then the use of the Fourier series as a tool in the design of nonrecursive filters is examined. It turns out that the use of the Fourier series by itself does not yield good designs but by applying the window technique described in Sec. 7.8 in conjunction with the Fourier series some moderately successful approximations can be obtained. The chapter concludes with the application of some classical numerical analysis formulas for the design of nonrecursive filters that can perform numerical interpolation, differentiation, or integration.

9.2 PROPERTIES OF CONSTANT-DELAY NONRECURSIVE FILTERS

Nonrecursive filters can be designed to have linear or nonlinear phase responses. However, linear-phase designs are typically preferred. In this section, it is shown that linear phase (or constant delay) can be achieved by ensuring that the impulse response has certain symmetries about its center point.

9.2.1 Impulse Response Symmetries

A nonrecursive causal filter of length N can be characterized by the transfer function

$$H(z) = \sum_{n=0}^{N-1} h(nT)z^{-n} \tag{9.1}$$

Its frequency response is given by

$$H(e^{j\omega T}) = M(\omega)e^{j\theta(\omega)} = \sum_{n=0}^{N-1} h(nT)e^{-j\omega nT} \tag{9.2}$$

where

$$M(\omega) = |H(e^{j\omega T})|$$

and

$$\theta(\omega) = \arg H(e^{j\omega T}) \tag{9.3}$$

The *phase* (or absolute) and *group* delays of a filter are given by

$$\tau_p = -\frac{\theta(\omega)}{\omega} \qquad \text{and} \qquad \tau_g = -\frac{d\theta(\omega)}{d\omega}$$

respectively (see Sec. 5.7).

For constant phase and group delays, the phase response must be linear, i.e.,

$$\theta(\omega) = -\tau\omega$$

and thus from Eqs. (9.2) and (9.3), we have

$$\theta(\omega) = -\tau\omega = \tan^{-1} \frac{-\sum_{n=0}^{N-1} h(nT)\sin\omega nT}{\sum_{n=0}^{N-1} h(nT)\cos\omega nT}$$

Consequently,

$$\tan\omega\tau = \frac{\sum_{n=0}^{N-1} h(nT)\sin\omega nT}{\sum_{n=0}^{N-1} h(nT)\cos\omega nT}$$

and accordingly

$$\sum_{n=0}^{N-1} h(nT)(\cos\omega nT \sin\omega\tau - \sin\omega nT \cos\omega\tau) = 0$$

or

$$\sum_{n=0}^{N-1} h(nT)\sin(\omega\tau - \omega nT) = 0$$

The solution of this equation is

$$\tau = \frac{(N-1)T}{2} \tag{9.4a}$$

$$h(nT) = h[(N-1-n)T] \qquad \text{for } 0 \le n \le N-1 \tag{9.4b}$$

as can be easily verified. Therefore, a nonrecursive filter can have constant phase and group delays *over the entire baseband*. It is only necessary for the impulse response to be *symmetrical* about the midpoint between samples $(N-2)/2$ and $N/2$ for even N or about sample $(N-1)/2$ for odd N. The required symmetry is illustrated in Fig. 9.1 for $N = 10$ and 11.

In contrast, recursive filters with constant phase or group delay are not easy to design as will be found out in Chaps. 11 and 12.

In most applications only the group delay needs to be constant in which case the phase response can have the form

$$\theta(\omega) = \theta_0 - \tau\omega$$

where θ_0 is a constant. On assuming that $\theta_0 = \pm\pi/2$, the above procedure yields a second class of constant-delay nonrecursive filters where

$$\tau = \frac{(N-1)T}{2} \tag{9.5a}$$

$$h(nT) = -h[(N-1-n)T] \tag{9.5b}$$

In this case, the impulse response is *antisymmetrical* about the midpoint between samples $(N-2)/2$ and $N/2$ for even N or about sample $(N-1)/2$ for odd N, as illustrated in Fig. 9.2.

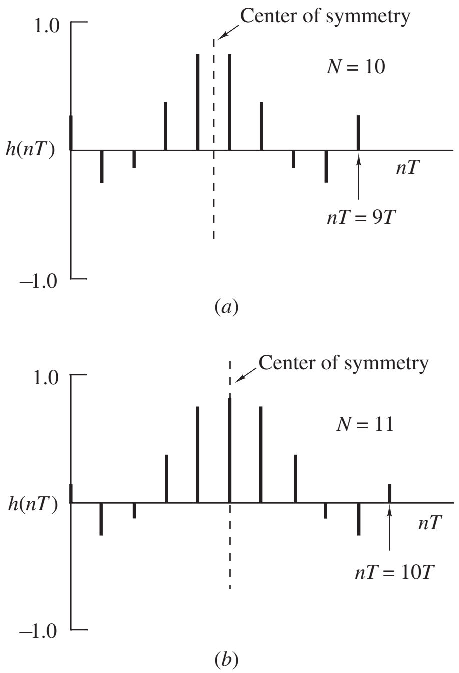


Figure 9.1 Impulse response for constant phase and group delays: (*a*) Even N, (*b*) odd N.

9.2.2 Frequency Response

The symmetries in the impulse response in Eqs. (9.4b) and (9.5b) lead to some simple expressions for the *frequency response* of nonrecursive filters as will now be demonstrated. For a *symmetrical* impulse response with N odd, Eq. (9.2) can be expressed as

$$H(e^{j\omega T}) = \sum_{n=0}^{(N-3)/2} h(nT)e^{-j\omega nT} + h\left[\frac{(N-1)T}{2}\right]e^{-j\omega(N-1)T/2} + \sum_{n=(N+1)/2}^{N-1} h(nT)e^{-j\omega nT} \qquad (9.6)$$

By using Eq. (9.4b) and then letting $N - 1 - n = m$ and $m = n$, the last summation in Eq. (9.6) can be expressed as

$$\begin{aligned}\sum_{n=(N+1)/2}^{N-1} h(nT)e^{-j\omega nT} &= \sum_{n=(N+1)/2}^{N-1} h[(N-1-n)T]e^{-j\omega nT}\\ &= \sum_{n=0}^{(N-3)/2} h(nT)e^{-j\omega(N-1-n)T} \qquad (9.7)\end{aligned}$$

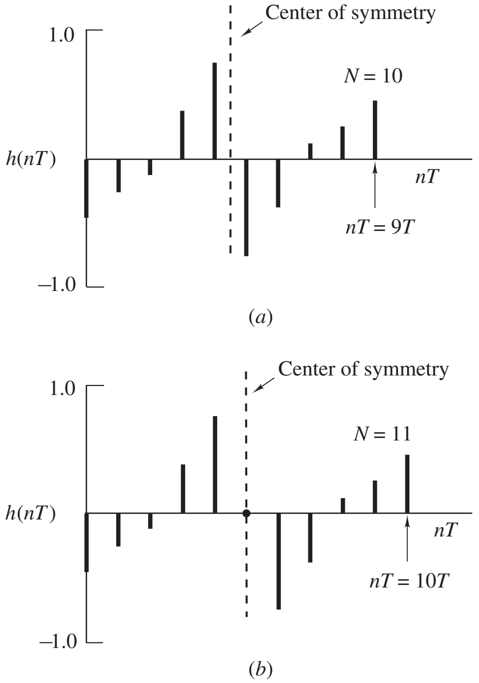


Figure 9.2 Alternative impulse response for constant group delay: (*a*) Even N, (*b*) odd N.

Now from Eqs. (9.6) and (9.7)

$$H(e^{j\omega T}) = e^{-j\omega(N-1)T/2}\left\{h\left[\frac{(N-1)T}{2}\right] + \sum_{n=0}^{(N-3)/2} 2h(nT)\cos\left[\omega\left(\frac{N-1}{2}-n\right)T\right]\right\}$$

and with $(N-1)/2 - n = k$, we have

$$H(e^{j\omega T}) = e^{-j\omega(N-1)T/2}\sum_{k=0}^{(N-1)/2} a_k \cos\omega kT$$

where
$$a_0 = h\left[\frac{(N-1)T}{2}\right] \tag{9.8a}$$

$$a_k = 2h\left[\left(\frac{N-1}{2}-k\right)T\right] \tag{9.8b}$$

Similarly, the frequency responses for the case of symmetrical impulse response with N even and for the two cases of *antisymmetrical* response simplify to the expressions summarized in Table 9.1.

Table 9.1 Frequency response of constant-delay nonrecursive filters

$h(nT)$	N	$H(e^{j\omega T})$
Symmetrical	Odd	$e^{-j\omega(N-1)T/2}\sum_{k=0}^{(N-1)/2} a_k \cos\omega kT$
	Even	$e^{-j\omega(N-1)T/2}\sum_{k=1}^{N/2} b_k \cos[\omega(k-\frac{1}{2})T]$
Antisymmetrical	Odd	$e^{-j[\omega(N-1)T/2-\pi/2]}\sum_{k=1}^{(N-1)/2} a_k \sin\omega kT$
	Even	$e^{-j[\omega(N-1)T/2-\pi/2]}\sum_{k=1}^{N/2} b_k \sin[\omega(k-\frac{1}{2})T]$
where $a_0 = h\left[\frac{(N-1)T}{2}\right]$	$a_k = 2h\left[\left(\frac{N-1}{2}-k\right)T\right]$	$b_k = 2h\left[\left(\frac{N}{2}-k\right)T\right]$

9.2.3 Location of Zeros

The impulse response constraints of Eqs. (9.4) and (9.5) impose certain restrictions on the zeros of $H(z)$. For odd N, Eqs. (9.1), (9.4b), and (9.5b) yield

$$H(z) = \frac{1}{z^{(N-1)/2}}\left\{\sum_{n=0}^{(N-3)/2} h(nT)\left(z^{(N-1)/2-n} \pm z^{-[(N-1)/2-n]}\right) + \frac{1}{2}h\left[\frac{(N-1)T}{2}\right](z^0 \pm z^0)\right\} \tag{9.9}$$

where the negative sign applies to the case of antisymmetrical impulse response. With $(N-1)/2-n = k$, Eq. (9.9) can be put in the form

$$H(z) = \frac{N(z)}{D(z)} = \frac{1}{z^{(N-1)/2}}\sum_{k=0}^{(N-1)/2}\frac{a_k}{2}(z^k \pm z^{-k})$$

where a_0 and a_k are given by Eqs. (9.8a) and (9.8b).

The zeros of $H(z)$ are the roots of

$$N(z) = \sum_{k=0}^{(N-1)/2} a_k(z^k \pm z^{-k})$$

If z is replaced by z^{-1} in $N(z)$, we have

$$N(z^{-1}) = \sum_{k=0}^{(N-1)/2} a_k(z^{-k} \pm z^k)$$

$$= \pm\sum_{k=0}^{(N-1)/2} a_k(z^k \pm z^{-k}) = \pm N(z)$$

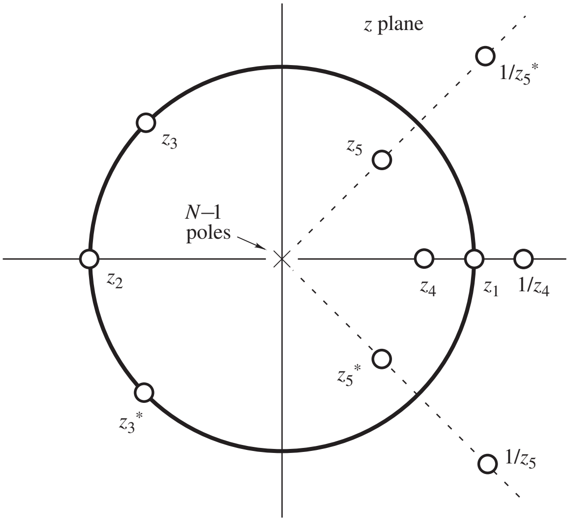


Figure 9.3 Typical zero-pole plot for a constant-delay nonrecursive filter.

The same relation holds for even N, as can be easily shown, and therefore if $z_i = r_i e^{j\psi_i}$ is a zero of $H(z)$, then $z_i^{-1} = e^{-j\psi_i}/r_i$ must also be a zero of $H(z)$. This has the following implications on the zero locations:

1. An arbitrary number of zeros can be located at $z_i = \pm 1$ since $z_i^{-1} = \pm 1$.
2. An arbitrary number of complex-conjugate pairs of zeros can be located on the unit circle since

$$(z - z_i)\left(z - z_i^*\right) = (z - e^{j\psi_i})(z - e^{-j\psi_i}) = \left(z - \frac{1}{z_i^*}\right)\left(z - \frac{1}{z_i}\right)$$

3. Real zeros off the unit circle must occur in reciprocal pairs.
4. Complex zeros off the unit circle must occur in groups of four, namely, z_i, z_i^*, and their reciprocals.

Polynomials with the above properties are often called *mirror-image polynomials*. A typical zero-pole plot for a constant-delay nonrecursive filter is shown in Fig. 9.3.

9.3 DESIGN USING THE FOURIER SERIES

Since the frequency response of a nonrecursive filter is a periodic function of ω with period ω_s, it can be expressed as a *Fourier series* (see Sec. 2.2). We can write

$$H(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} h(nT)e^{-j\omega nT} \tag{9.10}$$

where

$$h(nT) = \frac{1}{\omega_s}\int_{-\omega_s/2}^{\omega_s/2} H(e^{j\omega T})e^{j\omega nT}\, d\omega \tag{9.11}$$

and $\omega_s = 2\pi/T$. In Chap. 2, the Fourier series was applied for the time-domain representation of signals but in the present application it is applied for the frequency-domain representation of filters.

In effect, the roles of time and frequency are interchanged. If we let $e^{j\omega T} = z$ in Eq. (9.10)[1], we obtain

$$H(z) = \sum_{n=-\infty}^{\infty} h(nT)z^{-n} \tag{9.12}$$

Hence with an analytic representation for a required frequency response available, a corresponding transfer function can be readily derived. Unfortunately, however, this is *noncausal* and of *infinite order* since $h(nT)$ is defined over the range $-\infty < n < \infty$ according to Eq. (9.12). In order to achieve a finite-order transfer function, the series in Eq. (9.12) can be truncated by assigning

$$h(nT) = 0 \qquad \text{for } |n| > \frac{N-1}{2}$$

in which case

$$H(z) = h(0) + \sum_{n=1}^{(N-1)/2} [h(-nT)z^{n} + h(nT)z^{-n}] \tag{9.13}$$

Causality can be brought about by delaying the impulse response by $(N-1)T/2$ s, which translates into multiplying $H(z)$ by $z^{-(N-1)/2}$ by virtue of the time-shifting theorem of the z transform (Theorem 3.4), so that

$$H'(z) = z^{-(N-1)/2}H(z) \tag{9.14}$$

Since $|z^{-(N-1)/2}| = 1$ if $z = e^{j\omega T}$, the above modification does not change the amplitude response of the derived filter.

Note that if $H(e^{j\omega T})$ in Eq. (9.10) is an even function of ω, then the impulse response obtained is symmetrical about $n = 0$, and hence the filter has zero group delay. Consequently, the filter represented by the transfer function of Eq. (9.14) has constant group delay equal to $(N-1)T/2$.

The design approach just described is illustrated by the following example.

Example 9.1 Design a lowpass filter with a frequency response

$$H(e^{j\omega T}) \approx \begin{cases} 1 & \text{for } |\omega| \le \omega_c \\ 0 & \text{for } \omega_c < |\omega| \le \omega_s/2 \end{cases}$$

where ω_s is the sampling frequency.

■ **Solution**

From Eq. (9.11)

$$h(nT) = \frac{1}{\omega_s}\int_{-\omega_c}^{\omega_c} e^{j\omega nT}\,d\omega = \frac{1}{\omega_s}\left[\frac{e^{j\omega nT}}{jnT}\right]_{-\omega_c}^{\omega_c}$$

$$= \frac{1}{n\pi}\frac{(e^{j\omega_c nT} - e^{-j\omega_c nT})}{2j} = \frac{1}{n\pi}\sin\omega_c nT$$

[1] *This substitution is allowed by virtue of analytic continuation (see Sec. A.8).*

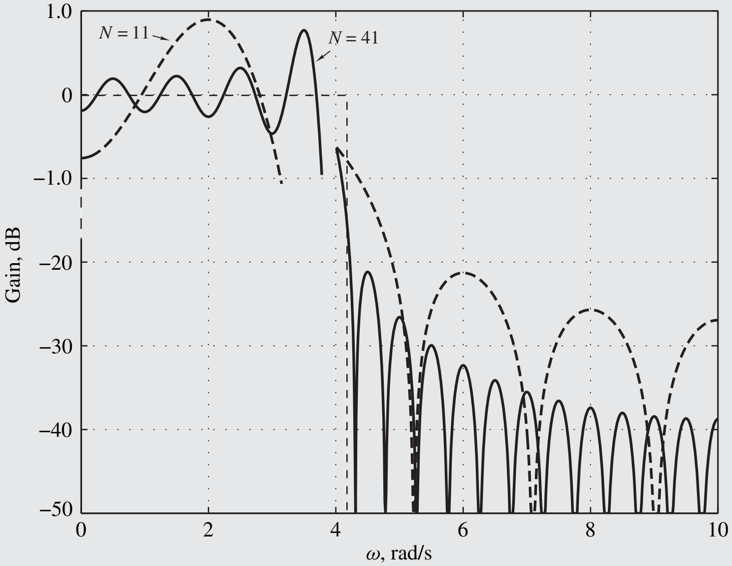


Figure 9.4 Amplitude response of lowpass filter (Example 9.1).

Hence Eqs. (9.13) and (9.14) yield

$$H(z) = z^{-(N-1)/2} \sum_{n=0}^{(N-1)/2} \frac{a_n}{2}(z^n + z^{-n}) \quad \blacksquare$$

where $$a_0 = h(0) \qquad a_n = 2h(nT)$$

The amplitude response of the lowpass filter obtained in Example 9.1 with ω_c and ω_s assumed to be 4 and 20 rad/s, respectively, is plotted in Fig. 9.4 for $N = 11$ and 41. The passband and stopband oscillations observed are due to *slow convergence* in the Fourier series, which in turn, is caused by the discontinuity at $\omega_c = 4$ rad/s. These are known as *Gibbs' oscillations*. As N is increased, the frequency of these oscillations is seen to increase, and at both low and high frequencies their amplitude is decreased. Also the transition between passband and stopband becomes steeper. However, the amplitudes of the passband and stopband ripples closest to the passband edge remain virtually unchanged as can be see in Fig. 9.4. Consequently, the quality of the filter obtained is not very good and ways must be found for the reduction of Gibbs' oscillations.

A rudimentary method is to avoid discontinuities in the idealized frequency response by introducing transition bands between passbands and stopbands [1]. For example, the response of the above lowpass filter could be redefined as

$$H(e^{j\omega T}) \approx \begin{cases} 1 & \text{for } |\omega| \le \omega_p \\ -\dfrac{\omega - \omega_a}{\omega_a - \omega_p} & \text{for } \omega_p < |\omega| < \omega_a \\ 0 & \text{for } \omega_a \le |\omega| \le \omega_s/2 \end{cases}$$

9.4 USE OF WINDOW FUNCTIONS

An alternative and easy-to-apply technique for the reduction of Gibbs' oscillations is to truncate the infinite-duration impulse response $h(nT)$ given by Eq. (9.11) through the use of a discrete-time window function $w(nT)$ such as those encountered in Sec. 7.8.2. If we let

$$h_w(nT) = w(nT)h(nT)$$

then the complex-convolution theorem (Theorem 3.10) gives

$$H_w(z) = \mathcal{Z}[w(nT)h(nT)] = \frac{1}{2\pi j} \oint_\Gamma H(v)W\left(\frac{z}{v}\right) v^{-1}\, dv \tag{9.15}$$

where Γ represents a contour in the common region of convergence of $H(v)$ and $W(z/v)$ and

$$H(z) = \mathcal{Z}h(nT) = \sum_{n=-\infty}^{\infty} h(nT)z^{-n} \tag{9.16a}$$

$$W(z) = \mathcal{Z}w(nT) = \sum_{n=-\infty}^{\infty} w(nT)z^{-n} \tag{9.16b}$$

If we let

$$v = e^{j\varpi T} \qquad \text{and} \qquad z = e^{j\omega T}$$

and assume that $H(v)$ and $W(z/v)$ converge on the unit circle of the v plane, Eq. (9.15) can be expressed as

$$H_w(e^{j\omega T}) = \frac{T}{2\pi} \int_0^{2\pi/T} H(e^{j\varpi T})W\left(e^{j(\omega-\varpi)T}\right) d\varpi \tag{9.17}$$

This is, of course, a convolution integral like the one in Eq. (7.20) and the effect of the window spectrum on the frequency response of the nonrecursive filter is very much analogous to the effect of the frequency spectrum of a nonperiodic continuous-time window on the frequency spectrum of the truncated continuous-time signal in Sec. 7.8.1.

Assuming that a lowpass filter with an idealized frequency response

$$H(e^{j\omega T}) = \begin{cases} 1 & \text{for } 0 \le |\omega| \le \omega_c \\ 0 & \text{for } \omega_c < |\omega| \le \omega_s/2 \end{cases}$$

is required, the graphical construction for the convolution integral assumes the form illustrated in Fig. 9.5. This is very similar to the graphical construction in Fig. 7.14 except for the fact that the frequency response of the filter and the frequency spectrum of the window function are periodic in the present application. As may be easily deduced following the steps in Sec. 7.8.1, the main-lobe width of the window will introduce transition bands at frequency points where the frequency response of the filter has discontinuities, i.e., at passband edges. On the other hand, the side ripples of the window will introduce ripples in the passband(s) of the filter whose amplitudes are directly related to the ripple ratio of the window function used.

A variety of window functions have been described in the literature in recent years and some of them are as follows [1, 2]:

1. Rectangular
2. von Hann[2]
3. Hamming
4. Blackman
5. Dolph-Chebyshev
6. Kaiser

The first four windows have only one adjustable parameter, the window length N. The last two, namely, the Dolph-Chebyshev and the Kaiser windows, have two parameters, the window length and one other parameter.

9.4.1 Rectangular Window

The *rectangular* window is given by

$$w_R(nT) = \begin{cases} 1 & \text{for } |n| \le \dfrac{N-1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{9.18}$$

and its frequency spectrum has been deduced in Sec. 7.8.2 as

$$W_R(e^{j\omega T}) = \frac{\sin(\omega NT/2)}{\sin(\omega T/2)} \tag{9.19}$$

Its main-lobe width is $2\omega_s/N$ and its ripple ratio remains relatively independent of N at approximately 22 percent for values of N in the range 11 to 101.

The rectangular window corresponds, of course, to the direct truncation of the Fourier series and the effect of direct truncation on $H(e^{j\omega T})$ is quite evident in Fig. 9.4. As N is increased the transition width between passband and stopband is decreased, an effect that is common in all windows. However, the amplitudes of the last passband and first stopband ripples remain virtually unchanged with increasing values of N, and this is a direct consequence of the fact that the ripple ratio of the rectangular window is virtually independent of N (see Fig. 7.10).

[2] *Due to Julius von Hann and often referred to inaccurately as the Hanning window function.*

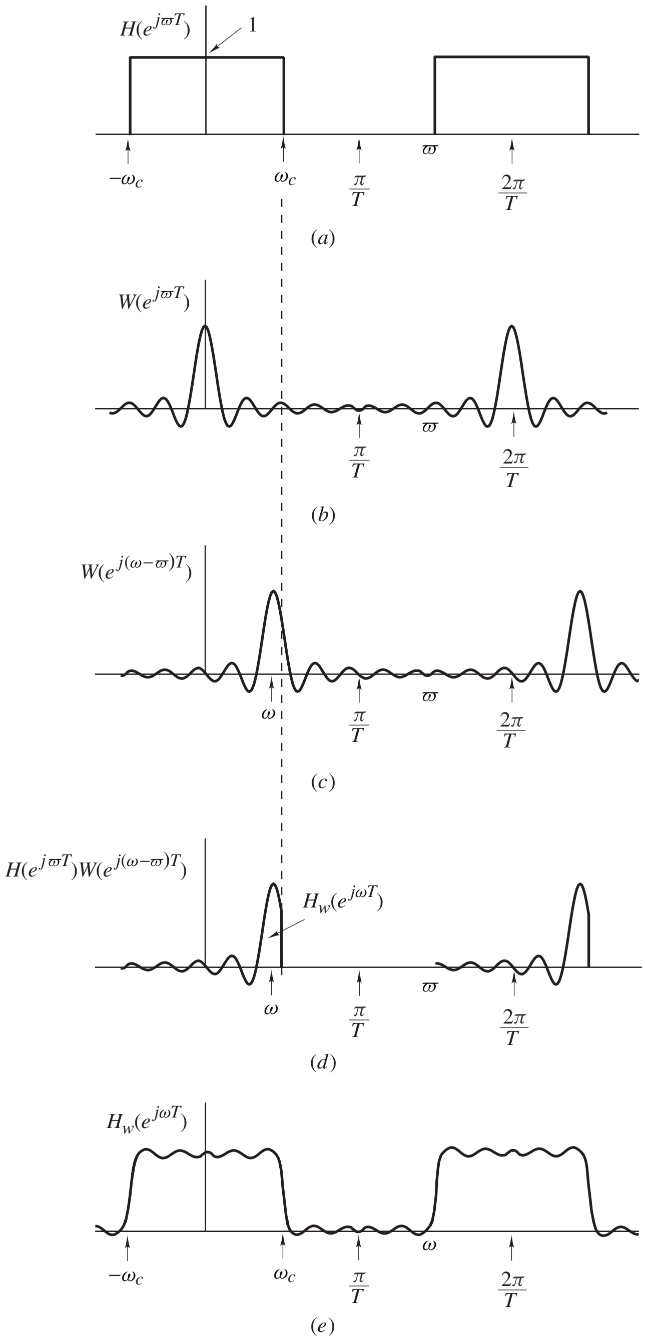


Figure 9.5 Convolution integral of Eq. (9.17).

9.4.2 von Hann and Hamming Windows

The *von Hann* and *Hamming* windows are essentially one and the same and are both given by the raised-cosine function

$$w_H(nT) = \begin{cases} \alpha + (1-\alpha)\cos\dfrac{2\pi n}{N-1} & \text{for } |n| \le \dfrac{N-1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{9.20}$$

where $\alpha = 0.5$ in the von Hann window and $\alpha = 0.54$ in the Hamming window. The small increase in the value of α from 0.5 to 0.54 in the latter window has a beneficial effect, namely, it reduces the ripple ratio by about 50 percent (see Table 9.2 below).

The spectrums of these windows can be related to that of the rectangular window. Equation (9.20) can be expressed as

$$\begin{aligned} w_H(nT) &= w_R(nT)\left[\alpha + (1-\alpha)\cos\frac{2\pi n}{N-1}\right] \\ &= \alpha w_R(nT) + \frac{1-\alpha}{2} w_R(nT)\left(e^{j2\pi n/(N-1)} + e^{-j2\pi n/(N-1)}\right) \end{aligned}$$

and on using the time-shifting theorem of the z transform (Theorem 3.4), we have

$$\begin{aligned} W_H(e^{j\omega T}) &= \mathcal{Z}[w_H(nT)]\Big|_{z=e^{j\omega T}} \\ &= \alpha W_R(e^{j\omega T}) + \frac{1-\alpha}{2} W_R\left(e^{j[\omega T - 2\pi/(N-1)]}\right) \\ &\quad + \frac{1-\alpha}{2} W_R\left(e^{j[\omega T + 2\pi/(N-1)]}\right) \end{aligned} \tag{9.21a}$$

Table 9.2 Summary of window parameters

Type of window	Main-lobe width	Ripple ratio, %		
		$N = 11$	$N = 21$	$N = 101$
Rectangular	$\dfrac{2\omega_s}{N}$	22.34	21.89	21.70
von Hann	$\dfrac{4\omega_s}{N}$	2.62	2.67	2.67
Hamming	$\dfrac{4\omega_s}{N}$	1.47	0.93	0.74
Blackman	$\dfrac{6\omega_s}{N}$	0.08	0.12	0.12

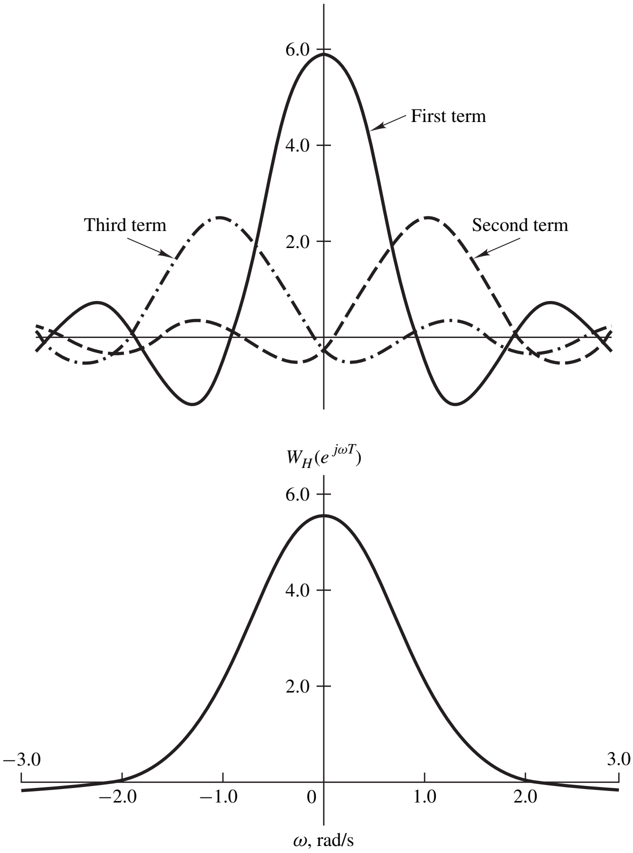


Figure 9.6 Spectrum of von Hann or Hamming window.

Now from Eqs. (9.19) and (9.21a), we get

$$W_H(e^{j\omega T}) = \frac{\alpha \sin(\omega NT/2)}{\sin(\omega T/2)} + \frac{1-\alpha}{2} \cdot \frac{\sin[\omega NT/2 - N\pi/(N-1)]}{\sin[\omega T/2 - \pi/(N-1)]}$$

$$+ \frac{1-\alpha}{2} \cdot \frac{\sin[\omega NT/2 + N\pi/(N-1)]}{\sin[\omega T/2 + \pi/(N-1)]} \tag{9.21b}$$

Consequently, the spectrums for the von Hann and Hamming windows can be formed by simply shifting $W_R(e^{j\omega T})$ first to the right and then to the left by $2\pi/(N-1)T$ and after that adding the three spectral components in Eq. (9.21b) as illustrated in Fig. 9.6. As can be observed, the second and third terms tend to cancel the first right and first left side lobes in $\alpha W_R(e^{j\omega T})$, and as a result both the von Hann and Hamming windows have reduced side lobe amplitudes compared with those of the rectangular window. For $N = 11$ and $\omega_s = 10$ rad/s, the ripple ratios for the two windows are 2.62 and 1.47 percent and change to 2.67 and 0.74 percent, respectively, for $N = 101$ (see Table 9.2).

The first term in Eq. (9.21b) is zero if

$$\omega = \frac{m\omega_s}{N}$$

and, similarly, the second and third terms are zero if

$$\omega = \left(m + \frac{N}{N-1}\right)\frac{\omega_s}{N} \qquad \text{and} \qquad \omega = \left(m - \frac{N}{N-1}\right)\frac{\omega_s}{N}$$

respectively, for $m = \pm 1, \pm 2, \ldots$. If $N \gg 1$, all three terms in Eq. (9.21b) have their first common zero at $|\omega| \approx 2\omega_s/N$, and hence the main-lobe width for the von Hann and Hamming windows is approximately $4\omega_s/N$.

9.4.3 Blackman Window

The *Blackman* window is similar to the preceding two and is given by

$$w_B(nT) = \begin{cases} 0.42 + 0.5\cos\dfrac{2\pi n}{N-1} + 0.08\cos\dfrac{4\pi n}{N-1} & \text{for } |n| \le \dfrac{N-1}{2} \\ 0 & \text{otherwise} \end{cases}$$

The additional cosine term leads to a further reduction in the amplitude of Gibbs' oscillations. The ripple ratio for $N = 11$ and $\omega_s = 10$ rad/s is 0.08 percent and changes to 0.12 percent for $N = 101$. The main-lobe width, however, is increased to about $6\omega_s/N$ (see Table 9.2).

As can be seen in Table 9.2, as the ripple ratio is decreased from one window to the next one, the main-lobe width is increased. This happens to be a fairly general trade-off among windows.

Example 9.2 Redesign the lowpass filter of Example 9.1 using the von Hann, Hamming, and Blackman windows.

■ **Solution**

The impulse response is the same as in Example 9.1, that is,

$$h(nT) = \frac{1}{n\pi}\sin\omega_c nT$$

On multiplying $h(nT)$ by the appropriate window function and then using Eqs. (9.15) and (9.14), in this order, we obtain

$$H'_w(z) = z^{-(N-1)/2}\sum_{n=0}^{(N-1)/2}\frac{a'_n}{2}(z^n + z^{-n}) \quad ■$$

where $\qquad a'_0 = w(0)h(0) \qquad$ and $\qquad a'_n = 2w(nT)h(nT)$

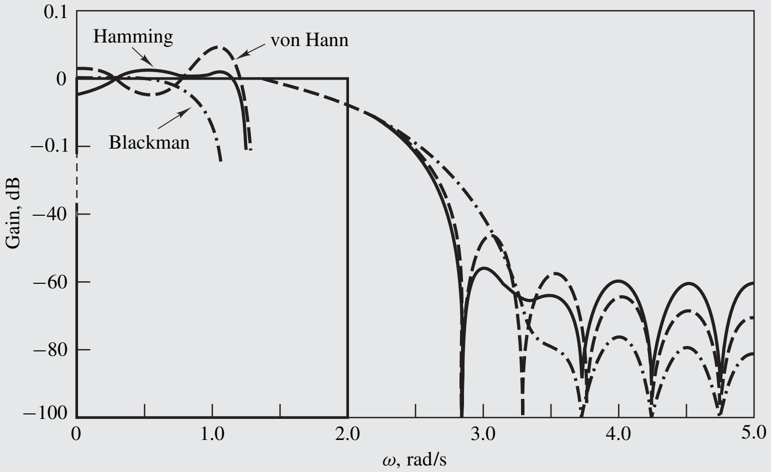


Figure 9.7 Amplitude response of lowpass filter (Example 9.2).

The amplitude responses of the three filters are given by

$$M(\omega) = \left| \sum_{n=0}^{(N-1)/2} a_n' \cos \omega n T \right|$$

These are plotted in Fig. 9.7 for $N = 21$ and $\omega_s = 10$ rad/s. As expected, the amplitude of the passband ripple is reduced, and the minimum stopband attenuation as well as the transition width are increased progressively from the von Hann to the Hamming to the Blackman window.

9.4.4 Dolph-Chebyshev Window

The windows considered so far have a ripple ratio which is practically independent of N, as can be seen in Table 9.2, and as a result the usefulness of these windows is limited. A more versatile window is the so-called *Dolph-Chebyshev* window [3]. This window is given by

$$w_{DC}(nT) = \frac{1}{N}\left[\frac{1}{r} + 2\sum_{i=1}^{(N-1)/2} T_{N-1}\left(x_0 \cos\frac{i\pi}{N}\right)\cos\frac{2n\pi i}{N}\right] \tag{9.22}$$

for $n = 0, 1, 2, \ldots, (N-1)/2$ where r is the required ripple ratio as a fraction and

$$x_0 = \cosh\left(\frac{1}{N-1}\cosh^{-1}\frac{1}{r}\right)$$

Function $T_k(x)$ is the kth-order Chebyshev polynomial associated with the Chebyshev approximation for recursive filters (see Sec. 10.4.1) and is given by

$$T_k(x) = \begin{cases} \cos(k\cos^{-1} x) & \text{for } |x| \le 1 \\ \cosh(\cosh^{-1} x) & \text{for } |x| > 1 \end{cases}$$

Evidently, an arbitrary ripple ratio can be achieved with this window and, as in other windows, the main-lobe width can be controlled by choosing the value of N.

The Dolph-Chebyshev window has two additional properties of interest. First, with N fixed, the main-lobe width is the *smallest* that can be achieved for a given ripple ratio; second, all the side lobes have the same amplitude, as can be seen in Fig. 9.8, that is, its amplitude spectrum is *equiripple*. A consequence of the first property is that filters designed by using this window have a narrow transition band. A consequence of the second property is that the approximation error tends to be somewhat more uniformly distributed with respect to frequency.

There is a practical issue in connection with most windows, including the Dolph-Chebyshev window, which needs to be addressed. We have assumed an ideal passband amplitude response of unity in the filters considered so far and typically the response of the designed filter is required to oscillate about unity. The value of the filter gain at any given frequency depends on the area of the window spectrum, as can be seen in Fig. 9.5, and if the passband gain is required to be approximately equal to unity, then the area of the window spectrum should be approximately equal to $2\pi/T$ to cancel out the factor $T/(2\pi)$ in the convolution integral of Eq. (9.17). For the Kaiser window, this turns out to be the case. However, in the case of the Dolph-Chebyshev window, the area of the window spectrum tends to

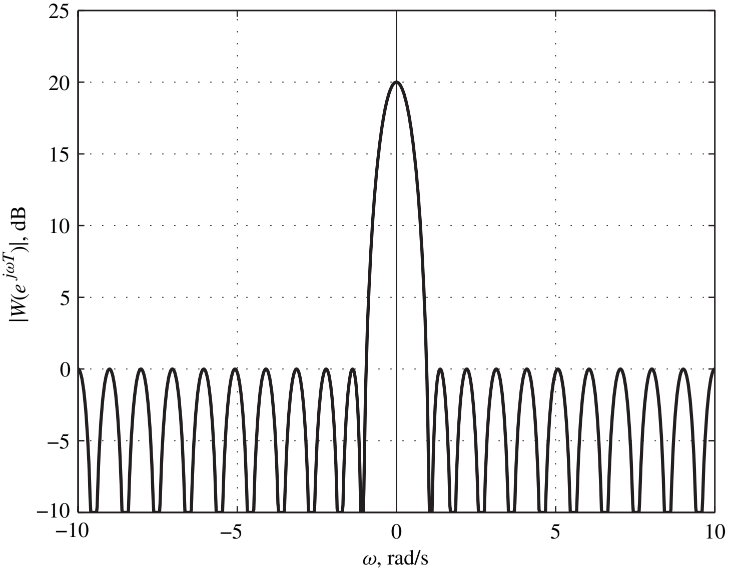


Figure 9.8 Amplitude spectrum for Dolph-Chebyshev window ($N = 21$, ripple ratio $= -20$ dB).

depend on the ripple ratio and, consequently, the passband gain will oscillate about some value other than unity. The problem can be easily circumvented by simply scaling the values of the impulse response by a suitable factor after the design is completed. This amounts to scaling the amplitude response by the same factor as can be readily verified. Depending on the application at hand, one may want the amplitude response to have a maximum value of unity (or 0 dB), or to oscillate about unity, or to do something else. In the first case, one would need to find the maximum value of the passband amplitude response as a ratio, say, $M_{\max}$ and then divide all the values of the modified impulse response by $M_{\max}$, that is,

$$h'_w(nT) = \frac{h_w(nT)}{M_{\max}} \qquad \text{for } -(N-1)/2 \le n \le (N-1)/2$$

On the other hand, if the passband amplitude response is required to oscillate about unity, then one would need to scale the impulse response values with respect to the average passband response by letting

$$h'_w(nT) = \frac{h_w(nT)}{M_{\text{AV}}} \qquad \text{where } M_{\text{AV}} = \tfrac{1}{2}(M_{\max} + M_{\min}) \tag{9.23}$$

and $M_{\min}$ is the minimum of the passband amplitude response. This scaling technique, which is also known as *normalization* of the amplitude response, is illustrated in the following example.

Example 9.3 (a) Using the Fourier-series method along with the Dolph-Chebyshev window, design a nonrecursive highpass filter assuming the idealized frequency response

$$H(e^{j\omega T}) = \begin{cases} 1 & \text{for } -\omega_s/2 < \omega < -\omega_c \\ 1 & \text{for } \omega_c < \omega < \omega_s/2 \\ 0 & \text{otherwise} \end{cases}$$

The required filter parameters are as follows:

- Ripple ratio: −20 dB
- ω_c : 6.0 rad/s
- ω_s : 20 rad/s
- $N = 21$

(b) Assuming that the passband of the filter extends from 6.8 to 10 rad/s, normalize the design obtained in part (a) so as to achieve an amplitude response that oscillates about unity. (c) Find the passband peak-to-peak ripple A_p in dB. (d) Assuming that the stopband extends from 0 to 5.5 rad/s, find the minimum stopband attenuation A_a in dB.

■ **Solution**

(a) From Eq. (9.11), we have

$$\begin{aligned}
h(nT) &= \frac{1}{\omega_s}\left[\int_{-\omega_s/2}^{-\omega_c} e^{j\omega nT}\,d\omega + \int_{\omega_c}^{\omega_s/2} e^{j\omega nT}\,d\omega\right]\\
&= \frac{1}{\omega_s}\left\{\left[\frac{e^{j\omega nT}}{jnT}\right]_{-\omega_s/2}^{-\omega_c} + \left[\frac{e^{j\omega nT}}{jnT}\right]_{\omega_c}^{\omega_s/2}\right\}\\
&= \frac{1}{n\pi}[\sin\omega_s nT/2 - \sin\omega_c nT]\\
&= \frac{1}{n\pi}[\sin n\pi - \sin\omega_c nT]
\end{aligned}$$

We note that $(\sin n\pi)/n\pi$ is always zero except for $n = 0$ and hence we get

$$h(nT) = \begin{cases} 1 - \frac{1}{n\pi}\sin\omega_c nT & \text{for } n = 0\\ -\frac{1}{n\pi}\sin\omega_c nT & \text{otherwise}\end{cases} \tag{9.24}$$

The ripple ratio in dB is given by $20\log r$ and hence

$$20\log r = -20 \qquad \text{or} \qquad r = 10^{-1} = 0.1$$

On using Eqs. (9.22) and (9.24), the design of Table 9.3 where $h_w(nT) = w_{DC}(nT)h(nT)$ can be obtained. ■

(b) The amplitude response can be computed by using the formula for a symmetrical impulse response of odd length in Table 9.1. Through a simple MATLAB m-file the maximum and minimum values of the passband amplitude response can be

Table 9.3 Numerical values of $h(nT)$ and $w_{DC}(nT)h(nT)$ (Example 9.3)

n	$h(nT) = h(-nT)$	$h_w(nT) = h_w(-nT)$
0	$4.000000E-1$	$2.343525E-1$
1	$-3.027307E-1$	$-1.758746E-1$
2	$9.354893E-2$	$5.298466E-2$
3	$6.236595E-2$	$3.384614E-2$
4	$-7.568267E-2$	$-3.865863E-2$
5	0.0	0.0
6	$5.045512E-2$	$2.155795E-2$
7	$-2.672827E-2$	$-1.012015E-2$
8	$-2.338723E-2$	$-7.661656E-3$
9	$3.363674E-2$	$9.278868E-3$
10	0.0	0.0

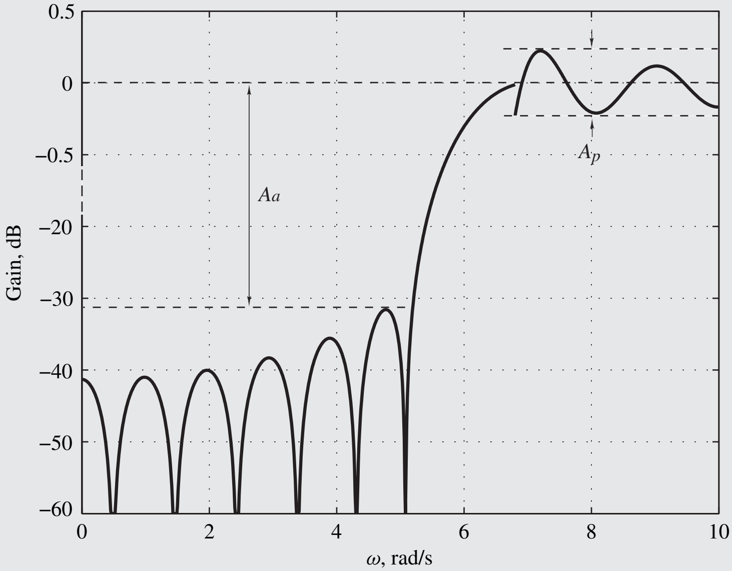


Figure 9.9 Amplitude response of lowpass filter (Example 9.3).

obtained as

$$M_{\max} = 0.6031 \qquad \text{and} \qquad M_{\min} = 0.5737$$

Hence the required scaling factor to normalize the passband amplitude response to unity is obtained from Eq. (9.23) as $M_{\mathrm{AV}} = 0.5884$. The amplitude response of the filter is plotted in Fig. 9.9. ■

(c) The peak-to-peak passband ripple in dB can be obtained as

$$A_p = 20\log M_{\max} - 20\log M_{\min} = 20\log\frac{M_{\max}}{M_{\min}} = 0.43 \text{ dB} \quad \blacksquare$$

(d) The minimum stopband attenuation A_a is defined as the negative of the maximum stopband gain and it can be computed as 0.01549. Hence we have

$$A_a = -20\log 0.01549 = 20\log\frac{1}{0.01549} = 36.2 \text{ dB} \quad \blacksquare$$

In view of the *equiripple* amplitude spectrum of the Dolph-Chebyshev window, one could expect to obtain an equiripple amplitude response for the filter. However, it does not work out that way because the relation between the amplitudes of the ripples in the filter response and those in the window spectrum is nonlinear. The nonlinear nature of this relation can be verified by examining the graphical construction of the convolution integral in Fig. 9.5.

9.4.5 Kaiser Window

The Kaiser window [4] and its properties have been described in Sec. 7.8.1. As will be shown below, this window can be used to design nonrecursive filters that satisfy prescribed specifications and it is, therefore, used widely. For this reason, its main characteristics are repeated here for easy reference. The window function is given by

$$w_K(nT) = \begin{cases} \dfrac{I_0(\beta)}{I_0(\alpha)} & \text{for } |n| \le \dfrac{N-1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{9.25}$$

where α is an independent parameter and

$$\beta = \alpha\sqrt{1 - \left(\frac{2n}{N-1}\right)^2} \qquad I_0(x) = 1 + \sum_{k=1}^{\infty}\left[\frac{1}{k!}\left(\frac{x}{2}\right)^k\right]^2$$

The exact spectrum of $w_K(nT)$ can be readily obtained from Eq. (9.16b) as

$$W_K(e^{j\omega T}) = w_K(0) + 2\sum_{n=1}^{(N-1)/2} w_K(nT)\cos\omega nT$$

and an approximate but closed-form formula was given in Sec. 7.8.2 (see Eq. (7.32)).

The ripple ratio can be varied continuously from the low value in the Blackman window to the high value in the rectangular window by simply varying the parameter α. Also, as in other windows, the main-lobe width, designated as B_m, can be adjusted by varying N. The influence of α on the ripple ratio and main-lobe width is illustrated in Fig. 7.17*a* and *b*.

An important advantage of the Kaiser window is that a method is available that can be used to design filters that will satisfy prescribed specifications [4]. The design method is based on the fact that while the ripple ratio affects both the passband ripple and the transition width between passband and stopband, the window length N affects only the transition width. Consequently, one can choose the α of the window through some empirical formulas to achieve the required passband or stopband ripple and then through another empirical formula one can choose the window length to achieve the desired transition width. The nuts and bolts of the method are as follows:

9.4.6 Prescribed Filter Specifications

In a filter designed through the use of the Kaiser window, the passband amplitude response oscillates between $1 - \delta$ and $1 + \delta$ whereas the stopband amplitude response oscillates between 0 and δ where δ is the amplitude of the largest passband ripple, which happens to be the same as the amplitude of the largest stopband ripple. Hence the vital characteristics of a lowpass filter can be completely specified as illustrated in Fig. 9.10*a* where 0 to ω_p and ω_a to $\omega_s/2$ define the passband and stopband, respectively. A prescribed set of specifications δ, ω_p, and ω_a can be achieved for some specified sampling frequency ω_s by choosing the parameter α and the length N of the Kaiser window such that the amplitude response never crosses into the shaded areas in Fig. 9.10*a*.

Typically in practice, the required filter characteristics are specified in terms of the peak-to-peak passband ripple A_p and the minimum stopband attenuation A_a in dB as defined in the solution

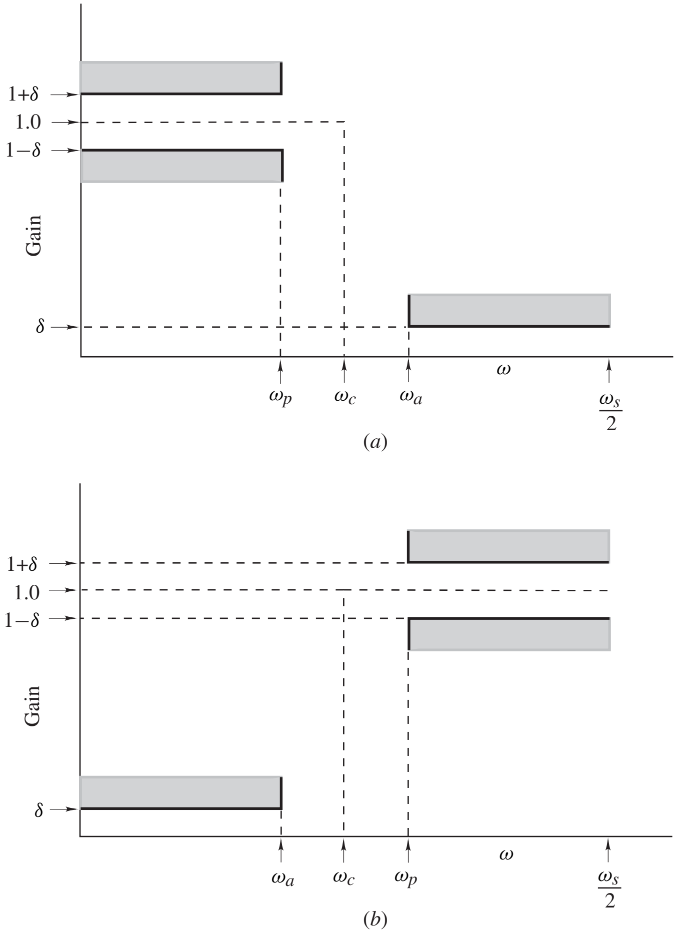


Figure 9.10 Idealized frequency responses: (*a*) Lowpass filter, (*b*) highpass filter.

of Example 9.3. For a lowpass filter specified by Fig. 9.10*a*, we have

$$A_p = 20 \log \frac{1+\delta}{1-\delta} \tag{9.26}$$

and

$$A_a = -20 \log \delta \tag{9.27}$$

respectively, and the transition width is given by

$$B_t = \omega_a - \omega_p$$

Given some arbitrary passband ripple and minimum stopband attenuation, say, $\tilde{A}_p$ and $\tilde{A}_a$, respectively, it may or may not be possible to achieve the required specifications exactly. If it is possible, that would be just fine. If it is not possible to get the exact specifications, the next best thing is to design a filter such that

$$A_p \le \tilde{A}_p \qquad \text{for } 0 \le \omega \le \omega_p$$

and

$$A_a \geq \tilde{A}_a \qquad \text{for } \omega_a \leq \omega \leq \omega_s/2$$

i.e., design a filter that would oversatisfy one or both specifications. This is a recurring theme in the design of filters both for nonrecursive as well as recursive.

A filter with a passband ripple equal to or less than $\tilde{A}_p$, a minimum stopband attenuation equal to or greater than $\tilde{A}_a$, and a transition width B_t can be readily designed by using the following procedure [4]:

1. Determine $h(nT)$ using the Fourier-series approach of Sec. 9.3 assuming an idealized frequency response

$$H(e^{j\omega T}) = \begin{cases} 1 & \text{for } |\omega| \leq \omega_c \\ 0 & \text{for } \omega_c < |\omega| \leq \omega_s/2 \end{cases}$$

(dashed line in Fig. 9.10*a*) where

$$\omega_c = \tfrac{1}{2}(\omega_p + \omega_a)$$

2. Choose δ in Eqs. (9.26) and (9.27) such that $A_p \leq \tilde{A}_p$ and $A_a \geq \tilde{A}_a$. A suitable value is

$$\delta = \min(\tilde{\delta}_p, \tilde{\delta}_a)$$

where $$\tilde{\delta}_p = \frac{10^{0.05\tilde{A}_p} - 1}{10^{0.05\tilde{A}_p} + 1} \qquad \text{and} \qquad \tilde{\delta}_a = 10^{-0.05\tilde{A}_a}$$

3. With the required δ defined, the actual stopband loss A_a in dB can be calculated using Eq. (9.27).
4. Choose parameter α as

$$\alpha = \begin{cases} 0 & \text{for } A_a \leq 21 \text{ dB} \\ 0.5842(A_a - 21)^{0.4} + 0.07886(A_a - 21) & \text{for } 21 < A_a \leq 50 \text{ dB} \\ 0.1102(A_a - 8.7) & \text{for } A_a > 50 \text{ dB} \end{cases}$$

5. Choose parameter D as

$$D = \begin{cases} 0.9222 & \text{for } A_a \leq 21 \text{ dB} \\ \dfrac{A_a - 7.95}{14.36} & \text{for } A_a > 21 \text{ dB} \end{cases}$$

Then select the lowest odd value of N that would satisfy the inequality

$$N \geq \frac{\omega_s D}{B_t} + 1$$

6. Form $w_K(nT)$ using Eq. (9.25).
7. Form

$$H'_w(z) = z^{-(N-1)/2} H_w(z) \qquad \text{where } H_w(z) = \mathcal{Z}[w_K(nT)h(nT)]$$

Example 9.4 Design a lowpass filter that would satisfy the following specifications:

- Maximum passband ripple in frequency range 0 to 1.5 rad/s: 0.1 dB
- Minimum stopband attenuation in frequency range 2.5 to 5.0 rad/s: 40 dB
- Sampling frequency: 10 rad/s

Solution

From step 1 and Example 9.1

$$h(nT) = \frac{1}{n\pi}\sin\omega_c nT \qquad \text{where } \omega_c = \tfrac{1}{2}(1.5+2.5) = 2.0 \text{ rad/s}$$

Step 2 gives

$$\tilde{\delta}_p = \frac{10^{0.05(0.1)}-1}{10^{0.05(0.1)}+1} = 5.7564\times10^{-3}$$

$$\tilde{\delta}_a = 10^{-0.05(40)} = 0.01$$

Hence

$$\delta = 5.7564\times10^{-3}$$

and from step 3

$$A_a = 44.797 \text{ dB}$$

Steps 4 and 5 yield

$$\alpha = 3.9524 \qquad D = 2.5660$$

Hence

$$N \geq \frac{10(2.566)}{1} + 1 = 26.66$$

or

$$N = 27$$

Finally steps 6 and 7 give

$$H'_w(z) = z^{-(N-1)/2}\sum_{n=0}^{(N-1)/2} h_w(nT)(z^n + z^{-n}) \quad \blacksquare$$

where

$$h_w(nT) = w_K(nT)h(nT)$$

The numerical values of $h(nT)$ and $w_K(nT)h(nT)$ are given in Table 9.4, and the amplitude response achieved is plotted in Fig. 9.11. This satisfies the prescribed specifications.

Table 9.4 Numerical values of $h(nT)$ and $w_K(nT)h(nT)$ (Example 9.4)

n	$h(nT) = h(-nT)$	$h_w(nT) = h_w(-nT)$
0	$4.000000E-1$	$4.000000E-1$
1	$3.027307E-1$	$2.996921E-1$
2	$9.354893E-2$	$8.983587E-2$
3	$-6.236595E-2$	$-5.690178E-2$
4	$-7.568267E-2$	$-6.420517E-2$
5	0.0	0.0
6	$5.045512E-2$	$3.450028E-2$
7	$2.672827E-2$	$1.577694E-2$
8	$-2.338723E-2$	$-1.155982E-2$
9	$-3.363674E-2$	$-1.343734E-2$
10	0.0	0.0
11	$2.752097E-2$	$6.235046E-3$
12	$1.559149E-2$	$2.395736E-3$
13	$-1.439214E-2$	$-1.326848E-3$

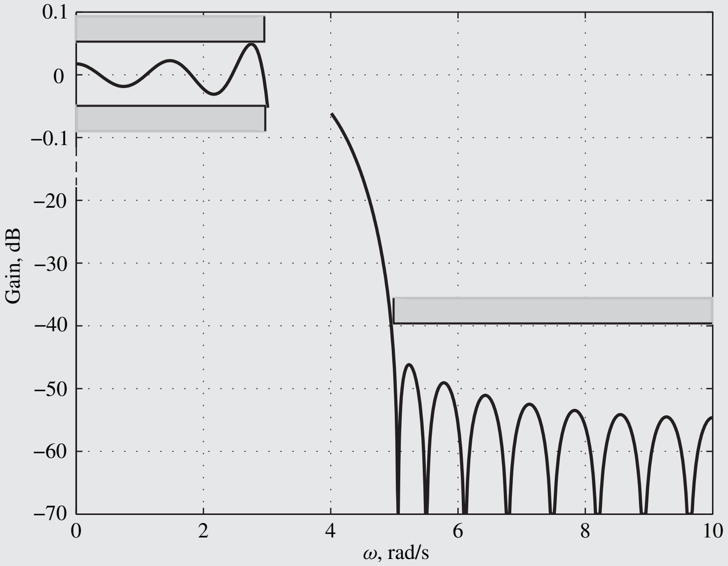


Figure 9.11 Amplitude response of lowpass filter (Example 9.4).

The above design procedure can be readily used for the design of *highpass* filters. For the specifications of Fig. 9.10*b*, the transition width and idealized frequency response in step 1 can be taken as

$$B_t = \omega_p - \omega_a \qquad \text{and} \qquad H(e^{j\omega T}) = \begin{cases} 1 & \text{for } -\omega_s/2 \le \omega \le -\omega_c \\ 1 & \text{for } \omega_c \le \omega \le \omega_s/2 \\ 0 & \text{otherwise} \end{cases}$$

where

$$\omega_c = \tfrac{1}{2}(\omega_a + \omega_p)$$

The remaining steps apply without modification.

The procedure can also be extended to the design of multiband filters such as bandpass and bandstop filters. This is possible on account of the fact that the amplitudes of the passband and stopband ripples and the transition widths between passbands and stopbands depend directly on the ripple ratio of the window and its length and are independent of the number of filter bands. Thus all one needs to do for a multiband filter is to design the filter on the basis of the narrowest transition width.

For the *bandpass* specifications of Fig. 9.12*a*, the design must be based on the narrower of the two transition bands, i.e.,

$$B_t = \min[(\omega_{p1} - \omega_{a1}),\ (\omega_{a2} - \omega_{p2})] \tag{9.28}$$

Hence

$$H(e^{j\omega T}) = \begin{cases} 1 & \text{for } -\omega_{c2} \le \omega \le -\omega_{c1} \\ 1 & \text{for } \omega_{c1} \le \omega \le \omega_{c2} \\ 0 & \text{otherwise} \end{cases} \tag{9.29}$$

where

$$\omega_{c1} = \omega_{p1} - \frac{B_t}{2} \qquad \omega_{c2} = \omega_{p2} + \frac{B_t}{2} \tag{9.30}$$

Similarly, for the *bandstop* specifications of Fig. 9.12*b*, we let

$$B_t = \min\,[(\omega_{a1} - \omega_{p1}), (\omega_{p2} - \omega_{a2})]$$

and

$$H(e^{j\omega T}) = \begin{cases} 1 & \text{for } 0 \le |\omega| \le \omega_{c1} \\ 0 & \text{for } \omega_{c1} < |\omega| < \omega_{c2} \\ 1 & \text{for } \omega_{c2} \le |\omega| \le \omega_s/2 \end{cases}$$

where

$$\omega_{c1} = \omega_{p1} + \frac{B_t}{2} \qquad \omega_{c2} = \omega_{p2} - \frac{B_t}{2}$$

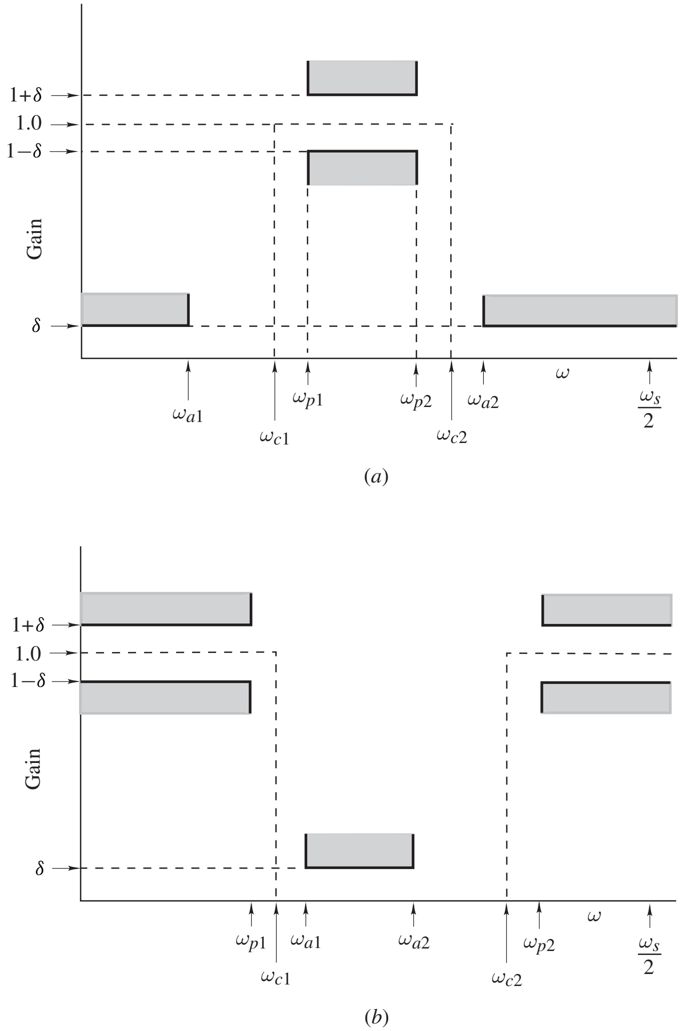


Figure 9.12 Idealized frequency responses: (*a*) Bandpass filter, (*b*) bandstop filter.

Example 9.5 Design a bandpass filter that would satisfy the following specifications:

- Minimum attenuation for $0 \leq \omega \leq 200$: 45 dB
- Maximum passband ripple for $400 < \omega < 600$: 0.2 dB
- Minimum attenuation for $700 \leq \omega \leq 1000$: 45 dB
- Sampling frequency: 2000 rad/s

■ **Solution**

From Eq. (9.28)

$$B_t = \min\,[(400-200),\ (700-600)] = 100$$

Hence from Eq. (9.30)

$$\omega_{c1} = 400 - 50 = 350 \text{ rad/s} \qquad \omega_{c2} = 600 + 50 = 650 \text{ rad/s}$$

Step 1 of the design procedure yields

$$h(nT) = \frac{1}{\omega_s}\int_{-\omega_s/2}^{\omega_s/2} H(e^{j\omega T})e^{j\omega nT}\,d\omega$$

and from Eq. (9.29), we get

$$\begin{aligned}
h(nT) &= \frac{1}{\omega_s}\left[\int_{-\omega_{c2}}^{-\omega_{c1}} e^{j\omega nT}\,d\omega + \int_{\omega_{c1}}^{\omega_{c2}} e^{j\omega nT}\,d\omega\right]\\
&= \frac{1}{\omega_s}\left[\frac{e^{-j\omega_{c1}nT}}{jnT} - \frac{e^{-j\omega_{c2}nT}}{jnT} + \frac{e^{j\omega_{c2}nT}}{jnT} - \frac{e^{j\omega_{c1}nT}}{jnT}\right]\\
&= \frac{1}{n\pi}\left[\frac{e^{j\omega_{c2}nT} - e^{-j\omega_{c2}nT}}{2j} - \frac{e^{j\omega_{c1}nT} - e^{-j\omega_{c2}nT}}{2j}\right]\\
h(nT) &= \frac{1}{n\pi}(\sin\omega_{c2}nT - \sin\omega_{c1}nT)
\end{aligned}$$

Now according to step 2,

$$\tilde{\delta}_p = \frac{10^{0.05(0.2)}-1}{10^{0.05(0.2)}+1} = 1.1512 \times 10^{-2}$$

$$\tilde{\delta}_a = 10^{-0.05(45)} = 5.6234 \times 10^{-3}$$

and

$$\delta = 5.6234 \times 10^{-3}$$

Thus from Eq. (9.27), we obtain

$$A_a = 45 \text{ dB}$$

The design can be completed as in Example 9.4. The resulting values for α, D, and N are

$$\alpha = 3.9754 \qquad D = 2.580 \qquad \text{and} \qquad N = 53 \quad ■$$

The amplitude response achieved is plotted in Fig. 9.13.

Note that if we let $\omega_{c1} = 0$ and $\omega_{c2} = \omega_c$ or $\omega_{c1} = \omega_c$ and $\omega_{c2} = \omega_s/2$ in the above expression for $h(nT)$, we get the impulse response for a lowpass or highpass filter, as may be expected (see Examples 9.1 and 9.3). Thus a computer program that can design bandpass filters can also be used to design lowpass and highpass filters.

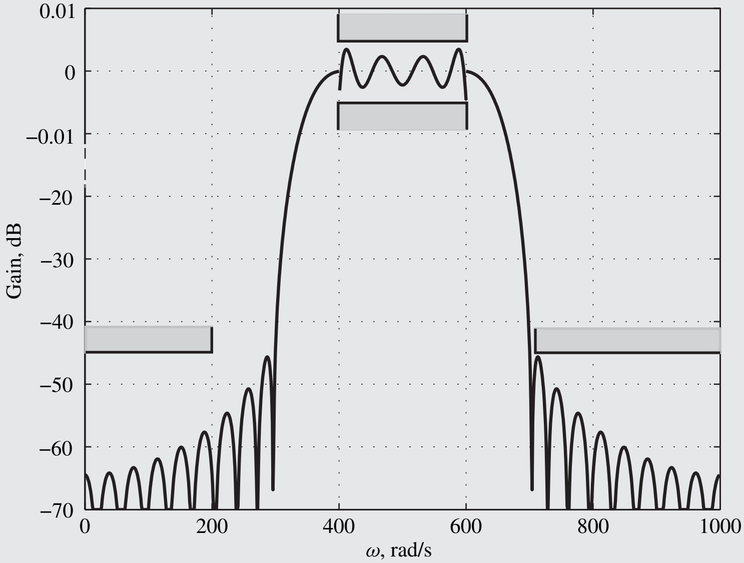


Figure 9.13 Amplitude response of bandpass filter (Example 9.5).

9.4.7 Other Windows

There are several other window functions in the literature that can be applied in the design of nonrecursive filters such as the Saramäki and ultraspherical windows [5, 6]. Like the Dolph-Chebyshev and Kaiser windows, the Saramäki window offers an independent parameter in addition to the window length. The ultraspherical window is more flexible than the others because it offers two independent parameters in addition to the window length. Consequently, it is possible to achieve a great variety of spectral characteristics with it [7], even to design better quality or more economical filters. The ultraspherical window includes the Dolph-Chebyshev and Saramäki windows as special cases and it is also closely related to the Kaiser window.

9.5 DESIGN BASED ON NUMERICAL-ANALYSIS FORMULAS

In signal processing, a continuous-time signal often needs to be interpolated, extrapolated, differentiated at some instant $t = t_1$, or integrated between two distinct instants t_1 and t_2. Such mathematical operations can be performed by using the many classical numerical-analysis formulas [8, 9, 10]. Formulas of this type, which are derived from the Taylor series, can be readily used for the design of nonrecursive filters.

The most fundamental numerical formulas are the formulas for interpolation since they form the basis of many other formulas, including formulas for differentiation and integration. The most commonly used interpolation formulas are the *Gregory-Newton*, *Bessel*, *Everett*, *Stirling*, and *Gauss* interpolation formulas. The value of $x(t)$ at $t = nT + pT$, where $0 \leq p < 1$, is given by the

Gregory-Newton formulas as

$$x(nT + pT) = (1 + \Delta)^p x(nT) = \left[1 + p\Delta + \frac{p(p-1)}{2!}\Delta^2 + \cdots\right] x(nT)$$

and

$$x(nT + pT) = (1 - \nabla)^{-p} x(nT) = \left[1 + p\nabla + \frac{p(p+1)}{2!}\nabla^2 + \cdots\right] x(nT)$$

where

$$\Delta x(nT) = x(nT + T) - x(nT) \qquad \text{and} \qquad \nabla x(nT) = x(nT) - x(nT - T)$$

are commonly referred to as the *forward* and *backward differences*, respectively. On the other hand, the Stirling formula yields

$$\begin{aligned} x(nT + pT) = {} & \left[1 + \frac{p^2}{2!}\delta^2 + \frac{p^2(p^2-1)}{4!}\delta^4 + \cdots\right] x(nT) \\ & + \frac{p}{2}\left[\delta x\left(nT - \tfrac{1}{2}T\right) + \delta x\left(nT + \tfrac{1}{2}T\right)\right] \\ & + \frac{p(p^2-1)}{2(3!)}\left[\delta^3 x\left(nT - \tfrac{1}{2}T\right) + \delta^3 x\left(nT + \tfrac{1}{2}T\right)\right] \\ & + \frac{p(p^2-1)(p^2-2^2)}{2(5!)}\left[\delta^5 x\left(nT - \tfrac{1}{2}T\right) + \delta^5 x\left(nT + \tfrac{1}{2}T\right)\right] + \cdots \end{aligned} \tag{9.31}$$

where

$$\delta x\left(nT + \tfrac{1}{2}T\right) = x(nT + T) - x(nT) \tag{9.32}$$

is known as the *central difference*.

The forward, backward, and central difference operators are, of course, linear and, therefore, higher-order differences can be readily obtained. For example,

$$\begin{aligned} \delta^3 x\left(nT + \tfrac{1}{2}T\right) &= \delta^2\left[\delta x\left(nT + \tfrac{1}{2}T\right)\right] \\ &= \delta^2[x(nT + T) - x(nT)] \\ &= \delta[\delta x(nT + T) - \delta x(nT)] \\ &= \delta\left\{x\left(nT + \tfrac{3}{2}T\right) - x\left(nT + \tfrac{1}{2}T\right)\right. \\ &\quad \left. - \left[x\left(nT + \tfrac{1}{2}T\right) - x\left(nT - \tfrac{1}{2}T\right)\right]\right\} \\ &= \delta x\left(nT + \tfrac{3}{2}T\right) - 2\delta x\left(nT + \tfrac{1}{2}T\right) + \delta x\left(nT - \tfrac{1}{2}T\right) \\ &= [x(nT + 2T) - x(nT + T)] - 2[x(nT + T) - x(nT)] \\ &\quad + [x(nT) - x(nT - T)] \\ &= x(nT + 2T) - 3x(nT + T) + 3x(nT) - x(nT - T) \end{aligned}$$

The first derivative of $x(t)$ with respect to time at $t = nT + pT$ can be expressed as

$$\begin{aligned}\left.\frac{dx(t)}{dt}\right|_{t=nT+pT} &= \frac{dx(nT+pT)}{dp} \times \frac{dp}{dt} \\ &= \frac{1}{T}\frac{dx(nT+pT)}{dp}\end{aligned} \tag{9.33}$$

and, therefore, the above interpolation formulas lead directly to corresponding differentiation formulas. Similarly, integration formulas can be derived by writing

$$\int_{nT}^{t_2} x(t)\,dt = T\int_0^{p_2} x(nT+pT)\,dp$$

where $\qquad nT < t_2 \le nT + T \qquad$ and $\qquad p_2 = \frac{t_2 - nT}{T}$

that is, $0 < p_2 \le 1$.

A nonrecursive filter that can perform *interpolation*, *differentiation*, or *integration* can now be obtained by expressing one of the above numerical formulas in the form of a difference equation. Let $x(nT)$ and $y(nT)$ be the input and output in a nonrecursive filter and assume that $y(nT)$ is equal to the desired function of $x(t)$, that is,

$$y(nT) = f[x(t)] \tag{9.34}$$

For example, if $y(nT)$ is required to be the first derivative of $x(t)$ at $t = nT + pT$, where $0 \le p \le 1$, we can write

$$y(nT) = \left.\frac{dx(t)}{dt}\right|_{t=nT+pT} \tag{9.35}$$

By choosing an appropriate formula for $f[x(t)]$ and then eliminating all the difference operators using their definitions, Eq. (9.34) can be put in the form

$$y(nT) = \sum_{i=-K}^{M} a_i x(nT - iT)$$

Thus the desired transfer function can be obtained as

$$H(z) = \sum_{n=-K}^{M} h(nT)z^{-n}$$

For the case of a forward- or central-difference formula, $H(z)$ is noncausal. Hence for real-time applications it will be necessary to multiply $H(z)$ by an appropriate negative power of z, which would convert a noncausal into a causal design.

Example 9.6 A signal $x(t)$ is sampled at a rate of $1/T$ Hz. Design a sixth-order differentiator with a time-domain response

$$y(nT) = \left.\frac{dx(t)}{dt}\right|_{t=nT}$$

Use the Stirling formula.

■ **Solution**

From Eqs. (9.31) and (9.33)

$$\begin{aligned}
y(nT) = \left.\frac{dx(t)}{dt}\right|_{t=nT} &= \frac{1}{2T}\left[\delta x\left(nT - \tfrac{1}{2}T\right) + \delta x\left(nT + \tfrac{1}{2}T\right)\right] \\
&\quad - \frac{1}{12T}\left[\delta^3 x\left(nT - \tfrac{1}{2}T\right) + \delta^3 x\left(nT + \tfrac{1}{2}T\right)\right] \\
&\quad + \frac{1}{60T}\left[\delta^5 x\left(nT - \tfrac{1}{2}T\right) + \delta^5 x\left(nT + \tfrac{1}{2}T\right)\right] + \cdots
\end{aligned}$$

Now, on using Eq. (9.32)

$$\begin{aligned}
\delta x\left(nT - \tfrac{1}{2}T\right) + \delta x\left(nT + \tfrac{1}{2}T\right) &= x(nT + T) - x(nT - T) \\
\delta^3 x\left(nT - \tfrac{1}{2}T\right) + \delta^3 x\left(nT + \tfrac{1}{2}T\right) &= x(nT + 2T) - 2x(nT + T) \\
&\quad + 2x(nT - T) - x(nT - 2T) \\
\delta^5 x\left(nT - \tfrac{1}{2}T\right) + \delta^5 x\left(nT + \tfrac{1}{2}T\right) &= x(nT + 3T) - 4x(nT + 2T) \\
&\quad + 5x(nT + T) - 5x(nT - T) \\
&\quad + 4x(nT - 2T) - x(nT - 3T)
\end{aligned}$$

Hence

$$\begin{aligned}
y(nT) = \frac{1}{60T}&[x(nT + 3T) - 9x(nT + 2T) + 45x(nT + T) \\
&-45x(nT - T) + 9x(nT - 2T) - x(nT - 3T)]
\end{aligned}$$

and, therefore

$$H(z) = \frac{1}{60T}(z^3 - 9z^2 + 45z - 45z^{-1} + 9z^{-2} - z^{-3}) \quad ■$$

Note that the differentiator has an antisymmetrical impulse response, i.e., it has a constant group delay, and it is also noncausal. A causal filter can be obtained by multiplying $H(z)$ by z^{-3}. The amplitude response of the differentiator is plotted in Fig. 9.14 for $\omega_s = 2\pi$.

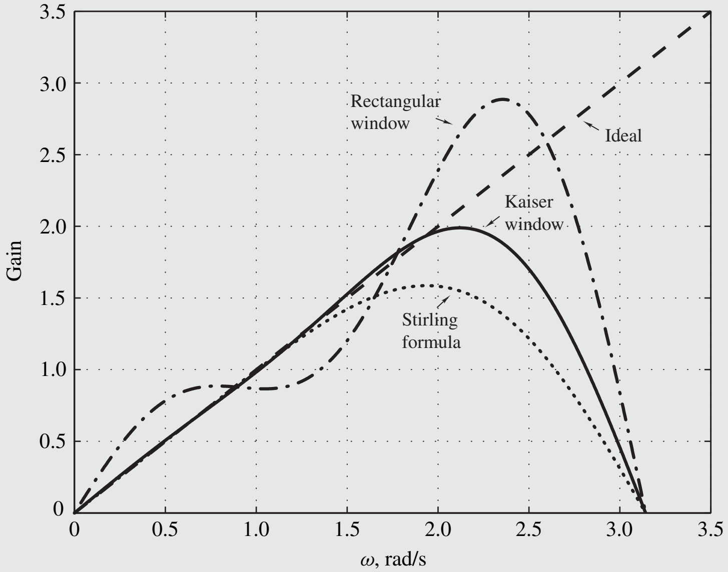


Figure 9.14 Amplitude response of digital differentiators (Examples 9.6 and 9.7).

Differentiators can also be designed by employing the Fourier series method of Sec. 9.3. An analog differentiator is characterized by the continuous-time transfer function

$$H(s) = s$$

Hence a corresponding digital differentiator can be designed by assigning

$$H(e^{j\omega T}) = j\omega \qquad \text{for} \qquad 0 \le |\omega| < \omega_s/2 \tag{9.36}$$

Then on assuming a periodic frequency response, the appropriate impulse response can be determined by using Eq. (9.11). Gibbs' oscillations due to the transition in $H(e^{j\omega T})$ at $\omega = \omega_s/2$ can be reduced, as before, by using the window technique.

Example 9.7 Redesign the differentiator of Example 9.6 by employing the Fourier-series method. Use (a) a rectangular window and (b) the Kaiser window with $\alpha = 3.0$.

■ **Solution**

(a) From Eqs. (9.36) and (9.11), we have

$$h(nT) = \frac{1}{\omega_s}\int_{-\omega_s/2}^{\omega_s/2} j\omega e^{j\omega nT}\, d\omega = -\frac{1}{\omega_s}\int_{0}^{\omega_s/2} 2\omega \sin(\omega nT)\, d\omega$$

On integrating by parts, we get

$$h(nT) = \frac{1}{nT}\cos \pi n - \frac{1}{n^2 \pi T}\sin \pi n$$

or

$$h(nT) = \begin{cases} 0 & \text{for } n = 0 \\ \dfrac{1}{nT}\cos \pi n & \text{otherwise} \end{cases}$$

Now if we use the rectangular window with $N = 7$, we deduce

$$H_w(z) = \frac{1}{6T}(2z^3 - 3z^2 + 6z - 6z^{-1} + 3z^{-2} - 2z^{-3}) \quad \blacksquare$$

(*b*) Similarly, the Kaiser window yields

$$H_w(z) = \sum_{n=-3}^{3} w_K(nT)h(nT)z^{-n} \quad \blacksquare$$

where $w_K(nT)$ can be computed using Eq. (9.25). The amplitude responses of the two differentiators are compared in Fig. 9.14 with the response of the differentiator obtained in Example 9.6.

As before, the parameter α in the Kaiser window can be increased to increase the in-band accuracy or decreased to increase the bandwidth. Thus the differentiator obtained with the Kaiser window has the important advantage that it can be adjusted to suit the application. The design of digital differentiators satisfying prescribed specifications is considered in Refs. [11, 12] (see also Sec. 15.9.3).

REFERENCES

[1] F. F. Kuo and J. F. Kaiser, *System Analysis by Digital Computer*, Chap. 7, New York: Wiley, 1966.

[2] R. B. Blackman, *Data Smoothing and Prediction*, Reading, MA: Addison-Wesley, 1965.

[3] C. L. Dolph, “A current distribution for broadside arrays which optimizes the relationship between beamwidth and side-lobe level,” *Proc. IRE*, vol. 34, pp. 335–348, June 1946.

[4] J. F. Kaiser, “Nonrecursive digital filter design using the I_0-sinh window function,” *in Proc. IEEE Int. Symp. Circuit Theory, 1974*, pp. 20–23.

[5] T. Saramäki, “Adjustable windows for the design of FIR filters—A tutorial,” *6th Mediterranean Electrotechnical Conference*, vol. 1, pp. 28–33, May 1991.

[6] R. L. Streit, “A two-parameter family of weights for nonrecursive digital filters and antennas,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, pp. 108–118, Feb. 1984.

[7] S. W. A. Bergen and A. Antoniou, “Design of Ultraspherical Window Functions with Prescribed Spectral Characteristics,” *Applied Journal of Signal Processing*, vol. 13, pp. 2053–2065, 2004.

[8] R. Butler and E. Kerr, *An Introduction to Numerical Methods*, London: Pitman, 1962.

[9] C. E. Fröberg, *Introduction to Numerical Analysis*, Reading, MA: Addison-Wesley, 1965.

[10] M. L. James, G. M. Smith, and J. C. Wolford, *Applied Numerical Methods for Digital Computation*, New York: Harper & Row, 1985.

[11] A. Antoniou, "Design of digital differentiators satisfying prescribed specifications," *Proc. Inst. Elect. Eng.*, Part E, vol. 127, pp. 24–30, Jan. 1980.

[12] A. Antoniou and C. Charalambous, "Improved design method for Kaiser differentiators and comparison with equiripple method," *Proc. Inst. Elect. Eng.*, Part E, vol. 128, pp. 190–196, Sept. 1981.

PROBLEMS

9.1. (*a*) A nonrecursive filter is characterized by the transfer function

$$H(z) = \frac{1 + 2z + 3z^2 + 4z^3 + 3z^4 + 2z^5 + z^6}{z^6}$$

Find the group delay.

(*b*) Repeat part (*a*) if

$$H(z) = \frac{1 - 2z + 3z^2 - 4z^3 + 3z^4 - 2z^5 + z^6}{z^6}$$

9.2. Figure P9.2 shows the zero-pole plots of two nonrecursive filters. Check each filter for phase-response linearity.

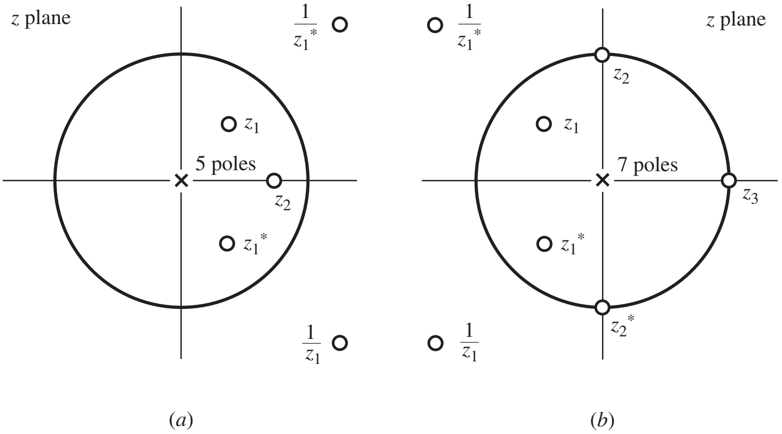


Figure P9.2

9.3. A nonrecursive bandstop digital filter can be designed by applying the Fourier series method to the idealized frequency response:

$$H(e^{j\omega T}) = \begin{cases} 1 & \text{for } |\omega| \le \omega_{c1} \\ 0 & \text{for } \omega_{c1} < |\omega| < \omega_{c2} \\ 1 & \text{for } \omega_{c2} \le |\omega| \le \omega_s/2 \end{cases}$$

(*a*) Obtain an expression for the impulse response of the filter.

(*a*) Obtain a causal transfer function assuming a filter length $N = 11$.

9.4. A nonrecursive digital filter can be designed by applying the Fourier series method to the idealized frequency response:

$$H(e^{j\omega T}) \approx \begin{cases} 0 & \text{for } |\omega| < \omega_{c1} \text{ rad/s} \\ 1 & \text{for } \omega_{c1} \le |\omega| \le \omega_{c2} \text{ rad/s} \\ 0 & \text{for } \omega_{c2} < |\omega| < \omega_{c3} \text{ rad/s} \\ 1 & \text{for } \omega_{c3} \le |\omega| \le \omega_{c4} \text{ rad/s} \\ 0 & \text{for } \omega_{c4} < |\omega| \le \omega_s/2 \text{ rad/s} \end{cases}$$

(*a*) Obtain an expression for the impulse response of the filter using the Fourier-series method.

(*a*) Obtain a causal transfer function assuming a filter length $N = 15$.

9.5. (*a*) Derive an exact expression for the spectrum of the Blackman window.

(*b*) Using the result in part (*a*) and assuming that $N \gg 1$, show that the main-lobe width for the Blackman window is approximately $6\omega_s/N$.

9.6. (*a*) Design a nonrecursive highpass filter in which

$$H(e^{j\omega T}) \approx \begin{cases} 1 & \text{for } 2.5 \le |\omega| \le 5.0 \text{ rad/s} \\ 0 & \text{for } |\omega| < 2.5 \text{ rad/s} \end{cases}$$

Use the rectangular window and assume that $\omega_s = 10$ rad/s and $N = 11$.

(*b*) Repeat part (*a*) with $N = 21$ and $N = 31$. Compare the three designs.

9.7. Redesign the filter of Prob. 9.6 using the von Hann, Hamming, and Blackman windows in turn. Assume that $N = 21$. Compare the three designs.

9.8. Design a nonrecursive bandpass filter in which

$$H(e^{j\omega T}) \approx \begin{cases} 0 & \text{for } |\omega| < 400 \text{ rad/s} \\ 1 & \text{for } 400 \le |\omega| \le 600 \text{ rad/s} \\ 0 & \text{for } 600 < |\omega| \le 1000 \text{ rad/s} \end{cases}$$

Use the von Hann window and assume that $\omega_s = 2000$ rad/s and $N = 21$. Check your design by plotting the amplitude response over the frequency range 0 to 1000 rad/s.

9.9. Design a nonrecursive bandstop filter with a frequency response

$$H(e^{j\omega T}) \approx \begin{cases} 1 & \text{for } |\omega| \le 300 \text{ rad/s} \\ 0 & \text{for } 300 < |\omega| < 700 \text{ rad/s} \\ 1 & \text{for } 700 \le |\omega| \le 1000 \text{ rad/s} \end{cases}$$

Use the Hamming window and assume that $\omega_s = 2000$ rad/s and $N = 21$. Check your design by plotting the amplitude response over the frequency range 0 to 1000 rad/s.

9.10. A digital filter is required with a frequency response like that depicted in Fig. P9.10.

(*a*) Obtain a nonrecursive design using the rectangular window assuming that $\omega_s = 10$ rad/s and $N = 21$.

(*b*) Repeat part *(a)* using a Dolph-Chebyshev window with a ripple ratio of -30 dB.

(*c*) Repeat part (*a*) using a Kaiser window with an α of 3.0.

(*d*) Compare the designs obtained in parts (*a*) to (*c*).

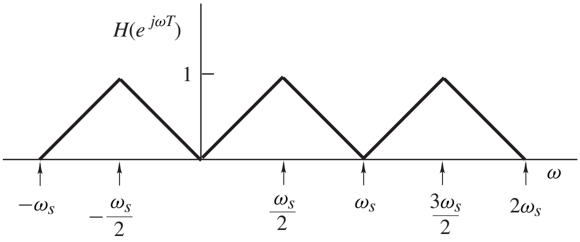


Figure P9.10

9.11. A digital filter with a frequency response like that depicted in Fig. P9.11 is required.

(*a*) Obtain a nonrecursive design using a rectangular window assuming that $\omega_s = 10$ rad/s and $N = 21$.

(*b*) Repeat part *(a)* using a Dolph-Chebyshev window with a ripple ratio of -20 dB.

(*c*) Repeat part (*a*) using a Kaiser window with an α of 2.0.

(*d*) Compare the designs obtained in parts (*a*) to (*c*).

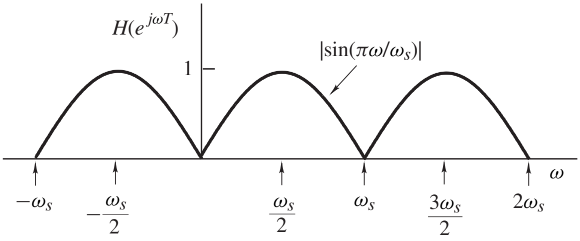


Figure P9.11

9.12. (*a*) Using the idealized amplitude response in Example 9.5, design a bandpass filter using the Dolph-Chebyshev window. The required filter specifications are as follows:

- Ripple ratio: -20 dB
- $\omega_{c1} = 3.0, \ \omega_{c2} = 7.0$ rad/s
- $N = 21$
- $\omega_s = 20$ rad/s

(*b*) Assuming that the passband extends from 3.8 to 6.8 modify the design in part (*a*) so as to achieve an amplitude response that oscillates about unity.

(*c*) Find the passband peak-to-peak ripple Ap in dB.

(*d*) Assuming that the lower and upper stopbands extend from 0 to 2.1 and 7.9 to 10 rad/s, respectively, find the minimum stopband attenuation.

9.13. (*a*) Repeat Prob. 9.12 assuming a ripple ratio of -25 dB.

(*b*) Compare the design of this problem with that of Prob. 9.12.

9.14. Show that the Kaiser window includes the rectangular window as a special case.

9.15. (*a*) Repeat Prob. 9.12 using a Kaiser window with $\alpha = 1.0$.

(*b*) Repeat Prob. 9.12 using a Kaiser window with $\alpha = 4.0$.

(*c*) Compare the designs in parts (*a*) and (*b*).

9.16. Design a nonrecursive lowpass filter that would satisfy the following specifications:

$$A_p \le 0.1 \text{ dB} \qquad A_a \ge 44.0 \text{ dB}$$

$$\omega_p = 20 \text{ rad/s} \qquad \omega_a = 30 \text{ rad/s} \qquad \omega_s = 100 \text{ rad/s}$$

9.17. Design a nonrecursive highpass filter that would satisfy the following specifications:

$$A_p \le 0.3 \text{ dB} \qquad A_a \ge 45.0 \text{ dB}$$

$$\omega_p = 3 \text{ rad/s} \qquad \omega_a = 2 \text{ rad/s} \qquad \omega_s = 10 \text{ rad/s}$$

9.18. Design a nonrecursive bandpass filter that would satisfy the following specifications:

$$A_p \le 0.5 \text{ dB} \qquad A_a \ge 35.0 \text{ dB} \qquad \omega_{p1} = 40 \text{ rad/s} \qquad \omega_{p2} = 60 \text{ rad/s}$$

$$\omega_{a1} = 20 \text{ rad/s} \qquad \omega_{a2} = 80 \text{ rad/s} \qquad \omega_s = 200 \text{ rad/s}$$

9.19. Design a nonrecursive bandstop filter that would satisfy the following specifications:

$$A_p \le 0.2 \text{ dB} \qquad A_a \ge 40 \text{ dB} \qquad \omega_{p1} = 1000 \text{ rad/s} \qquad \omega_{p2} = 4000 \text{ rad/s}$$

$$\omega_{a1} = 2000 \text{ rad/s} \qquad \omega_{a2} = 3000 \text{ rad/s} \qquad \omega_s = 10{,}000 \text{ rad/s}$$

9.20. (*a*) Show that

$$\mathcal{Z}\nabla^k x(nT) = (1 - z^{-1})^k X(z)$$

(*b*) A signal $x(t)$ is sampled at a rate of 2π rad/s. Design a sixth-order differentiator in which

$$y(nT) \approx \left.\frac{dx(t)}{dt}\right|_{t=nT}$$

Use the Gregory-Newton backward-difference formula.

(*c*) Repeat part (*b*) using the Stirling central-difference formula.

9.21. The phase response $\theta(\omega)$ of a digital filter is sampled at $\omega = n\Omega$ for $n = 0, 1, 2, \ldots$. Design a sixth-order digital differentiator that can be used to generate the group delay of the digital filter. Use the Stirling formula.

9.22. A signal $x(t)$ is sampled at a rate of 2π rad/s. Design a sixth-order integrator filter in which

$$y(nT) \approx \int_{nT}^{(n+1)T} x(t)\,dt$$

Use the Gregory-Newton backward-difference formula.

9.23. Two digital filters are to be cascaded. The sampling frequency in the first filter is 2π rad/s, and that in the second is 4π rad/s. Design a sixth-order interface using the Gregory-Newton backward-difference formula. *Hint*: Design an interpolating filter.

CHAPTER 10

APPROXIMATIONS FOR ANALOG FILTERS

10.1 INTRODUCTION

As mentioned in the introduction of Chap. 8, the available approximation methods for recursive digital filters can be classified as indirect or direct. Alternatively, they can be classified as noniterative or iterative. In indirect methods a discrete-time transfer function that would satisfy certain required specifications is deduced from a corresponding continuous-time transfer function through the application of certain transformations. In effect, indirect methods entail a closed-form formulation and they are, therefore, noniterative. The continuous-time transfer function is obtained by using one of several classical approximation methods for analog filters. On the other hand, in direct methods, a discrete-time transfer function is generated directly in the z domain usually using an optimization algorithm of some kind, i.e., direct methods are also iterative most of the time.

Indirect methods have a historical basis. As detailed in Chap. 1, analog filters began to emerge around 1915 and during the first half of the 20th century some really powerful analog-filter approximation methods were invented [1–5]. When digital filters appeared on the scene during the 1960s, it was quite natural for engineers to attempt to obtain digital-filter approximations by adapting, modifying, or transforming well-established analog-filter approximations. It is now clear, that these indirect methods have passed the test of time and are, as a consequence, very much a part of a modern DSP designer's tool kit. This hypothesis can be verified by counting the analog-filter approximation methods found in MATLAB, for example.

This chapter considers in some detail several analog-filter approximation methods that are suitable for the design of filters with piecewise-constant amplitude responses, i.e., filters whose passband and stopband gains are constant and zero, respectively, to within prescribed tolerances. The most frequently used approximation methods of this type are as follows:

1. Butterworth
2. Chebyshev
3. Inverse-Chebyshev
4. Elliptic
5. Bessel-Thomson

In the first four methods, attention is focused on deriving a continuous-time transfer function that would yield a specified amplitude response (or loss characteristic) and no particular attention is paid to the associated phase response. This is in contrast with the design of nonrecursive filters whereby the linearity of the phase response is imposed at the outset, as may be recalled from Chap. 9. In consequence, the phase response achieved through these analog-filter approximations turns out to be nonlinear and, as a result, the group delay tends to vary with frequency. This may present a problem in applications where phase distortion is undesirable (see Sec. 5.7). In the fifth approximation method, namely, the Bessel-Thomson [6] method, a constraint is imposed on the group delay associated with the transfer function, which results in a fairly *linear phase response* over a certain frequency range.

The chapter begins with an introductory section dealing with the terminology and characterization of analog filters. While digital-filter designers talk about amplitude responses and gains, their analog-filter counterparts are more inclined to deal with loss characteristics and losses. This is because passive *RLC* analog filters, the forefathers of all filters, can provide only loss which can vary from zero to some large positive value. However, there is also a practical reason in describing analog filters in terms of loss characteristics. The derivations of the necessary formulas for the various approximations are that much easier to handle. The treatment of the basics provided is somewhat cursory and it is intended as a refresher. The interested reader is referred to Refs. [1–5] and also to a survey article written by the author in Ref. [7] for a more detailed exposition.

The derivations provided deal with *lowpass* approximations since other types of approximations can be readily obtained through the application of transformations. Suitable transformations for the design of *highpass, bandpass*, and *bandstop* filters are described at the end of the chapter.

It should be mentioned that the derivation of the formulas for the elliptic approximation is quite demanding as it entails a basic understanding of *elliptic* functions. Fortunately, however, the formulas that give the transfer-function coefficients can be put in a fairly simple form that is easy to apply even for the uninitiated. The elliptic approximation is treated in detail here because it yields the lowest-order transfer function for filters that are required to have prescribed piecewise-constant loss specifications, which makes it the *optimal approximation* for such applications. The reader who is interested in the application of the method may skip the derivations and proceed to Sec. 10.6.6 for a step-by-step procedure for the design. The reader who is also interested in the derivation of this very important method may start by reading Appendix B which provides a brief review of the fundamentals of elliptic functions.

The application of analog-filter approximations in the design of recursive digital filters will be considered in Chaps. 11 and 12. Chapter 12 considers, in addition, a *delay-equalization technique* that can be used in conjunction with the above methods for the design of digital filters with approximately linear phase response. Optimization methods that can be used to design recursive filters and equalizers can be found in Chap. 16.

10.2 BASIC CONCEPTS

The basics of analog filters bear a one-to-one correspondence with the basics of digital filters, i.e., characterization, time-domain analysis, stability, frequency-domain analysis, and so on.

10.2.1 Characterization

An nth-order linear causal analog filter with input $v_i(t)$ and output $v_o(t)$ such as that in Fig. 10.1 can be characterized by a differential equation of the form

$$b_n \frac{d^n v_o(t)}{dt^n} + b_{n-1} \frac{d^{n-1} v_o(t)}{dt^{n-1}} + \cdots + b_0 v_o(t) = a_n \frac{d^n v_i(t)}{dt^n} + a_{n-1} \frac{d^{n-1} v_i(t)}{dt^{n-1}} + \cdots + a_0 v_i(t) \tag{10.1}$$

The coefficients $a_0,\ a_1, \ldots,\ a_n$ and $b_0,\ b_1, \ldots,\ b_n$ are functions of the element values and are real since the parameters of the filter (e.g., resistances, inductances, and so on) are real. The element values can be time-dependent in real life but are assumed to be time-invariant in theory.

10.2.2 Laplace Transform

The representation and analysis of discrete-time systems is facilitated through the use of the z transform. The transform of choice for analog filters and continuous-time systems in general is, of course, the Laplace transform which has already been encountered in Sec. 6.5.2. It is defined as

$$X(s) = \int_{-\infty}^{\infty} x(t) e^{-st}\, dt \tag{10.2}$$

where s is a complex variable of the form $s = \sigma + j\omega$. Signal $x(t)$ can be recovered from $X(s)$ by applying the inverse Laplace transform which is given by

$$x(t) = \frac{1}{2\pi j} \int_{C-j\infty}^{C+j\infty} X(s) e^{st}\, ds \tag{10.3}$$

where C is a positive constant.

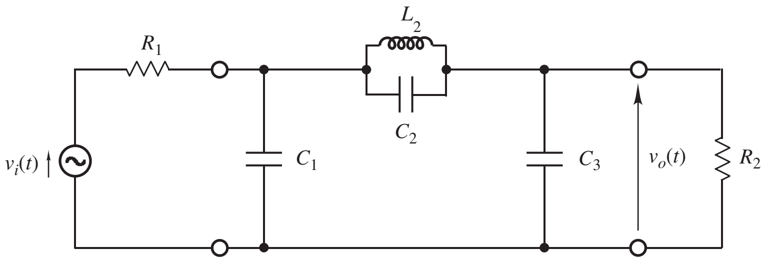


Figure 10.1 Passive *RLC* analog filter.

The Laplace transform can be obtained by letting $j\omega \rightarrow s$ in the Fourier transform (see Fig. 6.13) and, therefore, it is an *analytic continuation* of the latter transform (see Sec. A.8). As for the Fourier and z transforms, short-hand notations can be used for the Laplace transform, i.e.,

$$X(s) = \mathcal{L}x(t) \qquad \text{and} \qquad x(t) = \mathcal{L}^{-1}X(s) \qquad \text{or} \qquad X(s) \leftrightarrow x(t)$$

10.2.3 The Transfer Function

The Laplace transform of the kth derivative of some function of time $x(t)$ is given by

$$\mathcal{L}\left[\frac{d^k x(t)}{dt^k}\right] = s^k X(s) - s^{k-1}x(0) - s^{k-2}\left.\frac{dx(t)}{dt}\right|_{t=0} - \cdots - \left.\frac{d^{k-1}x(t)}{dt^{k-1}}\right|_{t=0}$$

where

$$x(0) \qquad \left.\frac{dx(t)}{dt}\right|_{t=0} \qquad \cdots \qquad \left.\frac{d^{k-1}x(t)}{dt^{k-1}}\right|_{t=0}$$

are said to be the *initial conditions* of $x(t)$. In an analog filter, initial conditions are associated with the presence of charges in capacitors and inductors. In the present context, the analog filter can be safely assumed to be initially relaxed and thus all initial conditions can be deemed to be zero. On applying the Laplace transform to the differential equation in Eq. (10.1), we obtain

$$(b_n s^n + b_{n-1}s^{n-1} + \cdots + b_0)V_o(s) = (a_n s^n + a_{n-1}s^{n-1} + \cdots + a_0)V_i(s)$$

and thus

$$\frac{V_o(s)}{V_i(s)} = \frac{\sum_{i=0}^{n} a_i s^i}{\sum_{i=0}^{n} b_i s^i} = H(s) \tag{10.4}$$

This equation defines the transfer function of the filter, $H(s)$, which can also be expressed in terms of its zeros and poles as

$$H(s) = \frac{N(s)}{D(s)} = H_0 \frac{\prod_{i=1}^{n}(s - z_i)}{\prod_{i=1}^{n}(s - p_i)} \tag{10.5}$$

The transfer function of a continuous-time system plays the same key role as that of a discrete-time system. It provides a complete description of the filter both in the time and frequency domains.

10.2.4 Time-Domain Response

The time-domain response of an analog filter can be expressed in terms of the time convolution as

$$v_o(t) = \int_{-\infty}^{\infty} h(\tau)v_i(t-\tau)\,d\tau$$

where $h(t)$ is the response of the filter to the continuous-time impulse function $\delta(t)$. Now from the time-convolution theorem of the Fourier transform (Theorem 2.14), we can write

$$V_o(j\omega) = H(j\omega)V_i(j\omega) \tag{10.6a}$$

that is, the Fourier transform (or frequency spectrum) of the output signal is equal to the Fourier transform of the impulse response times the Fourier transform of the input signal. If we now let $j\omega = s$, we obtain

$$V_o(s) = H(s)V_i(s) \tag{10.6b}$$

or

$$H(s) = \frac{V_o(s)}{V_i(s)} \tag{10.6c}$$

Eq. (10.6c) is essentially the same as Eq. (10.4) and, in effect, the transfer function of an analog filter is one and the same as the Laplace transform of the impulse response.

The response of an analog filter to an arbitrary excitation can be deduced by obtaining the inverse Laplace transform of $V_o(s)$ and from Eq. (10.6b), we have

$$v_o(t) = \mathcal{L}^{-1}[H(s)V_i(s)]$$

If (i) the singularities of $V_o(s)$ in the finite plane are poles, and (ii) $V_o(s) \to 0$ uniformly with respect to the angle of s as $|s| \to \infty$ with $\sigma \le C$, where C is a positive constant, then [8]

$$v_o(t) = \begin{cases} 0 & \text{for } t < 0 \\ \dfrac{1}{2\pi j}\displaystyle\int_\Gamma V_o(s)e^{st}\,ds & \text{for } t \ge 0 \end{cases} \tag{10.7}$$

where Γ is a contour in the counterclockwise sense made up of the part of the circle $s = Re^{j\theta}$ to the left of line $s = C$ and the segment of the line $s = C$ that overlaps the circle, as depicted in Fig. 10.2, and C and R are sufficiently large to ensure that Γ encloses all the finite poles of $V_o(s)$.

From the residue theorem (see Sec. A.7), the contour integral in Eq. (10.7) can be evaluated as

$$\frac{1}{2\pi j}\int_\Gamma V_o(s)e^{st}ds = \sum_{i=1}^{P} \underset{s=p_i}{\Re es}\ [V_o(s)e^{st}] \tag{10.8}$$

where P is the number of poles in $V_o(s)$.

Note that if the numerator degree of the transfer function is equal to the denominator degree, then condition (ii) above is violated and the inversion technique described cannot be applied. However, the problem can be readily circumvented by expressing $V_o(s)$ as

$$V_o(s) = R_\infty + V_o'(s)$$

where

$$R_\infty = \lim_{s\to\infty} Y(s)$$

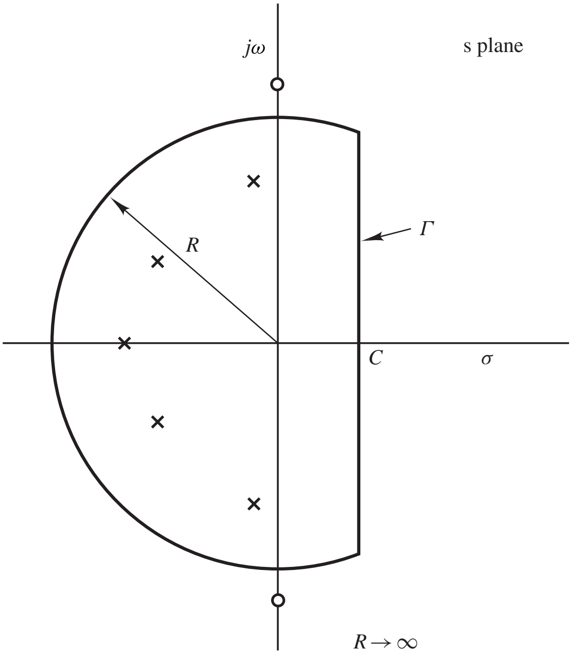


Figure 10.2 Contour Γ for the evaluation of the inverse Laplace transform.

As can be readily verified, in such a case

$$V_o'(s) = V_o(s) - R_\infty$$

would satisfy conditions (i) and (ii) above and thus the inverse Laplace transform of $Y(s)$ can be obtained as

$$v_o(t) = R_\infty \delta(t) + \mathcal{L}^{-1} V_o'(s)$$

The simplest way to obtain the time-domain response of a filter is to express $H(s)V_i(s)$ as a partial-fraction expansion and then invert the resulting fractions individually. If $V_o(s)$ has simple poles, we can write

$$V_o(s) = R_\infty + \sum_{i=1}^{P} \frac{R_i}{s - p_i} \tag{10.9}$$

where R_∞ is a constant and

$$R_i = \lim_{s \to p_i} \left[(s - p_i)\, V_o(s)\right]$$

is the residue of pole $s = p_i$. On applying the inversion formula in Eq. (10.7) to each partial fraction, we obtain

$$v_o(t) = R_\infty \delta(t) + u(t) \sum_{i=1}^{P} R_i e^{p_i t} \tag{10.10}$$

where $\delta(t)$ and $u(t)$ are the impulse function and unit step, respectively.

The impulse response $h(t)$ of an analog system, that is, $\mathcal{L}^{-1}H(s)$, is of as much importance as the impulse response of a discrete-time system since its absolute integrability is a necessary and sufficient condition for the stability of the system. In discrete-time systems, the absolute summability of the impulse response imposes the condition that the poles of the transfer function be located inside the unit circle. Similarly, the absolute integrability of the impulse response in analog systems imposes the condition that the poles of the transfer function be located in the left-half s plane.

Sometimes the unit-step response of an analog system may be required (see Prob. 11.9, for example). The Laplace transform of the unit step, $u(t)$, is $1/s$. Hence the unit-step response is obtained as

$$v_o(t) = \mathcal{R}u(t) = \mathcal{L}^{-1}\frac{H(s)}{s}$$

In certain applications, it may be necessary to deduce the initial or final value of a signal from its Laplace transform (see Sec. 11.3, for example). Given the Laplace transform $X(s)$ of a right-sided signal $x(t)$, the initial and final values of the signal can be obtained as

$$x(0+) = \lim_{s\to\infty} [sX(s)]$$

and

$$\lim_{t\to\infty} x(t) = \lim_{s\to 0}[sX(s)]$$

10.2.5 Frequency-Domain Analysis

The sinusoidal response of an analog filter can be obtained

$$v_o(t) = \mathcal{L}^{-1}[H(s)V_i(s)]$$

where

$$X(s) = \mathcal{L}[u(t)\sin\omega t] = \frac{\omega}{(s + j\omega)(s - j\omega)}$$

Through an analysis similar to that found in Sec. 5.5.1, it can be shown that the sinusoidal response of an analog filter comprises a transient and a steady-state component (see Prob. 10.1). If the analog filter is stable,[1] i.e., the poles of the transfer function are in the left-half s plane, the transient

[1] *Of course passive RLC analog filters cannot be unstable for the same reason that a piano note cannot persist forever but there are active analog filters that can become unstable.*

component approaches zero as t is increased and eventually the response of the filter assumes a steady state of the form

$$v_o(t) = M(\omega)\sin[\omega t + \theta(\omega)] \tag{10.11}$$

where

$$M(\omega) = |H(j\omega)| \qquad \text{and} \qquad \theta(\omega) = \arg H(j\omega)$$

are the gain and phase shift of the filter. As functions of frequency $M(\omega)$ and $\theta(\omega)$ are the amplitude and phase response, respectively, and function

$$H(j\omega) = M(\omega)e^{j\theta(\omega)}$$

which includes both the amplitude and phase responses, defines the frequency response.

Given an arbitrary filter characterized by a transfer function such as that in Eq. (10.5) with M simple zeros and N simple poles, we can write

$$H(j\omega) = M(\omega)e^{j\theta(\omega)} = \frac{H_0 \prod_{i=1}^{M}(j\omega - z_i)}{\prod_{i=1}^{N}(j\omega - p_i)} \tag{10.12}$$

By letting

$$j\omega - z_i = M_{z_i}e^{j\psi_{z_i}} \qquad \text{and} \qquad j\omega - p_i = M_{p_i}e^{j\psi_{p_i}}$$

we obtain

$$M(\omega) = \frac{|H_0| \prod_{i=1}^{M} M_{z_i}}{\prod_{i=1}^{N} M_{p_i}} \tag{10.13}$$

and

$$\theta(\omega) = \arg\ H_0 + \sum_{i=1}^{M} \psi_{z_i} - \sum_{i=1}^{N} \psi_{p_i} \tag{10.14}$$

where arg $H_0 = \pi$ if H_0 is negative. Thus the amplitude and phase responses of an analog filter can be determined by evaluating the transfer function on the imaginary axis of the s plane, as illustrated in Fig. 10.3.

As in discrete-time systems the group delay is defined as

$$\tau(\omega) = -\frac{d\theta(\omega)}{d\omega}$$

and as a function of frequency $\tau(\omega)$ is said to be the *delay characteristic*.

The approximation methods to be presented in this chapter have evolved hand in hand with realization methods for passive *RLC* analog filters such as that in Fig. 10.1. On the basis of energy considerations, $M(\omega)$ is always equal to or less than unity in these filters and thus the gain in dB is always equal to or less than zero. For this reason, the past literature on passive analog filters has been almost entirely in terms of the loss (or attenuation) $A(\omega)$ which is always equal to or greater

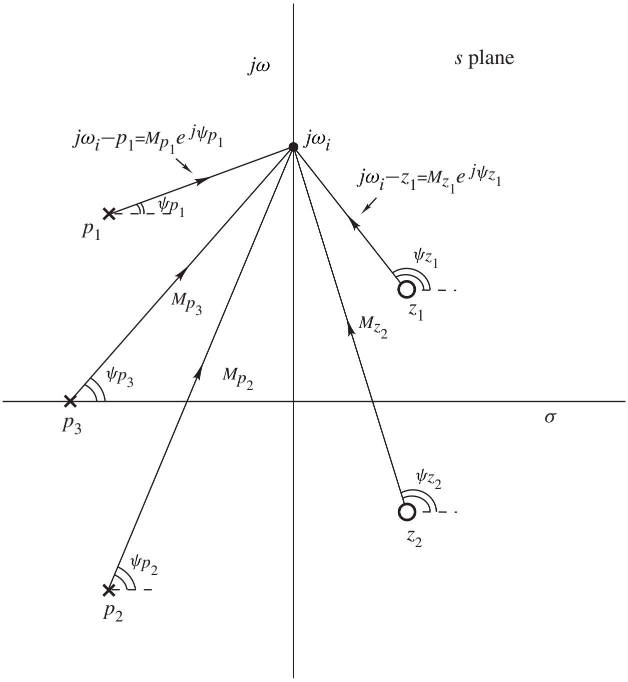


Figure 10.3 Evaluation of frequency response.

than zero since it is defined as the reciprocal of the gain in dB. The loss can be expressed as

$$A(\omega) = 20\log\left|\frac{V_i(j\omega)}{V_o(j\omega)}\right| = 20\log\frac{1}{|H(j\omega)|} = 10\log L(\omega^2)$$

where

$$L(\omega^2) = \frac{1}{H(j\omega)H(-j\omega)} \tag{10.15}$$

A plot of $A(\omega)$ versus ω is often referred to as a *loss characteristic.*

With $\omega = s/j$ in Eq. (10.15), the function

$$L(-s^2) = \frac{D(s)D(-s)}{N(s)N(-s)}$$

can be formed. This is called the *loss function* of the filter, and, as can be easily verified, its zeros are the poles of $H(s)$ and their negatives, whereas its poles are the zeros of $H(s)$ and their negatives. Typical zero-pole plots for $H(s)$ and $L(-s^2)$ are shown in Fig. 10.4.

10.2.6 Ideal and Practical Filters

The solution of the approximation problem for analog filters is facilitated by stipulating the existence of a set of idealized filters that can serve as models. An ideal lowpass filter is one that will pass

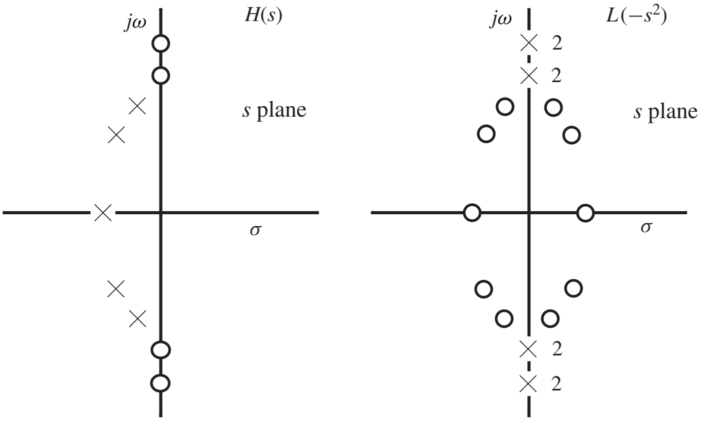


Figure 10.4 Typical zero-pole plots for $H(s)$ and $L(-s^2)$.

only low-frequency and reject high-frequency components. Such a filter would have zero loss in its passband and infinite loss in its stopband as depicted in Fig. 10.5*a*. The boundary between the passband and stopband, namely, ω_c, can be referred to as the *cutoff frequency.* Highpass, bandpass, and bandstop filters with loss characteristics like those depicted in Fig. 10.5*b* to *d* can similarly be defined.

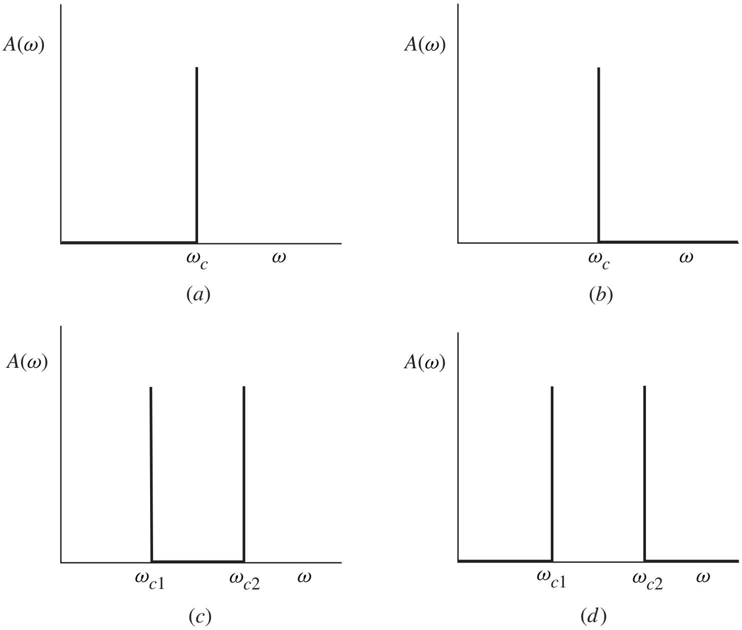


Figure 10.5 Ideal loss characteristics: (*a*) Lowpass, (*b*) highpass, (*c*) bandpass, (*d*) bandstop.

A practical lowpass filter differs from an ideal one in that the passband loss is not zero, the stopband loss is not infinite, and the transition between passband and stopband is gradual. The loss characteristic might assume the form shown in Fig. 10.6*a* where ω_p is the passband edge, ω_a is the stopband edge, A_p is the maximum passband loss, and A_a is the minimum stopband loss. The cutoff frequency ω_c is usually a loose demarcation boundary between passband and stopband, which can vary from the one type of approximation to the next usually on the basis of convenience. For example, it is often used to refer to the 3-dB frequency in Butterworth filters or the square root of $\omega_p\omega_a$ in the case of elliptic filters. Typical characteristics for practical highpass, bandpass, and bandstop filters are shown in Fig. 10.6*b* to *d*.

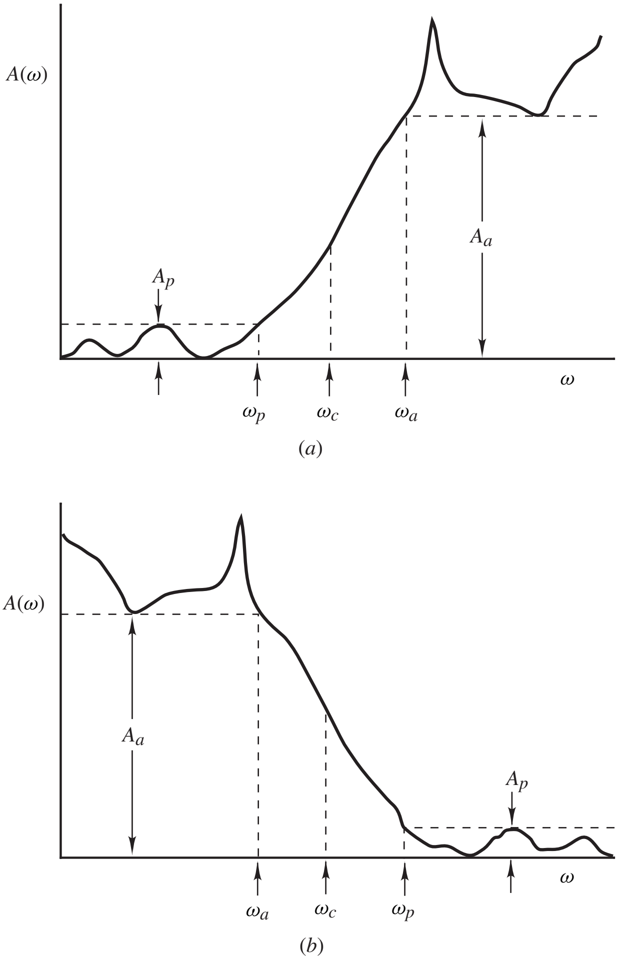


Figure 10.6 Nonideal loss characteristics: (*a*) Lowpass, (*b*) highpass.

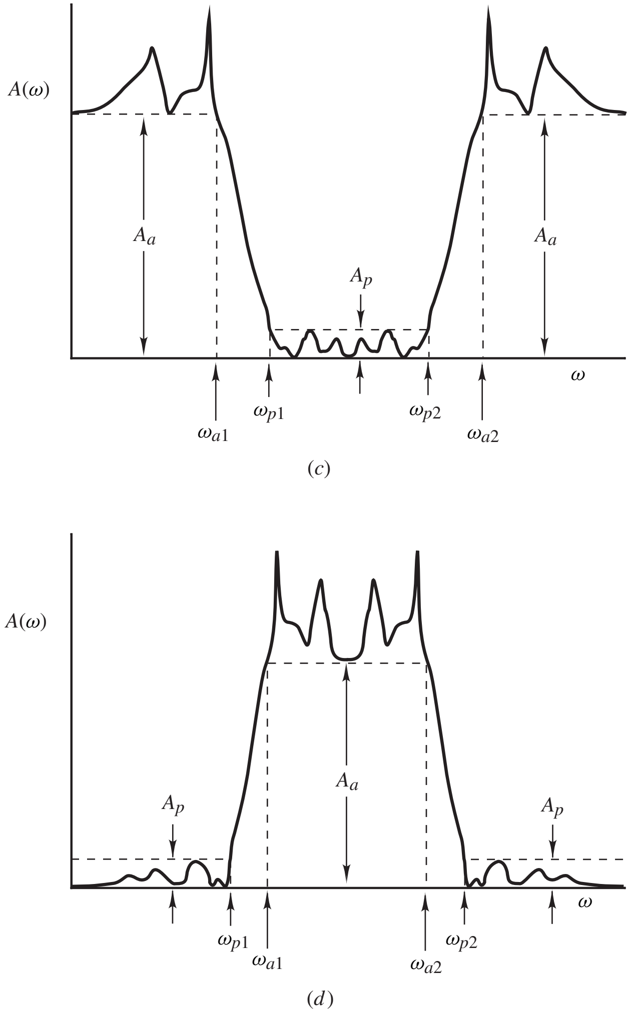


Figure 10.6 Cont'd Nonideal loss characteristics: (*c*) Bandpass, (*d*) bandstop.

10.2.7 Realizability Constraints

An *analog-filter approximation* is a realizable continuous-time transfer function such that the loss characteristic approaches one of the idealized characteristics in Fig. 10.5. A continuous-time transfer function is said to be *realizable* if it characterizes a stable and causal network. Such a transfer function is required to satisfy the following constraints:

1. It must be a rational function of s with real coefficients.
2. Its poles must lie in the left-half s plane.
3. The degree of the numerator polynomial must be equal to or less than that of the denominator polynomial.

In the following four sections, we focus our attention on *normalized* lowpass approximations; namely, Butterworth approximations in which the 3-dB cutoff frequency ω_c is equal to 1 rad/s, Chebyshev approximations in which the passband edge ω_p is equal to 1 rad/s, inverse-Chebyshev approximations in which the stopband edge ω_a is equal to 1 rad/s, elliptic approximations in which the cutoff frequency $\omega_c = \sqrt{(\omega_p\omega_a)}$ is equal to 1 rad/s, and Bessel-Thomson approximations in which the group delay as $\omega \to 0$ is equal to 1 s. Normalization keeps the sizes of numbers around unity which are easier to manage. Approximations for real-life practical filters can be obtained from the normalized ones through the use of transformations as described in Sec. 10.8. Approximations so obtained are sometimes said to be *denormalized.*

10.3 BUTTERWORTH APPROXIMATION

The simplest lowpass approximation, the *Butterworth* approximation, is derived by assuming that $L(\omega^2)$ is a polynomial of the form

$$L(\omega^2) = b_0 + b_1\omega^2 + \cdots + b_n\omega^{2n} \tag{10.16}$$

such that

$$\lim_{\omega^2\to 0} L(\omega^2) = 1$$

in a maximally flat sense.

10.3.1 Derivation

The Taylor series of $L(x+h)$, where $x = \omega^2$, is

$$L(x+h) = L(x) + h\frac{dL(x)}{dx} + \cdots + \frac{h^k}{k!}\frac{d^kL(x)}{dx^k}$$

The polynomial $L(x)$ approaches unity in a *maximally flat* sense as $x \to 0$ if its first $n-1$ derivatives are zero at $x = 0$. We may, therefore, assign

$$L(0) = 1$$

$$\left.\frac{d^kL(x)}{dx^k}\right|_{x=0} = 0 \qquad \text{for } k \le n-1$$

Thus from Eq. (10.16), we have

$$b_0 = 1 \qquad \text{and} \qquad b_1 = b_2 = \cdots = b_{n-1} = 0$$

or

$$L(\omega^2) = 1 + b_n\omega^{2n}$$

Now, for a normalized approximation in which

$$L(1) = 2$$

that is, $A(\omega) \approx 3\text{dB}$ at $\omega = 1$ rad/s, $b_n = 1$ and

$$L(\omega^2) = 1 + \omega^{2n} \tag{10.17}$$

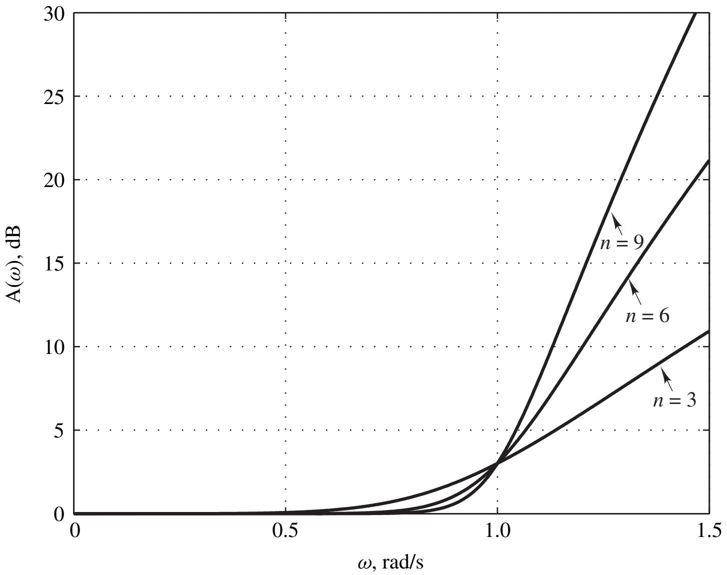


Figure 10.7 Typical Butterworth loss characteristics ($n = 3, 6, 9$).

Hence, the loss in a normalized lowpass Butterworth approximation is

$$A(\omega) = 10\log(1 + \omega^{2n}) \tag{10.18}$$

This is plotted in Fig. 10.7 for $n = 3, 6, 9$.

10.3.2 Normalized Transfer Function

With $\omega = s/j$ in Eq. (10.17), we have

$$L(-s^2) = 1 + (-s^2)^n = \prod_{i=1}^{2n}(s - z_i)$$

$$z_i = \begin{cases} e^{j(2i-1)\pi/2n} & \text{for even } n \\ e^{j(i-1)\pi/n} & \text{for odd } n \end{cases} \tag{10.19}$$

and since $|z_k| = 1$, the zeros of $L(-s^2)$ are located on the unit circle $|s| = 1$.

The normalized transfer function can be formed as

$$H_N(s) = \frac{1}{\prod_{i=1}^{n}(s - p_i)}$$

where p_i for $i = 1, 2, \ldots, n$ are the left-half s-plane zeros of $L(-s^2)$.

Example 10.1 Using the Butterworth approximation, find $H_N(s)$ for $(a)\ n = 5$ and $(b)\ n = 6$.

■ **Solution**

(a) For $n = 5$, Eq. (10.19) gives

$$\begin{aligned} z_i &= e^{j(i-1)\pi/5} \\ &= \cos\frac{(i-1)\pi}{5} + j\sin\frac{(i-1)\pi}{5} \end{aligned}$$

Hence, the zeros of the loss function are as follows:

$$z_1 = 1.0 \qquad z_2 = 0.809017 + j0.587785$$
$$z_3 = 0.309017 + j0.951057 \qquad z_4 = -0.309017 + j0.951057$$
$$z_5 = -0.809017 + j0.587785 \qquad z_6 = -1.0$$
$$z_7 = -0.809017 - j0.587785 \qquad z_8 = -0.309017 - j0.951057$$
$$z_9 = 0.309017 - j0.951057 \qquad z_{10} = 0.809017 - j0.587785$$

Dropping the right-hand s plane zeros of the loss function, we get

$$z_4 = -0.309017 + j0.951057 \qquad z_5 = -0.809017 + j0.587785$$
$$z_6 = -1.0 \qquad z_7 = -0.809017 - j0.587785$$
$$z_8 = -0.309017 - j0.951057$$

Now if we combine complex conjugate pairs of poles into factors, we obtain

$$\begin{aligned} H(s) = {} & \frac{1}{(s+1)} \cdot \frac{1}{(s+0.309017 - j0.951057)(s+0.309017 + j0.951057)} \\ & \cdot \frac{1}{(s+0.809017 - j0.587785)(s+0.809017 + j0.587785)} \\ = {} & \frac{1}{(s+1)} \cdot \frac{1}{(s^2+0.618034s+1)} \cdot \frac{1}{(s^2+1.618034s+1)} \quad ■ \end{aligned}$$

(b) Similarly, for $n = 6$, we have

$$\begin{aligned} z_i &= e^{j(2i-1)\pi/2n} \\ &= \cos\frac{(2i-1)\pi}{12} + j\sin\frac{(2i-1)\pi}{12} \end{aligned}$$

Hence

$$z_1 = 0.965928 + j0.258819 \qquad z_2 = 0.707107 + j0.707107$$
$$z_3 = 0.258819 + j0.965926 \qquad z_4 = -0.258819 + j0.965926$$

$$z_5 = -0.707107 + j0.707107 \qquad z_6 = -0.965928 + j0.258819$$
$$z_7 = -0.965928 - j0.258819 \qquad z_8 = -0.707107 - j0.707107$$
$$z_9 = -0.258819 - j0.965926 \qquad z_{10} = 0.258819 - j0.965926$$
$$z_{11} = 0.707107 - j0.707107 \qquad z_{12} = 0.965928 - j0.258819$$

Dropping the right-hand s plane zeros of the loss function, we get

$$z_4 = -0.258819 + j0.965926 \qquad z_5 = -0.707107 + j0.707107$$
$$z_6 = -0.965928 + j0.258819 \qquad z_7 = -0.965928 - j0.258819$$
$$z_8 = -0.707107 - j0.707107 \qquad z_9 = -0.258819 - j0.965926$$

Now if we combine complex conjugate pairs of poles into factors, we obtain

$$\begin{aligned} H(s) &= \frac{1}{(s + 0.258819 - j0.965926)(s + 0.258819 + j0.965926)} \\ &\quad \cdot \frac{1}{(s + 0.707107 - j0.707107)(s + 0.707107 + j0.707107)} \\ &\quad \cdot \frac{1}{(s + 0.965928 - j0.258819)(s + 0.965928 + j0.258819)} \\ &= \frac{1}{(s^2 + 0.517638s + 1)} \cdot \frac{1}{(s^2 + 1.414214s + 1)} \\ &\quad \cdot \frac{1}{(s^2 + 1.931852s + 1)} \quad \blacksquare \end{aligned}$$

The zero-pole plots of the loss function for the two examples are shown in Fig. 10.8.

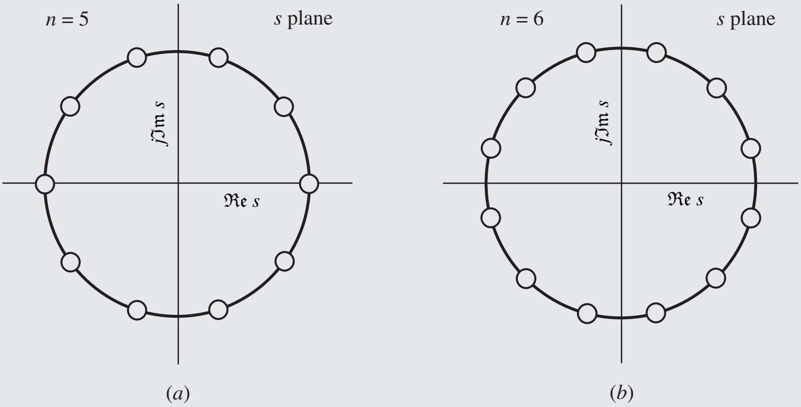


Figure 10.8 Zero-pole plots of loss function $L(-s^2)$ (Example 10.1).

10.3.3 Minimum Filter Order

Typically in practice, the required filter order is unknown. However, for Butterworth, Chebyshev, inverse-Chebyshev, and elliptic filters it can by easily deduced if the required specifications are known.

Let us assume that we need a Butterworth filter with a maximum passband loss A_p, minimum stopband loss A_a, passband edge ω_p, and stopband edge ω_a. As can be seen in Fig. 10.7, the loss in the Butterworth approximation is a monotonic increasing function, and thus the maximum passband loss occurs at the passband edge. Hence, we have

$$A(\omega_p) = 10\log(1+\omega_p^{2n}) \le A_p$$

Thus

$$\begin{aligned} 1+\omega_p^{2n} &\le 10^{A_p/10} \\ \omega_p^{2n} &\le 10^{0.1A_p} - 1 \\ 2n\log\omega_p &\le \log(10^{0.1A_p} - 1) \end{aligned}$$

For $\omega_p < 1$ and $A_p < 3.01$ dB, both sides in the above inequality are negative and if we express the above relation as

$$-2n\log\omega_p \ge -\log(10^{0.1A_p} - 1)$$

both sides will be positive. Solving for n, we get

$$n \ge \frac{[-\log(10^{0.1A_p} - 1)]}{(-2\log\omega_p)} \tag{10.20}$$

Similarly, the minimum stopband loss occurs at the stopband edge. Hence, n must be large enough to ensure that

$$A(\omega_a) = 10\log\left(1+\omega_a^{2n}\right) \ge A_a$$

Solving for n, we get

$$n \ge \frac{\log(10^{0.1A_a} - 1)}{2\log\omega_a} \tag{10.21}$$

In practice, we must, of course, satisfy both the passband and stopband specifications and, therefore, n must be chosen large enough to satisfy both Eq. (10.20) as well as Eq. (10.21).

It should be mentioned here that Eqs. (10.20) and (10.21) will not normally yield an integer but since the filter order *must* be an integer, the outcome of Eqs. (10.20) and (10.21) must be rounded up to the nearest integer. As a result of this rounding-up operation, the required specifications will be slightly oversatisfied. The actual maximum passband loss and actual minimum stopband loss can be found by evaluating the loss of the filter at the specified passband and stopband edges using Eq. (10.18).

Example 10.2 In an application a normalized Butterworth lowpass filter is required that would satisfy the following specification:

- Passband edge ω_p: 0.7 rad/s
- Stopband edge ω_a: 2.0 rad/s
- Maximum passband loss A_p: 0.5 dB
- Minimum stopband loss A_a: 30.0 dB

(*a*) Find the minimum filter order that would satisfy the specifications. (*b*) Calculate the actual maximum passband loss and minimum stopband loss. (*c*) Obtain the required transfer function.

■ **Solution**

(*a*) To ensure that the passband loss is equal to or greater than $A_p = 0.5$ dB the inequality in Eq. (10.20) must be satisfied, i.e.,

$$\begin{aligned} n &\geq \frac{[-\log(10^{0.1A_p} - 1)]}{(-2\log\omega_p)} \\ &\geq \frac{[-\log(10^{0.1*0.5} - 1)]}{(-2\log 0.7)} \\ &\geq 2.9489 \rightarrow 3 \end{aligned}$$

To ensure that the stopband loss is equal to or greater than $A_a = 30.0$ dB, Eq. (10.21) must be satisfied, i.e.,

$$\begin{aligned} n &\geq \frac{\log(10^{0.1A_a} - 1)}{2\log\omega_a} \\ &\geq \frac{\log(10^{0.1*30} - 1)}{2\log 2.0} \\ &\geq 4.9822 \rightarrow 5 \end{aligned}$$

In order to satisfy the passband as well as the stopband specifications, we choose the order n to be the larger of 3 and 5, that is, $n = 5$. ■

(*b*) Because of the monotonic increasing nature of the loss of the Butterworth approximation, the actual maximum passband loss occurs at the passband edge. Hence, Eq. (10.18) gives

$$A(\omega_p) = 10\log\left(1 + \omega_p^{2n}\right) = 10\log(1 + 0.7^{10}) = 0.1210 \text{ dB} \quad ■$$

Similarly, the actual minimum stopband loss occurs at the stopband edge and thus

$$A(\omega_a) = 10\log\left(1 + \omega_a^{2n}\right) = 10\log(1 + 2.0^{10}) = 30.11 \text{ dB} \quad ■$$

The Butterworth method, like the Bessel-Thomson method to follow, yields only one approximation for each filter order and, therefore, the required transfer function is the one found in Example 10.1, part (*a*).

10.4 CHEBYSHEV APPROXIMATION

In the Butterworth approximation, the loss is an increasing monotonic function of ω, and as a result the passband characteristic is lopsided, as can be seen in Fig. 10.7. A more balanced characteristic can be achieved by employing the *Chebyshev*[2] approximation in which the passband loss oscillates between zero and a prescribed maximum A_p. In effect, the Chebyshev approximation leads to a so-called *equiripple solution*.

10.4.1 Derivation

The loss characteristic in a fourth-order normalized Chebyshev approximation is of the form illustrated in Fig. 10.9, where $\omega_p = 1$. The loss is given by

$$A(\omega) = 10 \log L(\omega^2) \tag{10.22a}$$

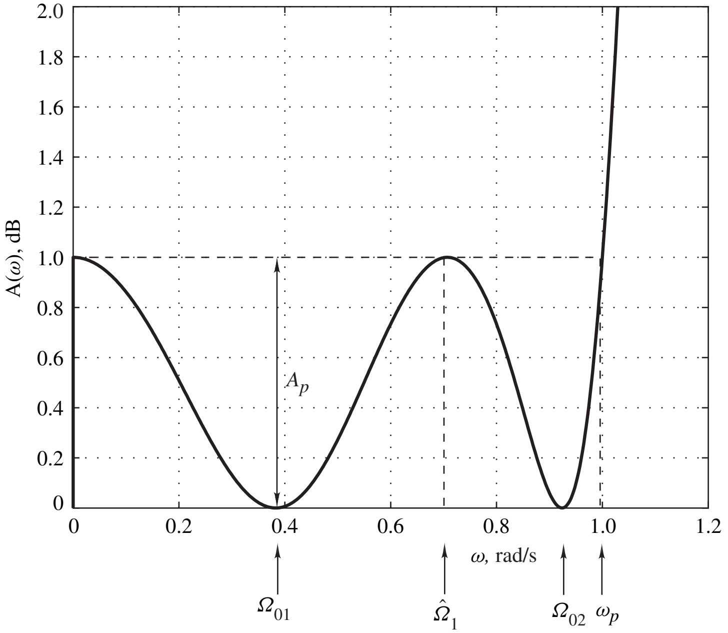


Figure 10.9 Loss characteristic of a fourth-order normalized Chebyshev filter.

[2] *Pafnuty Lvovitch Chebyshev (1821–1894) was a Russian mathematician who was born in Okatovo, a small town west of Moscow. In addition to his famous contribution to approximation theory, he contributed to number theory, integration, and probability theory, and studied the convergence of the Taylor series.*

where

$$L(\omega^2) = 1 + \varepsilon^2 F^2(\omega) \tag{10.22b}$$

and

$$\varepsilon^2 = 10^{0.1A_p} - 1 \tag{10.23}$$

$F(\omega)$, $L(\omega^2)$, and in turn $L(-s^2)$ are polynomials, and hence the normalized transfer function is of the form

$$H_N(s) = \frac{H_0}{D(s)}$$

where H_0 is a constant.

The derivation of $H_N(s)$ involves three general steps:

1. The exact form of $F(\omega)$ is deduced such that the desired loss characteristic is achieved.
2. The exact form of $L(\omega^2)$ is obtained.
3. The zeros of $L(-s^2)$ and, in turn, the poles of $H_N(s)$ are found.

Close examination of the Chebyshev loss characteristic depicted in Fig. 10.9 reveals that $F(\omega)$ and $L(\omega^2)$ must have the following properties:

Property 1: $F(\omega) = 0$ if $\omega = \pm\Omega_{01}, \pm\Omega_{02}$

Property 2: $F^2(\omega) = 1$ if $\omega = 0, \pm\hat{\Omega}_1, \pm 1$

Property 3: $\dfrac{dL(\omega^2)}{d\omega} = 0$ if $\omega = 0, \pm\Omega_{01}, \pm\hat{\Omega}_1, \pm\Omega_{02}$

From Property 1, $F(\omega)$ must be a polynomial of the form

$$F(\omega) = M_1\left(\omega^2 - \Omega_{01}^2\right)\left(\omega^2 - \Omega_{02}^2\right)$$

($M_1, M_2, \ldots, M_7$ represent miscellaneous constants in this analysis.) From Property 2, $1 - F^2(\omega)$ has zeros at $\omega = 0, \pm\hat{\Omega}_1, \pm 1$. Furthermore, the derivative of $1 - F^2(\omega)$ with respect to ω, namely,

$$\frac{d}{d\omega}[1 - F^2(\omega)] = -2F(\omega)\frac{dF(\omega)}{d\omega} = -\frac{1}{\varepsilon^2}\frac{dL(\omega^2)}{d\omega} \tag{10.24}$$

has zeros at $\omega = 0, \pm\Omega_{01}, \pm\hat{\Omega}_1, \pm\Omega_{02}$, according to Property 3. Consequently, $1 - F^2(\omega)$ must have at least double zeros at $\omega = 0, \pm\hat{\Omega}_1$. Therefore, we can write

$$1 - F^2(\omega) = M_2\omega^2\left(\omega^2 - \hat{\Omega}_1^2\right)^2(\omega^2 - 1)$$

Now from Eq. (10.24) and Properties 1 and 3, we get

$$\frac{dF(\omega)}{d\omega} = \frac{1}{2\varepsilon^2 F(\omega)} \frac{dL(\omega^2)}{d\omega} = M_3\omega\left(\omega^2 - \hat{\Omega}_1^2\right)$$

By combining the above results, we can form the differential equation

$$\left[\frac{dF(\omega)}{d\omega}\right]^2 = \frac{M_4[1 - F^2(\omega)]}{1 - \omega^2} \tag{10.25}$$

which is the basis of the fourth-order Chebyshev approximation.

The reader who is more interested in applying the Chebyshev approximation and less in its derivation can proceed to Sec. 10.4.3 where the general formulas for the nth-order Chebyshev approximation can be found.

To continue with the derivation, Eq. (10.25) can be expressed in terms of definite integrals as

$$M_5 \int_0^F \frac{dx}{\sqrt{1 - x^2}} + M_6 = \int_0^\omega \frac{dy}{\sqrt{1 - y^2}}$$

Hence, F and ω are interrelated by the equation

$$M_5 \cos^{-1} F + M_7 = \cos^{-1} \omega = \theta \tag{10.26}$$

i.e., for a given value of θ

$$\omega = \cos\theta \qquad \text{and} \qquad F = \cos\left(\frac{\theta}{M_5} - \frac{M_7}{M_5}\right)$$

What remains to be done is to determine constants M_5 and M_7. If $\omega = 0$, then $\theta = \pi/2$; and if $\omega = 1$, then $\theta = 0$, as depicted in Fig. 10.10. Now, F will correspond to $F(\omega)$ only if it has two zeros in the range $0 \le \theta \le \pi/2$ (Property 1), and its magnitude is unity if $\theta = 0, \pi/2$ (Property 2). Thus F must be of the form illustrated in Fig. 10.10. As can be seen, for $\theta = 0$

$$F = \cos\left(-\frac{M_7}{M_5}\right) = 1$$

or $M_7 = 0$. In addition, one period of F must be equal to one-quarter period of ω, that is,

$$2\pi M_5 = \frac{\pi}{2} \qquad \text{or} \qquad M_5 = \frac{1}{4}$$

Therefore, the exact form of $F(\omega)$ can be obtained from Eq. (10.26) as

$$F(\omega) = \cos(4\cos^{-1}\omega)$$

Alternatively, by expressing $\cos 4\theta$ in terms of $\cos\theta$, $F(\omega)$ can be put in the form

$$F(\omega) = 1 - 8\omega^2 + 8\omega^4$$

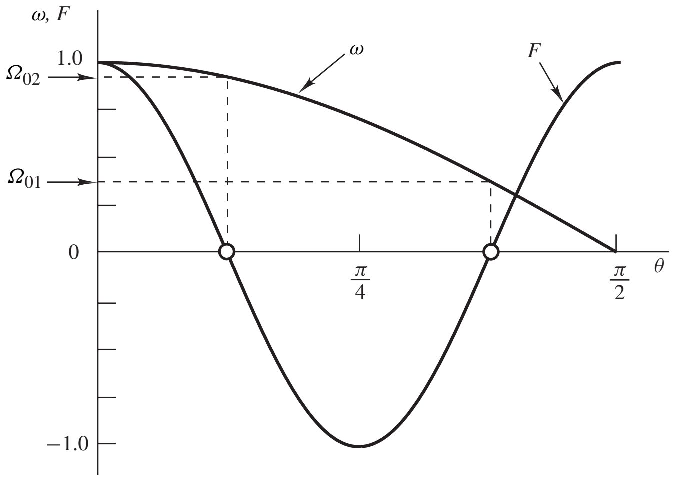


Figure 10.10 Plots of ω and F versus θ.

This polynomial is the fourth-order Chebyshev polynomial and is often designated as $T_4(\omega)$.[3]

Similarly, for an nth-order Chebyshev approximation, one can show that

$$F(\omega) = T_n(\omega) = \cos(n \cos^{-1} \omega)$$

and hence from Eq. (10.22b)

$$L(\omega^2) = 1 + \varepsilon^2[\cos(n \cos^{-1} \omega)]^2 \tag{10.27}$$

This relation gives the loss characteristic for $|\omega| \leq 1$. For $|\omega| > 1$, the quantity $\cos^{-1} \omega$ becomes complex, i.e.,

$$\cos^{-1} \omega = j\theta \tag{10.28}$$

and since

$$\omega = \cos j\theta = \frac{1}{2}(e^{j(j\theta)} + e^{-j(j\theta)}) = \cosh \theta$$

we have

$$\theta = \cosh^{-1} \omega$$

[3] *The use of T_n for the representation of Chebyshev polynomials has to do with the German spelling of the great mathematician's name, i.e., Tchebyscheff [2], which does not appear to be in use nowadays.*

Now from Eq. (10.28)

$$\cos^{-1}\omega = j\cosh^{-1}\omega$$

and

$$\cos(n\cos^{-1}\omega) = \cos(jn\cosh^{-1}\omega) = \cosh(n\cosh^{-1}\omega)$$

Thus for $|\omega| > 1$, Eq. (10.27) becomes

$$L(\omega^2) = 1 + \varepsilon^2[\cosh(n\cosh^{-1}\omega)]^2 \tag{10.29}$$

In summary, the loss in a normalized lowpass Chebyshev approximation is given by

$$A(\omega) = 10\log\left[1 + \varepsilon^2 T_n^2(\omega)\right] \tag{10.30}$$

where

$$T_n(\omega) = \begin{cases} \cos(n\cos^{-1}\omega) & \text{for } |\omega| \le 1 \\ \cosh(n\cosh^{-1}\omega) & \text{for } |\omega| > 1 \end{cases}$$

The loss characteristics for $n = 4$, $A_p = 1$ dB and $n = 7$, $A_p = 0.5$ dB are plotted in Fig. 10.11*a*. As can be seen

$$A(0) = \begin{cases} A_p & \text{for even } n \\ 0 & \text{for odd } n \end{cases}$$

as is generally the case in the Chebyshev approximation.

As an aside, note that in Fig. 10.11*a* the number of stationary points is exactly equal to the order of the approximation, that is, 4 or 7 for a fourth- or seventh-order approximation. This is a general property of the Chebyshev approximation which is imposed by the formulation of the approximation problem.

10.4.2 Zeros of Loss Function

With $\omega = s/j$, Eq. (10.29) becomes

$$L(-s^2) = 1 + \varepsilon^2\left[\cosh\left(n\cosh^{-1}\frac{s}{j}\right)\right]^2$$

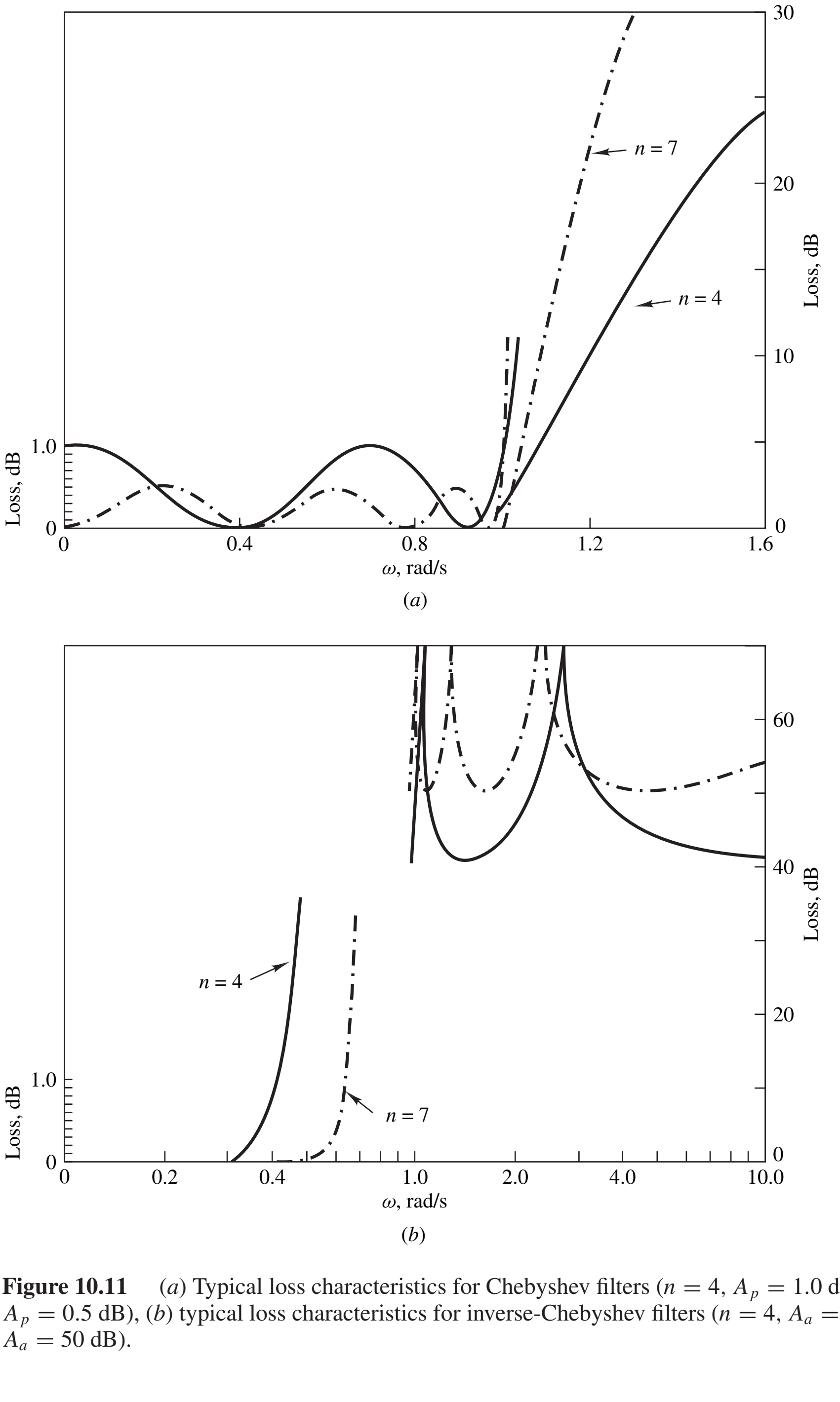
and if $s_i = \sigma_i + j\omega_i$ is a zero of $L(-s^2)$, we can write

$$u_i + jv_i = \cosh^{-1}(-j\sigma_i + \omega_i) \tag{10.31a}$$

$$\cosh[n(u_i + jv_i)] = \pm\frac{j}{\varepsilon} \tag{10.31b}$$

From Eq. (10.31a)

$$-j\sigma_i + \omega_i = \cosh(u_i + jv_i) = \cosh u_i \cos v_i + j\sinh u_i \sin v_i$$



(*a*)

(*b*)

Figure 10.11 (*a*) Typical loss characteristics for Chebyshev filters ($n = 4$, $A_p = 1.0$ dB and $n = 7$, $A_p = 0.5$ dB), (*b*) typical loss characteristics for inverse-Chebyshev filters ($n = 4$, $A_a = 40$ dB and $n = 7$, $A_a = 50$ dB).

or

$$\sigma_i = -\sinh u_i \sin v_i \tag{10.32}$$

and

$$\omega_i = \cosh u_i \cos v_i \tag{10.33}$$

Similarly, from Eq. (10.31b)

$$\cosh nu_i \cos nv_i + j \sinh nu_i \sin nv_i = \pm\frac{j}{\varepsilon}$$

or

$$\cosh nu_i \cos nv_i = 0 \tag{10.34a}$$

and

$$\sinh nu_i \sin nv_i = \pm\frac{1}{\varepsilon} \tag{10.34b}$$

The solution of Eq. (10.34a) is

$$v_i = \frac{(2i-1)\pi}{2n} \qquad \text{for } i = 1,\ 2,\ \ldots, n \tag{10.35a}$$

and since $\sin(nv_i) = \pm 1$, Eq. (10.34b) yields

$$u_i = u = \pm\frac{1}{n}\sinh^{-1}\frac{1}{\varepsilon} \tag{10.35b}$$

Therefore, from Eqs. (10.32), (10.33), (10.35a), and (10.35b)

$$\sigma_i = \pm\sinh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\varepsilon}\right)\sin\frac{(2i-1)\pi}{2n} \tag{10.36a}$$

$$\omega_i = \cosh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\varepsilon}\right)\cos\frac{(2i-1)\pi}{2n} \tag{10.36b}$$

for $i = 1, 2, \ldots, n$. Evidently,

$$\frac{\sigma_i^2}{\sinh^2 u} + \frac{\omega_i^2}{\cosh^2 u} = 1$$

i.e., the zeros of $L(-s^2)$ are located on an ellipse, as depicted in Fig. 10.12.

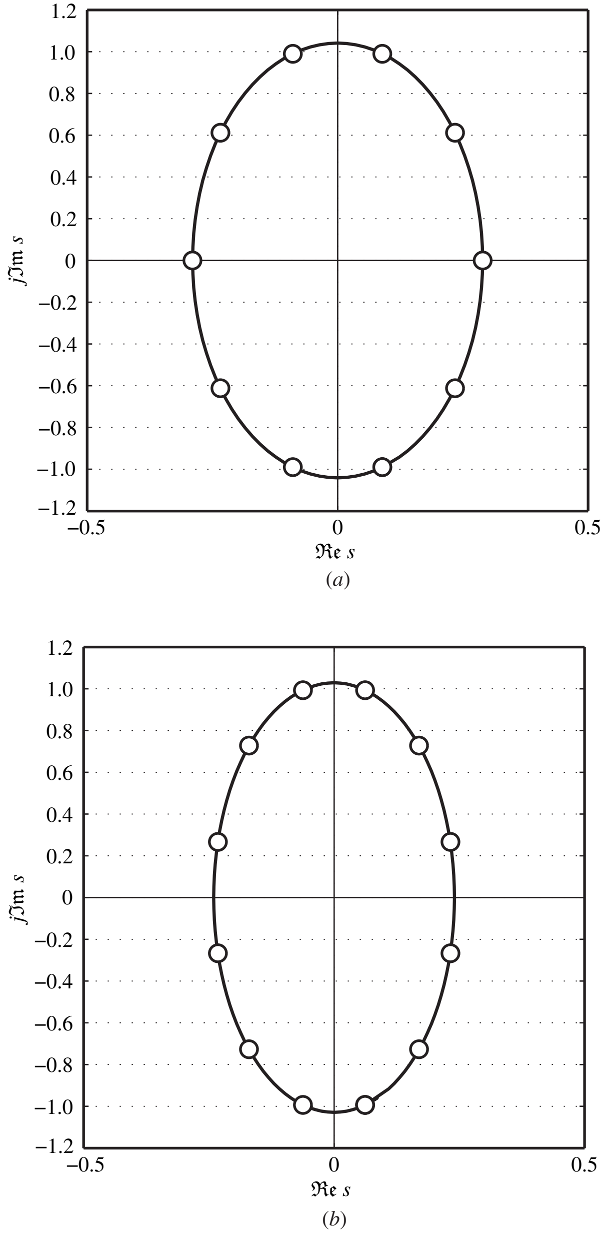


Figure 10.12 Zero-pole plot of $L(-s^2)$ for Chebyshev filter: (*a*) $n = 5$, $A_p = 1$ dB, (*b*) $n = 6$, $A_p = 1$ dB.

10.4.3 Normalized Transfer Function

The normalized transfer function $H_N(s)$ can at this point be formed by identifying the left-half s-plane zeros of the loss function, which happen to be the poles of the transfer function, as

$$H_N(s) = \frac{H_0}{D(s)\prod_i^r (s - p_i)(s - p_i^*)} \tag{10.37a}$$

$$= \frac{H_0}{D(s)\prod_i^r [s^2 - 2\,\Re e(p_i)s + |p_i|^2]} \tag{10.37b}$$

where

$$r = \begin{cases} \dfrac{n-1}{2} & \text{for odd } n \\ \dfrac{n}{2} & \text{for even } n \end{cases} \qquad \text{and} \qquad D_0(s) = \begin{cases} s - p_0 & \text{for odd } n \\ 1 & \text{for even } n \end{cases}$$

The poles and multiplier constant, H_0, can be calculated by using the following formulas in sequence:

$$\varepsilon = \sqrt{10^{0.1A_p} - 1} \tag{10.38}$$

$$p_0 = \sigma_{(n+1)/2} \tag{10.39}$$

with

$$\sigma_{(n+1)/2} = -\sinh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\varepsilon}\right) \tag{10.40}$$

$$p_i = \sigma_i + j\omega_i \qquad \text{for } i = 1, 2, \ldots, r \tag{10.41}$$

with

$$\sigma_i = -\sinh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\varepsilon}\right)\sin\frac{(2i-1)\pi}{2n} \tag{10.42a}$$

$$\omega_i = \cosh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\varepsilon}\right)\cos\frac{(2i-1)\pi}{2n} \tag{10.42b}$$

and

$$H_0 = \begin{cases} -p_0 \prod_{i=1}^r |p_i|^2 & \text{for odd } n \\ 10^{-0.05A_p}\prod_{i=1}^r |p_i|^2 & \text{for even } n \end{cases} \tag{10.43}$$

In the above formulation, constant H_0 is chosen to yield zero minimum passband loss. Formulas for the required hyperbolic functions and their inverses can be found in Sec. A.3.4.

Example 10.3 Obtain a fourth-order normalized Chebyshev approximation assuming a maximum passband loss of $A_p = 1.0$ dB.

■ **Solution**

From Eq. (10.23)

$$\frac{1}{\varepsilon} = x = \frac{1}{\sqrt{10^{0.1} - 1}} = 1.965227$$

and
$$\sinh^{-1}\frac{1}{\varepsilon} = \ln(x + \sqrt{x^2 + 1}) = 1.427975$$

Hence, Eqs. (10.42a) and (10.42b) give

$$\sigma_i = -0.364625 \sin\frac{(2i-1)\pi}{8}$$
$$\omega_i = 1.064402 \cos\frac{(2i-1)\pi}{8}$$

and from Eqs. (10.41) and (10.43), the poles and multiplier constant can be obtained as

$$p_1, p_1^* = -0.139536 \pm j0.983379$$
$$p_2, p_2^* = -0.336870 \pm j0.407329$$
$$H_0 = 10^{-0.05\times 1}\prod_{i=1}^{2}|p_i|^2 = 0.245653$$

Since $D_0(s) = 1$ for an even-order Chebyshev approximation, Eq. (10.37b) gives the required transfer function as

$$H_N(s) = H_0 \prod_{i=1}^{2}\frac{1}{s^2 + b_{1i}s + b_{0i}} \quad ■$$

where

$$b_{01} = 0.986505 \qquad b_{11} = 0.279072$$
$$b_{02} = 0.279398 \qquad b_{12} = 0.673740$$

10.4.4 Minimum Filter Order

In a normalized lowpass Chebyshev transfer function, the passband edge is fixed at $\omega_p = 1$ rad/s and an arbitrary maximum passband loss A_p dB can be achieved. Since the stopband loss is an increasing monotonic function of frequency as can be seen in Fig. 10.11a, the minimum stopband loss occurs at the stopband edge. From Eq. (10.30), we have

$$\begin{aligned} A(\omega_a) &= 10\log\left[1+\varepsilon^2 T_n^2(\omega_a)\right] \\ &= 10\log\left\{1+\varepsilon^2\left[\cosh\left(n\cosh^{-1}\omega_a\right)\right]^2\right\} \end{aligned} \tag{10.44}$$

Since the minimum stopband loss must be equal to or exceed A_a, we have

$$\begin{aligned} 10\log\{1+\varepsilon^2[\cosh(n\cosh^{-1}\omega_a)]^2\} &\geq A_a \\ 1+\varepsilon^2[\cosh(n\cosh^{-1}\omega_a)]^2 &\geq 10^{0.1A_a} \\ \cosh(n\cosh^{-1}\omega_a) &\geq \frac{\sqrt{10^{0.1A_a}-1}}{\varepsilon} \end{aligned}$$

and on eliminating ε using Eq. (10.23) and then solving for n, we obtain

$$n \geq \frac{\cosh^{-1}\sqrt{D}}{\cosh^{-1}\omega_a} \tag{10.45a}$$

where

$$D = \frac{10^{0.1A_a}-1}{10^{0.1A_p}-1} \tag{10.45b}$$

The required filter order is the lowest integer that would satisfy the above inequality. Once the filter order is determined, the actual minimum stopband loss can be obtained by substituting back the filter order in Eq. (10.44).

Example 10.4 An application calls for a normalized lowpass Chebyshev filter that would satisfy the following specifications:

- Passband edge ω_p: 1.0 rad/s
- Stopband edge ω_a: 2.0 rad/s
- Maximum passband loss A_p: 0.1 dB
- Minimum stopband loss A_a: 34.0 dB

(*a*) Find the minimum filter order. (*b*) Obtain the required transfer function. (*c*) Calculate the actual minimum stopband loss.

■ **Solution**

(*a*) From Eq. (10.45b)

$$D = \frac{10^{0.1\times 34}-1}{10^{0.1\times 0.1}-1} = 1.077958\times 10^5$$

Hence, Eq. (10.45a) gives

$$n \geq \frac{\cosh^{-1}\sqrt{1.077958\times 10^5}}{\cosh^{-1} 2.0} = 4.93 \rightarrow 5 \quad \blacksquare$$

(*b*) From Eq. (10.38), we have

$$\varepsilon = \sqrt{10^{0.1\times 0.1} - 1} = 0.152620 \qquad \text{or} \qquad \frac{1}{\varepsilon} = x = 6.552203$$

and
$$\sinh^{-1}\frac{1}{\varepsilon} = \ln(x + \sqrt{x^2+1}) = 2.578722$$

From Eqs. (10.40) and (10.41), we get

$$\sigma_3 = -0.538914$$

and

$$\sigma_i = -0.5389143 \sin\frac{(2i-1)\pi}{10}$$
$$\omega_i = 1.135970 \cos\frac{(2i-1)\pi}{10}$$

Thus Eqs. (10.39), (10.41), and (10.43) give the poles and multiplier constant as

$$p_0 = -0.538914$$
$$p_1, p_1^* = -0.166534 \pm j1.080372$$
$$p_2, p_2^* = -0.435991 \pm j0.667707$$
$$H_0 = -p_0 \prod_{i=1}^{2} |p_i|^2 = 0.409513$$

Therefore, from Eq. (10.37b) the required transfer function is obtained as

$$H_N(s) = \frac{H_0}{s + b_{00}} \prod_{i=1}^{2} \frac{1}{s^2 + b_{1i}s + b_{0i}} \qquad \blacksquare$$

where

$$b_{00} = 0.538914$$
$$b_{01} = 1.194937 \qquad b_{11} = 0.333067$$
$$b_{02} = 0.635920 \qquad b_{12} = 0.871982$$

(*c*) The actual minimum stopband loss can be obtained by evaluating the stopband loss at the stopband edge using the actual filter order. From Eq. (10.44), we get

$$A(\omega_a) = 10\log\{1 + (0.152620)^2[\cosh(5\cosh^{-1} 2.0)]^2\} = 34.85 \text{ dB} \qquad \blacksquare$$

10.5 INVERSE-CHEBYSHEV APPROXIMATION

A closely related approximation to the above is the *inverse-Chebyshev* approximation. This can actually be derived from the Chebyshev approximation but the derivation is left as an exercise to the reader (see Prob. 10.12). The passband loss in the inverse-Chebyshev is very similar to that of the Butterworth approximation, i.e., it is an increasing monotonic function of ω, while the stopband loss oscillates between infinity and a prescribed minimum loss A_a, as depicted in Fig. 10.11*b*. The loss is given by

$$A(\omega) = 10 \log \left[1 + \frac{1}{\delta^2 T_n^2(1/\omega)} \right] \tag{10.46}$$

where

$$\delta^2 = \frac{1}{10^{0.1A_a} - 1} \tag{10.47}$$

and the stopband extends from $\omega = 1$ to ∞.

10.5.1 Normalized Transfer Function

The normalized transfer function has a number of zeros on the $j\omega$ axis in this case and is given by

$$H_N(s) = \frac{H_0}{D_0(s)} \prod_{i=1}^{r} \frac{(s - 1/z_i)(s - 1/z_i^*)}{(s - 1/p_i)(s - 1/p_i^*)} \tag{10.48a}$$

$$= \frac{H_0}{D_0(s)} \prod_{i=1}^{r} \frac{s^2 - 2\,\Re e\left(\frac{1}{z_i}\right) s + \frac{1}{|z_i|^2}}{s^2 - 2\,\Re e\left(\frac{1}{p_i}\right) s + \frac{1}{|p_i|^2}} \tag{10.48b}$$

$$= \frac{H_0}{D_0(s)} \prod_{i=1}^{r} \frac{s^2 + \frac{1}{|z_i|^2}}{s^2 - 2\,\Re e\left(\frac{1}{p_i}\right) s + \frac{1}{|p_i|^2}} \tag{10.48c}$$

where

$$r = \begin{cases} \frac{n-1}{2} & \text{for odd } n \\ \frac{n}{2} & \text{for even } n \end{cases} \tag{10.48d}$$

and

$$D_0(s) = \begin{cases} s - \frac{1}{p_0} & \text{for odd } n \\ 1 & \text{for even } n \end{cases} \tag{10.48e}$$

If the filter order n and minimum stopband loss A_a are known, the multiplier constant H_0 and zeros and poles or transfer function coefficient can be obtained by using the following formulas in sequence:

$$\delta = \frac{1}{\sqrt{10^{0.1A_a} - 1}} \tag{10.49}$$

$$z_i = j \cos \frac{(2i-1)\pi}{2n} \qquad \text{for } 1,\ 2,\ \ldots,\ r \tag{10.50}$$

$$p_0 = \sigma_{(n+1)/2} \tag{10.51}$$

with

$$\sigma_{(n+1)/2} = -\sinh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\delta}\right) \tag{10.52}$$

$$p_i = \sigma_i + j\omega_i \qquad \text{for } 1,\ 2,\ \ldots,\ r \tag{10.53}$$

with

$$\sigma_i = -\sinh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\delta}\right)\sin\frac{(2i-1)\pi}{2n} \tag{10.54a}$$

$$\omega_i = \cosh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\delta}\right)\cos\frac{(2i-1)\pi}{2n} \tag{10.54b}$$

and

$$H_0 = \begin{cases} \frac{1}{-p_0}\prod_{i=1}^{r}\frac{|z_i|^2}{|p_i|^2} & \text{for odd } n \\ \prod_{i=1}^{r}\frac{|z_i|^2}{|p_i|^2} & \text{for even } n \end{cases} \tag{10.55}$$

The derivation of $H_N(s)$ is left as an exercise for the reader (see Prob. 10.9).

10.5.2 Minimum Filter Order

In a normalized lowpass inverse-Chebyshev transfer function, the stopband edge is fixed at $\omega_a = 1$ rad/s and an arbitrary minimum stopband loss A_a dB can be achieved for any given order. The minimum filter order is thus determined by the maximum loss allowed in the passband, namely, A_p dB.

The highest passband loss occurs at the passband edge and from Eq. (10.46)

$$\begin{aligned} A(\omega_p) &= 10\log\left[1+\frac{1}{\delta^2 T_n^2(1/\omega_p)}\right] \\ &= 10\log\left[1+\frac{1}{\delta^2[\cosh(n\cosh^{-1}1/\omega_p)]^2}\right] \end{aligned} \tag{10.56}$$

Hence, the minimum filter order must satisfy the inequality

$$10\log\left[1+\frac{1}{\delta^2[\cosh(n\cosh^{-1}1/\omega_p)]^2}\right] \le A_p$$

and if we solve for n, we obtain

$$n \ge \frac{\cosh^{-1}\sqrt{D}}{\cosh^{-1}(1/\omega_p)} \tag{10.57a}$$

where

$$D = \frac{10^{0.1A_a}-1}{10^{0.1A_p}-1} \tag{10.57b}$$

The minimum filter order is the lowest integer that would satisfy the above inequality. The actual maximum passband loss can be obtained by substituting the filter order obtained back in Eq. (10.56).

Example 10.5 An application requires a normalized lowpass inverse-Chebyshev filter that would satisfy the following specifications:

- Passband edge ω_p: 0.6 rad/s
- Stopband edge ω_a: 1.0 rad/s
- Maximum passband loss A_p: 1.0 dB
- Minimum stopband loss A_a: 35.0 dB

(*a*) Find the minimum filter order. (*b*) Obtain the required transfer function. (*c*) Calculate the actual maximum passband loss.

■ **Solution**

(*a*) From Eq. (10.57b)

$$D = \frac{10^{0.1\times 35.0} - 1}{10^{0.1\times 1.0} - 1} = 1.2209 \times 10^4$$

Hence, Eq. (10.57a) yields

$$n \geq \frac{\cosh^{-1}\sqrt{1.2209 \times 10^4}}{\cosh^{-1}\left(\frac{1}{0.6}\right)} = \frac{5.3981}{1.0986} = 4.9136 \rightarrow 5 \quad \blacksquare$$

(*b*) From Eqs. (10.48d) and (10.48e), we have

$$r = (n-1)/2 = 2 \qquad \text{and} \qquad D_0(s) = s - 1/p_0$$

and from Eqs. (10.49)–(10.55), we get

$$\begin{aligned}
\delta &= \frac{1}{\sqrt{10^{0.1\times 35.0} - 1}} = 0.017786\\
\sigma_3 &= -\sinh\left(\frac{1}{5}\sinh^{-1}\frac{1}{0.017786}\right) = -1.091354\\
z_i &= j\cos\frac{(2i-1)\pi}{10}\\
\sigma_i &= -\sinh\left(\frac{1}{5}\sinh^{-1}\frac{1}{0.017786}\right)\sin\frac{(2i-1)\pi}{10}\\
&= -1.091354\sin\frac{(2i-1)\pi}{10}\\
\omega_i &= \cosh\left(\frac{1}{5}\sinh^{-1}\frac{1}{0.017786}\right)\cos\frac{(2i-1)\pi}{2n}\\
&= 1.480221\cos\frac{(2i-1)\pi}{10}
\end{aligned}$$

Hence,

$$\begin{aligned}
p_0 &= \sigma_3 = -1.091354 \\
z_1 &= j\cos\frac{\pi}{10} = j0.951057 \\
z_2 &= j\cos\frac{3\pi}{10} = j0.587785 \\
p_1 &= -1.091354\sin\frac{\pi}{10} + j1.480221\cos\frac{\pi}{10} \\
&= -0.337247 + j1.407774 \\
p_2 &= -1.091354\sin\frac{3\pi}{10} + j1.480221\cos\frac{3\pi}{10} \\
&= -0.882924 + j0.870052
\end{aligned}$$

Therefore, the transfer function in Eq. (10.48c) assumes the form

$$\begin{aligned}
H_N(s) &= \frac{H_0\left(s^2+\frac{1}{|z_1|^2}\right)\left(s^2+\frac{1}{|z_2|^2}\right)}{\left(s-\frac{1}{\sigma_3}\right)\left[s^2-2\,\Re e\left(\frac{1}{p_1}\right)s+\frac{1}{|p_1|^2}\right]\left[s^2-2\,\Re e\left(\frac{1}{p_2}\right)s+\frac{1}{|p_2|^2}\right]} \\
&= \frac{H_0(s^2+a_{01})(s^2+a_{02})}{(s+b_{00})(s^2+b_{11}s+b_{01})(s^2+b_{12}s+b_{02})} \quad \blacksquare
\end{aligned}$$

where

$$a_{01} = \frac{1}{|z_1|^2} = 1.105573 \qquad a_{02} = \frac{1}{|z_2|^2} = 2.894427$$

$$b_{00} = -\frac{1}{\sigma_3} = 0.916293$$

$$b_{01} = \frac{1}{|p_1|^2} = 0.477199 \qquad b_{11} = -2\,\Re e\left(\frac{1}{p_1}\right) = 0.321868$$

$$b_{02} = \frac{1}{|p_2|^2} = 0.650811 \qquad b_{12} = -2\,\Re e\left(\frac{1}{p_2}\right) = 1.149232$$

$$\begin{aligned}
H_0 &= \frac{1}{-p_0}\prod_{i=1}^{2}\frac{|z_i|^2}{|p_i|^2} \\
&= 0.088928
\end{aligned}$$

(*c*) From Eq. (10.56), the maximum passband loss can be determined by evaluating the loss at the passband edge as

$$\begin{aligned}
A(\omega_p) &= 10\log\left\{1+\frac{1}{0.017786^2\left[\cosh\left(5\cosh^{-1}\frac{1}{0.6}\right)\right]^2}\right\} \\
&= 0.8427\text{ dB} \quad \blacksquare
\end{aligned}$$

10.6 ELLIPTIC APPROXIMATION

The Chebyshev approximation yields a much better passband characteristic and the inverse-Chebyshev approximation yields a much better stopband characteristic than the Butterworth approximation. A filter with an improved passband as well as an improved stopband loss characteristic can be obtained by using the *elliptic* approximation in which the passband loss oscillates between zero and a prescribed maximum A_p and the stopband loss oscillates between infinity and a prescribed minimum A_a.

The elliptic approximation is more efficient than the preceding two in that the transition between passband and stopband is steeper for a given approximation order.

Our approach to this approximation follows the formulation of Grossman [9], which, although involved, is probably the simplest available. The approach taken is first to deduce the fifth-order approximation and then generalize the results obtained to the nth odd-order approximation. After that the nth even-order approximation is given without the derivation. The section concludes with a practical procedure for obtaining elliptic transfer functions that would satisfy prescribed filter specifications.

10.6.1 Fifth-Order Approximation

The loss characteristic in a fifth-order normalized elliptic approximation is of the form depicted in Fig. 10.13, where

$$\omega_p = \sqrt{k} \qquad \omega_a = \frac{1}{\sqrt{k}} \qquad \omega_c = \sqrt{\omega_a \omega_p} = 1$$

The constants k and k_1 given by

$$k = \frac{\omega_p}{\omega_a}$$

and

$$k_1 = \left(\frac{10^{0.1A_p} - 1}{10^{0.1A_a} - 1} \right)^{1/2} \tag{10.58}$$

are the *selectivity factor* and *discrimination factor*, respectively. The loss is given by

$$A(\omega) = 10 \log L(\omega^2)$$

where

$$L(\omega^2) = 1 + \varepsilon^2 F^2(\omega) \tag{10.59}$$

and

$$\varepsilon^2 = 10^{0.1A_p} - 1 \tag{10.60}$$

Function $F(\omega)$ and in turn $L(\omega^2)$, $L(-s^2)$, and $H(s)$, which are polynomials in the Chebyshev approximation, are ratios of polynomials in the case of the elliptic approximation.

According to the elliptic loss characteristic of Fig. 10.13, the prerequisite properties of $F(\omega)$ and $L(\omega^2)$ are as follows:

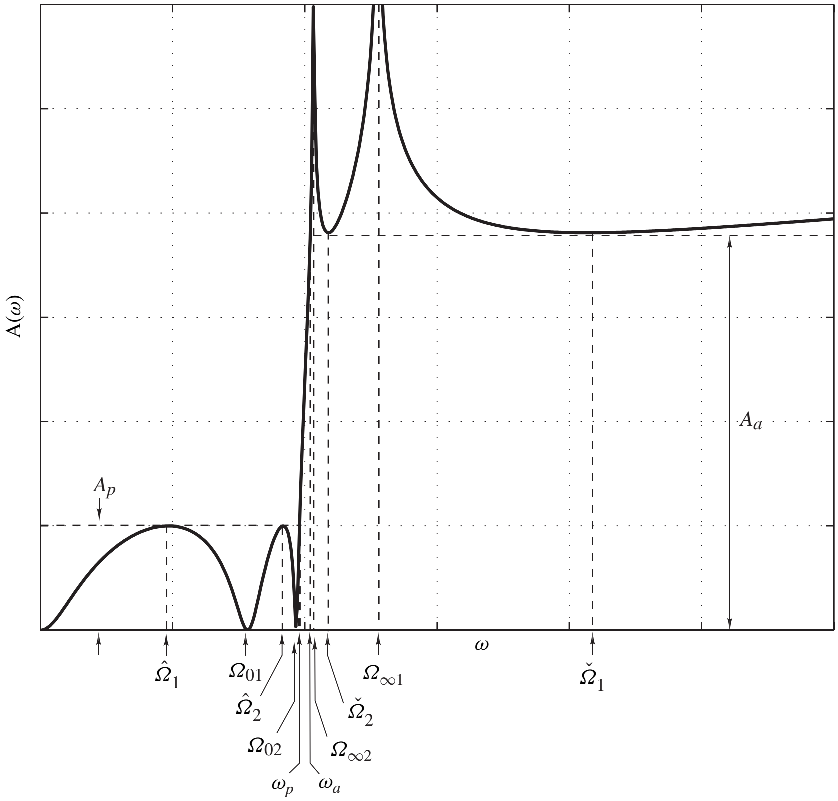


Figure 10.13 Loss characteristic of a fifth-order elliptic filter.

Property 1: $F(\omega) = 0$ if $\omega = 0, \pm\Omega_{01}, \pm\Omega_{02}$

Property 2: $F(\omega) = \infty$ if $\omega = \infty, \pm\Omega_{\infty 1}, \pm\Omega_{\infty 2}$

Property 3: $F^2(\omega) = 1$ if $\omega = \pm\hat{\Omega}_1, \pm\hat{\Omega}_2, \pm\sqrt{k}$

Property 4: $F^2(\omega) = \dfrac{1}{k_1^2}$ if $\omega = \pm\check{\Omega}_1, \pm\check{\Omega}_2, \pm\frac{1}{\sqrt{k}}$

Property 5: $\dfrac{dL(\omega^2)}{d\omega} = 0$ if $\omega = \pm\hat{\Omega}_1, \pm\hat{\Omega}_2, \pm\check{\Omega}_1, \pm\check{\Omega}_2$

By using each and every one of these properties we shall attempt to derive the exact form of $F(\omega)$. The approach is analogous to that used earlier in the Chebyshev approximation.[4]

From Properties 1 and 2, we obtain

$$F(\omega) = \frac{M_1\omega\left(\omega^2 - \Omega_{01}^2\right)\left(\omega^2 - \Omega_{02}^2\right)}{\left(\omega^2 - \Omega_{\infty 1}^2\right)\left(\omega^2 - \Omega_{\infty 2}^2\right)} \tag{10.61}$$

[4] *The DSP practitioner who is more interested in applying the elliptic approximation and less so in its derivation may proceed to Sec. 10.6.6 for the outcome of this exercise in mathematics.*

(M_1 to M_7 represent miscellaneous unknown constants that arise in the formulation of the problem at hand). Similarly, from Properties 2 and 3, we can write

$$1 - F^2(\omega) = \frac{M_2\left(\omega^2 - \hat{\Omega}_1^2\right)^2\left(\omega^2 - \hat{\Omega}_2^2\right)^2(\omega^2 - k)}{\left(\omega^2 - \Omega_{\infty 1}^2\right)^2\left(\omega^2 - \Omega_{\infty 2}^2\right)^2}$$

where the double zeros at $\omega = \pm\hat{\Omega}_1, \pm\hat{\Omega}_2$ are due to Property 5 (see Sec. 10.4.1). Similarly, from Properties 2, 4, and 5

$$1 - k_1^2 F^2(\omega) = \frac{M_3\left(\omega^2 - \check{\Omega}_1^2\right)^2\left(\omega^2 - \check{\Omega}_2^2\right)^2(\omega^2 - 1/k)}{\left(\omega^2 - \Omega_{\infty 1}^2\right)^2\left(\omega^2 - \Omega_{\infty 2}^2\right)^2}$$

and from Property 5

$$\frac{dF(\omega)}{d\omega} = \frac{M_4\left(\omega^2 - \hat{\Omega}_1^2\right)\left(\omega^2 - \hat{\Omega}_2^2\right)\left(\omega^2 - \check{\Omega}_1^2\right)\left(\omega^2 - \check{\Omega}_2^2\right)}{\left(\omega^2 - \Omega_{\infty 1}^2\right)^2\left(\omega^2 - \Omega_{\infty 2}^2\right)^2}$$

By combining the above results, we can form the important relation

$$\left[\frac{dF(\omega)}{d\omega}\right]^2 = \frac{M_5[1 - F^2(\omega)][1 - k_1^2 F^2(\omega)]}{(1 - \omega^2/k)(1 - k\omega^2)} \tag{10.62}$$

Alternatively, we can write

$$\int_0^F \frac{dx}{\sqrt{(1 - x^2)\left(1 - k_1^2 x^2\right)}} = \sqrt{M_5}\int_0^\omega \frac{dy}{\sqrt{(1 - y^2/k)(1 - ky^2)}} + M_7$$

and if $y = \sqrt{k}y'$, $y' = y$

$$\int_0^F \frac{dx}{\sqrt{(1 - x^2)(1 - k_1^2 x^2)}} = M_6\int_0^{\omega/\sqrt{k}} \frac{dy}{\sqrt{(1 - y^2)(1 - k^2 y^2)}} + M_7$$

These are *elliptic integrals* of the first kind, and they can be put in the more convenient form

$$\int_0^{\phi_1} \frac{d\theta_1}{\sqrt{1 - k_1^2 \sin^2\theta_1}} = M_6\int_0^\phi \frac{d\theta}{\sqrt{1 - k^2\sin^2\theta}} + M_7$$

by using the transformations

$$x = \sin\theta_1 \qquad F = \sin\phi_1 \qquad y = \sin\theta \qquad \frac{\omega}{\sqrt{k}} = \sin\phi$$

The above two integrals can assume complex values if complex values are allowed for ϕ_1 and ϕ. By letting

$$\int_0^{\phi} \frac{d\theta}{\sqrt{1-k^2\sin^2\theta}} = z \qquad \text{where } z = u + jv$$

the solution of the differential equation in Eq. (10.62) can be expressed in terms of a pair of simultaneous equations as

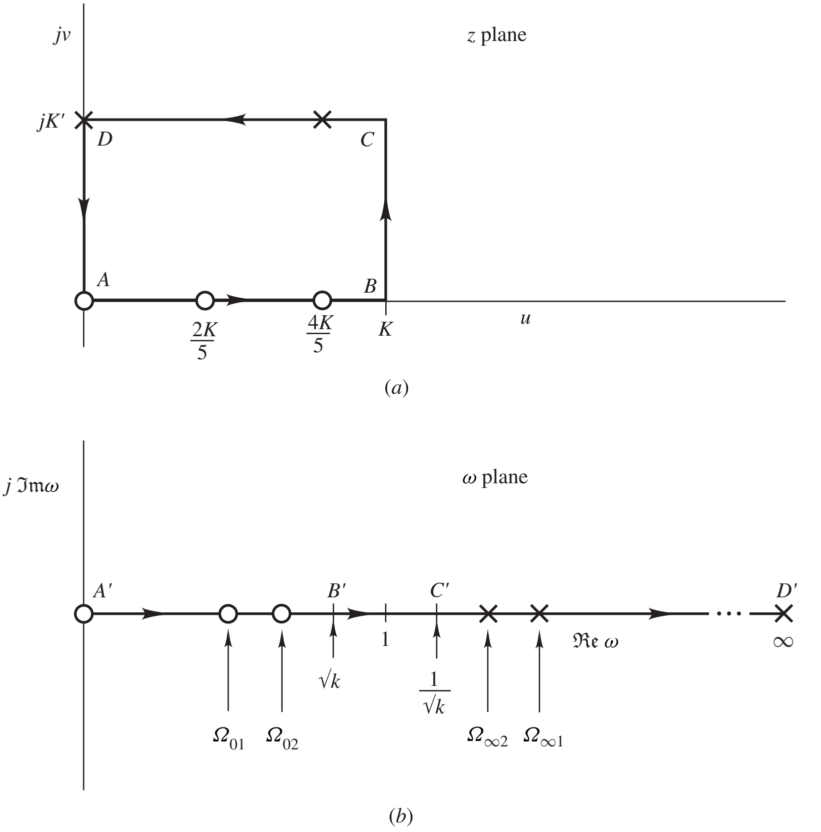
$$\frac{\omega}{\sqrt{k}} = \sin\phi = \operatorname{sn}(z, k) \tag{10.63}$$

$$F = \sin\phi_1 = \operatorname{sn}(M_6 z + M_7, k_1) \tag{10.64}$$

The entities at the right-hand side are *elliptic functions*.

Further progress in this analysis can be made by using the properties of elliptic functions as detailed in Appendix B.

As demonstrated in Sec. B.7, Eq. (10.63) is a transformation that maps trajectory $ABCD$ in Fig. 10.14*a* onto the positive real axis of the ω plane, as depicted in Fig. 10.14*b*. Since the behavior



(*a*)

(*b*)

Figure 10.14 Mapping properties of Eq. (10.63).

of $F(\omega)$ is known for all real values of ω, constants M_6 and M_7 can be determined. In turn, the exact form of $F(\omega)$ can be derived.

If $z = u$ and $0 \le u \le K$ (domain 1 in Sec. B.7), Eqs. (10.63) and (10.64) become

$$\omega = \sqrt{k}\,\mathrm{sn}(u, k) \tag{10.65}$$

$$F = \mathrm{sn}(M_6 u + M_7, k_1) \tag{10.66}$$

where ω and F have real periods of $4K$ and $4K_1/M_6$, respectively (see Sec. B.6). If $\omega = 0$, then $u = 0$; and if $\omega = \sqrt{k}$, then $u = K$, as illustrated in Fig. 10.15. Now, F will correspond to $F(\omega)$ if it has zeros at $u = 0$ and at two other points in the range $0 < u \le K$ (Property 1), and its magnitude is unity at $u = K$ (Property 3). Consequently, F must be of the form illustrated in Fig. 10.15. Clearly, for $u = 0$

$$F = \mathrm{sn}(M_7, k_1) = 0$$

or $M_7 = 0$. Furthermore, five quarter periods of F must be equal to one quarter period of ω, that is,

$$M_6 = \frac{5K_1}{K}$$

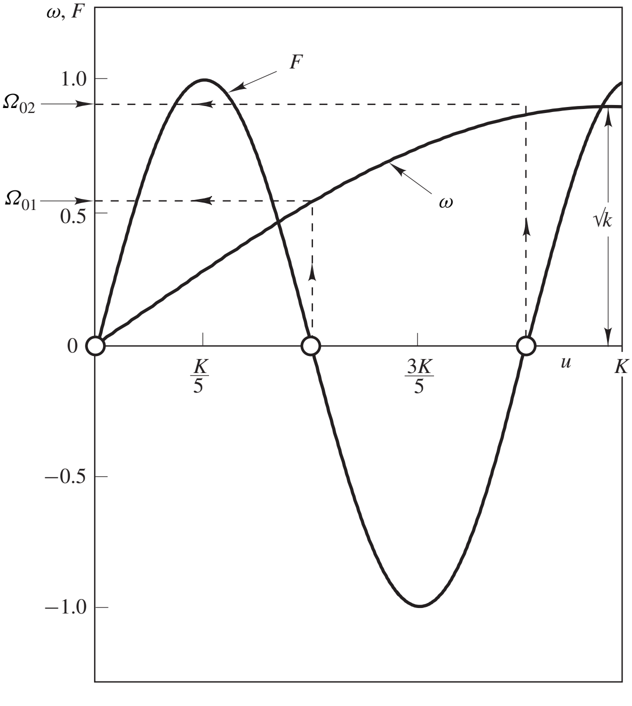


Figure 10.15 Plots of ω and F versus u.

and so from Eq. (10.66)

$$F = \text{sn}\left(\frac{5K_1 u}{K}, k_1\right)$$

Now F has z-plane zeros at

$$u = \frac{2Ki}{5} \qquad \text{for } i = 0, 1, 2$$

and, therefore, $F(\omega)$ must have ω-plane zeros (zero-loss frequencies) at

$$\Omega_{0i} = \sqrt{k}\,\text{sn}\left(\frac{2Ki}{5}, k\right) \qquad \text{for } i = 0, 1, 2$$

according to Eq. (10.65) (see Fig. 10.14).

If $z = u + jK'$ and $0 \le u \le K$ (domain 3 in Sec. B.7), Eqs. (10.63) and (10.64) assume the form

$$\omega = \frac{1}{\sqrt{k}\,\text{sn}(u, k)} \tag{10.67}$$

$$F = \text{sn}\left[\frac{5K_1(u + jK')}{K}, k_1\right] \tag{10.68}$$

If $\omega = \infty$, $u = 0$ and F must be infinite (Property 2), that is,

$$F = \text{sn}\left(\frac{j5K_1K'}{K}, k_1\right) = \infty$$

and from Eq. (B.19)

$$F = \frac{j\,\text{sn}(5K_1K'/K, k_1')}{\text{cn}(5K_1K'/K, k_1')} = \infty \qquad \text{where } k_1' = \sqrt{1 - k_1^2}$$

Hence, it is necessary that

$$\text{cn}\left(\frac{5K_1K'}{K}, k_1'\right) = 0$$

and, therefore, the relation

$$\frac{5K'}{K} = \frac{K_1'}{K_1} \tag{10.69}$$

must hold. The quantities K, K' are functions of k, and similarly K_1, K_1' are functions of k_1; in turn, k_1 is a function of A_p and A_a by definition. In effect, Eq. (10.69) constitutes an implicit constraint among filter specifications. We shall assume here that Eq. (10.69) holds. The implications of this assumption will be examined at a later point.

With Eq. (10.69) satisfied, Eq. (10.68) becomes

$$F = \operatorname{sn}\left(\frac{5K_1}{K}u + jK_1', k_1\right)$$

and after some manipulation

$$F = \frac{1}{k_1 \operatorname{sn}(5K_1 u/K, k_1)}$$

Evidently, $F = \infty$ if

$$u = 2Ki/5 \qquad \text{for } i = 0, 1, 2 \tag{10.70}$$

that is, F has poles at

$$z = \frac{2Ki}{5} + jK' \qquad \text{for } i = 0, 1, 2$$

as depicted in Fig. 10.14, and since line CD maps onto line $C'D'$, F corresponds to $F(\omega)$. That is, $F(\omega)$ has two poles in the range $1/\sqrt{k} \le \omega < \infty$ and one at $\omega = \infty$ (Property 2). The poles of $F(\omega)$ (infinite-loss frequencies) can be obtained from Eqs. (10.67) and (10.70) as

$$\Omega_{\infty i} = \frac{1}{\sqrt{k}\operatorname{sn}(2Ki/5, k)} \qquad \text{for } i = 0, 1, 2$$

Therefore, the infinite-loss frequencies are the reciprocals of the zero-loss frequencies, i.e.,

$$\Omega_{\infty i} = \frac{1}{\Omega_{0i}}$$

and by eliminating $\Omega_{\infty i}$ in Eq. (10.61), we have

$$F(\omega) = \frac{M_1'\omega\left(\omega^2 - \Omega_{01}^2\right)\left(\omega^2 - \Omega_{02}^2\right)}{\left(1 - \omega^2\Omega_{01}^2\right)\left(1 - \omega^2\Omega_{02}^2\right)} \tag{10.71}$$

The only unknown at this point is constant M_1'. With $z = K + jv$ and $0 \le v \le K'$ (domain 2 in Sec. B.7), Eqs. (10.63) and (10.64) can be put in the form

$$\omega = \frac{\sqrt{k}}{\operatorname{dn}(v, k')} \qquad \text{and} \qquad F = \operatorname{sn}\left[\frac{5K_1(K + jv)}{K}, k_1\right]$$

If $\omega = 1$, then $v = K'/2$ and $F(1) = M_1'$, according to Eq. (10.71). Hence,

$$M_1' = \operatorname{sn}\left(5K_1 + j\frac{5K'K_1}{2K}, k_1\right) \qquad \text{or} \qquad M_1' = \operatorname{sn}\left(K_1 + \frac{jK_1'}{2}, k_1\right)$$

according to Eqs. (10.69) and (B.8) and after some manipulation, we get

$$M_1' = \frac{1}{\mathrm{dn}(K_1'/2, k_1')} = \frac{1}{\sqrt{k_1}}$$

10.6.2 *N*th-Order Approximation (*n* Odd)

For an nth-order approximation with n odd, constant M_7 in Eq. (10.64) is zero, and n quarter periods of F must correspond to one quarter period of ω, that is,

$$M_6 = \frac{nK_1}{K}$$

Therefore, Eq. (10.64) assumes the form

$$F = \mathrm{sn}\left(\frac{nK_1 z}{K}, k_1\right) \tag{10.72}$$

where the relation

$$\frac{nK'}{K} = \frac{K_1'}{K_1}$$

must hold. The expression for $F(\omega)$ can be shown to be

$$F(\omega) = \frac{(-1)^r \omega}{\sqrt{k_1}} \prod_{i=1}^{r} \frac{\omega^2 - \Omega_i^2}{1 - \omega^2 \Omega_i^2}$$

where

$$r = \frac{n-1}{2}$$

and $$\Omega_i = \sqrt{k}\,\mathrm{sn}\left(\frac{2Ki}{n}, k\right) \qquad \text{for } i = 1, 2, \ldots, r$$

10.6.3 Zeros and Poles of $L(-s^2)$

The next task is to determine the zeros and poles of $L(-s^2)$. From Eqns. (10.59) and (10.72), the z-domain representation of the loss function can be expressed as

$$L(z) = 1 + \varepsilon^2\,\mathrm{sn}^2\left(\frac{nK_1 z}{K}, k_1\right)$$

and by factorizing

$$L(z) = \left[1 + j\varepsilon\,\mathrm{sn}\left(\frac{nK_1 z}{K}, k_1\right)\right]\left[1 - j\varepsilon\,\mathrm{sn}\left(\frac{nK_1 z}{K}, k_1\right)\right]$$

If z_1 is a root of the first factor, $-z_1$ must be a root of the second factor since the elliptic sine is an odd function of z. Consequently, the zeros of $L(z)$ can be determined by solving the equation

$$\mathrm{sn}\left(\frac{nK_1z}{K}, k_1\right) = \frac{j}{\varepsilon}$$

In practice, the value of k_1 is very small. For example, $k_1 \le 0.0161$ if $A_p \le 1$ dB and $A_a \ge 30$ dB and decreases further if A_p is reduced or A_a is increased. We can thus assume that $k_1 = 0$, in which case

$$\mathrm{sn}\left(\frac{nK_1z}{K}, 0\right) = \sin\frac{nK_1z}{K} = \frac{j}{\varepsilon}$$

where $K_1 = \pi/2$, according to Eq. (B.2). Alternatively,

$$-j\frac{n\pi z}{2K} = \sinh^{-1}\frac{1}{\varepsilon}$$

and on using the identity

$$\sinh^{-1} x = \ln(x + \sqrt{x^2 + 1})$$

and Eq. (10.60), we obtain one zero of $L(z)$ as

$$z_0 = jv_0$$

where

$$v_0 = \frac{K}{n\pi}\ln\frac{10^{0.05A_p} + 1}{10^{0.05A_p} - 1}$$

Now sn $(nK_1z/K, k_1)$ has a real period of $4K/n$, and as a result all z_i given by

$$z_i = z_0 + \frac{4Ki}{n} \qquad \text{for } i = 0, 1, 2, \ldots$$

must also be zeros of $L(z)$.

The zeros of $L(\omega^2)$ can be deduced by using the transformation between the z and ω planes, namely, Eq. (10.63). In turn, the zeros of $L(-s^2)$ can be obtained by letting $\omega = s/j$. For $i = 0$, there is a real zero of $L(-s^2)$ at $s = \sigma_0$, where

$$\sigma_0 = j\sqrt{k}\,\mathrm{sn}(jv_0, k) \tag{10.73}$$

and for $i = 1, 2, \ldots, n-1$ there are $n-1$ distinct complex zeros at $s = \sigma_i + j\omega_i$, where

$$\sigma_i + j\omega_i = j\sqrt{k}\,\mathrm{sn}\left(jv_0 + \frac{4Ki}{n}, k\right) \tag{10.74}$$

The remaining n zeros are negatives of zeros already determined.

For $n = 5$, the required values of the elliptic sine are

$$\begin{aligned}
&\operatorname{sn}\left(jv_0 + \frac{4K}{5}\right)\\
&\operatorname{sn}\left(jv_0 + \frac{8K}{5}\right) = \operatorname{sn}\left(jv_0 + 2K - \frac{2K}{5}\right) = -\operatorname{sn}\left(jv_0 - \frac{2K}{5}\right)\\
&\operatorname{sn}\left(jv_0 + \frac{12K}{5}\right) = \operatorname{sn}\left(jv_0 + 2K + \frac{2K}{5}\right) = -\operatorname{sn}\left(jv_0 + \frac{2K}{5}\right)\\
&\operatorname{sn}\left(jv_0 + \frac{16K}{5}\right) = \operatorname{sn}\left(jv_0 + 4K - \frac{4K}{5}\right) = \operatorname{sn}\left(jv_0 - \frac{4K}{5}\right)
\end{aligned}$$

Hence, Eq. (10.74) can be put in the form

$$\sigma_i + j\omega_i = j\sqrt{k}(-1)^i \operatorname{sn}\left(jv_0 \pm \frac{2Ki}{5}, k\right) \qquad \text{for } i = 1, 2$$

Similarly, for any odd value of n

$$\sigma_i + j\omega_i = j\sqrt{k}(-1)^i \operatorname{sn}\left(jv_0 \pm \frac{2Ki}{n}, k\right) \qquad \text{for } i = 1, 2, \ldots, \frac{n-1}{2}$$

Now with the aid of the addition formula (see Sec. B.5) we can show that

$$\sigma_i + j\omega_i = \frac{(-1)^i \sigma_0 V_i \pm j\Omega_i W}{1 + \sigma_0^2 \Omega_i^2} \qquad \text{for } i = 1, 2, \ldots, \frac{n-1}{2}$$

where

$$W = \sqrt{\left(1 + k\sigma_0^2\right)\left(1 + \frac{\sigma_0^2}{k}\right)} \tag{10.75}$$

$$V_i = \sqrt{\left(1 - k\Omega_i^2\right)\left(1 - \frac{\Omega_i^2}{k}\right)} \tag{10.76}$$

$$\Omega_i = \sqrt{k}\, \operatorname{sn}\left(\frac{2Ki}{n}, k\right) \tag{10.77}$$

A complete description of $L(-s^2)$ is available at this point. It has zeros at $s = \pm\sigma_0, \pm(\sigma_i + j\omega_i)$ and double poles at $s = \pm j/\Omega_i$, which can be evaluated by using the series representation of elliptic functions given in Sec. B.8. From Eq. (10.73) and (B.30), we have

$$\sigma_0 = \frac{-2q^{1/4} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \sinh[(2m+1)\Lambda]}{1 + 2\sum_{m=1}^{\infty} (-1)^m q^{m^2} \cosh 2m\Lambda} \tag{10.78}$$

where

$$\Lambda = \frac{1}{2n} \ln \frac{10^{0.05A_p} + 1}{10^{0.05A_p} - 1}$$

The parameter q, which is known as the *modular constant*, is given by

$$q = e^{-\pi K'/K} \tag{10.79}$$

Similarly, from Eqs. (10.77) and (B.30)

$$\Omega_i = \frac{2q^{1/4} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \sin \frac{(2m+1)\pi i}{n}}{1 + 2\sum_{m=1}^{\infty} (-1)^m q^{m^2} \cos \frac{2m\pi i}{n}} \tag{10.80}$$

for $i = 1, 2, \ldots, (n-1)/2$. The modular constant q can be determined by evaluating K and K' numerically. A quicker method, however, is to use the following procedure.

Since $\text{dn}(0, k) = 1$, Eq. (B.32) gives

$$\sqrt{k'} = \frac{1 - 2q + 2q^4 - 2q^9 + \cdots}{1 + 2q + 2q^4 + 2q^9 + \cdots} \tag{10.81}$$

Now, $q < 1$ since $K, K' > 0$, and hence a first approximation for q is

$$q_0 = \frac{1}{2} \left(\frac{1 - \sqrt{k'}}{1 + \sqrt{k'}} \right)$$

By eliminating $\sqrt{k'}$ using Eq. (10.81), rationalizing, and then performing long division we have

$$q \approx q_0 + 2q^5 - 5q^9 + 10q^{13}$$

Thus, if q_{m-1} is an approximation for q

$$q_m \approx q_0 + 2q_{m-1}^5 - 5q_{m-1}^9 + 10q_{m-1}^{13}$$

is a better approximation. By using this recursive relation repeatedly we can show that

$$q \approx q_0 + 2q_0^5 + 15q_0^9 + 150q_0^{13}$$

Since k is known, the quantities $k', q_0, q, \sigma_0, \Omega_i, \sigma_i$, and ω_i can be evaluated. Subsequently, the normalized transfer function $H_N(s)$ can be formed.

10.6.4 *N*th-Order Approximation (*n* Even)

So far we have been concerned with odd-order approximations. However, the results can be easily extended to the case of even n.

Function F is of the form

$$F = \text{sn}\left(\frac{nK_1}{K} z + K_1, k_1 \right)$$

where the relation

$$\frac{nK'}{K} = \frac{K_1'}{K_1}$$

must again hold. The expression for $F(\omega)$ in this case is given by

$$F(\omega) = \frac{(-1)^r}{\sqrt{k_1}} \prod_{i=1}^{r} \frac{\omega^2 - \Omega_i^2}{1 - \omega^2 \Omega_i^2}$$

where

$$r = \frac{n}{2} \quad \text{and} \quad \Omega_i = \sqrt{k}\, \text{sn}\left[\frac{(2i-1)K}{n}, k\right] \qquad \text{for } i = 1, 2, \ldots, r$$

The zeros of $L(-s^2)$ are

$$s_i = \pm(\sigma_i + j\omega_i)$$

where

$$\sigma_i + j\omega_i = \frac{\pm[\sigma_0 V_i + j(-1)^i \Omega_i W]}{1 + \sigma_0^2 \Omega_i^2}$$

The parameters W, V_i, and σ_0 are given by Eqs. (10.75), (10.76), and (10.78), as in the case of odd n, and the values of Ω_i can be computed by replacing i by $i - \frac{1}{2}$ in the right-hand side of Eq. (10.80).

10.6.5 Specification Constraint

The results of the preceding sections are based on the assumption that the relation

$$\frac{nK'}{K} = \frac{K_1'}{K_1} \tag{10.82}$$

holds. As pointed out earlier, this equation constitutes a constraint among filter specifications of the form

$$f_1(n, k) = f_2(A_p, A_a)$$

Consequently, if three of the four parameters are specified, the fourth is automatically fixed. It is thus of interest to put Eq. (10.82) in a more useful form that can be used to evaluate the corresponding fourth parameter.

From the definition of the elliptic sine $\text{sn}(K_1, k_1) = 1$ and from Eq. (B.30)

$$k_1 = 4\sqrt{q_1}\left(\frac{1 + q_1^2 + q_1^6 + \cdots}{1 + 2q_1 + 2q_1^4 + \cdots}\right)^2 \qquad \text{where} \quad q_1 = e^{-\pi K_1'/K_1}$$

In practice, k_1 is close to zero, k'_1 is close to unity, K'_1/K_1 is large, and, as a result, $q_1 \ll 1$. Hence, we can assume that

$$k_1 \approx 4\sqrt{q_1} \qquad \text{or} \qquad k_1^2 = 16q_1 = 16e^{-\pi K'_1/K_1}$$

By eliminating K'_1/K_1, using Eq. (10.82), we have

$$k_1^2 = 16e^{-\pi n K'/K}$$

and from Eq. (10.79)

$$k_1^2 = 16q^n$$

Therefore, from Eq. (10.58) the desired formula is

$$\frac{10^{0.1A_p} - 1}{10^{0.1A_a} - 1} = 16q^n \tag{10.83}$$

If n, k, and A_p are specified, the resulting minimum stopband loss is given by

$$A_a = 10\log\left(\frac{10^{0.1A_p} - 1}{16q^n} + 1\right) \tag{10.84}$$

The minimum stopband loss A_a is plotted versus k in Fig. 10.16*a* for various values of A_p in the range $0.125 \leq A_p \leq 5$ dB. On the other hand, Fig. 10.16*b* shows A_a versus k for various values of n in the range $2 \leq n \leq 10$. We note in Fig. 10.16*a* and *b* that for a fixed maximum passband loss or a fixed filter order, the minimum stopband loss is reduced if we attempt to increase the selectivity, i.e., make the transition characteristic between the passband and stopband steeper.

Alternatively, if k, A_a, and A_p are specified, the required approximation order must satisfy the inequality

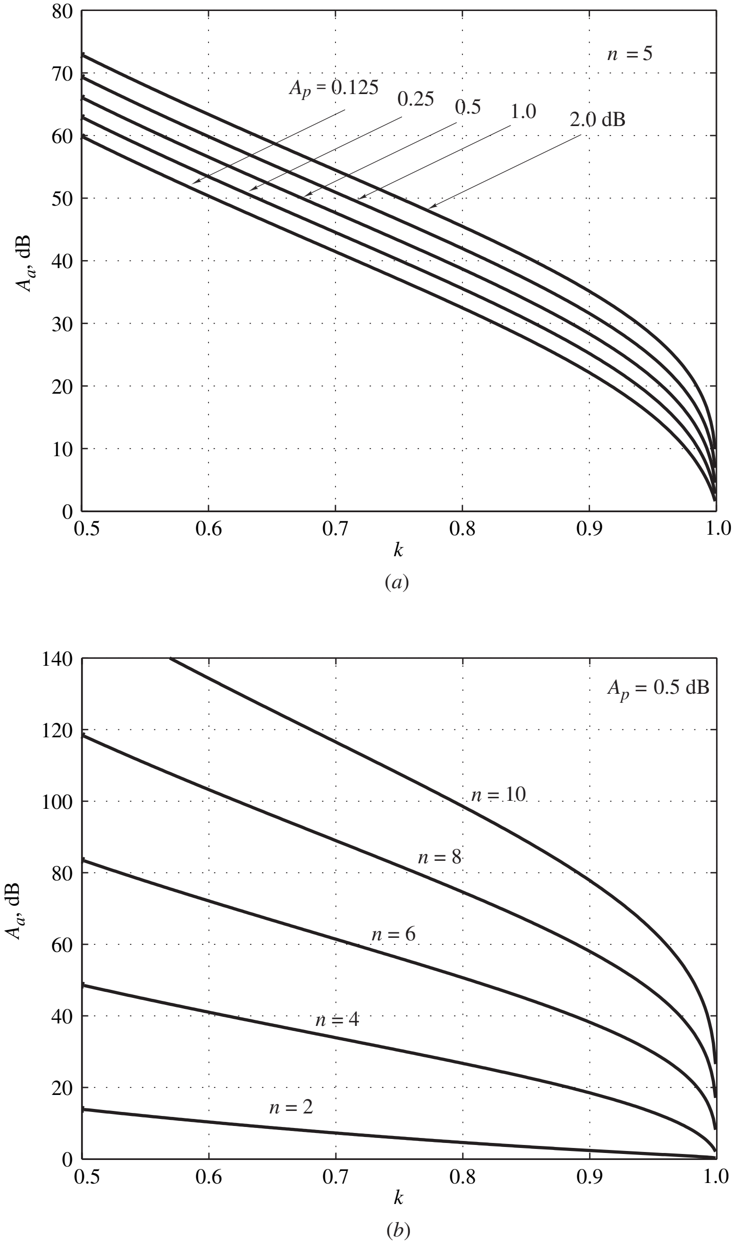
$$n \geq \frac{\log 16D}{\log(1/q)} \qquad \text{where} \;\; D = \frac{10^{0.1A_a} - 1}{10^{0.1A_p} - 1}$$

10.6.6 Normalized Transfer Function

The results obtained through the previous mathematical roller coaster can now be summarized in layman's language for the DSP practitioner.

An elliptic normalized lowpass filter with a selectivity factor k, a maximum passband loss of A_p dB, and a minimum stopband loss equal to or in excess of A_a dB has a transfer function of the form

$$H_N(s) = \frac{H_0}{D_0(s)} \prod_{i=1}^{r} \frac{s^2 + a_{0i}}{s^2 + b_{1i}s + b_{0i}} \tag{10.85}$$



(*a*)

(*b*)

Figure 10.16 Plots of A_a versus k: (*a*) $n = 5$, $A_p = 0.125$, 0.25, 0.5, 1.0, 2.0 dB, (*b*) $A_p = 0.5$ dB, $n = 2$, 4, 6, 8, 10.

where
$$r = \begin{cases} \dfrac{n-1}{2} & \text{for odd } n \\ \dfrac{n}{2} & \text{for even } n \end{cases}$$

and
$$D_0(s) = \begin{cases} s + \sigma_0 & \text{for odd } n \\ 1 & \text{for even } n \end{cases}$$

The transfer-function coefficients and multiplier constant H_0 can be computed by using the following formulas in sequence:

$$k' = \sqrt{1-k^2} \tag{10.86}$$

$$q_0 = \frac{1}{2}\left(\frac{1-\sqrt{k'}}{1+\sqrt{k'}}\right) \tag{10.87}$$

$$q = q_0 + 2q_0^5 + 15q_0^9 + 150q_0^{13} \tag{10.88}$$

$$D = \frac{10^{0.1A_a}-1}{10^{0.1A_p}-1} \tag{10.89}$$

$$n \geq \frac{\log 16D}{\log(1/q)} \tag{10.90}$$

$$\Lambda = \frac{1}{2n}\ln\frac{10^{0.05A_p}+1}{10^{0.05A_p}-1} \tag{10.91}$$

$$\sigma_0 = \left|\frac{2q^{1/4}\sum_{m=0}^{\infty}(-1)^m q^{m(m+1)}\sinh[(2m+1)\Lambda]}{1+2\sum_{m=1}^{\infty}(-1)^m q^{m^2}\cosh 2m\Lambda}\right| \tag{10.92}$$

$$W = \sqrt{\left(1+k\sigma_0^2\right)\left(1+\frac{\sigma_0^2}{k}\right)} \tag{10.93}$$

$$\Omega_i = \frac{2q^{1/4}\sum_{m=0}^{\infty}(-1)^m q^{m(m+1)}\sin\frac{(2m+1)\pi\mu}{n}}{1+2\sum_{m=1}^{\infty}(-1)^m q^{m^2}\cos\frac{2m\pi\mu}{n}} \tag{10.94}$$

where

$$\mu = \begin{cases} i & \text{for odd } n \\ i-\frac{1}{2} & \text{for even } n \end{cases} \qquad i = 1, 2, \ldots, r$$

$$V_i = \sqrt{\left(1-k\Omega_i^2\right)\left(1-\frac{\Omega_i^2}{k}\right)} \tag{10.95}$$

$$a_{0i} = \frac{1}{\Omega_i^2} \tag{10.96}$$

$$b_{0i} = \frac{(\sigma_0 V_i)^2 + (\Omega_i W)^2}{\left(1+\sigma_0^2\Omega_i^2\right)^2} \tag{10.97}$$

$$b_{1i} = \frac{2\sigma_0 V_i}{1 + \sigma_0^2 \Omega_i^2} \tag{10.98}$$

$$H_0 = \begin{cases} \sigma_0 \prod_{i=1}^{r} \dfrac{b_{0i}}{a_{0i}} & \text{for odd } n \\ 10^{-0.05A_p} \prod_{i=1}^{r} \dfrac{b_{0i}}{a_{0i}} & \text{for even } n \end{cases} \tag{10.99}$$

The actual minimum stopband loss is given by Eq. (10.84). The series in Eqs. (10.92) and (10.94) converge rapidly, and three or four terms are sufficient for most purposes.

Example 10.6 An elliptic filter is required satisfying the following specifications:

- Passband edge ω_p: $\sqrt{0.9}$ rad/s
- Stopband edge ω_a: $1/\sqrt{0.9}$ rad/s
- Maximum passband loss A_p: 0.1 dB
- Minimum stopband loss A_a: 50.0 dB

Form $H_N(s)$.

■ **Solution**

From Eqs. (10.86)–(10.90)

$$k = 0.9 \qquad k' = 0.435890 \qquad q_0 = 0.102330$$

$$q = 0.102352 \qquad D = 4{,}293{,}090 \qquad n \geq 7.92 \qquad \text{or} \qquad n = 8$$

From Eqs. (10.91)–(10.99) the transfer-function coefficients in Table 10.1 can be obtained. The corresponding loss characteristic is plotted in Fig. 10.17. The actual value of A_a is 50.82 dB according to Eq. (10.84). ■

Table 10.1 Coefficients of $H_N(s)$ (Example 10.6)

i	a_{0i}	b_{0i}	b_{1i}
1	$1.434825E+1$	$2.914919E-1$	$8.711574E-1$
2	2.231643	$6.123726E-1$	$4.729136E-1$
3	1.320447	$8.397386E-1$	$1.825141E-1$
4	1.128832	$9.264592E-1$	$4.471442E-2$
$H_0 = 2.876332E-3$			

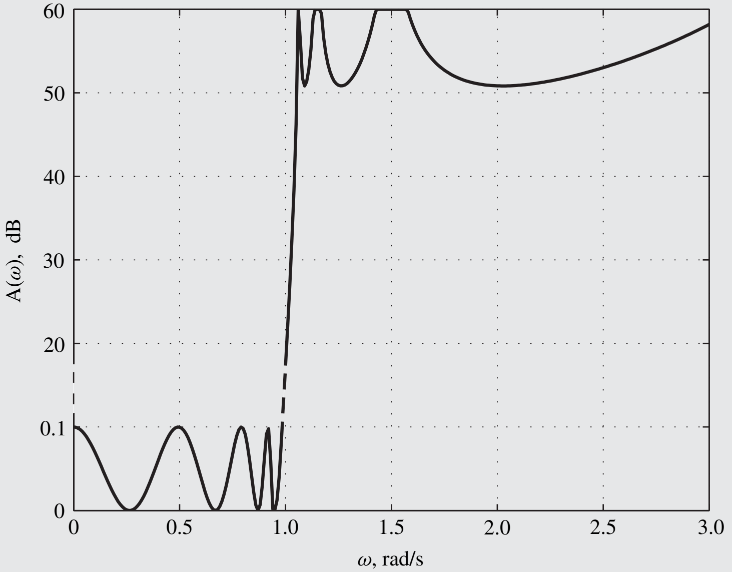


Figure 10.17 Loss characteristic of an eighth-order, elliptic filter (Example 10.6).

10.7 BESSEL-THOMSON APPROXIMATION

Ideally, the group delay of a filter should be independent of frequency, or, equivalently, the phase shift should be a linear function of frequency to minimize delay distortion (see Sec. 5.7). Since the only objective in the preceding three approximations is to achieve a specific loss characteristic, there is no reason for the phase characteristic to turn out to be linear. In fact, it turns out to be nonlinear as one might expect. Consequently, the delay tends to vary with frequency, in particular in the elliptic approximation.

Consider the transfer function

$$H(s) = \frac{b_0}{\sum_{i=0}^{n} b_i s^i} = \frac{b_0}{s^n B(1/s)} \tag{10.100}$$

where

$$b_i = \frac{(2n-i)!}{2^{n-i} i!(n-i)!} \tag{10.101}$$

Function $B(s)$ is a Bessel polynomial, and $s^n B(1/s)$ can be shown to have zeros in the left-half s plane. $B(1/j\omega)$ can be expressed in terms of *Bessel* functions [2, 10] as

$$B\left(\frac{1}{j\omega}\right) = \frac{1}{j^n}\sqrt{\frac{\pi\omega}{2}}[(-1)^n J_{-v}(\omega) - jJ_v(\omega)]e^{j\omega}$$

where $v = n + \frac{1}{2}$ and

$$J_v(\omega) = \omega^v \sum_{i=0}^{\infty} \frac{(-1)^i \omega^{2i}}{2^{2i+v} i! \Gamma(v+i+1)} \tag{10.102}$$

($\Gamma(\cdot)$ is the *gamma* function). Hence, from Eq. (10.100)

$$\begin{aligned}
|H(j\omega)|^2 &= \frac{2b_0^2}{\pi \omega^{2n+1} \left[J_{-v}^2(\omega) + J_v^2(\omega)\right]} \\
\theta(\omega) &= -\omega + \tan^{-1} \frac{(-1)^n J_v(\omega)}{J_{-v}(\omega)} \\
\tau(\omega) &= -\frac{d\theta(\omega)}{d\omega} = 1 - \frac{(-1)^n \left(J_{-v} J_v' - J_v J_{-v}'\right)}{J_{-v}^2(\omega) + J_v^2(\omega)}
\end{aligned}$$

Alternatively, from the properties of Bessel functions and Eq. (10.102) [2]

$$|H(j\omega)|^2 = 1 - \frac{\omega^2}{2n-1} + \frac{2(n-1)\omega^4}{(2n-1)^2(2n-3)} + \cdots \tag{10.103}$$

$$\tau(\omega) = 1 - \frac{\omega^{2n}}{b_0^2} |H(j\omega)|^2 \tag{10.104}$$

Clearly, as $\omega \to 0$, $|H(j\omega)| \to 1$ and $\tau(\omega) \to 1$. Furthermore, the first $n-1$ derivatives of $\tau(\omega)$ with respect to ω^2 are zero if $\omega = 0$, which makes the approximation maximally flat at the origin. This means that there is some frequency range $0 \leq \omega < \omega_p$ for which the delay is approximately constant. On the other hand, if $\omega \to \infty$, $|H(j\omega)| \to 1/(j\omega)^n \to 0$ and, therefore, $H(s)$ is a lowpass constant-delay approximation. This is sometimes referred to as the Bessel approximation since it uses a Bessel function. However, the possibility of using the function in Eq. (10.100) as a normalized lowpass approximation with a maximally flat group delay at he origin was proposed by Thomson [6] and its correct name should, therefore, be the Bessel-Thomson approximation.

Note that the formulas in Eqs. (10.103) and (10.104) are used here to demonstrate the maximally flat property of the group delay and have no other practical usefulness. For any other purpose, the amplitude and phase responses or the loss and delay characteristics should be obtained by using the transfer function in Eq. (10.100).

The Bessel-Thomson approximation has a normalized group delay of 1 s. However, any other delay can be achieved by replacing s by $\tau_0 s$ in Eq. (10.100). Typical loss and group-delay characteristics for the Bessel-Thomson approximation are plotted in Fig. 10.18 and 10.19, respectively.

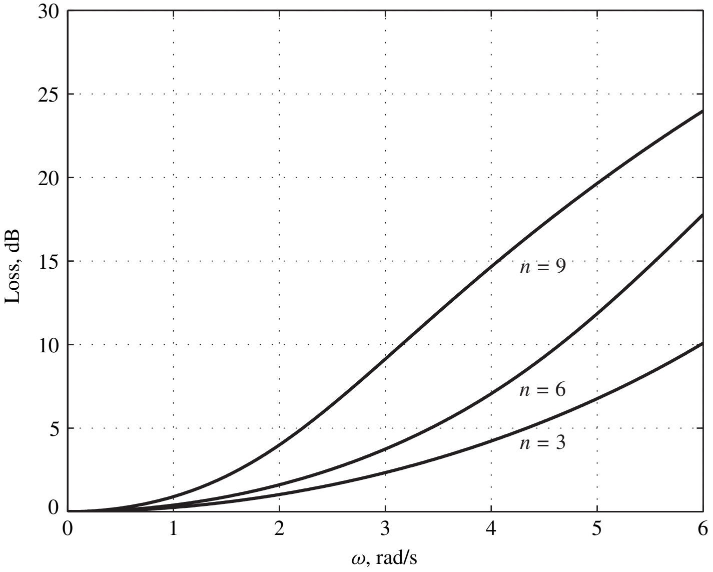


Figure 10.18 Loss characteristics of normalized Bessel-Thomson lowpass filters: $n = 3, 6, 9$.

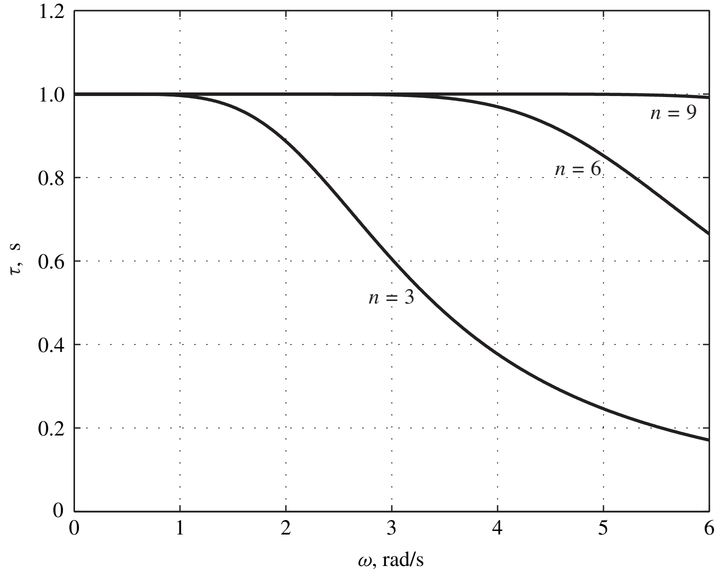


Figure 10.19 Delay characteristics of normalized Bessel-Thomson lowpass filters: $n = 3, 6, 9$.

Example 10.7 Form the Bessel-Thomson transfer function for $n = 6$.

■ **Solution**

From Eqs. (10.100) and (10.101), we obtain

$$H(s) = \frac{10,395}{10,395 + 10,395s + 4725s^2 + 1260s^3 + 210s^4 + 21s^5 + s^6} \quad ■$$

(See Fig. 10.18 and 10.19 for the loss and delay characteristics).

10.8 TRANSFORMATIONS

In the preceding sections, only normalized lowpass approximations have been considered. The reason is that denormalized lowpass, highpass, bandpass, and bandstop approximations can be easily derived by using *transformations* of the form

$$s = f(\bar{s})$$

10.8.1 Lowpass-to-Lowpass Transformation

Consider a normalized lowpass transfer function $H_N(s)$ with passband and stopband edges ω_p and ω_a, and let

$$s = \lambda\bar{s} \tag{10.105}$$

in $H_N(s)$. If $s = j\omega$, we have $\bar{s} = j\omega/\lambda$ and hence Eq. (10.105) maps the j axis of the s plane onto the j axis of the $\bar{s}$ plane. In particular, ranges 0 to $j\omega_p$ and $j\omega_a$ to $j\infty$ map onto ranges 0 to $j\omega_p/\lambda$ and $j\omega_a/\lambda$ to $j\infty$, respectively, as depicted in Fig. 10.20. Therefore,

$$H_{\text{LP}}(\bar{s}) = H_N(s)\Big|_{s=\lambda\bar{s}}$$

constitutes a denormalized lowpass approximation with passband and stopband edges ω_p/λ and ω_a/λ, respectively. A graphical illustration of the lowpass-to-lowpass transformation is shown in Fig. 10.21.

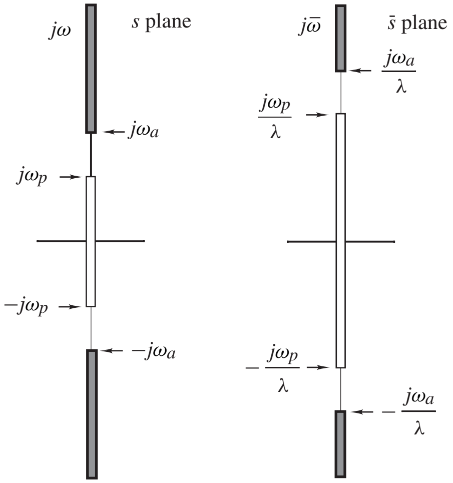
10.8.2 Lowpass-to-Bandpass Transformation

Now let

$$s = \frac{1}{B}\left(\bar{s} + \frac{\omega_0^2}{\bar{s}}\right)$$

in $H_N(s)$, where B and ω_0 are constants. If $s = j\omega$ and $\bar{s} = j\bar{\omega}$, we have

$$j\omega = \frac{j}{B}\left(\bar{\omega} - \frac{\omega_0^2}{\bar{\omega}}\right) \qquad \text{or} \qquad j\bar{\omega} = j\left[\frac{\omega B}{2} \pm \sqrt{\omega_0^2 + \left(\frac{\omega B}{2}\right)^2}\,\right]$$



$j\omega$
s plane
$j\omega_a$
$j\omega_p$
$-j\omega_p$
$-j\omega_a$
$j\bar{\omega}$
$\bar{s}$ plane
$\frac{j\omega_a}{\lambda}$
$\frac{j\omega_p}{\lambda}$
$-\frac{j\omega_p}{\lambda}$
$-\frac{j\omega_a}{\lambda}$

Figure 10.20 Lowpass-to-lowpass transformation: Mapping.

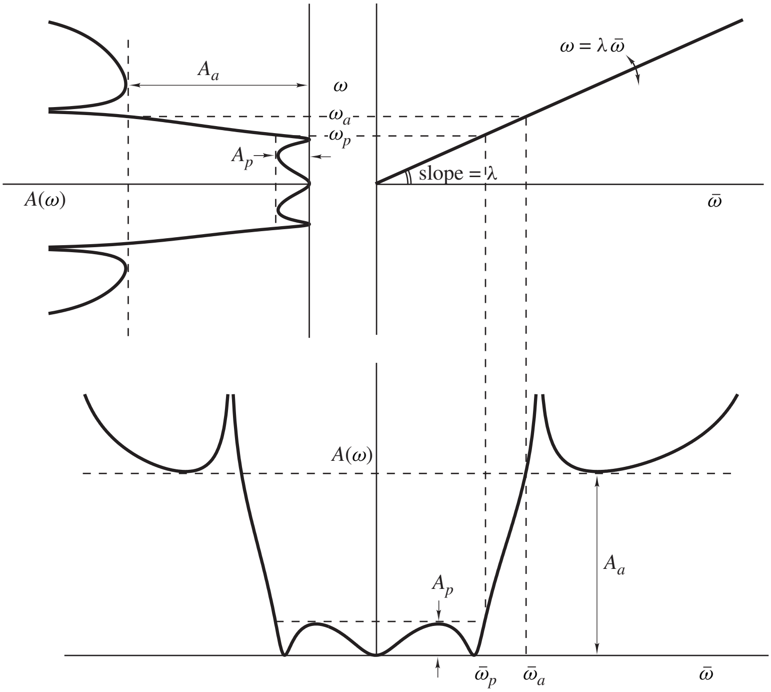


Figure 10.21 Lowpass-to-lowpass transformation: Graphical interpretation.

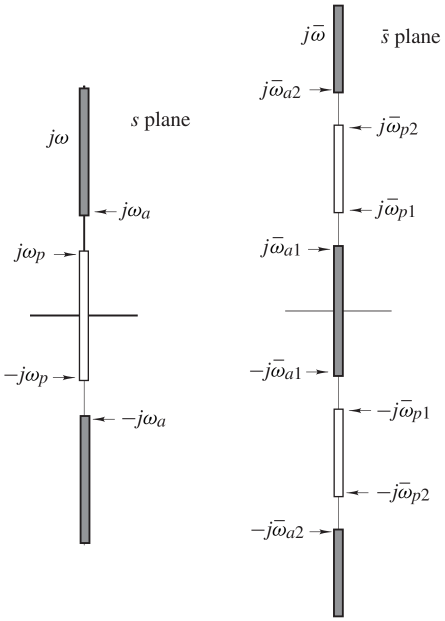


Figure 10.22 Lowpass-to-bandpass transformation: Mapping.

Table 10.2 Analog-filter transformations

Type	Transformation
LP to LP	$s = \lambda\bar{s}$
LP to HP	$s = \dfrac{\lambda}{\bar{s}}$
LP to BP	$s = \dfrac{1}{B}\left(\bar{s} + \dfrac{\omega_0^2}{\bar{s}}\right)$
LP to BS	$s = \dfrac{B\bar{s}}{\bar{s}^2 + \omega_0^2}$

Hence

$$\bar{\omega} = \begin{cases} \omega_0 & \text{if } \omega = 0 \\ \pm\bar{\omega}_{p1}, \pm\bar{\omega}_{p2} & \text{if } \omega = \pm\omega_p \\ \pm\bar{\omega}_{a1}, \pm\bar{\omega}_{a2} & \text{if } \omega = \pm\omega_a \end{cases}$$

where

$$\bar{\omega}_{p1}, \bar{\omega}_{p2} = \mp\frac{\omega_p B}{2} + \sqrt{\omega_0^2 + \left(\frac{\omega_p B}{2}\right)^2}$$

$$\bar{\omega}_{a1}, \bar{\omega}_{a2} = \mp\frac{\omega_a B}{2} + \sqrt{\omega_0^2 + \left(\frac{\omega_a B}{2}\right)^2}$$

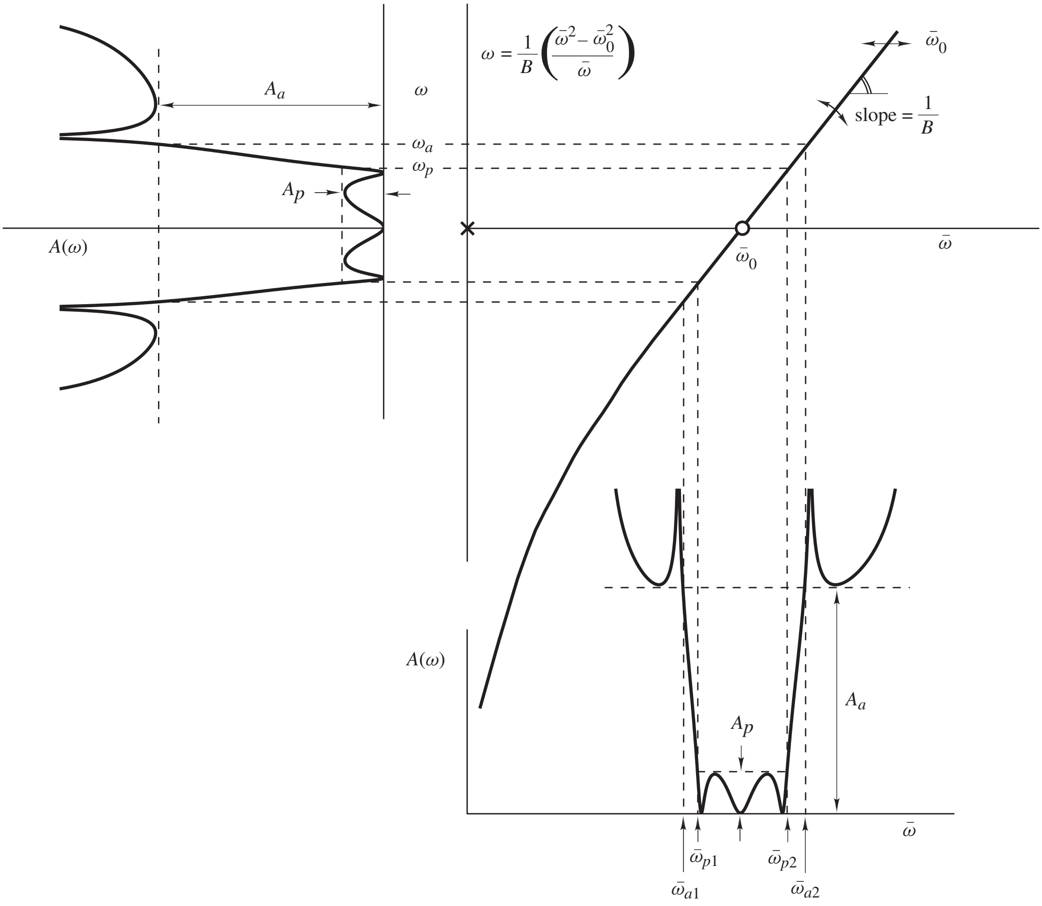


Figure 10.23 Lowpass-to-bandpass transformation: Graphical interpretation.

The mapping for $s = j\omega$ is thus of the form illustrated in Fig. 10.22, and consequently

$$H_{BP}(\bar{s}) = H_N(s)\Big|_{s=\frac{1}{B}\left(\bar{s}+\frac{\omega_0^2}{\bar{s}}\right)}$$

is a bandpass approximation with passband edges ω_{p1}, ω_{p2} and stopband edges ω_{a1}, ω_{a2}. A graphical illustration of the lowpass-to-bandpass transformation is shown in Fig. 10.23.

Similarly, the transformations in the second and fourth rows of Table 10.2 yield highpass and bandstop approximations.

REFERENCES

[1] E. A. Guillemin, *Synthesis of Passive Networks*, New York: Wiley, 1957.

[2] N. Balabanian, *Network Synthesis*, Englewood Cliffs, NJ: Prentice-Hall, 1958.

[3] L. Weinberg, *Network Analysis and Synthesis*, New York: McGraw-Hill, 1962.

[4] J. K. Skwirzynski, *Design Theory and Data for Electrical Filters*, London: Van Nostrand, 1965.

[5] R. W. Daniels, *Approximation Methods for Electronic Filter Design*, New York: McGraw-Hill, 1974.

[6] W. E. Thomson, "Delay networks having maximally flat frequency characteristics," *Proc. Inst. Elect. Eng.*, pt. 3, vol. 96, pp. 487–490, 1949.

[7] A. Antoniou, *General Characteristics of Filters* in *The Circuits and Systems Handbook*, ed. W.-K. Chen, Portland, OR: Book News, Inc., 2004.

[8] R. J. Schwarz and B. Friedland, *Linear Systems*, New York: McGraw-Hill, 1965.

[9] A. J. Grossman, "Synthesis of Tchebyscheff parameter symmetrical filters," *Proc. IRE*, vol. 45, pp. 454–473, Apr. 1957.

[10] G. N. Watson, *A Treatise on the Theory of Bessel Functions*, London: Cambridge University Press, 1948.

PROBLEMS

10.1. A stable analog system is characterized by the transfer function in Eq. (10.5). Show that the steady-state sinusoidal response of the system is given by Eq. (10.11).

10.2. A fourth-order lowpass Butterworth filter[5] is required.

(a) Obtain the normalized transfer function $H_N(s)$.

(b) Derive expressions for the loss and phase shift.

(c) Calculate the loss and phase shift at $\omega = 0.5$ rad/s.

(d) Obtain a corresponding denormalized transfer function $H_D(s)$ with a 3-dB cutoff frequency at 1000 rad/s.

10.3. A fifth-order Butterworth filter is required.

(a) Form $H(s)$.

(b) Plot the loss characteristic.

10.4. Filter specifications are often described pictorially as in Fig. P10.4, where ω_p and ω_a are desired passband and stopband edges, respectively, A_p is the maximum passband loss, and A_a is the minimum stopband loss. Find n and, in turn, form $H(s)$, if $\omega_p = 1$, $\omega_a = 3$ rad/s, $A_p = 3.0$, $A_a \geq 45$ dB. Use the Butterworth approximation.

10.5. In an application a normalized Butterworth lowpass filter is required that would satisfy the following specification:

- Passband edge ω_p: 0.6 rad/s
- Stopband edge ω_a: 2.5 rad/s
- Maximum passband loss A_p: 1.0 dB
- Minimum stopband loss A_a: 40.0 dB

(a) Find the minimum filter order that would satisfy the specifications.

(b) Calculate the actual maximum passband loss and minimum stopband loss.

(c) Obtain the required transfer function.

10.6. A third-order lowpass filter with passband edge $\omega_p = 1$ rad/s and passband ripple $A_p = 1.0$ dB is required. Obtain the poles and multiplier constant of the transfer function assuming a Chebyshev approximation.

[5] *The filters considered in this problem section are all analog filters.*

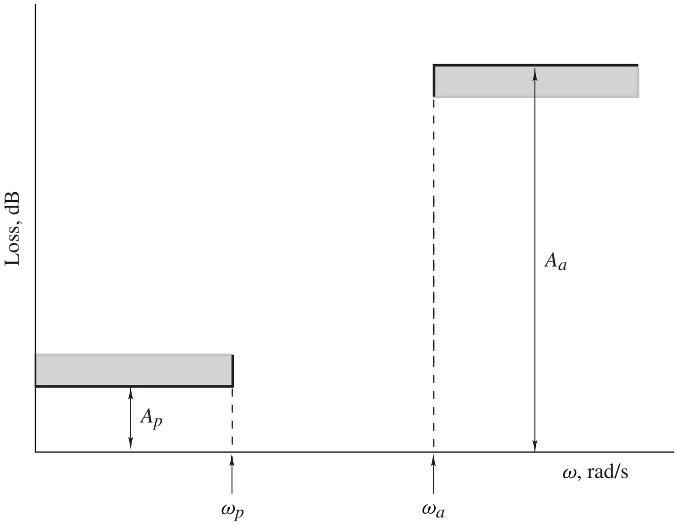


Figure P10.4

10.7. A fifth-order normalized lowpass Chebyshev filter is required.

(*a*) Form $H(s)$ if $A_p = 0.1$ dB.

(*b*) Plot the loss characteristic.

10.8. A Chebyshev filter that would satisfy the specifications of Fig. P10.8 is required. Find n and, in turn, form $H(s)$.

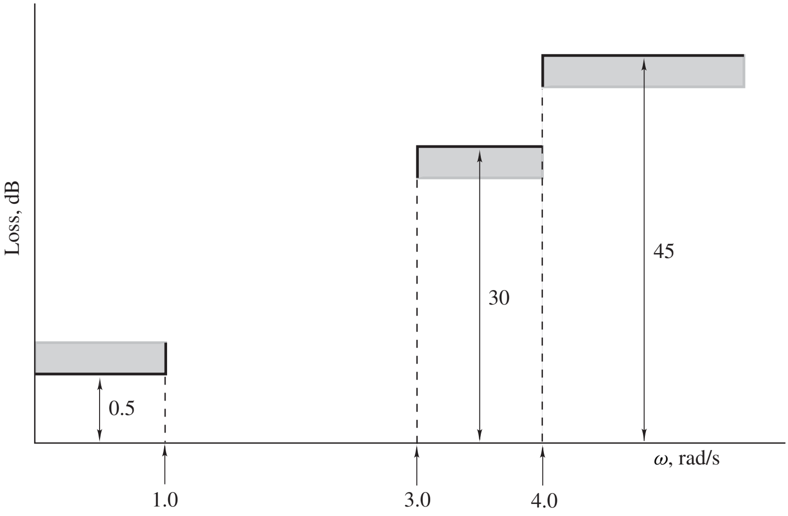


Figure P10.8

10.9. An application calls for a normalized Chebyshev lowpass filter that would satisfy the following specification:

- Passband edge ω_p: 1.0 rad/s
- Stopband edge ω_a: 2.2 rad/s
- Maximum passband loss A_p: 0.2 dB
- Minimum stopband loss A_a: 40.0 dB

(*a*) Find the minimum filter order that would satisfy the specifications.
(*b*) Calculate the actual maximum passband loss and minimum stopband loss.
(*c*) Obtain the required transfer function.

10.10. (*a*) Show that

$$T_{n+1}(\omega) = 2\omega T_n(\omega) - T_{n-1}(\omega)$$

(*b*) Hence demonstrate that the following relation [5] holds:

$$T_n(\omega) = \frac{n}{2}\sum_{r=0}^{K} \frac{(-1)^r(n-r-1)!}{r!(n-2r)!}(2\omega)^{n-2r} \qquad \text{where } K = \text{Int}\left(\frac{n}{2}\right)$$

(*c*) Obtain $T_{10}(\omega)$.

10.11. (*a*) Find $A(\omega)$ for the normalized lowpass Butterworth and Chebyshev approximations if $\omega \gg 1$.
(*b*) Show that $A(\omega)$ increases at the rate of $20n$ dB/decade in both cases.

10.12. The inverse-Chebyshev approximation can be derived by considering the loss function

$$A(\omega) = 10\log\left[1 + \frac{1}{\delta^2 T_n^2(\omega)}\right]$$

where

$$\delta^2 = \frac{1}{10^{0.1A_a} - 1}$$

(*a*) Show that $A(\omega)$ represents a highpass filter with an equiripple stopband loss, a monotonic increasing passband loss, and a stopband edge $\omega_a = 1$ rad/s.
(*b*) Show that the filter represented by $A(\omega)$ has a transfer function of the form

$$H_{HP}(s) = \frac{\prod_{i=1}^{n}(s - z_i)}{\prod_{i=1}^{n}(s - p_i)}$$

where z_i and p_i for $i = 1, 2, \ldots, n$ are given by Eqs. (10.50), (10.51), and (10.53), respectively.
(*c*) Show that $H_N(s) = H_{HP}(1/s)$ is the normalized lowpass transfer function for the inverse-Chebyshev approximation.

10.13. A fourth-order inverse-Chebyshev filter with a minimum stopband loss of 40 dB is required.
(*a*) Obtain the required transfer function.
(*b*) Find the 3-dB cutoff frequency.

10.14. An application requires a normalized inverse-Chebyshev lowpass filter that would satisfy the following specifications:

- Passband edge ω_p: 0.5 rad/s
- Stopband edge ω_a: 1.0 rad/s
- Maximum passband loss A_p: 0.5 dB
- Minimum stopband loss A_a: 30.0 dB

(*a*) Find the minimum filter order that would satisfy the specifications.

(*b*) Obtain the required transfer function.

(*c*) Calculate the actual maximum passband loss and minimum stopband loss.

10.15. (*a*) Write a MATLAB *m*-file that can be used to obtain the normalized elliptic transfer function for an arbitrary set of given specifications $\{k,\ A_p,\ A_a\}$ where k is the selectivity, A_p is maximum passband loss, and A_a is the minimum stopband loss. Your program should also compute the actual stopband loss.

(*b*) Use the program in part (*a*) to obtain elliptic transfer functions for two different sets of specifications that would result in an even- and an odd-order transfer function of order greater than 3.

(*c*) Plot the loss characteristics associated with the transfer functions obtained.

10.16. (*a*) A lowpass elliptic filter is required that would satisfy the specifications

$$n = 4 \qquad A_p = 1.0 \text{ dB} \qquad k = 0.7$$

Form $H(s)$.

(*b*) Determine the corresponding minimum stopband loss.

(*c*) Plot the loss characteristic.

10.17. In a particular application an elliptic lowpass filter is required. The specifications are

- Selectivity k: 0.6
- Maximum passband loss A_p: 0.5 dB
- Minimum stopband loss A_a: 40.0 dB

10.18. An elliptic lowpass filter that would satisfy the specifications

- Selectivity k: 0.95
- Maximum passband loss A_p: 0.3 dB
- Minimum stopband loss A_a: 60.0 dB

is required.

(*a*) Determine the order of the transfer function.

(*b*) Determine the actual loss.

(*c*) Obtain the transfer function.

10.19. (*a*) Obtain the normalized transfer function $H(s)$ for the eighth-order Bessel-Thomson approximation.

(*b*) Plot the corresponding phase characteristic.

10.20. (*a*) Obtain the normalized transfer function $H(s)$ for the ninth-order Bessel-Thomson approximation.

(*b*) Using the transfer function in part (*a*), obtain expressions (i) for the loss characteristic, (ii) for the phase response, and (iii) for the group delay characteristic.

(*c*) Using MATLAB or similar software, plot (i) the loss characteristic, (ii) the phase response, and (iii) the delay characteristic for the frequency range 0 to 6 rad/s.

10.21. Show that

$$H(s) = \frac{\sum_{i=0}^{n} b_i(-s)^i}{\sum_{i=0}^{n} b_i s^i}$$

where

$$b_i = \frac{(2n-i)!}{2^{n-i} i!(n-i)!}$$

is a constant-delay, allpass transfer function.

10.22. A constant-delay lowpass filter is required with a group delay of 1 ms. Form $H(s)$ using the sixth-order Bessel-Thomson approximation.

10.23. An normalized inverse-Chebyshev lowpass filter has a transfer function

$$H_N(s) = \frac{H_0}{s + b_{00}} \prod_{i=1}^{2} \frac{s^2 + a_{0i}}{s^2 + b_{1i}s + b_{0i}}$$

where

$$H_0 = 1.581147E-2 \qquad b_{00} = 5.957330E-1$$
$$a_{01} = 2.894427 \qquad b_{01} = 3.161351E-1 \qquad b_{11} = 8.586353E-1$$
$$a_{02} = 1.105573 \qquad b_{02} = 2.686568E-1 \qquad b_{12} = 2.787138E-1$$

(*a*) By using the lowpass-to-lowpass transformation, obtain a lowpass transfer function that would result in a stopband edge of 1000 Hz.

(*b*) By using MATLAB or similar software, find the passband edge of the transformed filter assuming a maximum passband loss of 1.0 dB.

10.24. A normalized lowpass Chebyshev filter has a transfer function

$$H_N(s) = \frac{H_0}{s + b_{00}} \prod_{i=1}^{2} \frac{1}{s^2 + b_{1i}s + b_{0i}}$$

where

$$H_0 = 0.287898 \qquad b_{00} = 0.461411$$
$$b_{01} = 1.117408 \qquad b_{11} = 0.285167$$
$$b_{02} = 0.558391 \qquad b_{12} = 0.746578$$

(*a*) By using the lowpass-to-highpass transformation, obtain a highpass transfer function that would result in a passband edge of 10,000 Hz.

(*b*) By using MATLAB or similar software, find (i) the maximum passband loss and (ii) the minimum stopband loss of the highpass filter assuming a stopband edge of 5800 Hz.

10.25. A normalized elliptic transfer function for which $k = 0.8$ and $A_p = 0.1$ dB is subjected to the lowpass-to-bandpass transformation. Find the passband and stopband edges of the bandpass filter if $B = 200$, $\omega_0 = 1000$ rad/s.

10.26. A normalized elliptic transfer function for which $k = 0.7$ and $A_p = 0.5$ dB is subjected to the lowpass-to-bandstop transformation. Find the passband and stopband edges of the bandstop filter if $B = 100$, $\omega_0 = 2000$ rad/s.

10.27. A normalized, third-order, elliptic, lowpass filter is characterized by the transfer function

$$H_N(s) = H_0 \frac{s^2 + a_{01}}{(s + b_{00})(s^2 + b_{11}s + b_{01})}$$

where

$$\begin{aligned} H_0 &= 6.710103E - 2 \\ b_{00} &= 3.715896E - 1 \qquad a_{01} = 2.687292 \\ b_{11} &= 3.044886E - 1 \qquad b_{01} = 4.852666E - 1 \end{aligned}$$

(*a*) Obtain a bandpass elliptic transfer function by applying the lowpass-to-bandpass transformation assuming that $B_0 = 1.153776E + 3$ and $\omega_0 = 1.445683E + 3$.

(*b*) By plotting the loss characteristic of the bandpass filter over the frequency range 0 to 4000 rad/s, find the maximum passband loss, the minimum stopband loss, the passband edges, and stopband edges of the filter.

10.28. A normalized, third-order, elliptic, lowpass filter is characterized by the transfer function

$$H_N(s) = H_0 \frac{s^2 + a_{01}}{(s + b_{00})(s^2 + b_{11}s + b_{01})}$$

where

$$\begin{aligned} H_0 &= 4.994427E - 2 \\ b_{00} &= 3.461194E - 1 \qquad a_{01} = 3.011577 \\ b_{11} &= 2.961751E - 1 \qquad b_{01} = 4.345639E - 1 \end{aligned}$$

(*a*) Obtain a bandstop elliptic transfer function by applying the lowpass-to-bandpass transformation assuming that $B_0 = 8.0E + 2$ and $\omega_0 = 7.885545E + 02$.

(*b*) By plotting the loss characteristic of the filter over the frequency range 0 to 2000 rad/s, find the maximum passband loss, the minimum stopband loss, the passband edges, and stopband edges of the bandstop filter.

10.29. A lowpass filter is required that would satisfy the following specifications:

- Passband edge ω_p: 2000 rad/s
- Stopband edge ω_a: 7000 rad/s
- Maximum passband loss A_p: 0.4 dB
- Minimum stopband loss A_a: 45.0 dB

(*a*) Assuming a Butterworth approximation, find the required order n and the value of the transformation parameter λ.

(*b*) Form $H(s)$.

10.30. Repeat Prob. 10.29 for the case of a Chebyshev approximation and compare the design obtained with that obtained in Prob. 10.29.

10.31. Repeat Prob. 10.29 for the case of an inverse-Chebyshev approximation and compare the design obtained with that obtained in Prob. 10.29.

10.32. Repeat Prob. 10.29 for the case of an elliptic approximation and compare the design obtained with that obtained in Prob. 10.29.

10.33. A highpass filter is required that would satisfy the following specifications:

- Passband edge ω_p: 2000 rad/s
- Stopband edge ω_a: 1000 rad/s

- Maximum passband loss A_p: 0.5 dB
- Minimum stopband loss A_a: 40.0 dB

(a) Assuming a Butterworth approximation, find the required order n and the value of the transformation parameter λ.

(b) Form $H(s)$.

10.34. Repeat Prob. 10.33 for the case of a Chebyshev approximation and compare the design obtained with that obtained in Prob. 10.33.

10.35. Repeat Prob. 10.33 for the case of an inverse-Chebyshev approximation and compare the design obtained with that obtained in Prob. 10.33.

10.36. Repeat Prob. 10.33 for the case of an elliptic approximation and compare the design obtained with that obtained in Prob. 10.33.

10.37. A bandpass filter is required that would satisfy the specifications depicted in Fig. P10.37. Assuming that the elliptic approximation is to be employed, find suitable values for ω_0, k, B, and n.

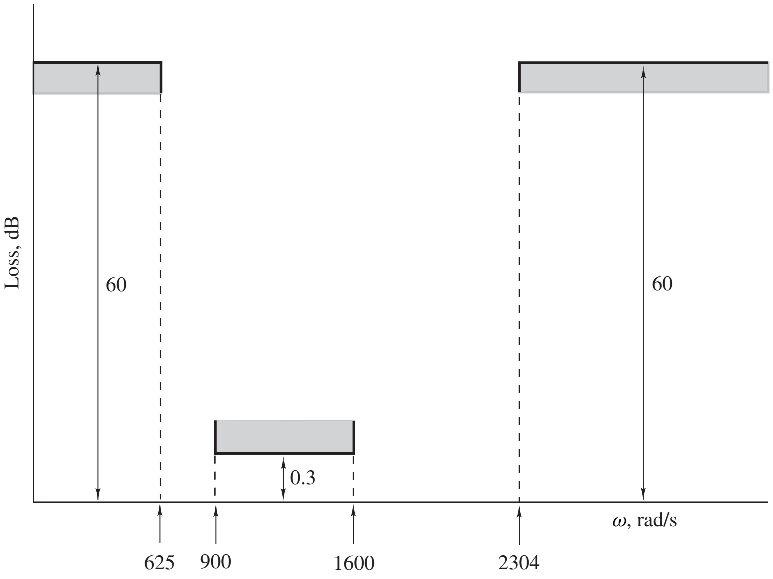


Figure P10.37

10.38. A bandpass filter is required that would satisfy the following specifications:

- Lower passband edge ω_{p1}: 9500 rad/s
- Upper passband edge ω_{p2}: 10,500 rad/s
- Lower stopband edge ω_{a1}: 5000 rad/s
- Lower stopband edge ω_{a2}: 15,000 rad/s
- Maximum passband loss A_p: 1.0 dB
- Minimum stopband loss A_a: 50.0 dB

(a) Assuming a Butterworth approximation, find the required order n and the value of the transformation parameters B and ω_0.

(b) Form $H(s)$.

10.39. Repeat Prob. 10.38 for the case of a Chebyshev approximation and compare the design obtained with that obtained in Prob. 10.38.

10.40. Repeat Prob. 10.38 for the case of an inverse-Chebyshev approximation and compare the design obtained with that obtained in Prob. 10.38.

10.41. Repeat Prob. 10.38 for the case of an elliptic approximation and compare the design obtained with that obtained in Prob. 10.38.

10.42. A bandstop filter is required that would satisfy the specifications depicted in Fig. P10.42. Assuming that the elliptic approximation is to be employed, find suitable values for $\omega_0,\ k,\ B$, and n.

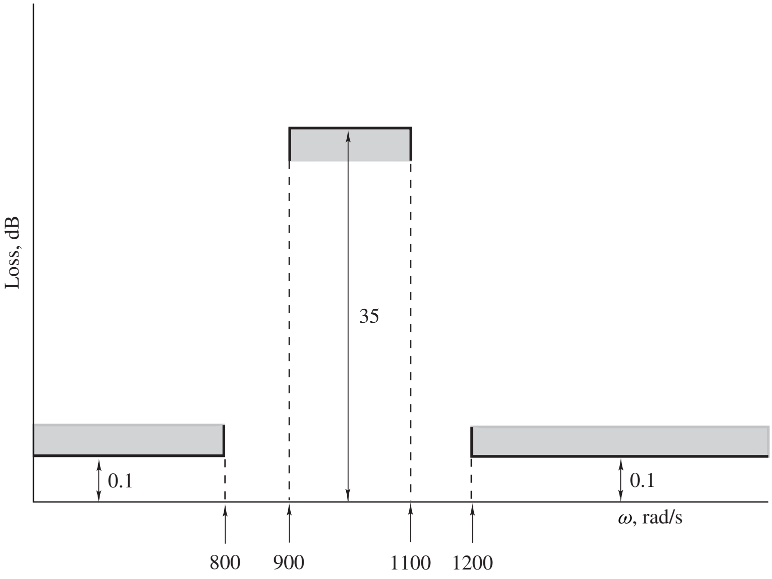


Figure P10.42

10.43. A bandstop filter is required that would satisfy the following specifications:

- Lower passband edge ω_{p1}: 20 rad/s
- Upper passband edge ω_{p2}: 80 rad/s
- Lower stopband edge ω_{a1}: 48 rad/s
- Lower stopband edge ω_{a2}: 52 rad/s
- Maximum passband loss A_p: 1.0 dB
- Minimum stopband loss A_a: 25.0 dB

(*a*) Assuming a Butterworth approximation, find the required order n and the value of the transformation parameters B and ω_0.

(*b*) Form $H(s)$.

10.44. Repeat Prob. 10.43 for the case of a Chebyshev approximation and compare the design obtained with that obtained in Prob. 10.43.

10.45. Repeat Prob. 10.43 for the case of an inverse-Chebyshev approximation and compare the design obtained with that obtained in Prob. 10.43.

10.46. Repeat Prob. 10.43 for the case of an elliptic approximation and compare the design obtained with that obtained in Prob. 10.43.

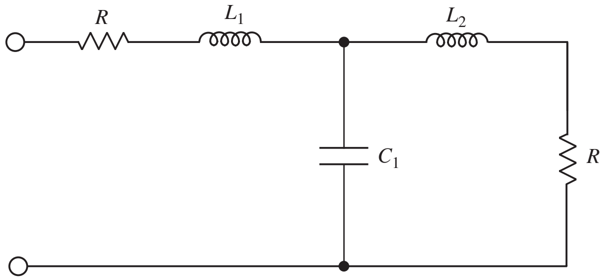


Figure P10.47

10.47. Figure P10.47 shows an LC filter.

(a) Derive a highpass LC filter.

(b) Derive a bandpass LC filter.

(c) Derive a bandstop LC filter.

CHAPTER 11

DESIGN OF RECURSIVE (IIR) FILTERS

11.1 INTRODUCTION

Approximation methods for the design of recursive (IIR) filters differ quite significantly from those used for the design of nonrecursive filters. The basic reason is that in the first case the transfer function is a ratio of polynomials of z whereas in the second case it is a polynomial of z^{-1}.

In *recursive filters*, the approximation problem is usually solved through indirect methods. First, a continuous-time transfer function that satisfies certain specifications is obtained using one of the standard analog-filter approximations described in Chap. 10. Then a corresponding discrete-time transfer function is obtained using one of the following methods [1–9]:

1. Invariant impulse-response method
2. Modified version of method 1
3. Matched-z transformation
4. Bilinear transformation

This chapter is concerned with the indirect approach to the design of recursive filters. It starts with the realizability constraints that must be satisfied by the discrete-time transfer function and then deals with the details of the aforementioned approximation methods. The chapter also describes a set of z-domain transformations that can be used to derive transformed lowpass, highpass, bandpass, or bandstop discrete-time transfer functions from a given lowpass discrete-time transfer function. It concludes with a general discussion on the choice between recursive and nonrecursive designs.

Iterative methods that are suitable for the design of nonrecursive and recursive filters are considered in Chaps. 15 and 16, respectively.

11.2 REALIZABILITY CONSTRAINTS

In order to be *realizable* by a recursive filter, a transfer function must satisfy the following constraints:

1. It must be a rational function of z with real coefficients.
2. Its poles must lie within the unit circle of the z plane.
3. The degree of the numerator polynomial must be equal to or less than that of the denominator polynomial.

The first constraint is actually artificial and is imposed by our assumption in Chaps. 1 and 4 that signals are real and that the constituent elements of a digital filter perform real arithmetic. If unit delays, adders, and multipliers are defined for complex signals in terms of complex arithmetic, then transfer functions with complex coefficients can be considered to be realizable [10, 11]. The second and third constraints will assure a stable and causal filter, respectively (see Secs. 5.3 and 5.2, respectively).

11.3 INVARIANT IMPULSE-RESPONSE METHOD

Consider the impulse modulated filter $\hat{F}_A$ of Fig. 11.1, where S is an ideal impulse modulator and F_A is an analog filter characterized by $H_A(s)$. $\hat{F}_A$ can be represented by a continuous-time transfer function $\hat{H}_A(s)$ or, equivalently, by a discrete-time transfer function $H_D(z)$, as shown in Sec. 6.9. From Eq. (6.53b)

$$\hat{H}_A(j\omega) = H_D(e^{j\omega T}) = \frac{h_A(0+)}{2} + \frac{1}{T}\sum_{k=-\infty}^{\infty} H_A(j\omega + jk\omega_s) \tag{11.1}$$

where $\omega_s = 2\pi/T$ is the sampling frequency and

$$\begin{aligned} h_A(t) &= \mathcal{L}^{-1}H_A(s) \\ h_A(0+) &= \lim_{s\to\infty}[sH_A(s)] \\ H_D(z) &= \mathcal{Z}h_A(nT) \end{aligned} \tag{11.2}$$

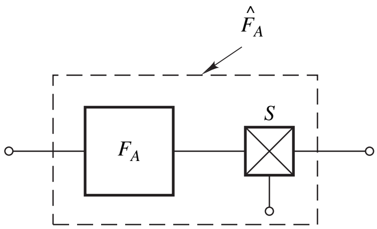


Figure 11.1 Impulse modulated filter.

Therefore, given an analog filter F_A, a corresponding digital filter, represented by $H_D(z)$, can be derived by using the following procedure:

1. Deduce $h_A(t)$, the impulse response of the analog filter.
2. Replace t by nT in $h_A(t)$.
3. Form the z transform of $h_A(nT)$.

If

$$H_A(j\omega) \approx 0 \qquad \text{for } |\omega| \geq \frac{\omega_s}{2} \tag{11.3a}$$

then

$$\sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} H_A(j\omega + jk\omega_s) \approx 0 \qquad \text{for } |\omega| < \frac{\omega_s}{2} \tag{11.3b}$$

If, in addition,

$$h_A(0+) = 0 \tag{11.4}$$

Eqs. (11.1), (11.3b), and (11.4) yield

$$\hat{H}_A(j\omega) = H_D(e^{j\omega T}) \approx \frac{1}{T} H_A(j\omega) \qquad \text{for } |\omega| < \frac{\omega_s}{2} \tag{11.5}$$

i.e., if $H_A(j\omega)$ is bandlimited, the baseband frequency response of the derived digital filter is approximately the same as that of the analog filter except that the gain of the digital filter is multiplied by the constant $1/T$. This constant can be eliminated by multiplying the numerator coefficients of $H_D(z)$ by T.

If the denominator degree in $H_A(s)$ exceeds the numerator degree by at least 2, the basic assumptions in Eqs. (11.3a) and (11.4) hold for some sufficiently high value of ω_s. If, in addition, the poles of $H_A(s)$ are simple, we can write

$$H_A(s) = \sum_{i=1}^{N} \frac{R_i}{s - p_i} \tag{11.6}$$

Hence from steps 1 and 2 above

$$h_A(t) = \mathcal{L}^{-1} H_A(s) = \sum_{i=1}^{N} R_i e^{p_i t} \qquad \text{and} \qquad h_A(nT) = \sum_{i=1}^{N} R_i e^{p_i nT}$$

(see Sec. 10.2.4). Subsequently, from step 3

$$H_D(z) = \mathcal{Z} h_A(nT) = \sum_{i=1}^{N} \frac{R_i z}{z - e^{Tp_i}} \tag{11.7}$$

Since complex-conjugate pairs of poles in $H_A(s)$ yield complex-conjugate values of R_i and e^{Tp_i}, the coefficients in $H_D(z)$ are real. Pole $p_i = \sigma_i + j\omega_i$ gives rise to a pole $\tilde{p}_i$ in $H_D(z)$, where

$$\tilde{p}_i = e^{Tp_i} = e^{T(\sigma_i + j\omega_i)}$$

and for $\sigma_i < 0$, $|\tilde{p}_i| < 1$. Hence a stable analog filter yields a stable digital filter. Also the numerator degree in $H_D(z)$ cannot exceed the denominator degree as can be easily verified, and $H_D(z)$ is therefore realizable.

The method described, which is known as the *invariant impulse-response* method, yields good results for Butterworth, Bessel-Thomson, or Chebyshev lowpass and bandpass filters for which the basic assumptions of Eqs. (11.3a) and (11.4) hold. An advantage of the method is that it preserves the phase response as well as the loss characteristic of the analog filter.

Example 11.1 Design a digital filter by applying the invariant impulse-response method to the Bessel-Thomson transfer function

$$H_A(s) = \frac{105}{105 + 105s + 45s^2 + 10s^3 + s^4}$$

(see Sec. 10.7). Employ a sampling frequency $\omega_s = 8$ rad/s; repeat with $\omega_s = 16$ rad/s.

■ **Solution**

The poles of $H_A(s)$ and the residues in Eq. (11.6) are

$$\begin{aligned} p_1, p_1^* &= -2.896211 \pm j0.8672341 \\ p_2, p_2^* &= -2.103789 \pm j2.657418 \\ R_1, R_1^* &= 1.663392 \mp j8.396299 \\ R_2, R_2^* &= -1.663392 \pm j2.244076 \end{aligned}$$

Hence from Eq. (11.7)

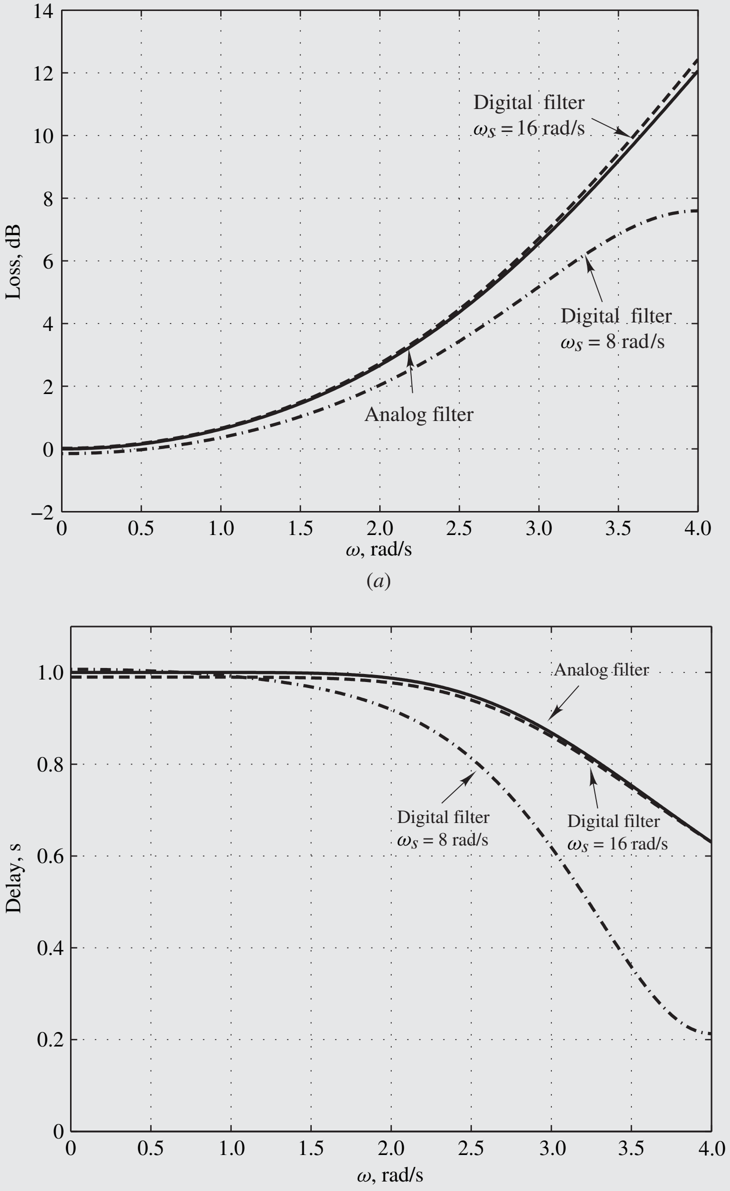
$$TH_D(z) = \sum_{j=1}^{2} \frac{a_{1j}z + a_{2j}z^2}{b_{0j} + b_{1j}z + z^2} \qquad \blacksquare$$

where coefficients a_{ij} and b_{ij} are given in Table 11.1. The transfer function is multiplied by T to eliminate the effect of constant $1/T$ in Eq. (11.5).

Table 11.1 Coefficients of $TH_D(z)$ (Example 11.1)

ω_s	j	a_{1j}	a_{2j}	b_{0j}	b_{1j}
8	1	$6.452333E-1$	2.612851	$1.057399E-2$	$-1.597700E-1$
	2	$-8.345233E-1$	-2.612851	$3.671301E-2$	$1.891907E-1$
16	1	$3.114550E-1$	1.306425	$1.028299E-1$	$-6.045080E-1$
	2	$-3.790011E-1$	-1.306425	$1.916064E-1$	$-4.404794E-1$

The loss and delay characteristics obtained are plotted in Fig. 11.2*a* and *b*, respectively. The higher sampling frequency gives better results because aliasing errors are less pronounced.



(*a*)

(*b*)

Figure 11.2 Example 11.1: (*a*) Loss characteristics, (*b*) delay characteristics.

11.4 MODIFIED INVARIANT IMPULSE-RESPONSE METHOD

Aliasing errors tend to restrict the application of the invariant impulse-response method to the design of *allpole filters*.[1] However, a *modified* version of the method is available, as will now be demonstrated, which can be applied to filters that also have zeros in the finite s plane.

Consider the transfer function

$$H_A(s) = \frac{H_0 N(s)}{D(s)} = \frac{H_0 \prod_{i=1}^{M}(s - z_i)}{\prod_{i=1}^{N}(s - p_i)} \tag{11.8}$$

where M can be as high as N. We can write

$$H_A(s) = \frac{H_0 H_{A1}(s)}{H_{A2}(s)}$$

where

$$H_{A1}(s) = \frac{1}{D(s)} \tag{11.9}$$

$$H_{A2}(s) = \frac{1}{N(s)} \tag{11.10}$$

Clearly, with $M, N \geq 2$ Eq. (11.2) yields

$$h_{A1}(0+) = 0 \qquad h_{A2}(0+) = 0$$

and furthermore

$$\left.\begin{aligned} H_{A1}(j\omega) &\approx 0 \\ H_{A2}(j\omega) &\approx 0 \end{aligned}\right\} \qquad \text{for } |\omega| \geq \frac{\omega_s}{2}$$

for some sufficiently high value of ω_s. Consequently, from Eq. (11.1) we can write

$$\left.\begin{aligned} \hat{H}_{A1}(j\omega) &= H_{D1}(e^{j\omega T}) \approx \frac{1}{T} H_{A1}(j\omega) \\ \hat{H}_{A2}(j\omega) &= H_{D2}(e^{j\omega T}) \approx \frac{1}{T} H_{A2}(j\omega) \end{aligned}\right\} \qquad \text{for } |\omega| < \frac{\omega_s}{2}$$

Therefore, we can form

$$H_D(z) = \frac{H_0 H_{D1}(z)}{H_{D2}(z)} \tag{11.11}$$

such that

$$H_D(e^{j\omega T}) = \frac{H_0 H_{D1}(e^{j\omega T})}{H_{D2}(e^{j\omega T})} \approx H_A(j\omega) \qquad \text{for } |\omega| < \frac{\omega_s}{2}$$

If the zeros and poles of $H_A(s)$ are simple, Eq. (11.7) gives

$$H_{D1}(z) = \sum_{i=1}^{N} \frac{A_i z}{z - e^{Tp_i}} = \frac{N_1(z)}{D_1(z)} \tag{11.12}$$

$$H_{D2}(z) = \sum_{i=1}^{M} \frac{B_i z}{z - e^{Tz_i}} = \frac{N_2(z)}{D_2(z)} \tag{11.13}$$

[1] *These are filters that have only poles in the finite s plane.*

Thus from Eqs. (11.11)–(11.13)

$$H_D(z) = \frac{H_0 N_1(z) D_2(z)}{N_2(z) D_1(z)} \tag{11.14}$$

The derived filter can be unstable since some of the zeros of $N_2(z)$ may be located on or outside the unit circle of the z plane, but the problem can be easily overcome. For an arbitrary pole of $H_D(z)$, say, p_i, we can write

$$\begin{aligned}|(e^{j\omega T} - p_i)| &= \left| -e^{j\omega T} p_i \left(e^{-j\omega T} - \frac{1}{p_i} \right) \right| \\ &= |p_i| \left| \left(e^{j\omega T} - \frac{1}{p_i^*} \right)^* \right| \\ &= |p_i| \left| \left(e^{j\omega T} - \frac{1}{p_i^*} \right) \right|\end{aligned}$$

If p_i is real, we have

$$|(e^{j\omega T} - p_i)| = |p_i| \left| \left(e^{j\omega T} - \frac{1}{p_i} \right) \right|$$

and if p_i and p_i^* are a complex-conjugate pair of poles, then

$$|(e^{j\omega T} - p_i)(e^{j\omega T} - p_i^*)| = |p_i|^2 \left| \left(e^{j\omega T} - \frac{1}{p_i} \right) \left(e^{j\omega T} - \frac{1}{p_i^*} \right) \right|$$

Hence any poles of $H_D(z)$ located outside the unit circle can be replaced by their reciprocals without changing the shape of the loss characteristic. This will introduce a constant vertical shift in the loss characteristic but the problem can be easily eliminated by adjusting H_0, the multiplier constant of the transfer function.

The method yields excellent results for elliptic filters. For this class of filters, polynomial $N_2(z)$ turns out to be a mirror-image polynomial with roots on the negative real axis (see Sec. 9.2.3) and, in effect, its roots occur in reciprocal pairs. This means that half of the roots of $N_2(z)$ would be located outside the unit circle of the z plane and since these roots are poles in the derived transfer function given by Eq. (11.14), the filter obtained would be unstable. However, the problem can be easily eliminated by applying the above stabilization technique. If $N_2(z)$ has K roots p_i that are located outside the unit circle, a stable filter can be obtained by replacing each p_i by $1/p_i$ and then dividing the multiplier constant H_0 in Eq. (11.14) by $\prod_1^K |p_i|$.

The main problem with the modified invariant impulse-response method has to do with the order of the filter obtained. Unfortunately, polynomials $N_1(z)$ and $N_2(z)$ tend to increase the order of $H_D(z)$, as can be seen in Eq. (11.14), and that makes the method uneconomical. The method is described in some detail here only because it provides a theoretical foundation for the matched-z transformation method as will be demonstrated in Sec. 11.5.

Example 11.2 The transfer function

$$H_A(s) = H_0 \prod_{j=1}^{3} \frac{a_{0j} + s^2}{b_{0j} + b_{1j}s + s^2}$$

where H_0, a_{0j}, and b_{1j} are given in Table 11.2, represents a lowpass elliptic filter satisfying the following specifications:

- Passband ripple: 0.1 dB
- Minimum stopband loss: 43.46 dB
- Passband edge: $\sqrt{0.8}$ rad/s
- Stopband edge: $1/\sqrt{0.8}$ rad/s

Employing the modified invariant impulse-response method, design a corresponding digital filter. Use $\omega_s = 7.5$ rad/s.

Table 11.2 Coefficients of $H_A(s)$ (Example 11.2)

j	a_{0j}	b_{0j}	b_{1j}
1	$1.199341E+1$	$3.581929E-1$	$9.508335E-1$
2	2.000130	$6.860742E-1$	$4.423164E-1$
3	1.302358	$8.633304E-1$	$1.088749E-1$
$H_0 = 6.713267E-3$			

■ **Solution**

From Eqs. (11.9) and (11.10)

$$H_{A1}(s) = \prod_{j=1}^{3} \frac{1}{b_{0j} + b_{1j}s + s^2}$$

$$H_{A2}(s) = \prod_{j=1}^{3} \frac{1}{a_{0j} + s^2}$$

The design can be accomplished by using the following procedure:

1. Find the poles and residues of $H_{A1}(s)$ and $H_{A2}(s)$.
2. Form $H_{D1}(z)$ and $H_{D2}(z)$ using Eqs. (11.12) and (11.13).
3. Replace zeros of $N_2(z)$ outside the unit circle by their reciprocals.
4. Adjust constant H_0 to achieve zero minimum passband loss.

Table 11.3 Coefficients of $H_D(z)$ (Example 11.2)

j	a_{0j}	a_{1j}	b_{0j}	b_{1j}
1	1.0	1.942528	$4.508735E-1$	-1.281134
2	1.0	$-7.530225E-1$	$6.903732E-1$	-1.303838
3	1.0	-1.153491	$9.128252E-1$	-1.362371
4	$3.248990E+1$	$1.955491E+1$	$5.611278E-2$	$7.751650E-1$
5	$1.331746E-2$	$3.971465E-1$	$5.611278E-2$	$7.751650E-1$
$H_0 = 3.847141E-4$				

With this procedure $H_D(z)$ can be deduced as

$$H_D(z) = H_0 \prod_{j=1}^{5} \frac{a_{0j} + a_{1j}z + z^2}{b_{0j} + b_{1j}z + z^2} \qquad \blacksquare$$

where H_0, a_{ij}, and b_{ij} are given in Table 11.3.

The loss characteristic achieved, plotted in Fig. 11.3*a* and *b*, is seen to be a faithful reproduction of the analog loss characteristic. For this filter, the conventional invariant impulse-response method gives unsatisfactory results because the assumptions of Eqs. (11.3a) and (11.4) are violated.

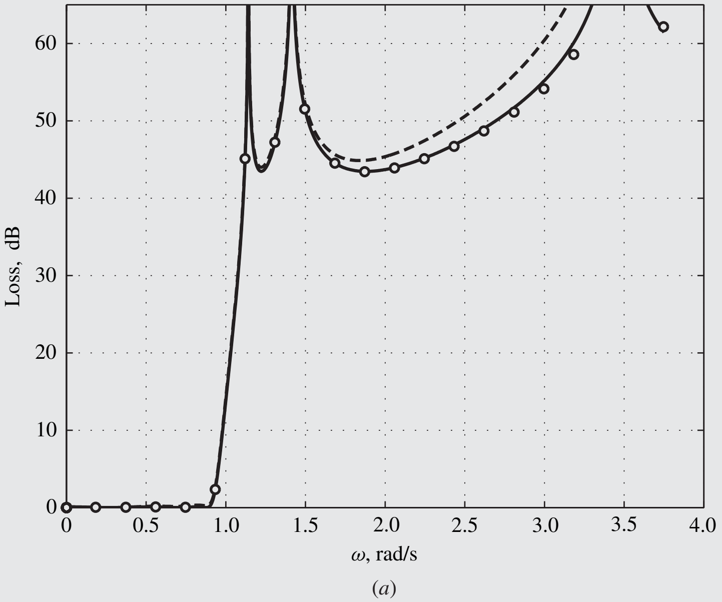


Figure 11.3 Examples 11.2 and 11.4: (*a*) Stopband characteristics.
— Analog filter; ○ ○ ○ modified impulse-invariant response method;
- - - - - matched-z transformation method.

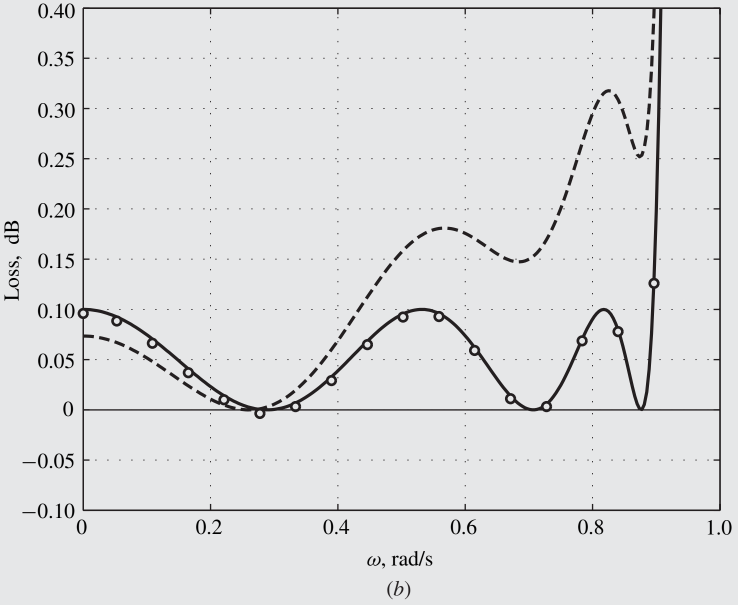


Figure 11.3 Cont'd Examples 11.2 and 11.4: (*b*) Passband characteristics.
— Analog filter; ○ ○ ○ modified impulse-invariant response method;
- - - - - matched-*z* transformation method.

11.5 MATCHED-*Z* TRANSFORMATION METHOD

An alternative approximation method for the design of recursive filters is the so-called *matched-z transformation method* [5, 9]. In this method, given a continuous-time transfer function like that in Eq. (11.8), a corresponding discrete-time transfer function can be formed as

$$H_D(z) = (z+1)^L \frac{H_0 \prod_{i=1}^{M}(z - e^{z_i T})}{\prod_{i=1}^{N}(z - e^{p_i T})} \tag{11.15}$$

where L is an integer. The value of L is equal to the number of zeros at $s = \infty$ in $H_A(s)$. Typical values for L are given in Table 11.4.

The matched-z transformation method had little or no theoretical foundation when it first found its way into the technical literature. It was probably observed that in the invariant impulse-response method, the ith pole of the digital filter, p_i', is related to the ith pole of the analog filter, p_i, through the relation

$$p_i' = e^{p_i T}$$

It did not take too long for someone to attempt to map the zeros of the analog filter in the same way by letting

$$z_i' = e^{z_i T}$$

Table 11.4 Typical values of L in Eq. (11.15)

Type of filter	Lowpass	Highpass	Bandpass	Bandstop
Butterworth	N	0	$N/2$	0
Chebyshev	N	0	$N/2$	0
Inverse-Chebyshev,				
N odd	1	0	N/A	N/A
N even	0	0	1 for $N/2$ odd 0 for $N/2$ even	0
Elliptic,				
N odd	1	0	N/A	N/A
N even	0	0	1 for $N/2$ odd 0 for $N/2$ even	0

This matched-z transformation seemed to work for highpass and bandstop filters and to make the method work for lowpass and bandpass filters as well the fi ddle factor' $(z+1)^L$ was introduced, which improved the situation.

If we now compare Eqs. (11.14) and (11.15), we note that the only difference between the transfer function of the modified invariant impulse-response method and the matched-z transformation method is that the ratio of polynomials, $N_1(z)/N_2(z)$, in Eq. (11.14) is replaced by the factor $(z+1)^L$. For the standard filter approximations, it turns out that

$$\frac{N_1(z)}{N_2(z)} \approx (z+1)^L$$

and this is why the matched-z transformation method works as well as it does.

Note that the matched-z transformation method cannot control the level of the amplitude response, i.e., if one starts with an analog filter that has a maximum passband gain of unity, the maximum passband gain in the derived digital filter is always something other than unity and, therefore, a correction is necessary after the design is completed to restore the maximum passband gain to unity. This can be easily achieved by finding the maximum passband gain of the derived digital filter, say, $M_{\max}$, and then changing the multiplier constant in Eq. 11.15 from H_0 to $H_0' = H_0/M_{\max}$.

The method is fairly simple to apply and gives reasonable results provided that a sufficiently large sampling frequency is used. Its main disadvantage is that it introduces a relatively large error in the passband loss as will be seen in the following example.

Example 11.3 The transfer function

$$H_A(s) = \frac{H_0 s^4}{\prod_{j=1}^{2}(s-p_j)(s-p_j^*)}$$

where

$$H_0 = 0.9885531 \qquad p_1, p_1^* = -2.047535 \pm j1.492958$$

$$p_2, p_2^* = -0.3972182 \pm j1.688095$$

represents a highpass Chebyshev filter with a passband edge of 2 rad/s and a passband ripple of 0.1 dB. Obtain a corresponding discrete-time transfer function employing the matched-z transformation method. Use a sampling frequency of 10 rad/s.

■ **Solution**

The value of L in Eq. (11.15) is generally zero for highpass filters, according to Table 11.4. Hence $H_D(z)$ can be readily formed as

$$H_D(z) = H_0 \frac{(1 - 2z + z^2)^2}{\prod_{j=1}^{2}(b_{0j} + b_{1j}z + z^2)} \quad ■$$

where

$$b_{01} = 7.630567 \times 10^{-2} \qquad b_{11} = -3.267079 \times 10^{-1}$$

$$b_{02} = 6.070409 \times 10^{-1} \qquad b_{12} = -7.608887 \times 10^{-1}$$

$$H_0 = 2.076398 \times 10^{-1}$$

The above value of H_0 was chosen to give a maximum passband gain of unity, which corresponds to a minimum passband loss of zero dB. The loss characteristic of the derived filter is compared with that of the analog filter in Fig. 11.4.

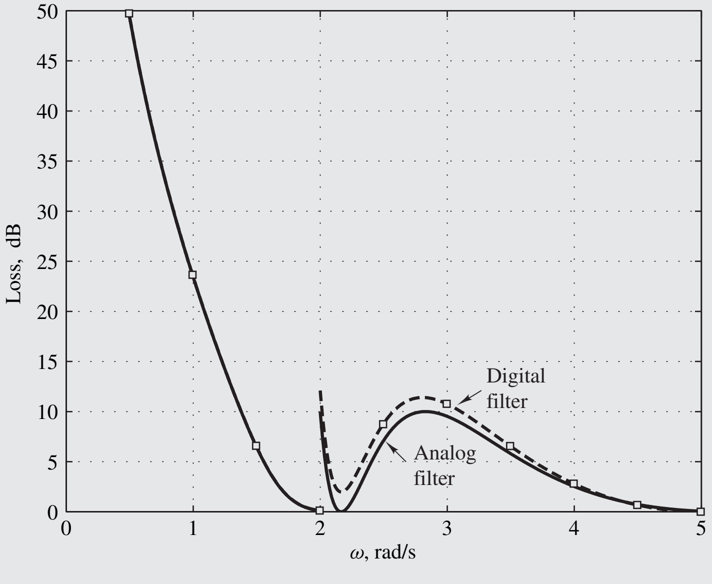


Figure 11.4 Loss characteristic (Example 11.3).

Example 11.4 Redesign the lowpass filter of Example 11.2 employing the matched-z transformation method.

■ **Solution**

From Eq. (11.15) $H_D(z)$ can be formed as

$$H_D(z) = H_0 \prod_{j=1}^{3} \frac{a_{0j} + a_{1j}z + z^2}{b_{0j} + b_{1j}z + z^2} \quad ■$$

where a_{1j} and b_{1j} are given by the first three rows in Table 11.3. For zero minimum passband loss, H_0 is given by

$$H_0 = 8.605074 \times 10^{-3}$$

The loss characteristic achieved is shown in Fig. 11.3 (dashed curve). As can be seen, it is significantly inferior to the loss characteristic obtained by using the modified invariant impulse-response method. However, as was mentioned earlier, the latter method leads to a higher-order transfer function which is more uneconomical.

11.6 BILINEAR-TRANSFORMATION METHOD

In the approximation method of Sec. 11.3, the derived digital filter has *exactly* the same impulse response as the original analog filter for $t = nT$. An approximation method will now be described whereby a digital filter is derived that has *approximately* the same time-domain response as the original analog filter *for any excitation.*

11.6.1 Derivation

Consider an *analog integrator* characterized by the transfer function

$$H_{AI}(s) = \frac{1}{s} \tag{11.16}$$

and assume that its response to an excitation $x(t)$ is $y(t)$, as depicted in Fig. 11.5.

The impulse response of the integrator is given by

$$\mathcal{L}^{-1}H_I(s) = h_I(t) = \begin{cases} 1 & \text{for } t \geq 0+ \\ 0 & \text{for } t \leq 0- \end{cases}$$

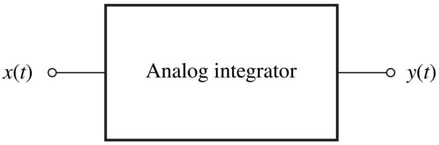


Figure 11.5 Analog integrator.

and its response at instant t to an arbitrary right-sided excitation $x(t)$, i.e., $x(t) = 0$ for $t < 0$, is given by the convolution integral (see Theorem 2.14)

$$y(t) = \int_0^t x(\tau)h_I(t-\tau)\,d\tau$$

(see Sec. 10.2.4). If $0+ < t_1 < t_2$, we can write

$$y(t_2) - y(t_1) = \int_0^{t_2} x(\tau)h_I(t_2-\tau)\,d\tau - \int_0^{t_1} x(\tau)h_I(t_1-\tau)\,d\tau \tag{11.17}$$

For $0+ < \tau \le t_1, t_2$

$$h_I(t_2-\tau) = h_I(t_1-\tau) = 1$$

and thus Eq. (11.17) simplifies to

$$y(t_2) - y(t_1) = \int_{t_1}^{t_2} x(\tau)\,d\tau$$

As $t_1 \to t_2$, from Fig. 11.6

$$y(t_2) - y(t_1) \approx \frac{t_2-t_1}{2}[x(t_1) + x(t_2)]$$

and on letting $t_1 = nT - T$ and $t_2 = nT$ the difference equation

$$y(nT) - y(nT-T) = \frac{T}{2}[x(nT-T) + x(nT)]$$

can be formed. This equation represents a '*digital integrator*' that has approximately the same time-domain response as the analog integrator for any excitation. By applying the z transform, we obtain

$$Y(z) - z^{-1}Y(z) = \frac{T}{2}[z^{-1}X(z) + X(z)]$$

and hence the transfer function of the digital integrator can be derived as

$$H_{DI}(z) = \frac{Y(z)}{X(z)} = \frac{T}{2}\left(\frac{z+1}{z-1}\right)$$

The above equation can be expressed as

$$H_{DI}(z) = \frac{1}{s}\bigg|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)} \tag{11.18}$$

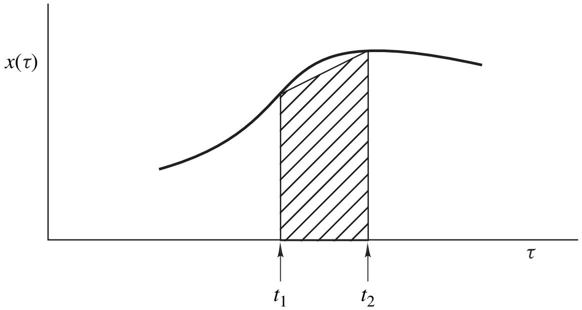


Figure 11.6 Response of analog integrator.

and, therefore, from Eqs. (11.16) and (11.18), we have

$$H_{DI}(z) = H_{AI}(s)\Big|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}$$

In effect, a digital integrator can be obtained from an analog integrator by simply applying the bilinear transformation[2]

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right) \tag{11.19}$$

to the transfer function of the analog integrator. As $T \to 0$, the shaded area in Fig. 11.6 would tend to approach the area under the curve and, consequently, the time-domain response of the digital integrator would tend to approach that of the analog integrator, as may be expected.

Applying the bilinear transformation to the transfer function of an arbitrary analog filter will yield a digital filter characterized by the discrete-time transfer function

$$H_D(z) = H_A(s)\Big|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)} \tag{11.20}$$

The digital filter so obtained will produce approximately the same time-domain response as the analog filter from which it is derived for any excitation. Furthermore, the time-domain response of the digital filter would tend to approach that of the analog filter as $T \to 0$.

The above time-domain analysis led to a transformation that can be used to obtain a digital filter from an analog one. All we know at this point is the relationship between the time-domain response of the digital filter with that of the analog filter. To make further progress, we must deduce the relationship between the frequency-domain response of the digital filter with that of the analog filter, and to do that we must examine the mapping properties of the bilinear transformation.

11.6.2 Mapping Properties of Bilinear Transformation

Equation (11.19) can be put in the form

$$z = \frac{2/T + s}{2/T - s}$$

and with $s = \sigma + j\omega$ we have

$$z = re^{j\theta}$$

where

$$r = \left[\frac{\left(\frac{2}{T}+\sigma\right)^2 + \omega^2}{\left(\frac{2}{T}-\sigma\right)^2 + \omega^2}\right]^{1/2}$$

and

$$\theta = \tan^{-1}\frac{\omega}{2/T + \sigma} + \tan^{-1}\frac{\omega}{2/T - \sigma} \tag{11.21}$$

[2] *The bilinear transformation is one of the standard conformal transformations (see Sec. A.9).*

Clearly

$$
\begin{array}{ll}
\text{if } \sigma > 0 & \text{then } r > 1 \\
\text{if } \sigma = 0 & \text{then } r = 1 \\
\text{if } \sigma < 0 & \text{then } r < 1
\end{array}
$$

i.e., the bilinear transformation maps

1) the open right-half s plane onto the region exterior to the unit circle $|z| = 1$ of the z plane,
2) the j axis of the s plane onto the unit circle $|z| = 1$, and
3) the open left-half s plane onto the interior of the unit circle $|z| = 1$.

For $\sigma = 0$, we have $r = 1$, and from Eq. (11.21) $\theta = 2\tan^{-1}(\omega T/2)$. Hence

$$
\begin{array}{ll}
\text{if } \omega = 0 & \text{then } \theta = 0 \\
\text{if } \omega \to +\infty & \text{then } \theta \to +\pi \\
\text{if } \omega \to -\infty & \text{then } \theta \to -\pi
\end{array}
$$

i.e., the origin of the s plane maps onto point $(1, 0)$ of the z plane and the positive and negative j axes of the s plane map onto the upper and lower semicircles $|z| = 1$, respectively. The transformation is illustrated in Fig. 11.7a and b.

From Property 2 above it follows that the maxima and minima of $|H_A(j\omega)|$ will be preserved in $|H_D(e^{j\Omega T})|$. Also if

$$M_1 \le |H_A(j\omega)| \le M_2$$

for some frequency range $\omega_1 \le \omega \le \omega_2$, then

$$M_1 \le |H_D(e^{j\Omega T})| \le M_2$$

for a corresponding frequency range $\Omega_1 \le \Omega \le \Omega_2$. Consequently, passbands or stopbands in the analog filter translate into passbands or stopbands in the digital filter.

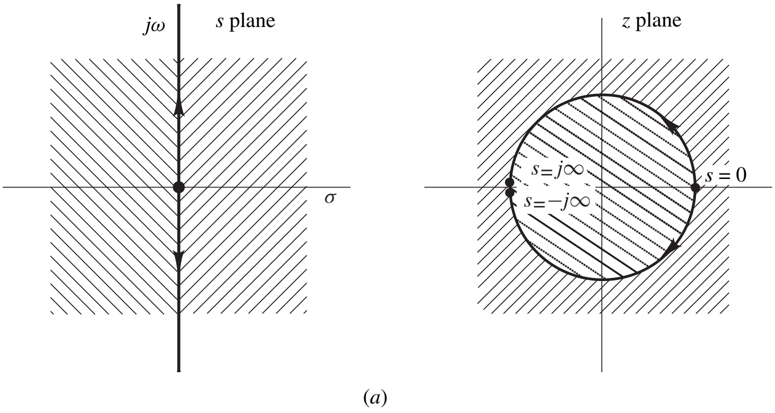
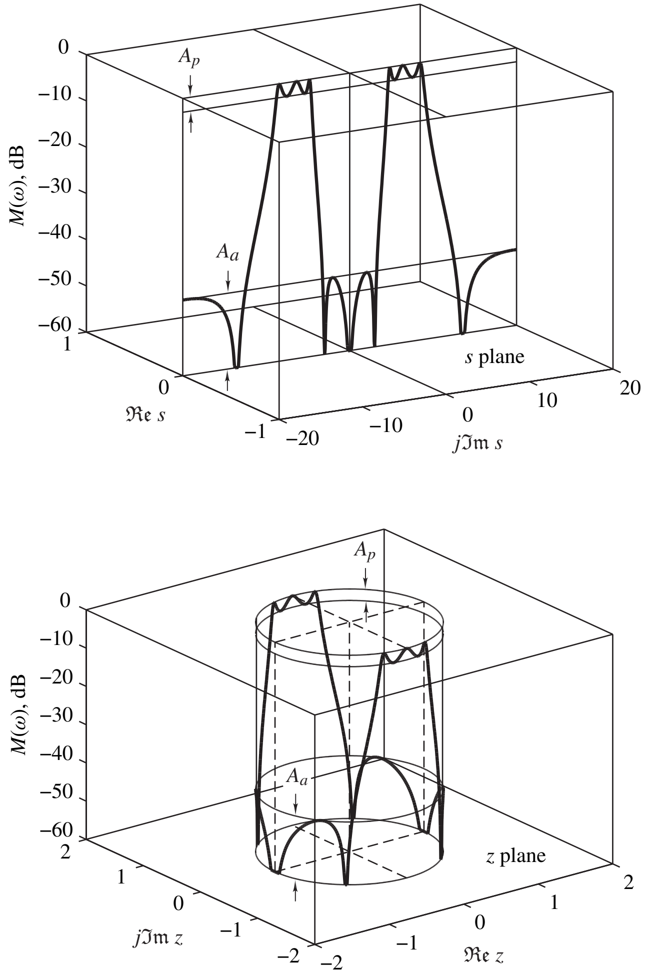


Figure 11.7 Bilinear transformation: (a) Mapping from s to z plane.



(*b*)

Figure 11.7 Cont'd Bilinear transformation: (*b*) Mapping of amplitude response of analog filter to the z domain.

From Property 3 it follows that a stable analog filter will yield a stable digital filter, and since the transformation has real coefficients, $H_D(z)$ will have real coefficients. Finally, the numerator degree in $H_D(z)$ cannot exceed the denominator degree and, therefore, $H_D(z)$ is a realizable transfer function.

11.6.3 The Warping Effect

Let ω and Ω represent the frequency variable in the analog filter and the derived digital filter, respectively. From Eq. (11.20)

$$H_D(e^{j\Omega T}) = H_A(j\omega)$$

provided that

$$\omega = \frac{2}{T} \tan \frac{\Omega T}{2} \tag{11.22}$$

For $\Omega < 0.3/T$

$$\omega \approx \Omega$$

and, as a result, the digital filter has the same frequency response as the analog filter. For higher frequencies, however, the relation between ω and Ω becomes nonlinear, as illustrated in Fig. 11.8, and distortion is introduced in the frequency scale of the digital filter relative to that of the analog filter. This is known as the *warping effect* [2, 5].

The influence of the warping effect on the amplitude response can be demonstrated by considering an analog filter with a number of uniformly spaced passbands centered at regular intervals, as in Fig. 11.8. The derived digital filter has the same number of passbands, but the center frequencies and bandwidths of higher-frequency passbands tend to be reduced disproportionately, as shown in Fig. 11.8.

If only the amplitude response is of concern, the warping effect can for all practical purposes be eliminated by *prewarping* the analog filter [2, 5]. Let $\omega_1, \omega_2, \ldots, \omega_i, \ldots$ be the passband and stopband edges in the analog filter. The corresponding passband and stopband edges in the digital

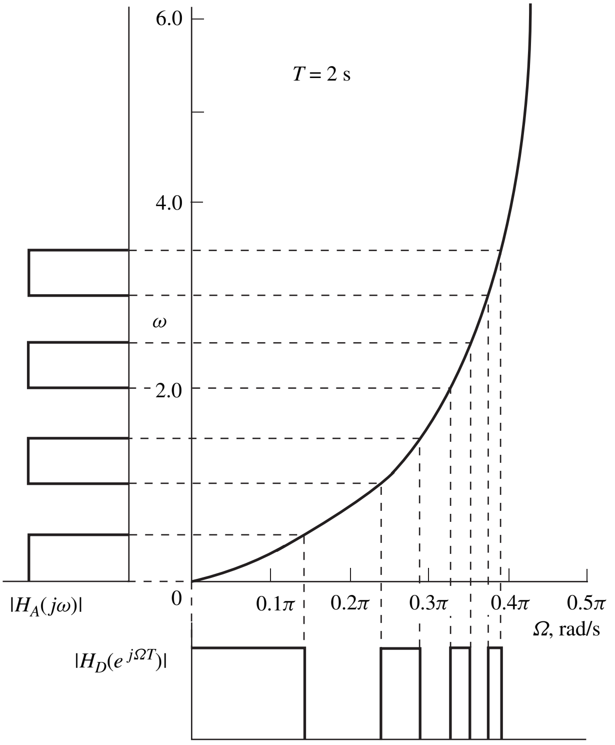


Figure 11.8 Influence of the warping effect on the amplitude response.

filter are given by Eq. (11.22) as

$$\Omega_i = \frac{2}{T}\tan^{-1}\frac{\omega_i T}{2} \qquad \text{for } i = 1, 2, \ldots \tag{11.23}$$

Consequently, if prescribed passband and stopband edges $\tilde{\Omega}_1, \tilde{\Omega}_2, \ldots, \tilde{\Omega}_i, \ldots$ are to be achieved in the digital filter, the analog filter must be prewarped before application of the bilinear transformation to ensure that

$$\omega_i = \frac{2}{T}\tan\frac{\tilde{\Omega}_i T}{2} \tag{11.24}$$

Under these circumstances

$$\Omega_i = \tilde{\Omega}_i$$

according to Eqs. (11.23) and (11.24), as required.

The bilinear transformation together with the prewarping technique is used in Chap. 12 to develop a detailed procedure for the design of Butterworth, Chebyshev, inverse-Chebyshev, and elliptic filters satisfying prescribed loss specifications.

The influence of the warping effect on the phase response can be demonstrated by considering an analog filter with linear phase response. As illustrated in Fig. 11.9, the phase response of the derived digital filter is nonlinear. Furthermore, little can be done to linearize it except by employing delay equalization (see Sec. 12.5.1). Consequently, if it is mandatory to preserve a linear phase response, the alternative methods of Secs. 11.3–11.4 should be considered.

Example 11.5 The transfer function

$$H_A(s) = \prod_{j=1}^{3}\frac{a_{0j} + s^2}{b_{0j} + b_{1j}s + s^2}$$

where a_{0j} and b_{ij} are given in Table 11.5, represents an elliptic bandstop filter with a passband ripple of 1 dB and a minimum stopband loss of 34.45 dB. Use the bilinear transformation to obtain a corresponding digital filter. Assume a sampling frequency of 10 rad/s.

Table 11.5 Coefficients of $H_A(s)$ (Example 11.5)

j	a_{0j}	b_{0j}	b_{1j}
1	6.250000	6.250000	2.618910
2	8.013554	$1.076433E+1$	$3.843113E-1$
3	4.874554	3.628885	$2.231394E-1$

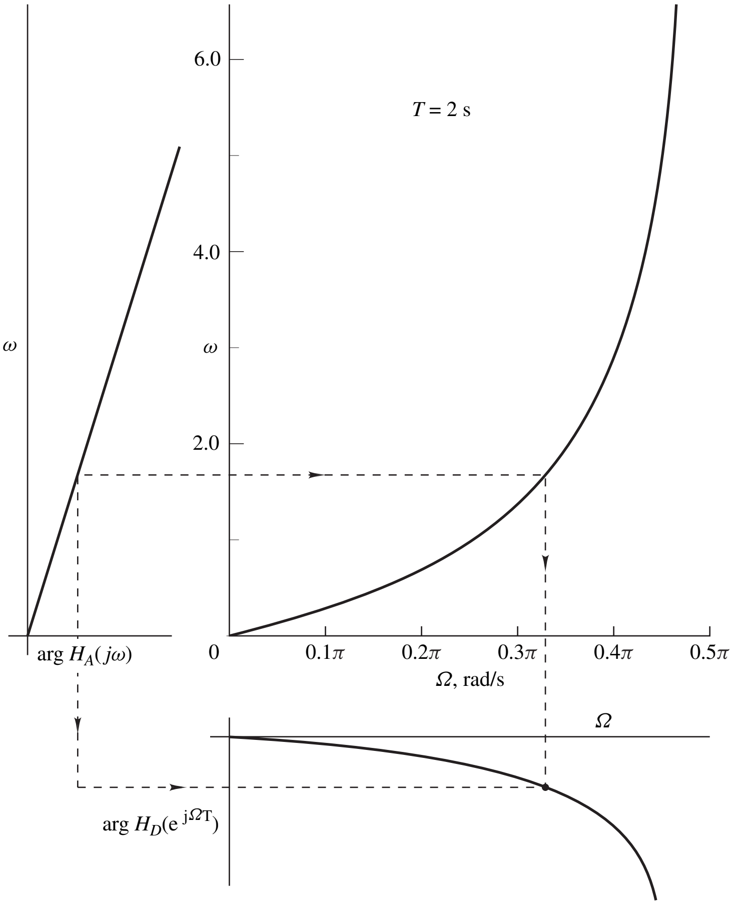


Figure 11.9 Influence of the warping effect on the phase response.

■ **Solution**

From Eq. (11.20), one can show that

$$H_D(z) = \prod_{j=1}^{3} \frac{a'_{0j} + a'_{1j}z + a'_{0j}z^2}{b'_{0j} + b'_{1j}z + z^2} \quad ■$$

where

$$a'_{0j} = \frac{a_{0j} + 4/T^2}{c_j} \qquad a'_{1j} = \frac{2(a_{0j} - 4/T^2)}{c_j}$$

$$b'_{0j} = \frac{b_{0j} - 2b_{1j}/T + 4/T^2}{c_j} \qquad b'_{1j} = \frac{2(b_{0j} - 4/T^2)}{c_j}$$

$$c_j = b_{0j} + \frac{2b_{1j}}{T} + \frac{4}{T^2}$$

The numerical values of a'_{ij} and b'_{ij} are given in Table 11.6. The loss characteristic of the derived digital filter is compared with that of the analog filter in Fig. 11.10. The expected lateral displacement in the characteristic of the digital filter is evident.

Table 11.6 Coefficients of $H_D(z)$ (Example 11.5)

j	a'_{0j}	a'_{1j}	b'_{0j}	b'_{1j}
1	$6.627508E-1$	$-3.141080E-1$	$3.255016E-1$	$-3.141080E-1$
2	$8.203382E-1$	$-1.915542E-1$	$8.893929E-1$	$5.716237E-2$
3	1.036997	$-7.266206E-1$	$9.018366E-1$	$-8.987781E-1$

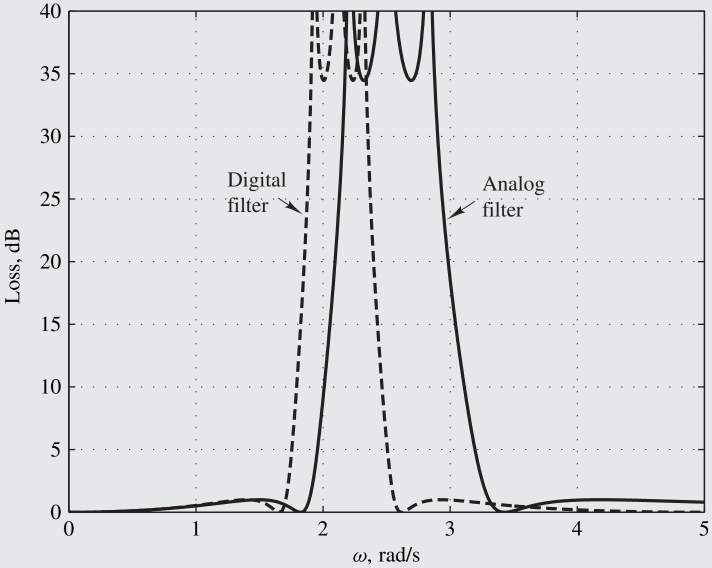


Figure 11.10 Loss characteristic (Example 11.5).

11.7 DIGITAL-FILTER TRANSFORMATIONS

A normalized lowpass analog filter can be transformed into a denormalized lowpass, highpass, bandpass, or bandstop filter by employing the transformations described in Sec. 10.8. Analogous transformations can be derived for digital filters as we shall now show. These are due to Constantinides [12].

11.7.1 General Transformation

Consider the transformation

$$z = f(\bar{z}) = e^{j\zeta\pi} \prod_{i=1}^{m} \frac{\bar{z} - a_i^*}{1 - a_i \bar{z}} \tag{11.25}$$

where ζ and m are integers and a_i^* is the complex conjugate of a_i. With $z = Re^{j\Omega T}$, $\bar{z} = re^{j\omega T}$, and $a_i = c_i e^{j\psi_i}$, Eq. (11.25) becomes

$$Re^{j\Omega T} = e^{j\zeta\pi} \prod_{i=1}^{m} \frac{re^{j\omega T} - c_i e^{-j\psi_i}}{1 - rc_i e^{j(\omega T + \psi_i)}}$$

and hence

$$R^2 = \prod_{i=1}^{m} \frac{r^2 + c_i^2 - 2rc_i \cos(\omega T + \psi_i)}{1 + (rc_i)^2 - 2rc_i \cos(\omega T + \psi_i)} \tag{11.26}$$

Evidently,

$$\begin{array}{llll} \text{if } R > 1 & \text{then } r^2 + c_i^2 > 1 + (rc_i)^2 & \text{or} & r > 1 \\ \text{if } R = 1 & \text{then } r^2 + c_i^2 = 1 + (rc_i)^2 & \text{or} & r = 1 \\ \text{if } R < 1 & \text{then } r^2 + c_i^2 < 1 + (rc_i)^2 & \text{or} & r < 1 \end{array}$$

In effect, Eq. (11.26) maps

1. the unit circle $|z| = 1$ onto the unit circle $|\bar{z}| = 1$,
2. the interior of $|z| = 1$ onto the interior of $|\bar{z}| = 1$, and
3. the exterior of $|z| = 1$ onto the exterior of $|\bar{z}| = 1$

as illustrated in Fig. 11.11.

Now consider a normalized lowpass filter characterized by $H_N(z)$ with a passband extending from 0 to Ω_p. On applying the above transformation we can form

$$H(\bar{z}) = H_N(z)\Big|_{z=f(\bar{z})} \tag{11.27}$$

With the poles of $H_N(z)$, located inside the unit circle $|z| = 1$, those of $H(\bar{z})$ will be located inside the unit circle $|\bar{z}| = 1$; that is, $H(\bar{z})$ will represent a stable filter. Furthermore, from item 1, if

$$M_1 \leq |H_N(e^{j\Omega T})| \leq M_2$$

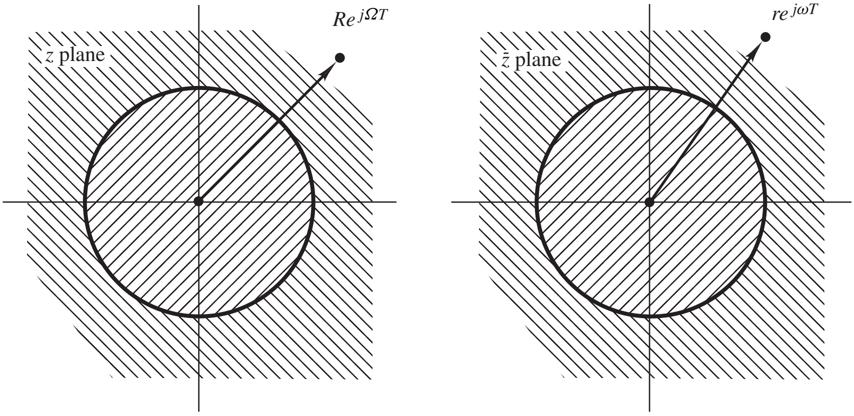


Figure 11.11 General z-domain transformation.

for some frequency range $\Omega_1 \leq \Omega \leq \Omega_2$, then

$$M_1 \leq |H(e^{j\omega T})| \leq M_2$$

for one or more corresponding ranges of ω; that is, the passband (stopband) in $H_N(z)$ will translate into one or more passbands (stopbands) in $H(\bar{z})$. Therefore, the above transformation can form the basis of a set of transformations that can be used to derive transformed lowpass, highpass, bandpass, and bandstop digital filters from a given lowpass digital filter.

11.7.2 Lowpass-to-Lowpass Transformation

The appropriate values for ζ, m, and a_i in Eq. (11.25) can be determined by examining the details of the necessary mapping. If $H(\bar{z})$ is to represent a *lowpass* filter with a passband edge ω_p, the mapping must be of the form shown in Fig. 11.12*a*, where solid lines denote passbands. As complex number $e^{j\Omega T}$ traces the unit circle in the z plane once, $e^{j\omega T}$ must trace the unit circle in the $\bar{z}$ plane once in the same sense. The transformation must thus be bilinear ($m = 1$) of the form

$$z = e^{j\zeta\pi} \frac{\bar{z} - a^*}{1 - a\bar{z}} \tag{11.28}$$

At points A and A', $z = \bar{z} = 1$, and at C and C', $z = \bar{z} = -1$ Hence Eq. (11.28) gives

$$1 = e^{j\zeta\pi} \frac{1 - a^*}{1 - a} \qquad \text{and} \qquad 1 = e^{j\zeta\pi} \frac{1 + a^*}{1 + a}$$

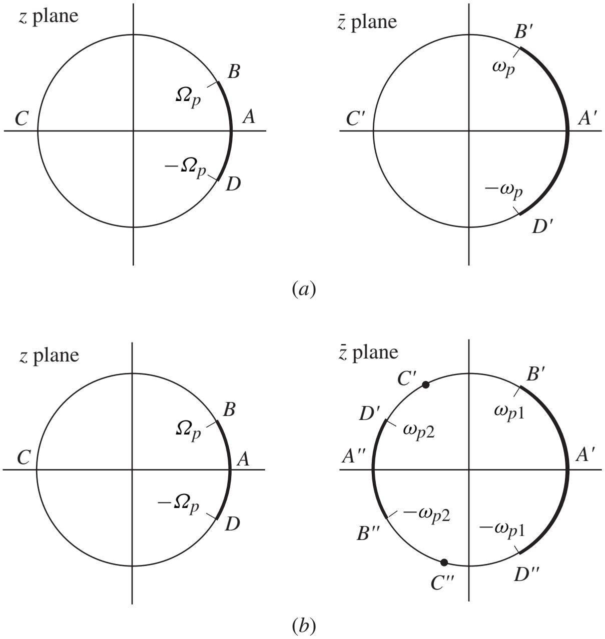


Figure 11.12 (*a*) Lowpass-to-lowpass transformation, (*b*) lowpass-to-bandstop transformation.

By solving these equations, we obtain

$$a = a^* \equiv \alpha \quad \zeta = 0$$

where α is a real constant. Thus Eq. (11.28) becomes

$$z = \frac{\bar{z} - \alpha}{1 - \alpha\bar{z}}$$

The necessary value for α can be determined by noting that at points B and B', we have $\Omega = \Omega_p$ and $\omega = \omega_p$, in which case

$$e^{j\Omega_p T} = \frac{e^{j\omega_p T} - \alpha}{1 - \alpha e^{j\omega_p T}}$$

or

$$\alpha = \frac{\sin[(\Omega_p - \omega_p)T/2]}{\sin[(\Omega_p + \omega_p)T/2]}$$

11.7.3 Lowpass-to-Bandstop Transformation

If a *bandstop* filter is required with passband edges ω_{p1} and ω_{p2}, the mapping must have the form shown in Fig. 11.12*b*. In order to introduce an upper passband in $H(\bar{z})$, $e^{j\Omega T}$ must trace the unit circle of the z plane twice for each revolution of $e^{j\omega T}$ in the $\bar{z}$ plane. Consequently, in this case, the transformation must be biquadratic ($m = 2$) of the form

$$z = e^{j\zeta\pi} \frac{\bar{z}^2 + \beta\bar{z} + \gamma}{1 + \beta\bar{z} + \gamma\bar{z}^2}$$

where β and γ are real constants. At points A and A', $z = \bar{z} = 1$ and

$$e^{j\zeta\pi} = 1$$

so that

$$z = \frac{\bar{z}^2 + \beta\bar{z} + \gamma}{1 + \beta\bar{z} + \gamma\bar{z}^2}$$

With $z = e^{j\Omega T}$ and $\bar{z} = e^{j\omega T}$

$$e^{j\Omega T} = \frac{e^{j2\omega T} + \beta e^{j\omega T} + \gamma}{e^{j2\omega T}(e^{-j2\omega T} + \beta e^{-j\omega T} + \gamma)}$$

Hence

$$\frac{\Omega T}{2} = \tan^{-1} \frac{\sin 2\omega T + \beta \sin \omega T}{\cos 2\omega T + \beta \cos \omega T + \gamma} - \omega T$$

and after some manipulation

$$\tan \frac{\Omega T}{2} = \frac{(1 - \gamma)\sin \omega T}{(1 + \gamma)\cos \omega T + \beta}$$

At points B and B', $\Omega = \Omega_p$ and $\omega = \omega_{p1}$, respectively, and as a result

$$\tan\frac{\Omega_p T}{2} = \frac{(1-\gamma)\sin\omega_{p1}T}{(1+\gamma)\cos\omega_{p1}T + \beta} \tag{11.29}$$

Likewise, at points D and D', $\Omega = -\Omega_p$ and $\omega = \omega_{p2}$, respectively, so that

$$\tan\frac{-\Omega_p T}{2} = \frac{(1-\gamma)\sin\omega_{p2}T}{(1+\gamma)\cos\omega_{p2}T + \beta} \tag{11.30}$$

Now by solving Eqs. (11.29) and (11.30), β and γ can be deduced as

$$\beta = -\frac{2\alpha}{1+k} \qquad \gamma = \frac{1-k}{1+k}$$

where

$$\alpha = \frac{\cos[(\omega_{p2}+\omega_{p1})T/2]}{\cos[(\omega_{p2}-\omega_{p1})T/2]} \qquad \text{and} \qquad k = \tan\frac{\Omega_p T}{2}\tan\frac{(\omega_{p2}-\omega_{p1})T}{2}$$

Lowpass-to-highpass and *lowpass-to-bandpass* transformations can similarly be derived. The complete set of transformations is summarized in Table 11.7.

Table 11.7 Constantinides transformations

Type	Transformation	α, k
LP to LP	$z = \dfrac{\bar{z}-\alpha}{1-\alpha\bar{z}}$	$\alpha = \dfrac{\sin[(\Omega_p-\omega_p)T/2]}{\sin[(\Omega_p+\omega_p)T/2]}$
LP to HP	$z = -\dfrac{\bar{z}-\alpha}{1-\alpha\bar{z}}$	$\alpha = \dfrac{\cos[(\Omega_p-\omega_p)T/2]}{\cos[(\Omega_p+\omega_p)T/2]}$
LP to BP	$z = -\dfrac{\bar{z}^2 - \frac{2\alpha k}{k+1}\bar{z} + \frac{k-1}{k+1}}{1 - \frac{2\alpha k}{k+1}\bar{z} + \frac{k-1}{k+1}\bar{z}^2}$	$\alpha = \dfrac{\cos[(\omega_{p2}+\omega_{p1})T/2]}{\cos[(\omega_{p2}-\omega_{p1})T/2]}$ $k = \tan\dfrac{\Omega_p T}{2}\cot\dfrac{(\omega_{p2}-\omega_{p1})T}{2}$
LP to BS	$z = \dfrac{\bar{z}^2 - \frac{2\alpha}{1+k}\bar{z} + \frac{1-k}{1+k}}{1 - \frac{2\alpha}{1+k}\bar{z} + \frac{1-k}{1+k}\bar{z}^2}$	$\alpha = \dfrac{\cos[(\omega_{p2}+\omega_{p1})T/2]}{\cos[(\omega_{p2}-\omega_{p1})T/2]}$ $k = \tan\dfrac{\Omega_p T}{2}\tan\dfrac{(\omega_{p2}-\omega_{p1})T}{2}$

11.7.4 Application

The Constantinides transformations can be readily applied to design filters with prescribed passband edges. The following procedure can be employed:

1. Obtain a lowpass transfer function $H_N(z)$ using any approximation method.
2. Determine the passband edge Ω_p in $H_N(z)$.
3. Form $H(\bar{z})$ according to Eq. (11.27) using the appropriate transformation.

An important feature of filters designed by using this procedure is that the passband edge in lowpass or highpass filters can be varied by varying a single parameter, namely, α. Similarly, both the lower and upper passband edges in bandpass or bandstop filters can be varied by varying only a pair of parameters, namely, α and k [13].

An alternative design procedure by which prescribed passband as well as stopband edges can be achieved is described in Chap. 12.

11.8 COMPARISON BETWEEN RECURSIVE AND NONRECURSIVE DESIGNS

Before a solution is sought for the approximation problem, a choice must be made between a recursive and a nonrecursive design [14]. In recursive filters the poles of the transfer function can be placed anywhere inside the unit circle. A consequence of this degree of freedom is that high selectivity (i.e., narrow transition bands) can easily be achieved with low-order transfer functions. In nonrecursive filters, on the other hand, with the poles fixed at the origin, high selectivity can be achieved only by using a relatively high order for the transfer function. For the same filter specification the required order in a nonrecursive design can be as high as 5 to 10 times that in a recursive design. For example, the bandpass-filter specification in Example 9.5 can be met using a nonrecursive filter of order 52 or a recursive elliptic filter of order 8. In practice, the cost of a digital filter tends to increase and its speed tends to decrease as the order of the transfer function is increased. Hence, for *high-selectivity* applications where the delay characteristic is of secondary importance, the choice is expected to be a recursive design.

Constant group delay is mandatory for certain applications, e.g., in data transmission and image processing (see Sec. 5.7). For such applications, the choice is between a nonrecursive design and an *equalized* recursive design. If computational efficiency is unimportant (e.g., if the amount of data to be processed is small), a nonrecursive design based on the methods considered in Chap. 9 may be entirely acceptable. However, if computational efficiency is of prime importance (e.g., in real-time applications or in applications where massive amounts of data are to be processed), an optimal nonrecursive design based on the Remez exchange algorithm described in Chap. 15 or an equalized recursive design based on the method of Sec. 16.8 must be selected. Optimal nonrecursive designs are easier to obtain than equalized recursive designs. However, computational efficiency is significantly better in equalized recursive designs, particularly if a high selectivity is required.

Nonrecursive filters are naturally suited for certain applications, e.g., to perform numerical operations like interpolation, extrapolation, differentiation and integration. Further, owing to the fact that their impulse response is of finite duration, nonrecursive filters can be implemented in terms of *fast-Fourier transforms* without the need for a window function (see Sec. 7.8.).

In certain applications, the choice between a nonrecursive and a recursive design may be determined by other factors. For example, nonrecursive filters are always stable, owing to the absence of feedback, and this feature alone makes them the filters of choice for the implementation of adaptive filters (see Sec. 18.5). In these applications, the characteristics of the adaptive filter change with time on-line and if it is possible for the filter to become unstable, it will at some point. Recursive filters can also be used as adaptive filters but a special mechanism has to be incorporated in the filter implementation to prevent instability from arising. Nonrecursive filters tend to be relatively insensitive to quantization errors. In addition, their realizations are simple and regular and are highly attractive for very-large-scale integrated circuit implementation (see Sec. 8.3).

REFERENCES

[1] J. F. Kaiser, "Design methods for sampled data filters," *Proc. 1st Allerton Conf. Circuit Syst. Theory*, pp. 221–236, Nov. 1963.

[2] R. M. Golden and J. F. Kaiser, "Design of wideband sampled-data filters," *Bell Syst. Tech. J.*, vol. 43, pp. 1533–1546, July 1964.

[3] C. M. Rader and B. Gold, "Digital filter design techniques in the frequency domain," *Proc. IEEE*, vol. 55, pp. 149–171, Feb. 1967.

[4] D. J. Nowak and P. E. Schmid, "Introduction to digital filters," *IEEE Trans. Electromagn. Compat.,* vol. 10, pp. 210–220, June 1968.

[5] R. M. Golden, "Digital filter synthesis by sampled-data transformation," *IEEE Trans. Audio Electroacoust.*, vol. 16, pp. 321–329, Sept. 1968.

[6] A. J. Gibbs, "An introduction to digital filters," *Aust. Telecommun. Res.,* vol. 3, pp. 3–14, Nov. 1969.

[7] A. J. Gibbs, "The design of digital filters," *Aust. Telecommun. Res.*, vol. 4, pp. 29–34, Mar. 1970.

[8] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975.

[9] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[10] T. H. Crystal and L. Ehrman, "The design and applications of digital filters with complex coefficients," *IEEE Trans. Audio Electroacoust.*, vol. 16, pp. 315–320, Sept. 1968.

[11] P. A. Regalia, S. K. Mitra, and J. Fadavi-Ardekani, "Implementation of real coefficient digital filters using complex arithmetic," *IEEE Trans. Circuits Syst.*, vol. 34, pp. 345–353, Apr. 1987.

[12] A. G. Constantinides, "Spectral transformations for digital filters," *Proc. Inst. Elect. Eng.,* vol. 117, pp. 1585–1590, Aug. 1970.

[13] R. E. Crochiere and P. Penfield, Jr., "On the efficient design of bandpass digital filter structures," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 23, pp. 380–381, Aug. 1975.

[14] Rabiner, L. R., J. F. Kaiser, O. Herrmann, and M. T. Dolan, "Some comparisons between FIR and IIR digital filters," *Bell Syst. Tech. J.*, vol. 53, pp. 305–331, Feb. 1974.

[15] S. S. Haykin and R. Carnegie, "New method of synthesising linear digital filters based on convolution integral," *Proc. Inst. Elect. Eng.*, vol. 117, pp. 1063–1072, June 1970.

[16] S. A. White, "New method of synthesising linear digital filters based on convolution integral," *Proc. Inst. Elect. Eng.*, vol. 118, p. 348, Feb. 1971.

[17] A. Antoniou and C. Shekher, "Invariant sinusoid approximation method for recursive digital filters," *Electron. Lett.*, vol. 9, pp. 498–500, Oct. 1973.

PROBLEMS

11.1. By using the invariant impulse-response method, derive a discrete-time transfer function from the continuous-time transfer function

$$H_A(s) = \frac{1}{(s+1)(s^2+s+1)}$$

The sampling frequency is 10 rad/s.

11.2. A continuous-time system is characterized by the transfer function

$$H_A(s) = \frac{1}{s^3+6s^2+11s+6}$$

By using the invariant impulse-response method, obtain the transfer function of a corresponding discrete-time system. The sampling frequency is 6π.

11.3. A continuous-time system has a transfer function

$$H_A(s) = \frac{5.0}{s+2.5} - \frac{5s+13}{s^2+4s+\frac{25}{4}}$$

Using the invariant impulse-response method, obtain the transfer function of a corresponding discrete-time system. The sampling frequency is 20 rad/s.

11.4. A Bessel-Thomson normalized lowpass filter has a transfer function

$$H_A(s) = \frac{R_0}{s-p_0} + \frac{R_1}{s-p_1} + \frac{R_1^*}{s-p_1^*}$$

where

$$R_0 = 4.53, \qquad p_0 = -2.32$$
$$R_1 = -2.26 - j6.24, \qquad p_1 = -1.84 + j1.75$$

(*a*) Using the invariant impulse-response method, obtain the transfer function of a corresponding digital filter. The sampling frequency is 20 rad/s.

(*b*) Give an approximate expression for the frequency response of the digital filter in terms of the frequency response of the analog filter, $H_A(j\omega)$, and state what determines the accuracy of your expression.

11.5. The sixth-order normalized Bessel-Thomson transfer function can be expressed as

$$H_A(s) = \sum_{i=1}^{3} \left(\frac{R_i}{s-p_i} + \frac{R_i^*}{s-p_i^*} \right)$$

where R_i and p_i are given in Table P11.5.

(*a*) Design a digital filter by using the invariant impulse-response method, assuming a sampling frequency of 10 rad/s.

(*b*) Plot the phase response of the digital filter.

Table P11.5

i	p_i	R_i
1	$-4.248359 + j0.867510$	$10.95923 - j39.42517$
2	$-3.735708 + j2.626272$	$-14.12677 + j12.70117$
3	$-2.515932 + j4.492673$	$3.16754 - j0.202460$

11.6. A continuous-time system has a transfer function

$$H_A(s) = \frac{s^2 - 3s + 3}{s^2 + 3s + 3}$$

(*a*) Find the amplitude response.

(*b*) Can one design a corresponding discrete-time system by using the invariant impulse-response method? If so, carry out the design employing a sampling frequency of 10 rad/s. Otherwise, explain the reasons for the failure of the method.

11.7. A lowpass digital filter that would satisfy the specifications of Fig. P11.7 is required.

(*a*) Obtain a design by applying the invariant impulse-response method to an appropriate Chebyshev approximation. The sampling frequency is 20,000 rad/s.

(*b*) Check your design by plotting the amplitude responses of the original analog filter and the designed digital filter.

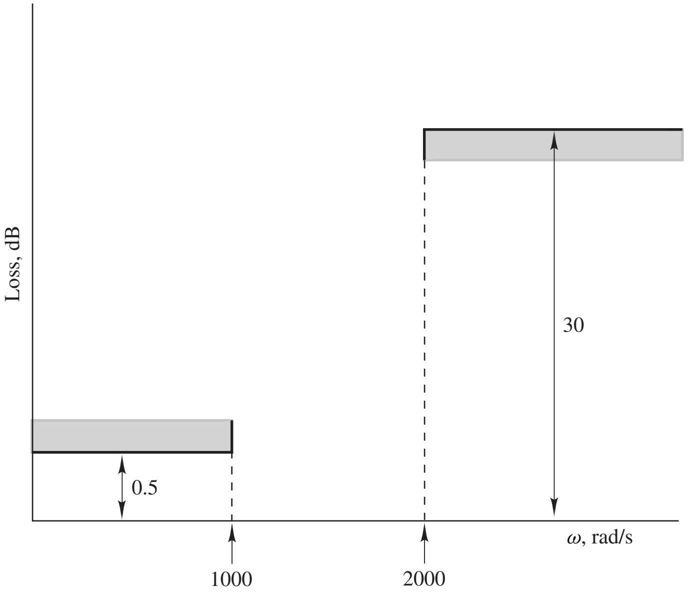


Figure P11.7

11.8. A bandpass filter is required with passband edges of 900 and 1600 rad/s and a maximum passband loss of 1.0 dB. Obtain a design by employing the invariant impulse-response method. Start with a second-order normalized lowpass Chebyshev approximation and neglect the effects of aliasing. A suitable sampling frequency is 10,000 rad/s.

11.9. Given an analog filter characterized by

$$H_A(s) = \sum_{i=1}^{N} \frac{R_i}{s - p_i}$$

a corresponding digital filter characterized by $H_D(z)$ can be derived such that

$$\mathcal{R}_A u(t)\Big|_{t=nT} = \mathcal{R}_D u(nT)$$

This is called the *invariant unit-step-response approximation method* [15, 16].

(*a*) Show that

$$H_D(e^{j\omega T}) \approx H_A(j\omega) \qquad \text{for } |\omega| < \frac{\omega_s}{2}$$

if $\omega \ll 1/T$ and

$$\frac{H_A(j\omega)}{j\omega} \approx 0 \qquad \text{for } |\omega| \geq \frac{\omega_s}{2}$$

(*b*) Show that

$$H_D(z) = \sum_{i=1}^{N} \frac{R_i'}{z - e^{p_i T}} \qquad \text{where } R_i' = \frac{(e^{p_i T} - 1)R_i}{p_i}$$

11.10. (*a*) Design a third-order digital filter by applying the invariant unit-step-response method (see Prob. 11.9) to the transfer function in Prob. 11.1. Assume that $\omega_s = 10$ rad/s.

(*b*) Compare the design with that obtained in Prob. 11.1.

11.11. Given an analog filter characterized by

$$H_A(s) = H_0 + \sum_{i=1}^{N} \frac{R_i}{s - p_i}$$

a corresponding digital filter characterized by $H_D(z)$ can be derived such that

$$\mathcal{R}_A u(t) \sin \omega_0 t \Big|_{t=nT} = \mathcal{R}_D u(nT) \sin \omega_0 nT$$

This is the so-called *invariant sinusoid-response approximation method* [17].

(*a*) Show that

$$H_D(e^{j\omega T}) \approx \frac{2\omega_0(\cos \omega T - \cos \omega_0 T)}{(\omega_0^2 - \omega^2)T \sin \omega_0 T} H_A(j\omega) \qquad \text{for } |\omega| < \frac{\omega_s}{2}$$

if $\omega \ll 1/T$ and

$$\frac{\omega_0 H_A(j\omega)}{\omega_0^2 - \omega^2} \approx 0 \qquad \text{for } |\omega| \geq \frac{\omega_s}{2}$$

(*b*) Show that

$$H_D(z) = H_0 + \sum_{i=1}^{N} \frac{U_i z + V_i}{z - e^{p_i T}}$$

where

$$U_i = (\omega_0 e^{p_i T} - p_i \sin \omega_0 T - \omega_0 \cos \omega_0 T)R_i'$$

$$V_i = [e^{p_i T}(p_i \sin \omega_0 T - \omega_0 \cos \omega_0 T) + \omega_0]R_i'$$

$$R_i' = \frac{R_i}{(p_i^2 + \omega_0^2) \sin \omega_0 T}$$

11.12. (*a*) Redesign the filter of Prob. 11.1 by employing the invariant sinusoid-response method (see Prob. 11.11). The value of ω_0 may be assumed to be 1 rad/s.

(*b*) Plot the resulting phase response.

11.13. Design a lowpass filter that would satisfy specifications of Fig. P11.7 by applying the modified invariant impulse-response method to a suitable elliptic approximation. The sampling frequency is 20,000 rad/s.

11.14. A given third-order lowpass Chebyshev filter has a transfer function

$$H(s) = \frac{H_0}{(s - p_0)(s - p_1)(s - p_1^*)}$$

where $H_0 = 0.49$, $p_0 = -0.494$, $p_1 = -0.247 + j0.966$, and p_1^* is the complex conjugate of p_1.

(*a*) Obtain a corresponding discrete-time transfer function using the matched-*z* transformation method assuming a sampling frequency of 20 rad/s.

(*b*) What are the advantages and disadvantages of the matched-*z* transformation method and the invariant impulse-response method?

11.15. (*a*) Design a digital filter by applying the matched-*z* transformation method to the inverse-Chebyshev filter of Example 10.5 assuming a sampling frequency of 20 rad/s.

(*b*) Check your design by plotting the amplitude responses of the original analog filter and the designed digital filter.

11.16. An analog elliptic lowpass filter with a passband edge $\omega_p = 0.7075$ rad/s has a transfer function of the form

$$H(s) = \frac{0.07488(s^2 + 2.577)}{(s + 0.3818)(s^2 + 0.3069s + 0.5053)}$$

(*a*) Obtain a continuous-time highpass transfer function with the same passband edge.

(*b*) Obtain a corresponding discrete-time highpass transfer function using the matched-*z* transformation method. The sampling frequency is 20 rad/s.

11.17. Design a lowpass filter that would satisfy the specifications of Fig. P11.7 by applying the matched-*z* transformation method to an appropriate Chebyshev approximation. The sampling frequency is 20,000 rad/s.

11.18. Design a highpass filter that would satisfy the specifications of Fig. P11.18. Use the matched-*z* transformation method along with an elliptic approximation. Assume that $\omega_s = 6000$ rad/s.

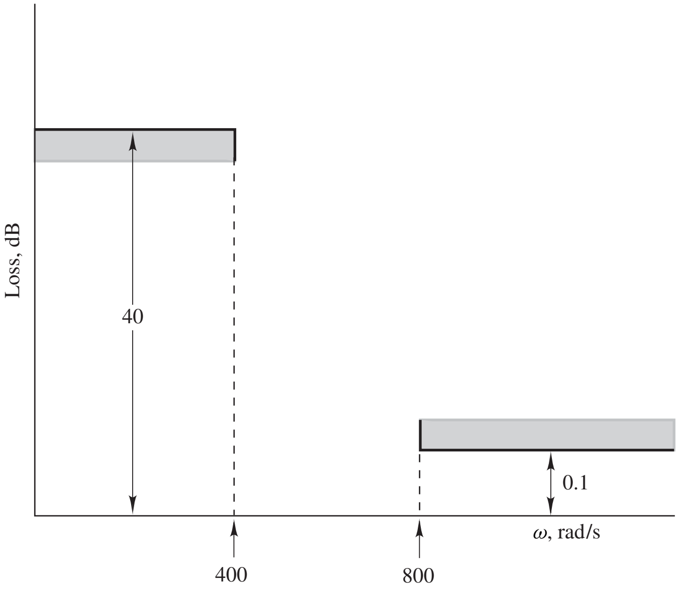


Figure P11.18

11.19. (*a*) Obtain a digital-filter network by applying the bilinear transformation to the transfer function

$$H_A(s) = \frac{s^2}{s^2 + \sqrt{2}s + 1}$$

(*b*) Evaluate the gain of the filter for $\omega = 0$ and $\omega = \pi/T$.

11.20. (*a*) Obtain a discrete-time transfer function by applying the bilinear transformation of the transfer function of Prob. 11.1. The sampling frequency is 4π rad/s.

(*b*) Determine the gain and phase-shift of the filter at $\omega = 0$ and $\omega = \pi$ rad/s.

11.21. Design a digital filter by applying the bilinear transformation to the Chebyshev transfer function of Example 10.3. The sampling frequency is 10 rad/s.

11.22. Design a digital filter by applying the bilinear transformation to the inverse-Chebyshev transfer function of Example 10.5. The sampling frequency is 20 rad/s.

11.23. The lowpass transfer function of Example 10.6 is subjected to the bilinear transformation.

(*a*) Assuming that $\omega_s = 10$ rad/s, find the resulting passband and stopband edges and also the infinite-loss frequencies.

(*b*) Determine the effective selectivity factor[3] for the digital filter designed.

(*c*) Find the minimum value of ω_s if the passband and stopband edges in the digital filter are to be within ±1 percent of the corresponding values in the analog filter.

11.24. Redesign the highpass filter in Prob. 11.16, part (*a*), using the bilinear transformation.

11.25. An analog elliptic lowpass filter has the following specifications:

- Passband edge $\omega_p = 0.6325$ rad/s
- Stopband edge $\omega_a = 1.5811$ rad/s
- Maximum passband loss = 0.5 dB
- Minimum stopband loss = 37.7 dB

The filter is characterized by the transfer function

$$H(s) = \frac{0.05917(s^2 + 3.2634)}{(s + 0.4209)(s^2 + 0.3618s + 0.4587)}$$

(*a*) Design a corresponding lowpass digital filter using the bilinear-transformation method assuming a sampling frequency of 2π rad/s.

(*b*) Find the passband and stopband edges of the digital filter.

(*c*) Find the maximum passband loss and minimum stopband loss of the digital filter.

11.26. (*a*) Obtain a continuous-time fourth-order highpass transfer function with a 3-dB cutoff frequency at 1 rad/s using the Butterworth approximation.

(*b*) Obtain a corresponding discrete-time transfer function using the bilinear transformation. Assume a sampling frequency of 10 rad/s.

(*c*) Determine the exact 3-dB cutoff frequency of the digital filter.

11.27. Derive the lowpass-to-highpass transformation of Table 11.7.

11.28. Derive the lowpass-to-bandpass transformation of Table 11.7.

11.29. $H_D(z)$ represents a lowpass filter with a passband edge Ω_p. Show that $H_D(-z)$ represents a highpass filter with a passband edge $\Omega_p - \omega_s/2$.

11.30. The transfer function

$$H(z) = H_0 \prod_{j=1}^{2} \frac{a_{0j} + a_{1j}z + a_{0j}z^2}{b_{0j} + b_{1j}z + z^2}$$

[3] *This is the ratio of the actual passband edge to the actual stopband edge.*

where a_{ij} and b_{ij} are given in Table P11.30, represents a lowpass digital filter with a passband edge of 1 rad/s if $\omega_s = 2\pi$ rad/s. By using the lowpass-to-highpass transformation in Table 11.7, design a highpass filter with a passband edge of 2 rad/s if $\omega_s = 2\pi$ rad/s.

Table P11.30

j	a_{0j}	a_{1j}	b_{0j}	b_{1j}
1	$1.722415E-1$	$3.444829E-1$	$4.928309E-1$	-1.263032
2	$1.727860E-1$	$3.455720E-1$	$7.892595E-1$	$-9.753164E-1$
$H_0 = 3.500865E-1$				

11.31. By using the transfer function of Prob. 11.30 obtain a lowpass cascade canonic structure whose passband edge can be varied by varying a single parameter.

This page intentionally left blank

CHAPTER 12

RECURSIVE (IIR) FILTERS SATISFYING PRESCRIBED SPECIFICATIONS

12.1 INTRODUCTION

The previous chapter has shown that given an analog filter, a corresponding digital filter can be readily obtained by using the bilinear transformation. This design method preserves the maxima and minima of the amplitude response and, as a consequence, passbands and stopbands in the analog filter translate into corresponding passbands and stopbands in the digital filter; furthermore, the passband ripple and minimum stopband attenuation in the analog filter are preserved in the digital filter, and the latter filter is stable if the former is stable. Owing to these important advantages, the bilinear transformation method is one of the most important methods for the design of digital filters, if not the most important. As was demonstrated in Chap. 11, the main problem with the method is the so-called *warping effect* which introduces frequency-scale distortion. If $\omega_1, \omega_2, \ldots, \omega_i, \ldots$ are the passband and stopband edges in the analog filter, then, according to Eq. (11.23), the corresponding passband and stopband edges in the derived digital filter are given by

$$\Omega_i = \frac{2}{T} \tan^{-1} \frac{\omega_i T}{2} \qquad i = 1, 2, \ldots$$

Consequently, if prescribed passband and stopband edges $\tilde{\Omega}_1,\ \tilde{\Omega}_2,\ \ldots,\ \tilde{\Omega}_i,\ \ldots$ are to be achieved, the analog filter must be prewarped before the application of the bilinear transformation to ensure that its band edges are given by

$$\omega_i = \frac{2}{T}\tan\frac{\tilde{\Omega}_i T}{2}$$

If this could be accomplished, then the band edges of the digital filter would assume their prescribed values Ω_i since

$$\begin{aligned}\Omega_i &= \frac{2}{T}\tan^{-1}\frac{\omega_i T}{2}\\ &= \frac{2}{T}\tan^{-1}\left(\frac{T}{2}\cdot\frac{2}{T}\tan\frac{\tilde{\Omega}_i T}{2}\right)\\ &= \tilde{\Omega}_i \qquad \text{for } i = 1,\ 2,\ \ldots\end{aligned}$$

The design of lowpass, highpass, bandpass, and bandstop filters is usually accomplished in two steps. First a normalized continuous-time lowpass transfer function is transformed into a denormalized lowpass, highpass, bandpass, or bandstop transfer function employing one of the standard analog-filter transformations described in Sec. 10.8. Then the bilinear transformation is applied. Prewarping can be effected by choosing the parameters in the analog-filter transformations appropriately.

This chapter considers the details of the above design procedure. Formulas are derived for the parameters of the analog-filter transformations for Butterworth, Chebyshev, inverse-Chebyshev, and elliptic filters, which simplify the design of digital filters satisfying prescribed specifications [1].

12.2 DESIGN PROCEDURE

Consider a normalized analog lowpass filter characterized by $H_N(s)$ with a loss

$$A_N(\omega) = 20\log\frac{1}{|H_N(j\omega)|}$$

and assume that

$$0 \le A_N(\omega) \le A_p \qquad \text{for } 0 \le |\omega| \le \omega_p$$

$$A_N(\omega) \ge A_a \qquad \text{for } \omega_a \le |\omega| \le \infty$$

as illustrated in Fig. 12.1. A corresponding denormalized lowpass (LP), highpass (HP), bandpass (BP), or bandstop (BS) digital filter with the same passband ripple and the same minimum stopband loss can be derived by using the following steps:

1. Form

$$H_X(\bar{s}) = H_N(s)\Big|_{s=f_X(\bar{s})} \tag{12.1}$$

where $f_X(\bar{s})$ is one of the transformations given in Table 12.1 (see Sec. 10.8).

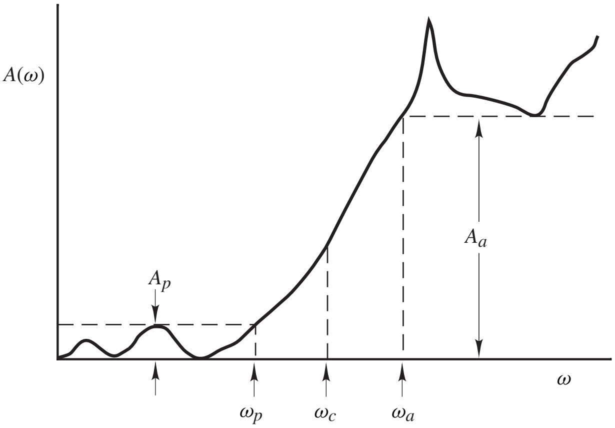


Figure 12.1 Loss characteristic of normalized LP analog filter.

2. Apply the bilinear transformation to $H_X(\bar{s})$, that is,

$$H_D(z) = H_X(\bar{s})\Big|_{\bar{s}=\frac{2}{T}\left(\frac{z-1}{z+1}\right)} \tag{12.2}$$

If the derived filter is to have prescribed passband and stopband edges, the parameters λ, ω_0, and B in Table 12.1 and the order of $H_N(s)$ must be chosen appropriately. Formulas for these parameters for the various standard types of analog-filter approximations are derived in the following section.

12.3 DESIGN FORMULAS

12.3.1 Lowpass and Highpass Filters

Consider the *lowpass-filter* specifications of Fig. 12.2, where $\tilde{\Omega}_p$ and $\tilde{\Omega}_a$ are the desired passband and stopband edges, and assume that the above design procedure yields a transfer function $H_D(z)$ such that

$$A_D(\Omega) = 20 \log \frac{1}{|H_D(e^{j\Omega T})|}$$

Table 12.1 Standard forms of $f_X(\bar{s})$

X	$f_X(\bar{s})$
LP	$\lambda\bar{s}$
HP	$\dfrac{\lambda}{\bar{s}}$
BP	$\dfrac{1}{B}\left(\bar{s}+\dfrac{\omega_0^2}{\bar{s}}\right)$
BS	$\dfrac{B\bar{s}}{\bar{s}^2+\omega_0^2}$

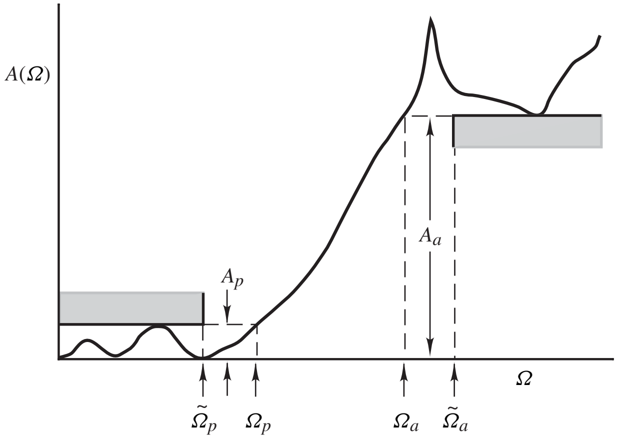


Figure 12.2 Loss characteristic of LP digital filter.

where
$$0 \le A_D(\Omega) \le A_p \qquad \text{for } 0 \le |\Omega| \le \Omega_p$$
$$A_D(\Omega) \ge A_a \qquad \text{for } \Omega_a \le |\Omega| \le \omega_s/2$$

as shown in Fig. 12.2.

If we let $s = j\omega$ and $\bar{s} = j\bar{\omega}$ in Eq. (12.1), then from Table 12.1

$$|H_{\text{LP}}(j\bar{\omega})| = |H_N(j\omega)|$$

provided that
$$\omega = \lambda\bar{\omega}$$

Hence
$$\omega_p = \lambda\bar{\omega}_p \tag{12.3}$$
$$\omega_a = \lambda\bar{\omega}_a \tag{12.4}$$

where $\bar{\omega}_p$ and $\bar{\omega}_a$ denote the passband and stopband edges, respectively, in $H_{\text{LP}}(\bar{s})$. If we now let $z = e^{j\Omega T}$ and $\bar{s} = j\bar{\omega}$ in Eq. (12.2), we get

$$|H_D(e^{j\Omega T})| = |H_{\text{LP}}(j\bar{\omega})|$$

provided that
$$\bar{\omega} = \frac{2}{T}\tan\frac{\Omega T}{2}$$

Thus
$$\bar{\omega}_p = \frac{2}{T}\tan\frac{\Omega_p T}{2} \tag{12.5}$$
$$\bar{\omega}_a = \frac{2}{T}\tan\frac{\Omega_a T}{2} \tag{12.6}$$

and from Eqs. (12.3)–(12.6), we obtain

$$\omega_p = \frac{2}{T}\lambda \tan \frac{\Omega_p T}{2} \tag{12.7}$$

$$\omega_a = \frac{2}{T}\lambda \tan \frac{\Omega_a T}{2} \tag{12.8}$$

From Eqs. (12.7) and (12.8), parameter λ of the lowpass-to-lowpass (LP-to-LP) transformation and the selectivity of the normalized analog LP filter can be deduced as

$$\lambda = \frac{T\omega_p}{2\tan(\Omega_p T/2)} \tag{12.9}$$

and

$$\frac{\omega_p}{\omega_a} = \frac{\tan(\Omega_p T/2)}{\tan(\Omega_a T/2)} \tag{12.10}$$

respectively.

The digital LP filter will satisfy the prescribed specifications if we force the *actual* passband and stopband edges Ω_p and Ω_a to be equal to the *prescribed* passband and stopband edges $\tilde{\Omega}_p$ and $\tilde{\Omega}_a$, respectively, but, in practice, it is usually difficult to fix both the passband and stopband edges at the prescribed values. A feasible alternative is to fix the passband edge at the prescribed value and to make the stopband edge equal to or less than the prescribed stopband edge. This can be accomplished by assigning

$$\Omega_p = \tilde{\Omega}_p \qquad \text{and} \qquad \Omega_a \le \tilde{\Omega}_a$$

In this way, the loss at $\Omega = \tilde{\Omega}_p$ will be exactly equal to the minimum value specified and the loss at $\Omega = \tilde{\Omega}_a$ will be equal to or exceed the minimum value specified as can be seen in Fig. 12.2.

With the above assignment, Eq. (12.9) gives the required value of the LP-to-LP transformation as

$$\lambda = \frac{T\omega_p}{2\tan(\tilde{\Omega}_p T/2)}$$

On the other hand, Eq. (12.10) gives

$$\frac{\omega_p}{\omega_a} = \frac{\tan(\Omega_p T/2)}{\tan(\Omega_a T/2)} \ge \frac{\tan(\tilde{\Omega}_p T/2)}{\tan(\tilde{\Omega}_a T/2)} \tag{12.11}$$

or

$$\frac{\omega_p}{\omega_a} \ge K_0 \quad \text{where} \quad K_0 = \frac{\tan(\tilde{\Omega}_p T/2)}{\tan(\tilde{\Omega}_a T/2)}$$

The inequality in Eq. (12.11) can be justified by noting that Ω_a and $\tilde{\Omega}_a$ are both less than the Nyquist frequency, $\omega_s/2$, and hence the values of $\Omega_a T/2$ and $\tilde{\Omega}_a T/2$ are in the range 0 to $\pi/2$. Since $\tan(\Omega T/2)$ is a monotonic increasing function of Ω over the range $0 < \Omega T/2 < \pi/2$ or

Table 12.2 Lowpass and highpass filters

LP	$\dfrac{\omega_p}{\omega_a} \geq K_0$
	$\lambda = \dfrac{\omega_p T}{2\tan(\tilde{\Omega}_p T/2)}$
HP	$\dfrac{\omega_p}{\omega_a} \geq \dfrac{1}{K_0}$
	$\lambda = \dfrac{2\omega_p \tan(\tilde{\Omega}_p T/2)}{T}$
where	$K_0 = \dfrac{\tan(\tilde{\Omega}_p T/2)}{\tan(\tilde{\Omega}_a T/2)}$

$0 < \Omega < \omega_s/2$ and $\tilde{\Omega}_a \geq \Omega_a$, according to the above assignment, it follows that $\tan(\tilde{\Omega}_a T/2) \geq \tan(\Omega_a T/2)$. This inequality states, in effect, that the selectivity of the normalized analog LP filter, ω_p/ω_a, must be greater than or equal to constant K_0, and this constraint imposes a lower limit on the order of $H_N(s)$, as will be shown later.

The preceding approach can be readily extended to *highpass filters*. The formulas for λ and ω_p/ω_a for LP and HP filters are summarized in Table 12.2.

12.3.2 Bandpass and Bandstop Filters

Now consider the *bandpass-filter* specifications of Fig. 12.3, where $\tilde{\Omega}_{p1}$, $\tilde{\Omega}_{p2}$ and $\tilde{\Omega}_{a1}$, $\tilde{\Omega}_{a2}$ represent the desired passband and stopband edges, respectively, and assume that the derived filter satisfies the

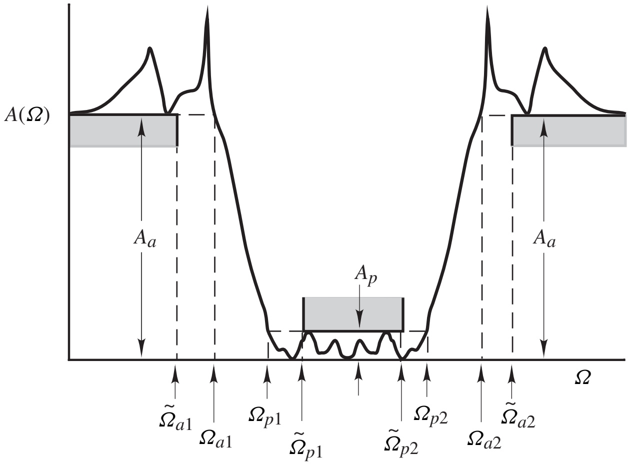


Figure 12.3 Loss characteristic of bandpass digital filter.

conditions

$$0 \le A_D(\Omega) \le A_p \qquad \text{for } \Omega_{p1} \le |\Omega| \le \Omega_{p2}$$
$$A_D(\Omega) \ge A_a \qquad \text{for } 0 \le |\Omega| \le \Omega_{a1} \qquad \text{and} \qquad \Omega_{a2} \le |\Omega| \le \omega_s/2$$

as shown in Fig. 12.3.

From Eq. (12.1) and Table 12.1

$$|H_{\text{BP}}(j\bar{\omega})| = |H_N(j\omega)|$$

provided that

$$\omega = \frac{1}{B}\left(\bar{\omega} - \frac{\omega_0^2}{\bar{\omega}}\right)$$

and by solving for $\bar{\omega}$, we obtain

$$\bar{\omega} = \frac{\omega B}{2} \pm \sqrt{\omega_0^2 + \left(\frac{\omega B}{2}\right)^2} \tag{12.12}$$

With $\qquad \omega = \pm\omega_p \qquad \text{or} \qquad \pm\,\omega_a$

the positive passband and stopband edges in $H_{\text{BP}}(\bar{s})$ can be deduced from Eq. (12.12) as

$$\bar{\omega}_{p1}, \bar{\omega}_{p2} = \mp\frac{\omega_p B}{2} + \sqrt{\omega_0^2 + \omega_p^2\left(\frac{B}{2}\right)^2}$$

and

$$\bar{\omega}_{a1}, \bar{\omega}_{a2} = \mp\frac{\omega_a B}{2} + \sqrt{\omega_0^2 + \omega_a^2\left(\frac{B}{2}\right)^2}$$

respectively. As can be readily verified

$$\bar{\omega}_{p1}\bar{\omega}_{p2} = \omega_0^2 \tag{12.13}$$
$$\bar{\omega}_{a1}\bar{\omega}_{a2} = \omega_0^2 \tag{12.14}$$
$$\bar{\omega}_{p2} - \bar{\omega}_{p1} = \omega_p B \tag{12.15}$$
$$\bar{\omega}_{a2} - \bar{\omega}_{a1} = \omega_a B \tag{12.16}$$

From Eq. (12.2)

$$|H_D(e^{j\Omega T})| = |H_{\text{BP}}(j\bar{\omega})|$$

if

$$\bar{\omega} = \frac{2}{T}\tan\frac{\Omega T}{2}$$

and hence

$$\bar{\omega}_{p1} = \frac{2}{T} \tan \frac{\Omega_{p1} T}{2} \tag{12.17}$$

$$\bar{\omega}_{p2} = \frac{2}{T} \tan \frac{\Omega_{p2} T}{2} \tag{12.18}$$

$$\bar{\omega}_{a1} = \frac{2}{T} \tan \frac{\Omega_{a1} T}{2} \tag{12.19}$$

$$\bar{\omega}_{a2} = \frac{2}{T} \tan \frac{\Omega_{a2} T}{2} \tag{12.20}$$

We can now assign

$$\Omega_{p1} = \tilde{\Omega}_{p1} \qquad \text{and} \qquad \Omega_{p2} = \tilde{\Omega}_{p2} \tag{12.21}$$

From Eqs. (12.15), (12.17), and (12.18), we obtain

$$B = \frac{2K_A}{T\omega_p} \tag{12.22}$$

where
$$K_A = \tan \frac{\tilde{\Omega}_{p2} T}{2} - \tan \frac{\tilde{\Omega}_{p1} T}{2} \tag{12.23}$$

Similarly, from Eqs. (12.13), (12.17), and (12.18)

$$\omega_0 = \frac{2\sqrt{K_B}}{T} \tag{12.24}$$

where
$$K_B = \tan \frac{\tilde{\Omega}_{p1} T}{2} \tan \frac{\tilde{\Omega}_{p2} T}{2} \tag{12.25}$$

With the passband edge of the normalized LP filter, ω_p, the sampling period $T (= 2\pi/\omega_s)$, and the specified passband edges for the digital filter, $\tilde{\Omega}_{p1}$ and $\tilde{\Omega}_{p2}$, known, parameters B and ω_0 of the LP-to-BP transformation can be determined.

In order to determine the minimum order of the normalized LP filter that would satisfy the specifications, we need to derive an expression for the selectivity of the LP filter as was done in the case of lowpass filters in the previous section. From Eqs. (12.13) and (12.14), we have

$$\bar{\omega}_{a1}\bar{\omega}_{a2} = \bar{\omega}_{p1}\bar{\omega}_{p2}$$

and from Eqs. (12.17)–(12.20)

$$\tan \frac{\Omega_{a1} T}{2} \tan \frac{\Omega_{a2} T}{2} = \tan \frac{\Omega_{p1} T}{2} \tan \frac{\Omega_{p2} T}{2}$$

but since the passband edges of the BP digital filter, Ω_{p1} and Ω_{p2}, have already been fixed (see Eq. (12.21)), the right-hand side of the above equation has also been fixed and, therefore, it is not in general possible to also fix the stopband edges, Ω_{a1} and Ω_{a2}, at the prescribed values. The alternative is to find suitable values for Ω_{a1} and Ω_{a2} such that

$$\Omega_{a1} \geq \tilde{\Omega}_{a1} \qquad \text{and} \qquad \Omega_{a2} \leq \tilde{\Omega}_{a2}$$

without violating Eqs. (12.14) and (12.16). In this way, the loss at $\Omega = \tilde{\Omega}_{a1}$ and $\tilde{\Omega}_{a2}$ will be equal to or exceed the minimum specified values as can be seen in Fig. 12.3.

If

$$\Omega_{a1} \geq \tilde{\Omega}_{a1}$$

then Eq. (12.19) gives

$$\bar{\omega}_{a1} = \frac{2}{T} \tan \frac{\Omega_{a1} T}{2} \geq \frac{2}{T} \tan \frac{\tilde{\Omega}_{a1} T}{2} \tag{12.26}$$

since $\tan(\Omega T/2)$ is a monotonic increasing function over the range 0 to $\omega_s/2$, as was shown in the case of LP filters. Let us assume that Ω_{a2} is chosen such that Eqs. (12.14) and (12.16) are satisfied. If we eliminate $\bar{\omega}_{a2}$ in Eq. (12.16) using Eq. (12.14) and then eliminate $\bar{\omega}_{a1}$ using Eq. (12.19), we get

$$\begin{aligned} \omega_a &= \frac{\omega_0^2 - \bar{\omega}_{a1}^2}{B\bar{\omega}_{a1}} \\ &= \frac{\omega_0^2 - (2/T)^2 \tan^2(\Omega_{a1} T/2)}{(2B/T)\tan(\Omega_{a1} T/2)} \end{aligned} \tag{12.27}$$

Therefore, from Eqs. (12.26) and (12.27), we have

$$\omega_a \leq \frac{\omega_0^2 - (2/T)^2 \tan^2(\tilde{\Omega}_{a1} T/2)}{(2B/T)\tan(\tilde{\Omega}_{a1} T/2)}$$

and on eliminating ω_0 and B using Eqs. (12.24) and (12.22), we deduce

$$\frac{\omega_p}{\omega_a} \geq K_1$$

where

$$K_1 = \frac{K_A \tan(\tilde{\Omega}_{a1} T/2)}{K_B - \tan^2(\tilde{\Omega}_{a1} T/2)} \tag{12.28}$$

and constants K_A and K_B are given by Eqs. (12.23) and (12.25), respectively. In other words, in order to satisfy the required specifications at the lower stopband edge, Ω_{a1}, the selectivity of the normalized analog LP filter must be equal to or exceed constant K_1.

On the other hand, if

$$\Omega_{a2} \leq \tilde{\Omega}_{a2}$$

then from Eq. (12.20)

$$\bar{\omega}_{a2} = \frac{2}{T}\tan\frac{\Omega_{a2}T}{2} \le \frac{2}{T}\tan\frac{\tilde{\Omega}_{a2}T}{2} \tag{12.29}$$

since $\Omega_{a2} \le \tilde{\Omega}_{a2}$ and $\tan \Omega T/2$ is a monotonic increasing function over the range $0 < \Omega < \omega_s/2$. Let us assume that Ω_{a1} is chosen such that Eqs. (12.14) and (12.16) are satisfied. If we eliminate $\bar{\omega}_{a1}$ in Eq. (12.16) using Eq. (12.14) and then eliminate $\bar{\omega}_{a2}$ using Eq. (12.20), we get

$$\begin{aligned}\omega_a &= \frac{\bar{\omega}_{a2}^2 - \omega_0^2}{B\bar{\omega}_{a2}}\\ &= \frac{(2/T)^2\tan^2(\Omega_{a2}T/2) - \omega_0^2}{(2B/T)\tan(\Omega_{a2}T/2)}\end{aligned} \tag{12.30}$$

Therefore, from Eqs. (12.29) and (12.30), we have

$$\omega_a \le \frac{(2/T)^2\tan^2(\tilde{\Omega}_{a2}T/2) - \omega_0^2}{(2B/T)\tan(\tilde{\Omega}_{a2}T/2)}$$

and on eliminating ω_0 and B, we conclude that

$$\frac{\omega_p}{\omega_p} \ge K_2$$

where

$$K_2 = \frac{K_A\tan(\tilde{\Omega}_{a2}T/2)}{\tan^2(\tilde{\Omega}_{a2}T/2) - K_B} \tag{12.31}$$

That is, in order to satisfy the required specifications at the upper stopband edge, Ω_{a2}, the selectivity of the normalized analog LP filter must be equal to or exceed constant K_2.

Summarizing, if $\Omega_{a1} \ge \tilde{\Omega}_{a1}$ and Ω_{a2} is assumed to satisfy the required constraints, then $\omega_p/\omega_a \ge K_1$; and if $\Omega_{a2} \le \tilde{\Omega}_{a2}$ and Ω_{a1} is assumed to satisfy the required constraints, then $\omega_p/\omega_a \ge K_2$. These relations also hold in the reverse order, that is, if $\omega_p/\omega_a \ge K_1$ then $\Omega_{a1} \ge \Omega_{a1}$ and if $\omega_p/\omega_a \ge K_2$ then $\Omega_{a2} \le \tilde{\Omega}_{a2}$. Therefore, if we ensure that

$$\frac{\omega_p}{\omega_a} \ge K \qquad \text{where } K = \max(K_1, K_2)$$

then $$\Omega_{a1} \ge \tilde{\Omega}_{a1} \qquad \text{and} \qquad \Omega_{a2} \le \tilde{\Omega}_{a2}$$

as required. In words, in order to satisfy the specifications at the lower stopband edge, we need a selectivity of at least K_1 for the normalized LP filter and to satisfy the specifications at the upper stopband edge, we need a selectivity of at least K_2. Therefore, choosing the larger value of the selectivity would ensure that the specifications are satisfied at both stopband edges.

The appropriate value for K is easily deduced from Eqs. (12.28) and (12.31) as

$$K = \begin{cases} K_1 & \text{if } K_C \ge K_B\\ K_2 & \text{if } K_C < K_B\end{cases}$$

Table 12.3 Bandpass and bandstop filters

BP	$\omega_0 = \dfrac{2\sqrt{K_B}}{T}$	
	$\dfrac{\omega_p}{\omega_a} \geq \begin{cases} K_1 & \text{if } K_C \geq K_B \\ K_2 & \text{if } K_C < K_B \end{cases}$	
	$B = \dfrac{2K_A}{T\omega_p}$	
BS	$\omega_0 = \dfrac{2\sqrt{K_B}}{T}$	
	$\dfrac{\omega_p}{\omega_a} \geq \begin{cases} \dfrac{1}{K_2} & \text{if } K_C \geq K_B \\ \dfrac{1}{K_1} & \text{if } K_C < K_B \end{cases}$	
	$B = \dfrac{2K_A\omega_p}{T}$	
where	$K_A = \tan\dfrac{\tilde{\Omega}_{p2}T}{2} - \tan\dfrac{\tilde{\Omega}_{p1}T}{2}$	$K_B = \tan\dfrac{\tilde{\Omega}_{p1}T}{2}\tan\dfrac{\tilde{\Omega}_{p2}T}{2}$
	$K_C = \tan\dfrac{\tilde{\Omega}_{a1}T}{2}\tan\dfrac{\tilde{\Omega}_{a2}T}{2}$	$K_1 = \dfrac{K_A\tan(\tilde{\Omega}_{a1}T/2)}{K_B - \tan^2(\tilde{\Omega}_{a1}T/2)}$
	$K_2 = \dfrac{K_A\tan(\tilde{\Omega}_{a2}T/2)}{\tan^2(\tilde{\Omega}_{a2}T/2) - K_B}$	

where

$$K_C = \tan\frac{\tilde{\Omega}_{a1}T}{2}\tan\frac{\tilde{\Omega}_{a2}T}{2}$$

The same approach can also be applied to deduce appropriate formulas for *bandstop filters*. The required design formulas for BP and BS filters are summarized in Table 12.3.

The formulas derived so far are very general and apply to any normalized LP filter that has a loss characteristic of a form illustrated in Fig. 12.1. Specific formulas for Butterworth, Chebyshev, inverse-Chebyshev, and elliptic filters can fairly easily be obtained as will now be demonstrated.

12.3.3 Butterworth Filters

The loss in a normalized *Butterworth* filter is given by

$$A_N(\omega) = 10\log(1 + \omega^{2n})$$

(see Sec. 10.3) where n is the order of the transfer function. Evaluating $A_N(\omega)$ for $\omega = \omega_p$ and ω_a, we get

$$A_N(\omega_p) = A_p = 10\log\bigl(1+\omega_p^{2n}\bigr) \qquad \text{and} \qquad A_N(\omega_a) = A_a = 10\log\bigl(1+\omega_a^{2n}\bigr)$$

Solving for ω_p and ω_a, we obtain

$$\omega_p = (10^{0.1A_p} - 1)^{1/2n} \qquad \text{and} \qquad \omega_a = (10^{0.1A_a} - 1)^{1/2n} \tag{12.32}$$

Hence from Eq. (12.32) and Tables 12.2 and 12.3, we have

$$\frac{\omega_p}{\omega_a} = \left(\frac{10^{0.1A_p} - 1}{10^{0.1A_a} - 1}\right)^{1/2n} \geq K$$

where K depends on the type of filter and is given in terms of constants K_0, K_1, and K_2 in Table 12.4. Solving for n, we obtain

$$n \geq \frac{\log D}{2\log(1/K)}$$

where

$$D = \frac{10^{0.1A_a} - 1}{10^{0.1A_p} - 1} \tag{12.33}$$

The smallest integer that would satisfy the above equation is the minimum filter order that would satisfy the required specifications.

Table 12.4 Butterworth filters

LP	$K = K_0$
HP	$K = \dfrac{1}{K_0}$
BP	$K = \begin{cases} K_1 & \text{if } K_C \geq K_B \\ K_2 & \text{if } K_C < K_B \end{cases}$
BS	$K = \begin{cases} \dfrac{1}{K_2} & \text{if } K_C \geq K_B \\ \dfrac{1}{K_1} & \text{if } K_C < K_B \end{cases}$
$n \geq \dfrac{\log D}{2\log(1/K)} \qquad D = \dfrac{10^{0.1A_a} - 1}{10^{0.1A_p} - 1}$	
$\omega_p = (10^{0.1A_p} - 1)^{1/2n}$	

12.3.4 Chebyshev Filters

In normalized *Chebyshev* filters, we have

$$A_N(\omega) = 10\log\,[1 + \varepsilon^2 T_n^2(\omega)]$$

where $$T_n(\omega) = \cosh\,(n\cosh^{-1}\omega) \qquad \text{for } \omega_p \le \omega < \infty$$

with $$\varepsilon^2 = 10^{0.1A_p} - 1$$

and $$\omega_p = 1 \tag{12.34}$$

(see Sec. 10.4). For $\omega = \omega_a$

$$A_N(\omega_a) = A_a = 10\log\,\{1 + (10^{0.1A_p} - 1)[\cosh\,(n\cosh^{-1}\omega_a)]^2\}$$

or $$\omega_a = \cosh\left(\frac{1}{n}\cosh^{-1}\sqrt{D}\right) \tag{12.35}$$

Thus from Eqs. (12.34) and (12.35) and Tables 12.2 and 12.3, we obtain

$$\frac{\omega_p}{\omega_a} = \frac{1}{\cosh\left(\frac{1}{n}\cosh^{-1}\sqrt{D}\right)} \ge K$$

where K is given in terms of constants K_0, K_1, and K_2 in Table 12.5. Solving for n, the order of the normalized LP filter that would satisfy the specifications must satisfy the inequality

$$n \ge \frac{\cosh^{-1}\sqrt{D}}{\cosh^{-1}(1/K)}$$

Table 12.5 Chebyshev filters

LP	$K = K_0$
HP	$K = \dfrac{1}{K_0}$
BP	$K = \begin{cases} K_1 & \text{if } K_C \ge K_B \\ K_2 & \text{if } K_C < K_B \end{cases}$
BS	$K = \begin{cases} \dfrac{1}{K_2} & \text{if } K_C \ge K_B \\ \dfrac{1}{K_1} & \text{if } K_C < K_B \end{cases}$
$n \ge \dfrac{\cosh^{-1}\sqrt{D}}{\cosh^{-1}(1/K)}$	$D = \dfrac{10^{0.1A_a} - 1}{10^{0.1A_p} - 1}$
$\omega_p = 1$	

where $\cosh^{-1} x$ can be evaluated using the identity

$$\cosh^{-1} x = \ln(x + \sqrt{x^2 - 1})$$

12.3.5 Inverse-Chebyshev Filters

In normalized *inverse-Chebyshev* filters, the loss is given by

$$A_N(\omega) = 10 \log \left[1 + \frac{1}{\delta^2 T_n^2(1/\omega)} \right]$$

$$\text{where} \qquad T_n(1/\omega) = \cosh [n \cosh^{-1}(1/\omega)] \qquad \text{for } 0 < \omega < \omega_a$$

$$\text{with} \qquad \delta^2 = \frac{1}{10^{0.1A_a} - 1}$$

$$\text{and} \qquad \omega_a = 1 \tag{12.36}$$

(see Sec. 10.5). For $\omega = \omega_p$, we can write

$$A_N(\omega_p) = 10 \log \left[1 + \frac{1}{\delta^2 T_n^2(1/\omega_p)} \right]$$

Now solving for ω_p, we obtain

$$\omega_p = \frac{1}{\cosh \left(\frac{1}{n} \cosh^{-1} \sqrt{D} \right)} \tag{12.37}$$

and from Eqs. (12.36) and (12.37)

$$\frac{\omega_p}{\omega_a} = \frac{1}{\cosh \left(\frac{1}{n} \cosh^{-1} \sqrt{D} \right)} \ge K$$

Therefore, the order of the normalized LP filter must satisfy the inequality

$$n \ge \frac{\cosh^{-1} \sqrt{D}}{\cosh^{-1}(1/K)}$$

where K in terms of constants K_0, K_1, and K_2 is given in Table 12.5. Evidently, the formula for n is the same as for Chebyshev filters and thus inverse-Chebyshev filters can be designed in the same way as Chebyshev ones using the parameters in Table 12.5 except that the value for ω_p given by Eq. (12.37) should be used instead of unity.

12.3.6 Elliptic Filters

In *elliptic filters*, the selectivity factor is defined as

$$k = \frac{\omega_p}{\omega_a}$$

Table 12.6 Elliptic filters

	k		ω_p
LP	K_0		$\sqrt{K_0}$
HP	$\dfrac{1}{K_0}$		$\dfrac{1}{\sqrt{K_0}}$
BP	K_1	if $K_C \geq K_B$	$\sqrt{K_1}$
	K_2	if $K_C < K_B$	$\sqrt{K_2}$
BS	$\dfrac{1}{K_2}$	if $K_C \geq K_B$	$\dfrac{1}{\sqrt{K_2}}$
	$\dfrac{1}{K_1}$	if $K_C < K_B$	$\dfrac{1}{\sqrt{K_1}}$
$n \geq \dfrac{\log 16D}{\log(1/q)}$		$D = \dfrac{10^{0.1A_a} - 1}{10^{0.1A_p} - 1}$	

(see Sec. 10.6) and from Tables 12.2 and 12.3, we have

$$k \geq K$$

where $K = K_0,\ 1/K_0,\ \ldots$. Since any value in the range 0 to 1, except for unity, is a permissible value for k, we can assign the lowest acceptable value, namely,

$$k = K$$

as in Table 12.6, which will lead to the lowest order normalized LP analog filter.

With k chosen, the value of ω_p is also fixed, that is,

$$\omega_p = \sqrt{k}$$

Finally, with k, A_p, and A_a known the necessary value for n can be computed by using the formula in Table 12.6 (see Sec. 10.6.5).

12.4 DESIGN USING THE FORMULAS AND TABLES

The formulas and tables developed in the preceding section lead to the following simple design procedure:

1. Using the prescribed specifications, determine n and ω_p, and for elliptic filters also k, from Tables 12.4 to 12.6 (use Eq. (12.37) to calculate ω_p for inverse-Chebyshev filters).
2. Determine λ for LP and HP filters using Table 12.2 or B and ω_0 for BP and BS filters using Table 12.3.

3. Form the normalized transfer function (see Chap. 10).
4. Apply the analog-filter transformation in Eq. (12.1).
5. Apply the bilinear transformation in Eq. (12.2).

The procedure yields LP and HP filters that satisfy the specifications exactly at the passband edge and oversatisfy the specifications at the stopband edge. In the case of bandpass and bandstop filters, the specifications are satisfied exactly at the two passband edges and are oversatisfied at the stopband edges.

Example 12.1 Design a highpass filter that would satisfy the following specifications:

$$A_p = 1 \text{ dB} \qquad A_a = 45 \text{ dB} \qquad \tilde{\Omega}_p = 3.5 \text{ rad/s}$$
$$\tilde{\Omega}_a = 1.5 \text{ rad/s} \qquad \omega_s = 10 \text{ rad/s}$$

Use a Butterworth, a Chebyshev, and then an elliptic approximation.

■ **Solution**

Butterworth filter: The sampling period is

$$T = \frac{2\pi}{\omega_s} = \frac{2\pi}{10}$$

From Table 12.2

$$K_0 = \frac{\tan(3.5\pi/10)}{\tan(1.5\pi/10)} = 3.851840$$

Hence from Table 12.4

$$D = \frac{10^{4.5} - 1}{10^{0.1} - 1} = 1.221270 \times 10^5$$
$$n \geq \frac{\log D}{2 \log K_0} \approx 4.34 \rightarrow 5$$
$$\omega_p = (10^{0.1} - 1)^{0.1} = 8.736097 \times 10^{-1}$$

Now from Table 12.2

$$\lambda = \frac{2}{T}\omega_p \tan\frac{\tilde{\Omega}_p T}{2} = 5.457600$$

Chebyshev filter: From Table 12.5

$$n \geq \frac{\cosh^{-1}\sqrt{D}}{\cosh^{-1} K_0} = \frac{\ln\left(\sqrt{D} + \sqrt{D-1}\right)}{\ln\left(K_0 + \sqrt{K_0^2 - 1}\right)} \approx 3.24 \rightarrow 4$$

$$\omega_p = 1$$

Hence from Table 12.2

$$\lambda = 6.247183$$

Elliptic filter: From Table 12.6

$$k = \frac{1}{K_0} = 2.596162 \times 10^{-1}$$

Now, from Eqs. (10.86)–(10.88), we obtain

$$k' = \sqrt{1 - k^2} = 9.657119 \times 10^{-1}$$

$$q_0 = \frac{1}{2}\left(\frac{1 - \sqrt{k'}}{1 + \sqrt{k'}}\right) = 4.361108 \times 10^{-3}$$

$$q = q_0 + 2q_0^5 + \cdots \approx q_0$$

Hence from Table 12.6

$$n \geq \frac{\log 16D}{\log (1/q)} \approx 2.67 \to 3$$

$$\omega_p = \sqrt{k} = 5.095255 \times 10^{-1}$$

and from Table 12.2

$$\lambda = 3.183099$$

At this point the normalized LP Butterworth, Chebyshev, and elliptic approximations can be obtained (see Chap. 10). The designs can be completed by applying the LP-to-HP transformation and after that the bilinear transformation to each of the three approximations.

The transfer functions $H_N(s)$, $H_{HP}(\bar{s})$, and $H_D(z)$ can be put in the form

$$H_0 \prod_{j=1}^{J} \frac{a_{0j} + a_{1j}w + a_{2j}w^2}{b_{0j} + b_{1j}w + b_{2j}w^2} \qquad \blacksquare$$

where H_0 is a multiplier constant and $w = s,\ \bar{s}$, or z. The coefficients of $H_N(s)$ can be computed as in Table 12.7, those of $H_{\mathrm{HP}}(\bar{s})$ as in Table 12.8, and those of $H_D(z)$ as in Table 12.9.

The loss characteristics of the three filters are plotted in Fig. 12.4. The actual minimum stopband loss for the Butterworth, Chebyshev, and elliptic filter are 52.70, 58.45, and 52.90 dB, respectively.

Table 12.7 Coefficients of $H_N(s)$ (Example 12.1)

	j	a_{0j}	a_{1j}	a_{2j}	b_{0j}	b_{1j}	b_{2j}
Butterworth	1	1	0	0	1	1.0	0
	2	1	0	0	1	1.618034	1
	3	1	0	0	1	$6.180340E-1$	1
	$H_0 = 1.0$						
Chebyshev	1	1	0	0	$2.793981E-1$	$6.737394E-1$	1
	2	1	0	0	$9.865049E-1$	$2.790720E-1$	1
	$H_0 = 2.456533E-1$						
Elliptic	1	1.0	0	0	$2.573050E-1$	1.0	0
	2	5.091668	0	1	$2.592344E-1$	$2.442048E-1$	1
	$H_0 = 1.310029E-2$						

Table 12.8 Coefficients of $H_{HP}(\bar{s})$ (Example 12.1)

	j	a_{0j}	a_{1j}	a_{2j}	b_{0j}	b_{1j}	b_{2j}
Butterworth	1	0	1	0	5.457600	1.0	0
	2	0	0	1	$2.978540E+1$	8.830582	1
	3	0	0	1	$2.978540E+1$	3.372982	1
	$H_0 = 1.0$						
Chebyshev	1	0	0	1	$1.396835E+2$	$1.506443E+1$	1
	2	0	0	1	$3.956118E+1$	1.767263	1
	$H_0 = 8.912509E-1$						
Elliptic	1	0.0	1	0	$1.237091E+1$	1.0	0
	2	1.989941	0	1	$3.908478E+1$	2.998552	1
	$H_0 = 1.0$						

Table 12.9 Coefficients of $H_D(z)$ (Example 12.1)

	j	a_{0j}	a_{1j}	a_{2j}	b_{0j}	b_{1j}	b_{2j}
Butterworth	1	−1	1	0	$2.632312E-1$	1.0	0
	2	1	−2	1	$1.735936E-1$	$5.778156E-1$	1
	3	1	−2	1	$5.760838E-1$	$7.759805E-1$	1
	$H_0 = 1.097518E-2$						
Chebyshev	1	1	−2	1	$5.150704E-1$	1.310140	1
	2	1	−2	1	$7.966193E-1$	1.063983	1
	$H_0 = 8.363241E-3$						
Elliptic	1	−1	1.0	0	$5.907039E-1$	1.0	0
	2	1	−1.343365	1	$6.751385E-1$	$9.854281E-1$	1
	$H_0 = 4.221731E-2$						

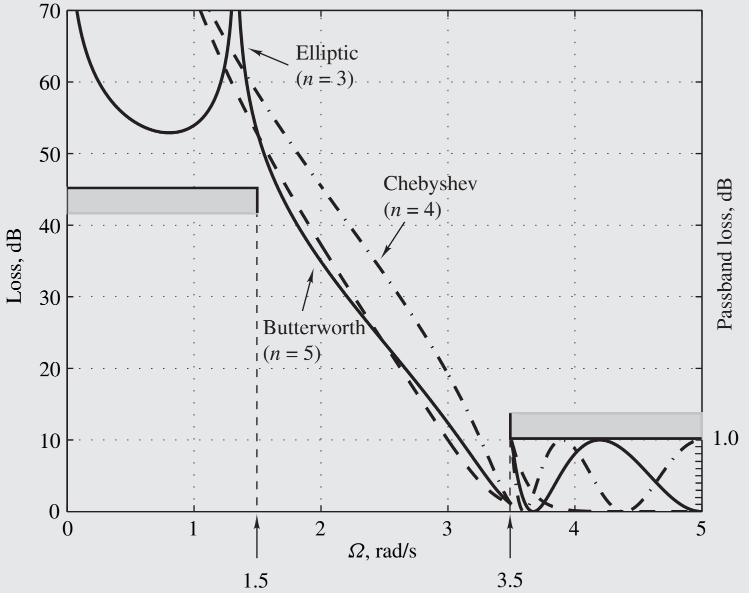


Figure 12.4 Loss characteristics of highpass filters (Example 12.1).

Example 12.2 Design an elliptic bandpass filter that would satisfy the following specifications:

$$A_p = 1 \text{ dB} \qquad A_a = 45 \text{ dB} \qquad \tilde{\Omega}_{p1} = 900 \text{ rad/s} \qquad \tilde{\Omega}_{p2} = 1100 \text{ rad/s}$$

$$\tilde{\Omega}_{a1} = 800 \text{ rad/s} \qquad \tilde{\Omega}_{a2} = 1200 \text{ rad/s} \qquad \omega_s = 6000 \text{ rad/s}$$

■ **Solution**

From Table 12.3

$$K_A = \tan\frac{1100\pi}{6000} - \tan\frac{900\pi}{6000} = 1.398821 \times 10^{-1}$$

$$K_B = \tan\frac{900\pi}{6000}\tan\frac{1100\pi}{6000} = 3.308897 \times 10^{-1}$$

$$K_C = \tan\frac{800\pi}{6000}\tan\frac{1200\pi}{6000} = 3.234776 \times 10^{-1}$$

Hence $K_C < K_B$ and from Table 12.6, we get

$$k = K_2 = \frac{K_A \tan(\tilde{\Omega}_{a2}T/2)}{\tan^2(\tilde{\Omega}_{a2}T/2) - K_B} = 5.159570 \times 10^{-1}$$

$$\omega_p = \sqrt{K_2} = 7.183016 \times 10^{-1}$$

D is the same as in Example 12.1, that is, $D = 1.221270 \times 10^5$, and from Eqs. (10.86)–(10.88), we have

$$k' = \sqrt{1-k^2} = 8.566144 \times 10^{-1}$$

$$q_0 = \frac{1}{2}\left(\frac{1-\sqrt{k'}}{1+\sqrt{k'}}\right) = 1.933628 \times 10^{-2}$$

$$q = q_0 + 2q_0^5 + \cdots \approx 1.933629 \times 10^{-2}$$

Hence $$n \geq \frac{\log 16D}{\log(1/q)} \approx 3.67 \rightarrow 4$$

Now, from Table 12.3

$$\omega_0 = \frac{2\sqrt{K_B}}{T} = 1.098609 \times 10^3 \qquad \text{and} \qquad B = \frac{2K_A}{T\omega_p} = 3.719263 \times 10^2$$

The parameters of the elliptic approximation and the LP-to-BP transformation are, at this point, available. Hence the elliptic approximation can be obtained (see Sec. 10.6.6).

Table 12.10 Coefficients of $H_D(z)$ (Example 12.2)

j	a_{0j}	a_{1j}	b_{0j}	b_{1j}
1	1.0	-1.602667	$9.268668E-1$	$-8.886598E-1$
2	1.0	$-7.283103E-3$	$9.306057E-1$	-1.046605
3	1.0	-1.331115	$9.738539E-1$	$-8.048914E-1$
4	1.0	$-5.885571E-1$	$9.767824E-1$	-1.160308
$H_0 = 3.444154E-3$				

Applying the LP-to-BP transformation followed by the bilinear transformation yields the transfer function of the required filter as

$$H_D(z) = H_0 \prod_{j=1}^{4} \frac{a_{0j} + a_{1j}z + z^2}{b_{0j} + b_{1j}z + z^2} \quad \blacksquare$$

The numerical values of the coefficients are given in Table 12.10. The loss characteristic of the filter is plotted in Fig. 12.5. The actual minimum stopband loss is 50.64 dB.

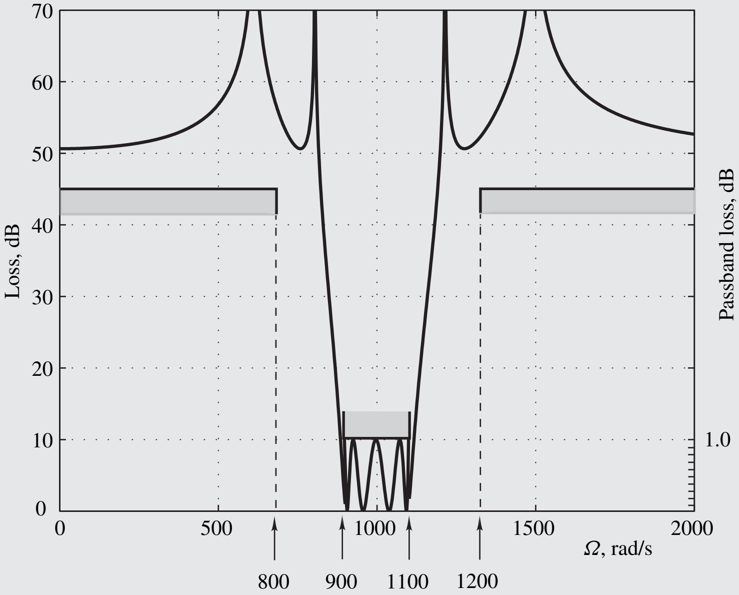


Figure 12.5 Loss characteristic of elliptic bandpass filter (Example 12.2).

Example 12.3 Design a Chebyshev bandstop filter that would satisfy the following specifications:

$$A_p = 0.5\text{ dB} \qquad A_a = 40\text{ dB} \qquad \tilde{\Omega}_{p1} = 350\text{ rad/s} \qquad \tilde{\Omega}_{p2} = 700\text{ rad/s}$$

$$\tilde{\Omega}_{a1} = 430\text{ rad/s} \qquad \tilde{\Omega}_{a2} = 600\text{ rad/s} \qquad \omega_s = 3000\text{ rad/s}$$

- **Solution**

The sampling period is given by

$$T = \frac{2\pi}{\omega_s} = \frac{2\pi}{3000}$$

Hence

$$\frac{\tilde{\Omega}_{p1}T}{2} = \frac{35\pi}{300} \qquad \frac{\tilde{\Omega}_{p2}T}{2} = \frac{70\pi}{300}$$

$$\frac{\tilde{\Omega}_{a1}T}{2} = \frac{43\pi}{300} \qquad \frac{\tilde{\Omega}_{a2}T}{2} = \frac{60\pi}{300}$$

From Table 12.3, we have

$$K_A = \tan\frac{70\pi}{300} - \tan\frac{35\pi}{300} = 5.165400 \times 10^{-1}$$

$$K_B = \tan\frac{35\pi}{300}\tan\frac{70\pi}{300} = 3.456327 \times 10^{-1}$$

$$K_C = \tan\frac{43\pi}{300}\tan\frac{60\pi}{300} = 3.512244 \times 10^{-1}$$

Since $K_C > K_B$, according to Table 12.5, we need to compute $K = 1/K_2$, that is,

$$K = \frac{1}{K_2} = \frac{\tan^2(\tilde{\Omega}_{a2}T/2) - K_B}{K_A\tan(\tilde{\Omega}_{a2}T/2)} = 4.855769 \times 10^{-1}$$

From Table 12.5,

$$D = \frac{10^{4.0} - 1}{10^{0.05} - 1} = 8.194662 \times 10^4$$

$$n = \frac{\cosh^{-1}\sqrt{D}}{\cosh^{-1} 1/K} = 4.70 \rightarrow 5$$

$$\omega_p = 1$$

Now from Table 12.3, the parameters of the LP-to-BS transformation can be obtained as

$$\omega_0 = \frac{2\sqrt{K_B}}{T} = 5.614083 \times 10^2$$

$$B = \frac{2K_A\omega_p}{T} = 4.932594 \times 10^2$$

Table 12.11 Coefficients of $H_D(z)$ (Example 12.3)

j	a_{0j}	a_{1j}	b_{0j}	b_{1j}
1	1.0	$-9.725792E-1$	$-2.887281E-2$	$-4.722491E-1$
2	1.0	$-9.725792E-1$	$6.230100E-1$	$5.028889E-2$
3	1.0	$-9.725792E-1$	$7.543570E-1$	-1.400163
4	1.0	$-9.725792E-1$	$9.168994E-1$	$-2.175109E-1$
5	1.0	$-9.725792E-1$	$9.428927E-1$	-1.435926

$H_0 = 2.225052\text{E}-1$

By obtaining the appropriate Chebyshev approximation (see Sec. 10.4.3) and then applying the LP-to-BS transformation followed by the bilinear transformation the transfer function is of the required digital filter can be obtained as

$$H_D(z) = H_0 \prod_{j=1}^{5} \frac{a_{0j} + a_{1j}z + z^2}{b_{0j} + b_{1j}z + z^2} \quad \blacksquare$$

where the coefficients a_{ij} and b_{ij} are given in Table 12.11. The loss characteristic of the filter is plotted in Fig. 12.6. The actual minimum stopband loss is 43.50 dB.

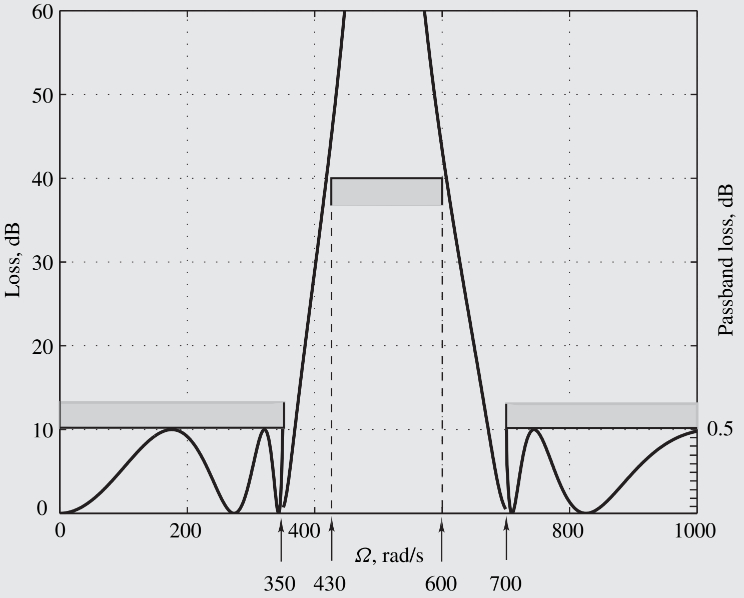


Figure 12.6 Loss characteristic of Chebyshev bandstop filter (Example 12.3).

12.5 CONSTANT GROUP DELAY

The phase response in filters designed by using the method described in this chapter is in general quite nonlinear because of two reasons. First, the Butterworth, Chebyshev, inverse-Chebyshev, and elliptic approximations are inherently nonlinear-phase approximations. Second, the warping effect tends to increase the nonlinearity of the phase response. As a consequence, the group delay tends to vary with frequency and the application of these filters tends to introduce delay distortion (see Sec. 5.7).

Constant group-delay filters can sometimes be designed by using constant-delay approximations such as the Bessel-Thomson approximation with design methods that preserve the linearity in the phase response of the analog filter, e.g., the invariant impulse-response method. However, a constant delay and prescribed loss specifications are usually difficult to achieve simultaneously, particularly if bandpass or bandstop high-selectivity filters are desired.

12.5.1 Delay Equalization

The design of constant-delay analog filters satisfying prescribed loss specifications is almost invariably accomplished in two steps. First a filter is designed satisfying the loss specifications ignoring the group delay. Then a delay equalizer is designed which can be used in cascade with the filter to compensate for variations in the group delay of the filter. The same technique can also be used in digital filters.

Let $H_F(z)$ and $H_E(z)$ be the transfer functions of the filter and equalizer, respectively. The group delays of the filter and equalizer are given by

$$\tau_F(\omega) = -\frac{d\theta_F(\omega)}{d\omega} \qquad \text{and} \qquad \tau_E(\omega) = -\frac{d\theta_E(\omega)}{d\omega}$$

respectively, where

$$\theta_F(\omega) = \arg H_F(e^{j\omega T}) \qquad \text{and} \qquad \theta_E(\omega) = \arg H_E(e^{j\omega T})$$

The overall transfer function of the filter-equalizer combination is

$$H_{FE}(z) = H_F(z)H_E(z)$$

Hence

$$|H_{FE}(e^{j\omega T})| = |H_F(e^{j\omega T})||H_E(e^{j\omega T})|$$

and

$$\theta_{FE}(\omega) = \theta_F(\omega) + \theta_E(\omega) \tag{12.38}$$

Now from Eq. (12.38), the overall group delay of the filter-equalizer combination can be obtained as

$$\tau_{FE}(\omega) = \tau_F(\omega) + \tau_E(\omega)$$

Therefore, a digital filter that satisfies prescribed loss specifications and has constant group delay with respect to some passband $\omega_{p1} \leq \omega \leq \omega_{p2}$ can be designed using the following steps:

1. Design a filter satisfying the loss specifications using the procedure in Sec. 12.4.
2. Design an equalizer with

$$|H_E(e^{j\omega T})| = 1 \qquad \text{for } 0 \leq \omega \leq \omega_s/2$$

and

$$\tau_E(\omega) = \tau - \tau_F(\omega) \qquad \text{for } \omega_{p1} \leq \omega \leq \omega_{p2} \tag{12.39}$$

where τ is a constant.

From step 2, $H_E(z)$ must be an *allpass* transfer function of the form

$$H_E(z) = \prod_{j=1}^{M} \frac{1 + c_{1j}z + c_{0j}z^2}{c_{0j} + c_{1j}z + z^2} \tag{12.40}$$

The equalizer can be designed by finding a set of values for c_{0j}, c_{1j}, τ, and M such that (*a*) Eq. (12.39) is satisfied to within a prescribed error in order to achieve approximately constant group delay with respect to the passband, and (*b*) the poles of $H_E(z)$ are inside the unit circle of the z plane to ensure that the equalizer is stable. Equalizers can be designed by using optimization methods as will be demonstrated in Sec. 16.8.

Note that delay equalization is unnecessary for stopbands since signals that pass through stopbands are normally deemed to be noise and delay distortion in noise is of no concert.

12.5.2 Zero-Phase Filters

In nonreal-time applications, the problem of delay distortion can be eliminated in a fairly simple manner by designing the filter as a cascade arrangement of two filters characterized by $H(z)$ and $H(z^{-1})$, as depicted in Fig. 12.7*a*. Since $H(e^{-j\omega T})$ is the complex conjugate of $H(e^{j\omega T})$, the frequency response of the cascade arrangement can be expressed as

$$H_0(e^{j\omega T}) = H(e^{j\omega T})H(e^{-j\omega T}) = |H(e^{j\omega T})|^2$$

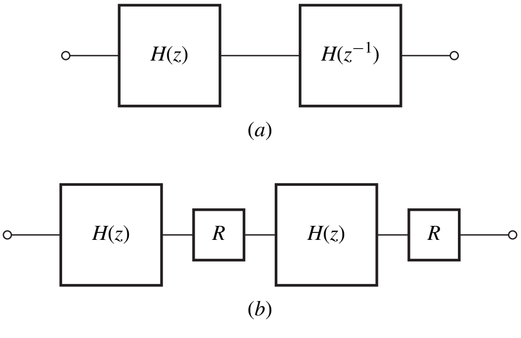


Figure 12.7 (*a*) Zero-phase filter, (*b*) implementation.

In other words, the frequency response of the arrangement is real and, as a result, the filter has *zero phase response* and, therefore, it would introduce zero delay. If a filter with passband ripple A_p and minimum stopband loss A_a is required, the design can be readily completed by obtaining a transfer function with passband ripple $A_p/2$ and minimum stopband loss $A_a/2$, since the two filters in Fig. 12.7a have identical amplitude responses.

If the impulse response of the first filter is $h(nT)$, then that of the second filter is $h(-nT)$, as can be readily demonstrated (see Prob. 12.20), and if the first filter is causal, the second one is noncausal. Hence the cascade of Fig. 12.7a can be implemented, as depicted in Fig. 12.7b, where devices R are used to reverse the signals at the input and output of the second filter. In this arrangement, the first filter introduces a certain delay, which depends on the frequency, and thus a certain amount of delay distortion is introduced. The second filter introduces exactly the same delay as the first, but, since the signal is fed backward, the delay is actually a time advance and, therefore, cancels the delay of the first filter.

The scheme of Fig. 12.7 is suitable for nonreal-time applications since it uses a noncausal filter. An alternative approach for the design of constant-delay filters that can be used for nonreal- or real-time applications is to use nonrecursive approximations which are explored in Chaps. 9 and 15.

12.6 AMPLITUDE EQUALIZATION

In many applications, a filter is required to operate in cascade with a channel or system that does not have a constant amplitude response (e.g., a D/A converter, Fig. 6.17d). If the transfer function of such a channel is $H_C(z)$ and the passband of the channel-filter combination extends from ω_{p1} to ω_{p2}, then the transfer function of the filter must be chosen such that

$$|H_C(e^{j\omega T})H_F(e^{j\omega T})| = 1 \qquad \text{for } \omega_{p1} \le \omega \le \omega_{p2}$$

to within a prescribed tolerance in order to keep the amplitude distortion to an acceptable level (see Sec. 5.7). If the variation in the amplitude response of the channel is small, it may be possible to solve the problem by taking the channel loss into account when the filter specifications are formulated. Alternatively, if the variation of the amplitude response of the channel is large, then the filter may have to be tuned or redesigned using one of the optimization methods described in Chap. 16 (e.g., see Example 16.3).

REFERENCES

[1] A. Antoniou, “Design of elliptic digital filters: Prescribed specifications,” *Proc. Inst. Elect. Eng.*, Part G, vol. 124, pp. 341–344, Apr. 1977 (see vol. 125, p. 504, June 1978 for errata).

PROBLEMS

12.1. Design a lowpass digital filter that would satisfy the specifications of Fig. P12.1. Use a Butterworth approximation.

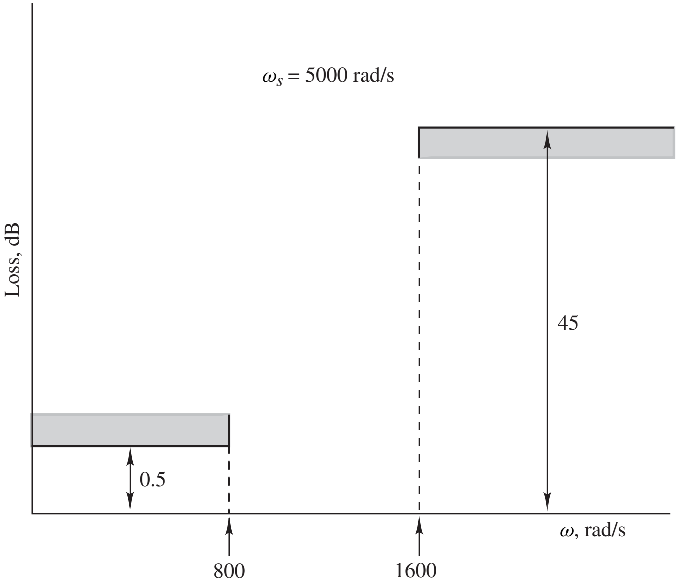


Figure P12.1

12.2. Redesign the filter of Prob. 12.1 using a Chebyshev approximation.

12.3. Redesign the filter of Prob. 12.1 using an inverse-Chebyshev approximation.

12.4. Redesign the filter of Prob. 12.1 using an elliptic approximation.

12.5. Design a highpass digital filter that would satisfy the specifications of Fig. P12.5. Use a Butterworth approximation.

12.6. Redesign the filter of Prob. 12.5 using a Chebyshev approximation.

12.7. Redesign the filter of Prob. 12.5 using an inverse-Chebyshev approximation.

12.8. Redesign the filter of Prob. 12.5 using an elliptic approximation.

12.9. Design a bandpass digital filter that would satisfy the specifications of Fig. P12.9. Use a Butterworth approximation.

12.10. Redesign the filter of Prob. 12.9 using a Chebyshev approximation.

12.11. Redesign the filter of Prob. 12.9 using an inverse-Chebyshev approximation.

12.12. Redesign the filter of Prob. 12.9 using an elliptic approximation.

12.13. Design a bandstop digital filter that would satisfy the specifications of Fig. P12.13. Use a Butterworth approximation.

12.14. Redesign the filter of Prob. 12.13 using a Chebyshev approximation.

12.15. Redesign the filter of Prob. 12.13 using an inverse-Chebyshev approximation.

12.16. Redesign the filter of Prob. 12.13 using an elliptic approximation.

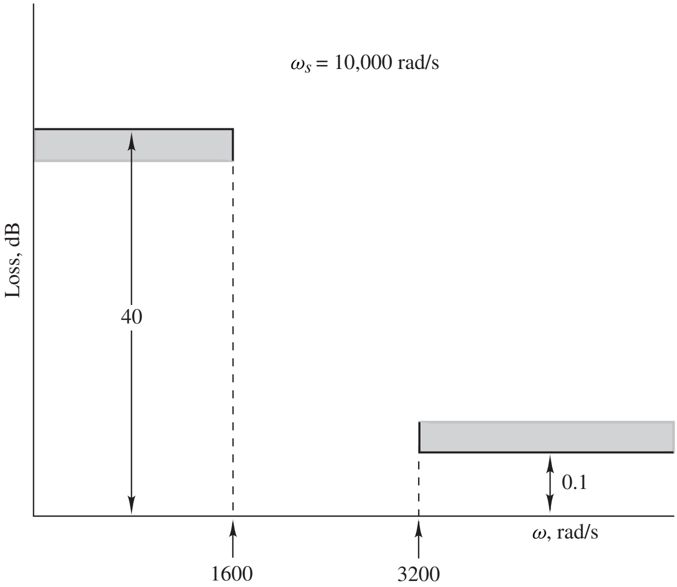


Figure P12.5

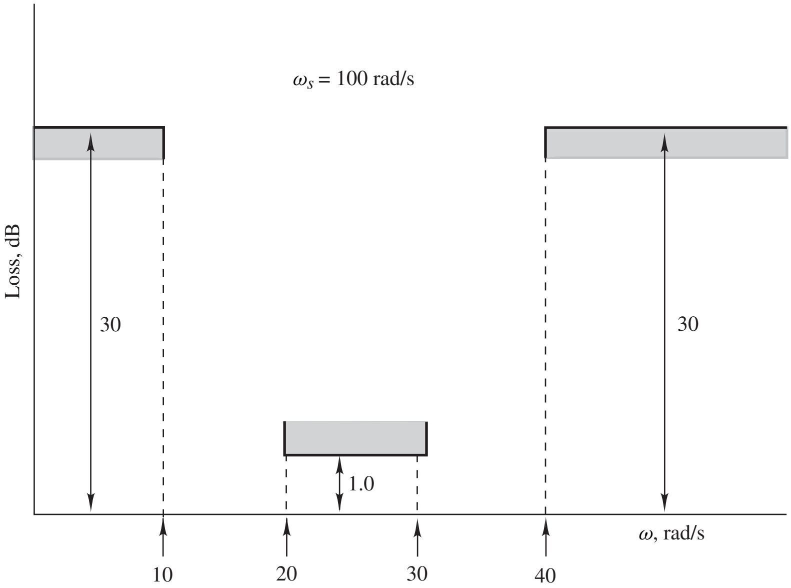


Figure P12.9

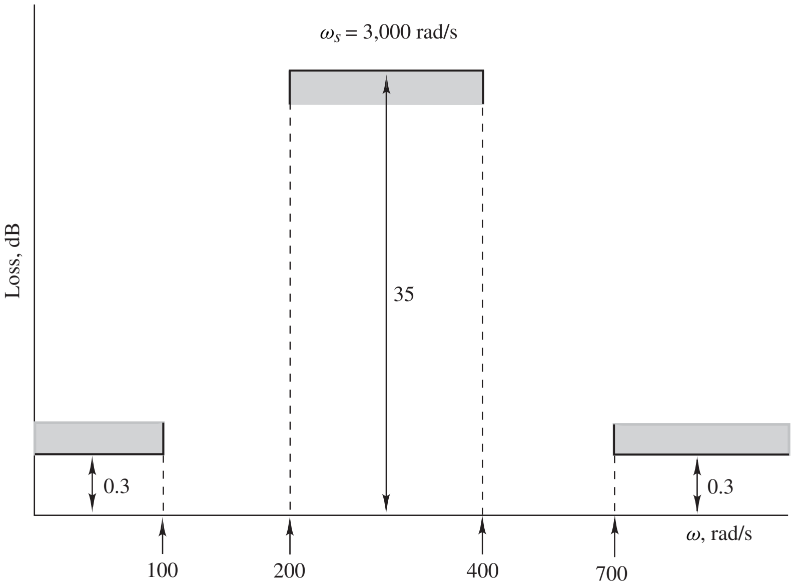


Figure P12.13

12.17. Derive the formulas of Table 12.2 for highpass filters.

12.18. Derive the formulas of Table 12.3 for bandstop filters.

12.19. Shaw that the transfer function of Eq. (12.40) is an allpass transfer function.

12.20. A digital filter with an impulse response $h(nT)$ has a transfer function $H(z)$. Show that a filter with a transfer function $H(z^{-1})$ has an impulse response $h(-nT)$.

This page intentionally left blank

CHAPTER 13

RANDOM SIGNALS

13.1 INTRODUCTION

The methods of analysis considered so far assume deterministic signals. Frequently in digital filters and communication systems in general random signals are encountered, e.g., the noise generated by an analog-to-digital (A/D) converter or the noise generated by an amplifier. Signals of this type can assume an infinite number of waveforms, and measurement will at best yield a set of typical waveforms. Despite the lack of a complete description, many statistical attributes of a random signal can be determined from a statistical description of the signal.

The time- and frequency-domain statistical attributes of random signals as well as the effect of filtering on such signals can be studied by using the concept of a *random process*.

This chapter provides a brief description of random processes. The main results are presented in terms of continuous-time random signals and are then extended to discrete-time signals by using the interrelation between the Fourier and z transforms. The chapter begins with a brief summary of the essential features of random variables. Detailed discussions of random variables and processes can be found in [1–5].

13.2 RANDOM VARIABLES

Consider an experiment which may have a finite or infinite number of random outcomes, and let ζ_1, $\zeta_2, \ldots$ be the possible outcomes. A set S comprising all the possible ζ can be constructed, and a number $\mathbf{x}(\zeta)$ can be assigned to each ζ according to some rule. The function $\mathbf{x}(\zeta)$, or simply $\mathbf{x}$, whose domain is set S and whose range is a set of numbers is called a *random variable*. Typical random variables are the coordinates of the hit position in an experiment of target practice or the speed and direction of the wind at some specified instant in a given region or at some specified location over a period of time. Specific random variables that will be studied in some detail in Chap. 14 are the errors introduced by the quantization of signals and filter coefficients.

13.2.1 Probability-Distribution Function

A random variable $\mathbf{x}$ may assume values in a certain range (x_1, x_2), where x_1 can be as low as $-\infty$ and x_2 as high as $+\infty$. The probability of observing random variable $\mathbf{x}$ below or at value x is referred to as the *probability-distribution function* of $\mathbf{x}$ and is denoted as

$$P_{\mathbf{x}}(x) = \Pr\,[\mathbf{x} \le x]$$

13.2.2 Probability-Density Function

The derivative of $P_{\mathbf{x}}(x)$ with respect to x is called the *probability-density function* of $\mathbf{x}$ and is denoted as

$$p_{\mathbf{x}}(x) = \frac{dP_{\mathbf{x}}(x)}{dx}$$

A fundamental property of $p_{\mathbf{x}}(x)$ is

$$\int_{-\infty}^{\infty} p_{\mathbf{x}}(x)\,dx = 1$$

since the range $(-\infty, +\infty)$ must necessarily include the value of $\mathbf{x}$. Also

$$\mathrm{P}_{\mathbf{x}}[x_1 \le \mathbf{x} \le x_2] = \int_{x_1}^{x_2} p_{\mathbf{x}}(x)\,dx$$

13.2.3 Uniform Probability Density

In many situations there is no preferred value or range for the random variable. In such a case, the probability density is said to be *uniform* and is given by

$$p_{\mathbf{x}}(x) = \begin{cases} \dfrac{1}{x_2 - x_1} & \text{for } x_1 \le x \le x_2 \\ 0 & \text{otherwise} \end{cases}$$

13.2.4 Gaussian Probability Density

Very common in nature is the *Gaussian* probability density given by

$$p_{\mathbf{x}}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\eta)^2/2\sigma^2} \qquad \text{for } -\infty \le x \le \infty \tag{13.1}$$

The parameters σ and η are constants.

There are many other important probability-density functions, e.g., *binomial*, *Poisson*, and *Rayleigh* [1], but these are beyond the scope of this book.

13.2.5 Joint Distributions

An experiment may have two sets of random outcomes, say, $\zeta_{x1}, \zeta_{x2}, \ldots$ and ζ_{y1}, $\zeta_{y2}, \ldots$. For example, in an experiment of target practice, the hit position can be described in terms of two coordinates. Experiments of this type necessitate two random variables, say, $\mathbf{x}$ and $\mathbf{y}$. The probability of observing $\mathbf{x}$ and $\mathbf{y}$ below or at x and y, respectively, is said to be the *joint distribution function* of $\mathbf{x}$ and $\mathbf{y}$ and is denoted as

$$P_{\mathbf{xy}}(x, y) = \Pr\,[\mathbf{x} \le x, \mathbf{y} \le y]$$

The joint probability-density function of $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$p_{\mathbf{xy}}(x, y) = \frac{\partial^2 P_{\mathbf{xy}}(x, y)}{\partial x \partial y}$$

The range $(-\infty, \infty)$ must include the values of $\mathbf{x}$ and $\mathbf{y}$, and hence

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_{\mathbf{xy}}(x, y)\, dx\, dy = 1$$

The probability of observing $\mathbf{x}$ and $\mathbf{y}$ in the ranges $x_1 \le x \le x_2$ and $y_1 \le y \le y_2$, respectively, is given by

$$\Pr\,[x_1 \le \mathbf{x} \le x_2, y_1 \le \mathbf{y} \le y_2] = \int_{y_1}^{y_2} \int_{x_1}^{x_2} p_{\mathbf{xy}}(x, y)\, dx\, dy$$

Two random variables $\mathbf{x}$ and $\mathbf{y}$ representing outcomes $\zeta_{x1}, \zeta_{x2}, \ldots$ and $\zeta_{y1}, \zeta_{y2}, \ldots$ of an experiment are said to be *statistically independent* if the occurrence of any outcome ζ_x does not influence the occurrence of any outcome ζ_y and vice versa. A necessary and sufficient condition for statistical independence is

$$p_{\mathbf{xy}}(x, y) = p_{\mathbf{x}}(x) p_{\mathbf{y}}(y) \tag{13.2}$$

13.2.6 Mean Values and Moments

The *mean* or *expected value* of random variable $\mathbf{x}$ is defined as

$$E\{\mathbf{x}\} = \int_{-\infty}^{\infty} x p_{\mathbf{x}}(x)\, dx$$

Similarly, if a random variable $\mathbf{z}$ is a function of two other random variables $\mathbf{x}$ and $\mathbf{y}$, that is,

$$\mathbf{z} = f(\mathbf{x}, \mathbf{y})$$

then

$$E\{\mathbf{z}\} = \int_{-\infty}^{\infty} z p_{\mathbf{z}}(z)\, dz \tag{13.3}$$

If $\mathbf{z}$ is a single-valued function of $\mathbf{x}$ and $\mathbf{y}$ and $x \le \mathbf{x} \le x+dx$, $y \le \mathbf{y} \le y+dy$, then $z \le \mathbf{z} \le z+dz$. Hence

$$\Pr\,[z \le \mathbf{z} \le z + dz] = \Pr\,[x \le \mathbf{x} \le x + dx, y \le \mathbf{y} \le y + dy]$$

or

$$p_{\mathbf{z}}(z)dz = p_{\mathbf{xy}}(x, y)dx\,dy$$

and from Eq. (13.3)

$$E\{\mathbf{z}\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)p_{\mathbf{xy}}(x, y)\,dx\,dy$$

Actually this is a general relation that holds for multivalued functions as well [1]. For

$$\mathbf{z} = \mathbf{xy}$$

we have

$$E\{\mathbf{xy}\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyp_{\mathbf{xy}}(x, y)\,dx\,dy$$

and if variables **x** and **y** are statistically independent, then the use of Eq. (13.2) yields

$$E\{\mathbf{xy}\} = \int_{-\infty}^{\infty} xp_{\mathbf{x}}(x)\,dx \int_{-\infty}^{\infty} yp_{\mathbf{y}}(y)\,dy = E\{\mathbf{x}\}E\{\mathbf{y}\} \tag{13.4}$$

The nth moment of **x** is defined as

$$E\{\mathbf{x}^n\} = \int_{-\infty}^{\infty} x^n p_{\mathbf{x}}(x)\,dx$$

The second moment is usually referred to as the *mean square* of **x**.

The nth central moment of **x** is defined as

$$E\{(\mathbf{x} - E\{\mathbf{x}\})^n\} = \int_{-\infty}^{\infty} (x - E\{\mathbf{x}\})^n p_{\mathbf{x}}(x)\,dx \tag{13.5}$$

The second central moment is commonly referred to as the *variance* and is given by

$$\begin{aligned}\sigma_{\mathbf{x}}^2 &= E\{(\mathbf{x} - E\{\mathbf{x}\})^2\}\\ &= E\{\mathbf{x}^2 - 2\mathbf{x}E\{\mathbf{x}\} + (E\{\mathbf{x}\})^2\}\\ &= E\{\mathbf{x}^2\} - (E\{\mathbf{x}\})^2 \end{aligned}\tag{13.6}$$

If

$$\mathbf{z} = a_1\mathbf{x}_1 + a_2\mathbf{x}_2$$

where a_1, a_2 are constants and $\mathbf{x}_1$, $\mathbf{x}_2$ are statistically independent random variables, then from Eqs. (13.4) and (13.5), we have

$$\sigma_{\mathbf{z}}^2 = a_1^2\sigma_{\mathbf{x}_1}^2 + a_2^2\sigma_{\mathbf{x}_2}^2$$

In general, if

$$\mathbf{z} = \sum_{i=1}^{n} a_i \mathbf{x}_i$$

and variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are statistically independent, then

$$\sigma_{\mathbf{z}}^2 = \sum_{i=1}^{n} a_i^2 \sigma_{\mathbf{x}_i}^2 \tag{13.7}$$

Example 13.1 (*a*) Find the mean and variance for a random variable with a uniform probability density given by

$$p_{\mathbf{x}}(x) = \begin{cases} \dfrac{1}{x_2 - x_1} & \text{for } x_1 \leq x \leq x_2 \\ 0 & \text{otherwise} \end{cases}$$

(*b*) Repeat part (*a*) for a random variable with a Gaussian probability density

$$p_{\mathbf{x}}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\eta)^2/2\sigma^2} \qquad \text{for } -\infty \leq x \leq \infty$$

■ **Solution**

(*a*) From the definition of the mean, we have

$$E\{\mathbf{x}\} = \int_{x_1}^{x_2} \frac{x}{x_2 - x_1}\, dx = \frac{1}{2}(x_1 + x_2) \quad ■ \tag{13.8}$$

Similarly, the mean square can be deduced as

$$E\{\mathbf{x}^2\} = \int_{x_1}^{x_2} \frac{x^2}{x_2 - x_1}\, dx = \frac{x_2^3 - x_1^3}{3(x_2 - x_1)} \tag{13.9}$$

and from Eq. (13.6), we obtain

$$\sigma_{\mathbf{x}}^2 = \frac{(x_2 - x_1)^2}{12} \quad ■ \tag{13.10}$$

(*b*) In this case, we can write

$$E\{\mathbf{x}\} = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} x e^{-(x-\eta)^2/2\sigma^2}\, dx$$

and with $x = y + \eta$

$$E\{\mathbf{x}\} = \frac{1}{\sigma\sqrt{2\pi}}\left(\int_{-\infty}^{\infty} ye^{-y^2/2\sigma^2}\,dy + \eta\int_{-\infty}^{\infty} e^{-y^2/2\sigma^2}\,dy\right)$$

The first integral is zero because the integrand is an odd function of y whereas the second integral is equal to $\sigma\sqrt{2\pi}$ according to standard tables of integrals. Hence

$$E\{\mathbf{x}\} = \eta \quad \blacksquare$$

Now

$$E\{\mathbf{x}^2\} = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty} x^2 e^{-(x-\eta)^2/2\sigma^2}\,dx$$

and, as before,

$$E\{\mathbf{x}^2\} = \sigma^2 + \eta^2 \qquad \text{or} \qquad \sigma_{\mathbf{x}}^2 = \sigma^2 \quad \blacksquare$$

13.3 RANDOM PROCESSES

A random process is an extension of the concept of a random variable. Consider an experiment with possible random outcomes $\zeta_1, \zeta_2, \ldots$. A set S comprising all ζ can be constructed and a waveform $\mathbf{x}(t, \zeta)$ can be assigned to each ζ according to some rule. The set of waveforms obtained is called an *ensemble*, and each individual waveform is said to be a *sample function*. Set S, the ensemble, and the probability description associated with S constitute a *random process*.

The concept of a random process can be illustrated by an example. Suppose that a large number of radio receivers of a particular model are receiving a carrier signal transmitted by a broadcasting station. With the receivers located at different distances from the broadcasting station, the amplitude and phase of the received carrier will be different at each receiver. As a result, the set of the received waveforms, illustrated in Fig. 13.1, can be described by

$$\mathbf{x}(t, \zeta) = \mathbf{z}\cos(\omega_c t + \mathbf{y})$$

where $\mathbf{z}$ and $\mathbf{y}$ are random variables and $\zeta = \zeta_1, \zeta_2, \ldots$. The set of all possible waveforms that might be received constitutes an ensemble and the ensemble together with the probability densities of $\mathbf{z}$ and $\mathbf{y}$ constitutes a random process.

13.3.1 Notation

A random process can be represented by $\mathbf{x}(t, \zeta)$ or in a simplified notation by $\mathbf{x}(t)$. Depending on the circumstances, $\mathbf{x}(t, \zeta)$ can represent one of four things as follows:

1. The *ensemble*, if t and ζ are variables.

2. A *sample function*, if t is variable and ζ is fixed.

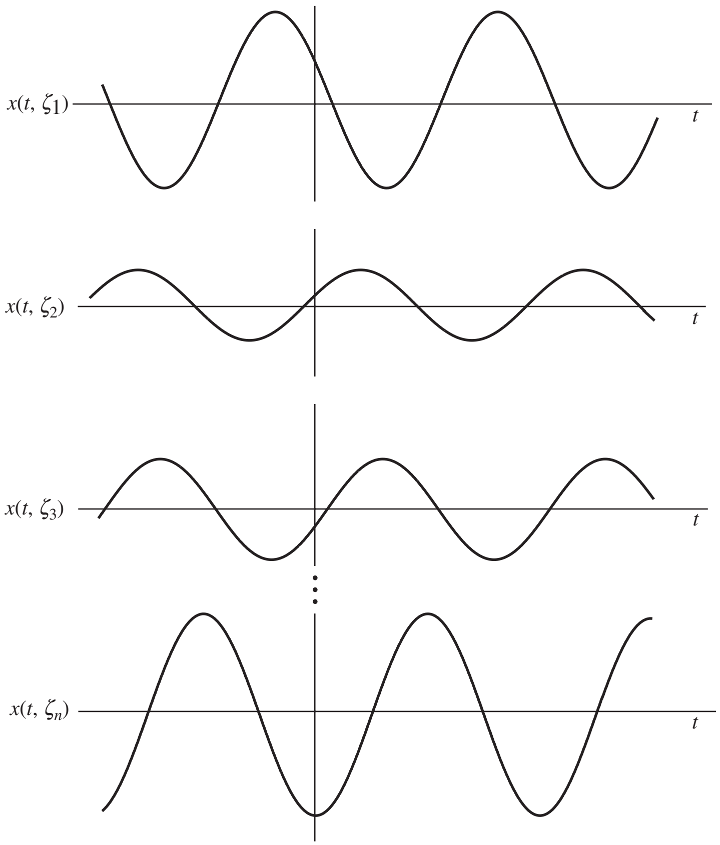


Figure 13.1 A random process.

3. A *random variable*, if t is fixed and ζ is variable.

4. A *single number*, if t and ζ are fixed.

13.4 FIRST- AND SECOND-ORDER STATISTICS

For a fixed value of t, $\mathbf{x}(t)$ is a random variable representing the instantaneous values of the various sample functions over the ensemble. The probability distribution and probability density of $\mathbf{x}(t)$ are denoted as

$$P(x;t) = \Pr[\mathbf{x}(t) \le x] \qquad \text{and} \qquad p(x;t) = \frac{\partial P(x;t)}{\partial x}$$

respectively. These two equations constitute the *first-order statistics* of the random process.

At any two instants t_1 and t_2, $\mathbf{x}(t_1)$ and $\mathbf{x}(t_2)$ are distinct random variables. Their joint probability distribution and joint probability density depend on t_1 and t_2 in general, and are denoted as

$$P(x_1, x_2; t_1, t_2) = \Pr[\mathbf{x}(t_1) \le x_1,\ \mathbf{x}(t_2) \le x_2]$$

and

$$p(x_1, x_2; t_1, t_2) = \frac{\partial^2 P(x_1, x_2; t_1, t_2)}{\partial x_1 \partial x_2}$$

respectively. These two equations constitute the *second-order statistics* of the random process.

Similarly, at any k instants $t_1, t_2, \ldots, t_k$, the quantities $\mathbf{x}_1, \mathbf{x}_2, \ldots,$ and $\mathbf{x}_k$ are distinct random variables. Their joint probability distribution and joint probability density depend on $t_1, t_2, \ldots, t_k$ and can be defined as before. These quantities constitute the *kth-order statistics* of the random process.

Example 13.2 Find the first-order probability density $p(x; t)$ for random process

$$\mathbf{x}(t) = \mathbf{y}t - 2$$

where $\mathbf{y}$ is a random variable with a probability density

$$p_{\mathbf{y}}(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \qquad \text{for } -\infty \le y \le \infty$$

■ **Solution**

If x and y are possible values of $\mathbf{x}(t)$ and $\mathbf{y}$, then

$$x = yt - 2 \qquad \text{or} \qquad y = \frac{1}{t}(x + 2)$$

From Fig. 13.2

$$\Pr[x \le \mathbf{x} \le x + |dx|] = \Pr[y \le \mathbf{y} \le y + |dy|]$$

i.e.,

$$p_{\mathbf{x}}(x)|dx| = p_{\mathbf{y}}(y)|dy| \qquad \text{or} \qquad p_{\mathbf{x}}(x) = \frac{p_{\mathbf{y}}(y)}{|dx/dy|}$$

Since

$$\frac{dx}{dy} = t$$

we obtain

$$p(x; t) = p_{\mathbf{x}}(x) = \frac{1}{|t|\sqrt{2\pi}} e^{-(x+2)^2/2t^2} \qquad \text{for } -\infty \le x \le \infty \quad ■$$

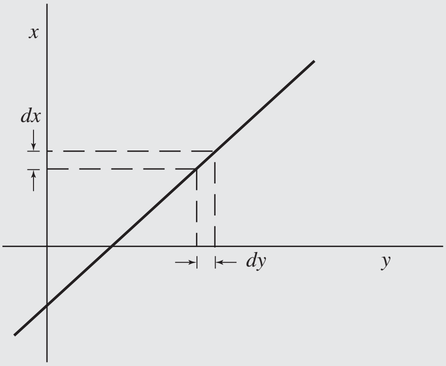


Figure 13.2 Function $x = yt - 2$ (Example 13.2).

Example 13.3 Find the first-order probability density $p(x; t)$ of the random process

$$\mathbf{x}(t) = \cos(\omega_c t + \mathbf{y})$$

where **y** is a random variable with probability density

$$p_{\mathbf{y}}(y) = \begin{cases} \dfrac{1}{2\pi} & \text{for } 0 \le y \le 2\pi \\ 0 & \text{otherwise} \end{cases}$$

■ **Solution**

If x and y are possible values of $\mathbf{x}(t)$ and $\mathbf{y}$, then

$$x = \cos(\omega_c t + y)$$

and from Fig. 13.3, we get

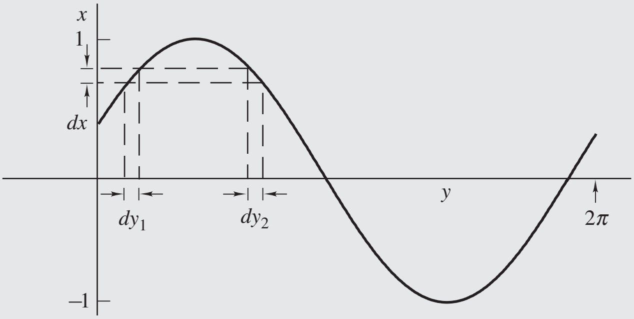


Figure 13.3 Function $x = \cos(\omega_c t + y)$ (Example 13.3).

$$\Pr[x \le \mathbf{x} \le x + |dx|] = \Pr[y_1 \le \mathbf{y} \le y_1 + |dy_1|]$$
$$+\Pr[y_2 \le \mathbf{y} \le y_2 + |dy_2|]$$

or
$$p_{\mathbf{x}}(x)|dx| = p_{\mathbf{y}}(y_1)|dy_1| + p_{\mathbf{y}}(y_2)|dy_2|$$

Hence
$$p_{\mathbf{x}}(x) = \frac{p_{\mathbf{y}}(y_1)}{|x'(y_1)|} + \frac{p_{\mathbf{y}}(y_2)}{|x'(y_2)|}$$

where
$$x'(y) = \frac{dx}{dy} = -\sin(\omega_c t + y) = -\sqrt{1 - x^2}$$

Since
$$p_{\mathbf{y}}(y_1) = p_{\mathbf{y}}(y_2) = p_{\mathbf{y}}(y) \qquad \text{and} \qquad |x'(y_1)| = |x'(y_2)| = |x'(y)|$$

we obtain
$$p(x;t) = p_{\mathbf{x}}(t) = \begin{cases} \frac{1}{\pi\sqrt{1 - x^2}} & \text{for } |x| < 1 \\ 0 & \text{otherwise} \end{cases} \qquad \blacksquare$$

13.5 MOMENTS AND AUTOCORRELATION

The first-order statistics give the mean, mean square, and other moments of a random process at any instant t. From Sec. 13.2.6

$$E\{\mathbf{x}(t)\} = \int_{-\infty}^{\infty} x p(x;t)\, dx$$

$$E\{\mathbf{x}^2(t)\} = \int_{-\infty}^{\infty} x^2 p(x;t)\, dx$$

The second-order statistics give the *autocorrelation function* of a random process, which is defined as

$$r_{\mathbf{x}}(t_1, t_2) = E\{\mathbf{x}(t_1)\mathbf{x}(t_2)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 x_2 p(x_1, x_2; t_1, t_2)\, dx_1\, dx_2$$

The autocorrelation is a measure of the interdependence between the instantaneous signal values at $t = t_1$ and those at $t = t_2$. This is the most important attribute of a random process, as it leads to a frequency-domain description of the process.

Example 13.4 (*a*) Find the mean, mean square, and autocorrelation for the random process in Example 13.2. (*b*) Repeat part (*a*) for the process of Example 13.3.

■ **Solution**

(*a*) The probability density of $\mathbf{x}(t)$ has been obtained in Example 13.2 as

$$p(x;t) = p_{\mathbf{x}}(x) = \frac{1}{|t|\sqrt{2\pi}} e^{-(x+2)^2/2t^2} \qquad \text{for } -\infty \le x \le \infty$$

Now the mean and mean square of a random variable $\mathbf{x}$ with a Gaussian probability density

$$p_{\mathbf{x}}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\eta)^2/2\sigma^2} \qquad \text{for } -\infty \le x \le \infty$$

have been obtained in Example 13.1 as

$$E\{\mathbf{x}\} = \eta \qquad \text{and} \qquad E\{\mathbf{x}^2\} = \sigma^2 + \eta^2$$

respectively. Thus, by comparison, the mean and mean square of $\mathbf{x}(t)$ can be readily obtained as

$$E\{\mathbf{x}(t)\} = -2 \qquad \text{and} \qquad E\{\mathbf{x}^2(t)\} = t^2 + 4 \quad ■$$

The autocorrelation is given by

$$r_{\mathbf{x}}(t_1, t_2) = E\{(\mathbf{y}t_1 - 2)(\mathbf{y}t_2 - 2)\} = t_1 t_2 E\{\mathbf{y}^2\} - 2(t_1 + t_2)E\{\mathbf{y}\} + 4$$

and since $\mathbf{y}$ is a random variable with a probability density

$$p_{\mathbf{y}}(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \qquad \text{for } -\infty \le y \le \infty$$

(see Example 13.2), we have

$$E\{\mathbf{y}\} = 0 \qquad \text{and} \qquad E\{\mathbf{y}^2\} = 1$$

and

$$r_{\mathbf{x}}(t_1, t_2) = t_1 t_2 + 4 \quad ■$$

(*b*) The probability density of $\mathbf{x}(t)$ was obtained in Example 13.3 as

$$p(x;t) = p_{\mathbf{x}}(t) = \begin{cases} \frac{1}{\pi\sqrt{1-x^2}} & \text{for } |x| < 1 \\ 0 & \text{otherwise} \end{cases}$$

Thus the mean and mean square of $\mathbf{x}(t)$ can be readily obtained as

$$E\{\mathbf{x}(t)\} = \frac{1}{\pi}\int_{-1}^{1} \frac{x}{\sqrt{1-x^2}}\,dx = 0 \quad \blacksquare \tag{13.11}$$

$$E\{\mathbf{x}^2(t)\} = \frac{1}{\pi}\int_{-1}^{1} \frac{x^2}{\sqrt{1-x^2}}\,dx = \tfrac{1}{2} \quad \blacksquare$$

The autocorrelation can be expressed as

$$\begin{aligned} r_{\mathbf{x}}(t_1, t_2) &= E\{\cos(\omega_c t_1 + \mathbf{y})\cos(\omega_c t_2 + \mathbf{y})\} \\ &= \tfrac{1}{2}\cos(\omega_c t_1 - \omega_c t_2) + \tfrac{1}{2}E\{\cos(\omega_c t_1 + \omega_c t_2 + 2\mathbf{y})\} \end{aligned}$$

Now

$$\bar{\mathbf{x}}(t) = \cos(\omega_c t_1 + \omega_c t_2 + 2\mathbf{y})$$

is a random variable of the same type as $\mathbf{x}(t)$ in Example 13.3, whose probability density can be obtained as

$$p(\bar{x}; t) = p_{\bar{\mathbf{x}}}(t) = \begin{cases} \frac{1}{\pi\sqrt{1-\bar{x}^2}} & \text{for } |\bar{x}| < 1 \\ 0 & \text{otherwise} \end{cases}$$

(see Example 13.3), and hence $E\bar{x}(t) = 0$. Therefore,

$$r_{\mathbf{x}}(t_1, t_2) = \tfrac{1}{2}\cos\omega_c\tau \qquad \text{where } \tau = t_1 - t_2 \quad \blacksquare \tag{13.12}$$

13.6 STATIONARY PROCESSES

A random process is said to be *strictly stationary* if $\mathbf{x}(t)$ and $\mathbf{x}(t+T)$ have the same statistics (all orders) for any value of T. If the mean of $\mathbf{x}(t)$ is constant and its autocorrelation depends only on $t_2 - t_1$, that is,

$$E\{\mathbf{x}(t)\} = \text{ constant} \qquad E\{\mathbf{x}(t_1)\mathbf{x}(t_2)\} = r_{\mathbf{x}}(t_2 - t_1)$$

the process is called *wide-sense stationary*. A strictly stationary process is also stationary in the wide sense; however, the converse is not necessarily true. The process of Example 13.4, part (*a*), is wide-sense stationary; however, that of Example 13.4, part (*b*), is not stationary.

13.7 FREQUENCY-DOMAIN REPRESENTATION

The frequency-domain representation of deterministic signals is normally in terms of amplitude, phase, and energy-density spectrums (see Chap. 2). Although such representations are possible for random processes [1], they are avoided in practice because of the mathematical difficulties associated with infinite-energy signals (see Sec. 6.2). Usually, random processes are represented in terms of power-density spectra.

Consider a signal $x(t)$ and let

$$x_{T_0}(t) = \begin{cases} x(t) & \text{for } |t| \le T_0 \\ 0 & \text{otherwise} \end{cases}$$

The average power of $x(t)$ over the interval $[-T_0, T_0]$ is

$$P_{T_0} = \frac{1}{2T_0}\int_{-T_0}^{T_0} x^2(t)\,dt = \frac{1}{2T_0}\int_{-\infty}^{\infty} x_{T_0}^2(t)\,dt$$

and by virtue of Parseval's formula (see Theorem 2.16)

$$P_{T_0} = \int_{-\infty}^{\infty} \frac{|X_{T_0}(j\omega)|^2}{2T_0}\,\frac{d\omega}{2\pi}$$

where

$$X_{T_0}(j\omega) = \mathcal{F}x_{T_0}(t)$$

Evidently, the elemental area in the above integral, namely,

$$\frac{|X_{T_0}(j\omega)|^2}{2T_0}\,\frac{d\omega}{2\pi} = \frac{|X_{T_0}(j\omega)|^2}{2T_0}df$$

represents average power (f is the frequency in hertz). Therefore, the quantity

$$\frac{|X_{T_0}(j\omega)|^2}{2T_0}$$

represents the average power per unit bandwidth (in hertz) and can be referred to as the *power spectral density* (PSD) of $x_{T_0}(t)$. If $x_{T_0}(t)$ and $x(t)$ are sample functions of random processes $\mathbf{x}_{T_0}(t)$ and $\mathbf{x}(t)$, respectively, we can define

$$\text{PSD of } \mathbf{x}_{T_0}(t) = E\left\{\frac{|X_{T_0}(j\omega)|^2}{2T_0}\right\}$$

and since $\mathbf{x}_{T_0}(t) \to \mathbf{x}(t)$ as $T_0 \to \infty$, we obtain

$$\text{PSD of } \mathbf{x}(t) = S_{\mathbf{x}}(\omega) = \lim_{T_0\to\infty} E\left\{\frac{|X_{T_0}(j\omega)|^2}{2T_0}\right\} \tag{13.13}$$

The function $S_{\mathbf{x}}(\omega)$ is said to be the *power-density spectrum* of the process.

For stationary processes, the *PSD is the Fourier transform of the autocorrelation function*, as we shall now demonstrate. From Eq. (13.13)

$$\begin{aligned} S_{\mathbf{x}}(\omega) &= \lim_{T_0\to\infty} E\left\{\frac{X_{T_0}(j\omega)X_{T_0}^*(j\omega)}{2T_0}\right\} \\ &= \lim_{T_0\to\infty} \frac{1}{2T_0} E\left\{\int_{-T_0}^{T_0} \mathbf{x}(t_2)e^{-j\omega t_2}\,dt_2 \int_{-T_0}^{T_0} \mathbf{x}(t_1)e^{j\omega t_1}\,dt_1\right\} \\ &= \lim_{T_0\to\infty} \frac{1}{2T_0} \int_{-T_0}^{T_0}\int_{-T_0}^{T_0} E\{\mathbf{x}(t_1)\mathbf{x}(t_2)\}e^{-j\omega(t_2-t_1)}\,dt_1\,dt_2 \end{aligned}$$

For a wide-sense-stationary process, we have

$$E\{\mathbf{x}(t_1)\mathbf{x}(t_2)\} = r_{\mathbf{x}}(t_2 - t_1)$$

and hence we can write

$$S_{\mathbf{x}}(\omega) = \lim_{T_0\to\infty} \frac{1}{2T_0} \int_{-T_0}^{T_0}\int_{-T_0}^{T_0} f(t_2 - t_1)\,dt_1\,dt_2 \tag{13.14}$$

where

$$f(t_2 - t_1) = r_{\mathbf{x}}(t_2 - t_1)e^{-j\omega(t_2-t_1)} \tag{13.15}$$

The preceding double integral represents the volume under the surface $y = f(t_2 - t_1)$ and above the square region in Fig. 13.4. Since $f(t_2 - t_1)$ is constant on any line of the form

$$t_2 = t_1 + c$$

the volume over the elemental area bounded by the square region and the lines

$$t_2 = t_1 + \tau \qquad \text{and} \qquad t_2 = t_1 + \tau + d\tau$$

is approximately constant. From the geometry of Fig. 13.4, we note that the elemental area dA is the difference between the areas of two overlapping equilateral right-angled triangles. For $\tau \geq 0$, the sides of the larger and smaller triangles are $2T_0 - \tau$ and $2T_0 - (\tau + d\tau)$, respectively, and hence

$$\begin{aligned} dA &= \frac{1}{2}(2T_0 - \tau)^2 - \frac{1}{2}[2T_0 - (\tau + d\tau)]^2 \\ &= (2T_0 - \tau)d\tau + \frac{1}{2}(d\tau)^2 \\ &\approx (2T_0 - \tau)d\tau \end{aligned}$$

Similarly, for $\tau < 0$, we get

$$dA \approx (2T_0 + \tau)d\tau$$

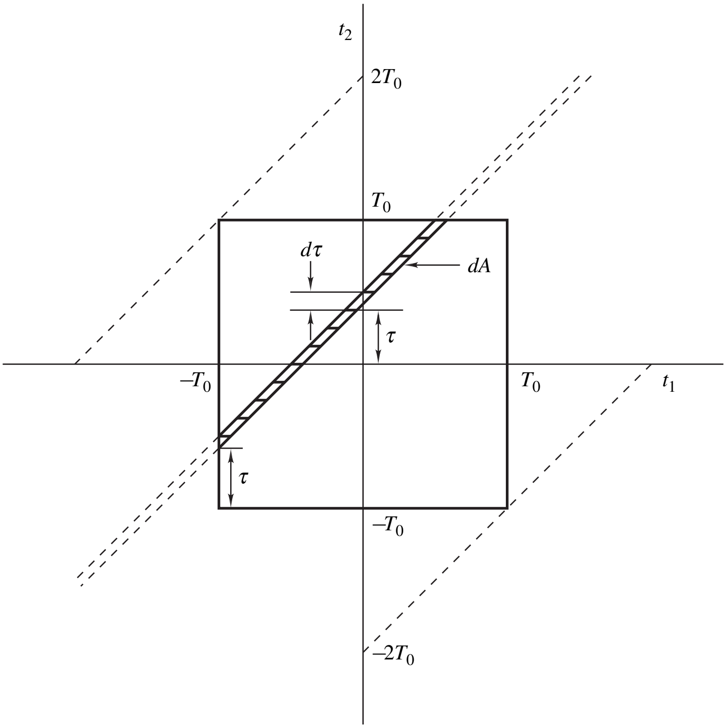


Figure 13.4 Domain of $y = f(t_2 - t_1)$.

and in general, as $d\tau \to 0$, we can write

$$dA = (2T_0 - |\tau|)d\tau$$

Hence the elemental volume for $t_2 - t_1 = \tau$ is

$$dV = f(\tau)(2T_0 - |\tau|)\,d\tau$$

In order to obtain the entire volume under the surface $y = f(t_2 - t_1)$ and above the square region in Fig. 13.4, τ must be increased from $-2T_0$ to $+2T_0$; thus Eq. (13.14) can be expressed as

$$S_{\mathbf{x}}(\omega) = \lim_{T_0\to\infty} \frac{1}{2T_0} \int_{-2T_0}^{2T_0} f(\tau)(2T_0 - |\tau|)\,d\tau$$

$$= \int_{-\infty}^{\infty} f(\tau) \lim_{T_0\to\infty} \left(1 - \frac{|\tau|}{2T_0}\right) d\tau = \int_{-\infty}^{\infty} f(\tau)\,d\tau$$

Therefore, from Eq. (13.15)

$$S_{\mathbf{x}}(\omega) = \int_{-\infty}^{\infty} r_{\mathbf{x}}(\tau)e^{-j\omega\tau}\,d\tau \tag{13.16}$$

and if

$$\int_{-\infty}^{\infty} |r_{\mathbf{x}}(\tau)|\, d\tau < \infty$$

we can write

$$r_{\mathbf{x}}(\tau) = E\{\mathbf{x}(t)\mathbf{x}(t+\tau)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{\mathbf{x}}(\omega)e^{j\omega\tau}\, d\omega \tag{13.17}$$

i.e.,

$$r_{\mathbf{x}}(\tau) \leftrightarrow S_{\mathbf{x}}(\omega)$$

by virtue of the convergence theorem of the Fourier transform (Theorem 2.5). The formula in Eq. (13.16) is known as the *Wiener-Khinchine relation*.

Example 13.5 Find the PSD of the process in Example 13.3.

■ **Solution**

The autocorrelation of the process was obtained in Example 13.4, part (*b*), as

$$r(\tau) = \tfrac{1}{2}\cos\omega_c\tau$$

(see Eq. (13.12)). Hence from Eq. (13.16) and Table 6.2

$$S_{\mathbf{x}}(\omega) = \mathcal{F}r(\tau) = \frac{\pi}{2}[\delta(\omega+\omega_c) + \delta(\omega-\omega_c)] \quad ■$$

The autocorrelation is an even function of τ, that is,

$$r_{\mathbf{x}}(\tau) = r_{\mathbf{x}}(-\tau)$$

as can be easily shown, and $S_{\mathbf{x}}(\omega)$ is an even function of ω by definition. Equations (13.16) and (13.17) can thus be written as

$$S_{\mathbf{x}}(\omega) = \int_{-\infty}^{\infty} r_{\mathbf{x}}(\tau)\cos(\omega\tau)\, d\tau$$

$$r_{\mathbf{x}}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{\mathbf{x}}(\omega)\cos(\omega\tau)\, d\omega$$

i.e., $S_{\mathbf{x}}(\omega)$ is real. If $\omega = 0$, then

$$S_{\mathbf{x}}(0) = \int_{-\infty}^{\infty} r_{\mathbf{x}}(\tau)\, d\tau$$

i.e., *the total area under the autocorrelation function equals the PSD at zero frequency*. The average power of $\mathbf{x}(t)$ is given by

$$\text{Average power} = E\{\mathbf{x}^2(t)\} = r_{\mathbf{x}}(0) = \int_{-\infty}^{\infty} S_{\mathbf{x}}(\omega)\,\frac{d\omega}{2\pi}$$

as is to be expected.

A random process whose PSD is constant at all frequencies is said to be a *white-noise process*. If

$$S_{\mathbf{x}}(\omega) = K$$

we have

$$r_{\mathbf{x}}(\tau) = K\delta(\tau)$$

i.e., *the autocorrelation of a white-noise process is an impulse at the origin*.

13.8 DISCRETE-TIME RANDOM PROCESSES

The concept of a random process can be readily extended to discrete-time random signals by simply assigning discrete-time waveforms to the possible outcomes of an experiment. The *mean, mean square,* and *autocorrelation* of a discrete-time process $\mathbf{x}(nT)$ can be expressed as

$$\begin{aligned}
E\{\mathbf{x}(nT)\} &= \int_{-\infty}^{\infty} xp(x; nT)\,dx \\
E\{\mathbf{x}^2(nT)\} &= \int_{-\infty}^{\infty} x^2 p(x; nT)\,dx \\
r_{\mathbf{x}}(kT) &= E\{\mathbf{x}(nT)\mathbf{x}(nT + kT)\}
\end{aligned}$$

A *frequency-domain* representation for a discrete-time process can be deduced by using the interrelations between the z transform and the Fourier transform (see Sec. 6.5.1). We can write

$$\mathcal{Z}r_{\mathbf{x}}(kT) = \sum_{k=-\infty}^{\infty} r(kT)z^{-k} = R_{\mathbf{x}}(z)$$

and from Eq. (6.43c)

$$R_{\mathbf{x}}(e^{j\omega T}) = \sum_{k=-\infty}^{\infty} r(kT)e^{-j\omega k} = \mathcal{F}\hat{r}_{\mathbf{x}}(\tau) = \hat{S}_{\mathbf{x}}(\omega) \tag{13.18}$$

where

$$\hat{r}_{\mathbf{x}}(\tau) = E\{\hat{\mathbf{x}}(t)\hat{\mathbf{x}}(t+\tau)\}$$

$$\hat{\mathbf{x}}(t) = \sum_{n=-\infty}^{\infty} \mathbf{x}(nT)\delta(t-nT)$$

$$\tau = kT$$

(see Sec. 6.5). Therefore, from Eqs. (13.13) and (13.18)

$$R_{\mathbf{x}}(e^{j\omega T}) = \lim_{T_0\to\infty} E\left\{\frac{|\hat{X}_{T_0}(j\omega)|^2}{2T_0}\right\}$$

and

$$\hat{X}_{T_0}(j\omega) = \mathcal{F}\hat{\mathbf{x}}_{T_0}(t)$$

and

$$\hat{\mathbf{x}}_{T_0}(t) = \begin{cases} \hat{\mathbf{x}}(t) & \text{for } |t| \le T_0 \\ 0 & \text{otherwise} \end{cases}$$

In effect, the *z transform of the autocorrelation of discrete-time process* $\mathbf{x}(nT)$ *evaluated on the unit circle* $|z| = 1$ *is numerically equal to the PSD of the impulse-modulated process* $\hat{\mathbf{x}}(t)$. This quantity can be referred to as the PSD of discrete-time process $\mathbf{x}(nT)$ and can be represented by $S_{\mathbf{x}}(e^{j\omega T})$ by analogy with the PSD of continuous-time process $\mathbf{x}(t)$ which is represented by $S_{\mathbf{x}}(\omega)$. Consequently, we can write

$$\mathcal{Z}r_{\mathbf{x}}(kT) = S_{\mathbf{x}}(z)$$

where

$$r_{\mathbf{x}}(kT) = \frac{1}{2\pi j}\oint_{\Gamma} S_{\mathbf{x}}(z)z^{k-1}\,dz \tag{13.19a}$$

by virtue of Eq. (3.6). If $\mathbf{x}(t)$ were a voltage or current waveform, then $E\{\mathbf{x}^2(t)\}$ would represent the average energy that would be delivered in a 1-Ω resistor. Consequently, the quantity $E\{\mathbf{x}^2(nT)\}$ is said to be the *power* in $\mathbf{x}(nT)$. It can be obtained by evaluating the autocorrelation function at $k = 0$, that is,

$$E\{\mathbf{x}^2(nT)\} = r_{\mathbf{x}}(0) = \frac{1}{2\pi j}\oint_{\Gamma} S_{\mathbf{x}}(z)z^{-1}\,dz \tag{13.19b}$$

13.9 FILTERING OF DISCRETE-TIME RANDOM SIGNALS

If a discrete-time random signal is processed by a digital filter, we expect the PSD of the output signal to be related to that of the input signal. This indeed is the case, as will now be shown.

Consider a filter characterized by $H(z)$, and let $\mathbf{x}(n)$ and $\mathbf{y}(n)$ be the input and output processes, respectively. From the convolution summation (see Eq. (4.36b))

$$\mathbf{y}(i) = \sum_{p=-\infty}^{\infty} h(p)\mathbf{x}(i-p) \qquad \mathbf{y}(j) = \sum_{q=-\infty}^{\infty} h(q)\mathbf{x}(j-q)$$

and hence

$$E\{\mathbf{y}(i)\mathbf{y}(j)\} = E\left\{\sum_{q=-\infty}^{\infty}\sum_{p=-\infty}^{\infty} h(p)h(q)\mathbf{x}(i-p)\mathbf{x}(j-q)\right\}$$

With $j = i + k$ and $q = p + n$, we have

$$r_{\mathbf{y}}(k) = \sum_{n=-\infty}^{\infty}\sum_{p=-\infty}^{\infty} h(p)h(p+n)E\{\mathbf{x}(i-p)\mathbf{x}(i-p+k-n)\}$$

or

$$r_{\mathbf{y}}(k) = \sum_{n=-\infty}^{\infty} g(n)r_{\mathbf{x}}(k-n)$$

where

$$g(n) = \sum_{p=-\infty}^{\infty} h(p)h(p+n)$$

The use of the real-convolution theorem of the z transform (Theorem 3.7) gives

$$S_{\mathbf{y}}(z) = \mathcal{Z}r_{\mathbf{y}}(k) = \mathcal{Z}g(k)\mathcal{Z}r_{\mathbf{x}}(k) = G(z)S_{\mathbf{x}}(z) \tag{13.20}$$

Now

$$G(z) = \mathcal{Z}\sum_{p=-\infty}^{\infty} h(p)h(p+n) = \sum_{n=-\infty}^{\infty}\sum_{p=-\infty}^{\infty} h(p)h(p+n)z^{-n}$$

and with $n = k - p$

$$G(z) = \sum_{k=-\infty}^{\infty} h(k)z^{-k}\sum_{p=-\infty}^{\infty} h(p)(z^{-1})^{-p} = H(z)H(z^{-1}) \tag{13.21}$$

Therefore, from Eqs. (13.20) and (13.21) we get

$$S_{\mathrm{y}}(z) = H(z)H(z^{-1})S_{\mathbf{x}}(z) \tag{13.22}$$

or

$$S_{\mathbf{y}}(e^{j\omega T}) = |H(e^{j\omega T})|^2 S_{\mathbf{x}}(e^{j\omega T})$$

i.e., the *PSD of the output process is equal to the squared amplitude response of the filter times the PSD of the input process.*

Example 13.6 The output of a digital filter is given by

$$y(n) = x(n) + 0.8y(n-1)$$

The input of the filter is a random signal with zero mean and variance $\sigma_{\mathbf{x}}^2$; successive values of $x(n)$ are statistically independent. (*a*) Find the output PSD. (*b*) Obtain an expression for the average output power.

■ **Solution**

(*a*) The autocorrelation of the input signal is

$$r_{\mathbf{x}}(k) = E\{\mathbf{x}(n)\mathbf{x}(n+k)\}$$

For $k = 0$

$$r_{\mathbf{x}}(k) = E\{\mathbf{x}^2(n)\} = \sigma_{\mathbf{x}}^2$$

For $k \neq 0$, the use of Eq. (13.4) gives

$$r_{\mathbf{x}}(k) = E\{\mathbf{x}(n)\}E\{\mathbf{x}(n+k)\} = 0$$

Hence $$r_{\mathbf{x}}(k) = \sigma_{\mathbf{x}}^2\delta(k) \qquad \text{and} \qquad S_{\mathbf{x}}(z) = \sigma_{\mathbf{x}}^2$$

Now from Eq. (13.22)

$$S_{\mathbf{y}}(z) = \sigma_{\mathbf{x}}^2 H(z)H(z^{-1}) \quad \blacksquare$$

where $$H(z) = \frac{z}{z-0.8}$$

(*b*) From Eq. (13.19b)

$$\text{Output power} = E\{\mathbf{y}^2(n)\} = r_{\mathbf{y}}(0) = \frac{1}{2\pi j}\oint_{\Gamma} \sigma_{\mathbf{x}}^2 H(z)H(z^{-1})z^{-1}\,dz$$

and if Γ is taken to be the unit circle $|z| = 1$ we can let $z = e^{j\omega T}$, in which case

$$\text{Output power} = \frac{1}{\omega_s}\int_0^{\omega_s} \sigma_{\mathbf{x}}^2 H(e^{j\omega T})H(e^{-j\omega T})\,d\omega \quad \blacksquare$$

A simple numerical method for the evaluation of the output power can be found in Ref. [6].

REFERENCES

[1] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York: McGraw-Hill, 1991.

[2] W. B. Davenport, Jr., and W. L. Root, *Random Signals and Noise*, New York: McGraw-Hill, 1958.

[3] B. P. Lathi, *An Introduction to Random Signals and Communication Theory*, Scranton: International Textbook, 1968.

[4] G. R. Cooper and C. D. McGillem, *Probabilistic Methods of Signal and System Analysis*, New York: Holt, Reinhart and Winston, 1971.

[5] H. Stark and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 2002.

[6] K. J. Åström, E. I. Jury, and R. G. Agniel, "A numerical method for the evaluation of complex integrals," *IEEE Trans. Automatic Control*, vol. 15, pp. 468–471, Aug. 1970.

PROBLEMS

13.1. A random variable **x** has a probability-density function

$$p_{\mathbf{x}}(x) = \begin{cases} Ke^{-x} & \text{for } 1 \le x \le \infty \\ 0 & \text{otherwise} \end{cases}$$

(*a*) Find K.

(*b*) Find Pr $[0 \le \mathbf{x} \le 2]$.

13.2. A random variable **x** has a probability-density function

$$p_{\mathbf{x}}(x) = \begin{cases} \dfrac{1}{q} & \text{for } 0 \le x \le q \\ 0 & \text{otherwise} \end{cases}$$

Find its mean, mean square, and variance.

13.3. Find the mean, mean square, and variance for the random variable of Prob. 13.1.

13.4. Demonstrate the validity of Eq. (13.7).

13.5. A Gaussian random variable **x** has a mean η and a variance σ^2. Show that

$$P_{\mathbf{x}}(x_1 - \eta) = 1 - P_{\mathbf{x}}(\eta - x_1)$$

where $P_{\mathbf{x}}(x)$ is the probability-distribution function of a Gaussian random variable with zero mean.

13.6. A Gaussian random variable **x** has $\eta = 0$ and $\sigma = 2$.

(*a*) Find Pr $[\mathbf{x} \ge 2]$.

(*b*) Find Pr $[|\mathbf{x}| \ge 2]$.

(*c*) Find x_1 if Pr $[|\mathbf{x}| \le x_1] = 0.95$.

13.7. The random variable of Prob. 13.5 satisfies the relations

$$\Pr[\mathbf{x} \le 60] = 0.2 \qquad \Pr[\mathbf{x} \ge 90] = 0.1$$

Find η and σ^2.

13.8. A random variable **x** has a Rayleigh probability-density function given by

$$p_{\mathbf{x}}(x) = \begin{cases} \dfrac{xe^{-x^2/2\alpha^2}}{\alpha^2} & \text{for } 0 \le x \le \infty \\ 0 & \text{otherwise} \end{cases}$$

Show that

(a) $E\{\mathbf{x}\} = \alpha\sqrt{\dfrac{\pi}{2}}$

(b) $E\{\mathbf{x}^2\} = 2\alpha^2$

(c) $\sigma_{\mathbf{x}}^2 = \left(2 - \dfrac{\pi}{2}\right)\alpha^2$

13.9. A random process is given by

$$\mathbf{x}(t) = \mathbf{y}e^{-t}u(t - \mathbf{z})$$

where **y** and **z** are random variables uniformly distributed in the range $(-1, 1)$. Sketch five sample functions.

13.10. A random process is given by

$$\mathbf{x}(t) = 2 + \frac{\mathbf{y}t}{\sqrt{2}}$$

where **y** is a random variable with a probability-density function

$$p_{\mathbf{y}}(y) = \frac{1}{\sqrt{2\pi}}e^{-y^2/2} \qquad \text{for } -\infty \le y \le \infty$$

Find the first-order probability-density function of $\mathbf{x}(t)$.

13.11. A random process is given by

$$\mathbf{x}(t) = \mathbf{z}\cos(\omega_0 t + \mathbf{y})$$

Find the first-order probability-density function of $\mathbf{x}(t)$.

(a) If **z** is a random variable distributed uniformly in the range $(-1, 1)$ and **y** is a constant.

(b) If **y** is a random variable distributed uniformly in the range $(-\pi, \pi)$ and **z** is a constant.

13.12. Find the mean, mean square, and autocorrelation for the process in Prob. 13.10. Is the process stationary?

13.13. Repeat Prob. 13.12 for the processes in Prob. 13.11.

13.14. A stationary discrete-time random process is given by

$$\mathbf{x}(nT) = E\{\mathbf{x}(nT)\} + \mathbf{x}_0(nT)$$

where $\mathbf{x}_0(nT)$ is a zero-mean process. Show that

(a) $r_{\mathbf{x}}(0) = E\{\mathbf{x}^2(nT)\}$

(b) $r_{\mathbf{x}}(-kT) = r_{\mathbf{x}}(kT)$

(c) $r_{\mathbf{x}}(0) \ge |r_{\mathbf{x}}(kT)|$

(d) $r_{\mathbf{x}}(kT) = [E\{\mathbf{x}(nT)\}]^2 + r_{\mathbf{x}_0}(kT)$

13.15. Explain the physical significance of

(a) $E\{\mathbf{x}(nT)\}$

(b) $E^2\{\mathbf{x}(nT)\}$

(c) $E\{\mathbf{x}^2(nT)\}$

(d) $\sigma_{\mathbf{x}}^2 = E\{\mathbf{x}^2(nT)\} - [E\{\mathbf{x}(nT)\}]^2$

13.16. A discrete-time random process is given by

$$\mathbf{x}(nT) = 3 + 4nT\mathbf{y}$$

where **y** is a random variable with a probability-density function

$$p_{\mathbf{y}}(y) = \frac{1}{2\sqrt{2\pi}} e^{-(y-4)^2/8} \qquad \text{for } -\infty \leq y \leq \infty$$

Find its mean, mean square, and autocorrelation.

13.17. A discrete-time random process is given by

$$\mathbf{x}(nT) = \mathbf{z}\cos\left(\omega_0 nT + \frac{\pi}{8}\right)$$

where **z** is a random variable distributed uniformly in the range (0, 1). Find the mean, mean square, and autocorrelation of $\mathbf{x}(nT)$. Is the process stationary?

13.18. A discrete-time random process is given by

$$\mathbf{x}(nT) = \sqrt{2}\cos(\omega_0 nT + \mathbf{y})$$

where **y** is a random variable uniformly distributed in the range $(-\pi, \pi)$.

(*a*) Find the mean, mean square, and autocorrelation of $\mathbf{x}(nT)$.

(*b*) Show that the process is wide-sense stationary.

(*c*) Find the PSD of $\mathbf{x}(nT)$.

13.19. The random process of Prob. 13.18 is processed by a digital filter characterized by

$$H(e^{j\omega T}) = \begin{cases} 1 & \text{for } |\omega| \leq \omega_c \\ 0 & \text{otherwise} \end{cases}$$

Sketch the input and output power-density spectrums if $\omega_0 \leq \omega_c$.

13.20. A random process $\mathbf{x}(nT)$ with a probability-density function

$$p_{\mathbf{x}}(x; nT) = \begin{cases} 1 & \text{for } \frac{1}{2} \leq x \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

is applied at the input of the filter depicted in Fig. P13.20. Find the output PSD if $\mathbf{x}(nT)$ and $\mathbf{x}(kT)$ $(n \neq k)$ are statistically independent.

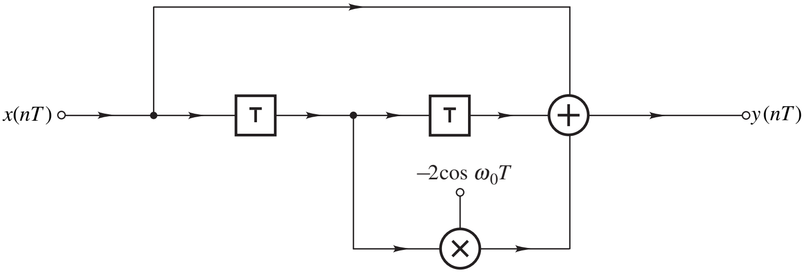


Figure P13.20

This page intentionally left blank

CHAPTER 14

EFFECTS OF FINITE WORD LENGTH IN DIGITAL FILTERS

14.1 INTRODUCTION

In software as well as hardware digital-filter implementations, numbers are stored in finite-length registers. Consequently, if coefficients and signal values cannot be accommodated in the available registers, they must be quantized before they can be stored. Number quantization gives rise to three types of errors:

1. Coefficient-quantization errors
2. Product-quantization errors
3. Input-quantization errors

The transfer-function coefficients are normally evaluated to a high degree of precision during the approximation step. If *coefficient quantization* is applied, the frequency response of the resulting filter may differ appreciably from the desired response, and if the quantization step is coarse, the filter may actually fail to meet the desired specifications.

Product-quantization errors arise at the outputs of multipliers. Each time a signal represented by b_1 digits is multiplied by a coefficient represented by b_2 digits, a product having as many as b_1+b_2 digits is generated. Since a uniform register length must, in practice, be used throughout the filter, each multiplier output must be rounded or truncated before processing can continue. These errors tend to propagate through the filter and give rise to output noise commonly referred to as output *roundoff noise*.

Input-quantization errors arise in applications where digital filters are used to process continuous-time signals. These are the errors inherent in the analog-to-digital conversion process (see Sec. 6.9).

This chapter begins with a review of the various number systems and types of arithmetic that can be used in digital-filter implementations. It then describes various methods of analysis and design that can be applied to quantify and minimize the effects of quantization. Section 14.3 deals with a method of analysis that can be used to evaluate the effect of coefficient quantization and Sec. 14.4 describes two families of filter structures that are relatively insensitive to coefficient quantization. Section 14.5 deals with methods by which roundoff noise caused by product quantization can be evaluated, and Secs. 14.6–14.8 describe methods by which roundoff noise can be reduced or minimized. In Sec. 14.9, two types of parasitic oscillations known as *quantization* and *overflow limit cycles* are considered in some detail and methods for their elimination are described.

14.2 NUMBER REPRESENTATION

The hardware implementation of digital filters, like the implementation of other digital hardware, is based on the binary-number representation.

14.2.1 Binary System

In general, any number N can be expressed as

$$N = \sum_{i=-m}^{n} b_i r^i \tag{14.1}$$

where

$$0 \le b_i \le r - 1$$

If distinct symbols are assigned to the permissible values of b_i, the number N can be represented by the notation

$$N = (b_n b_{n-1} \cdots b_0 . b_{-1} \cdots b_{-m})_r \tag{14.2}$$

The parameter r is said to be the *radix* of the representation, and the point separating N into two parts is called the *radix point*.

If $r = 10$, Eq. (14.2) becomes the decimal representation of N and the radix point is the decimal point. Similarly, if $r = 2$ Eq. (14.2) becomes the binary representation of N and the radix point is referred to as the *binary point*. The common symbols used to represent the two permissible values of b_i are 0 and 1. These are called *bits*.

A mixed decimal number can be converted into a binary number through the following steps:

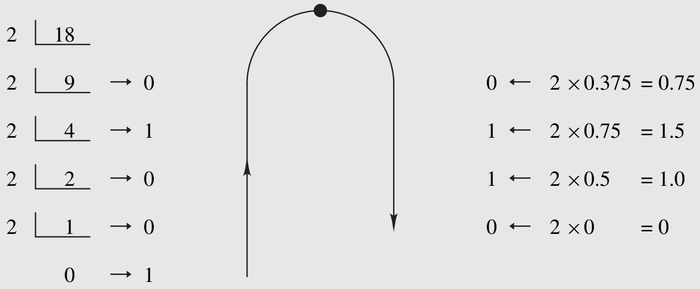
1. Divide the integer part by 2 repeatedly and arrange the resulting remainders in the reverse order.
2. Multiply the fraction part by 2 and remove the resulting integer part; repeat as many times as necessary, and then arrange the integers obtained in the forward order.

A binary number can be converted into a decimal number by using Eq. (14.1).

Example 14.1 (*a*) Form the binary representation of $N = 18.375_{10}$. (*b*) Form the decimal representation of $N = 11.101_2$.

■ **Solution**

(*a*) The binary representation can be carried out as follows:



Hence, we get

$$18.375_{10} = 10010.011_2 \quad ■$$

(*b*) From Eq. (14.1)

$$11.101_2 = 1(2^1) + 1(2^0) + 1(2^{-1}) + 0(2^{-2}) + 1(2^{-3}) = 3.625_{10} \quad ■$$

The most basic electronic memory device is the *flip-flop* which can be either in a low or a high state. By assigning a 0 to the low state and a 1 to the high state, a single-bit binary number can be stored. By arranging n flip-flops in juxtaposition, as in Fig. 14.1*a*, a register can be formed that will store an n-bit number.

A rudimentary 4-bit digital-filter implementation is shown in Fig. 14.1*b*. Registers R_y and R_p are used to store the past output $y(n-1)$ and the multiplier coefficient p, respectively. The output of the multiplier at steady state is $by(n-1)$. Once a new input sample is received, the adder goes into action to form the new output $y(n)$, which is then used to update register R_y. Subsequently, the multiplier is triggered into operation and the product $by(n-1)$ is formed. The cycle is repeated when a new input sample is received.

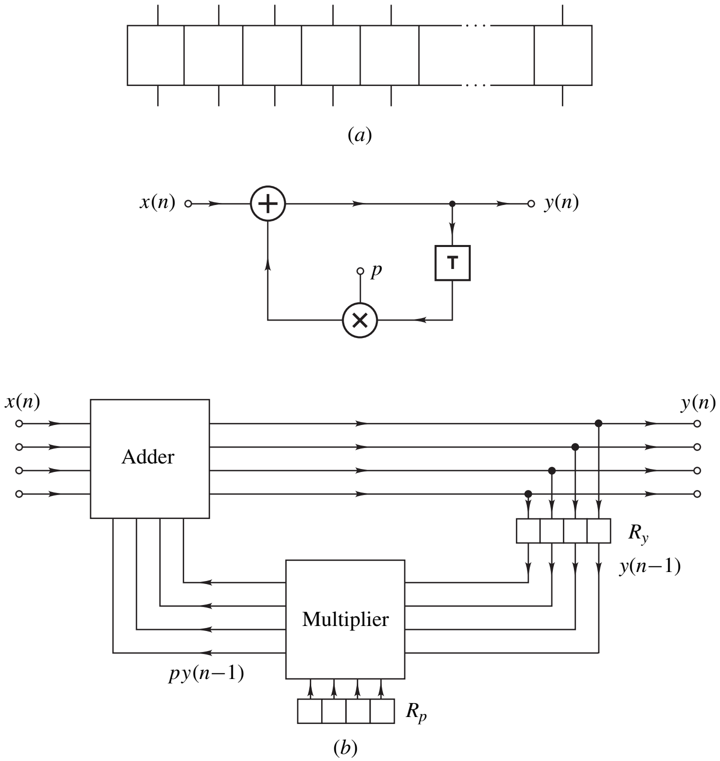


Figure 14.1 (*a*) Register, (*b*) rudimentary digital-filter implementation.

A filter implementation like that in Fig. 14.1*b* can assume many forms, depending on the type of machine arithmetic used. The arithmetic can be of the *fixed-point* or *floating-point* type and in each case various conventions can be used for the representation of negative numbers. The two types of arithmetic differ in the way numbers are stored in registers and in the way by which they are manipulated by the digital hardware.

14.2.2 Fixed-Point Arithmetic

In fixed-point arithmetic, the numbers are usually assumed to be proper fractions. Integers and mixed numbers are avoided because (1) the number of bits representing an integer cannot be reduced by rounding or truncation without destroying the number and (2) mixed numbers are more difficult to multiply. For these reasons, the binary point is usually set between the first and second bit positions in the register, as depicted in Fig. 14.2*a*. The first position is reserved for the sign of the number.

Depending on the representation of negative numbers, fixed-point arithmetic can assume three forms:

1. Signed magnitude
2. One's complement
3. Two's complement

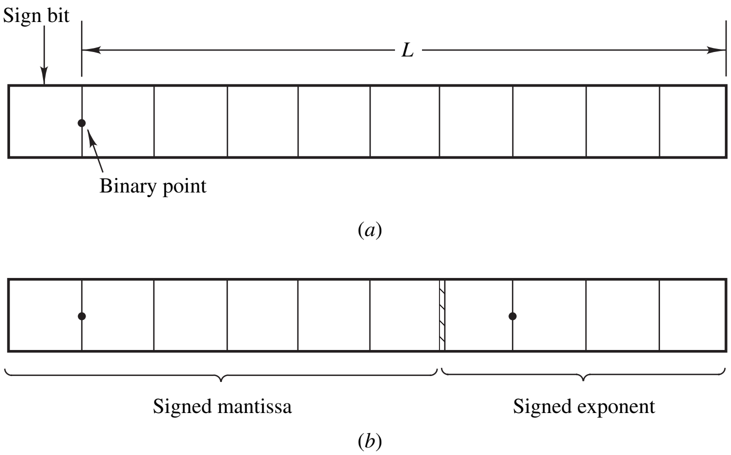


Figure 14.2 Storage of (*a*) fixed-point numbers, (*b*) floating-point numbers.

In the *signed-magnitude* arithmetic a fractional number

$$N = \pm 0.b_{-1}b_{-2}\cdots b_{-m}$$

is represented as

$$N_{sm} = \begin{cases} 0.b_{-1}b_{-2}\cdots b_{-m} & \text{for } N \geq 0 \\ 1.b_{-1}b_{-2}\cdots b_{-m} & \text{for } N \leq 0 \end{cases}$$

The most significant bit is said to be the *sign bit*; e.g., if $N = +0.1101$ or -0.1001, then $N_{sm} = 0.1101$ or 1.1001.

The *one's-complement* representation of a number N is defined as

$$N_1 = \begin{cases} N & \text{for } N \geq 0 \\ 2 - 2^{-L} - |N| & \text{for } N \leq 0 \end{cases} \tag{14.3}$$

where L, referred to as the *word length*, is the number of bit locations in the register to the right of the binary point. The binary form of $2-2^{-L}$ is a string of 1s filling the $L+1$ locations of the register. Thus, the one's complement of a negative number can be deduced by representing the number by $L+1$ bits, including zeros if necessary, and then complementing (changing 0s into 1s and 1s into 0s) all bits; e.g., if $N = -0.11010$, then $N_1 = 1.00101$ for $L = 5$ and $N_1 = 1.00101111$ for $L = 8$.

The *two's-complement* representation is similar. We now have

$$N_2 = \begin{cases} N & \text{for } N \geq 0 \\ 2 - |N| & \text{for } N < 0 \end{cases}$$

The two's complement of a negative number can be formed by adding 1 at the least significant position of the one's complement. Similarly, a negative number can be recovered from its two's complement by complementing and then adding 1 at the least significant position.

The possible numbers that can be stored in a 4-bit register together with their decimal equivalents are listed in Table 14.1. Some peculiarities of the three systems are evident. The signed-magnitude and the one's-complement systems have two representations for zero whereas the two's-complement system has only one. On the other hand, −1 is represented in the two's-complement system but not in the other two.

The merits and demerits of the three types of arithmetic can be envisaged by examining how arithmetic operations are performed in each case.

One's-complement addition of any two numbers is carried out by simply adding their one's complements bit by bit. A carry bit at the most significant position, if one is generated, is added at the least significant position (*end-around carry*). Two's-complement addition is exactly the same except that a carry bit at the most significant position is ignored. Signed-magnitude addition, on the other hand, is much more complicated as it involves sign checks as well as complementing and end-around carry [1].

In the one's- or two's-complement arithmetic, direct multiplication of the complements does not always yield the product, and as a consequence special algorithms must be employed. By contrast, signed-magnitude multiplication is accomplished by simply multiplying the magnitudes of the two numbers bit by bit and then adjusting the sign bit of the product.

Table 14.1 Decimal equivalents of numbers 0.000 to 1.111

Binary number	Decimal equivalent (eighths)		
	Signed magnitude	**One's complement**	**Two's complement**
0.000	0	0	0
0.001	1	1	1
0.010	2	2	2
0.011	3	3	3
0.100	4	4	4
0.101	5	5	5
0.110	6	6	6
0.111	7	7	7
1.000	−0	−7	−8
1.001	−1	−6	−7
1.010	−2	−5	−6
1.011	−3	−4	−5
1.100	−4	−3	−4
1.101	−5	−2	−3
1.110	−6	−1	−2
1.111	−7	−0	−1

Example 14.2 Form the sum 0.53125 + (−0.40625) using the one's- and two's-complement additions assuming a word length of 5 bits.

■ **Solution**

$$0.53125_{10} = 0.10001_2$$

$$0.40625_{10} = 0.01101_2$$

	One's complement	**Two's complement**
0.53125	0.10001	0.10001
−0.40625	1.10010	1.10011
0.12500	↓ 0.00011	1 ← 0.00100
	⟶ 1	
	0.00100	

An important feature of the one's- or two's-complement addition is that a machine-representable sum $S = n_1 + n_2 + \cdots + n_i + \cdots$ will always be evaluated correctly, even if overflow does occur in the evaluation of partial sums.

Example 14.3 Form the sum $\frac{7}{8} + \frac{4}{8} + (-\frac{6}{8})$ using the two's-complement addition. Assume $L = 3$.

■ **Solution**

From Table 14.1

$$\begin{array}{rrrrl} & 7/8 & & 0.111 & \\ + & 4/8 & & 0.100 & \\ \hline & 11/8 & & 1.011 & \text{incorrect partial sum} \\ - & 6/8 & + & 1.010 & \\ \hline & 5/8 & & 0.101 & \text{correct sum} \quad \blacksquare \end{array}$$

14.2.3 Floating-Point Arithmetic

There are two basic disadvantages in a fixed-point arithmetic: (1) The range of numbers that can be handled is small; e.g., in the two's-complement representation the smallest number is −1 and the largest is $1 - 2^{-L}$. (2) The percentage error produced by truncation or rounding tends to increase as the magnitude of the number is decreased. For example, if numbers 0.11011010 and 0.000110101 are both truncated such that only 4 bits are retained to the right of the binary point, the respective errors will be 4.59 and 39.6 percent.

These problems can be alleviated to a large extent by using a floating-point arithmetic. In this type of arithmetic, a number N is expressed as

$$N = M \times 2^e \tag{14.4}$$

where e is an integer and

$$\frac{1}{2} \le M < 1$$

M and e are referred to as the *mantissa* and *exponent*, respectively. For example, numbers 0.00110101 and 1001.11 are represented by 0.110101×2^{-2} and 0.100111×2^4, respectively. Negative numbers are handled in the same way as in fixed-point arithmetic.

Floating-point numbers are stored in registers, as depicted in Fig. 14.2*b*. The register is subdivided into two segments, one for the signed mantissa and one for the signed exponent.

Floating-point addition is carried out by shifting the mantissa of the smaller number to the right and increasing the exponent until the exponents of the two numbers are equal. The mantissas are then added to form the sum, which is subsequently put back into the normalized representation of Eq. (14.4). Multiplication is accomplished by multiplying mantissas, adding exponents, and then readjusting the product.

It should be mentioned that the above scheme is just one of many. Nowadays most computers and digital hardware are designed to work with the IEEE floating-point representation format known as the ANSI/IEEE Standard 754-1985, which is, in effect, a variation of the above scheme. In the IEEE format, an arbitrary number N is represented as

$$N = S \times 2^{e+bias}$$

where S is a signed mixed number in the range 1 to $2 - 2^{-L}$ where L is the number of significant bits allowed in the fractional part of S and $bias$ is an integer. S is commonly referred to as the *significand* [1].

The two obvious differences between the IEEE representation and the generic one in Eq. (14.4) is that, first, S has an integer part and, second, a biased representation is used for the exponent.

Normalizing numbers to the range 1 to $2 - 2^{-L}$, that is, $1.000\cdots000$ to $1.111\cdots111$, would cause the integer part of all numbers to be unity and, as a consequence, it would not need to be stored. Thus number precision can be increased by one bit without increasing the cost of hardware. The missing unity can be assumed by the software or hardware implementation.

The use of a biased exponent eliminates the need to deal with negative numbers, which simplifies the arithmetic a great deal, but it introduces another problem. If the exponent in some application is allowed to assume values, say, between -256 to $+256$, a bias of 256 can be added to all exponents to achieve an exponent range of 0 to 512. In such a scheme, exponents e_1 and e_2 would be represented by $e_1 + bias$ and $e_2 + bias$. In the case of floating-point multiplication, the exponents would need to be added in which case a sum of $e_1 + e_2 + 2bias$ would be produced. Evidently, in order to achieve a ‘biased’ sum, an amount *bias* would need to be subtracted from the sum. In the case of floating-point division, an exponent e_2 would need to be subtracted from an exponent e_1 and, consequently, a difference $e_1 - e_2$ would be produced. In effect, in the case of subtraction, an amount *bias* would need to be added to get a biased representation. Multiplication and division of biased numbers would, of course, need much more attention, as can be readily verified, but, fortunately, these operations are not needed when dealing with the exponents in floating-point arithmetic.

Summarizing, the use of a biased representation eliminates the need to deal with negative exponents but it introduces the problem of keeping taps on the flow of the bias and the question remains as to whether a biased representation is more cost effective. It turns out that if a bias which is a power of two minus 1 is used, i.e., $bias = 2^{L_e} - 1$ where L_e is an integer, correcting for the bias is greatly simplified and the IEEE format becomes the floating-point representation of choice. Further details about the IEEE format as well as more information on computer arithmetic in general can be found in Parhami [1].

Floating-point arithmetic leads to increased dynamic range and improved precision of processing. Unfortunately, it also leads to increased cost of hardware (or more complicated software) and to reduced speed of processing. The reason is that more software/hardware is needed since both the significand (or mantissa) and exponent have to be manipulated. For software non-real-time implementations on general purpose digital computers, floating-point arithmetic is always preferred since neither the cost of hardware nor the speed of processing is a significant factor.

14.2.4 Number Quantization

Once the register length in a fixed-point implementation is assigned, the set of machine representable numbers is fixed. If the word length is L bits (excluding the sign bit), the smallest number variation that can be represented is a 1 at the least significant register position, which corresponds to 2^{-L}. Therefore, any number consisting of B bits (excluding the sign bit), where $B > L$, must be quantized. This can be accomplished (1) by *truncating* all bits that cannot be accommodated in the register, and (2) by *rounding* the number to the nearest machine-representable number.

Obviously, if a number x is quantized, an error ε will be introduced given by

$$\varepsilon = x - Q[x] \tag{14.5}$$

where $Q[x]$ denotes the quantized value of x. The range of ε tends to depend on the type of number representation and also on the type of quantization. Let us examine the various possibilities, starting with truncation.

As can be seen in Table 14.1, the representation of positive numbers is identical in all three fixed-point representations. Since truncation can only reduce a positive number, ε is positive. Its maximum value occurs when all disregarded bits are 1s, in which case

$$0 \le \varepsilon_T \le 2^{-L} - 2^{-B} \qquad \text{for } x \ge 0$$

For negative numbers the three representations must be considered individually. For the signed-magnitude representation, truncation will decrease the magnitude of the number or increase its signed value, and hence $Q[x] > x$ or

$$-(2^{-L} - 2^{-B}) \le \varepsilon_T \le 0 \qquad \text{for } x < 0$$

The one's-complement representation of a negative number

$$x = -\sum_{i=1}^{B} b_{-i}2^{-i} \tag{14.6}$$

(where $b_{-i} = 0$ or 1) is obtained from Eq. (14.3) as

$$x_1 = 2 - 2^{-L} - \sum_{i=1}^{B} b_{-i}2^{-i}$$

If all the disregarded bits are 0s, obviously $\varepsilon = 0$. At the other extreme if all the disregarded bits are 1s, we have

$$Q[x_1] = 2 - 2^{-L} - \sum_{i=1}^{B} b_{-i}2^{-i} - (2^{-L} - 2^{-B})$$

Consequently, the decimal equivalent of $Q[x_1]$ is

$$Q[x] = -\left[\sum_{i=1}^{B} b_{-i}2^{-i} + (2^{-L} - 2^{-B})\right] \tag{14.7}$$

and, therefore, from Eqs. (14.5)–(14.7)

$$0 \le \varepsilon_T \le 2^{-L} - 2^{-B} \qquad \text{for } x < 0$$

The same inequality holds for two's-complement numbers, as can easily be shown. In summary, for signed-magnitude numbers

$$-q < \varepsilon_T < q$$

where $q = 2^{-L}$ is the quantization step, whereas for one's- or two's-complement numbers

$$0 \le \varepsilon_T < q$$

Evidently, quantization errors can be kept as low as desired by using a sufficiently large value of L.

For rounding, the quantization error can be positive as well as negative by definition, and its maximum value is $q/2$. If numbers lying halfway between quantization levels are rounded up, we have

$$-\frac{q}{2} \le \varepsilon_R < \frac{q}{2} \tag{14.8}$$

Rounding can be effected, in practice, by adding 1 at position $L + 1$ and then truncating the number to L bits.

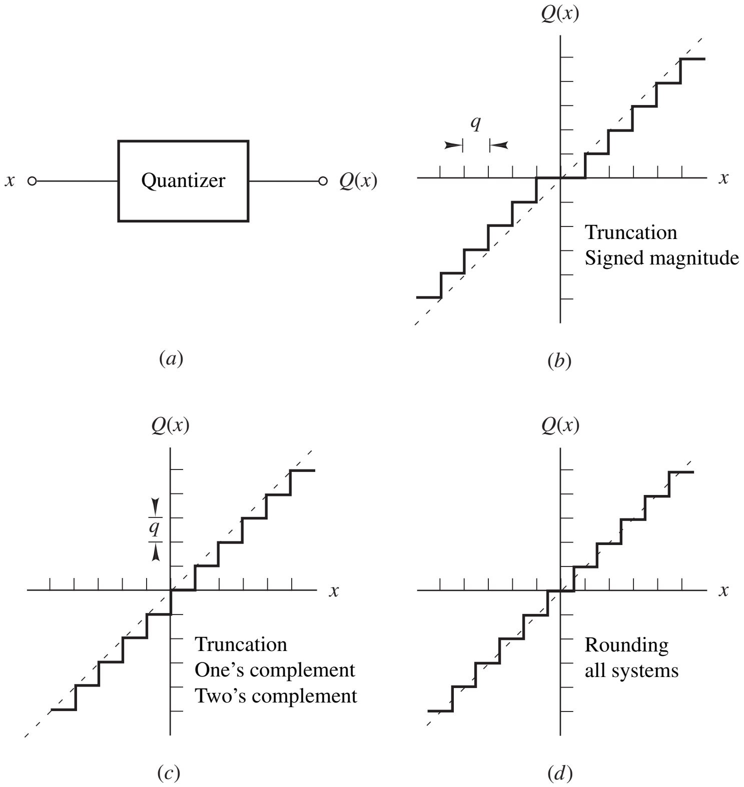


Figure 14.3 Number quantization: (*a*) Quantizer, (*b*) to (*d*) $Q(x)$ versus x.

A convenient way of visualizing the process of quantization is to imagine a quantizer with input x and output $Q[x]$. Depending on the type of quantization, the transfer characteristic of the device can assume one of the forms illustrated in Fig. 14.3.

The range of quantization error in floating-point arithmetic can be evaluated by using a similar approach.

14.3 COEFFICIENT QUANTIZATION

Coefficient-quantization errors introduce perturbations in the zeros and poles of the transfer function, which in turn manifest themselves as errors in the frequency response. Product-quantization errors, on the other hand, can be regarded as noise sources that give rise to output roundoff noise. Since the importance of the two types of errors can vary considerably from application to application, it is frequently advantageous to use different word lengths for the coefficient and signal values. The coefficient word length can be chosen to satisfy prescribed frequency-response specifications, whereas the signal word length can be chosen to satisfy a signal-to-noise ratio specification.

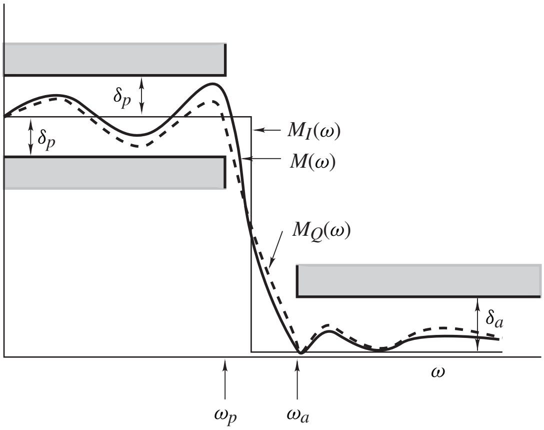


Figure 14.4 Coefficient quantization.

Consider a digital filter characterized by $H(z)$ and let

$$M(\omega) = |H(e^{j\omega T})| = \text{amplitude response without quantization}$$

$$M_Q(\omega) = \text{amplitude response with quantization}$$

$$M_I(\omega) = \text{ideal amplitude response}$$

$$\delta_p\ (\delta_a) = \text{passband (stopband) tolerance on amplitude response}$$

These quantities are illustrated in Fig. 14.4.

The effect of coefficient quantization is to introduce an error ΔM in $M(\omega)$ given by

$$\Delta M = M(\omega) - M_Q(\omega)$$

The maximum permissible value of $|\Delta M|$, denoted by $\Delta M_{\max}(\omega)$, can be deduced from Fig. 14.4 as

$$\Delta M_{\max}(\omega) = \begin{cases} \delta_p - |M(\omega) - M_I(\omega)| & \text{for } \omega \leq \omega_p \\ \delta_a - |M(\omega) - M_I(\omega)| & \text{for } \omega \geq \omega_a \end{cases}$$

and if

$$|\Delta M| \leq \Delta M_{\max}(\omega) \tag{14.9}$$

for $0 \leq \omega \leq \omega_p$ and $\omega_a \leq \omega \leq \omega_s/2$, the desired specification will be met. The *optimum* word length can thus be determined exactly by evaluating $|\Delta M|$ as a function of frequency for successively larger values of the word length until Eq. (14.9) is satisfied. Evidently, this is a trial-and-error approach and may entail considerable computation.

An alternative approach is to employ a statistical method proposed by Avenhaus [2] and later modified by Crochiere [3]. This method yields a fairly accurate estimate of the required word length and is, in general, more efficient than the exact method described. Its details follow.

Consider a fixed-point implementation and assume that quantization is carried out by rounding. From Eq. (14.8) the error in coefficient c_i ($i = 1, 2, \dots, m$), denoted as Δc_i, can assume any value in the range $-q/2$ to $+q/2$; that is, Δc_i is a random variable. If the probability density of Δc_i is assumed to be uniform, that is,

$$p(\Delta c_i) = \begin{cases} \frac{1}{q} & \text{for } -\frac{q}{2} \le \Delta c_i \le \frac{q}{2} \\ 0 & \text{otherwise} \end{cases}$$

then from Eqs. (13.8) and (13.10)

$$E\{\Delta c_i\} = 0 \tag{14.10}$$

$$\sigma^2_{\Delta c_i} = \frac{q^2}{12} \tag{14.11}$$

The variation ΔM in $M(\omega)$ is also a random variable. By virtue of Taylor's theorem we can write

$$\Delta M = \sum_{i=1}^{m} \Delta c_i S_{c_i}^M$$

where the quantity

$$S_{c_i}^M = \frac{\partial M(\omega)}{\partial c_i}$$

is known as the *sensitivity of the amplitude response $M(\omega)$ with respect to variations in coefficient c_i*. Evidently,

$$E\{\Delta M\} = \sum_{i=1}^{m} S_{c_i}^M E\{\Delta c_i\} = 0$$

according to Eq. (14.10). If Δc_i and Δc_j ($i \ne j$) are assumed to be statistically independent, then from Eq. (13.7)

$$\sigma^2_{\Delta M} = \sum_{i=1}^{m} \sigma^2_{\Delta c_i} \left(S_{c_i}^M\right)^2$$

and, therefore, from Eq. (14.11)

$$\sigma^2_{\Delta M} = \frac{q^2 S_T^2}{12} \tag{14.12}$$

where

$$S_T^2 = \sum_{i=1}^{m} \left(S_{c_i}^M\right)^2 \tag{14.13}$$

For a large value of m, ΔM is approximately Gaussian by virtue of the *central-limit theorem* [4], and since $E\{\Delta M\} = 0$, Eq. (13.1) gives

$$p(\Delta M) = \frac{1}{\sigma_{\Delta M}\sqrt{2\pi}} e^{-\Delta M^2/2\sigma_{\Delta M}^2} \qquad \text{for } -\infty \le \Delta M \le \infty$$

Consequently, ΔM will be in some range $-\Delta M_1 \le \Delta M \le \Delta M_1$ with a probability y given by

$$y = \Pr\left[|\Delta M| \le \Delta M_1\right] = \frac{2}{\sigma_{\Delta M}\sqrt{2\pi}} \int_0^{\Delta M_1} e^{-\Delta M^2/2\sigma_{\Delta M}^2}\, d(\Delta M) \tag{14.14}$$

With the variable transformation

$$\Delta M = x\sigma_{\Delta M} \qquad \Delta M_1 = x_1\sigma_{\Delta M} \tag{14.15}$$

Equation (14.14) can be put in the standard form

$$y = \frac{2}{\sqrt{2\pi}} \int_0^{x_1} e^{-x^2/2}\, dx$$

Once an acceptable confidence factor y is selected, the corresponding value of x_1 can be obtained from published tables or by using a numerical method. The quantity ΔM_1 is essentially a statistical bound on ΔM, and if the word length is chosen such that

$$\Delta M_1 \le \Delta M_{\max}(\omega) \tag{14.16}$$

the desired specifications will be satisfied to within a confidence factor y. The resulting word length can be referred to as the *statistical word length*. A statistical bound on the quantization step can be deduced from Eqs. (14.12), (14.15), and (14.16) as

$$q \le \frac{\sqrt{12}\Delta M_{\max}(\omega)}{x_1 S_T} \tag{14.17}$$

The register length should be sufficiently large to accommodate the quantized value of the largest coefficient; so let

$$Q[\max c_i] = \sum_{i=-K}^{J} b_i 2^i$$

where b_J and $b_{-K} \ne 0$. The required word length must be

$$L = 1 + J + K \tag{14.18}$$

and since $q = 2^{-K}$ or

$$K = \log_2 \frac{1}{q} \tag{14.19}$$

Eqs. (14.17)–(14.19) now give the desired result as

$$L \geq L(\omega) = 1 + J + \log_2 \frac{x_1 S_T}{\sqrt{12}\Delta M_{\max}(\omega)}$$

A reasonable agreement between the statistical and exact word lengths is achieved by using $x_1 = 2$ [3, 5]. This value of x_1 corresponds to a confidence factor of 0.95.

The amplitude-response sensitivities $S_{c_i}^M$ in Eq. (14.13) can be efficiently computed as follows. The sensitivity of the frequency response with respect to a multiplier coefficient c can be expressed as

$$S_c^H(e^{j\omega T}) = \frac{\partial H(e^{j\omega T})}{\partial c} = \Re e\left[S_c^H(e^{j\omega T})\right] + j\Im m\left[S_c^H(e^{j\omega T})\right]$$

and if

$$H(e^{j\omega T}) = M(\omega)e^{j\theta(\omega)}$$

we can show that

$$\begin{aligned}\Re e\left[S_c^H(e^{j\omega T})\right] &= [\cos\theta(\omega)]\frac{\partial M(\omega)}{\partial c} - M(\omega)[\sin\theta(\omega)]\frac{\partial\theta(\omega)}{\partial c}\\ \Im m\left[S_c^H(e^{j\omega T})\right] &= [\sin\theta(\omega)]\frac{\partial M(\omega)}{\partial c} + M(\omega)[\cos\theta(\omega)]\frac{\partial\theta(\omega)}{\partial c}\end{aligned}$$

Therefore,

$$S_c^M = \frac{\partial M(\omega)}{\partial c} = [\cos\theta(\omega)]\ \Re e\, S_c^H(e^{j\omega T}) + [\sin\theta(\omega)]\ \Im m\, S_c^H(e^{j\omega T})$$

and

$$S_c^\theta = \frac{\partial\theta(\omega)}{\partial c} = \frac{1}{M(\omega)}\{[\cos\theta(\omega)]\ \Im m\, S_c^H(e^{j\omega T}) - [\sin\theta(\omega)]\ \Re e\, S_c^H(e^{j\omega T})\}$$

where S_c^θ is the sensitivity of the phase response $\theta(\omega)$ with respect to coefficient c. Now given an arbitrary digital-filter network incorporating a multiplier with a coefficient c, the sensitivity of the transfer function of the network can be obtained by using the transpose approach as

$$S_c^H = \frac{\partial H(z)}{\partial c} = H_{12}(z)H_{34}(z)$$

where $H_{12}(z)$ and $H_{34}(z)$ are the transfer functions form the input of the network to the input of the multiplier and from the output of the multiplier to the output of the network, respectively (see pp. 125–128 of [6]).

With the transfer function sensitivities known, the amplitude-response sensitivities $S_{c_i}^M$ can be deduced and thus S_T, q, and K can be evaluated using Eqs. (14.13), (14.17), and (14.19), respectively. In turn, the statistical word length in Eq. (14.18) can be obtained.

The statistical word length is a convenient figure of merit of a specific filter structure. It can serve as a sensitivity measure in studies where a general comparison of various structures is desired. It can also be used as an objective function in word-length optimization algorithms [3].

A different approach for the study of quantization effects was proposed by Jenkins and Leon [7]. In this approach a computer-aided analysis scheme is used to generate confidence-interval error bounds on the time-domain response of the filter. The method can be used to study the effects of coefficient or product quantization in fixed-point or floating-point implementations. Furthermore, the quantization can be by rounding or truncation.

14.4 LOW-SENSITIVITY STRUCTURES

The effects of coefficient quantization are most serious in applications where the poles of the transfer function are located close to the unit circle $|z| = 1$. In such applications, small changes in the coefficients can cause large changes in the frequency response of the filter, and in extreme cases they can actually cause the filter to become unstable. In this section, we show that second-order structures can be derived whose sensitivity to coefficient quantization is much lower than that of the standard direct realizations described in Chap. 8. These structures can be used in the cascade or parallel realizations for the design of high-selectivity or narrow-band filters.

Let $M(\omega)$ be the amplitude response of a digital-filter structure and assume that b is a multiplier constant. Now let $\Delta M(\omega)$ be the change in $M(\omega)$ due to a quantization error Δb in b. The *normalized sensitivity* of $M(\omega)$ with respect to b is defined as

$$\begin{aligned}\bar{S}_b^M &= \lim_{\Delta b \to 0} \frac{\dfrac{\Delta M(\omega)}{M(\omega)}}{\dfrac{\Delta b}{b}} \\ &= \frac{b}{M(\omega)} \frac{\partial M(\omega)}{\partial b} \end{aligned} \tag{14.20}$$

and for small values of Δb, we have

$$\frac{\Delta M(\omega)}{M(\omega)} \approx \frac{\Delta b}{b} \bar{S}_b^M \tag{14.21}$$

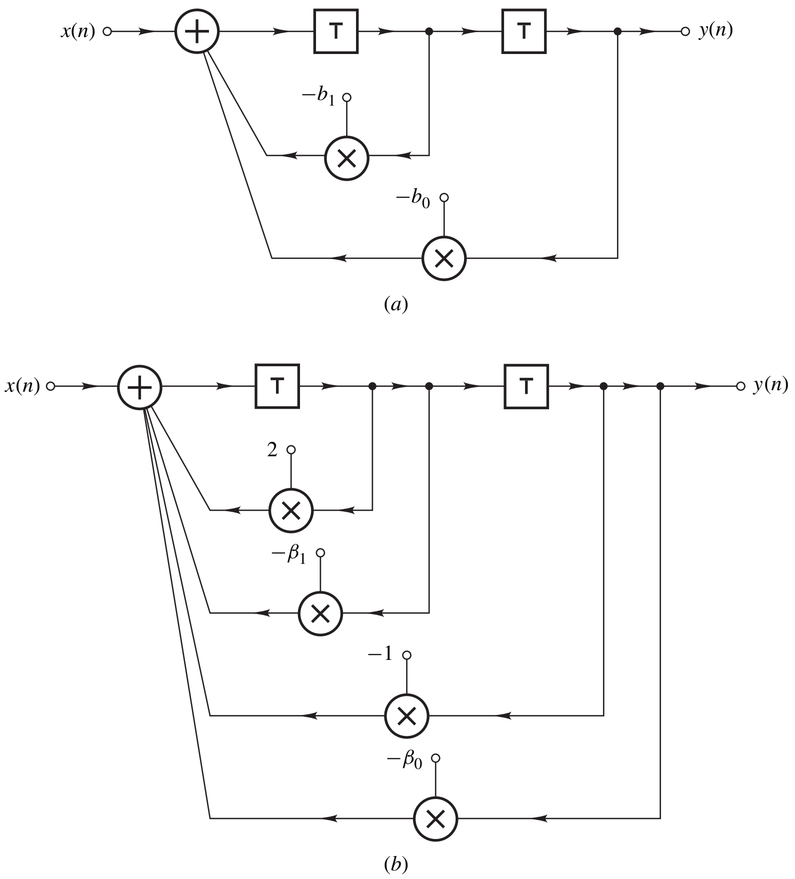
The normalized sensitivity can be used to compare different structures.

Consider the direct realization of Fig. 14.5*a*. Straightforward analysis gives the transfer function

$$H(z) = \frac{1}{z^2 + b_1 z + b_0}$$

and hence the amplitude response of the realization can be readily obtained as

$$M(\omega) = \frac{1}{\left[1 + b_0^2 + b_1^2 + 2b_1(1 + b_0)\cos\omega T + 2b_0 \cos 2\omega T\right]^{1/2}} \tag{14.22}$$



(*a*)

(*b*)

Figure 14.5 (*a*) Second-order direct realization, (*b*) corresponding low-sensitivity realization.

Using Eqs. (14.20) and (14.22), the normalized sensitivities of $M(\omega)$ with respect to b_0 and b_1 can be obtained as

$$\bar{S}_{b_0}^{M} = -b_0(b_0 + b_1 \cos \omega T + \cos 2\omega T)[M(\omega)]^2$$

$$\bar{S}_{b_1}^{M} = -b_1[b_1 + (1 + b_0) \cos \omega T][M(\omega)]^2$$

A modified version of the structure in Fig. 14.5*a* can be obtained by replacing each of the multipliers by two multipliers in parallel, as shown in Fig. 14.5*b*, as suggested by Agarwal and Burrus [8]. The transfer function of the original structure will be maintained in the new structure if

$$b_0 = 1 + \beta_0 \qquad \text{and} \qquad b_1 = \beta_1 - 2$$

and from Eq. (14.20)

$$\begin{aligned}\bar{S}_{\beta_0}^{M} &= \frac{\beta_0}{M(\omega)}\frac{\partial M(\omega)}{\partial \beta_0} = \frac{\beta_0}{b_0}\frac{\partial b_0}{\partial \beta_0} \times \frac{b_0}{M(\omega)}\frac{\partial M(\omega)}{\partial b_0}\\ &= \frac{\beta_0}{1+\beta_0}\bar{S}_{b_0}^{M} \qquad (14.23)\end{aligned}$$

and

$$\begin{aligned}\bar{S}_{\beta_1}^{M} &= \frac{\beta_1}{M(\omega)}\frac{\partial M(\omega)}{\partial \beta_1} = \frac{\beta_1}{b_1}\frac{\partial b_1}{\partial \beta_1} \times \frac{b_1}{M(\omega)}\frac{\partial M(\omega)}{\partial b_1}\\ &= \frac{\beta_1}{\beta_1-2}\bar{S}_{b_1}^{M} \qquad (14.24)\end{aligned}$$

Now if the poles of the transfer function are located close to the point $z = 1$, as may be the case in a narrow-band lowpass filter of high selectivity, then $b_0 \approx 1$ and $b_1 \approx -2$. As a consequence, β_0 and β_1 will be small and, therefore, from Eqs. (14.23) and (14.24)

$$|\bar{S}_{\beta_0}^{M}| \ll |\bar{S}_{b_0}^{M}| \qquad \text{and} \qquad |\bar{S}_{\beta_1}^{M}| \ll |\bar{S}_{b_1}^{M}|$$

In effect, if coefficients β_0 and β_1 are represented to the same degree of precision as coefficients b_0 and b_1, then the use of the structure in Fig. 14.5*b* instead of that in Fig. 14.5*a* leads to a significant reduction in the sensitivity to quantization errors, as can be seen from Eq. (14.21). The same degree of precision in the representation of the coefficients can be achieved by using either floating-point or fixed-point arithmetic. In the latter case, each multiplier coefficient should be scaled up to eliminate any zeros between the binary point and the most significant nonzero bit and the product scaled down by a corresponding shift operation.

The structure of Fig. 14.5*b*, like other structures in which all the outputs of multipliers are inputs to one and the same adder, has the advantage that the quantization of products can be carried out using one quantizer at the output of the adder instead of one quantizer at the output of each multiplier. Structures of this type are suitable for the application of *error-spectrum shaping*, which is a technique for the reduction of roundoff noise (see Sec. 14.8).

The disadvantage of the structure of Fig. 14.5*b* is that the low-sensitivity property can be achieved only if the poles of the transfer function are close to point $z = 1$. A family of structures that are suitable for the application of error-spectrum shaping and simultaneously lead to low sensitivity for a variety of pole locations close to the unit circle $|z| = 1$ can be obtained from the general second-order configuration depicted in Fig. 14.6 by using a method reported by Diniz and Antoniou [9]. In this configuration, branches A, B, C, D, and E represent unit delays or machine-representable multiplier constants, such as, 0, ±1, or ±2.

The structure of Fig. 14.6 realizes the transfer function

$$H(z) = \frac{N(z)}{D(z)} \qquad (14.25)$$

where $N(z)$ depends on the choice of multiplier coefficients c_0 to c_2 and

$$D(z) = z^2(1 - BD - AC - m_1A + ABE + m_2AB + ABCD + m_1ABD) \qquad (14.26)$$

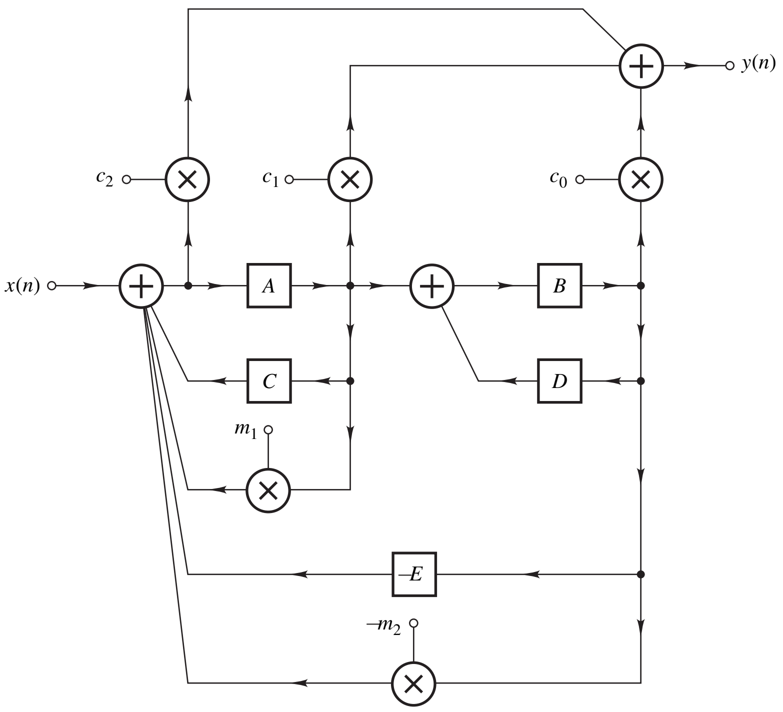


Figure 14.6 General second-order direct realization.

Assuming that $H(z)$ is of the form

$$H(z) = \frac{a_2z^2 + a_1z + a_0}{z^2 + b_1z + b_0} \tag{14.27}$$

and then comparing Eq. (14.25) with Eq. (14.27), a number of second-order structures can be deduced. In order to avoid delay-free loops (see Sec. 4.8.1) and keep the number of delays to the minimum of two, the constraints

$$A = z^{-1} \qquad \text{and} \qquad B \qquad \text{or} \qquad D = z^{-1}$$

must be satisfied. Therefore, two cases are possible, namely, Case I where $A = B = z^{-1}$ and Case II where $A = D = z^{-1}$.

14.4.1 Case I

For Case I, polynomial $D(z)$ of Eq. (14.26) assumes the form

$$D(z) = z^2 - z(C + D + m_1) + CD + m_1D + m_2 + E$$

and to achieve low sensitivity, multipliers C, D, and E must be chosen as

$$C + D = I_R[-b_1] \qquad \text{and} \qquad E = I_R[b_0 + b_1D + D^2] \tag{14.28}$$

where $I_R[x]$ is the closest integer to x. Equation (14.28) forces the values of m_1 and m_2 to be as low as possible and, as in the structure of Fig. 14.5*b*, low sensitivity is assured.

If the poles are close to point $z = 1$, then $b_1 \approx -2$ and $b_0 \approx 1$, and so

$$C + D = 2$$

We can thus assign

$$C = 1 \qquad D = 1 \qquad \text{and} \qquad E = 0$$

This choice of coefficients gives the structure of Fig. 14.5*b*, which is suitable for values of b_1 in the range $-2.0 < b_1 < -1.5$. Proceeding in the same way, the 15 structures in Table 14.2 can be deduced [9]. Structure I-2, like I-1, was reported in [8].

14.4.2 Case II

For Case II, polynomial $D(z)$ of Eq. (14.26) assumes the form

$$D(z) = z^2 - z(B + C + m_1 - m_2B - BE) + BC + m_1B$$

Table 14.2 Structures for Case I

Structure	C	D	E	Range of b_1
I-1	1	1	0	$-2.0 < b_1 < -1.5$
I-2	2	0	1	
I-3	0	2	1	$-2.0 < b_1 < -1.75$
I-4	0	2	2	$-1.75 < b_1 < -1.5$
I-5	1	0	1	$-1.5 < b_1 < -0.5$
I-6	0	1	1	
I-7	0	0	1	$-0.5 < b_1 < 0.5$
I-8	−1	1	2	
I-9	1	−1	2	
I-10	−1	0	1	$0.5 < b_1 < 1.5$
I-11	0	−1	1	
I-12	0	−2	2	$1.5 < b_1 < 1.75$
I-13	−2	0	1	$1.5 < b_1 < 2.0$
I-14	−1	−1	0	
I-15	0	−2	1	$1.75 < b_1 < 2.0$

Table 14.3 Structures for Case II

Structure	B	C	E	Range of b_1
II-1	1	1	0	$-2.0 < b_1 < -1.5$
II-2	1	1	1	$-1.5 < b_1 < -0.5$
II-3	1	1	2	$-0.5 < b_1 < 0$
II-4	-1	-1	2	$0 < b_1 < 0.5$
II-5	-1	-1	1	$0.5 < b_1 < 1.5$
II-6	-1	-1	0	$1.5 < b_1 < 2.0$

and to achieve low sensitivity, constants B, C, and E must be chosen as

$$B = 1 \qquad C = 1 \qquad \text{and} \qquad E = I_R[b_1 + b_0 + 1] \tag{14.29}$$

for poles with positive real part, and

$$B = -1 \qquad C = -1 \qquad \text{and} \qquad E = -I_R[b_1 - b_0 - 1] \tag{14.30}$$

for poles with negative real part. Using Eqs. (14.29) and (14.30), the structures of Table 14.3 can be deduced [9]. Structure II-1 was reported by Nishimura, Hirano, and Pal [10].

Different biquadratic transfer functions can be realized by using the formulas in Table 14.4.

In the above approach, the poles of the transfer function have been assumed to be close to the unit circle of the z plane. An alternative approach for selecting the optimum structure for a given transfer function, which is applicable for any pair of poles in the unit circle, was described by Ramana Rao and Eswaran [11].

Table 14.4 Realization of biquadratic transfer functions

Multiplier constant	Case I	Case II
c_0	$a_0 + a_1 D + a_2 D^2$	$a_2 + \dfrac{a_1}{B} + \dfrac{a_0}{B^2}$
c_1	$a_1 + a_2 D$	$-\dfrac{a_0}{B}$
c_2	a_2	a_2
m_1	$-b_1 - C - D$	$\dfrac{b_0}{B} - C$
m_2	$b_0 + b_1 D + D^2 - E$	$1 + \dfrac{b_1}{B} + \dfrac{b_0}{B^2} - E$

14.5 PRODUCT QUANTIZATION

The output of a finite-word-length multiplier can be expressed as

$$Q[c_i x(n)] = c_i x(n) + e(n)$$

where $c_i x(n)$ and $e(n)$ are the exact product and quantization error, respectively. A machine multiplier can thus be represented by the model depicted in Fig. 14.7*a*, where $e(n)$ is a noise source.

Consider the filter structure of Fig. 14.7*b* and assume a fixed-point implementation. Each multiplier can be replaced by the model of Fig. 14.7*a*, as in Fig. 14.7*c*. If product quantization is carried out by rounding, each noise signal $e_i(n)$ can be regarded as a random process with uniform probability density, that is,

$$p(e_i; n) = \begin{cases} \frac{1}{q} & \text{for } -\frac{q}{2} \le e_i(n) \le \frac{q}{2} \\ 0 & \text{otherwise} \end{cases}$$

Hence, from Eqs. (13.8) and (13.9) and Sec. 13.8, we have

$$E\{e_i(n)\} = 0 \tag{14.31}$$

$$E\left\{e_i^2(n)\right\} = \frac{q^2}{12} \tag{14.32}$$

$$r_{e_i}(k) = E\{e_i(n)e_i(n+k)\} \tag{14.33}$$

If the signal levels throughout the filter are much larger than q, the following reasonable assumptions can be made: (1) $e_i(n)$ and $e_i(n+k)$ are statistically independent for any value of n $(k \ne 0)$, and (2) $e_i(n)$ and $e_j(n+k)$ are statistically independent for any value of n or k $(i \ne j)$. Let us examine the implications of these assumptions starting with the first assumption. From Eqs. (14.31)–(14.33) and Eq. (13.4)

$$r_{e_i}(0) = E\left\{e_i^2(n)\right\} = \frac{q^2}{12}$$

and
$$r_{e_i}(k)\Big|_{k\ne0} = E\{e_i(n)\}E\{e_i(n+k)\} = 0$$

i.e.,
$$r_{e_i}(k) = \frac{q^2}{12}\delta(k)$$

where $\delta(k)$ is the impulse function. Therefore, the power spectral density (PSD) of $e_i(n)$ is

$$S_{e_i}(z) = \mathcal{Z} r_{e_i}(k) = \frac{q^2}{12} \tag{14.34}$$

that is, $e_i(n)$ *is a white-noise process*.

Let us now consider the implications of the second assumption. The autocorrelation of sum $e_i(n) + e_j(n)$ is

$$\begin{aligned} r_{e_i+e_j}(k) &= E\{[e_i(n) + e_j(n)][e_i(n+k) + e_j(n+k)]\} \\ &= E\{e_i(n)e_i(n+k)\} + E\{e_i(n)\}E\{e_j(n+k)\} + E\{e_j(n)\}E\{e_i(n+k)\} \\ &\quad + E\{e_j(n)e_j(n+k)\} \end{aligned}$$

or
$$r_{e_i+e_j}(k) = r_{e_i}(k) + r_{e_j}(k)$$

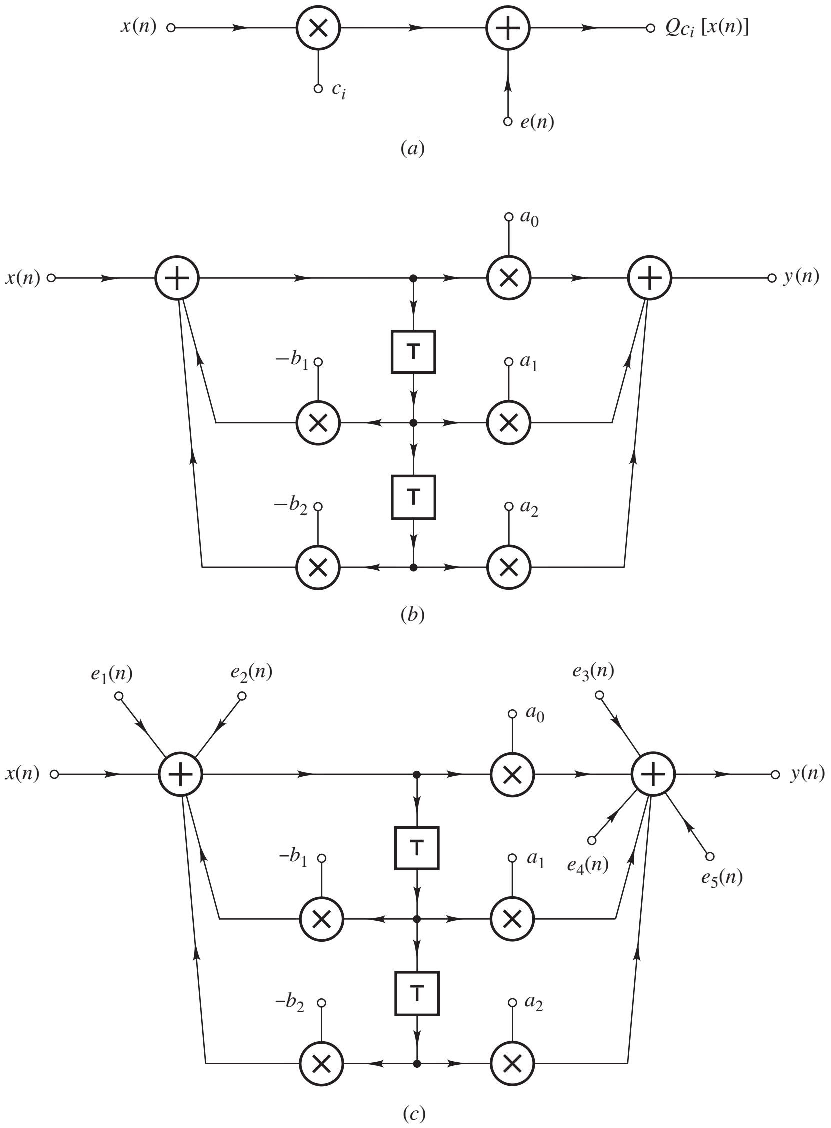


Figure 14.7 Product quantization: (*a*) Noise model for a multiplier, (*b*) second-order canonic section, (*c*) noise model for a second-order canonic section.

Therefore

$$S_{e_i+e_j}(z) = \mathcal{Z}[r_{e_i}(k) + r_{e_j}(k)] = S_{e_i}(z) + S_{e_j}(z)$$

i.e., *the PSD of a sum of two statistically independent processes is equal to the sum of their respective PSDs*. In effect, superposition can be employed.

Now from Fig. 14.7*c* and Eq. (13.22)

$$S_y(z) = H(z)H(z^{-1})\sum_{i=1}^{2} S_{e_i}(z) + \sum_{i=3}^{5} S_{e_i}(z)$$

where $H(z)$ is the transfer function of the filter, and hence from Eq. (14.34) the output PSD is given by

$$S_y(z) = \frac{q^2}{6}H(z)H(z^{-1}) + \frac{q^2}{4}$$

The above approach is applicable to any filter structure. Furthermore, it can be used to study the effects of input quantization.

14.6 SIGNAL SCALING

If the amplitude of any internal signal in a fixed-point implementation is allowed to exceed the dynamic range, *overflow* will occur and the output signal will be severely distorted. On the other hand, if all the signal amplitudes throughout the filter are unduly low, the filter will be operating inefficiently and *the signal-to-noise ratio will be poor*. Therefore, for optimum filter performance suitable *signal scaling* must be employed to adjust the various signal levels.

A scaling technique applicable to one's- or two's-complement implementations was proposed by Jackson [12]. In this technique a scaling multiplier is used at the input of a filter section, as in Fig. 14.8, with its constant λ chosen such that amplitudes of multiplier inputs are bounded by M if $|x(n)| \le M$. Under these circumstances, adder outputs are also bounded by M and cannot overflow. This is due to the fact that a machine-representable sum is always evaluated correctly in one's- or two's-complement arithmetic, even if overflow does occur in one of the partial sums (see Example 14.3). There are two methods for the determination of λ, as follows.

14.6.1 Method A

Consider the filter section of Fig. 14.8, where $v(n)$ is a multiplier input. The transfer function between nodes 1 and 2 can be denoted by $F(z)$. From the convolution summation

$$v(n) = \sum_{k=0}^{\infty} \lambda f(k)x(n-k) \tag{14.35}$$

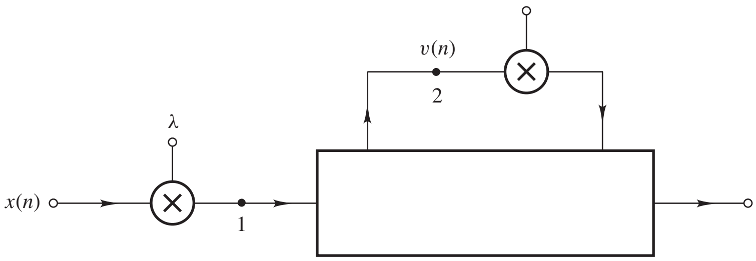


Figure 14.8 Signal scaling.

where $$f(n) = \mathcal{Z}^{-1}F(z)$$

Evidently

$$|v(n)| \le \sum_{k=0}^{\infty} |\lambda f(k)| \cdot |x(n-k)|$$

and if

$$|x(n)| \le M$$

then

$$|v(n)| \le M \sum_{k=0}^{\infty} |\lambda f(k)|$$

Thus a sufficient condition for $|v(n)| \le M$ is

$$\sum_{k=0}^{\infty} |\lambda f(k)| \le 1$$

or

$$\lambda \le \frac{1}{\sum_{k=0}^{\infty} |f(k)|} \tag{14.36}$$

Now consider the specific signal

$$x(n-k) = \begin{cases} M & \text{for } \lambda f(k) > 0 \\ -M & \text{for } \lambda f(k) < 0 \end{cases}$$

where $M > 0$. From Eq. (14.35)

$$v(n) = M \sum_{k=0}^{\infty} |\lambda f(k)|$$

and, therefore, $|v(n)| \le M$ if and only if Eq. (14.36) holds. Signal scaling can be applied by calculating the infinite sum of the magnitude of the impulse response from the input of the filter to the input of each multiplier and then evaluating λ using the largest sum so obtained in Eq. (14.36).

The above method guarantees that overflow will never occur as long as the input is bounded as prescribed. Unfortunately, the signal levels at the various nodes can be quite low and since quantization errors are independent of the signal level, a reduced signal-to-noise ratio may result. In addition, the computation of the sum in Eq. (14.36) is not usually straightforward.

14.6.2 Method B

The second and more efficient method for the evaluation of λ is based on *L_p-norm notation*. The L_p norm of an arbitrary periodic function $A(e^{j\omega T})$ with period ω_s is defined as

$$\|A\|_p = \left[\frac{1}{\omega_s}\int_0^{\omega_s} |A(e^{j\omega T})|^p \, d\omega\right]^{1/p}$$

where $p \geq 1$. It exists if

$$\int_0^{\omega_s} |A(e^{j\omega T})|^p \, d\omega < \infty$$

and if $A(e^{j\omega T})$ is continuous, then the limit

$$\lim_{p \to \infty} \|A\|_p = \|A\|_\infty = \max_{0 \leq \omega \leq \omega_s} |A(e^{j\omega T})| \tag{14.37}$$

exists, as can be easily demonstrated (see Prob. 14.22). Usually, $A(e^{j\omega T})$ is obtained by evaluating function $A(z)$ on the unit circle $z = e^{j\omega T}$ and $\|A\|_p$ is often referred to as the L_p norm of either $A(e^{j\omega T})$ or $A(z)$.

Now let

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \qquad \text{with } a < |z| < b$$

$$F(z) = \sum_{n=-\infty}^{\infty} f(n) z^{-n} \qquad \text{with } c < |z| < b$$

where $c < 1$ for a stable filter and $b > 1$. From Eq. (14.35)

$$V(z) = \lambda F(z) X(z) \qquad \text{with } d < |z| < b$$

where $d = \max\,(a, c)$. The inverse z transform of $V(z)$ is

$$v(n) = \frac{1}{2\pi j} \oint_\Gamma \lambda F(z) X(z) z^{n-1} \, dz \tag{14.38}$$

where Γ is a contour in the annulus of convergence. If $a < 1$, Γ can be taken to be the unit circle $|z| = 1$. With $z = e^{j\omega T}$ Eq. (14.38) becomes

$$v(n) = \frac{1}{\omega_s} \int_0^{\omega_s} \lambda F(e^{j\omega T}) X(e^{j\omega T}) e^{jn\omega T} \, d\omega$$

We can thus write

$$|v(n)| \leq \left[\max_{0 \leq \omega \leq \omega_s} |X(e^{j\omega T})|\right] \frac{1}{\omega_s} \int_0^{\omega_s} |\lambda F(e^{j\omega T})| \, d\omega \tag{14.39}$$

or

$$|v(n)| \leq \left[\max_{0 \leq \omega \leq \omega_s} |\lambda F(e^{j\omega T})|\right] \frac{1}{\omega_s} \int_0^{\omega_s} |X(e^{j\omega T})| \, d\omega \tag{14.40}$$

and by virtue of the *Schwarz inequality* [12], we can write

$$|v(n)| \leq \left[\frac{1}{\omega_s} \int_0^{\omega_s} |\lambda F(e^{j\omega T})|^2 d\omega\right]^{1/2} \left[\frac{1}{\omega_s} \int_0^{\omega_s} |X(e^{j\Omega T})|^2 \, d\Omega\right]^{1/2} \tag{14.41}$$

If L_p-norm notation is used, Eqs. (14.39)–(14.41) can be put in the compact form

$$|v(n)| \leq \|X\|_\infty \|\lambda F\|_1 \qquad |v(n)| \leq \|X\|_1 \|\lambda F\|_\infty \qquad |v(n)| \leq \|X\|_2 \|\lambda F\|_2$$

In fact, these inequalities are particular cases of the *Holder inequality* [12, 13]

$$|v(n)| \leq \|X\|_q \|\lambda F\|_p \tag{14.42}$$

where the relation

$$p = \frac{q}{q-1} \tag{14.43}$$

must hold.

Equation (14.42) is valid for any transfer function $\lambda F(z)$ including $\lambda F(z) = 1$, in which case $v(n) = x(n)$ and $\|1\|_p = 1$ for all $p \geq 1$. Consequently, from Eq. (14.42)

$$|x(n)| \leq \|X\|_q \qquad \text{for all } q \geq 1$$

Now if

$$|x(n)| \leq \|X\|_q \leq M$$

Eq. (14.42) gives

$$|v(n)| \leq M\|\lambda F\|_p$$

Therefore,

$$|v(n)| \leq M$$

provided that

$$\|\lambda F\|_p \leq 1$$

or

$$\lambda \leq \frac{1}{\|F\|_p} \qquad \text{for } \|X\|_q \leq M \tag{14.44}$$

where Eq. (14.43) must hold.

14.6.3 Types of Scaling

Depending on the values of p and q, two types of scaling can be identified, namely, L_2 *scaling* if $p = q = 2$ and L_∞ *scaling* if $p = \infty$ and $q = 1$.

From the definition of the L_p norm and Eq. (14.37), we have

$$\begin{aligned}\|F\|_2 = \left[\frac{1}{\omega_s}\int_0^{\omega_s} |F(e^{j\omega T})|^2 d\omega\right]^{1/2} &\leq \left\{\frac{1}{\omega_s}\int_0^{\omega_s} \left[\max_{0\leq\omega\leq\omega_s} |F(e^{j\omega T})|\right]^2 d\omega\right\}^{1/2}\\ &\leq \left(\frac{1}{\omega_s}\int_0^{\omega_s} \|F\|_\infty^2\, d\omega\right)^{1/2}\\ &\leq \|F\|_\infty\end{aligned}$$

or

$$\frac{1}{\|F\|_2} \geq \frac{1}{\|F\|_\infty}$$

As a consequence, L_2 scaling usually yields larger scaling constants than L_∞ scaling. This means that the signal levels at the various nodes are usually larger, and thus a better signal-to-noise ratio can be achieved. However, L_2 scaling is more likely to cause overflow than L_∞ scaling. The circumstances in which these two types of scaling are applicable are examined next.

If $x(n)$ is obtained by sampling a random or deterministic, finite-energy, bandlimited, continuous-time signal $x(t)$ such that

$$X_A(j\omega) = \mathcal{F}x(t) = 0 \qquad \text{for } |\omega| \geq \omega_s/2 \tag{14.45}$$

we can write

$$\begin{aligned}\|X\|_2 &= \left[\frac{1}{\omega_s}\int_0^{\omega_s} |X(e^{j\omega T})|^2\,d\omega\right]^{1/2}\\ &= \left[\frac{1}{\omega_s}\int_{-\omega_s/2}^{\omega_s/2} |X(e^{j\omega T})|^2\,d\omega\right]^{1/2}\end{aligned}$$

where $X(z) = \mathcal{Z}x(n)$. From Eq. (6.46a), we have

$$X(e^{j\omega T}) = \frac{1}{T}X_A(j\omega) \qquad \text{for } |\omega| < \omega_s/2$$

and hence

$$\begin{aligned}\|X\|_2 &= \left[\frac{1}{2\pi T}\int_{-\omega_s/2}^{\omega_s/2} |X_A(j\omega)|^2\,d\omega\right]^{1/2}\\ &= \left[\frac{1}{2\pi T}\int_{-\infty}^{\infty} |X_A(j\omega)|^2\,d\omega\right]^{1/2}\end{aligned}$$

On using Parseval's formula (see Theorem 2.16), we obtain

$$\|X\|_2 = \left[\frac{1}{T}\int_{-\infty}^{\infty} |x(t)|^2\,dt\right]^{1/2} \tag{14.46}$$

For a finite-energy signal, the above integral converges. Therefore, Eq. (14.42) holds with $p = 2$ and $q = 2$, and L_2 scaling is applicable.

If $x(n)$ is obtained by sampling a continuous-time signal $x(t)$ whose energy content is not finite (e.g., a sinusoidal signal) the integral in Eq. (14.46) does not converge, $\|X\|_2$ does not exist, and L_2 scaling is not applicable; therefore, if such a signal is applied to a structure incorporating L_2 scaling, then signal overflow may occur. If $x(t)$ is bounded and bandlimited, Eq. (14.45) is satisfied, and hence we can write

$$\begin{aligned}\|X\|_1 &= \frac{1}{\omega_s}\int_{-\omega_s/2}^{\omega_s/2} |X(e^{j\omega T})|\,d\omega\\ &= \frac{1}{2\pi}\int_{-\omega_s/2}^{\omega_s/2} |X_A(j\omega)|\,d\omega \end{aligned} \tag{14.47}$$

i.e., $\|X\|_1$ exists and Eq. (14.42) holds with $p = \infty$ and $q = 1$, and L_∞ scaling is applicable. The amplitude spectrum of $x(t)$ may become unbounded if $x(t)$ is a sinusoidal signal, in which case $X_A(j\omega)$ has poles on the $j\omega$ axis, or if $x(t)$ is constant, in which case $X_A(j\omega)$ is an impulse function. However, in both of these cases $\|X\|_1$ exists, as will now be demonstrated.

If $x(t) = M\cos\omega_0 nT$ where $0 \leq \omega_0 \leq \omega_s/2$, we have

$$X_A(j\omega) = \pi M[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]$$

and Eq. (14.47) gives

$$\begin{aligned}\|X\|_1 &= \frac{1}{2\pi}\int_{-\omega_s/2}^{\omega_s/2} |\pi M[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]|\,d\omega \\ &= M\end{aligned}$$

On the other hand, if $x(t) = M$, then

$$X_A(j\omega) = 2\pi M\delta(\omega)$$

and

$$\|X\|_1 = \frac{1}{2\pi}\int_{-\omega_s/2}^{\omega_s/2} |2\pi M\delta(\omega)|\,d\omega = M$$

Therefore, if we select λ such that

$$\|\lambda F\|_\infty = \max_{0\le\omega\le\omega_s} |\lambda F(e^{j\omega T})| \le 1$$

then

$$v(n) \le M$$

This result is to be expected. With a sinusoidal input and the gain between the input and node 2 in Fig. 14.8 equal to or less than unity, the signal at node 2 will be a sinusoid with an amplitude equal to or less than M.

14.6.4 Application of Scaling

If there are m multipliers in the filter of Fig. 14.8, then $|v_i(n)| \le M$ provided that

$$\lambda_i \le \frac{1}{\|F_i\|_p}$$

for $i = 1, 2, \ldots, m$. Therefore, in order to ensure that all multiplier inputs are bounded by M we must assign

$$\lambda = \min(\lambda_1, \lambda_2, \ldots, \lambda_m)$$

or

$$\lambda = \frac{1}{\max(\|F_1\|_p, \|F_2\|_p, \ldots, \|F_m\|_p)} \tag{14.48}$$

In the case of parallel or cascade realizations, efficient scaling can be accomplished by using one scaling multiplier per section.

Example 14.4 Deduce the scaling formulation for the cascade filter of Fig. 14.9a assuming that $p = \infty$ and $q = 1$.

■ **Solution**

The only critical signals are $y'_j(n)$ and $y_j(n)$ since the inputs of the feedback multipliers are delayed versions of $y'_j(n)$. The filter can be represented by the signal flow graph of

Fig. 14.9*b*, where

$$F'_j(z) = \frac{z^2}{z^2 + b_{1j}z + b_{2j}} \qquad F_j(z) = \frac{(z+1)^2}{z^2 + b_{1j}z + b_{2j}}$$

By using Eq. (14.48), we obtain

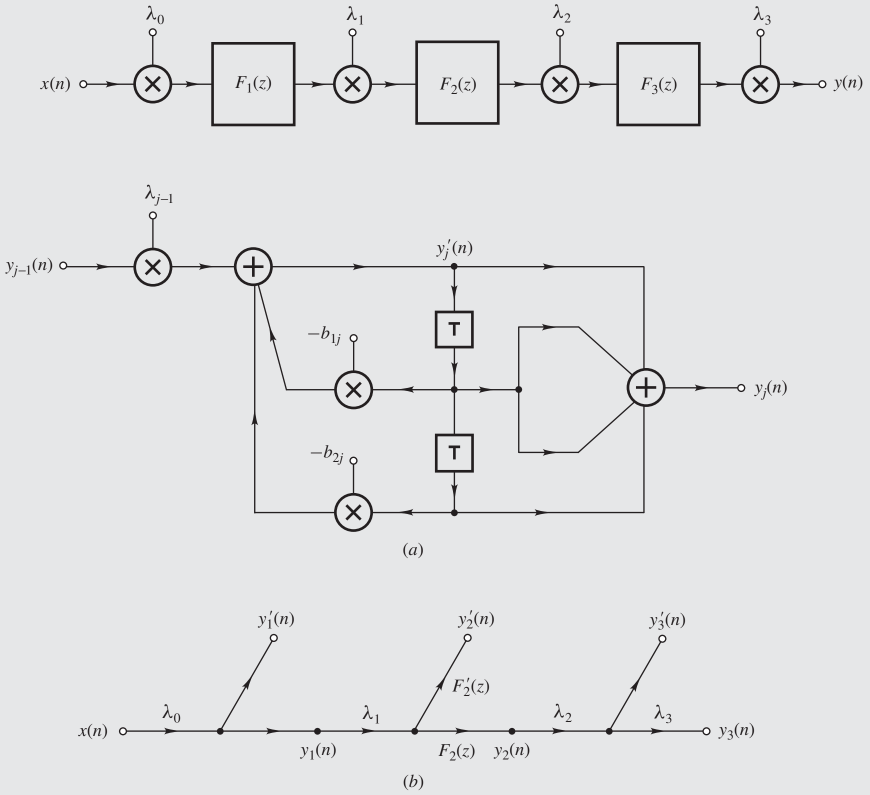
$$\lambda_0 = \frac{1}{\max\left(\|F'_1\|_\infty,\ \|F_1\|_\infty\right)} \qquad \lambda_1 = \frac{1}{\lambda_0 \max\left(\|F_1F'_2\|_\infty,\ \|F_1F_2\|_\infty\right)}$$

$$\lambda_2 = \frac{1}{\lambda_0\lambda_1 \max\left(\|F_1F_2F'_3\|_\infty,\ \|F_1F_2F_3\|_\infty\right)} \qquad \blacksquare$$

The scaling constants can be evaluated by noting that

$$\|\Pi F_i\|_\infty = \max_{0\le\omega\le\omega_s} |\Pi F_i(e^{j\omega T})|$$

according to Eq. (14.37).



(*a*)

(*b*)

Figure 14.9 (*a*) Cascade filter, (*b*) signal flow-graph representation.

The scaling constants are usually chosen to be the *nearest powers of 2* satisfying the overflow constraints. In this way, scaling multiplications can be reduced to simple data shifts.

In cascade filters, the *ordering of sections* has an influence on scaling, which in turn has an influence on the output noise. Analytical techniques for determining the optimum sequential ordering have not yet been devised. Nevertheless, some guidelines suggested by Jackson [14] lead to a good ordering.

14.7 MINIMIZATION OF OUTPUT ROUNDOFF NOISE

The level of output roundoff noise in fixed-point implementations can be reduced by increasing the word length. An alternative approach is to assume a general structure and vary its topology or parameters in such a way as to minimize the output roundoff noise. A method of this type that leads to *optimal* state-space structures was proposed by Mullis and Roberts [15]. The method is based on a state-space noise formulation reported by these authors and Hwang [16] at approximately the same time, and the principles involved are detailed below. The method is applicable to the general Nth-order realization but for the sake of simplicity it will be presented in terms of the second-order case.

A second-order state-space realization can be represented by the signal flow graph in Fig. 14.10 where $e_i(n)$ for $i = 1, 2$, and 3 are noise sources due to the quantization of products. From Sec. 4.8.2, the filter can be represented by the equations

$$\mathbf{q}(n+1) = \mathbf{A}\mathbf{q}(\mathbf{n}) + \mathbf{b}x(n) + \mathbf{e}(n) \tag{14.49a}$$

$$y(n) = \mathbf{c}^T\mathbf{q}(\mathbf{n}) + dx(n) + e_3(n) \tag{14.49b}$$

where $\mathbf{e}^T(n) = [e_1(n)\ e_2(n)]$. Let $F_1(z)$, $F_2(z)$ and $G_1(z)$, $G_2(z)$ be the transfer functions from the input to nodes $q_1(n)$, $q_2(n)$ and from nodes $e_1(n)$, $e_2(n)$ to the output, respectively. In terms of this notation, the column vectors $\mathbf{f}(z)$ and $\mathbf{g}(z)$ can be formed as

$$\mathbf{f}^T(z) = [F_1(z)\ F_2(z)] \qquad \text{and} \qquad \mathbf{g}^T(z) = [G_1(z)\ G_2(z)] \tag{14.50}$$

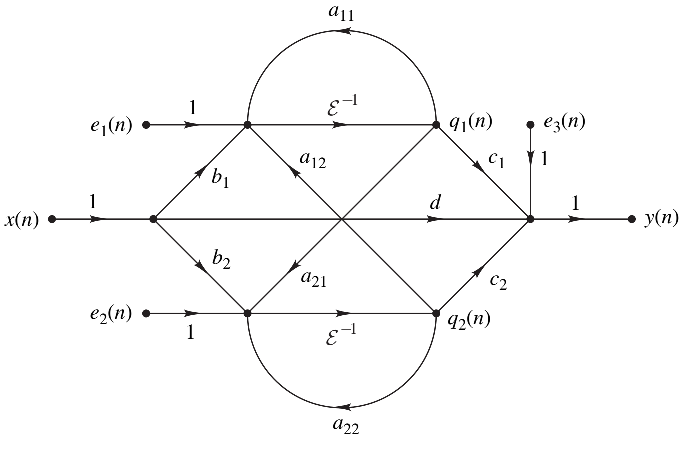


Figure 14.10 Second-order state-space realization.

and from Eq. (14.49), we obtain

$$\mathbf{f}(z) = (z\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} \qquad \text{and} \qquad \mathbf{g}(z) = (z\mathbf{I} - \mathbf{A}^T)^{-1}\mathbf{c} \tag{14.51}$$

(see Prob. 14.23).

Now if the realization of Fig. 14.10 is represented by the set $\{\mathbf{A}, \mathbf{b}, \mathbf{c}^T, d\}$ and the state vector $\mathbf{q}(n)$ is subjected to a transformation of the form $\tilde{\mathbf{q}}(n) = \mathbf{T}\mathbf{q}(n)$, a new realization

$$\{\tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}^T, \tilde{d}\}$$

is obtained where

$$\tilde{\mathbf{A}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1} \qquad \tilde{\mathbf{b}} = \mathbf{T}\mathbf{b} \qquad \tilde{\mathbf{c}}^T = \mathbf{c}^T\mathbf{T}^{-1} \qquad \tilde{d} = d \tag{14.52}$$

and from Eq. (14.49), one can show that

$$\tilde{\mathbf{f}}(z) = \mathbf{T}\mathbf{f}(z) \qquad \text{and} \qquad \tilde{\mathbf{g}}(z) = \mathbf{T}^{-1}\mathbf{g}(z) \tag{14.53}$$

(see Prob. 14.24). The realization $\{\tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}^T, \tilde{d}\}$ has minimum output roundoff noise subject to L_2-norm scaling if and only if

$$\tilde{\mathbf{W}} = \mathbf{D}\tilde{\mathbf{K}}\mathbf{D} \tag{14.54}$$

and

$$\tilde{K}_{ii}\tilde{W}_{ii} = \tilde{K}_{jj}\tilde{W}_{jj} \qquad \text{for all } i, j \tag{14.55}$$

where $\mathbf{D}$ is a diagonal matrix and $\tilde{\mathbf{K}} = \{\tilde{K}_{ij}\}$ and $\tilde{\mathbf{W}} = \{\tilde{W}_{ij}\}$ are the matrices given by

$$\tilde{\mathbf{K}} = \frac{1}{2\pi j}\oint_\Gamma \tilde{\mathbf{f}}(z)\tilde{\mathbf{f}}^T(z^{-1})z^{-1}\,dz \tag{14.56}$$

and

$$\tilde{\mathbf{W}} = \frac{1}{2\pi j}\oint_\Gamma \tilde{\mathbf{g}}(z)\tilde{\mathbf{g}}^T(z^{-1})z^{-1}\,dz \tag{14.57}$$

respectively [15]. Matrices $\mathbf{K}$ and $\mathbf{W}$ are known as the *reachability* and *observability gramians*, respectively.

From Eq. (14.44), L_2 scaling can be applied by ensuring that

$$\|\tilde{F}_i\|_2 = 1 \qquad \text{for all } i \tag{14.58}$$

and from Eqs. (14.56) and (14.58), we have

$$\begin{aligned}\tilde{K}_{ii} &= \frac{1}{2\pi j}\oint_\Gamma \tilde{F}_i(z)\tilde{F}_i(z^{-1})z^{-1}\,dz \\ &= \frac{1}{\omega_s}\int_0^{\omega_s} |\tilde{F}_i(e^{j\omega T})|^2\,d\omega \\ &= \|\tilde{F}_i\|_2^2 = 1 \end{aligned} \tag{14.59}$$

Therefore, the condition for minimum output roundoff noise in Eq. (14.55) assumes the form

$$\tilde{W}_{ii} = \tilde{W}_{jj} \qquad \text{for all } i,\ j \tag{14.60}$$

and from Eq. (14.57), we have

$$\|\tilde{G}_i\|_2^2 = \|\tilde{G}_j\|_2^2 \qquad \text{for all } i,\ j$$

In effect, *the output noise is minimum if the individual contributions due to the different noise sources are all equal*, as may be expected.

The application of the above method to the Nth-order general state-space realization would require $N^2 + 2N + 1$ multipliers, as opposed to $2N + 1$ in parallel or cascade canonic structures. That is, the method is uneconomical. Recognizing this problem, Mullis and Roberts applied their methodology to obtain so-called *block-optimal* parallel and cascade structures that require only $4N + 1$ and $9N/2$ multipliers, respectively. Unfortunately, in both cases the realization process is relatively complicated; in addition, in the latter case the pairing of zeros and poles into biquadratic transfer functions and the ordering of second-order sections are not optimized, and to be able to obtain a structure that is *fully* optimized the designer must undertake a large number of designs. A practical approach to this problem is to obtain second-order sections that are individually optimized and then use sections of this type in parallel or cascade for the realization of Nth-order transfer functions. Realizations so obtained are said to be *section-optimal*. This approach gives optimal parallel structures since in this case the output noise is independent of the pairing of zeros and poles and the ordering of sections; furthermore, as was shown by Jackson, Lindgren, and Kim [17], with some experience the approach gives suboptimal cascade structures that are nearly as good as corresponding block-optimal cascade structures.

Optimized second-order sections can be obtained by noting that Eq. (14.54) is satisfied if and only if $\mathbf{D} = \rho\mathbf{I}$, according to Eqs. (14.59) and (14.60); hence, Eq. (14.54) can be expressed as

$$\tilde{\mathbf{W}} = \rho^2\tilde{\mathbf{K}} \tag{14.61}$$

Since $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{K}}$ are symmetric matrices with equal diagonal elements, Eq. (14.61) assumes the form

$$\tilde{\mathbf{W}} = \rho^2\mathbf{J}\tilde{\mathbf{K}}\mathbf{J} \tag{14.62}$$

where

$$\mathbf{J} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

for a second-order realization. Eq. (14.62) is satisfied by a network in which

$$\tilde{\mathbf{A}}^T = \mathbf{J}\tilde{\mathbf{A}}\mathbf{J}$$

and

$$\tilde{\mathbf{c}} = \rho\mathbf{J}\tilde{\mathbf{b}}$$

If $\tilde{\mathbf{A}} = \{\tilde{a}_{ij}\}$, $\tilde{\mathbf{b}} = \{\tilde{b}_i\}$, and $\tilde{\mathbf{c}}^T = \{\tilde{c}_i\}$, then the preceding conditions yield

$$\tilde{a}_{11} = \tilde{a}_{22}$$

and

$$\frac{\tilde{b}_1}{\tilde{b}_2} = \frac{\tilde{c}_2}{\tilde{c}_1}$$

If $\{\hat{\mathbf{A}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}^T, \hat{d}\}$ represents a specific realization that satisfies these conditions, then applying the scaling transformation

$$\mathbf{T} = \begin{bmatrix} \|\hat{F}_1\|_2^{-1} & 0 \\ 0 & \|\hat{F}_2\|_2^{-1} \end{bmatrix} \tag{14.63}$$

results in a structure that satisfies Eqs. (14.54), (14.59), and (14.60) simultaneously and, therefore, is optimal for L_2 scaling. It should be mentioned that if the transformation

$$\mathbf{T} = \begin{bmatrix} \|\hat{F}_1\|_\infty^{-1} & 0 \\ 0 & \|\hat{F}_2\|_\infty^{-1} \end{bmatrix} \tag{14.64}$$

is used instead, the structure obtained is not optimal for L_∞ scaling, although good results are usually obtained.

A biquadratic second-order transfer function with complex-conjugate poles can be expressed as

$$H(z) = \frac{\gamma_1 z + \gamma_0}{z^2 + \beta_1 z + \beta_0} + \delta \tag{14.65}$$

and on the basis of the above principles, Jackson et al. [17] obtained the following optimal state-space realization:

$$\hat{a}_{11} = \hat{a}_{22} = -\beta_1/2 \tag{14.66a}$$

$$\hat{a}_{12} = (1 + \gamma_0)(K_1 \pm K_2)/\gamma_1^2 \tag{14.66b}$$

$$\hat{a}_{21} = [K_1 \mp K_2]^2/(1 + \gamma_0) \tag{14.66c}$$

$$\hat{b}_1 = \frac{1}{2}(1 + \gamma_0) \qquad \hat{b}_2 = \frac{1}{2}\gamma_1 \tag{14.66d}$$

$$\hat{c}_1 = \frac{\gamma_1}{1 + \gamma_0} \qquad \hat{c}_2 = 1 \tag{14.66e}$$

$$\hat{d} = \delta \tag{14.66f}$$

$$K_1 = \gamma_0 - \frac{1}{2}\beta_1\gamma_1$$

$$K_2 = \sqrt{\left(\gamma_0^2 - \gamma_0\gamma_1\beta_1 + \beta_0\gamma_1^2\right)}$$

An arbitrary parallel or cascade design can be obtained by expressing the individual biquadratic transfer functions as in Eq. (14.65) and then using the scaling transformation

$$\mathbf{T} = \begin{bmatrix} \|\hat{F}_{1i}\|_p^{-1} & 0 \\ 0 & \|\hat{F}_{2i}\|_p^{-1} \end{bmatrix}$$

with $p = 2$ or ∞ for each section, where $F_{1i}(z)$ and $F_{2i}(z)$ are the transfer functions between the input of the filter and the state-variable nodes 1 and 2, respectively, of the ith section.

14.8 APPLICATION OF ERROR-SPECTRUM SHAPING

An alternative approach for the reduction of output roundoff noise is through the application of a technique known as *error-spectrum shaping* [18, 19]. This technique involves the generation of a roundoff-error signal and the application of local feedback for the purpose of controlling and manipulating the output roundoff noise. The technique entails additional hardware which increases in direct proportion to the number of adders in the structure. Consequently, only structures in which the outputs of all multipliers are inputs to one and the same adder are suitable for the application of error-spectrum shaping. The most well-known structure of this type is the classical direct realization. Other structures of this type are the low-sensitivity structures described in Sec. 14.4.

The application of error-spectrum shaping to the direct realization of Fig. 14.11*a* is illustrated in Fig. 14.11*b*. Signals and coefficients are assumed to be in fixed-point format using L bits for the magnitude and one bit for the sign, and each of the two adders A_1 and A_2 can add products of $2L$ bits to produce a sum of $2L$ bits. Quantizer Q_1 rounds the output of adder A_1 to L bits and simultaneously generates a scaled-up version of the quantization error which is fed back to adder A_1 through the β subnetwork. Quantizer Q_2, on the other hand, scales down and rounds the output of adder A_2 to $2L$ bits. A suitable scaling factor for the β subnetwork is 2^L since the leading L bits of the quantization error are zeros. Constant λ is used to scale the input of quantizer Q_1. Assuming L_2 signal scaling, then

$$\lambda = \frac{1}{\|H\|_2} \tag{14.67}$$

where $H(z)$ is the transfer function of the structure in Fig. 14.11*a*.

A noise model for the configuration in Fig. 14.11*b* can be readily obtained as shown in Fig. 14.11*c*, where $-q_i/2 \le e_i(n) \le q_i/2$ with $q_1 = 2^{-L}$ and $q_2 = 2^{-2L}$. Hence, the PSDs of signals $e_1(n)$ and $e_2(n)$ are given by

$$S_{e_i}(z) = \sigma_{e_i}^2 = \frac{q_i^2}{12}$$

As in Sec. 14.5, the PSD of the output noise can be obtained as

$$S_n(z) = \sum_{i=1}^{2} \frac{q_i^2}{12} H_i(z) H_i(z^{-1}) \tag{14.68}$$

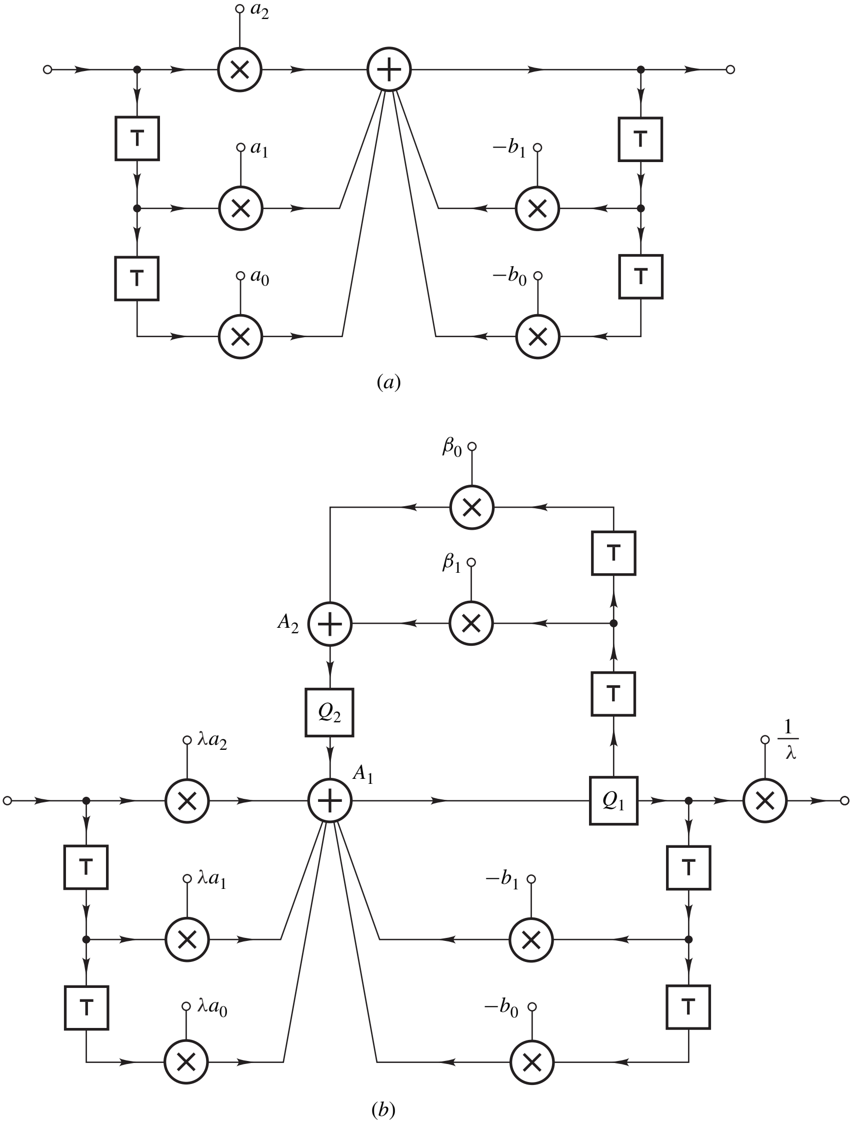
where

$$H_1(z) = \frac{1}{\lambda}\left(\frac{z^2 + \beta_1 z + \beta_0}{z^2 + b_1 z + b_0}\right) \tag{14.69}$$

and

$$H_2(z) = \frac{1}{\lambda(z^2 + b_1 z + b_0)} \tag{14.70}$$

are the transfer functions from noise sources $e_1(n)$ and $e_2(n)$ to the output, respectively. The output noise power is numerically equal to the autocorrelation of the output noise evaluated at $k = 0$ and



(*a*)

(*b*)

Figure 14.11 (*a*) Second-order direct realization, (*b*) application of error-spectrum shaping.

from Eqs. (13.19a) and (14.68), we obtain

$$
\begin{aligned}
r_n(0) = \sigma_n^2 &= \frac{1}{2\pi j} \oint_\Gamma S_n(z) z^{-1}\, dz \\
&= \frac{1}{2\pi j} \oint_\Gamma \sum_{i=1}^{2} \frac{q_i^2}{12} H_i(z) H_i(z^{-1}) z^{-1}\, dz \\
&= \sum_{i=1}^{2} \frac{q_i^2}{12} \| H_i \|_2^2 \qquad (14.71)
\end{aligned}
$$

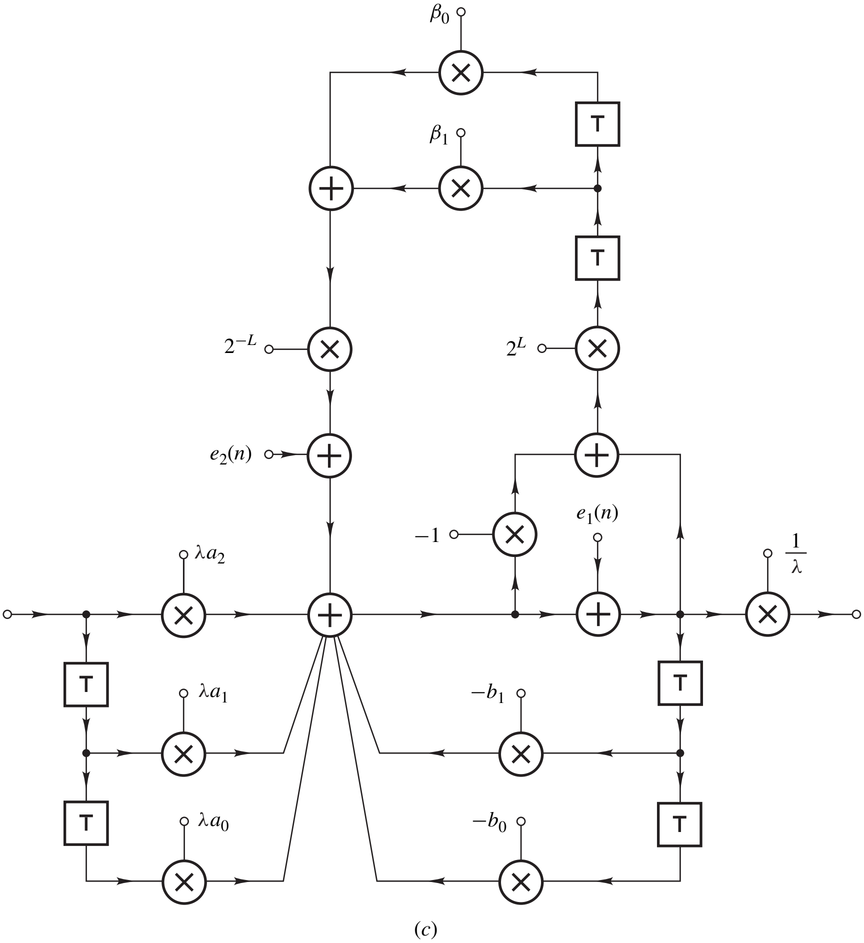


Figure 14.11 Cont'd (*c*) Noise model.

For a random input signal whose amplitude is uniformly distributed in the range (−1, 1), we have $r_x(k) = \sigma_x^2 = 1/3$; hence the output power due to the signal is given by

$$r_y(0) = \sigma_y^2 = \frac{1}{3}\|H\|_2^2 \tag{14.72}$$

Now from Eq. (14.67) and Eqs. (14.69)–(14.72), the signal-to-noise ratio can be obtained as

$$\text{SNR} = \frac{\sigma_y^2}{\sigma_n^2} = \frac{4 \times 2^{2L}}{\left\|\dfrac{z^2+\beta_1 z+\beta_0}{z^2+b_1 z+b_0}\right\|_2^2 + 2^{-2L}\left\|\dfrac{1}{z^2+b_1 z+b_0}\right\|_2^2}$$

If the parameters β_1 and β_2 are chosen to be equal to b_1 and b_2, respectively, then the signal-to-noise ratio is maximized, as demonstrated by Higgins and Munson [19].

Expressions for the coefficients of the error-spectrum shaping network for the case of cascade structures have been derived in [20].

14.9 LIMIT-CYCLE OSCILLATIONS

In the methods of analysis presented in Sec. 14.5, we made the fundamental assumption that signal levels are much larger than the quantization step throughout the filter. This allowed us to assume statistically independent noise signals from sample to sample and from source to source. On many occasions, signal levels can become very low or constant, at least for short periods of time, e.g., during pauses in speech and music signals. Under such circumstances, quantization errors tend to become highly correlated and can actually cause a filter to lock in an unstable mode whereby a steady output oscillation is generated. This phenomenon is known as the *deadband effect*, and the oscillation generated is commonly referred to as *quantization* or *granularity limit cycle*.

Quantization limit cycles are low-level oscillations whose amplitudes can be reduced by increasing the word length of the implementation. Another type of oscillation that can cause serious problems is sometimes brought about by overflow in the arithmetic devices used. Oscillations of this type are known as *overflow limit cycles* and their amplitudes can be quite large, sometimes as large as the maximum signal handling capacity of the hardware.

In this section, we examine the mechanisms by which quantization and overflow limit cycles can be generated and present methods for their elimination.

14.9.1 Quantization Limit Cycles

The deadband effect can be studied by using a technique developed by Jackson [21]. Consider the first-order filter of Fig. 14.12*a*. The transfer function and difference equation of the filter are given by

$$H(z) = \frac{H_0 z}{z - b}$$

and

$$y(n) = H_0 x(n) + b y(n-1) \tag{14.73}$$

respectively. The impulse response is

$$h(n) = H_0(b)^n$$

If $b = 1$ or -1, the filter is unstable and has an impulse response

$$h(n) = \begin{cases} H_0 & \text{for } b = 1 \\ H_0(-1)^n & \text{for } b = -1 \end{cases}$$

With $H_0 = 10.0$ and $b = -0.9$, the exact impulse response given in the second column of Table 14.5 can be obtained.

Now, assume that the filter is implemented using fixed-point decimal arithmetic, where each product $by(n-1)$ is rounded to the nearest integer according to the rule

$$Q[|by(n-1)|] = \text{Int}\,[|by(n-1)| + 0.5] \tag{14.74}$$

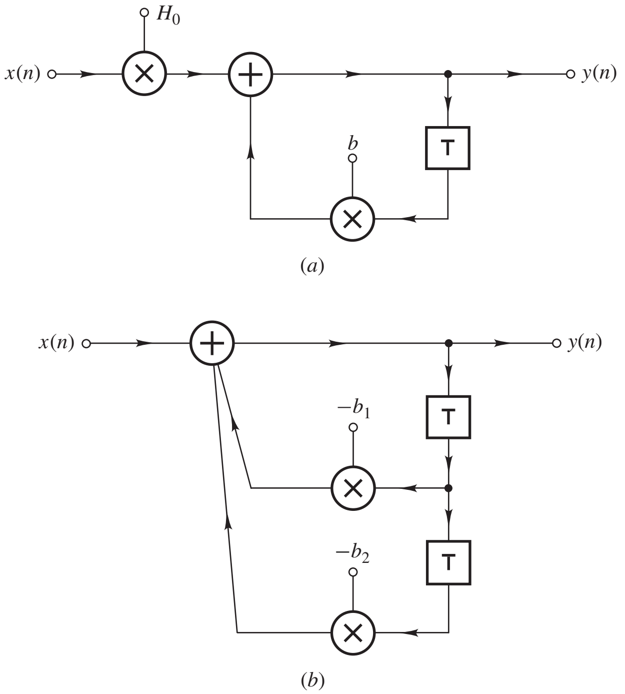


Figure 14.12 (*a*) First-order filter, (*b*) second-order filter.

With $H_0 = 10.0$ and $b = -0.9$, the response in the third column of Table 14.5 is obtained. As can be seen, for $n \geq 5$ the response oscillates between $+5$ and -5 and, in a sense, quantization has rendered the filter unstable.

If Eq. (14.73) is assumed to hold during the unstable mode, the effective value of b must be 1 for $b > 0$ or -1 for $b < 0$. If this is the case

$$Q[|by(n-1)|] = |y(n-1)|$$

and from Eq. (14.74)

$$\text{Int}\,[|b| \cdot |y(n-1)| + 0.5] = |y(n-1)|$$

or

$$\text{Int}\,[|y(n-1)| - (1-|b|)|y(n-1)| + 0.5] = |y(n-1)|$$

This equation can be satisfied if

$$0 \leq -(1-|b|)|y(n-1)| + 0.5 < 1$$

and by using the left-hand inequality, we conclude that

$$|y(n-1)| \leq \frac{0.5}{1-|b|} = k$$

Since $y(n-1)$ is an integer, instability cannot arise if $|b| < 0.5$. On the other hand, if $|b| \geq 0.5$, the response will tend to decay to zero once the input is removed, and eventually $y(n-1)$ will assume

Table 14.5 Impulse response of first-order filter

n	$h(n)$	$Q[h(n)]$
0	10.0	10.0
1	−9.0	−9.0
2	8.1	8.0
3	−7.29	−7.0
4	6.561	6.0
5	−5.9049	−5.0
6	5.31441	5.0
7	−4.782969	−5.0
⋮	⋮	⋮
100	2.65614×10^{-4}	5.0

values in the so-called *deadband range* $[-k, k]$. When this happens, the filter will become unstable. Any tendency of $|y(n-1)|$ to exceed k will restore stability, but in the absence of an input signal the response will again decay to a value within the deadband. Thus the filter will lock into a limit cycle of amplitude equal to or less than k. Since the effective value of b is $+1$ for $0.5 \leq b < 1$ or -1 for $-1 < b \leq -0.5$, the frequency of the limit cycle will be 0 or $\omega_s/2$.

For the second-order filter of Fig. 14.12*b*, we have

$$H(z) = \frac{z^2}{z^2 + b_1 z + b_0}$$

and

$$y(n) = x(n) - b_1 y(n-1) - b_0 y(n-2) \tag{14.75}$$

If the poles are complex, then

$$h(n) = \frac{r^n}{\sin\theta} \sin[(n+1)\theta]$$

where

$$r = \sqrt{b_0}$$

and

$$\theta = \cos^{-1} - \frac{b}{2\sqrt{b_0}}$$

For $b_0 = 1$, the impulse response is a sinusoid with constant amplitude and frequency

$$\omega_0 = \frac{1}{T}\cos^{-1} - \frac{b_1}{2} \tag{14.76}$$

This is sometimes referred to as the *resonant frequency* of the filter.

In second-order filters, there are two distinct limit-cycle modes. In one mode, a limit cycle with frequency 0 or $\omega_s/2$ is generated, and a limit cycle whose frequency is related to the resonant frequency ω_0 is generated in the other.

If the filter is implemented using fixed-point decimal arithmetic and each of the products $b_1 y(n-1)$ and $b_0 y(n-2)$ is rounded to the nearest integer according to the rule in Eq. (14.74), then Eq. (14.75) yields

$$y(n) = x(n) - Q[b_1 y(n-1)] - Q[b_0 y(n-2)]$$

The filter can sustain a zero-input limit cycle of amplitude y_0 ($y_0 > 0$) and frequency 0 or $\omega_s/2$ if

$$y_0 = \pm Q[b_1 y_0] - Q[b_0 y_0] \tag{14.77}$$

where the plus sign applies for limit cycles of frequency $\omega_s/2$ (see Prob. 14.29). Regions of the (b_0, b_1) plane that satisfy this equation and the corresponding values of y_0 are shown in Fig. 14.13*a*. The domain inside the triangle represents stable filters, as can be easily shown (see Eqs. (14.91a) and (14.91c)).

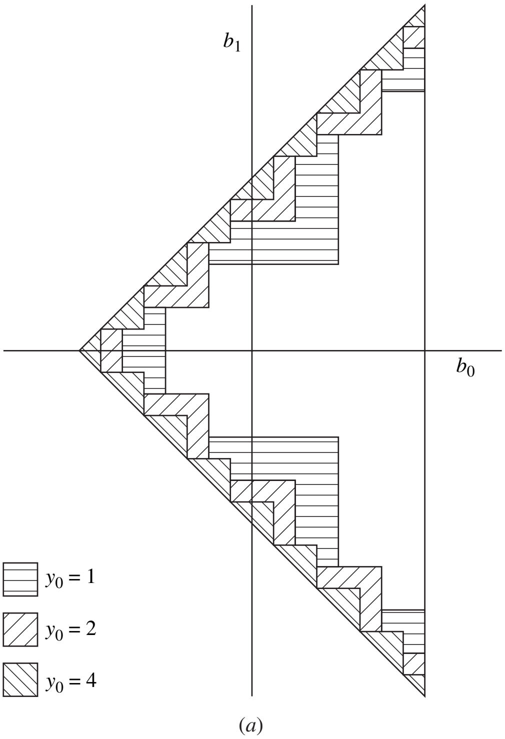


Figure 14.13 Regions of the (b_0, b_1) plane that yield quantization limit cycles: (*a*) Regions that satisfy Eq. (14.77).

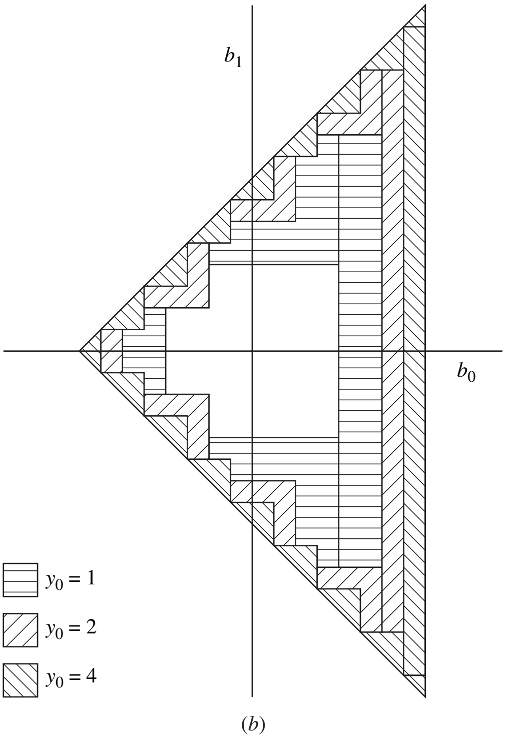


Figure 14.13 Cont'd Regions of the (b_0, b_1) plane that yield quantization limit cycles: (b) Regions that satisfy Eqs. (14.77) and (14.78).

If $e_1(n)$ and $e_2(n)$ are the quantization errors in products $b_1 y_0$ and $b_0 y_0$, respectively, then Eq. (14.77) gives

$$\pm b_1 = \frac{y_0 \pm e_1(n) \pm e_2(n)}{y_0} + b_0$$

and since $-0.5 < e_i(n) \leq 0.5$, a necessary but not sufficient condition for the existence of a limit cycle of frequency 0 or $\omega_s/2$ is obtained as

$$|b_1| \geq \frac{y_0 - 1}{y_0} + b_0$$

The second limit-cycle mode involves the quantization of product $b_0 y(n-2)$. If

$$Q[|b_0 y(n-2)|] = |y(n-2)|$$

then the effective value of b_0 is unity and, as in the first-order case, a condition for the existence of limit cycles can be deduced as

$$|y(n-2)| \leq \frac{0.5}{1 - |b_0|} = k \tag{14.78}$$

With k an integer, values of b_0 in the ranges

$$0.5 \leq |b_0| < 0.75$$

$$0.75 < |b_0| < 0.833$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\frac{2k-1}{2k} \leq |b_0| < \frac{2k+1}{2(k+1)}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

will yield deadbands $[-1, 1]$, $[-2, 2]$, ..., $[-k, k]$, ..., respectively. Regions of the (b_0, b_1) plane that satisfy both Eqs. (14.77) and (14.78) are depicted in Fig. 14.13b.

If the poles are close to the unit circle, the limit cycle is approximately sinusoidal with a frequency close to the resonant frequency given by Eq. (14.76).

For signed-magnitude binary arithmetic, Eq. (14.78) becomes

$$|y(n-2)| \leq \frac{q}{2(1-|b_0|)}$$

where q is the quantization step.

14.9.2 Overflow Limit Cycles

In one's- or two's- complement fixed-point implementations, the transfer characteristic of adders is periodic, as illustrated in Fig. 14.14a; as a consequence, if the inputs to an adder are sufficiently

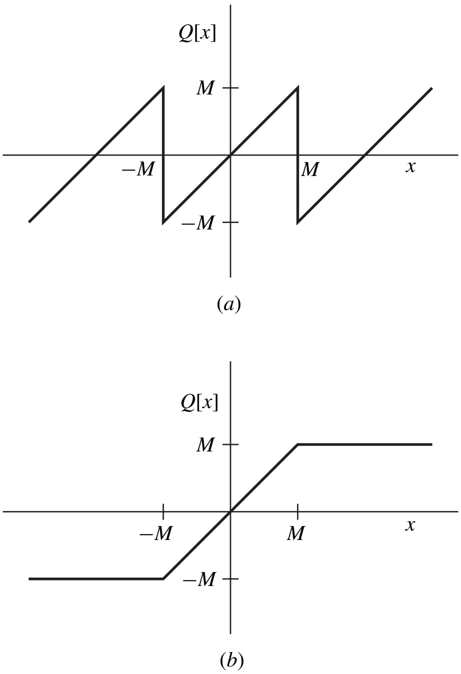


Figure 14.14 (a) Transfer characteristic of one's- or two's-complement fixed-point adder, (b) transfer characteristic of adder incorporating saturation mechanism.

large to cause overflow, unexpected results can occur. Under certain circumstances, oscillations of large amplitude can be sustained, which are known as *overflow limit-cycle oscillations*. These were identified and studied quite early in the development of digital filters by Ebert, Mazo, and Taylor [22]. The generation of overflow limit cycles is demonstrated by the following example.

Example 14.5 A second-order digital filter characterized by Eq. (14.75) with $b_1 = -1.375$ and $b_0 = 0.625$ is implemented in terms of two's-complement fixed-point arithmetic using a word length of 6 bits, excluding the sign bit. The quantization of products is carried out by rounding. Show that if $x(n) = 0$, $y(-2) = -43/64$, and $y(-1) = 43/64$, the filter will sustain an overflow limit cycle.

■ **Solution**

Using the difference equation, output $y(n)$ given in column 2 of Table 14.6 can be readily computed. Evidently, $y(4) = y(-2)$ and $y(5) = y(-1)$ and, therefore, a sustained oscillation of amplitude 43/64 and frequency $\omega_s/2$ will be generated.

Table 14.6 Overflow limit cycle in second-order filter

n	$64y(n)$	$64\tilde{y}(n)$
−2	−43	−43
−1	43	43
0	−42	63
1	43	60
2	−43	44
3	42	23
4	−43	4
5	43	− 8
6	−42	−14
7	43	−14
8	−43	−10
9	42	− 5
10	−43	− 3
11	43	− 1
12	−42	1
13	43	2
14	−43	2
15	42	2

14.9.3 Elimination of Quantization Limit Cycles

Quantization limit-cycle oscillations received considerable attention from researchers in the past, and two general approaches for minimizing or eliminating their effects have evolved. One approach entails the use of a sufficiently large signal word length to ensure that the amplitude of the limit-cycle is small enough to meet some system specification imposed by the application. Bounds on the limit-cycle amplitude that can be used in this approach have been deduced by Sandberg and Kaiser [23], Long and Trick [24], and Green and Turner [25]. The other approach entails the elimination of limit cycles altogether. Quantization limit cycles can be eliminated by using appropriate signal quantization schemes in specific structures, whereas overflow limit cycles can be eliminated by incorporating suitable saturation mechanisms in arithmetic devices.

An important method for the elimination of zero-input limit cycles was proposed by Meerkötter [26] and was later used by Mills, Mullis, and Roberts [27], and Vaidyanathan and Liu [28] to show that there are several realizations that support the elimination of limit-cycle oscillations. In this method, a Lyapunov function related to the stored power is constructed and is then used to demonstrate that under certain conditions limit cycles cannot be sustained. The principles involved are as follows.

Consider the digital filter shown in Fig. 14.15 and assume that block A is a linear subnetwork containing adders, multipliers, and interconnections but no unit delays. Further, assume that signal quantization and overflow control are carried out by quantizers Q_k for $k = 1, 2, \ldots, N$ placed at the inputs of the unit delays as shown. The state-space characterization of the filter can be expressed as

$$\mathbf{v}(n) = \mathbf{A}\mathbf{q}(\mathbf{n}) + \mathbf{b}x(n)$$

$$y(n) = \mathbf{c}^T\mathbf{q}(\mathbf{n}) + dx(n)$$

and if $x(n) = 0$, we can write

$$\mathbf{v}(n) = \mathbf{A}\mathbf{q}(\mathbf{n}) \tag{14.79}$$

$$\mathbf{q}(n+1) = \tilde{\mathbf{v}}(n) \tag{14.80}$$

where $\mathbf{A} = \{a_{ij}\}$ and $\tilde{v}_k(n)$ is related to $v_k(n)$ by some nonlinear and possibly time-varying functional relation of the form

$$\tilde{v}_k(n) = Q_k[v_k(n)] \qquad \text{for } k = 1, 2, \ldots, N \tag{14.81}$$

The quadratic form

$$p[\mathbf{q}(n)] = \mathbf{q}^T(n)\mathbf{D}\mathbf{q}(n) \tag{14.82}$$

where $\mathbf{D}$ is an $N \times N$ positive definite diagonal matrix, is related to the power stored in the unit delays at instant nT, and changes in this quantity can provide information about the stability of the

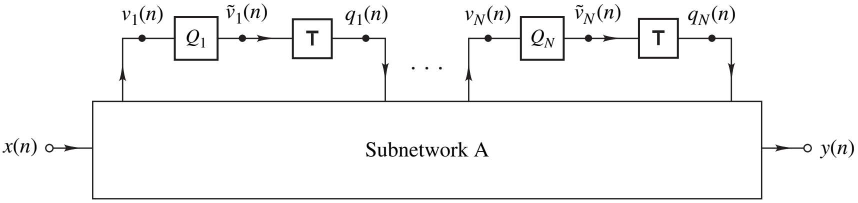


Figure 14.15 *N*th-order digital filter incorporating nonlinearities.

filter under zero-input conditions. The increase in $p[\mathbf{q}(n)]$ in one filter cycle can be expressed as

$$\Delta p[\mathbf{q}(n)] = p[\mathbf{q}(n+1)] - p[\mathbf{q}(n)] \tag{14.83}$$

and from Eqs. (14.80), (14.82), and (14.83), we have

$$\Delta p[\mathbf{q}(n)] = -\mathbf{q}^T(n)\mathbf{D}\mathbf{q}(n) + \tilde{\mathbf{v}}^T(n)\mathbf{D}\tilde{\mathbf{v}}(n) \tag{14.84}$$

Hence, Eqs. (14.79) and (14.84) yield

$$\begin{aligned}\Delta p[\mathbf{q}(n)] &= -\mathbf{q}^T(n)\mathbf{D}\mathbf{q}(n) + \tilde{\mathbf{v}}^T(n)\mathbf{D}\tilde{\mathbf{v}}(n) + [\mathbf{A}\mathbf{q}(n)]^T\mathbf{D}[\mathbf{A}\mathbf{q}(n)] - \mathbf{v}^T(n)\mathbf{D}\mathbf{v}(n)\\ &= -\mathbf{q}^T(n)(\mathbf{D} - \mathbf{A}^T\mathbf{D}\mathbf{A})\mathbf{q}(n) - \sum_{k=1}^{N}[v_k^2(n) - \tilde{v}_k^2(n)]d_{kk} \qquad (14.85)\end{aligned}$$

where d_{kk} for $k = 1, 2, \ldots, N$ are the diagonal elements of $\mathbf{D}$.

Now if

$$\mathbf{q}^T(n)(\mathbf{D} - \mathbf{A}^T\mathbf{D}\mathbf{A})\mathbf{q}(n) \geq 0 \tag{14.86}$$

and signals $v_k(n)$ are quantized such that

$$|\tilde{v}_k(n)| \leq |v_k(n)| \qquad \text{for} \qquad k = 1, 2, \ldots, N \tag{14.87}$$

then Eq. (14.85) yields

$$\Delta p[\mathbf{q}(n)] \leq 0 \tag{14.88}$$

that is, the power stored in the unit delays cannot increase. Since a digital filter is a finite-state machine, signals $q_k(n)$ must after a finite number of filter cycles either become permanently zero or oscillate periodically. In the first case, there are no limit cycle oscillations. In the second case, at least one $q_k(n)$, say $q_l(n)$, must oscillate periodically. However, from Eq. (14.88), we conclude that the amplitude of the oscillation must decrease with each filter cycle by some fixed amount until $q_l(n)$ becomes permanently zero after a finite number of filter cycles. Therefore, Eq. (14.86) and the conditions in Eq. (14.87) constitute a sufficient set of conditions for the elimination of limit cycles. A realization satisfying Eq. (14.86) is said to *support the elimination of zero-input limit cycles*. The conditions in Eq. (14.87) can be imposed by quantizing the state variables using magnitude truncation.

For a stable filter, the magnitudes of the eigenvalues of $\mathbf{A}$ are less than unity and Eq. (14.86) is satisfied if a positive definite diagonal matrix $\mathbf{D}$ can be found such that matrix $\mathbf{D} - \mathbf{A}^T\mathbf{D}\mathbf{A}$ is positive semidefinite [27, 28]. For second-order filters, this condition is satisfied if

$$a_{12}a_{21} \geq 0 \tag{14.89a}$$

or

$$a_{12}a_{21} < 0 \qquad \text{and} \qquad |a_{11} - a_{22}| + \det(\mathbf{A}) \leq 1 \tag{14.89b}$$

There are quite a few realizations that support the elimination of zero-input limit cycles. Some examples are: normal state-space structures in which

$$\mathbf{A} = \begin{bmatrix} \alpha & -\beta \\ -\beta & \alpha \end{bmatrix}$$

with $\beta > 0$ [29–31]; realizations that minimize the output roundoff noise such as those in [15, 17] (see Sec. 14.7); and lattice realizations [28, 31, 32].

Example 14.6 The structure shown in Fig. 14.16 realizes the biquadratic transfer function

$$H(z) = \frac{z^2 + a_1 z + a_0}{z^2 + b_1 z + b_0}$$

where

$$a_1 = -(\alpha_1 + \alpha_2) \tag{14.90a}$$

$$a_0 = 1 + \alpha_1 - \alpha_2 \tag{14.90b}$$

$$b_1 = -(\beta_1 + \beta_2) \tag{14.90c}$$

$$b_0 = 1 + \beta_1 - \beta_2 \tag{14.90d}$$

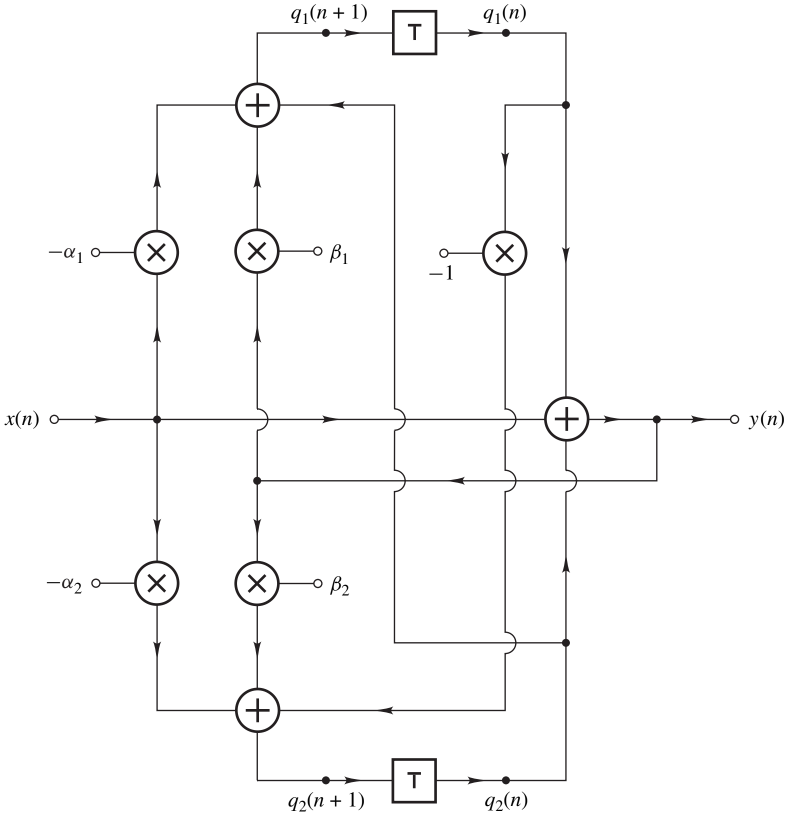


Figure 14.16 Biquadratic realization due to Meerkötter.

and is due to Meerkötter [26]. Show that the structure supports the elimination of zero-input limit cycles.

■ **Solution**

Straightforward analysis gives the state-space characterization of the structure as

$$\mathbf{q}(n+1) = \mathbf{A}\mathbf{q}(\mathbf{n}) + \mathbf{b}x(n)$$
$$y(n) = \mathbf{c}^T\mathbf{q}(\mathbf{n}) + dx(n)$$

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \beta_1 & (\beta_1+1) \\ (\beta_2-1) & \beta_2 \end{bmatrix}$$
$$\mathbf{b} = \begin{bmatrix} (\beta_1-\alpha_1) \\ (\beta_2-\alpha_2) \end{bmatrix} \qquad \mathbf{c} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \text{and} \qquad d = 1$$

The filter is stable if and only if

$$1 - b_0 > 0 \tag{14.91a}$$
$$1 + b_1 + b_0 > 0 \tag{14.91b}$$
$$1 - b_1 + b_0 > 0 \tag{14.91c}$$

as can be easily shown by using the Jury-Marden stability criterion (see Sec. 5.3.7. From Eq. (14.90), we can show that

$$a_{12}a_{21} = (\beta_1+1)(\beta_2-1) = \frac{1}{4}\left[b_1^2 - (1+b_0)^2\right]$$

and since $1 + b_0 > b_1$, according to Eq. (14.91c), we conclude that $a_{12}a_{21} < 0$. Hence, zero-input limit cycles can be eliminated by using magnitude truncation only if the condition in Eq. (14.89b) is satisfied. Simple manipulation now yields

$$|a_{11} - a_{22}| + \det(\mathbf{A}) = |b_0 - 1| + b_0 - 1 + 1 = 1$$

since $b_0 - 1$ is negative according to Eq. (14.91a); that is, Eq. (14.89b) is satisfied with the equal sign and, therefore, the structure supports the elimination of zero-input limit cycles. ■

Limit cycles can also be generated if the input assumes a constant value for a certain period of time. Limit cycles of this type, which include zero-input limit cycles as a special case, are referred to as *constant-input limit cycles*; they can be eliminated by using techniques described by Verkroost [33], Turner [34], and Diniz and Antoniou [35]. A state-space realization of the transfer function in Eq. (14.65) that supports the elimination of zero- and constant-input limit cycles is illustrated in

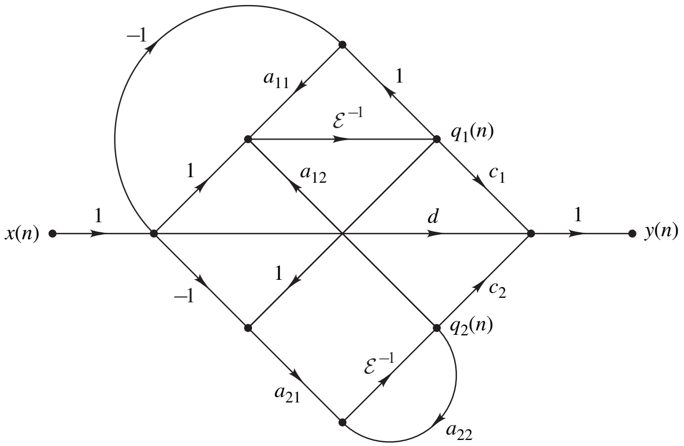


Figure 14.17 Second-order state-space realization that supports the elimination of zero- and constant-input limit cycles.

Fig. 14.17, where

$$a_{11} = a_{22} = -\beta_1/2 \tag{14.92a}$$

$$a_{12} = -\zeta/\sigma \qquad a_{21} = \sigma\zeta \tag{14.92b}$$

$$c_1 = \frac{\gamma_1 + \gamma_0}{1 + \beta_1 + \beta_0} \qquad c_2 = \frac{(2 + \beta_1)\gamma_0 - (\beta_1 + 2\beta_0)\gamma_1}{2\sigma\zeta(1 + \beta_1 + \beta_0)} \tag{14.92c}$$

$$d = \delta \tag{14.92d}$$

$$\zeta = \sqrt{\left(\beta_0 - \frac{\beta_1^2}{4}\right)}$$

Constant σ can be used to achieve optimal scaling. This structure is optimal or nearly optimal with respect to roundoff noise and is, in addition, slightly more economical than the state-space realization given by Eqs. (14.66a)–(14.66f) (see [35] for more details).

14.9.4 Elimination of Overflow Limit Cycles

Overflow limit cycles can be avoided to a large extent by applying strict scaling rules, e.g., using scaling method A in Sec. 14.6.1, to as far as possible prevent overflow from occurring. The problem with this approach is that signal levels throughout the filter are low; as a result, a poor signal-to-noise ratio is achieved. The preferred solution is to allow overflow on occasion but prevent the limit-cycle oscillations from occurring. A solution of this type reported in [22] involves incorporating a saturation mechanism in the design of adders so as to achieve a transfer characteristic of the type depicted in Fig. 14.14*b* where

$$Q[x] = \begin{cases} x & \text{if } |x| < M \\ M & \text{if } |x| \ge M \end{cases}$$

If this type of adder is used in the filter of Example 14.5, output $\tilde{y}(n)$ given in column 3 of Table 14.6 will be obtained. Evidently, the overflow limit cycle will be eliminated but a quantization limit cycle of amplitude 2/64 and frequency 0 will be present. This is due to the fact that this amplitude satisfies Eq. (14.77), as can be easily verified.

A concept that is closely related to overflow oscillations is the stability of the forced response of a nonlinear system or filter. If $\tilde{v}(n)$ and $v(n)$ are the state variables in Fig. 14.15, first with and then without the quantizers installed, the *forced response* of the filter is said to be stable if

$$\lim_{n\to\infty} [\tilde{v}(n) - v(n)] = 0$$

In practical terms, the stability of the forced response implies that transients due to overflow effects tend to die out once the cause of the overflow has been removed.

Claasen, Mecklenbräuker, and Peek [36] have shown that if a filter incorporating certain nonlinearities, e.g., overflow nonlinearities, is stable under zero-input conditions, then the forced response is also stable with respect to a corresponding set of nonlinearities. On the basis of this equivalence, if a digital filter of the type shown in Fig. 14.15 is stable under zero-input conditions, i.e., it satisfies Eq. (14.86) subject to the conditions in Eq. (14.87), then the forced response is also stable provided that the nonlinearities in Eq. (14.81) satisfy the conditions

$$\begin{aligned} 2 - x < Q_k[x] &\le 1 \qquad \text{for } 1 < x < 3 \\ -2 - x > Q_k[x] &\ge -1 \qquad \text{for } -3 < x < -1 \\ -1 \le Q_k[x] &\le 1 \qquad \text{for } |x| \ge 3 \end{aligned}$$

for $k = 1, 2, \ldots, N$, as illustrated in Fig. 14.18.

The stability of the forced response implies freedom from overflow limit cycles. It should be mentioned, however, that Claasen et al. deduced the above equivalence on the assumption that there is an infinite time separation between successive occurrences of overflow. Consequently, the above conditions may not guarantee the absence of overflow limit cycles if overflow occurs while the filter is recovering from a previous overflow.

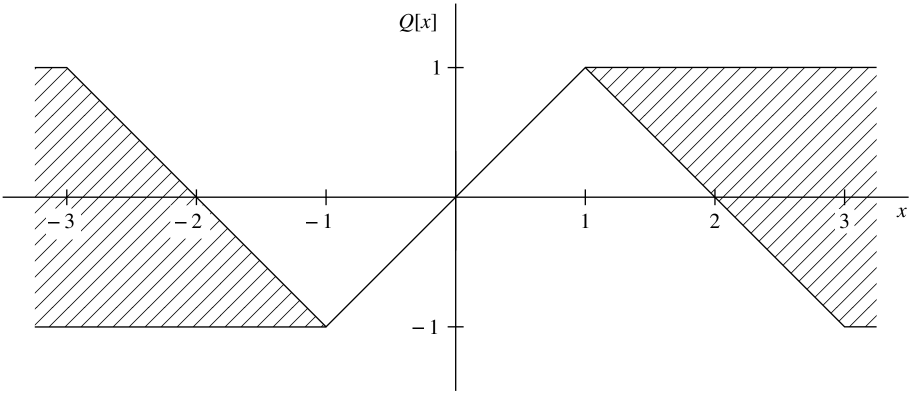


Figure 14.18 Transfer characteristic that guarantees the stability of the forced response.

REFERENCES

[1] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, New York: Oxford University Press, 2000.

[2] E. Avenhaus, "On the design of digital filters with coefficients of limited word length," *IEEE Trans. Audio Electroacoust.*, vol. 20, pp. 206–212, Aug. 1972.

[3] R. E. Crochiere, "A new statistical approach to the coefficient word length problem for digital filters," *IEEE Trans. Circuits Syst.*, vol. 22, pp. 190–196, Mar. 1975.

[4] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York: McGraw-Hill, 1991.

[5] R. E. Crochiere and A. V. Oppenheim, "Analysis of linear digital networks," *Proc. IEEE*, vol. 63, pp. 581–595, Apr. 1975.

[6] A. Antoniou, *Digital Filters: Analysis, Design, and Applications*, New York: McGraw-Hill, 1993.

[7] W. K. Jenkins and B. J. Leon, "An analysis of quantization error in digital filters based on interval algebras," *IEEE Trans. Circuits Syst.*, vol. 22, pp. 223–232, Mar. 1975.

[8] R. C. Agarwal and C. S. Burrus, "New recursive digital filter structures having very low sensitivity and roundoff noise," *IEEE Trans. Circuits Syst.*, vol. 22, pp. 921–927, Dec. 1975.

[9] P. S. R. Diniz and A. Antoniou, "Low-sensitivity digital-filter structures which are amenable to error-spectrum shaping," *IEEE Trans. Circuits Syst.*, vol. 32, pp. 1000–1007, Oct. 1985.

[10] S. Nishimura, K. Hirano, and R. N. Pal, "A new class of very low sensitivity and low roundoff noise recursive digital filter structures," *IEEE Trans. Circuits Syst.*, vol. 28, pp. 1152–1158, Dec. 1981.

[11] Y. V. Ramana Rao and C. Eswaran, "A pole-sensitivity based method for the design of digital filters for error-spectrum shaping," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1017–1020, July 1989.

[12] L. B. Jackson, "On the interaction of roundoff noise and dynamic range in digital filters," *Bell Syst. Tech. J.*, vol. 49, pp. 159–184, Feb. 1970.

[13] G. Bachman and L. Naria, *Functional Analysis*, New York: Academic, 1966.

[14] L. B. Jackson, "Roundoff-noise analysis for fixed-point digital filters realized in cascade or parallel form," *IEEE Trans. Audio Electroacoust.*, vol. 18, pp. 107–122, June 1970.

[15] C. T. Mullis and R. A. Roberts, "Synthesis of minimum roundoff noise fixed point digital filters," *IEEE Trans. Circuits Syst.*, vol. 23, pp. 551–562, Sept. 1976.

[16] S. Y. Hwang, "Roundoff noise in state-space digital filtering: A general analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, pp. 256–262, June 1976.

[17] L. B. Jackson, A. G. Lindgren, and Y. Kim, "Optimal synthesis of second-order state-space structures for digital filters," *IEEE Trans. Circuits Syst.*, vol. 26, pp. 149–153, Mar. 1979.

[18] T. Thong and B. Liu, "Error spectrum shaping in narrow-band recursive filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 25, pp. 200–203, Apr. 1977.

[19] W. E. Higgins and D. C. Munson, Jr., "Noise reduction strategies for digital filters: Error spectrum shaping versus the optimal linear state-space formulation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, pp. 963–973, Dec. 1982.

[20] W. E. Higgins and D. C. Munson, Jr., "Optimal and suboptimal error spectrum shaping for cascade-form digital filters," *IEEE Trans. Circuits Syst.*, vol. 31, pp. 429–437, May 1984.

[21] L. B. Jackson, “An analysis of limit cycles due to multiplication rounding in recursive digital filters,” *Proc. 7th Annu. Allerton Conf. Circuit Syst. Theory*, pp. 69–78, 1969.

[22] P. M. Ebert, J. E. Mazo, and M. G. Taylor, “Overflow oscillations in digital filters,” *Bell Syst. Tech. J.*, vol. 48, pp. 2999–3020, Nov. 1969.

[23] I. W. Sandberg and J. F. Kaiser, “A bound on limit cycles in fixed-point implementations of digital filters,” *IEEE Trans. Audio Electroacoust.*, vol. 20, pp. 110–114, June 1972.

[24] J. L. Long and T. N. Trick, “An absolute bound on limit cycles due to roundoff errors in digital filters,” *IEEE Trans. Audio Electroacoust.*, vol. 21, pp. 27–30, Feb. 1973.

[25] B. D. Green and L. E. Turner, “New limit cycle bounds for digital filters,” *IEEE Trans. Circuits Syst.*, vol. 35, pp. 365–374, Apr. 1988.

[26] K. Meerkötter, “Realization of limit cycle-free second-order digital filters,” *in Proc. IEEE Int. Symp. Circuits and Systems, 1976*, pp. 295–298.

[27] W. L. Mills, C. T. Mullis, and R. A. Roberts, “Digital filter realizations without overflow oscillations,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, pp. 334–338, Aug. 1978.

[28] P. P. Vaidyanathan and V. Liu, “An improved sufficient condition for absence of limit cycles in digital filters,” *IEEE Trans. Circuits Syst.*, vol. 34, pp. 319–322, Mar. 1987.

[29] C. M. Rader and B. Gold, “Effects of parameter quantization on the poles of a digital filter,” *Proc. IEEE*, vol. 55, pp. 688–689, May 1967.

[30] C. W. Barnes and A. T. Fam, “Minimum norm recursive digital filters that are free of overflow limit cycles,” *IEEE Trans. Circuits Syst.*, vol. 24, pp. 569–574, Oct. 1977.

[31] A. H. Gray, Jr. and J. D. Markel, “Digital lattice and ladder filter synthesis,” *IEEE Trans. Audio Electroacoust.*, vol. 21, pp. 491–500, Dec. 1973.

[32] A. H. Gray, Jr., “Passive cascaded lattice digital filters,” *IEEE Trans. Circuits Syst.*, vol. 27, pp. 337–344, May 1980.

[33] G. Verkroost, “A general second-order digital filter with controlled rounding to exclude limit cycles for constant input signals,” *IEEE Trans. Circuits Syst.*, vol. 24, pp. 428–431, Aug. 1977.

[34] L. E. Turner, “Elimination of constant-input limit cycles in recursive digital filters using a generalised minimum norm,” *Proc. Inst. Elect. Eng.*, Part G, vol. 130, pp. 69–77, June 1983.

[35] P. S. R. Diniz and A. Antoniou, “More economical state-space digital-filter structures which are free of constant-input limit cycles,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, pp. 807–815, Aug. 1986.

[36] T. A. C. M. Claasen, W. F. G. Mecklenbräuker, and J. B. H. Peek, “On the stability of the forced response of digital filters with overflow nonlinearities,” *IEEE Trans. Circuits Syst.*, vol. 22, pp. 692–696, Aug. 1975.

PROBLEMS

14.1. (*a*) Convert the decimal numbers

$$730.796875 \qquad \text{and} \qquad -3521.8828125$$

into binary representation.

(*b*) Convert the binary numbers

$$11011101.011101 \qquad \text{and} \qquad -100011100.1001101$$

into decimal representation.

14.2. Deduce the signed-magnitude, one's-complement, and two's-complement representations of (*a*) 0.810546875 and (*b*) −0.9462890625. Assume a word length $L = 10$.

14.3. The two's complement of a number x can be designated as

$$\tilde{x} = x_0.x_1x_2 \cdots x_L$$

(*a*) Show that

$$x = -x_0 + \sum_{i=1}^{L} x_i 2^{-i}$$

(*b*) Find x if $\tilde{x} = 0.1110001011$.

(*c*) Find x if $\tilde{x} = 1.1001110010$.

14.4. Assuming that $L = 7$, perform the following operations by using one's- and two's-complement additions.

(*a*) 0.6015625 – 0.4218750

(*b*) −0.359375 + (−0.218750)

14.5. The two's complement of x is given by

$$\tilde{x} = x_0.x_1x_2 \cdots x_L$$

(*a*) Show that

$$\text{Two's complement } (2^{-1}x) = \begin{cases} 2^{-1}\tilde{x} & \text{if } x_0 = 0 \\ 1 + 2^{-1}\tilde{x} & \text{if } x_0 = 1 \end{cases}$$

(*b*) Find the two's complement of $2^{-4}x$ if $\tilde{x} = 1.00110$.

14.6. (*a*) The register length in a fixed-point digital-filter implementation is 9 bits (including the sign bit), and the arithmetic is of the two's-complement type. Find the largest and smallest machine-representable decimal numbers.

(*b*) Show that the addition 0.8125 + 0.65625 will cause overflow.

(*c*) Show that the addition 0.8125 + 0.65625 + (−0.890625) will be evaluated correctly despite the overflow in the first partial sum.

14.7. The mantissa and exponent register segments in a floating-point implementation are 8 and 4 bits long, respectively.

(*a*) Deduce the register contents for −0.0234375, −5.0, 0.359375, and 11.5.

(*b*) Determine the dynamic range of the implementation.

Both mantissa and exponent are stored in signed-magnitude form.

14.8. A floating-point number

$$x = M \times 2^e \qquad \text{where } M = \sum_{i=1}^{B} b_{-i} 2^{-i}$$

is to be stored in a register whose mantissa and exponent segments comprise $L + 1$ and $e + 1$ bits, respectively. Assuming signed-magnitude representation and quantization by rounding, find the range of the quantization error.

14.9. A filter section is characterized by the transfer function

$$H(z) = H_0 \frac{(z+1)^2}{z^2 + b_1 z + b_0}$$

where

$$H_0 = -0.01903425 \qquad b_0 = 0.8638557 \qquad b_1 = -0.5596596$$

(*a*) Find the quantization error for each coefficient if signed-magnitude fixed-point arithmetic is to be used. Assume quantization by truncation and a word length $L = 6$ bits.

(*b*) Repeat part (*a*) if the quantization is to be by rounding.

14.10. (*a*) Realize the transfer function of Prob. 14.9 by using a canonic structure.

(*b*) The filter obtained in part (*a*) is implemented by using the arithmetic described in Prob. 14.9*a*. Plot the amplitude-response error versus frequency for $10 \le \omega \le 30$ rad/s. The sampling frequency is 100 rad/s.

(*c*) Repeat part (*b*), assuming quantization by rounding.

(*d*) Compare the results obtained in parts (*b*) and (*c*).

14.11. (*a*) The transfer function

$$H(z) = \frac{z^2 + 2z + 1}{z^2 + b_1 z + b_0} \qquad \text{where } b_1 = -r\sqrt{2} \qquad \text{and} \qquad b_0 = r^2$$

is to be realized by using the canonic structure of Fig. 14.7*b*. Find the sensitivities $S_{b_1}^H(z)$ and $S_{b_0}^H(z)$.

(*b*) The section is to be implemented by using fixed-point arithmetic, and the coefficient quantization is to be by rounding. Compute the statistical word length $L(\omega)$ for $0.7 \le r \le 0.95$ in steps of 0.05. Assume that $\Delta M_{\max}(\omega) = 0.02$, $x_1 = 2$ (see Sec. 14.3).

(*c*) Plot the statistical word length versus r and discuss the results achieved.

14.12. (*a*) Using Tables 14.2 and 14.3, obtain all possible low-sensitivity direct realizations of the transfer function in Prob. 14.9.

(*b*) The realizations in part (*a*) are to be implemented in terms of signed-magnitude fixed-point arithmetic using a word length $L = 6$, and quantization is to be by rounding. The sampling frequency is 100 rad/s. Plot the amplitude-response error versus frequency for $10 \le \omega \le 30$ rad/s for each realization.

(*c*) On the basis of the results in part (*b*), select the least sensitive of the possible realizations.

(*d*) Compare the realization selected in part (*c*) with the canonic realization obtained in Prob. 14.10.

14.13. The transfer function

$$H(z) = H_0 \prod_{i=1}^{3} \frac{(z+1)^2}{z^2 + b_{1i} z + b_{0i}}$$

where a_i, b_{0i}, and b_{1i} are given in Table P14.13, represents a lowpass Butterworth filter.

Table P14.13

i	b_{0i}	b_{1i}
1	$2.342170E-1$	$-9.459200E-1$
2	$3.753184E-1$	-1.054062
3	$7.148954E-1$	-1.314318
$H_0 = 5.796931E-4$		

(*a*) Realize the transfer function using three canonic sections in cascade.

(*b*) The realization in part (*a*) is to be implemented in terms of fixed-point signed-magnitude arithmetic using a word length $L = 8$ bits, and coefficient quantization is to be by rounding. The sampling frequency is 10^4 rad/s. Plot the amplitude-response error versus frequency for $0 \le \omega \le 10^3$ rad/s.

14.14. (*a*) Realize the transfer function in Prob. 14.13 using structure II-2 of Table 14.3.

(*b*) The realization in part (*a*) is to be implemented as in part (*b*) of Prob. 14.13. Plot the amplitude-response error versus frequency for $0 \le \omega \le 10^3$ rad/s.

(*c*) Compare the realization in part (*a*) with the cascade canonic realization of Prob. 14.13 with respect to sensitivity and the number of arithmetic operations.

14.15. The response of an A/D converter to a signal $x(t)$ is given by

$$y(n) = x(n) + e(n)$$

where $x(n)$ and $e(n)$ are random variables uniformly distributed in the ranges $-1 \leq x(n) \leq 1$ and $-2^{-(L+1)} \leq e(n) \leq 2^{-(L+1)}$, respectively.

(*a*) Find the signal-to-noise ratio. This is defined as

$$\text{SNR} = 10 \log \frac{\text{average signal power}}{\text{average noise power}}$$

(*b*) Find the PSD of $y(n)$ if $x(n)$, $e(n)$, $x(k)$, and $e(k)$ are statistically independent.

14.16. The filter section of Prob. 14.9 is to be scaled using the scheme in Fig. 14.8.

(*a*) Find λ for L_∞ scaling.

(*b*) Find λ for L_2 scaling using a frequency-domain method.

(*c*) Find λ for L_2 scaling using a time-domain method. (Hint: Use Parceval's discrete-time formula (Theorem 3.11))

(*d*) Compare the methods in parts (*b*) and (*c*).

(*e*) Compare the values of λ obtained with L_∞ and L_2 scaling and comment on the advantages and disadvantages of the two types of scaling.

14.17. The canonic realization of Prob. 14.13 is to be scaled according to the scheme in Fig. 14.9 using the L_∞ norm.

(*a*) Find the scaling constants λ_0, λ_1, and λ_2.

(*b*) The scaled realization is to be implemented in terms of fixed-point arithmetic and product quantization is to be by rounding. Plot the relative, output-noise PSD versus frequency. This is defined as

$$\text{RPSD} = 10 \log \frac{S_y(e^{j\omega T})}{S_e(e^{j\omega T})}$$

where $S_y(e^{j\omega T})$ is the PSD of output noise and $S_e(e^{j\omega T})$ is the PSD of a single noise source. The sampling frequency is 10^4 rad/s.

14.18. Repeat Prob. 14.17 using L_2 scaling and compare the results with those obtained in Prob. 14.17.

14.19. The low-sensitivity realization of Prob. 14.14 is to be scaled according to the scheme in Fig. 14.9 using the L_2 norm.

(*a*) Find the scaling constants λ_0, λ_1, and λ_2.

(*b*) The scaled realization is to be implemented in terms of fixed-point arithmetic and product quantization is to be by rounding. Plot the relative, output-noise PSD versus frequency.

14.20. The transfer function

$$H(z) = \prod_{i=1}^{3} \frac{a_{0i} z^2 + a_{1i} z + a_{0i}}{z^2 + b_{1i} z + b_{0i}}$$

where a_{0i}, a_{1i}, b_{0i} and b_{1i} are given in Table P14.16 represents a bandstop elliptic filter.

Table P14.16

i	a_{0i}	a_{1i}	b_{0i}	b_{1i}
1	$4.623281E-1$	$7.859900E-9$	$-7.534381E-2$	$7.859900E-9$
2	$4.879171E-1$	$5.904108E-2$	$8.051571E-1$	$8.883641E-1$
3	1.269926	$-1.536691E-1$	$8.051571E-1$	$-8.883640E-1$

(*a*) Realize the transfer function using three canonic sections in cascade.

(*b*) Determine the scaling constants. Assume the section order implied by the transfer function and use L_∞ scaling. The sampling frequency is 18 rad/s.

(*c*) Plot the relative output-noise PSD versus frequency.

14.21. The transfer function

$$H(z) = \prod_{i=1}^{3} \frac{a_{0i}z^2 + a_{1i}z + 1}{z^2 + a_{1i}z + a_{0i}}$$

where a_{0i} and a_{1i} are given in Table P14.17, represents a digital equalizer. Repeat parts (*a*) to (*c*) of Prob. 14.16. The sampling frequency is 2.4π rad/s.

Table P14.17

i	a_{0i}	a_{1i}
1	0.973061	−1.323711
2	0.979157	−1.316309
3	0.981551	−1.345605

14.22. Demonstrate the validity of Eq. (14.37).

14.23. Show that the column vectors $\mathbf{f}(z)$ and $\mathbf{g}(z)$ defined in Eq. (14.50) are given by the expressions in Eq. (14.51).

14.24. The vector $\mathbf{q}(n)$ in the state-space realization $\{\mathbf{A}, \mathbf{b}, \mathbf{c}^T, d\}$ is subjected to the transformation $\tilde{\mathbf{q}}(n) = \mathbf{T}\mathbf{q}(n)$.

(*a*) Show that the transformed realization $\{\tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}^T, \tilde{d}\}$ is given by Eq. (14.52.)

(*b*) Show that the transformed vectors $\tilde{\mathbf{f}}(z)$ and $\tilde{\mathbf{g}}(z)$ are given by Eq. (14.53).

14.25. (*a*) Obtain a state-space section-optimal realization of the lowpass filter in Prob. 14.13.

(*b*) Apply L_2 scaling to the realization.

(*c*) The scaled realization is to be implemented in terms of fixed-point arithmetic and product quantization is to be by rounding. Plot the relative, output-noise PSD versus frequency.

(*d*) Compare the results with those obtained in the case of the direct canonic realization in Prob. 14.18.

14.26. (*a*) Apply error-spectrum shaping to the scaled cascade canonic realization obtained in Prob. 11.13.

(*b*) The modified realization is to be implemented in terms of fixed-point arithmetic and product quantization is to be by rounding. Compute and plot the relative output-noise PSD versus frequency assuming the L_2 scaling obtained in Prob. 11.18.

(*c*) Compare the results with those obtained without error-spectrum shaping in Prob. 11.18.

14.27. A second-order filter characterized by Eq. (14.75) with $b_1 = -1.343503$ and $b_0 = 0.9025$ is to be implemented using signed-magnitude decimal arithmetic. Quantization is to be performed by rounding each product to the nearest integer, and $\omega_s = 2\pi$ rad/s.

(*a*) Estimate the peak-to-peak amplitude and frequency of the limit cycle by using Jackson's approach.

(*b*) Determine the actual amplitude and frequency of the limit cycle by simulation.

(*c*) Compare the results obtained in parts (*a*) and (*b*).

14.28. Repeat Prob. 14.27 for the coefficients $b_1 = -1.8$ and $b_0 = 0.99$.

14.29. A second-order filter represented by Eq. (14.75) is implemented in terms of fixed-point decimal arithmetic.

(*a*) Show that the filter can sustain zero-input limit cycles of amplitude y_0 ($y_0 > 0$) and frequency 0 or $\omega_s/2$ if Eq. (14.77) is satisfied.

(*b*) Find y_0 if $b_1 = -1.375$ and $b_0 = 0.625$.

14.30. The second-order realization shown in Fig. 4.5*c* can under certain conditions support the elimination of zero-input limit cycles. Deduce these conditions.

14.31. Show that the state-space realization of Eqs. (14.92a)–(14.92d) supports the elimination of zero-input limit cycles.

14.32. Realize the lowpass filter of Prob. 14.13 using Meerkötter's structure shown in Fig. 14.16.

14.33. Design a sinusoidal oscillator by using a digital filter in cascade with a bandpass filter. The frequency of oscillation is required to be $\omega_s/10$.

CHAPTER 15

DESIGN OF NONRECURSIVE FILTERS USING OPTIMIZATION METHODS

15.1 INTRODUCTION

The window method for the design of nonrecursive filters described in Chap. 9 is based on a closed-form solution and, as a result, it is easy to apply and entails a relatively insignificant amount of computation. Unfortunately, it usually leads to suboptimal designs whereby the filter order required to satisfy a set of given specifications is not the lowest that can be achieved. Consequently, the number of arithmetic operations required per output sample is not minimum, and the computational efficiency and speed of operation of the filter are not as high as could be.

This chapter deals with a method for the design of nonrecursive filters known as the *weighted-Chebyshev* method. In this method, an error function is formulated for the desired filter in terms of a linear combination of cosine functions and is then minimized by using a very efficient multivariable optimization algorithm known as the *Remez exchange algorithm*. When convergence is achieved, the error function becomes equiripple as in other types of Chebyshev solutions (see Sec. 10.4). The amplitude of the error in different frequency bands of interest is controlled by applying weighting to the error function.

The weighted-Chebyshev method is very flexible and can be used to obtain optimal solutions for most types of nonrecursive filters, e.g., digital differentiators, Hilbert transformers, and lowpass, highpass, bandpass, bandstop, and multiband filters with piecewise-constant amplitude responses. Furthermore, like the methods of Chaps. 12 and 16, it can be used to design filters with arbitrary amplitude responses. In common with other optimization methods, the weighted-Chebyshev method requires a large amount of computation; however, as the cost of computation is becoming progressively cheaper and cheaper with time, this disadvantage is not a very serious one.

The development of the weighted-Chebyshev method began with a paper by Herrmann published in 1970 [1], which was followed soon after by a paper by Hofstetter, Oppenheim, and Siegel [2]. These contributions were followed by a series of papers, during the seventies, by Parks, McClellan, Rabiner, and Herrmann [3–8]. These developments led, in turn, to the well-known McClellan-Parks-Rabiner computer program for the design of nonrecursive filters, documented in [9], which has found widespread applications. The approach to weighted-Chebyshev filters presented in this chapter is based on that reported in Refs. [3, 6, 8], and includes several enhancements proposed by the author in Refs. [10, 11].

15.2 PROBLEM FORMULATION

Consider a nonrecursive filter characterized by the transfer function

$$H(z) = \sum_{n=0}^{N-1} h(nT)z^{-n} \tag{15.1}$$

and assume that N is odd, the impulse response is symmetrical, and $\omega_s = 2\pi$. Since $T = 2\pi/\omega_s = 1$ s, the frequency response of the filter can be expressed as

$$H(e^{j\omega T}) = e^{-jc\omega}P_c(\omega)$$

where

$$\begin{aligned} P_c(\omega) &= \sum_{k=0}^{c} a_k \cos k\omega \qquad\qquad (15.2)\\ a_0 &= h(c) \\ a_k &= 2h(c-k) \qquad \text{for } k = 1, 2, \ldots, c \\ c &= (N-1)/2 \end{aligned}$$

(see Table 9.1).

If $e^{-jc\omega}D(\omega)$ is the desired frequency response and $W(\omega)$ is a weighting function, an error function $E(\omega)$ can be constructed as

$$E(\omega) = W(\omega)[D(\omega) - P_c(\omega)] \tag{15.3}$$

If $|E(\omega)|$ is minimized such that

$$|E(\omega)| \le \delta_p$$

with respect to some compact subset of the frequency interval $[0, \pi]$, say, $\mathbf{\Omega}$, a filter can be obtained in which

$$|E_0(\omega)| = |D(\omega) - P_c(\omega)| \le \frac{\delta_p}{|W(\omega)|} \qquad \text{for } \omega \in \mathbf{\Omega} \tag{15.4}$$

15.2.1 Lowpass and Highpass Filters

The amplitude response of an equiripple lowpass filter is of the form illustrated in Fig. 15.1, where δ_p and δ_a are the amplitudes of the passband and stopband ripples, and ω_p and ω_a are the passband and stopband edges, respectively. Hence, we require

$$D(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_p \\ 0 & \text{for } \omega_a \le \omega \le \pi \end{cases} \tag{15.5a}$$

with

$$|E_0(\omega)| \le \begin{cases} \delta_p & \text{for } 0 \le \omega \le \omega_p \\ \delta_a & \text{for } \omega_a \le \omega \le \pi \end{cases} \tag{15.5b}$$

Therefore, from Eqs. (15.4) and (15.5b), we can deduce the weighting function as

$$W(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_p \\ \delta_p/\delta_a & \text{for } \omega_a \le \omega \le \pi \end{cases} \tag{15.6}$$

Similarly, for highpass filters, we obtain

$$D(\omega) = \begin{cases} 0 & \text{for } 0 \le \omega \le \omega_a \\ 1 & \text{for } \omega_p \le \omega \le \pi \end{cases}$$

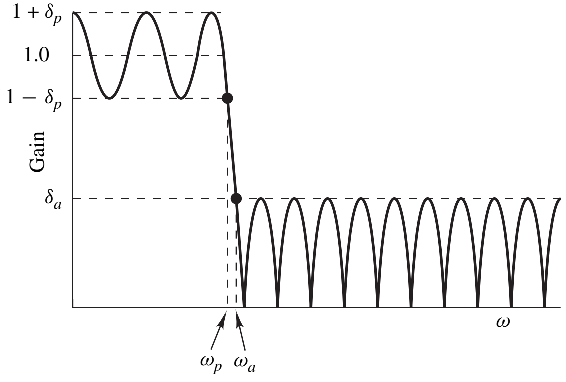


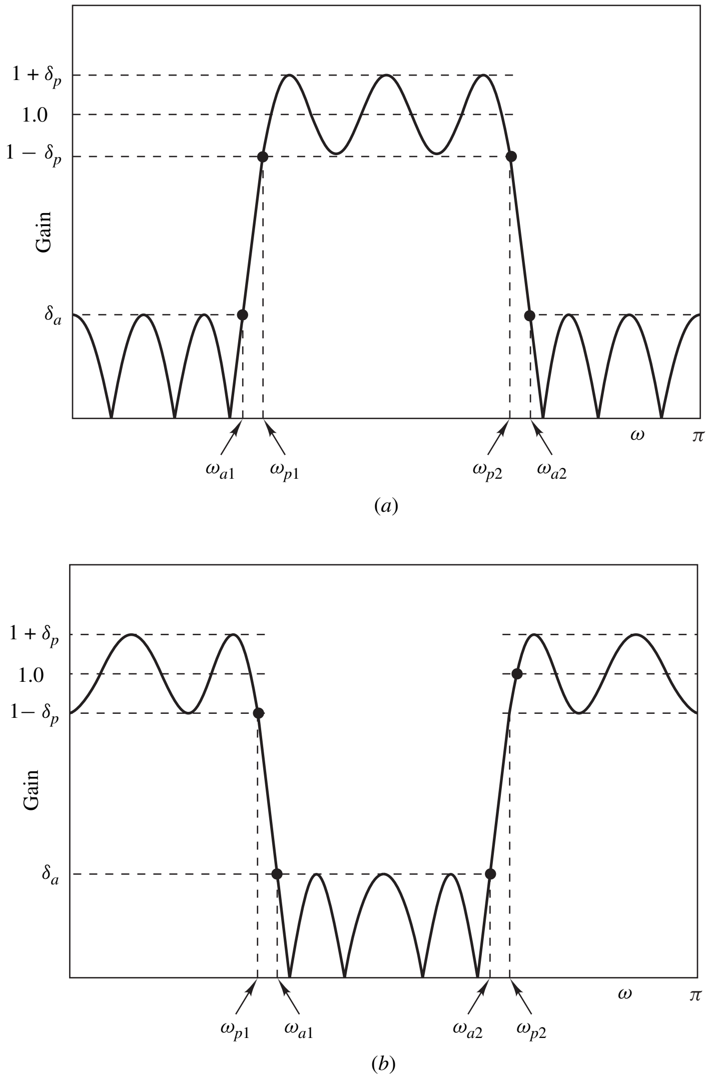
Figure 15.1 Amplitude response of an equiripple lowpass filter.

and

$$W(\omega) = \begin{cases} \delta_p/\delta_a & \text{for } 0 \le \omega \le \omega_a \\ 1 & \text{for } \omega_p \le \omega \le \pi \end{cases} \tag{15.7}$$

15.2.2 Bandpass and Bandstop Filters

The amplitude responses of equiripple bandpass and bandstop filters assume the forms illustrated in Fig. 15.2*a* and *b*, respectively, where δ_p and δ_a are the passband and stopband ripples, respectively,



(*a*)

(*b*)

Figure 15.2 Amplitude responses of equiripple filters: (*a*) Bandpass filter, (*b*) bandstop filter.

ω_{p1} and ω_{p2} are the passband edges, and ω_{a1} and ω_{a2} are the stopband edges. For bandpass filters

$$D(\omega) = \begin{cases} 0 & \text{for } 0 \le \omega \le \omega_{a1} \\ 1 & \text{for } \omega_{p1} \le \omega \le \omega_{p2} \\ 0 & \text{for } \omega_{a2} \le \omega \le \pi \end{cases}$$

$$W(\omega) = \begin{cases} \delta_p/\delta_a & \text{for } 0 \le \omega \le \omega_{a1} \\ 1 & \text{for } \omega_{p1} \le \omega \le \omega_{p2} \\ \delta_p/\delta_a & \text{for } \omega_{a2} \le \omega \le \pi \end{cases} \tag{15.8}$$

and for bandstop filters

$$D(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_{p1} \\ 0 & \text{for } \omega_{a1} \le \omega \le \omega_{a2} \\ 1 & \text{for } \omega_{p2} \le \omega \le \pi \end{cases}$$

$$W(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_{p1} \\ \delta_p/\delta_a & \text{for } \omega_{a1} \le \omega \le \omega_{a2} \\ 1 & \text{for } \omega_{p2} \le \omega \le \pi \end{cases} \tag{15.9}$$

15.2.3 Alternation Theorem

An effective approach for the solution of the optimization problem at hand is to solve the minimax problem

$$\underset{\mathbf{x}}{\text{minimize}} \;\; \{\max_{\omega} |E(\omega)|\} \tag{15.10}$$

where

$$\mathbf{x} = [a_0 \; a_1 \; \dots \; a_c]^T$$

The solution of this problem exists by virtue of the so-called *alternation theorem* [12] which is as follows:

Theorem 15.1 Alternation Theorem ***If $P_c(\omega)$ is a linear combination of $r = c + 1$ cosine functions of the form***

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega$$

then a necessary and sufficient condition that $P_c(\omega)$ be the unique, best, weighted-Chebyshev approximation to a continuous function $D(\omega)$ on Ω, where Ω is a compact subset of the frequency interval $[0, \pi]$, is that the weighted error function $E(\omega)$ exhibits at least $r + 1$

extremal frequencies ***in Ω, that is, there must exist at least r + 1 points $\hat{\omega}_i$ in Ω such that***

$$\hat{\omega}_0 < \hat{\omega}_1 < \ldots < \hat{\omega}_r$$
$$E(\hat{\omega}_i) = -E(\hat{\omega}_{i+1}) \qquad \textit{for } i = 0, 1, \ldots, r-1$$

and

$$|E(\hat{\omega}_i)| = \max_{\omega \in \Omega} |E(\omega)| \qquad \textit{for } i = 0, 1, \ldots, r \quad \blacktriangle$$

From the alternation theorem and Eq. (15.3), we can write

$$E(\hat{\omega}_i) = W(\hat{\omega}_i)[D(\hat{\omega}_i) - P_c(\hat{\omega}_i)] = (-1)^i \delta \tag{15.11}$$

for $i = 0, 1, \ldots, r$, where δ is a constant. This system of equations can be put in matrix form as

$$\begin{bmatrix} 1 & \cos\hat{\omega}_0 & \cos 2\hat{\omega}_0 & \cdots & \cos c\hat{\omega}_0 & \dfrac{1}{W(\hat{\omega}_0)} \\ 1 & \cos\hat{\omega}_1 & \cos 2\hat{\omega}_1 & \cdots & \cos c\hat{\omega}_1 & \dfrac{-1}{W(\hat{\omega}_1)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos\hat{\omega}_r & \cos 2\hat{\omega}_r & \cdots & \cos c\hat{\omega}_r & \dfrac{(-1)^r}{W(\hat{\omega}_r)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_c \\ \delta \end{bmatrix} = \begin{bmatrix} D(\hat{\omega}_0) \\ D(\hat{\omega}_1) \\ \vdots \\ D(\hat{\omega}_r) \end{bmatrix} \tag{15.12}$$

If the extremal frequencies (or extremals for short) were known, coefficients a_k and, in turn, the frequency response of the filter could be computed using Eq. (15.2). The solution of this system of equations exists since the above $(r+1) \times (r+1)$ matrix is known to be nonsingular [12].

15.3 REMEZ EXCHANGE ALGORITHM

The Remez exchange algorithm is an *iterative multivariable* algorithm which is naturally suited for the solution of the minimax problem in Eq. (15.10). It is based on the second optimization method of Remez [13] and involves the following basic steps:

Algorithm 1: Basic Remez exchange algorithm

1. Initialize extremals $\hat{\omega}_0, \hat{\omega}_1, \ldots, \hat{\omega}_r$ and ensure that an extremal is assigned at each band edge.
2. Locate the frequencies $\overset{\frown}{\omega}_0, \overset{\frown}{\omega}_1, \ldots, \overset{\frown}{\omega}_\rho$ at which $|E(\omega)|$ is maximum and $|E(\overset{\frown}{\omega}_i)| \geq \delta$. These frequencies are *potential* extremals for the next iteration.
3. Compute the convergence parameter

$$Q = \frac{\max|E(\overset{\frown}{\omega}_i)| - \min|E(\overset{\frown}{\omega}_i)|}{\max|E(\overset{\frown}{\omega}_i)|} \tag{15.13}$$

   where $i = 0, 1, \ldots, \rho$.
4. Reject $\rho - r$ *superfluous* potential extremals $\overset{\frown}{\omega}_i$ according to an appropriate rejection criterion and renumber the remaining $\overset{\frown}{\omega}_i$ sequentially; then set $\hat{\omega}_i = \overset{\frown}{\omega}_i$ for $i = 0, 1, \ldots, r$.

5. If $Q > \varepsilon$, where ε is a convergence tolerance (say $\varepsilon = 0.01$), repeat from step 2; otherwise continue to step 6.

6. Compute $P_c(\omega)$ using the last set of extremals; then deduce $h(n)$, the impulse response of the required filter, and stop. ●

The amount of computation required by the algorithm tends to depend quite heavily on the initialization scheme used in step 1, the search method used for the location of the maxima of the error function in step 2, and the criterion used to reject superfluous frequencies $\widehat{\omega}_i$ in step 4.

15.3.1 Initialization of Extremals

The simplest scheme for the initialization of extremals $\hat{\omega}_i$ for $i = 0, 1, \ldots, r$ is to assume that they are uniformly spaced in the frequency bands of interest. If there are J distinct bands in the required filter of widths $B_1, B_2, \ldots, B_J$ and extremals are to be located at the left-hand and right-hand band edges of each band, the total bandwidth, that is, $B_1 + B_2 + \cdots + B_J$, should be divided into $r + 1 - J$ intervals. Under these circumstances, the average interval between adjacent extremals is

$$W_0 = \frac{1}{r + 1 - J} \sum_{j=1}^{J} B_j$$

Since the quantities B_j / W_0 need not be integers, the use of W_0 for the generation of the extremals will almost always result in a fractional interval in each band. This problem can be avoided by rounding the number of intervals B_j / W_0 to the nearest integer and then readjusting the frequency interval for the corresponding band accordingly. This can be achieved by letting the numbers of intervals in bands j and J be

$$m_j = \text{Int}\left(\frac{B_j}{W_0} + 0.5\right) \qquad \text{for } j = 1, 2, \ldots, J - 1 \tag{15.14a}$$

and

$$m_J = r - \sum_{j=1}^{J-1}(m_j + 1) \tag{15.14b}$$

respectively, and then recalculating the frequency intervals for the various bands as

$$W_j = \frac{B_j}{m_j} \qquad \text{for } j = 1, 2, \ldots, J \tag{15.15}$$

A more sophisticated initialization scheme which was found to give good results is described in Ref. [14].

15.3.2 Location of Maxima of the Error Function

The frequencies $\widehat{\omega}_i$, which *must include maxima at band edges* if $|E(\widehat{\omega}_i)| \geq |\delta|$, can be located by simply evaluating $|E(\omega)|$ over a dense set of frequencies. A reasonable number of frequency points that yields sufficient accuracy in the determination of the frequencies $\widehat{\omega}_i$ is $8(N + 1)$. This corresponds to about 16 frequency points per ripple of $|E(\omega)|$. A suitable frequency interval for the jth band is $w_j = W_j/S$ with $S = 16$.

The above *exhaustive* step-by-step search can be implemented in terms of Algorithm 2 below where ω_{Lj} and ω_{Rj} are the left-hand and right-hand edges in band j; W_j is the interval between adjacent extremals and m_j is the number of intervals W_j in band j; w_j is the interval between successive samples of $|E(\omega)|$ in interval W_j and S is the number of intervals w_j in each interval W_j; N_j is the total number of intervals w_j in band j; and J is the number of bands.

Algorithm 2: Exhaustive step-by-step search

```
1. Set $N_j = m_j S$, $w_j = B_j/N_j$, and $e = 0$.
2. For each of bands $1, 2, \ldots, j, \ldots, J$ do:
    For each of frequencies $\omega_{1j} = \omega_{Lj}, \omega_{2j} = \omega_{Lj} + w_j, \ldots, \omega_{ij} = \omega_{Lj} + (i-1)w_j, \ldots, \omega_{(N_j+1)j} = \omega_{Rj}$, set $\widehat{\omega}_e = \omega_{ij}$ and $e = e + 1$ provided that $|E(\omega_{ij})| \geq |\delta|$ and one of the following conditions holds:
        (a) Case $\omega_{ij} = \omega_{Lj}$: if $|E(\omega_{ij})|$ is maximum at $\omega_{ij} = \omega_{Lj}$ (i.e., $|E(\omega_{Lj})| > |E(\omega_{Lj} + \varepsilon)|$);
        (b) Case $\omega_{Lj} < \omega_{ij} < \omega_{Rj}$: if $|E(\omega)|$ is maximum at $\omega = \omega_{ij}$ (i.e., $|E(\omega_{ij} - w_j)| < |E(\omega_{ij})| > |E(\omega_{ij} + w_j)|$);
        (c) Case $\omega_{ij} = \omega_{Rj}$: if $|E(\omega_{ij})|$ is maximum at $\omega_{ij} = \omega_{Rj}$ (i.e., $|E(\omega_{Rj})| > |E(\omega_{Rj} - \varepsilon)|$. ●
```

The parameter ε in steps 2(a) and 2(c) is a small positive constant and a value $10^{-2}w_j$ was found to yield satisfactory results.

In practice, $|E(\omega)|$ is maximum at an interior left-hand band edge[1] if its first derivative at the band edge is negative, and a mirror-image situation applies at an interior right-hand band edge. In such cases, $|E(\omega)|$ has a zero immediately to the right or left of the band edge and the inequality in step 2(a) or 2(c) may sometimes fail to identify a maximum. However, the problem can be avoided by using the inequality $|E(\omega_{Lj} - \varepsilon)| > |E(\omega_{Lj})|$ in step 2(a) and $|E(\omega_{Rj})| < |E(\omega_{Rj} + \varepsilon)|$ in step 2(c) for interior band edges. An alternative approach to the problem is to use gradient information based on the formulas given in Sec. 15.6.

In rare circumstances, a maximum of $|E(\omega)|$ may occur between a band edge and the first sample point. Such a maximum may be missed by Algorithm 2 but the problem can be easily identified since the number of potential extremals will then be less than the minimum. The remedy is to check the number of potential extremals at the end of each iteration and if it is found to be less than $r + 1$, the density of sample points, i.e., S, is doubled and the iteration is repeated. If the problem persists, the process is repeated until the required number of potential extremals is obtained. If a value of S equal to or less than 256 does not resolve the problem, the loss of potential extremals is most likely due to some other reason.

An important precaution in the implementation of the preceding as well as the subsequent search methods is to ensure that extremals belong to the dense set of frequency points to avoid numerical ill-conditioning in the computation of $E(\omega)$ (see Eqs. (15.11) and (15.17)). In addition, the condition $|E(\omega_{ij})| \geq |\delta|$ should be replaced by $|E(\omega_{ij})| > |\delta| - \varepsilon_1$, where ε_1 is a small positive constant, say, 10^{-6}, to ensure that no maxima are missed owing to roundoff errors.

[1] *An interior band edge is one in the range* $0 < \omega < \pi$, *that is, not at* $\omega = 0$ *or* π.

The search method is very reliable and its use in Algorithm 1 leads to a *robust* algorithm since the entire frequency axis is searched using a dense set of frequency points. Its disadvantage is that it requires a considerable amount of computation and is, therefore, inefficient. Improved search methods will be considered in Sec. 15.4.

A more efficient version of Algorithm 2 is obtained by maintaining all the interior band edges as extremals throughout the optimization independently of the behavior of the error function at the band edges. However, the algorithm obtained tends to fail more frequently than Algorithm 2.

15.3.3 Computation of $|E(\omega)|$ and $P_c(\omega)$

In steps 2 and 6 of the basic Remez algorithm (Algorithm 1), $|E(\omega)|$ and $P_c(\omega)$ need to be evaluated. This can be done by determining coefficients a_k by inverting the matrix in Eq. (15.12). This approach is inefficient and may be subject to numerical ill-conditioning, in particular, if δ is small and N is large. An alternative and more efficient approach is to deduce δ analytically and then interpolate $P_c(\omega)$ on the r frequency points using the *barycentric* form of the *Lagrange interpolation* formula. The necessary formulation is as follows.

Parameter δ can be deduced as

$$\delta = \frac{\sum_{k=0}^{r} \alpha_k D(\hat{\omega}_k)}{\sum_{k=0}^{r} \frac{(-1)^k \alpha_k}{W(\hat{\omega}_k)}} \tag{15.16}$$

and $P_c(\omega)$ is given by

$$P_c(\omega) = \begin{cases} C_k & \text{for } \omega = \hat{\omega}_0, \hat{\omega}_1, \ldots, \hat{\omega}_{r-1} \\ \dfrac{\sum_{k=0}^{r-1} \frac{\beta_k C_k}{x - x_k}}{\sum_{k=0}^{r-1} \frac{\beta_k}{x - x_k}} & \text{otherwise} \end{cases} \tag{15.17}$$

where

$$\alpha_k = \prod_{i=0,\ i\neq k}^{r} \frac{1}{x_k - x_i} \tag{15.18}$$

$$C_k = D(\hat{\omega}_k) - (-1)^k \frac{\delta}{W(\hat{\omega}_k)} \tag{15.19}$$

$$\beta_k = \prod_{i=0,\ i\neq k}^{r-1} \frac{1}{x_k - x_i} \tag{15.20}$$

with

$$x = \cos\omega \qquad \text{and} \qquad x_i = \cos\hat{\omega}_i \qquad \text{for } i = 0, 1, 2, \ldots, r$$

In step 2 of the Remez algorithm, $|E(\omega)|$ often needs to be evaluated at a frequency that was an extremal in the previous iteration. For these cases, the magnitude of the error function is simply $|\delta|$, according to Eq. (15.11), and need not be evaluated.

15.3.4 Rejection of Superfluous Potential Extremals

The solution of Eq. (15.12) can be obtained only if *precisely* $r + 1$ extremals are available. By differentiating $E(\omega)$, one can show that in a filter with one frequency band of interest (e.g., a digital differentiator) the number of maxima in $|E(\omega)|$ (potential extremals in step 2 of Algorithm 1) is $r+1$. In the weighted-Chebyshev method, band edges at which $|E(\omega)|$ is maximum and $|E(\omega)| \geq |\delta|$ are treated as potential extremals (see Algorithm 2). Therefore, whenever the number of frequency bands is increased by one, the number of potential extremals is increased by 2, that is, for a filter with J bands there can be as many as $r + 2J - 1$ frequencies $\widehat{\omega}_i$ and a maximum of $2J - 2$ superfluous $\widehat{\omega}_i$ may occur. This problem is overcome by rejecting $\rho - r$ of the potential extremals $\widehat{\omega}_i$, if $\rho > r$, in step 4 of the algorithm.

A simple rejection scheme is to reject the $\rho - r$ frequencies $\widehat{\omega}_i$ that yield the lowest $|E(\widehat{\omega}_i)|$ and then renumber the remaining $\widehat{\omega}_i$ from 0 to r [8]. This strategy is based on the well-known fact that the magnitude of the error in a given band is inversely related to the density of extremals in that band, i.e., a low density of extremals results in a large error and a high density results in a small error. Conversely, a low band error is indicative of a high density of extremals, and rejecting superfluous $\widehat{\omega}_i$ in such a band is the appropriate course of action.

A problem with the above scheme is that whenever a frequency remains an extremal in two successive iterations, $|E(\omega)|$ assumes the value of $|\delta|$ in the second iteration by virtue of Eq. (15.11). In practice, there are almost always several frequencies that remain extremals from one iteration to the next, and the value of $|E(\omega)|$ at these frequencies will be the same. Consequently, the rejection of potential extremals on the basis of the magnitude of the error can become arbitrary and may lead to the rejection of potential extremals in bands where the density of extremals is low. This tends to increase the number of iterations, and it may even prevent the algorithm from converging on occasion. This problem can to some extent be alleviated by rejecting only potential extremals that are not band edges.

An alternative rejection scheme based on the above strategy, which was found to give excellent results for 2-band and 3-band filters, involves ranking the frequency bands in the order of lowest average band error, dropping the band with the highest average error from the list, and then rejecting potential extremals, one per band, in a cyclic manner starting with the band with the lowest average error [11]. The steps involved are as follows:

Algorithm 3: Alternative rejection scheme for superfluous potential extremals

1. Compute the average band errors

$$E_j = \frac{1}{\nu_j} \sum_{\widehat{\omega}_i \in \Omega_j} |E(\widehat{\omega}_i)| \qquad \text{for } j = 1, 2, \ldots, J$$

where $\mathbf{\Omega}_j$ is the set of potential extremals in band j given by

$$\Omega_j = \{\widehat{\omega}_i : \omega_{Lj} \leq \widehat{\omega}_i \leq \omega_{Rj}\}$$

ν_j is the number of potential extremals in band j, and J is the number of bands.

2. Rank the J bands in the order of lowest average error and let $l_1, l_2, \ldots, l_J$ be the ranked list obtained, i.e., l_1 and l_J are the bands with the lowest and highest average error, respectively.

3. Reject one $\widehat{\omega}_i$ in each of bands $l_1, l_2, \ldots, l_{J-1}, l_1, l_2, \ldots$ until $\rho - r$ superfluous $\widehat{\omega}_i$ are rejected. In each case, reject the $\widehat{\omega}_i$, other than a band edge, that yields the lowest $|E(\widehat{\omega}_i)|$ in the band. ●

For example, if $J = 3$, $\rho - r = 3$, and the average errors for bands 1, 2, and 3 are 0.05, 0.08, and 0.02, then $\widehat{\omega}_i$ are rejected in bands 3, 1, and 3. Note that potential extremals are not rejected in band 2, which is the band of highest average error.

15.3.5 Computation of Impulse Response

The impulse response in step 6 of Algorithm 1 can be determined by noting that function $P_c(\omega)$ is the frequency response of a noncausal version of the required filter. The impulse response of this filter, represented by $h_0(n)$ for $-c \le n \le c$, can be determined by computing $P_c(k\Omega)$ for $k = 0, 1, 2, \ldots, c$, where $\Omega = 2\pi/N$, and then using the *inverse discrete Fourier transform*. It can be shown that

$$h_0(n) = h_0(-n) = \frac{1}{N}\left[P_c(0) + \sum_{k=1}^{c} 2P_c(k\Omega)\cos\left(\frac{2\pi kn}{N}\right)\right] \tag{15.21}$$

for $n = 0, 1, 2, \ldots, c$ (see Prob. 15.1). Therefore, the impulse response of the required causal filter is given by

$$h(n) = h_0(n - c)$$

for $n = 0, 1, 2, \ldots, N - 1$.

15.4 IMPROVED SEARCH METHODS

For a filter of length N, with the number of intervals w_j in each interval W_j equal to S, the exhaustive step-by-step search of Sec. 15.3.2 (Algorithm 2) requires about $S \times (N + 1)/2$ function evaluations, where each function evaluation entails $N - 1$ additions, $(N + 1)/2$ multiplications, and $(N + 1)/2$ divisions (see Eq. (15.17)). A Remez optimization usually requires four to eight iterations for lowpass or highpass filters, 6 to 10 iterations for bandpass filters, and 8 to 12 iterations for bandstop filters. Further, if prescribed specifications are to be achieved and the appropriate value of N is unknown, typically two to four Remez optimizations have to be performed (see Sec. 15.7). For example, if $N = 101$, $S = 16$, number of Remez optimizations $= 4$, iterations per optimization $= 6$, the design would entail 24 iterations, 19,200 function evaluations, 1.92×10^6 additions, 0.979×10^6 multiplications, and 0.979×10^6 divisions. This is in addition to the computation required for the evaluation of δ and coefficients α_k, C_k, and β_k once per iteration. In effect, the amount of computation required to complete a design is quite substantial. In this section, alternative search techniques which reduce the amount of computation to a fraction of that required by the exhaustive search described in the previous section, are described.

15.4.1 Selective Step-by-Step Search

When Eq. (15.12) is solved, the error function $|E(\omega)|$ is forced to satisfy the alternation theorem of Sec. 15.2.3. This theorem can be satisfied in several ways. The most likely possibility is illustrated in Fig. 15.3a, where ω_{Lj} and ω_{Rj} are the left-hand and right-hand edges, respectively, of the jth frequency band. In this case, ω_{Lj} and ω_{Rj} are extremal frequencies and there is strict alternation between maxima and zeros of $|E(\omega)|$. Additional maxima of $|E(\omega)|$ can be introduced under the following circumstances:

1. To the right of $\omega = 0$ (first band), if there is an extremal and $|E(\omega)|$ has a minimum at $\omega = 0$, as depicted in Fig. 15.3*b* (see properties of $|P_c(\omega)|$ in Sec. 15.6);
2. To the left of $\omega = \pi$ (last band), if there is an extremal and $|E(\omega)|$ has a minimum at $\omega = \pi$, as depicted in Fig. 15.3*c* (see Sec. 15.6);
3. At $\omega = 0$, if there is no extremal at $\omega = 0$, as depicted in Fig. 15.3*d*;
4. At $\omega = \pi$, if there is no extremal at $\omega = \pi$, as depicted in Fig. 15.3*e*;
5. To the right of an interior left-hand edge, as depicted in Fig. 15.3*f*;
6. To the left of an interior right-hand edge, as depicted in Fig. 15.3*g*;
7. At $\omega = \omega_{Lj}$, if there is no extremal at $\omega = \omega_{Lj}$, as depicted in Fig. 15.3*h*;
8. At $\omega = \omega_{Rj}$, if there is no extremal at $\omega = \omega_{Rj}$, as depicted in Fig. 15.3*i*;
9. Two consecutive *new* maxima at the interior of a band between two adjacent extremals, as depicted in Fig. 15.3*j*.

The maxima in Fig. 15.3*a* can be located by searching in the neighborhood of each extremal frequency using gradient information since there is a one-to-one correspondence between extremals and maxima of $|E(\omega)|$. If the first derivative is positive (negative), there is a maximum of $|E(\omega)|$ to the right (left) of the extremal, which can be readily located by increasing (decreasing) the frequency in steps w_j until $|E(\omega)|$ begins to decrease. The maxima in items (1) and (2) in the above list can be found by searching to the right of $\omega = 0$ in the first case or to the left of $\omega = \pi$ in the second case, if the second derivative is positive at $\omega = 0$ or π. Similarly, the maxima in (3) and (4) can be identified by checking whether $|E(\omega)|$ has a maximum and $|E(\omega)| \geq |\delta|$ at $\omega = 0$ in the first case or at $\omega = \pi$ in the second case. The maxima in (5) and (6) can be found by searching to the right of an interior left-hand edge if the first derivative is positive or to the left of a right-hand interior edge if the first derivative is negative. Similarly, the maxima in (7) and (8) can be identified by checking whether the first derivative is negative at $\omega = \omega_{Lj}$ in the first case and positive at $\omega = \omega_{Rj}$ in the second case, and $|E(\omega)| \geq |\delta|$ in each of the two cases.

If a *selective* step-by-step search based on the above principles is used in Algorithm 1, then at the start of the optimization the distance between a typical extremal $\hat{\omega}_i$ and the nearby maximum point $\widehat{\omega}_i$ will be less than half the period of the corresponding ripple of $|E(\omega)|$, owing to the relative symmetry of the ripples of the error function. In effect, in the first iteration only half of the combined width of the different bands needs to be searched. This will reduce the number of function evaluations by more than 50 percent relative to that required by the exhaustive search of Sec. 15.3.2 without degrading the accuracy of the optimization in any way. As the optimization progresses and the solution is approached, extremal $\hat{\omega}_i$ and maximum point $\widehat{\omega}_i$ tend to coincide and, therefore, the cumulative length of the frequency range that has to be searched is progressively reduced, thereby resulting in further economies in the number of function evaluations. In the last iteration, only two or three function evaluations are needed (including derivatives) per ripple. As a result, the total number of function evaluations can be reduced by 65 to 70 percent relative to that required by the exhaustive search [10].

A selective search of the type just described will miss maxima of the type in item (9) of the above list and the algorithm will fail. However, the problem can be overcome relatively easily. Maxima of the type in (9) can sometimes occur in the stopbands of bandstop filters, and it was found

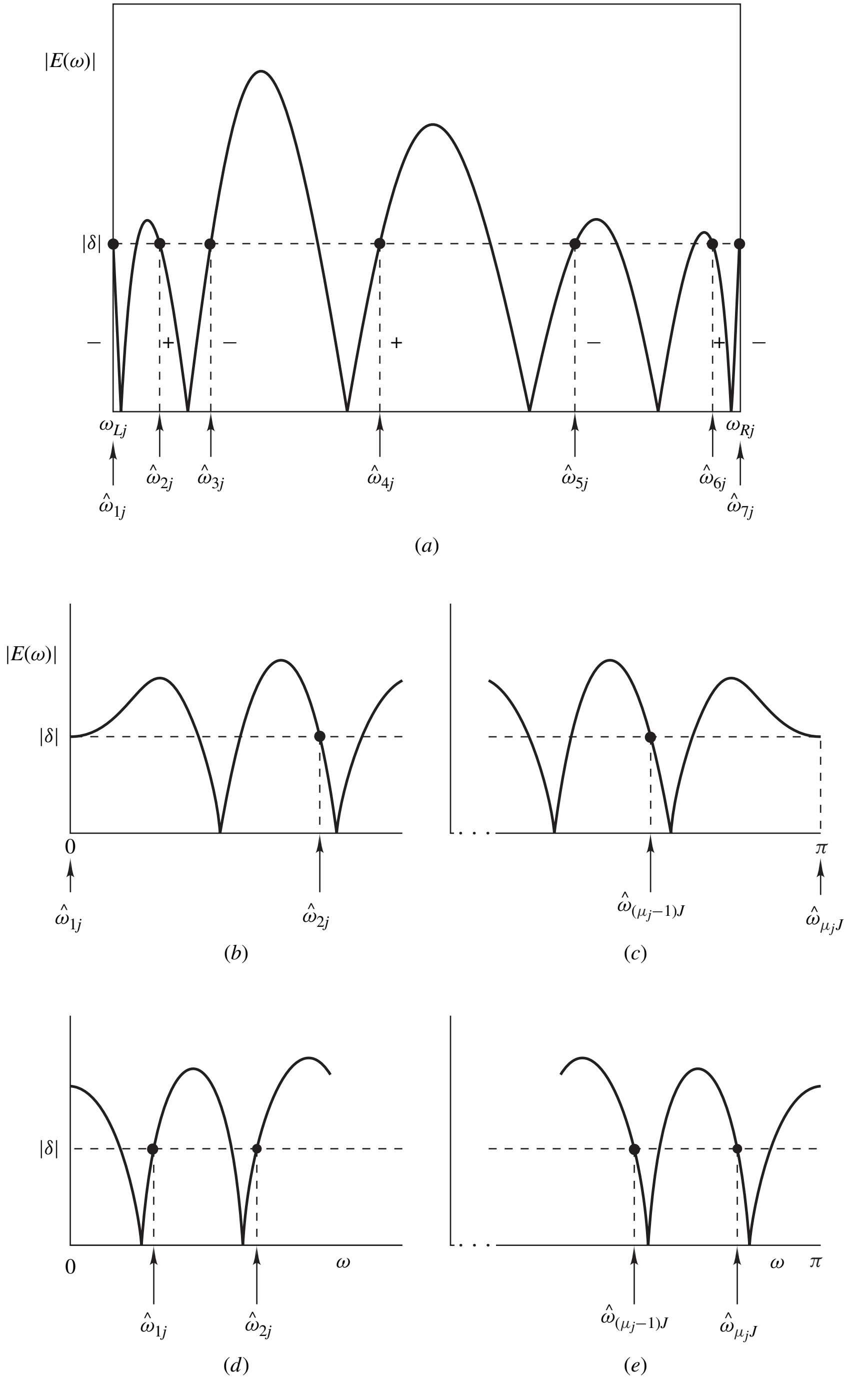


Figure 15.3 Types of maxima in $|E(\omega)|$.

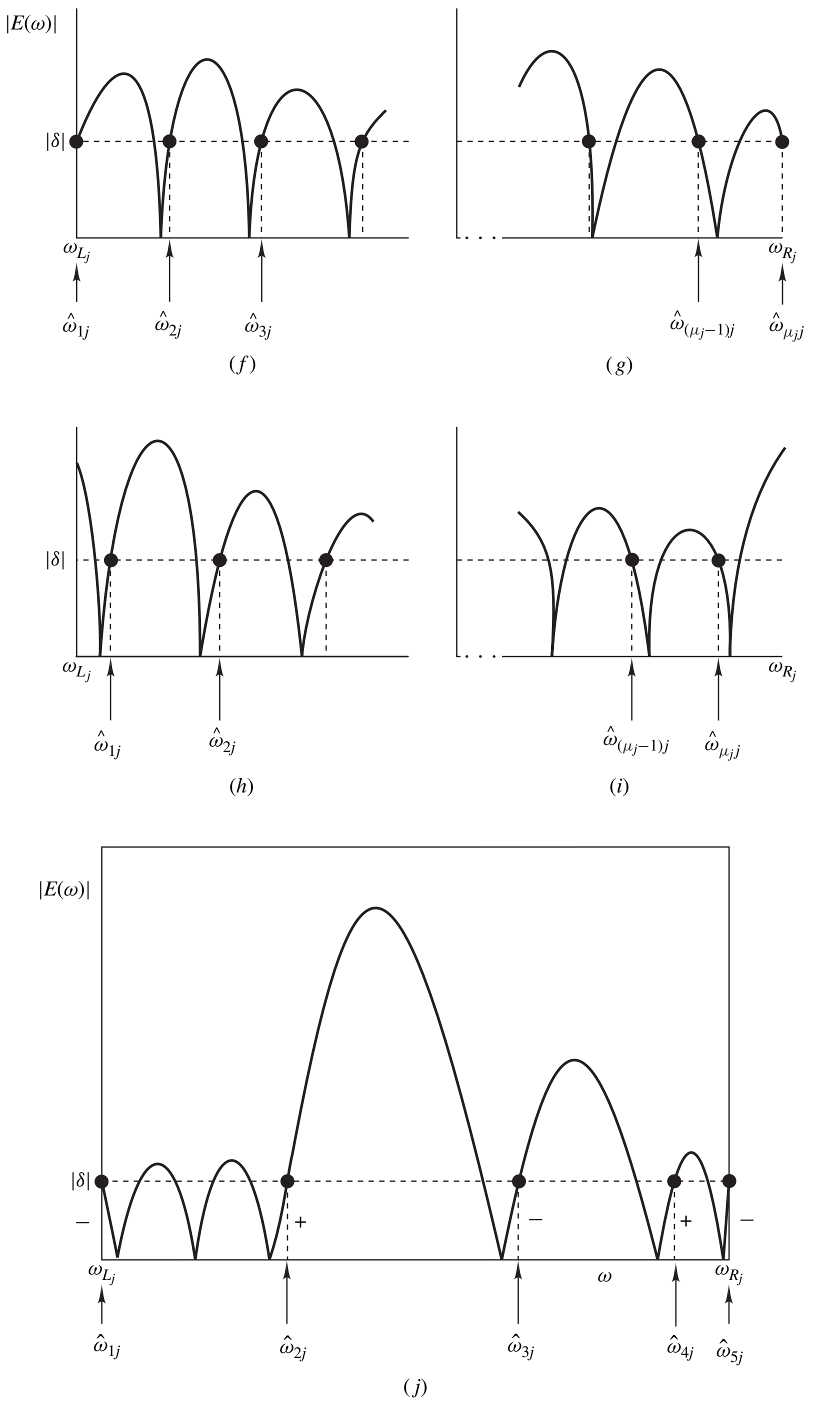


Figure 15.3 Cont'd Types of maxima in $|E(\omega)|$.

possible to reduce the number of failures by increasing somewhat the density of extremals in the stopband relative to the density of extremals in the passbands [11]. An alternative approach, which was found to give good results, is to check the distance between adjacent potential extremals at the end of the search; if the difference exceeds the initial difference by a significant amount (say if $(\widehat{\omega}_{(k+1)} - \widehat{\omega}_k) > RW_j$ for some k, where R is a constant in the range 1.5 to 2.0), then an exhaustive search is undertaken between $\widehat{\omega}_k$ and $\widehat{\omega}_{(k+1)}$ to locate any missed maxima.

15.4.2 Cubic Interpolation

This section deals with yet another search method that can further increase the computational efficiency of the Remez algorithm. The method is based on *cubic interpolation* [11].

Assume that the error function, depicted in Fig. 15.4, can be represented by the third-order polynomial

$$|E(\omega)| = M = a + b\omega + c\omega^2 + d\omega^3 \tag{15.22}$$

where $a, b, c,$ and d are constants. The first derivative of M with respect to ω is obtained from Eq. (15.22) as

$$\frac{dM}{d\omega} = G = b + 2c\omega + 3d\omega^2$$

Hence, the frequencies at which M has stationary points are given by

$$\bar{\omega} = \frac{1}{3d}\left[-c \pm \sqrt{(c^2 - 3bd)}\right] \tag{15.23}$$

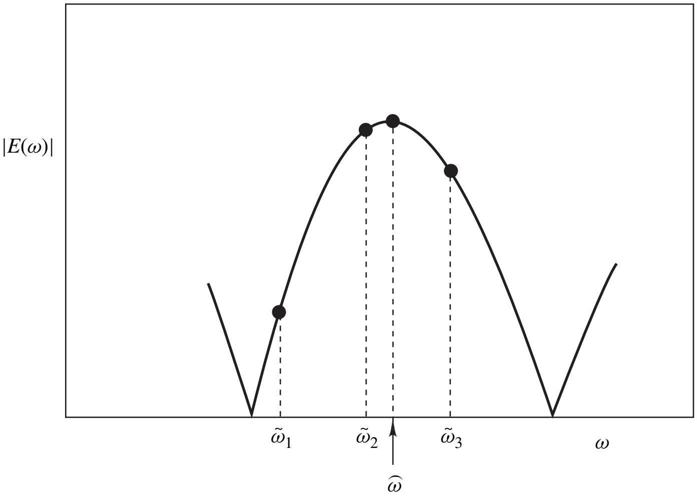


Figure 15.4 Frequency points for cubic interpolation.

Assuming that $d \neq 0$, the stationary point that corresponds to a maximum point, designated as $\widehat{\omega}$, can be selected by noting that M is maximum when

$$\frac{d^2M}{d\omega^2} = 2c + 6d\widehat{\omega} < 0$$

or

$$\widehat{\omega} < -\frac{c}{3d} \tag{15.24}$$

Evidently, if constants $b, c,$ and d are known, $\widehat{\omega}$ can be readily determined. If we assume that $\tilde{\omega}_1$, $\tilde{\omega}_2$, and $\tilde{\omega}_3$ are distinct frequencies, we can write

$$M\Big|_{\omega=\tilde{\omega}_k} = M_k = a + b\tilde{\omega}_k + c(\tilde{\omega}_k)^2 + d(\tilde{\omega}_k)^3$$

for $k = 1, 2,$ and 3 and

$$G\Big|_{\omega=\tilde{\omega}_1} = G_1 = b + 2c\tilde{\omega}_1 + 3d(\tilde{\omega}_1)^2$$

By solving this system of simultaneous equations, we can show that

$$d = \frac{\beta - \gamma}{\theta - \psi} \tag{15.25}$$

$$c = \beta - \theta d \tag{15.26}$$

$$b = G_1 - 2c\tilde{\omega}_1 - 3d(\tilde{\omega}_1)^2 \tag{15.27}$$

where

$$\beta = \frac{(M_2 - M_1) + G_1(\tilde{\omega}_1 - \tilde{\omega}_2)}{(\tilde{\omega}_1 - \tilde{\omega}_2)^2} \tag{15.28}$$

$$\gamma = \frac{(M_3 - M_1) + G_1(\tilde{\omega}_1 - \tilde{\omega}_3)}{(\tilde{\omega}_1 - \tilde{\omega}_3)^2} \tag{15.29}$$

$$\theta = \frac{2(\tilde{\omega}_1)^2 - \tilde{\omega}_2(\tilde{\omega}_1 + \tilde{\omega}_2)}{(\tilde{\omega}_1 - \tilde{\omega}_2)} \tag{15.30}$$

$$\psi = \frac{2(\tilde{\omega}_1)^2 - \tilde{\omega}_3(\tilde{\omega}_1 + \tilde{\omega}_3)}{(\tilde{\omega}_1 - \tilde{\omega}_3)} \tag{15.31}$$

By evaluating constants $\beta, \gamma, \theta, \psi, d, c,$ and b and then using Eqs. (15.23) and (15.24), $\widehat{\omega}$ can be determined.

The search just described entails four function evaluations, three for M_1, M_2, M_3, and one for G_1, per external. The method is useful as a possible replacement of the selective search only if it gives a fairly accurate estimate of $\widehat{\omega}$. To achieve this goal, the cubic interpolation should not be used near band edges where $|E(\omega)|$ is not well behaved (see Fig. 15.3).

If the above cubic interpolation is to be used to find the maximum of $|E(\omega)|$ in the neighborhood of extremal $\hat{\omega}_i$, the most suitable value for frequency $\tilde{\omega}_1$ is the extremal itself since the value of M is known at this frequency and need not be computed. It is given by Eq. (15.11) as

$$M|_{\omega=\hat{\omega}_i} = |\delta|$$

The frequency $\tilde{\omega}_3$ should bracket the required maximum point $\widehat{\omega}_i$, that is, $|\hat{\omega}_i| < |\widehat{\omega}_i| < |\tilde{\omega}_3|$, but it should not bracket the adjacent zero of the error function. It should be recalled at this point that as the solution is approached, $\widehat{\omega}_i$ tends to move closer to $\hat{\omega}_i$. Therefore, the accuracy of the cubic interpolation can be improved by reducing the interval $\tilde{\omega}_3 - \tilde{\omega}_1$ as the solution is approached. Such an *adjustable bracket* can be formed by using the convergence parameter Q of Eq. (15.13), which is known to reduce from 1 to 0 as the solution is approached. A formula for $\tilde{\omega}_3$, which was found to give good results [11] is

$$\tilde{\omega}_3 = \begin{cases} \hat{\omega}_i + \dfrac{Q}{2}(\hat{\omega}_{i+1} - \hat{\omega}_i) & \text{for } G_1 > 0 \\ \hat{\omega}_i - \dfrac{Q}{2}(\hat{\omega}_i - \hat{\omega}_{i-1}) & \text{for } G_1 < 0 \end{cases} \tag{15.32}$$

Frequency $\tilde{\omega}_2$ can be placed at the center of the frequency range $\tilde{\omega}_1$ to $\tilde{\omega}_3$, that is,

$$\tilde{\omega}_2 = \tfrac{1}{2}(\tilde{\omega}_1 + \tilde{\omega}_3) \tag{15.33}$$

The computational efficiency of the cubic-interpolation method described remains constant from iteration to iteration since the number of function evaluations required to perform an interpolation is constant. At the start of the optimization, the cubic-interpolation search is more efficient than the selective step-by-step method. However, as the solution is approached the number of function evaluations required by the selective search is progressively reduced, as was stated earlier, and at some point the selective search becomes more efficient. A prudent strategy under these circumstances is to use the cubic-interpolation search at the start of the optimization and switch over to the selective step-by-step search when some suitable criterion is satisfied. Extensive experimental results have shown that computational advantage can be gained by using the cubic-interpolation search if $Q > 0.65$, and the selective search otherwise [11]. The use of the cubic interpolation search along with the selective step-by-step search of the preceding section can reduce the number of function evaluations by 70 to 75 percent relative to that required by the exhaustive search.

15.4.3 Quadratic Interpolation

An alternative method for the location of the maxima of $|E(\omega)|$ that was found to work well is based on a *two-stage quadratic interpolation* search. However, the computational efficiency that can be achieved with this approach was found to be somewhat inferior relative to the above one-stage cubic-interpolation search.

15.4.4 Improved Formulation

In the problem formulation considered so far, the extremals $\hat{\omega}_0, \hat{\omega}_1, \ldots, \hat{\omega}_r$ are treated as a 1-D array and are numbered sequentially from 0 to r. Through the rejection of superfluous extremals, as detailed in the previous sections, the distribution of extremals can change from iteration to iteration. In order to evaluate δ and coefficients C_k in Eqs. (15.16) and (15.19) correctly, it is necessary to monitor and track the indices of the first and last extremal of each band throughout the optimization. This tends to complicate the implementation of the Remez algorithm quite significantly. The problem can be eliminated by representing the extremals in terms of a 2-D array of the form

$$\hat{\mathbf{\Omega}} = \begin{bmatrix} \hat{\omega}_{11} & \hat{\omega}_{12} & \cdots & \hat{\omega}_{1j} & \cdots & \hat{\omega}_{1J} \\ \hat{\omega}_{21} & \hat{\omega}_{22} & \cdots & \hat{\omega}_{2j} & \cdots & \hat{\omega}_{2J} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \hat{\omega}_{\mu_1 1} & \hat{\omega}_{\mu_2 2} & \cdots & \hat{\omega}_{\mu_j j} & \cdots & \hat{\omega}_{\mu_J J} \end{bmatrix}$$

where the jth column represents the extremals of the jth band, μ_j is the number of extremals in the jth band, and J is the number of bands. The use of this notation necessitates that the formulas for δ and $P_c(\omega)$ be modified accordingly. From Eqs. (15.16)–(15.20) one can show that (see Probs. 15.2 and 15.3)

$$\delta = \frac{\sum_{\{k,\,m\}\in\mathbf{K}_r} \alpha_{km} D(\hat{\omega}_{km})}{\sum_{\{k,\,m\}\in\mathbf{K}_r} \frac{(-1)^q \alpha_{km}}{W(\hat{\omega}_{km})}} \tag{15.34}$$

and

$$P_c(\omega) = \begin{cases} C_{km} & \text{for } \omega \in \hat{\mathbf{\Omega}} \\ \dfrac{\sum_{\{k,\,m\}\in\mathbf{K}_{r-1}} \frac{\beta_{km} C_{km}}{x - x_{km}}}{\sum_{\{k,\,m\}\in\mathbf{K}_{r-1}} \frac{\beta_{km}}{x - x_{km}}} & \text{otherwise} \end{cases} \tag{15.35}$$

where

$$\beta_{km} = \prod_{\{i,\,j\}\in\mathbf{I}_{r-1}} \frac{1}{x_{km} - x_{ij}} \tag{15.36}$$

$$\alpha_{km} = \begin{cases} \beta_{km} & \text{if } k = \mu_J \text{ and } m = J \\ \dfrac{\beta_{km}}{x_{km} - x_{\mu_J J}} & \text{otherwise} \end{cases} \tag{15.37}$$

$$C_{km} = D(\hat{\omega}_{km}) - (-1)^q \frac{\delta}{W(\hat{\omega}_{km})} \tag{15.38}$$

with

$$q = \begin{cases} k - 1 & \text{if } m = 1 \\ k - 1 + \sum_{j=1}^{m-1} \mu_j & \text{if } m \geq 2 \end{cases} \tag{15.39}$$

and

$$x = \cos\omega \qquad x_{ij} = \cos\hat{\omega}_{ij} \qquad \text{for } \{i,\,j\} \in \mathbf{I}_r$$

In the above formulation, $\mathbf{K}_r$, $\mathbf{K}_{r-1}$, $\mathbf{I}_r$, and $\mathbf{I}_{r-1}$ are sets given by

$$\mathbf{K}_r = \{\{k, m\} : (1 \le k \le \mu_m) \text{ and } (1 \le m \le J)\} \tag{15.40}$$

$$\mathbf{K}_{r-1} = \{\{k, m\} : (1 \le k \le l) \text{ and } (1 \le m \le J)\} \tag{15.41}$$

$$\mathbf{I}_r = \{\{i, j\} : (1 \le i \le \mu_j) \text{ and } (1 \le j \le J)\} \tag{15.42}$$

and

$$\mathbf{I}_{r-1} = \{\{i, j\} : (1 \le i \le h) \text{ and } (1 \le j \le J) \text{ and } (i \ne k \text{ or } j \ne m)\} \tag{15.43}$$

with

$$l = \begin{cases} \mu_J - 1 & \text{for } m = J \\ \mu_m & \text{otherwise} \end{cases}$$

and

$$h = \begin{cases} \mu_J - 1 & \text{for } j = J \\ \mu_j & \text{otherwise} \end{cases}$$

15.5 EFFICIENT REMEZ EXCHANGE ALGORITHM

The above principles will now be used to construct an efficient Remez exchange algorithm. As in Algorithm 2, ω_{Lj} and ω_{Rj} are the left- and right-hand edges in band j; W_j is the interval between adjacent extremals and m_j is the number of intervals W_j in band j; w_j is the interval between successive samples in interval W_j, and S is the number of intervals w_j in each interval W_j; N_j is the total number of intervals w_j in band j; and J is the number of bands. The frequencies $\hat{\omega}_{1j}, \hat{\omega}_{2j}, \ldots, \hat{\omega}_{\mu_j j}$ are the current extremals and $\widehat{\omega}_{1j}, \widehat{\omega}_{2j}, \ldots, \widehat{\omega}_{\nu_j j}$ are the potential extremals for the next iteration in band j. The magnitude of the error function and its first and second derivatives with respect to ω are denoted as

$$M = |E(\omega)| \qquad G_1 = \frac{d|E(\omega)|}{d\omega} \qquad G_2 = \frac{d^2|E(\omega)|}{d\omega^2}$$

The improved algorithm consists of a main part called MAIN which calls routines EXTREMALS, SELECTIVE, and CUBIC. The steps involved are detailed below.

Algorithm 4: Efficient Remez exchange algorithm

MAIN

```
M1.
    (a) Initialize $S$, say, $S = 16$, and set $Q = 1$.
    (b) For $j = 1, 2, \ldots, J$ do:
        Compute $m_j$ and $W_j$ for $j = 1, 2, \ldots, J$ using Eqs. (15.14) and (15.15), respectively.
        Initialize extremals by letting $\hat{\omega}_{1j} = \omega_{Lj}, \ldots, \hat{\omega}_{ij} = \omega_{Lj} + (i-1)W_j, \ldots, \hat{\omega}_{\mu_j j} = \omega_{Rj} = \omega_{Lj} + m_j W_j$.
        Set $N_j = m_j S$ and $w_j = B_j / N_j$.
```

```
M2.
    (a) Compute coefficients $\beta_{kj}$, $\alpha_{kj}$, and $C_{kj}$ for $j = 1, 2, \ldots, J$ using Eqs. (15.36)–(15.38).
    (b) Compute $\delta$ using Eq. (15.34).
M3. Call EXTREMALS.
M4.
    (a) Set $\rho = \nu_1 + \nu_2 + \ldots \nu_J$.
    (b) Reject $\rho-(r+1)$ superfluous potential extremals[2] using Algorithm 3, renumber the remaining
        $\widehat{\omega}_{ij}$ sequentially, and update $\mu_j$ if necessary.
    (c) Update extremals by letting $\hat{\omega}_{ij} = \widehat{\omega}_{ij}$ for $i = 1, 2, \ldots, \mu_j$ and $j = 1, 2, \ldots, J$.
M5.
    (a) Compute $Q$ using Eq. (15.13).
    (b) If $Q > 0.01$, go to step M2.
M6.
    (a) Compute $P_c(k\Omega)$ for $k = 0, 1, \ldots, r-1$ using Eq. (15.35).
    (b) Compute $h(n)$ using Eq. (15.21).
    (c) Stop.
```

EXTREMALS

```
E1. For each of bands $1, 2, \ldots, j, \ldots, J$ do:
    (A) Set $e = 0$.
    (B) For each of extremals $\hat{\omega}_{1j}, \hat{\omega}_{2j}, \ldots, \hat{\omega}_{ij}, \ldots, \hat{\omega}_{\mu_j j}$ do:
        (a) Case $\hat{\omega}_{ij} = \hat{\omega}_{1j}$:
                If $\hat{\omega}_{ij} = \omega_{Lj}$, then do:
                    Case $j = 1$ (first band):
                    If $G_2 < 0$, then set $e = e + 1$ and $\widehat{\omega}_{ej} = \hat{\omega}_{ij}$; otherwise call SELECTIVE.
                    Case $j \neq 1$ (other bands):
                    If $G_1 > 0$, then call SELECTIVE; otherwise set $e = e + 1$ and $\widehat{\omega}_{ej} = \hat{\omega}_{ij}$.
                If $\hat{\omega}_{ij} \neq \omega_{Lj}$, then call SELECTIVE.
        (b) Case $\hat{\omega}_{1j} < \hat{\omega}_{ij} < \hat{\omega}_{\mu_j j}$:
                If $Q < 0.65$, then call SELECTIVE; otherwise call CUBIC.
                If $flag\,0 = 1$ (CUBIC was unsuccessful in generating a good estimate of the maximum
                point), then call SELECTIVE.
        (c) Case $\hat{\omega}_{ij} = \hat{\omega}_{\mu_j j}$:
                If $\hat{\omega}_{ij} = \omega_{Rj}$, then do:
                    Case $j = J$ (last band):
                    If $G_2 < 0$, then set $e = e + 1$ and $\widehat{\omega}_{ej} = \hat{\omega}_{ij}$; otherwise call SELECTIVE.
                    Case $j \neq J$ (other bands):
                    If $G_1 < 0$, then call SELECTIVE; otherwise set $e = e + 1$ and $\widehat{\omega}_{ej} = \hat{\omega}_{ij}$.
                If $\hat{\omega}_{ij} \neq \omega_{Rj}$, then call SELECTIVE.
    (C) Check for an additional potential extremal at the left-hand edge of band $j$: If $\hat{\omega}_{1j}$ and $\widehat{\omega}_{1j} \neq$
        $\omega_{Lj}$, $|E(\omega_{Lj})| > |E(\omega_{Lj} + w_j)|$, and $|E(\omega_{Lj})| \geq |\delta|$, then set $e = e + 1$ and insert new
        potential extremal at $\omega = \omega_{Lj}$.
```

[2] *The difference between the number of superfluous extremals in step 4 of Algorithm 1 and step M4(b) of Algorithm 4 is due to the fact that the count of potential extremals starts with 0 in Algorithm 1 and with 1 in Algorithm 4.*

```
        (D) Check for an additional potential extremal at the right-hand edge of band $j$: If $\hat{\omega}_{\mu_j j}$ and
            $\widehat{\omega}_{ej} \neq \omega_{Rj}$, $|E(\omega_{Rj} - w_j)| < |E(\omega_{Rj})|$, and $|E(\omega_{Rj})| \geq |\delta|$, then insert new potential
            extremal at $\omega = \omega_{Rj}$ and set $e = e + 1$.
        (E) Check for additional potential extremals in band $j$:
            (a) For $k = 1, 2, \ldots, e - 1$ check if

                    $$\widehat{\omega}_{(k+1)j} - \widehat{\omega}_{kj} > RW_j$$

                For each value of $k$ for which the inequality is satisfied, use an exhaustive search between
                frequencies $\widehat{\omega}_{kj}$ and $\widehat{\omega}_{(k+1)j}$ (see Algorithm 2). For each new maximum of $M$ such that
                $|E(\omega)| \geq |\delta|$, insert a new potential extremal sequentially between $\widehat{\omega}_{kj}$ and $\widehat{\omega}_{(k+1)j}$ and
                set $e = e + 1$ ($R$ is a constant in the range 1.5 to 2.0).
            (b) If there is a large gap (larger than $RW_j$) between the left-hand edge and the first potential
                extremal, check for additional potential extremals in the range $\omega_{Lj} < \omega < \widehat{\omega}_{1j}$; for
                each new maximum such that $|E(\omega)| \geq |\delta|$, insert a new potential extremal sequentially
                between $\omega_{Lj}$ and $\widehat{\omega}_{1j}$ and set $e = e + 1$.
            (c) If there is a large gap (larger than $RW_j$) between the last potential extremal and the right-
                hand edge, check for additional potential extremals in the range $\widehat{\omega}_{ej} < \omega < \omega_{Rj}$; for
                each new maximum such that $|E(\omega)| \geq |\delta|$, insert a new potential extremal sequentially
                between $\widehat{\omega}_{ej}$ and $\omega_{Rj}$ and set $e = e + 1$.
        (F) Set $\nu_j = e$.
E2.     Return.

SELECTIVE

    S1. If ($G_1 > 0$ and $\hat{\omega}_{ij} \neq 0$) or ($G_2 > 0$ and $\hat{\omega}_{ij} = 0$), then increase $\omega$ in steps $w_j$ until a maximum
        of $M$ is located. Set $e = e + 1$ and assign the frequency of this maximum to $\widehat{\omega}_e$. If no maximum is located
        in the frequency range $\hat{\omega}_{ij} \leq \omega <$ ($\hat{\omega}_{(i+1)j}$ or $\omega_{Rj}$), discontinue the search.
    S2. If ($G_1 < 0$ and $\hat{\omega}_{ij} \neq \pi$) or ($G_2 > 0$ and $\hat{\omega}_{ij} = \pi$), then decrease $\omega$ in steps $w_j$ until
        a maximum of $M$ is located. Set $e = e + 1$ and assign the frequency of this maximum to $\widehat{\omega}_e$. If no
        maximum is located in the frequency range ($\omega_{Lj}$ or $\hat{\omega}_{(i-1)j}$) $\leq \omega < \hat{\omega}_{ij}$, discontinue the search.
    S3. Return.

CUBIC

    C1. Set $flag0 = 0$.
    C2. Set $\tilde{\omega}_1 = \hat{\omega}_{ij}$ and compute frequencies $\tilde{\omega}_3$ and $\tilde{\omega}_2$ using Eqs. (15.32) and (15.33).
    C3. Compute constants $\beta, \gamma, \theta$, and $\psi$ using Eqs. (15.28)–(15.31).
    C4. Compute constants $d$, $c$, and $b$ using Eqs. (15.25)–(15.27). If $3bd > c^2$ (third-order polynomial
        has no maximum), then set $flag0 = 1$ and return.
    C5. Compute $\widehat{\omega}$ using Eqs. (15.23) and (15.24). If frequency $\widehat{\omega}$ is outside the interval $[\tilde{\omega}_1, \tilde{\omega}_3]$
        (estimate of the maximum point is unreliable), then set $flag0 = 1$ and return.
    C6. Set $\widehat{\omega} = w_j \times \text{Int}\,(\widehat{\omega}/w_j + 0.5)$.
    C7. Set $e = e + 1$ and $\widehat{\omega}_{ej} = \widehat{\omega}$.
    C8. Return. ●
```

Step E1(B)(a) checks for maxima at or near the left-hand edge of each band for the cases illustrated in Fig. 15.3a, b, d, f, and h. Step E1(B)(b) locates the interior maxima in Fig. 15.3a that correspond to extremals $\hat{\omega}_{2j}$ to $\hat{\omega}_{(\mu_j-1)j}$. Step E1(*B*)(*c*) checks for maxima at or near the right-hand edge of each band for the cases illustrated in Fig. 15.3*a*, *c, e, g*, and *i*. Step E1(*C*) checks for a new maximum at left-hand edge ω_{Lj} in the special case where there is no extremal and a maximum has not been picked up already at this frequency by step E1(*B*)(*a*). Such a situation can arise as shown in Fig. 15.3*d* where step E1(*B*)(*a*) will pick up the maximum at the right of point $\omega = \hat{\omega}_{1j}$, since $G_1 > 0$, but miss the maximum at $\omega = 0$. A similar situation can arise as illustrated in Fig. 15.3*h*. Step E1(*D*) checks for a new maximum at right-hand edge ω_{Rj} for the case where there is no extremal and a maximum has not been picked up already at this frequency by step E1(*B*)(*c*). Such a situation can arise as shown in Fig. 15.3*e* where step E1(*B*)(*c*) will pick up the maximum at the left of point $\omega{=}\hat{\omega}_{\mu_J J}$, since $G_1{<}0$, but miss the maximum at $\omega{=}\pi$. A similar situation can arise as illustrated in Fig. 15.3*i*. Steps E1(*E*)(*a*) to E1(*E*)(*c*) check for any missed maxima, like the maxima between $\hat{\omega}_{1j}$ and $\hat{\omega}_{2j}$ in Fig. 15.3*j*, in cases where the interval between any two adjacent maxima, between the left-hand edge and the first maximum, or between the last maximum and the right-hand edge is significantly larger than the average interval between adjacent extremals.

When the ripple of the error function is seriously skewed (e.g., near band edges in the first or second iteration) routine CUBIC may yield a poor estimate of the maximum point, and on rare occasions the third-order polynomial may not have a maximum. If either of these cases is detected, CUBIC is aborted and SELECTIVE is called in its place. CUBIC will almost always yield a value of $\widehat{\omega}$ between two adjacent sample points. In order to ensure that each potential extremal is a member of the set of sample points, $\widehat{\omega}$ is rounded to the nearest sample point in step C6. This makes the estimate produced by CUBIC compatible with that produced by SELECTIVE and prevents numerical ill-conditioning in the evaluation of $E(\omega)$, G_1, and G_2. The CUBIC interpolation routine may be disabled by modifying step E1(*B*)(*b*).

Extensive experimentation by the author has shown the above algorithm to be quite robust. It never failed in the design of 81 2-band filters chosen at random, it failed twice in the design of 67 3-band filters, three times in the design of 50 4-band filters, and three times in the design of 33 5-band filters. Lack of convergence is usually brought about by a cyclic pattern of rejected potential extremals, but the problem can be easily overcome by changing one of the specified filter parameters slightly, e.g., a passband or stopband edge or the order of the filter.

15.6 GRADIENT INFORMATION

Routines SELECTIVE and CUBIC in the above algorithm rely heavily on the *first and second derivatives* of $|E(\omega)|$ with respect to ω. From Eq. (15.3), we have

$$\frac{d|E(\omega)|}{d\omega} = \operatorname{sgn}\,[D(\omega) - P_c(\omega)]\left[\frac{dD(\omega)}{d\omega} - \frac{dP_c(\omega)}{d\omega}\right] \tag{15.44}$$

and

$$\frac{d^2|E(\omega)|}{d\omega^2} = \operatorname{sgn}\,[D(\omega) - P_c(\omega)]\left[\frac{d^2D(\omega)}{d\omega^2} - \frac{d^2P_c(\omega)}{d\omega^2}\right] \tag{15.45}$$

where

$$\operatorname{sgn}\,(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ -1 & \text{for } x < 0. \end{cases}$$

The first and second derivatives of $|E(\omega)|$ under different circumstances can be obtained from the following properties of $|P_c(\omega)|$ [10].

15.6.1 Property 1

For any frequency including extremal $\hat{\omega}_{\mu_J J}$ (last extremal of last band) but excluding all other extremals,

$$\frac{dP_c(\omega)}{d\omega} = \frac{d(\omega)n_1(\omega) - d_1(\omega)n(\omega)}{d^2(\omega)} \tag{15.46}$$

where

$$n(\omega) = \sum_{\{k,m\}\in\mathbf{K}_{r-1}} \frac{\beta_{km}C_{km}}{x - x_{km}}$$

$$d(\omega) = \sum_{\{k,m\}\in\mathbf{K}_{r-1}} \frac{\beta_{km}}{x - x_{km}}$$

$$n_1(\omega) = \sin\omega \sum_{\{k,m\}\in\mathbf{K}_{r-1}} \frac{\beta_{km}C_{km}}{(x - x_{km})^2}$$

$$d_1(\omega) = \sin\omega \sum_{\{k,m\}\in\mathbf{K}_{r-1}} \frac{\beta_{km}}{(x - x_{km})^2}$$

and $\mathbf{K}_{r-1}$ is given by Eq. (15.41).

15.6.2 Property 2

For all extremal frequencies except $\hat{\omega}_{\mu_J J}$,

$$\left.\frac{dP_c(\omega)}{d\omega}\right|_{\omega=\hat{\omega}_{ij}} = \frac{\sin(\hat{\omega}_{ij})}{\beta_{ij}} \sum_{\{k,m\}\in\mathbf{K}_{r-1}^{ij}} \frac{\beta_{km}(C_{ij} - C_{km})}{x_{ij} - x_{km}} \tag{15.47}$$

where

$$\mathbf{K}_{r-1}^{ij} = \{\{k,m\} : (1 \le k \le l) \text{ and } (1 \le m \le J) \text{ and } (k \ne i \text{ or } m \ne j)\}$$

with

$$l = \begin{cases} \mu_J - 1 & \text{for } m = J \\ \mu_m & \text{otherwise} \end{cases}$$

15.6.3 Property 3

For $\omega = 0$ or π, it follows from properties 1 and 2 that

$$\frac{dP_c(\omega)}{d\omega} = 0 \tag{15.48}$$

15.6.4 Property 4

For $\omega = 0$ if no extremal occurs at zero or for $\omega = \pi$ under all circumstances,

$$\frac{d^2 P_c(\omega)}{d\omega^2} = \frac{d(\omega)n_2(\omega) - d_2(\omega)n(\omega)}{d^2(\omega)} \tag{15.49}$$

where

$$n_2(\omega) = \cos\omega \sum_{\{k,\,m\}\in\mathbf{K}_{r-1}} \frac{\beta_{km} C_{km}}{(x - x_{km})^2}$$

$$d_2(\omega) = \cos\omega \sum_{\{k,\,m\}\in\mathbf{K}_{r-1}} \frac{\beta_{km}}{(x - x_{km})^2}$$

15.6.5 Property 5

If there is an extremal at $\omega = 0$, then

$$\left.\frac{d^2 P_c(\omega)}{d\omega^2}\right|_{\omega=0} = \frac{1}{\beta_{11}} \sum_{\{k,\,m\}\in\mathbf{K}^{11}_{r-1}} \frac{\beta_{km}(C_{11} - C_{km})}{x - x_{km}} \tag{15.50}$$

where $\mathbf{K}^{11}_{r-1} = \mathbf{K}^{ij}_{r-1}$ with $i = j = 1$.

Example 15.1 Design a nonrecursive equiripple highpass filter using the Remez algorithm (*a*) with the exhaustive search of Sec. 15.3.2, (*b*) with the selective step-by-step search of Sec. 15.4.1, and (*c*) with the selective step-by-step search in conjunction with the cubic-interpolation search of Sec. 15.4.2. Compare the results obtained. The required specifications are as follows:

- Filter length N: 21
- Passband edge ω_p: 2.0 rad/s
- Stopband edge ω_a: 1.0 rad/s
- Ratio δ_p/δ_a: 18.0
- Sampling frequency ω_s: 2π rad/s

■ **Solution**

The design in (*a*) was carried out using Algorithm 1 in conjunction with Algorithms 2 and 3, whereas the designs in (*b*) and (*c*) were carried out using Algorithm 4, first without and then with routine CUBIC, respectively. The progress of the design is illustrated in Table 15.1. As can be seen, the exhaustive and selective search methods required four iterations each, whereas the selective search in conjunction with cubic interpolation required five iterations. However, the number of function evaluations (evaluations of $P_c(\omega)$ using

Table 15.1 Progress in design of highpass filter (Example 15.1)

Iter. No.	Exhaustive search		Selective search		Sel. with cub. interpolation	
	Q	FE's	Q	FE's	Q	FE's
1	0.9379	154	0.9379	70	0.9379	49
2	0.5792	154	0.5792	48	0.6519	28
3	0.0846	154	0.0846	29	0.0756	28
4	0.0000	154	0.0000	20	0.0309	22
5	—	—	—	—	0.0000	20
Total FE's		616		167		147

Eq. (15.35) plus evaluations of G_1 and G_2 using Eqs. (15.44) and (15.45), respectively) decreased from 616 in the first method to 167 in the second and to 147 in the third. In the Remez algorithm, approximately 80 to 90 percent of the computational effort involves function evaluations. In effect, relative to that required by the exhaustive search, the use of the selective step-by-step search reduced the amount of computation by about 73 percent, and the use of the selective step-by-step search in conjunction with the cubic-interpolation search reduced the amount of computation by about 76 percent.

The three methods resulted in approximately the same impulse responses, as can be seen in Table 15.2, and the passband ripple and minimum stopband attenuation obtained in each case were 0.073 dB and 72.6 dB, respectively. The amplitude response of the filter is illustrated in Fig. 15.5. ■

Table 15.2 Impulse response of highpass filter (Example 15.1)

n	$h_0(n) = h_0(-n)$	
	Exhaustive or selective search	Selective with cubic interpolation
0	$4.976192E-1$	$4.976160E-1$
1	$-3.120628E-1$	$-3.120636E-1$
2	$2.462999E-3$	$2.466692E-3$
3	$8.853907E-2$	$8.854032E-2$
4	$-2.605336E-3$	$-2.609410E-3$
5	$-3.790087E-2$	$-3.790082E-2$
6	$2.553469E-3$	$2.555600E-3$
7	$1.553835E-2$	$1.553849E-2$
8	$-2.126568E-3$	$-2.127674E-3$
9	$-5.222708E-3$	$-5.222820E-3$
10	$1.898114E-3$	$1.898558E-3$

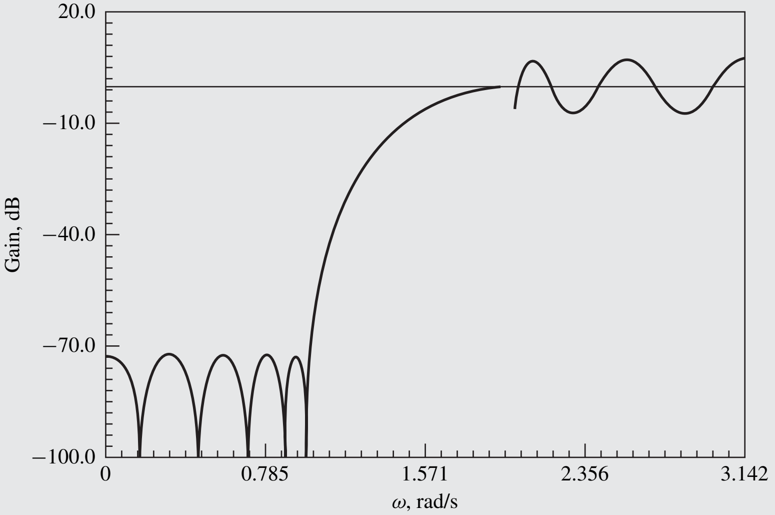


Figure 15.5 Amplitude response of equiripple highpass filter (Example 15.1) (the passband gain is multiplied by the factor 200 to show the passband ripple).

Example 15.2 Design a nonrecursive equiripple bandpass filter using the Remez algorithm (*a*) with the exhaustive search, (*b*) with the selective step-by-step search, and (*c*) with the selective step-by-step search in conjunction with the cubic-interpolation search. Compare the results obtained. The required specifications are as follows:

- Filter length N: 33
- Lower passband edge ω_{p1}: 1.00 rad/s
- Upper passband edge ω_{p2}: 2.00 rad/s
- Lower stopband edge ω_{a1}: 0.63 rad/s
- Upper stopband edge ω_{a2}: 2.40 rad/s
- Ratio δ_p/δ_a: 23.0
- Sampling frequency ω_s: 2π rad/s

■ **Solution**

As in Example 15.1, the design in (*a*) was carried out using Algorithm 1 in conjunction with Algorithms 2 and 3, whereas the designs in (*b*) and (*c*) were carried out using Algorithm 4 first without and then with routine CUBIC. The progress of the design is illustrated in Table 15.3. In this example, each of the three methods required eight iterations, and in each case there was a superfluous maximum at the end of the fourth iteration. The problem was eliminated by rejecting the second maximum of the third band, using the rejection method detailed in Sec. 15.3.4 (see Algorithm 3). As can

Table 15.3 Progress in design of bandpass filter (Example 15.2)

Iter. No.	Exhaustive search		Selective search		Selective with cubic interpolation	
	Q	FE's	Q	FE's	Q	FE's
1	0.8970	231	0.8970	99	0.8906	42
2	0.6109	231	0.6109	63	0.6098	42
3	0.4556	231	0.4556	54	0.3940	59
4	0.2975	231	0.2975	40	0.2924	41
			$\widehat{\omega}_{23}$ rejected			
5	0.6329	231	0.6329	45	0.6342	45
6	0.4035	231	0.4035	44	0.4019	44
7	0.1268	231	0.1268	40	0.1293	40
8	0.0063	231	0.0063	32	0.0078	32
Total FE's		1848		417		345

be seen in Table 15.3, the number of function evaluations decreased from 1848 in the first method to 417 in the second method to 345 in the third method. In effect, the use of the selective step-by-step search reduced the amount of computation by about 77 percent, and the use of the selective step-by-step search in conjunction with the cubic-interpolation search reduced the amount of computation by about 81 percent, relative to that required by the exhaustive search.

The three methods resulted in approximately the same impulse responses, as can be seen in Table 15.4. The amplitude response of the filter is illustrated in Fig. 15.6;

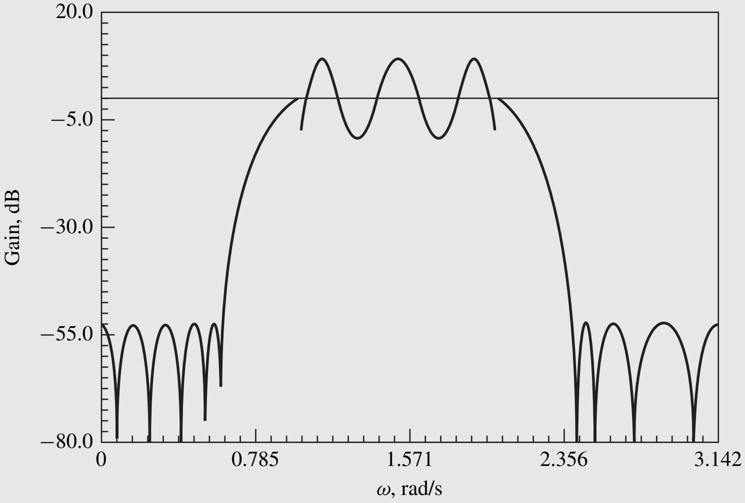


Figure 15.6 Amplitude response of equiripple bandpass filter (Example 15.2) (the passband gain is multiplied by the factor 20 to show the passband ripple).

the passband ripple and minimum stopband attenuation obtained in each case were 0.934 dB and 52.6 dB, respectively. ■

Table 15.4 Impulse response of bandpass filter (Example 15.2)

n	$h_0(n) = h_0(-n)$	
	Exhaustive or selective search	**Selective with cubic interpolation**
0	$4.095939E - 1$	$4.095922E - 1$
1	$2.529508E - 2$	$2.529615E - 2$
2	$-2.978313E - 1$	$-2.978309E - 1$
3	$-3.828648E - 2$	$-3.828785E - 2$
4	$7.734350E - 2$	$7.734476E - 2$
5	$-1.885007E - 3$	$-1.885265E - 3$
6	$5.491991E - 2$	$5.491921E - 2$
7	$3.246312E - 2$	$3.246380E - 2$
8	$-4.740273E - 2$	$-4.740238E - 2$
9	$-1.517104E - 2$	$-1.517257E - 2$
10	$-4.296619E - 3$	$-4.294720E - 3$
11	$-1.645571E - 2$	$-1.645695E - 2$
12	$2.126770E - 2$	$2.126769E - 2$
13	$2.137354E - 2$	$2.137482E - 2$
14	$-8.758516E - 3$	$-8.760408E - 3$
15	$-8.493829E - 3$	$-8.492478E - 3$
16	$-3.891420E - 5$	$-3.931393E - 5$

15.7 PRESCRIBED SPECIFICATIONS

Given a filter length N, a set of passband and stopband edges, and a ratio δ_p/δ_a, a nonrecursive filter with approximately piecewise-constant amplitude-response specifications can be readily designed. While the filter obtained will have passband and stopband edges at the correct locations and the ratio δ_p/δ_a will be as required, the amplitudes of the passband and stopband ripples are highly unlikely to be precisely as specified. An acceptable design can be obtained by predicting the value of N on the basis of the required specifications and then designing filters for increasing or decreasing values of N until the lowest value of N that satisfies the specifications is found.

A reasonably accurate *empirical* formula for the prediction of N for the case of lowpass and highpass filters, due to Herrmann, Rabiner, and Chan [15], is

$$N = \text{Int}\left[\frac{(D - FB^2)}{B} + 1.5\right] \tag{15.51}$$

where

$$\begin{aligned}
B &= |\omega_a - \omega_p|/2\pi \\
D &= [0.005309(\log \delta_p)^2 + 0.07114 \log \delta_p - 0.4761] \log \delta_a \\
&\quad -[0.00266(\log \delta_p)^2 + 0.5941 \log \delta_p + 0.4278] \\
F &= 0.51244(\log \delta_p - \log \delta_a) + 11.012
\end{aligned}$$

This formula can also be used to predict the filter length in the design of bandpass, bandstop, and multiband filters in general. In these filters, a value of N is computed for each transition band between a passband and stopband or a stopband and passband using Eq. (15.51) and the largest value of N so obtained is taken to be the predicted filter length. *Prescribed specifications* can be achieved by using the following design algorithm:

Algorithm 5: Design of filters satisfying prescribed specifications

```
1. Compute $N$ using Eq. (15.51); if $N$ is even, set $N = N + 1$.
2. Design a filter of length $N$ using Algorithm 4 and determine the minimum value of $\delta$, say $\breve{\delta}$.
   $(A)$ If $\breve{\delta} > \delta_p$, then do:
      $(a)$ Set $N = N + 2$, design a filter of length $N$ using Algorithm 4, and find $\breve{\delta}$;
      $(b)$ If $\breve{\delta} \le \delta_p$, then go to step 3; else, go to step 2$(A)(a)$.
   $(B)$ If $\breve{\delta} < \delta_p$, then do:
      $(a)$ Set $N = N - 2$, design a filter of length $N$ using Algorithm 4, and find $\breve{\delta}$;
      $(b)$ If $\breve{\delta} > \delta_p$, then go to step 4; else, go to step 2$(B)(a)$.
3. Use the last set of extremals and the corresponding value of $N$ to obtain the impulse response of the
   required filter and stop.
4. Use the last but one set of extremals and the corresponding value of $N$ to obtain the impulse response
   of the required filter and stop. ●
```

Example 15.3 In an application, a nonrecursive equiripple bandstop filter is required, which should satisfy the following specifications:

- Odd filter length
- Maximum passband ripple A_p: 0.5 dB
- Minimum stopband attenuation A_a: 50.0 dB

- Lower passband edge $\omega_{p1}:$ 0.8 rad/s
- Upper passband edge $\omega_{p2}:$ 2.2 rad/s
- Lower stopband edge $\omega_{a1}:$ 1.2 rad/s
- Upper stopband edge $\omega_{a2}:$ 1.8 rad/s
- Sampling frequency $\omega_s:$ 2π rad/s

Design the lowest-order filter that will satisfy the specifications.

■ **Solution**

The use of Algorithm 4 in conjunction with Algorithm 5 gave a filter of length 35. The progress of the design is illustrated in Table 15.5. The impulse response of the filter obtained is given in Table 15.6. The corresponding amplitude response is depicted in Fig. 15.7; the passband ripple and minimum stopband attenuation achieved are 0.4342 and 51.23 dB, respectively, and are within the specified limits. ■

Table 15.5 Progress in design of bandstop filter (Example 15.3)

N	Iters.	FE's	A_p, dB	A_a, dB
31	10	582	0.5055	49.91
33	7	376	0.5037	49.94
35	9	545	0.4342	51.23

Table 15.6 Impulse response of bandstop filter (Example 15.3)

n	$h_0(n) = h_0(-n)$	n	$h_0(n) = h_0(-n)$
0	$6.606345E-1$	9	$2.806340E-2$
1	$-2.307038E-2$	10	$-2.276572E-2$
2	$2.711461E-1$	11	$-9.924812E-3$
3	$4.306831E-2$	12	$-1.047638E-3$
4	$-1.198723E-1$	13	$-1.412229E-2$
5	$-1.829974E-2$	14	$1.284774E-2$
6	$-4.974998E-3$	15	$1.096745E-2$
7	$-2.016415E-2$	16	$8.260758E-4$
8	$4.593774E-2$	17	$3.482212E-3$

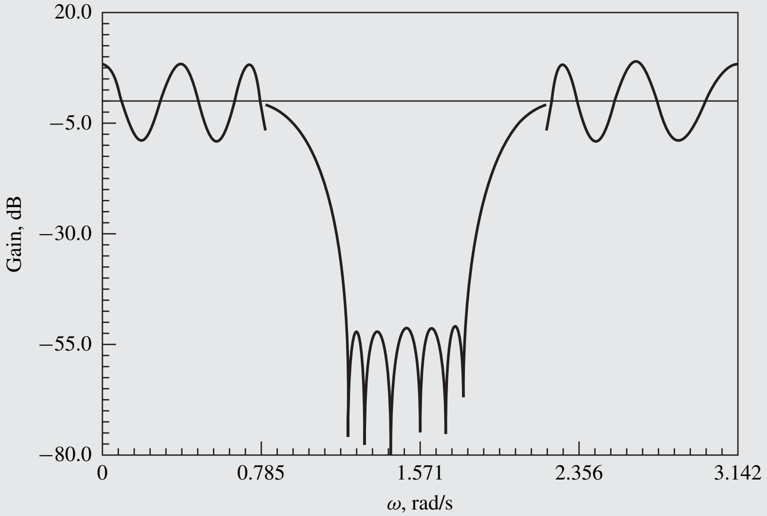


Figure 15.7 Amplitude response of equiripple bandstop filter (Example 15.3) (the passband gain is multiplied by the factor 40 to show the passband ripple).

15.8 GENERALIZATION

As was demonstrated in Chap. 9, there are four types of constant-delay nonrecursive filters. The impulse response can be *symmetrical* or *antisymmetrical*, and the filter length can be *odd* or *even*. In the preceding sections, we considered the design of filters with symmetrical impulse response and odd length. In this section, we show that the Remez algorithm can also be applied for the design of the three other types of filters.

15.8.1 Antisymmetrical Impulse Response and Odd Filter Length

Assuming that $\omega_s = 2\pi$, the frequency response of a nonrecursive filter with *antisymmetrical* impulse response and *odd* length can be expressed as

$$H(e^{j\omega T}) = e^{-jc\omega} j P_c'(\omega)$$

where

$$P_c'(\omega) = \sum_{k=1}^{c} a_k \sin k\omega \tag{15.52}$$

$$a_k = 2h(c-k) \qquad \text{for } k = 1, 2, \ldots, c$$

$$c = (N-1)/2$$

(see Table 9.1).

A filter with a desired frequency response $e^{-jc\omega}jD(\omega)$ can be designed by constructing the error function

$$E(\omega) = W(\omega)[D(\omega) - P'_c(\omega)] \tag{15.53}$$

and then minimizing $|E(\omega)|$ with respect to some compact subset of the frequency interval $[0, \pi]$. From Eq. (15.52), $P'_c(\omega)$ can be expressed as [6]

$$P'_c(\omega) = \sin\omega\, P_{c-1}(\omega) \tag{15.54}$$

where

$$P_{c-1}(\omega) = \sum_{k=0}^{c-1} \tilde{c}_k \cos k\omega \tag{15.55a}$$

and

$$a_1 = \tilde{c}_0 - \tfrac{1}{2}\tilde{c}_2 \tag{15.55b}$$

$$a_k = \tfrac{1}{2}(\tilde{c}_{k-1} - \tilde{c}_{k+1}) \qquad \text{for } k = 2, 3, \ldots, c-2 \tag{15.55c}$$

$$a_{c-1} = \tfrac{1}{2}\tilde{c}_{c-2} \tag{15.55d}$$

$$a_c = \tfrac{1}{2}\tilde{c}_{c-1} \tag{15.55e}$$

Hence Eq. (15.53) can be put in the form

$$E(\omega) = \tilde{W}(\omega)[\tilde{D}(\omega) - \tilde{P}(\omega)] \tag{15.56}$$

where

$$\begin{aligned}
\tilde{W}(\omega) &= Q(\omega)W(\omega)\\
\tilde{D}(\omega) &= D(\omega)/Q(\omega)\\
\tilde{P}(\omega) &= P_{c-1}(\omega)\\
Q(\omega) &= \sin\omega
\end{aligned}$$

Evidently, Eq. (15.56) is of the same form as Eq. (15.3), and on proceeding as in Sec. 15.2 one can obtain the system of equations

$$\begin{bmatrix}
1 & \cos\hat{\omega}_0 & \cos 2\hat{\omega}_0 & \cdots & \cos(c-1)\hat{\omega}_0 & \dfrac{1}{\tilde{W}(\hat{\omega}_0)}\\
1 & \cos\hat{\omega}_1 & \cos 2\hat{\omega}_1 & \cdots & \cos(c-1)\hat{\omega}_1 & \dfrac{-1}{\tilde{W}(\hat{\omega}_1)}\\
\vdots & \vdots & \vdots & & \vdots & \vdots\\
1 & \cos\hat{\omega}_r & \cos 2\hat{\omega}_r & \cdots & \cos(c-1)\hat{\omega}_r & \dfrac{(-1)^r}{\tilde{W}(\hat{\omega}_r)}
\end{bmatrix}
\begin{bmatrix} a_0\\ a_1\\ \vdots\\ a_{c-1}\\ \delta \end{bmatrix}
=
\begin{bmatrix} \tilde{D}(\hat{\omega}_0)\\ \tilde{D}(\hat{\omega}_1)\\ \vdots\\ \tilde{D}(\hat{\omega}_r) \end{bmatrix}$$

where $r = c$ is the number of cosine functions in $P_{c-1}(\omega)$. This system of equations is the same as that in Eq. (15.12) except that the number of extremals has been reduced from $c+2$ to $c+1$; therefore, the application of the Remez algorithm follows the methodology detailed in Secs. 15.2 and 15.3.

The use of Algorithm 1 or 4 yields the optimum $P_{c-1}(\omega)$ and from Eq. (15.54), the cosine function $P'_c(\omega)$ can be formed. Now $jP'_c(\omega)$ is the frequency response of a noncausal version of the required filter. The impulse response of this filter can be obtained as

$$h_0(n) = -h_0(-n) = -\frac{1}{N}\left[\sum_{k=1}^{c} 2P'_c(k\Omega)\sin\left(\frac{2\pi kn}{N}\right)\right] \tag{15.57}$$

for $n = 0, 1, 2, \ldots, c$, where $\Omega = 2\pi/N$, by using the inverse discrete Fourier transform. The impulse response of the corresponding causal filter is given by

$$h(n) = h_0(n-c)$$

for $n = 0, 1, 2, \ldots, N-1$.

15.8.2 Even Filter Length

The frequency response of a filter with *symmetrical* impulse response and *even* length is given by

$$H(e^{j\omega T}) = e^{-jc\omega}P_d(\omega)$$

where

$$P_d(\omega) = \sum_{k=1}^{d} b_k \cos\left(k - \frac{1}{2}\right)\omega$$

$$b_k = 2h(d-k) \qquad \text{for } k = 1, 2, \ldots, d$$

$$d = N/2$$

(see Table 9.1). $P_d(\omega)$ can be expressed as

$$P_d(\omega) = \cos\frac{\omega}{2}\,P_{d-1}(\omega)$$

where

$$P_{d-1}(\omega) = \sum_{k=0}^{d-1} \tilde{b}_k \cos k\omega \tag{15.58a}$$

and

$$b_1 = \tilde{b}_0 + \tfrac{1}{2}\tilde{b}_1 \tag{15.58b}$$

$$b_k = \frac{1}{2}(\tilde{b}_{k-1} + \tilde{b}_k) \qquad \text{for } k = 2, 3, \ldots, d-1 \tag{15.58c}$$

$$b_d = \tfrac{1}{2}\tilde{b}_{d-1} \tag{15.58d}$$

Proceeding as in the case of antisymmetrical impulse response, an error function of the form given in Eq. (15.56) can be constructed with

$$\tilde{P}(\omega) = P_{d-1}(\omega)$$

and

$$Q(\omega) = \cos\frac{\omega}{2}$$

Similarly, if the impulse response is *antisymmetrical* and the filter length is *even*, we have

$$H(e^{j\omega T}) = e^{-jc\omega} j P'_d(\omega)$$

where

$$\begin{aligned} P'_d(\omega) &= \sum_{k=1}^{d} b_k \sin\left(k - \tfrac{1}{2}\right)\omega \\ b_k &= 2h(d-k) \qquad \text{for } k = 1, 2, \ldots, d \\ d &= N/2 \end{aligned}$$

$P'_d(\omega)$ can now be expressed as

$$P'_d(\omega) = \sin\frac{\omega}{2}\, P_{d-1}(\omega)$$

where

$$P_{d-1}(\omega) = \sum_{k=0}^{d-1} \tilde{d}_k \cos k\omega \tag{15.59a}$$

and

$$b_1 = \tilde{d}_0 - \tfrac{1}{2}\tilde{d}_1 \tag{15.59b}$$

$$b_k = \tfrac{1}{2}(\tilde{d}_{k-1} - \tilde{d}_k) \qquad \text{for } k = 2, 3, \ldots, d-1 \tag{15.59c}$$

$$b_d = \tfrac{1}{2}\tilde{d}_{d-1} \tag{15.59d}$$

As in the previous case, an error function of the form given in Eq. (15.56) can be obtained with

$$\tilde{P}(\omega) = P_{d-1}(\omega)$$

and

$$Q(\omega) = \sin\frac{\omega}{2}$$

The various polynomials for the four types of nonrecursive filters are summarized in Table 15.7.

Table 15.7 Functions $H(e^{j\omega T})$, $Q(\omega)$, and $\tilde{P}(\omega)$ for the various types of nonrecursive filters

$h(n)$	N	$H(e^{j\omega T})$	$Q(\omega)$	$\tilde{P}(\omega)$
Symmetrical	odd	$e^{-jc\omega}P_c(\omega)$	1	$P_c(\omega)=\sum_{k=0}^{c}\ a_k\cos k\omega$
	even	$e^{-jc\omega}P_d(\omega)$ $P_d(\omega)=\sum_{k=1}^{d}\ b_k\cos\left(k-\frac{1}{2}\right)\omega$	$\cos\frac{\omega}{2}$	$P_{d-1}(\omega)=\sum_{k=0}^{d-1}\tilde{b}_k\cos k\omega$
Antisymmetrical	odd	$e^{-jc\omega}jP'_c(\omega)$ $P'_c(\omega)=\sum_{k=1}^{c}\ a_k\sin k\omega$	$\sin\omega$	$P_{c-1}(\omega)=\sum_{k=0}^{c-1}\tilde{c}_k\cos k\omega$
	even	$e^{-jc\omega}jP'_d(\omega)$ $P'_d(\omega)=\sum_{k=1}^{d}\ b_k\sin\left(k-\frac{1}{2}\right)\omega$	$\sin\dfrac{\omega}{2}$	$P_{d-1}(\omega)=\sum_{k=0}^{d-1}\tilde{d}_k\cos k\omega$

$a_0=h(c)$ $\quad a_k=2h(c-k)$ $\quad c=(N-1)/2$
$b_k=2h(d-k)$ $\quad d=N/2$

15.9 DIGITAL DIFFERENTIATORS

The Remez algorithm can be easily applied for the design of *equiripple digital differentiators*. The ideal frequency response of a causal differentiator is of the form $e^{-jc\omega}jD(\omega)$ where

$$D(\omega)=\omega \qquad \text{for } 0<|\omega|<\pi \tag{15.60}$$

and $c=(N-1)/2$ (see Sec. 9.5). From Table 15.7, we note that differentiators can be designed in terms of filters with antisymmetrical impulse response of either odd or even length.

15.9.1 Problem Formulation

Assuming odd filter length, Eqs. (15.53) and (15.60) give the error function

$$E(\omega)=W(\omega)[\omega-P'_c(\omega)] \qquad \text{for } 0<\omega\le\omega_p$$

where ω_p is the required bandwidth. Constant absolute or relative error may be required, depending on the application at hand. Hence $W(\omega)$ can be chosen to be either unity or $1/\omega$. In the latter case, $E(\omega)$ can be expressed as

$$E(\omega)=1-\frac{1}{\omega}P'_c(\omega) \qquad \text{for } 0<\omega\le\omega_p$$

and from Eq. (15.54)

$$E(\omega)=1-\frac{\sin\omega}{\omega}P_{c-1}(\omega) \qquad \text{for } 0<\omega\le\omega_p \tag{15.61}$$

Therefore, the error function can be expressed as in Eq. (15.56) with

$$\tilde{W}(\omega) = \frac{1}{\tilde{D}(\omega)} = \frac{\sin\omega}{\omega}$$
$$\tilde{P}(\omega) = P_{c-1}(\omega)$$

15.9.2 First Derivative

In Algorithm 4, the first derivative of $|E(\omega)|$ with respect to ω is required. From Eq. (15.61), one can show that

$$\frac{d|E(\omega)|}{d\omega} = \text{sgn}\left[1 - \frac{\sin\omega}{\omega}P_{c-1}(\omega)\right] \times \left[\frac{\sin\omega - \omega\cos\omega}{\omega^2}P_{c-1}(\omega) - \frac{\sin\omega}{\omega}\frac{dP_{c-1}(\omega)}{d\omega}\right] \tag{15.62}$$

The first derivative of $P_{c-1}(\omega)$ can be computed by using the formulas in Sec. 15.6, except that the number of extremals is reduced from $c+2$ to $c+1$. The value of $P_{c-1}(\omega)$ can be computed by using Eq. (15.35) with c replaced by $c-1$. If $\hat{\omega}_i$ is an extremal, then Eq. (15.61) yields

$$P_{c-1}(\hat{\omega}_i) = [1 - (-1)^i\delta]\frac{\hat{\omega}_i}{\sin\hat{\omega}_i}$$

since $E(\hat{\omega}_i) = (-1)^i\delta$.

In Algorithm 4, the second derivative of $|E(\omega)|$ with respect to ω is used to determine whether there is a maximum or minimum at $\omega = 0$. For differentiators, this information is more easily determined by computing the quantity

$$G_2' = |E(w_1)| - |E(0)|$$

where w_1 is the interval between successive samples. Depending on whether G_2' is positive or negative, $|E(\omega)|$ has a minimum or maximum at $\omega = 0$.

15.9.3 Prescribed Specifications

A digital differentiator is fully specified by the constraint

$$|E(\omega)| \le \delta_p \qquad \text{for } 0 < \omega \le \omega_p$$

where δ_p is the maximum passband error and ω_p is the bandwidth of the differentiator.

The differentiator length N that will just satisfy the required specifications is not normally known a priori and, although it may be determined on a hit and miss basis, a large number of designs may need to be carried out. In filters with approximately piecewise-constant amplitude responses, N can be predicted using the empirical formula of Eq. (15.51). In the case of differentiators, N can be predicted by noting a useful property of digital differentiators. If δ and δ_1 are the maximum passband errors in differentiators of lengths N and N_1, respectively, then the quantity $\ln(\delta/\delta_1)$ is *approximately linear* with respect to $N - N_1$ for a wide range of values of N_1 and ω_p, as illustrated in Fig. 15.8. Assuming linearity, we can show that [16]

$$N = N_1 + \frac{\ln(\delta/\delta_1)}{\ln(\delta_2/\delta_1)}(N_2 - N_1) \tag{15.63}$$

where δ_2 is the maximum passband error in a differentiator of length N_2.

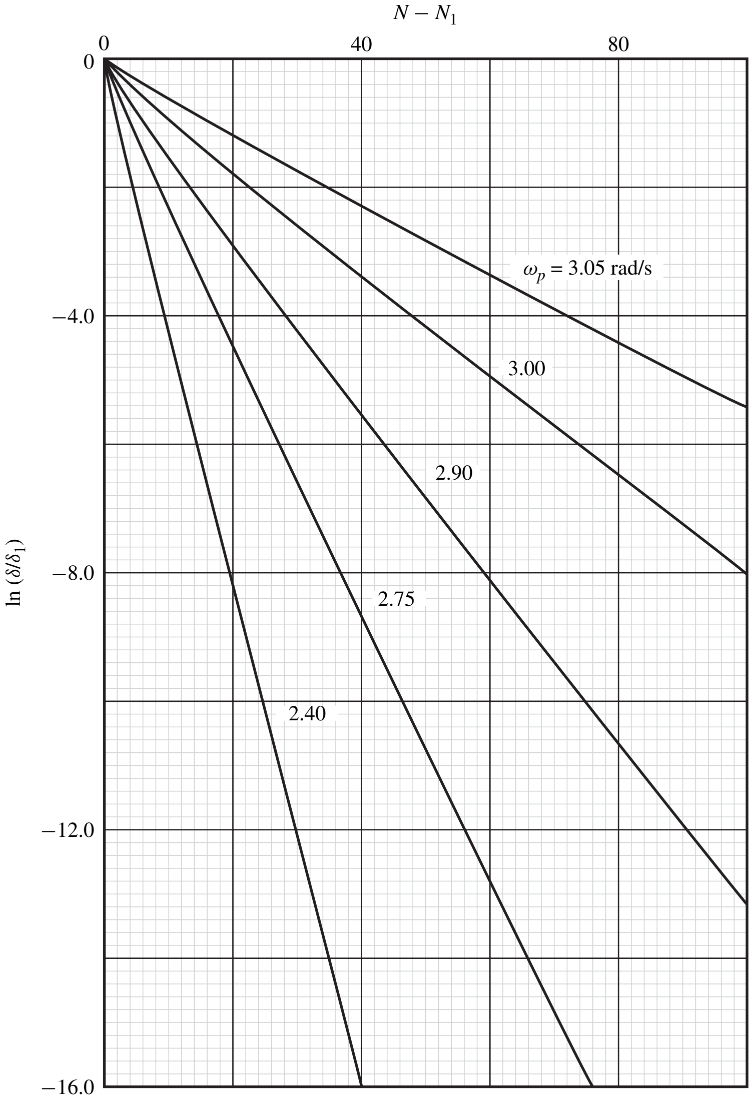


Figure 15.8 Variation of $\ln(\delta/\delta_1)$ versus $N - N_1$ for different values of ω_p and $N_1 = 11$.

By designing two low-order differentiators, a fairly accurate prediction of the required value of N can be obtained by using Eq. (15.63). A design algorithm based on this formula is as follows:

Algorithm 6: Design of digital differentiators satisfying prescribed specifications

1. Design a differentiator of length N_1, and find δ_1.
2. Design a differentiator of length $N_2 = N_1 + 2$ and find δ_2.
3. If $\delta_2 \leq \delta_p < \delta_1$, go to step 7.
4. Set $\delta = \delta_p$ and compute N using Eq. (15.63); set $N_3 = \text{Int}\,(N + 0.5)$; if N_3 is even and a differentiator of odd length is required, then set $N_3 = N_3 + 1$.
5. Design a differentiator of length N_3 and find δ_3.
   - (A) If $\delta_3 > \delta_p$, then do:
     - (a) Set $N_3 = N_3 + 2$, design a differentiator of length N_3, and find δ_3;
     - (b) If $\delta_3 \leq \delta_p$, then go to step 6; else, go to step $5(A)(a)$.
   - (B) If $\delta_3 < \delta_p$, then do:
     - (a) Set $N_3 = N_3 - 2$, design a differentiator of length N_3, and find δ_3;
     - (a) If $\delta_3 > \delta_p$, then go to step 7; else, go to step $5(B)(a)$.
6. Use the last set of extremals and the corresponding value of N to obtain the impulse response of the required differentiator and stop.
7. Use the last but one set of extremals and the corresponding value of N to obtain the impulse response of the required differentiator and stop. ●

Example 15.4 In an application, a digital differentiator is required which should satisfy the following specifications:

- Odd differentiator length
- Bandwidth ω_p: 2.5 rad/s
- Maximum passband ripple δ_p: 1.0×10^{-6}
- Sampling frequency ω_s: 2π rad/s

Design the lowest-order differentiator that will satisfy the specifications.

■ **Solution**

The design was carried out using Algorithm 6 in conjunction with Algorithm 4; in Algorithm 4 the relative error of Eq. (15.61) was minimized. The progress of the design is illustrated in Table 15.8. First, differentiators of lengths 21 and 23 were designed and the

Table 15.8 Progress in design of digital (differentiator Example 15.4)

N	Iters.	FE's	δ_p
21	4	141	$7.649E-4$
23	5	187	$3.786E-4$
43	5	616	$4.078E-7$
41	6	538	$8.069E-7$
39	6	500	$1.582E-6$

required N to satisfy the specifications was predicted to be 43 using Eq. (15.63). This differentiator length was found to oversatisfy the specifications, and designs for lengths 41 and 39 were then carried out. The design for $N = 39$ violates the specifications, as can be seen in Table 15.8; therefore, the optimum differentiator length is 41. The impulse response of this differentiator is given in Table 15.9. The amplitude response and passband relative error of the differentiator are plotted in Fig. 15.9a and b. ■

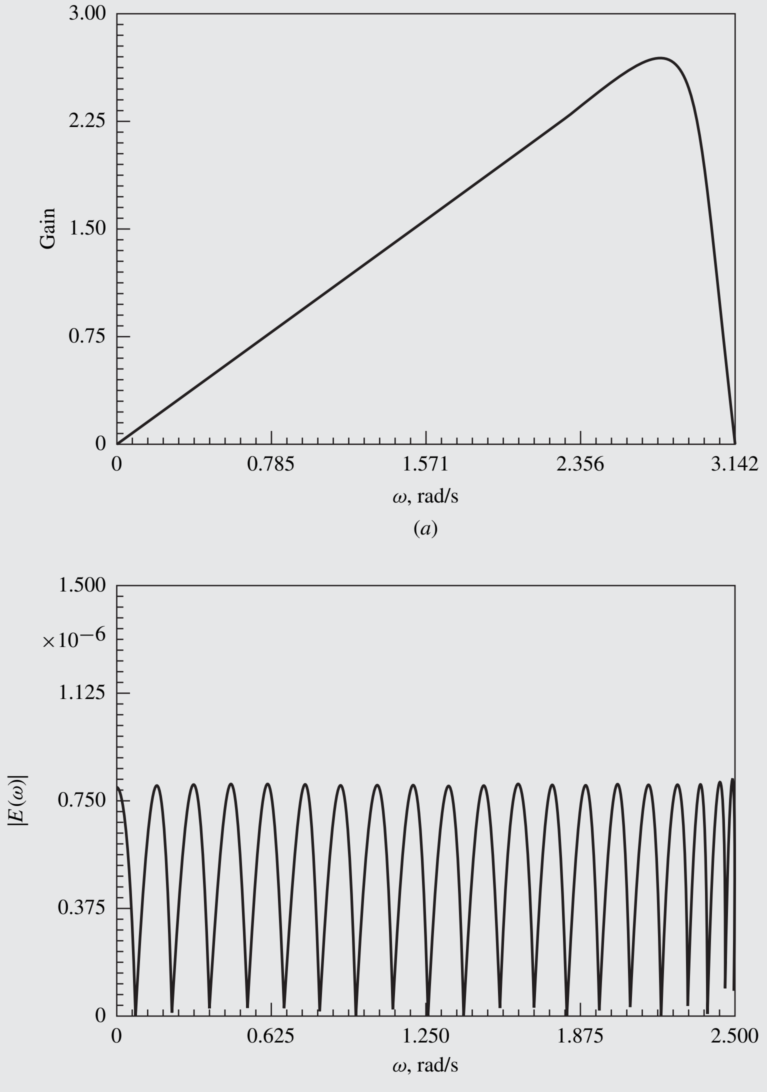


Figure 15.9 Design of digital differentiator (Example 15.4): (*a*) Amplitude response, (*b*) passband relative error.

Table 15.9 Impulse response of digital differentiator (Example 15.4)

n	$h_0(n) = -h_0(-n)$	n	$h_0(n) = -h_0(-n)$
0	0.0	11	$-1.305326E - 2$
1	$-9.852395E - 1$	12	$7.955151E - 3$
2	$4.710789E - 1$	13	$-4.626299E - 3$
3	$-2.914014E - 1$	14	$2.544983E - 3$
4	$1.966634E - 1$	15	$-1.309224E - 3$
5	$-1.371947E - 1$	16	$6.197315E - 4$
6	$9.651420E - 2$	17	$-2.633737E - 4$
7	$-6.751749E - 2$	18	$9.638584E - 5$
8	$4.653727E - 2$	19	$-2.795288E - 5$
9	$-3.138375E - 2$	20	$4.916591E - 6$
10	$2.058332E - 2$		

15.10 ARBITRARY AMPLITUDE RESPONSES

Very frequently nonrecursive filters are required whose amplitude responses cannot be described by analytical functions. For example, in the design of two-dimensional filters (see Sec. 18.6) through the singular-value decomposition [17, 18], the required two-dimensional filter is obtained by designing a set of one-dimensional digital filters whose amplitude responses turn out to have arbitrary shapes. In these applications, the desired amplitude response $D(\omega)$ is specified in terms of a table that lists a prescribed set of frequencies and the corresponding values of the required filter gain. Filters of this class can be readily designed by employing some interpolation scheme that can be used to evaluate $D(\omega)$ and its first derivative with respect to ω at any ω. A suitable scheme is to fit a set of third-order polynomials to the prescribed amplitude response. An interpolation scheme of this type is used in the design of recursive filters in the next chapter and is described in detail in Sec. 16.6.

15.11 MULTIBAND FILTERS

The algorithms presented in the previous sections can also be used to design *multiband* filters. While there is no theoretical upper limit on the number of bands, in practice, the design tends to become more and more difficult as the number of bands is increased. The reason is that the difference between the number of possible maxima in the error function and the number of extremals increases linearly with the number of bands, e.g., if the number of bands is 8, then the difference is 14 (see Sec. 15.3.4). As a consequence, the number of potential extremals that need to be rejected is large and the available rejection techniques become somewhat inefficient. The end result is that the number of iterations is increased quite significantly, and convergence is slow and sometimes impossible.

In mathematical terms, the problem discussed in the previous paragraph is attributed to the fact that, in the weighted-Chebyshev methods considered in this chapter, the approximating polynomial becomes seriously *underdetermined* if the number of bands exceeds three. The problem can be overcome by using the generalized Remez method described in Ref. [14]. This method is based on a different formulation of the design problem and leads to three types of equiripple filters, namely, *maximal-ripple*, *extra-ripple*, and *weighted-Chebyshev* filters. In the case of maximal-ripple filters, the approximating polynomial is fully determined; in the extra-ripple case, it is less underdetermined than the approximating polynomial in the methods described. Therefore, for filters with more than five bands, the method in Ref. [14] is preferred.

Example 15.5 In an application, a nonrecursive equiripple 5-band filter is required which should satisfy the specifications in Table 15.10. The sampling frequency is 2π. Design the lowest-order filter that will satisfy the specifications.

Table 15.10 Specifications of 5-band filter (Example 15.5)

Band:	1	2	3	4	5
$D(\omega)$	1.00	0.00	1.00	0.00	1.00
A_p, dB	0.50	—	0.75	—	1.00
A_a, dB	—	50.00	—	30.00	—
ω_L, rad/s	0.00	0.80	1.50	2.10	2.80
ω_R, rad/s	0.60	1.25	1.90	2.60	π

■ **Solution**

The use of Algorithm 4 in conjunction with Algorithm 5 gave a filter of length 61. The progress of the design is illustrated in Table 15.11. The impulse response of the filter obtained is given in Table 15.12, and the corresponding amplitude response is plotted in Fig. 15.10. As can be seen, the required specifications are satisfied. ■

Table 15.11 Progress in design of 5-band filter (Example 15.5)

N	Iters.	FE's	A_{p1}, dB	A_{a2}, dB	A_{p3}, dB	A_{a4}, dB	A_{p5}, dB
61	9	913	0.453	50.46	0.679	30.86	0.905
59	19	2219	0.539	49.35	0.808	29.35	1.077

Table 15.12 Impulse response of 5-band filter (Example 15.5)

n	$h_0(n) = h_0(-n)$	n	$h_0(n) = h_0(-n)$
0	$5.608208E-1$	16	$-8.164458E-4$
1	$4.013174E-2$	17	$-3.884179E-4$
2	$1.006767E-1$	18	$2.625242E-3$
3	$4.198731E-2$	19	$-1.130791E-2$
4	$2.414087E-1$	20	$9.190432E-3$
5	$-1.248415E-1$	21	$8.761118E-3$
6	$-1.019101E-1$	22	$6.476604E-3$
7	$6.608448E-3$	23	$9.610168E-3$
8	$-1.355327E-2$	24	$-1.976094E-2$
9	$4.780217E-3$	25	$-1.075689E-2$
10	$-1.549769E-2$	26	$3.013727E-3$
11	$3.468520E-2$	27	$-2.707701E-3$
12	$-8.299265E-4$	28	$-2.549441E-3$
13	$4.694733E-2$	29	$-9.605488E-3$
14	$2.641761E-3$	30	$1.495353E-2$
15	$-5.336269E-2$	–	–

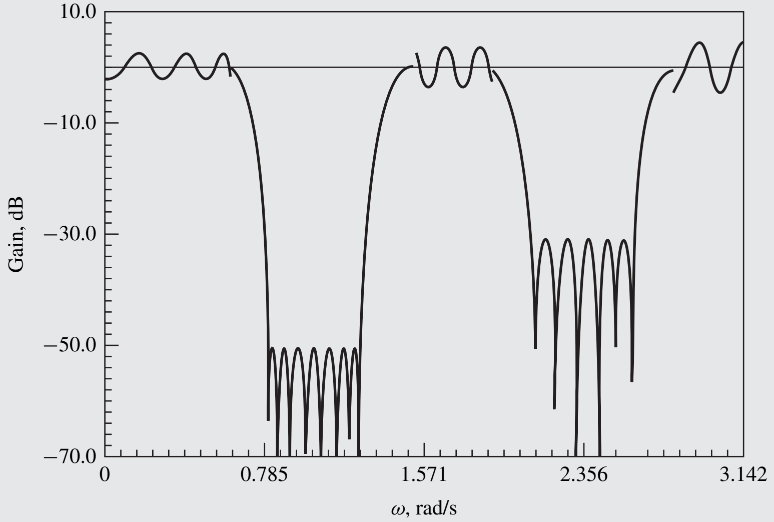


Figure 15.10 Amplitude response of equiripple 5-band filter (Example 15.5) (the passband gain is multiplied by the factor 10 to show the passband ripple).

The required filter order for multiband filters can be predicted by using the formula in Eq. (15.51), as was stated earlier. A generalized version of this formula, which gives improved results, can be found in Ref. [14].

REFERENCES

[1] O. Herrmann, “Design of nonrecursive digital filters with linear phase,” *Electron. Lett.*, vol. 6, pp. 182–184, May 1970.

[2] E. Hofstetter, A. Oppenheim, and J. Siegel, “A new technique for the design of non-recursive digital filters,” *5th Annual Princeton Conf. Information Sciences and Systems*, pp. 64–72, Mar. 1971.

[3] T. W. Parks and J. H. McClellan, “Chebyshev approximation for nonrecursive digital filters with linear phase,” *IEEE Trans. Circuit Theory*, vol. 19, pp. 189–194, Mar. 1972.

[4] T. W. Parks and J. H. McClellan, “A program for the design of linear phase finite impulse response digital filters,” *IEEE Trans. Audio Electroacoust.*, vol. 20, pp. 195–199, Aug. 1972.

[5] L. R. Rabiner and O. Herrmann, “On the design of optimum FIR low-pass filters with even impulse response duration,” *IEEE Trans. Audio Electroacoust.*, vol. 21, pp. 329–336, Aug. 1973.

[6] J. H. McClellan and T. W. Parks, “A unified approach to the design of optimum FIR linear-phase digital filters,” *IEEE Trans. Circuit Theory*, vol. 20, pp. 697–701, Nov. 1973.

[7] J. H. McClellan, T. W. Parks, and L. R. Rabiner, “A computer program for designing optimum FIR linear phase digital filters,” *IEEE Trans. Audio Electroacoust.*, vol. 21, pp. 506–526, Dec. 1973.

[8] L. R. Rabiner, J. H. McClellan, and T. W. Parks, “FIR digital filter design techniques using weighted Chebyshev approximation,” *Proc. IEEE*, vol. 63, pp. 595–610, Apr. 1975.

[9] J. H. McClellan, T. W. Parks, and L. R. Rabiner, “FIR linear phase filter design program,” *Programs for Digital Signal Processing*, New York: IEEE Press, pp. 5.1-1–5.1-13, 1979.

[10] A. Antoniou, “Accelerated procedure for the design of equiripple nonrecursive digital filters,” *Proc. Inst. Elect. Eng.*, Part G, vol. 129, pp. 1–10, Feb. 1982 (see vol. 129, p. 107, June 1982 for errata).

[11] A. Antoniou, “New improved method for the design of weighted-Chebyshev, nonrecursive, digital filters,” *IEEE Trans. Circuits Syst.*, vol. 30, pp. 740–750, Oct. 1983.

[12] E. W. Cheney, *Introduction to Approximation Theory*, New York: McGraw-Hill, pp. 72–100, 1996.

[13] E. Ya. Remes, *General Computational Methods for Tchebycheff Approximation*, Kiev, 1957 (Atomic Energy Commission Translation 4491, pp. 1–85).

[14] D. J. Shpak and A. Antoniou, “A generalized Reméz method for the design of FIR digital filters,” *IEEE Trans. Circuits Syst.*, vol. 37, pp. 161–174, Feb. 1990.

[15] O. Herrmann, L. R. Rabiner, and D. S. K. Chan, “Practical design rules for optimum finite impulse response low-pass digital filters,” *Bell Syst. Tech. J.*, vol. 52, pp. 769–799, Jul.-Aug. 1973.

[16] A. Antoniou and C. Charalambous, “Improved design method for Kaiser differentiators and comparison with equiripple method,” *Proc. Inst. Elect. Eng.*, Part E, vol. 128, pp. 190–196, Sept. 1981.

[17] A. Antoniou and W.-S. Lu, “Design of two-dimensional digital filters by using the singular value decomposition,” *IEEE Trans. Circuits Syst.*, vol. 34, pp. 1191–1198, Oct. 1987.

[18] W.-S. Lu, H.-P. Wang, and A. Antoniou, "Design of two-dimensional FIR digital filters using the singular-value decomposition," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 35–46, Jan. 1990.

ADDITIONAL REFERENCES

Adams, J. W., "FIR digital filters with least-squares stopbands subject to peak-gain constraints," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 376–388, Apr. 1991.

Karam, L. J. and J. H. McClellan,"Complex Chebyshev approximation for FIR filter design," *IEEE Trans. Circuits Syst.*-II, vol. 42, pp. 207–216, Mar. 1995.

W.-S. Lu, "Design of FIR filters with discrete coefficients: A semidefinite programming relaxation approach," *in Proc. IEEE Int. Symp. Circuits and Systems,* 2001, vol. 2, pp. 297–300, Sydney, Australia, May 2001.

PROBLEMS

15.1. A noncausal nonrecursive filter has a frequency response $P_c(\omega)$. The filter has a symmetrical impulse response represented by $h_0(n)$ for $-c \le n \le c$, where $c = (N-1)/2$. Using the inverse discrete Fourier transform, show that the impulse response of the filter is given by Eq. (15.21).

15.2. Show that δ and $P_c(\omega)$ given by Eqs. (15.16) and (15.17) can be expressed as in Eqs. (15.34) and (15.35), respectively.

15.3. Show that coefficients β_{km}, α_{km}, and C_{km}, which are used to compute δ and $P_c(\omega)$, can be expressed as in Eqs. (15.36)–(15.38).

15.4. Write a computer program based on the Remez algorithm (Algorithm 1) that can be used for the design of filters. Use the exhaustive step-by-step search method in Algorithm 2 in conjunction with the scheme in Algorithm 3 for the rejection of superfluous potential extremals. Then use a routine that will reject the $\rho - r$ superfluous potential extremals $\widehat{\omega}_i$ on the basis of the lowest error $|E(\widehat{\omega}_i)|$ (see Sec. 15.3.4) as an alternative rejection scheme and check whether there is a change in the computational efficiency of the program.

15.5. Show that for any frequency including the last extremal of the last band but excluding all other extremals, the first derivative of $P_c(\omega)$ with respect to ω is given by the formula in Eq. (15.46).

15.6. Show that for all extremals other than the last extremal of the last band, the first derivative of $P_c(\omega)$ with respect to ω is given by the formula in Eq. (15.47).

15.7. Show that the first derivative of $P_c(\omega)$ with respect to ω is zero at $\omega = 0$ and $\omega = \pi$ (see Eq. (15.48)). Hence show that $|E(\omega)|$ has a local maximum or minimum at these frequencies.

15.8. Show that for $\omega = 0$ if no extremal occurs at zero or for $\omega = \pi$ under all circumstances, the second derivative of $P_c(\omega)$ with respect to ω is given by Eq. (15.49).

15.9. Show that if there is an extremal at $\omega = 0$, then the second derivative of $P_c(\omega)$ with respect to ω at $\omega = 0$ is given by Eq. (15.50).

15.10. Write a computer program based on the Remez algorithm that can be used for the design of filters. Use the selective step-by-step search method of Sec. 15.4.1.

15.11. The cubic-interpolation search of Sec. 15.4.2 requires the evaluation of constants d, c, b, β, γ, θ, and ψ given by Eqs. (15.25)–(15.31). Derive the formulas for these constants.

15.12. Modify the program of Prob. 15.10 to include the cubic-interpolation search of Sec. 15.4.2 (see Algorithm 4).

15.13. Design a nonrecursive equiripple lowpass filter using the Remez algorithm (*a*) with the exhaustive search of Sec. 15.3.2, (*b*) with the selective step-by-step search of Sec. 15.4.1, and (*c*) with the selective step-by-step search in conjunction with the cubic-interpolation search of Sec. 15.4.2. Compare the results obtained. The required specifications are as follows:

- Filter length N: 21
- Passband edge ω_p: 1.0 rad/s
- Stopband edge ω_a: 1.5 rad/s
- Ratio δ_p/δ_a: 18.0
- Sampling frequency ω_s: 2π rad/s

15.14. Design a nonrecursive equiripple bandstop filter using the Remez algorithm (*a*) with the exhaustive search, (*b*) with the selective step-by-step search, and (*c*) with the selective step-by-step search in conjunction with the cubic-interpolation search. Compare the results obtained. The required specifications are as follows:

- Filter length N: 33
- Lower passband edge ω_{p1}: 0.8 rad/s
- Upper passband edge ω_{p2}: 2.1 rad/s
- Lower stopband edge ω_{a1}: 1.2 rad/s
- Upper stopband edge ω_{a2}: 1.8 rad/s
- Ratio δ_p/δ_a: 23.0
- Sampling frequency ω_s: 2π rad/s

15.15. Modify the program in Prob. 15.10 to include an option for the design of filters satisfying prescribed specifications. Use Algorithm 5.

15.16. In an application, a nonrecursive equiripple highpass filter is required, which should satisfy the following specifications:

- Odd filter length
- Maximum passband ripple A_p: 0.1 dB
- Minimum stopband attenuation A_a: 50.0 dB
- Passband edge ω_p: 1.8 rad/s
- Stopband edge ω_a: 1.0 rad/s
- Sampling frequency ω_s: 2π rad/s

Design the lowest-order filter that will satisfy the specifications.

15.17. In an application, a nonrecursive equiripple bandpass filter is required, which should satisfy the following specifications:

- Odd filter length
- Maximum passband ripple A_p: 0.1 dB
- Minimum stopband attenuation A_a: 60.0 dB
- Lower passband edge ω_{p1}: 1.0 rad/s
- Upper passband edge ω_{p2}: 1.6 rad/s
- Lower stopband edge ω_{a1}: 0.6 rad/s
- Upper stopband edge ω_{a2}: 2.0 rad/s
- Sampling frequency ω_s: 2π rad/s

Design the lowest-order filter that will satisfy the specifications.

15.18. Show that the sine polynomial $P'_c(\omega)$ of Eq. (15.52) can be expressed as in Eq. (15.54) where $P_{c-1}(\omega)$ is given by Eq. (15.55a).

15.19. A noncausal nonrecursive filter has a frequency response $jP_c(\omega)$. The filter has an antisymmetrical impulse response represented by $h_0(n)$ for $-c \leq n \leq c$, where $c = (N-1)/2$. Using the inverse discrete Fourier transform, show that the impulse response of the filter is given by Eq. (15.57).

15.20. The relative error in the design of digital differentiators is given by Eq. (15.61). Show that the first derivative of $|E(\omega)|$ with respect to ω is given by Eq. (15.62).

15.21. Write a computer program based on the Remez algorithm that can be used for the design of digital differentiators. Use the selective step-by-step search method in conjunction with the cubic-interpolation search.

15.22. Using the program in Prob. 15.21, design a digital differentiator of length $N = 41$ and bandwidth $\omega_p = 3.0$ rad/s. The sampling frequency is 2π rad/s.

15.23. If δ and δ_1 are the maximum passband errors in digital differentiators of lengths N and N_1, respectively, then the quantity $\ln(\delta/\delta_1)$ is approximately linear with respect to $N - N_1$, as can be seen in Fig. 15.8. Assuming linearity, derive the prediction formula of Eq. (15.63).

15.24. Modify the program in Prob. 15.21 to include an option for the design of digital differentiators satisfying prescribed specifications. Use Algorithm 6.

15.25. In an application, a digital differentiator is required, which should satisfy the following specifications:

- Odd differentiator length
- Bandwidth ω_p: 2.75 rad/s
- Maximum passband ripple δ_p: 1.0×10^{-4}
- Sampling frequency ω_s: 2π rad/s

Design the lowest-order differentiator that will satisfy the specifications.

15.26. In an application, a nonrecursive equiripple 4-band filter is required, which should satisfy the specifications in Table P15.26. The sampling frequency is 2π. Design the lowest-order filter that will satisfy the specifications.

Table P15.26

Band:	**1**	**2**	**3**	**4**
$D(\omega)$	0.0	1.0	0.0	1.0
A_p, dB	—	0.1	—	0.4
A_a, dB	50.0	—	55.0	—
ω_L, rad/s	0.0	1.2	2.0	2.8
ω_R, rad/s	0.8	1.6	2.4	π

15.27. In an application, a nonrecursive equiripple 5-band filter is required, which should satisfy the specifications in Table P15.27. The sampling frequency is $\omega_s = 2\pi$. Design the lowest-order filter that will satisfy the specifications.

Table P15.27

Band:	**1**	**2**	**3**	**4**	**5**
$D(\omega)$	1.0	0.0	1.0	0.0	1.0
A_p, dB	0.8	—	0.4	—	1.0
A_a, dB	—	50.0	—	30.0	—
ω_L, rad/s	0.0	0.8	1.6	2.2	2.9
ω_R, rad/s	0.4	1.2	1.9	2.6	π

CHAPTER 16

DESIGN OF RECURSIVE FILTERS USING OPTIMIZATION METHODS

16.1 INTRODUCTION

In Chaps. 11 and 12, several methods for the solution of the approximation problem in recursive filters have been described. These methods lead to a complete description of the transfer function in closed form, either in terms of its zeros and poles or its coefficients. They are, as a consequence, very efficient and lead to very precise designs. Their main disadvantage is that they are applicable only for the design of filters with piecewise-constant amplitude responses, i.e., filters whose passband and stopband gains are constant and zero, respectively, to within prescribed tolerances.

An alternative approach for the solution of the approximation problem in digital filters is through the application of *optimization methods* [1–5]. In these methods, a discrete-time transfer function is assumed and an error function is formulated on the basis of some desired amplitude and/or phase response. A norm of the error function is then minimized with respect to the transfer-function coefficients. As the value of the norm approaches zero, the resulting amplitude or phase response approaches the desired amplitude or phase response. These methods are *iterative* and, as a result, they usually involve a large amount of computation. However, unlike the closed-form methods of Chaps. 11 and 12, they are suitable for the design of filters having arbitrary amplitude or phase responses. Furthermore, they often yield superior designs.

In this chapter, the application of optimization methods for the design of recursive digital filters is considered. The chapter begins with an introductory section that deals with the formulation of the design problem as an optimization problem, and then proceeds with fairly detailed descriptions of algorithms that can be used to solve the optimization problem. The algorithms presented are based on the so-called *quasi-Newton* method which has been explored by Davidon, Fletcher, Powell, Broyden, and others. The exposition of the material begins with algorithms that are primarily of conceptual value and gradually proceeds to algorithms of increasing complexity and scope. It concludes with some highly sophisticated algorithms that are practical, flexible, efficient, and reliable. Throughout the chapter, emphasis is placed on the application of the algorithms rather than their theoretical foundation and convergence properties. Readers who are interested in a more mathematical treatment of the subject may consult one of the standard textbooks on optimization theory and practice [6–10].

16.2 PROBLEM FORMULATION

Assume that the amplitude response of a recursive filter is required to approach some specified amplitude response as closely as possible. Such a filter can be designed in two general steps, as follows:

1. An objective function which is dependent on the difference between the actual and specified amplitude response is formulated.
2. The objective function obtained is minimized with respect to the transfer-function coefficients.

An *N*th-order recursive filter with N even can be represented by the transfer function

$$H(z) = H_0 \prod_{j=1}^{J} \frac{a_{0j} + a_{1j}z + z^2}{b_{0j} + b_{1j}z + z^2} \tag{16.1}$$

where a_{ij} and b_{ij} are real coefficients, $J = N/2$, and H_0 is a positive multiplier constant. The amplitude response of the filter can be expressed as

$$M(\mathbf{x}, \omega) = |H(e^{j\omega T})| \tag{16.2}$$

where

$$\mathbf{x} = [a_{01}\ a_{11}\ b_{01}\ b_{11}\ \cdots\ b_{1J}\ H_0]^T$$

is a column vector with $4J + 1$ elements and ω is the frequency.

Let $M_0(\omega)$ be the specified amplitude response and, for the sake of exposition, assume that it is piecewise continuous, as illustrated in Fig. 16.1. The difference between $M(\mathbf{x}, \omega)$ and $M_0(\omega)$ is, in effect, the approximation error and can be expressed as

$$e(\mathbf{x}, \omega) = M(\mathbf{x}, \omega) - M_0(\omega) \tag{16.3}$$

By sampling $e(\mathbf{x}, \omega)$ at frequencies $\omega_1,\ \omega_2, \ldots, \omega_K$, as depicted in Fig. 16.1, the column vector

$$\mathbf{E}(\mathbf{x}) = [e_1(\mathbf{x})\ e_2(\mathbf{x})\ \ldots\ e_K(\mathbf{x})]^T$$

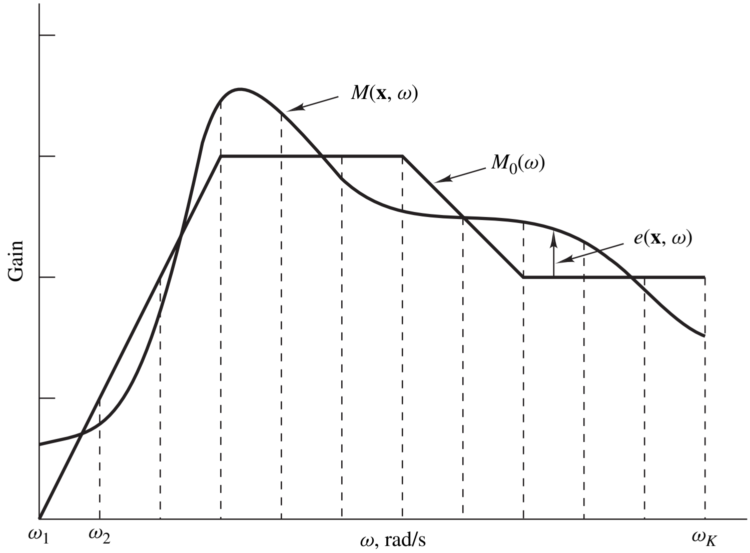


Figure 16.1 Formulation of error function.

can be formed where

$$e_i(\mathbf{x}) = e(\mathbf{x}, \omega_i) \tag{16.4}$$

for $i = 1, 2, \ldots, K$.

The approximation problem at hand can be solved by finding a point $\mathbf{x} = \breve{\mathbf{x}}$ such that

$$e_i(\breve{\mathbf{x}}) \approx 0$$

for $i = 1, 2, \ldots, K$. Assuming that a solution exists, a suitable *objective function* must first be formed which should satisfy a number of fundamental requirements. It should be a *scalar* quantity, and its minimization with respect to $\mathbf{x}$ should lead to the minimization of all the elements of $\mathbf{E}(\mathbf{x})$ in some sense. Further, it is highly desirable that it be *differentiable*. An objective function satisfying these requirements can be defined in terms of the L_p *norm* of $\mathbf{E}(\mathbf{x})$ as

$$\Psi(\mathbf{x}) = L_p = ||\mathbf{E}(\mathbf{x})||_p = \left[\sum_{i=1}^{K} |e_i(\mathbf{x})|^p\right]^{1/p}$$

where p is an integer.

Several special cases of the L_p norm are of particular interest. The L_1 *norm*, namely,

$$L_1 = \sum_{i=1}^{K} |e_i(\mathbf{x})|$$

is the sum of the magnitudes of the elements of $\mathbf{E}(\mathbf{x})$; the L_2 *norm* given by

$$L_2 = \left[\sum_{i=1}^{K} |e_i(\mathbf{x})|^2\right]^{1/2}$$

is the well-known *Euclidean norm*; and L_2^2 is the *sum of the squares* of the elements of $\mathbf{E}(\mathbf{x})$. In the case where $p = \infty$ and

$$\widehat{E}(\mathbf{x}) = \max_{1\le i\le K} |e_i(\mathbf{x})| \neq 0$$

we can write

$$\begin{aligned} L_\infty &= \lim_{p\to\infty} \left\{\sum_{i=1}^{K} |e_i(\mathbf{x})|^p\right\}^{1/p} \\ &= \widehat{E}(\mathbf{x}) \lim_{p\to\infty} \left\{\sum_{i=1}^{K} \left[\frac{|e_i(\mathbf{x})|}{\widehat{E}(\mathbf{x})}\right]^p\right\}^{1/p} \end{aligned} \tag{16.5}$$

Since each of the terms in the above summation is equal to or less than unity, we have

$$L_\infty = \widehat{E}(\mathbf{x})$$

With an objective function available, the required design can be obtained by solving the optimization problem

$$\underset{\mathbf{x}}{\text{minimize}}\ \Psi(\mathbf{x}) \tag{16.6}$$

If $\Psi(\mathbf{x})$ is defined in terms of L_2^2, a *least-squares* solution is obtained; if the L_∞ norm is used, a so-called *minimax* solution is obtained, since in this case the largest element in $\mathbf{E}(\mathbf{x})$ is minimized.

In digital filters, the magnitude of the largest amplitude-response error is usually required to be as small as possible and, therefore, minimax solutions are preferred.

16.3 NEWTON'S METHOD

The optimization problem of Eq. (16.6) can be solved by using an *unconstrained* optimization algorithm. Various classes of these algorithms have been developed in recent years, ranging from *steepest-descent* to *conjugate-direction* algorithms [6–10]. An important class of optimization algorithms that have been found to be very effective for the design of digital filters is the class of *quasi-Newton* algorithms. These are based on Newton's method for finding the minimum in quadratic *convex* functions.[1]

Consider a function $f(\mathbf{x})$ of n variables, where $\mathbf{x} = [x_1\ x_2\ \cdots\ x_n]^T$ is a column vector, and let $\boldsymbol{\delta} = [\delta_1\ \delta_2\ \cdots\ \delta_n]^T$ be a change in $\mathbf{x}$. If $f(\mathbf{x}) \in C^2$, that is, $f(\mathbf{x})$ has continuous second derivatives,

[1] *A two-variable convex function is one that represents a surface whose shape resembles a punch bowl.*

its Taylor series at point $\mathbf{x} + \boldsymbol{\delta}$ is given by

$$f(\mathbf{x}+\boldsymbol{\delta}) = f(\mathbf{x}) + \sum_{i=1}^{n} \frac{\partial f(\mathbf{x})}{\partial x_i}\delta_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}\delta_i\delta_j + o\left(||\boldsymbol{\delta}||_2^2\right) \tag{16.7}$$

where the remainder $o(||\boldsymbol{\delta}||_2^2)$ approaches zero faster than $||\boldsymbol{\delta}||_2^2$. If the remainder is negligible and a *stationary point* exists in the neighborhood of some point $\mathbf{x}$, it can be determined by differentiating $f(\mathbf{x} + \boldsymbol{\delta})$ with respect to elements δ_k for $k = 1, 2, \ldots, n$, and setting the result to zero. From Eq. (16.7), we obtain

$$\frac{\partial f(\mathbf{x}+\boldsymbol{\delta})}{\partial \delta_k} = \frac{\partial f(\mathbf{x})}{\partial x_k} + \sum_{i=1}^{n} \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_k}\delta_i = 0$$

for $k = 1, 2, \ldots, n$. This equation can be expressed in matrix form as

$$\mathbf{g} = -\mathbf{H}\boldsymbol{\delta} \tag{16.8}$$

where

$$\mathbf{g} = \boldsymbol{\nabla} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} & \frac{\partial f(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}^T$$

and

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{bmatrix}$$

are the *gradient* vector and *Hessian* matrix (or simply the gradient and Hessian) of $f(\mathbf{x})$, respectively. Therefore, the value of $\boldsymbol{\delta}$ that yields the stationary point of $f(\mathbf{x})$ can be obtained from Eq. (16.8) as

$$\boldsymbol{\delta} = -\mathbf{H}^{-1}\mathbf{g} \tag{16.9}$$

This equation will give the solution if and only if the following two conditions hold:

(i) The remainder $o(||\boldsymbol{\delta}||_2^2)$ in Eq. (16.7) can be neglected.

(ii) The Hessian is nonsingular.

If $f(\mathbf{x})$ is a *quadratic* function, its second partial derivatives are constants, i.e., $\mathbf{H}$ is a constant symmetric matrix, and its third and higher derivatives are zero. Therefore, condition (i) holds. If $f(\mathbf{x})$ has a stationary point and the sufficiency conditions for a minimum hold at the stationary point, then the Hessian matrix is *positive definite* and, therefore, *nonsingular*. Under these circumstances, given an arbitrary point $\mathbf{x} \in E^n$,[2] the *minimum point* can be obtained as $\breve{\mathbf{x}} = \mathbf{x} + \boldsymbol{\delta}$ by using Eq. (16.9).

If $f(\mathbf{x})$ is a general nonquadratic convex function that has a minimum at point $\breve{\mathbf{x}}$, then in the neighborhood $||\mathbf{x} - \breve{\mathbf{x}}||_2 < \epsilon$ the remainder in Eq. (16.7) becomes negligible and the second partial derivatives of $f(\mathbf{x})$ become approximately constant. As a result, in this domain function $f(\mathbf{x})$ behaves as if it were a quadratic function and conditions (i) and (ii) are again satisfied. Therefore, for any point $\mathbf{x}$ such that $||\mathbf{x} - \breve{\mathbf{x}}||_2 < \epsilon$, the use of Eq. (16.9) will yield an accurate estimate of the minimum point.

If a general function $f(\mathbf{x})$ is to be minimized and an arbitrary point $\mathbf{x} \in E^n$ is assumed, condition (i) and/or condition (ii) may be violated. If condition (i) is violated, then the use of Eq. (16.9) will not give the solution; if condition (ii) is violated, then Eq. (16.9) either has an infinite number of solutions or has no solutions at all. These problems can be overcome by using an iterative procedure in which the value of the function is progressively reduced by applying a series of corrections to $\mathbf{x}$ until a point in the neighborhood of the solution is obtained. When the remainder in Eq. (16.7) becomes negligible, an accurate estimate of the solution can be obtained by using Eq. (16.9). A suitable strategy to achieve this goal is based on the fundamental property that if $\mathbf{H}$ is positive definite, then $\mathbf{H}^{-1}$ is also positive definite. Furthermore, in such a case it can be shown through the use of the Taylor series that the direction pointed by the vector $-\mathbf{H}^{-1}\mathbf{g}$ of Eq. (16.9), which is known as the *Newton direction*, is a *descent direction* of $f(\mathbf{x})$. As a consequence, if at some initial point $\mathbf{x}$, $\mathbf{H}$ is positive definite, a reduction can be achieved in $f(\mathbf{x})$ by simply applying a correction of the form $\boldsymbol{\delta} = \alpha\mathbf{d}$ to $\mathbf{x}$, where α is a positive factor and $\mathbf{d} = -\mathbf{H}^{-1}\mathbf{g}$. On the other hand, if $\mathbf{H}$ is not positive definite, it can be forced to become positive definite by means of some algebraic manipulation (e.g., it can be changed to the unity matrix) and, as before, a reduction can be achieved in $f(\mathbf{x})$. In either case, the largest possible reduction in $f(\mathbf{x})$ with respect to the direction $\mathbf{d}$ can be achieved by choosing variable α such that $f(\mathbf{x} + \alpha\mathbf{d})$ is minimized. This can be done by using one of many available one-dimensional minimization algorithms (also known as *line searches*) [6–10]. Repeating these steps a number of times will yield a value of $\mathbf{x}$ in the neighborhood of the solution and eventually the solution itself. An algorithm based on these principles, known as the *Newton algorithm*, is as follows:

Algorithm 1: Basic Newton algorithm

1. Input $\mathbf{x}_0$ and ε. Set $k = 0$.
2. Compute the gradient $\mathbf{g}_k$ and Hessian $\mathbf{H}_k$. If $\mathbf{H}_k$ is not positive definite, force it to become positive definite.
3. Compute $\mathbf{H}_k^{-1}$ and $\mathbf{d}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k$.
4. Find α_k, the value of α that minimizes $f(\mathbf{x}_k + \alpha\,\mathbf{d}_k)$, using a line search.
5. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\delta}_k$, where $\boldsymbol{\delta}_k = \alpha_k\mathbf{d}_k$, and compute $f_{k+1} = f(\mathbf{x}_{k+1})$.
6. If $||\alpha_k\mathbf{d}_k||_2 < \varepsilon$, then output $\breve{\mathbf{x}} = \mathbf{x}_{k+1}$, $f(\breve{\mathbf{x}}) = f_{k+1}$, and stop.
   Otherwise, set $k = k + 1$ and repeat from step 2. ●

[2] *E^n represents the n-dimensional Euclidean space.*

The algorithm is terminated if the L_2 norm of $\alpha_k \mathbf{d}_k$, i.e., the magnitude of the change in **x**, is less than ε. The parameter ε is said to be the *termination tolerance* and is a small positive constant whose value is determined by the application under consideration.[3] In certain applications, a termination tolerance on the objective function itself, e.g., $|f_{k+1} - f_k| < \varepsilon$, may be preferable and sometimes termination tolerances may be imposed on the magnitudes of both the changes in **x** and the objective function.

So far, we have tacitly assumed that the optimization problem under consideration has a unique or *global* minimum. In practice, the problem may have more than one local minimum, sometimes a large number of minima, and on occasion a well-defined minimum may not even exist. We must, therefore, abandon the expectation that we shall always be able to obtain the best solution available. The best we can hope for is a solution that satisfies a number of the required specifications.

Example 16.1 (*a*) Show that the function

$$f(\mathbf{x}) = x_1^2 + 2x_1x_2 + 2x_2^2 + 2x_1 + x_2$$

has a minimum. (*b*) Find the minimum of the function using Algorithm 1 with $\mathbf{x}_0 = [0\ 0]^T$ as initial point.

■ **Solution**

(*a*) From basic calculus, the stationary points of a function are the points at which the gradient is equal to zero. If the Hessian at a specific stationary point is *positive definite*, *negative definite*, or *indefinite*, then the stationary point is a *minimum*, *maximum*, or *saddle point*; alternatively, if the Hessian is *positive* or *negative semidefinite*, then the stationary point can be either a *maximum* or a *minimum point*.

The partial derivatives of $f(\mathbf{x})$ are given by

$$\frac{\partial f}{\partial x_1} = 2x_1 + 2x_2 + 2 \qquad \text{and} \qquad \frac{\partial f}{\partial x_2} = 2x_1 + 4x_2 + 1$$

At a stationary point $\tilde{\mathbf{x}}$, the gradient **g** is zero; hence, we obtain $\tilde{\mathbf{x}} = [-1.5\ \ 0.5]^T$.

The Hessian can be deduced as

$$\mathbf{H} = \begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix}$$

Since the *principal minor determinants* of **H** are positive, the Hessian is positive definite (see Sec. 5.3.6), and so $\tilde{\mathbf{x}}$ is a minimum point.

(*b*) The gradient at $\mathbf{x}_0^T = [0\ 0]$ is $\mathbf{g}_0 = [2\ 1]^T$. The inverse of $\mathbf{H}_0$ is given by

$$\mathbf{H}_0^{-1} = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 0.5 \end{bmatrix}$$

[3] *Parameters ε, ε_1, and ε_2 represent termination tolerances throughout the chapter.*

and hence the Newton direction can be obtained from step 3 of Algorithm 1 as $\mathbf{d}_0 = -\mathbf{H}_0^{-1}\mathbf{g}_0 = [-1.5\ 0.5]^T$. The function under consideration is quadratic and the solution can be obtained with $\alpha_0 = 1$. From step 5, $\mathbf{x}_1 = \breve{\mathbf{x}} = [-1.5\ 0.5]^T$ and $f(\breve{\mathbf{x}}) = f_1 = -1.25$. Note that Algorithm 1 will need two iterations to stop since the termination test in step 6 will not be satisfied until the second iteration.

16.4 QUASI-NEWTON ALGORITHMS

The Newton algorithm described in the preceding section has three major disadvantages. First, both the first and second partial derivatives of $f(\mathbf{x})$ must be computed in each iteration in order to construct the gradient and Hessian, respectively. Second, in each iteration the Hessian must be checked for positive definiteness and, if it is found to be nonpositive definite, it must be forced to become positive definite. Third, matrix inversion is required in each iteration. By contrast, in quasi-Newton algorithms only the first derivatives need to be computed, and it is unnecessary to manipulate or invert the Hessian. Consequently, for general problems other than convex quadratic problems, quasi-Newton algorithms are much more efficient and are preferred.

Quasi-Newton algorithms, like the Newton algorithm, are developed for the convex quadratic problem and are then extended to the general problem. The fundamental principle in these algorithms is that the direction of search is based on an $n \times n$ matrix $\mathbf{S}$ that serves the same purpose as the inverse Hessian in the Newton algorithm. This matrix is constructed using available data and is contrived to be an approximation of $\mathbf{H}^{-1}$. Furthermore, as the number of iterations is increased, $\mathbf{S}$ becomes progressively a more and more accurate representation of $\mathbf{H}^{-1}$. For convex quadratic objective functions, $\mathbf{S}$ becomes identical to $\mathbf{H}^{-1}$ in $n + 1$ iterations where n is the number of variables.

16.4.1 Basic Quasi-Newton Algorithm

Let the gradients of $f(\mathbf{x})$ at points $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ be $\mathbf{g}_k$ and $\mathbf{g}_{k+1}$, respectively. If

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\delta}_k$$

then the Taylor series gives the elements of $\mathbf{g}_{k+1}$ as

$$g_{(k+1)m} = g_{km} + \sum_{i=1}^{n} \frac{\partial g_{km}}{\partial x_{k\,i}} \delta_{k\,i} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 g_{km}}{\partial x_{k\,i} \partial x_{kj}} \delta_{k\,i} \delta_{kj} + o\big(||\boldsymbol{\delta}||_2^2\big)$$

for $m = 1, 2, \ldots, n$. Now if $f(\mathbf{x})$ is quadratic, the second and higher derivatives of $f(\mathbf{x})$ are constant and zero, respectively, and as a result the second and higher derivatives of g_{km} are zero. Thus

$$g_{(k+1)m} = g_{km} + \sum_{i=1}^{n} \frac{\partial g_{km}}{\partial x_{k\,i}} \delta_{k\,i}$$

and since

$$g_{km} = \frac{\partial f_k}{\partial x_{km}}$$

we have

$$g_{(k+1)m} = g_{km} + \sum_{i=1}^{n} \frac{\partial^2 f_k}{\partial x_{k\,i} \partial x_{km}} \delta_{k\,i}$$

for $m = 1, 2, \ldots, n$. Therefore, $\mathbf{g}_{k+1}$ is given by

$$\mathbf{g}_{k+1} = \mathbf{g}_k + \mathbf{H}\boldsymbol{\delta}_k$$

where $\mathbf{H}$ is the Hessian of $f(\mathbf{x})$. Alternatively, we can write

$$\boldsymbol{\gamma}_k = \mathbf{H}\boldsymbol{\delta}_k \tag{16.10}$$

where

$$\boldsymbol{\delta}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$$

and

$$\boldsymbol{\gamma}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$$

The above analysis has shown that, if the gradient of $f(\mathbf{x})$ is known at two points $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$, a relation can be deduced that provides a certain amount of information about $\mathbf{H}$, namely, Eq. (16.10). Since $\mathbf{H}$ is a real symmetric matrix with $n \times (n + 1)/2$ unknowns and Eq. (16.10) provides only n equations, $\mathbf{H}$ cannot be determined uniquely through the use of Eq. (16.10). This problem can be overcome by evaluating the gradient sequentially at $n + 1$ points, say at $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n$, such that the changes in $\mathbf{x}$, namely,

$$\begin{aligned}
\boldsymbol{\delta}_0 &= \mathbf{x}_1 - \mathbf{x}_0 \\
\boldsymbol{\delta}_1 &= \mathbf{x}_2 - \mathbf{x}_1 \\
\vdots\ \ & \qquad \vdots \\
\boldsymbol{\delta}_{n-1} &= \mathbf{x}_n - \mathbf{x}_{n-1}
\end{aligned}$$

form a set of linearly independent vectors. Under these circumstances, Eq. (16.10) yields

$$\left[\boldsymbol{\gamma}_0\ \boldsymbol{\gamma}_1\ \ldots\ \boldsymbol{\gamma}_{n-1}\right] = \mathbf{H}\left[\boldsymbol{\delta}_0\ \boldsymbol{\delta}_1\ \cdots\ \boldsymbol{\delta}_{n-1}\right]$$

Therefore, $\mathbf{H}$ can be uniquely determined as

$$\mathbf{H} = \left[\boldsymbol{\gamma}_0\ \boldsymbol{\gamma}_1\ \cdots\ \boldsymbol{\gamma}_{n-1}\right]\left[\boldsymbol{\delta}_0\ \boldsymbol{\delta}_1\ \cdots\ \boldsymbol{\delta}_{n-1}\right]^{-1} \tag{16.11}$$

The above principles lead to the following algorithm:

Algorithm 2: Alternative Newton algorithm

```
1. Input $\mathbf{x}_{00}$ and $\varepsilon$. Input a set of $n$ linearly independent vectors $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{n-1}$. Set $k = 0$.
2. Compute $\mathbf{g}_{k0}$.
3. For $i = 0$ to $n - 1$ do:
   a. Set $\mathbf{x}_{k(i+1)} = \mathbf{x}_{k\,i} + \boldsymbol{\delta}_i$.
   b. Compute $\mathbf{g}_{k(i+1)}$.
   c. Set $\boldsymbol{\gamma}_{k\,i} = \mathbf{g}_{k(i+1)} - \mathbf{g}_{k\,i}$.
4. Compute $\mathbf{H}_k$ using Eq. (16.11). If $\mathbf{H}_k$ is not positive definite, force it to become positive definite.
5. Determine $\mathbf{S}_k = \mathbf{H}_k^{-1}$.
```

6. Set $\mathbf{d}_k = -\mathbf{S}_k\mathbf{g}_{k0}$ and find α_k, the value of α that minimizes $f(\mathbf{x}_{k0} + \alpha\,\mathbf{d}_k)$, using a line search.

7. Set $\mathbf{x}_{(k+1)0} = \mathbf{x}_{k0} + \alpha_k\mathbf{d}_k$ and compute $f_{(k+1)0} = f(\mathbf{x}_{(k+1)0})$.

8. If $\|\alpha_k\mathbf{d}_k\|_2 < \varepsilon$, then output $\breve{\mathbf{x}} = \mathbf{x}_{(k+1)0}$, $f(\breve{\mathbf{x}}) = f_{(k+1)0}$, and stop.
Otherwise, set $k = k + 1$ and repeat from step 2. ●

Algorithm 2 is essentially an alternative implementation of the Newton method in which the generation of $\mathbf{H}^{-1}$ is accomplished using computed data instead of the second derivatives. However, as in Algorithm 1, for the general nonquadratic problem it is necessary to check, manipulate, and invert the Hessian in every iteration. In addition, we now need to provide a set of linearly independent vectors to the algorithm, namely, $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{n-1}$. In other words, though of considerable conceptual value, the algorithm is of little practical usefulness.

Further progress toward the development of the quasi-Newton method can be made by generating the matrix $\mathbf{H}^{-1}$ from computed data using a set of linearly independent vectors $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{n-1}$ that are themselves generated from available data. This objective can be accomplished by generating the vectors

$$\boldsymbol{\delta}_k = -\mathbf{S}_k\mathbf{g}_k \tag{16.12}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\delta}_k \tag{16.13}$$

and

$$\boldsymbol{\gamma}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$$

and then making an additive correction to $\mathbf{S}_k$ of the form

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \mathbf{C}_k \tag{16.14}$$

for $k = 0, 1, \ldots, n-1$. If a correction matrix $\mathbf{C}_k$ can be found such that the conditions

$$\mathbf{S}_{k+1}\boldsymbol{\gamma}_i = \boldsymbol{\delta}_i \qquad \text{for } 0 \le i \le k \tag{16.15}$$

are satisfied and the vectors $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{n-1}$ and $\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{n-1}$ generated by this process are linearly independent, then for the case $k = n-1$ we can write

$$\mathbf{S}_n\left[\boldsymbol{\gamma}_0\ \boldsymbol{\gamma}_1\ \ldots\ \boldsymbol{\gamma}_{n-1}\right] = \left[\boldsymbol{\delta}_0\ \boldsymbol{\delta}_1\ \ldots\ \boldsymbol{\delta}_{n-1}\right]$$

or

$$\mathbf{S}_n = \left[\boldsymbol{\delta}_0\ \boldsymbol{\delta}_1\ \ldots\ \boldsymbol{\delta}_{n-1}\right]\left[\boldsymbol{\gamma}_0\ \boldsymbol{\gamma}_1\ \ldots\ \boldsymbol{\gamma}_{n-1}\right]^{-1} \tag{16.16}$$

Now from Eqs. (16.11) and (16.16), we have

$$\mathbf{S}_n = \mathbf{H}^{-1} \tag{16.17}$$

and if $k = n$, Eqs. (16.12) and (16.17) yield the Newton direction

$$\boldsymbol{\delta}_n = -\mathbf{H}^{-1}\mathbf{g}_n \tag{16.18}$$

Therefore, subject to conditions (i) and (ii) stated earlier, the solution of a convex quadratic problem can be obtained from Eqs. (16.13) and (16.18) as

$$\breve{\mathbf{x}} = \mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{H}^{-1}\mathbf{g}_n$$

The above principles lead to the basic quasi-Newton algorithm which is as follows:

Algorithm 3: Basic quasi-Newton algorithm

1. Input $\mathbf{x}_0$ and ε. Set $\mathbf{S}_0 = \mathbf{I}_n$ and $k = 0$. Compute $\mathbf{g}_0$.
2. Set $\mathbf{d}_k = -\mathbf{S}_k\mathbf{g}_k$ and find α_k, the value of α that minimizes $f(\mathbf{x}_k + \alpha\,\mathbf{d}_k)$, using a line search.
3. Set $\boldsymbol{\delta}_k = \alpha_k\mathbf{d}_k$ and $\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\delta}_k$, and compute $f_{k+1} = f(\mathbf{x}_{k+1})$.
4. If $\|\boldsymbol{\delta}_k\|_2 < \varepsilon$, then output $\breve{\mathbf{x}} = \mathbf{x}_{k+1}$, $f(\breve{\mathbf{x}}) = f_{k+1}$ and stop.
5. Compute $\mathbf{g}_{k+1}$ and set $\boldsymbol{\gamma}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$.
6. Compute $\mathbf{S}_{k+1} = \mathbf{S}_k + \mathbf{C}_k$.
7. Check $\mathbf{S}_{k+1}$ for positive definiteness and if it is found to be nonpositive definite force it to become positive definite.
8. Set $k = k + 1$ and go to step 2. ●

In step 2, the vector $-\mathbf{S}_k\mathbf{g}_k$ is denoted as $\mathbf{d}_k$, instead of $\boldsymbol{\delta}_k$ as in Eq. (16.12), and $f(\mathbf{x}_k + \alpha\mathbf{d}_k)$ is minimized with respect to α. The purpose of this modification is to make the algorithm applicable to the general nonquadratic problem where $-\mathbf{S}_k\mathbf{g}_k$ may not be the Newton direction. Matrix $\mathbf{S}_k$ is required to be positive definite for each k to ensure that vector $\mathbf{d}_k$ is a descent direction in each iteration. To obtain a descent direction in the first iteration, $\mathbf{S}_0$ is assumed to be the $n \times n$ unity matrix in step 1. Vector $\boldsymbol{\gamma}_k$ in step 5 is required for the computation of correction matrix $\mathbf{C}_k$ in step 6, as will be demonstrated in Sec. 16.4.2 below.

Algorithm 3 eliminates the need to input a set of linearly independent vectors $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{n-1}$ and, in addition, the inversion of $\mathbf{H}_k$ is replaced by an additive correction to $\mathbf{S}_k$. However, matrices $\mathbf{S}_1, \mathbf{S}_2, \ldots$ need to be checked for positive definiteness and may need to be manipulated. This can be easily done in practice by diagonalizing $\mathbf{S}_{k+1}$ and then replacing any nonpositive diagonal elements by corresponding positive ones. However, this would increase the computational load quite significantly.

16.4.2 Updating Formulas for Matrix $\mathbf{S}_{k+1}$

The updating formula for matrix $\mathbf{S}_{k+1}$ of Eq. (16.14) must satisfy strict requirements to be useful in Algorithm 3. As was stated earlier, for a convex quadratic problem, Eq. (16.15) must be satisfied and the vectors $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{n-1}$ and $\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{n-1}$ must be linearly independent. The derivation and properties of updating formulas of this type have received considerable attention during the past 30 years or so, and several distinct formulas have appeared in the literature. Early in the development of the subject, the so-called *rank-one* formula was proposed, in which the correction matrix $\mathbf{C}_k$ is of rank one. This has largely been replaced in recent years by *rank-two* formulas, like the Davidon-Fletcher-Powell (DFP) and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formulas [6–10]. A very important property of these two formulas is that a positive definite matrix $\mathbf{S}_k$ yields a positive definite $\mathbf{S}_{k+1}$ not only for convex quadratic problems but also for the general nonquadratic problem, provided that the line search in step 2 of the algorithm is exact (see Fletcher [6] for proof). This property also holds in the case where an *inexact* line search is used in step 2, except that a scalar quantity inherent in the computation of $\mathbf{C}_k$ must be forced to remain positive. The usefulness of this property in Algorithm 3 is obvious: the checking and manipulation of $\mathbf{S}_{k+1}$ in step 7 of the algorithm become unnecessary, and hence a considerable amount of computation can be avoided.

The DFP and BFGS updating formulas are given by

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k} - \frac{\mathbf{S}_k \boldsymbol{\gamma}_k \boldsymbol{\gamma}_k^T \mathbf{S}_k}{\boldsymbol{\gamma}_k^T \mathbf{S}_k \boldsymbol{\gamma}_k} \tag{16.19}$$

and

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \left(1 + \frac{\boldsymbol{\gamma}_k^T \mathbf{S}_k \boldsymbol{\gamma}_k}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k}\right) \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k} - \frac{\left(\boldsymbol{\delta}_k \boldsymbol{\gamma}_k^T \mathbf{S}_k + \mathbf{S}_k \boldsymbol{\gamma}_k \boldsymbol{\delta}_k^T\right)}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k} \tag{16.20}$$

respectively. A condition that guarantees the positive definiteness of $\mathbf{S}_{k+1}$ in both formulas is

$$\boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k = \boldsymbol{\delta}_k^T \mathbf{g}_{k+1} - \boldsymbol{\delta}_k^T \mathbf{g}_k > 0 \tag{16.21}$$

This will be put to good use in Algorithm 5.

16.4.3 Inexact Line Searches

In optimization algorithms in general, the bulk of the computational effort is spent executing line searches. Consequently, the amount of computation required to solve a problem tends to depend critically on the efficiency and precision of the line search used. If a high-precision line search is mandatory in a certain algorithm, then the algorithm can spend a considerable amount of computational effort minimizing the objective function with respect to scalar α. For this reason, low-precision or inexact line searches are usually preferable, provided of course that their use does not affect the convergence properties of the algorithm. Quasi-Newton algorithms have been found to be quite tolerant to line-search imprecision. As a result, inexact line searches are almost always used in these algorithms. An important line search of this type will now be examined.

Let

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\, \mathbf{d}_k$$

where $\mathbf{d}_k$ is a given descent direction vector and α is an independent variable, and assume that $f(\mathbf{x}_{k+1})$ is a *unimodal function*[4] of α, with a minimum at some point $\alpha = \breve{\alpha}$ where $\breve{\alpha} > 0$, as depicted in Fig. 16.2*a*. The linear approximation of the Taylor series for $f(\mathbf{x}_{k+1})$ is of the form

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) + \alpha\, \mathbf{g}_k^T \mathbf{d}_k \tag{16.22}$$

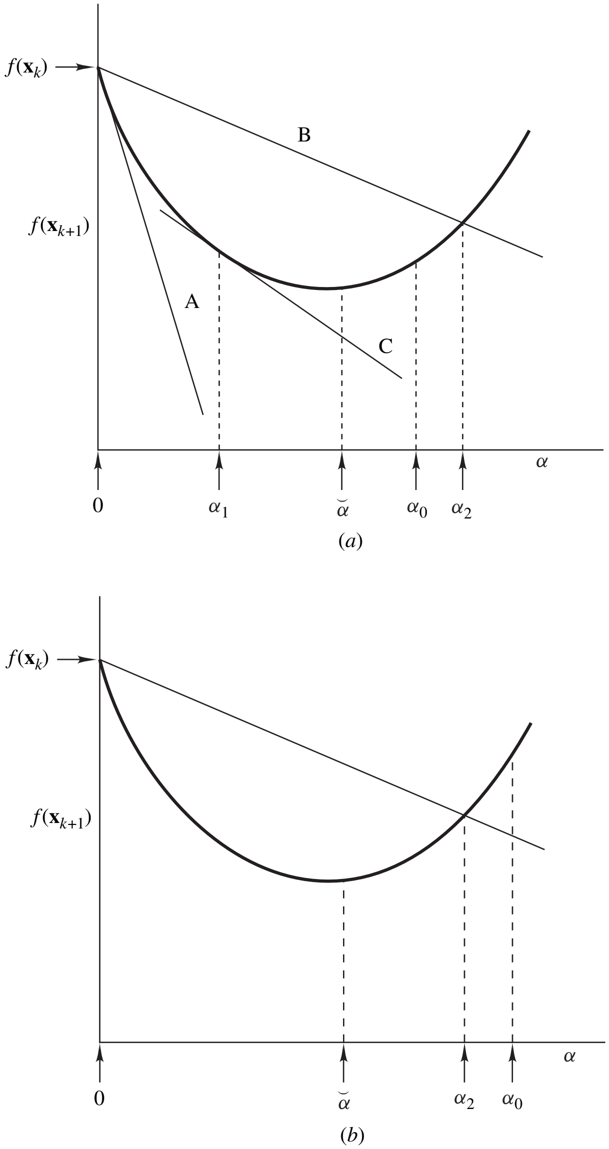
where

$$\mathbf{g}_k^T \mathbf{d}_k = \left.\frac{df(\mathbf{x}_k + \alpha\, \mathbf{d}_k)}{d\alpha}\right|_{\alpha=0}$$

is the slope at the origin of $f(\mathbf{x}_k + \alpha\, \mathbf{d}_k)$ as a function of α. Eq. (16.22) represents line A depicted in Fig. 16.2*a*. Similarly, the equation

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) + \rho\, \alpha\, \mathbf{g}_k^T \mathbf{d}_k \tag{16.23}$$

[4] *A unimodal function is one that has only one minimum.*



(*a*)

(*b*)

Figure 16.2 Inexact line search: (*a*) Case where the conditions in Eqs. (16.25) and (16.26) are both satisfied, (*b*) case where the condition in Eq. (16.25) is violated.

where $0 \leq \rho \leq 0.5$ represents a line (line B in Fig. 16.2*a*) whose slope ranges from 0 to $0.5\mathbf{g}_k^T\mathbf{d}_k$, depending on the value of ρ. Let us assume that this line intersects the curve in Fig. 16.2*a* at point $\alpha = \alpha_2$. On the other hand, the equation

$$\mathbf{g}_{k+1}^T\mathbf{d}_k = \sigma\, \mathbf{g}_k^T\mathbf{d}_k \tag{16.24}$$

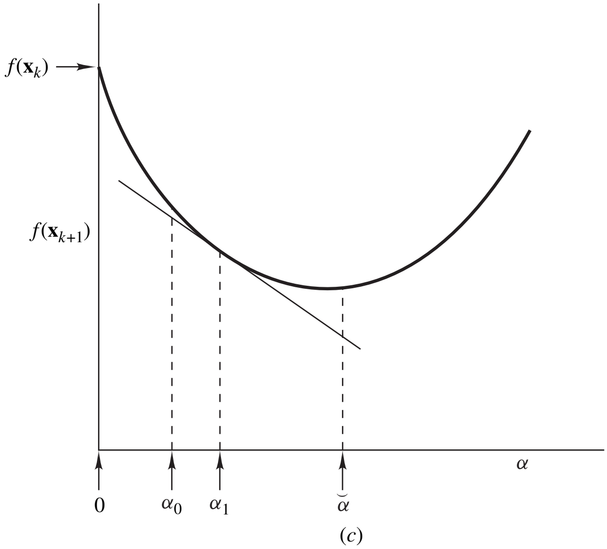


Figure 16.2 Cont'd Inexact line search: (*c*) Case where the condition in Eq. (16.26) is violated.

where $0 < \sigma < 1$, and $\sigma \geq \rho$ relates the derivative of $f(\mathbf{x}_{k+1})$ at some point $\alpha = \alpha_1$ to the derivative of the function at $\alpha = 0$ and represents line C in Fig. 16.2*a*. Since $0 < \sigma < 1$, we have $0 < \alpha_1 < \breve{\alpha}$.

Equations (16.23) and (16.24) define an interval $[\alpha_1, \alpha_2]$ that brackets the minimum point. Consequently, the two equations can be used as a termination criterion in a line search, much like the use of a termination tolerance on **x** or $f(\mathbf{x})$ in Algorithms 1 to 3. This possibility will now be examined.

Let us assume that a mechanism is available by which an estimate of $\breve{\alpha}$, say α_0, can be generated. If the actual value of $f(\mathbf{x}_{k+1})$ at $\alpha = \alpha_0$ is less than the value predicted by the linear approximation of Eq. (16.23), that is,

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \rho\, \alpha_0 \mathbf{g}_k^T \mathbf{d}_k \tag{16.25}$$

then $\alpha_0 \leq \alpha_2$. On the other hand, if the actual slope at $\alpha = \alpha_0$ is less negative (more positive) than the slope of the line in Eq. (16.24), that is,

$$\mathbf{g}_{k+1}^T \mathbf{d}_k \geq \sigma\, \mathbf{g}_k^T \mathbf{d}_k \tag{16.26}$$

then $\alpha_1 \leq \alpha_0$. Under these circumstances, we have $\alpha_1 \leq \alpha_0 \leq \alpha_2$, as depicted in Fig. 16.2*a*, and a certain reduction in $f(\mathbf{x}_{k+1})$ is achieved, which can be considered to be acceptable. In other words, if both Eqs. (16.25) and (16.26) are satisfied, then α_0 can be accepted as a reasonable approximation of $\breve{\alpha}$.

If either of the conditions in Eqs. (16.25) and (16.26) is violated, then α_0 is outside the interval $[\alpha_1, \alpha_2]$ and the reduction in $f(\mathbf{x}_{k+1})$ can be considered to be unacceptable. If the condition in Eq. (16.25) is violated, then $\alpha_0 > \alpha_2$, as depicted in Fig. 16.2*b*; since $0 < \breve{\alpha} < \alpha_0$, a better estimate for $\breve{\alpha}$ (say $\breve{\alpha}_0$) can be deduced by using some interpolation formula. If the condition in Eq. (16.26) is violated, then $0 < \alpha_0 < \alpha_1$, as depicted in Fig. 16.2*c*; in this case, a better estimate $\breve{\alpha}_0$ can be deduced by using some extrapolation formula. With a new estimate for $\breve{\alpha}$ available, the conditions in Eqs. (16.25) and (16.26) can be checked again and, if either of the two is not satisfied, the process is repeated. When an estimate of $\breve{\alpha}$ is found that satisfies both Eqs. (16.25) and (16.26), the search is terminated. The precision of such a line search can be controlled by choosing the values of ρ and σ since these parameters control the length of interval $[\alpha_1, \alpha_2]$.

Interpolation and *extrapolation* formulas that can be used in the above approach can be readily deduced by assuming a quadratic representation for $f(\mathbf{x}_k + \alpha\,\mathbf{d}_k)$. If the value of this function and its derivative with respect to α are known at two points, say, at $\alpha = \alpha_L$ and $\alpha = \alpha_0$ where $\alpha_L < \alpha_0$, then for $\alpha_0 > \alpha_2$ we can show that

$$\breve{\alpha}_0 = \alpha_L + \frac{(\alpha_0 - \alpha_L)^2 f'_L}{2[f_L - f_0 + (\alpha_0 - \alpha_L) f'_L]} \tag{16.27}$$

and for $\alpha_0 < \alpha_1$

$$\breve{\alpha}_0 = \alpha_0 + \frac{(\alpha_0 - \alpha_L) f'_0}{(f'_L - f'_0)} \tag{16.28}$$

where

$$\begin{aligned} f_L &= f(\mathbf{x}_k + \alpha_L \mathbf{d}_k) \\ f'_L &= f'(\mathbf{x}_k + \alpha_L \mathbf{d}_k) = \mathbf{g}(\mathbf{x}_k + \alpha_L \mathbf{d}_k)^T \mathbf{d}_k \\ f_0 &= f(\mathbf{x}_k + \alpha_0 \mathbf{d}_k) \\ f'_0 &= f'(\mathbf{x}_k + \alpha_0 \mathbf{d}_k) = \mathbf{g}(\mathbf{x}_k + \alpha_0 \mathbf{d}_k)^T \mathbf{d}_k \end{aligned}$$

An inexact line search due to Fletcher [6] based on the above principles is as follows:

Algorithm 4: Fletcher inexact line search

```
1. Input $\mathbf{x}_k$ and $\mathbf{d}_k$. Initialize algorithm parameters $\rho, \sigma, \tau$, and $\chi$. Set $\alpha_L = 0$ and $\alpha_U = 10^{99}$. Compute $\mathbf{g}_k$.
2. Compute $f_L = f(\mathbf{x}_k + \alpha_L \mathbf{d}_k)$ and $f'_L = \mathbf{g}(\mathbf{x}_k + \alpha_L \mathbf{d}_k)^T \mathbf{d}_k$.
3. Initialize $\alpha_0$, say $\alpha_0 = 1$ .
4. Compute $f_0 = f(\mathbf{x}_k + \alpha_0 \mathbf{d}_k)$.
5. (Interpolation)
   If $f_0 > f_L + \rho(\alpha_0 - \alpha_L) f'_L$, then do:
   a. If $\alpha_0 < \alpha_U$, then set $\alpha_U = \alpha_0$.
   b. Compute $\breve{\alpha}_0$ using Eq. (16.27).
   c. Compute $\breve{\alpha}_{0L} = \alpha_L + \tau(\alpha_U - \alpha_L)$; if $\breve{\alpha}_0 < \breve{\alpha}_{0L}$, then set $\breve{\alpha}_0 = \breve{\alpha}_{0L}$.
   d. Compute $\breve{\alpha}_{0U} = \alpha_U - \tau(\alpha_U - \alpha_L)$; if $\breve{\alpha}_0 > \breve{\alpha}_{0U}$, then set $\breve{\alpha}_0 = \breve{\alpha}_{0U}$.
   e. Set $\alpha_0 = \breve{\alpha}_0$ and go to step 4.
6. Compute $f'_0 = \mathbf{g}(\mathbf{x}_k + \alpha_0 \mathbf{d}_k)^T \mathbf{d}_k$.
7. (Extrapolation)
   If $f'_0 < \sigma f'_L$, then do:
   a. Compute $\Delta\alpha_0 = (\alpha_0 - \alpha_L) f'_0/(f'_L - f'_0)$ (see Eq. (16.28)).
   b. If $\Delta\alpha_0 < \tau(\alpha_0 - \alpha_L)$, then set $\Delta\alpha_0 = \tau(\alpha_0 - \alpha_L)$.
   c. If $\Delta\alpha_0 > \chi(\alpha_0 - \alpha_L)$, then set $\Delta\alpha_0 = \chi(\alpha_0 - \alpha_L)$.
   d. Compute $\breve{\alpha}_0 = \alpha_0 + \Delta\alpha_0$.
   e. Set $\alpha_L = \alpha_0$, $\alpha_0 = \breve{\alpha}_0$, $f_L = f_0$, $f'_L = f'_0$ and go to step 4.
8. Output $\alpha_0$ and $f_0$, and stop. ●
```

Assuming that $\mathbf{d}_k$ is a descent direction of $f(\mathbf{x})$ at point $\mathbf{x}_k$, the algorithm will carry out interpolations and/or extrapolations as necessary, which will progressively reduce the value of $f(\mathbf{x}_k + \alpha\,\mathbf{d}_k)$. When the conditions in Eqs. (16.25) and (16.26) are simultaneously satisfied, the algorithm terminates. The algorithm maintains a running bracket $[\alpha_L, \alpha_U]$ on the minimum point such that $\alpha_L \le \breve{\alpha}_0 \le \alpha_U$; if the interpolation formula yields a value of $\breve{\alpha}_0$ outside this interval or very close to the lower or upper limit, a more reasonable value is assigned to $\breve{\alpha}_0$ in step 5*c* or 5*d*. Similarly, if the value of $\Delta\alpha_0$ predicted in step 7*a* is negative, very small or very large, a more reasonable value is assigned to $\Delta\alpha_0$ in step 7*b* or 7*c*. The precision of the line search depends on the values of ρ and σ. Small values like $\rho = \sigma = 0.1$ yield a high-precision line search, whereas the values $\rho = 0.15$ and $\sigma = 0.9$ yield a somewhat imprecise one. Suitable values for τ and χ are 0.1 and 9, respectively. Further details about this line search can be found in the first edition of Fletcher [6]. A closely related inexact line search proposed by Al-Baali and Fletcher can be found in Ref. [11] (see also second edition of Fletcher [6]).

16.4.4 Practical Quasi-Newton Algorithm

A practical quasi-Newton algorithm that eliminates the problems associated with Algorithms 1 to 3 is detailed below. This is based on Algorithm 3 and uses a slightly modified version of Algorithm 4 as inexact line search. The algorithm is flexible, efficient, and very reliable, and is readily applicable for the design of digital filters and equalizers, as will be shown in Secs. 16.7 and 16.8.

Algorithm 5: Practical quasi-Newton algorithm

```
1. (Initialize algorithm)
   a. Input $\mathbf{x}_0$ and $\varepsilon_1$.
   b. Set $k = m = 0$.
   c. Set $\rho = 0.1$, $\sigma = 0.7$, $\tau = 0.1$, $\chi = 0.75$, $\widehat{M} = 600$, and $\varepsilon_2 = 10^{-10}$.
   d. Set $\mathbf{S}_0 = \mathbf{I}_n$.
   e. Compute $f_0$ and $\mathbf{g}_0$, and set $m = m + 2$. Set $f_{00} = f_0$ and $\Delta f_0 = f_0$.
2. (Initialize line search)
   a. Set $\mathbf{d}_k = -\mathbf{S}_k\mathbf{g}_k$.
   b. Set $\alpha_L = 0$ and $\alpha_U = 10^{99}$.
   c. Set $f_L = f_0$ and compute $f'_L = \mathbf{g}(\mathbf{x}_k + \alpha_L\mathbf{d}_k)^T\mathbf{d}_k$.
   d. (Estimate $\alpha_0$)
      If $|f'_L| > \varepsilon_2$, then compute $\alpha_0 = -2\Delta f_0/f'_L$; otherwise, set $\alpha_0 = 1$.
      If $\alpha_0 \le 0$ or $\alpha_0 > 1$, then set $\alpha_0 = 1$.
3. Set $\boldsymbol{\delta}_k = \alpha_0\mathbf{d}_k$ and compute $f_0 = f(\mathbf{x}_k + \boldsymbol{\delta}_k)$.
   Set $m = m + 1$.
4. (Interpolation)
   If $f_0 > f_L + \rho(\alpha_0 - \alpha_L)f'_L$ and $|(f_L - f_0)| > \varepsilon_2$ and $m < \widehat{M}$, then do:
   a. If $\alpha_0 < \alpha_U$, then set $\alpha_U = \alpha_0$.
   b. Compute $\breve{\alpha}_0$ using Eq. (16.27).
   c. Compute $\breve{\alpha}_{0L} = \alpha_L + \tau(\alpha_U - \alpha_L)$; if $\breve{\alpha}_0 < \breve{\alpha}_{0L}$, then set $\breve{\alpha}_0 = \breve{\alpha}_{0L}$.
   d. Compute $\breve{\alpha}_{0U} = \alpha_U - \tau(\alpha_U - \alpha_L)$; if $\breve{\alpha}_0 > \breve{\alpha}_{0U}$, then set $\breve{\alpha}_0 = \breve{\alpha}_{0U}$.
   e. Set $\alpha_0 = \breve{\alpha}_0$ and go to step 3.
5. Compute $f'_0 = \mathbf{g}(\mathbf{x}_k + \alpha_0\mathbf{d}_k)^T\mathbf{d}_k$ and set $m = m + 1$.
6. (Extrapolation)
   If $f'_0 < \sigma f'_L$ and $|(f_L - f_0)| > \varepsilon_2$ and $m < \widehat{M}$, then do:
   a. Compute $\Delta\alpha_0 = (\alpha_0 - \alpha_L)f'_0/(f'_L - f'_0)$ (see Eq. (16.28)).
```

b. If $\Delta\alpha_0 \leq 0$, then set $\breve{\alpha}_0 = 2\alpha_0$; otherwise, set $\breve{\alpha}_0 = \alpha_0 + \Delta\alpha_0$.
c. Compute $\breve{\alpha}_{0U} = \alpha_0 + \chi(\alpha_U - \alpha_0)$; if $\breve{\alpha}_0 > \breve{\alpha}_{0U}$, then set $\breve{\alpha}_0 = \breve{\alpha}_{0U}$.
d. Set $\alpha_L = \alpha_0, \alpha_0 = \breve{\alpha}_0, f_L = f_0, f'_L = f'_0$ and go to step 3.

7. (Check termination criteria and output results)
a. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\delta}_k$.
b. Set $\Delta f_0 = f_{00} - f_0$.
c. If ($\|\boldsymbol{\delta}_k\|_2 < \varepsilon_1$ and $|\Delta f_0| < \varepsilon_1$) or $m \geq \widehat{M}$, then output $\breve{\mathbf{x}} = \mathbf{x}_{k+1}$, $f(\breve{\mathbf{x}}) = f_{k+1}$, and stop.
d. Set $f_{00} = f_0$.

8. (Prepare for next iteration)
a. Compute $\mathbf{g}_{k+1}$ and set $\boldsymbol{\gamma}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$.
b. Compute $D = \boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k$; if $D \leq 0$, then set $\mathbf{S}_{k+1} = \mathbf{I}_n$; otherwise, compute $\mathbf{S}_{k+1}$using Eq. (16.19) or Eq. (16.20).
c. Set $k = k + 1$ and go to step 2. ●

Index m maintains a count of the number of function evaluations and is increased by one for each evaluation of f_0 or f'_0 in step 3 or 5, and $\widehat{M}$ is the maximum number of function evaluations allowed. When m becomes greater than $\widehat{M}$, the algorithm stops.

The estimate of α_0 in step 2*d* can be obtained by assuming that the function $f(\mathbf{x}_k + \alpha\,\mathbf{d}_k)$ can be represented by a quadratic polynomial of α and that the reduction achieved in $f(\mathbf{x}_k + \alpha\,\mathbf{d}_k)$ by changing α from 0 to α_0 is equal to Δf_0, the total reduction achieved in the previous iteration (see Prob. 16.11). This estimate can sometimes be quite inaccurate and may in certain circumstances become negative due to numerical ill-conditioning. For these reasons, if the estimate is equal to or less than zero or greater than unity, it is replaced by unity.

The quadratic extrapolation in step 6 of the algorithm may sometimes predict a maximum point at some negative value of α instead of a minimum point at some positive value of α (see Prob. 16.12). If such a case is identified in step 6*b*, the value of $2\alpha_0$ is assigned to $\breve{\alpha}_0$ to ensure that α is changed in the direction of descent. If α_U is fixed by the interpolation, the minimum point cannot exceed this value; and, if extrapolation results in an unreasonably large value of $\breve{\alpha}_0$, it is replaced by the value $\breve{\alpha}_{0U}$ computed in step 6*c*.

While a positive definite matrix $\mathbf{S}_k$ will ensure that $\mathbf{d}_k$ is a direction of descent of function $f(\mathbf{x})$ at point $\mathbf{x}_k$, in some rare occasions the function $f(\mathbf{x}_k + \alpha\,\mathbf{d}_k)$ may not have a well-defined minimum point. On the other hand, when the value of the function is very small, numerical ill-conditioning may arise occasionally due to roundoff errors. To avoid these problems, interpolation or extrapolation is carried out only if the expected reduction in the function $f(\mathbf{x}_k + \alpha\,\mathbf{d}_k)$ is larger than ε_2 and an upper limit in the number of function evaluations has not been exceeded.

If the DFP or BFGS updating formula is used in step 8*b* and the condition in Eq. (16.21) is satisfied, then a positive definite matrix $\mathbf{S}_k$ will result in a positive definite $\mathbf{S}_{k+1}$, as was stated earlier. We will now demonstrate that if the Fletcher inexact line search is used and the search is not terminated until the inequality in Eq. (16.26) is satisfied, then Eq. (16.21) is, indeed, satisfied. When the search is terminated in the kth iteration, we have $\alpha_0 \equiv \alpha_k$ and from step 3 of the algorithm $\boldsymbol{\delta}_k = \alpha_k \mathbf{d}_k$. Now from Eqs. (16.21) and (16.26), we obtain

$$\begin{aligned}\boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k &= \boldsymbol{\delta}_k^T \mathbf{g}_{k+1} - \boldsymbol{\delta}_k^T \mathbf{g}_k \\ &= \alpha_k \left(\mathbf{g}_{k+1}^T \mathbf{d}_k - \mathbf{g}_k^T \mathbf{d}_k\right) \\ &\geq \alpha_k(\sigma - 1)\mathbf{g}_k^T \mathbf{d}_k\end{aligned}$$

If $\mathbf{d}_k$ is a descent direction, then $\mathbf{g}_k^T\mathbf{d}_k < 0$ and $\alpha_k > 0$. Since $\sigma < 1$, we conclude that

$$\boldsymbol{\delta}_k^T\boldsymbol{\gamma}_k > 0$$

Under these circumstances, the positive definiteness of $\mathbf{S}_k$ is assured. In exceptional circumstances, the inexact line search in Algorithm 5 may not force the condition in Eq. (16.26) if the quantity $|(f_L - f_0)|$ is less than ε_2, and a nonpositive definite $\mathbf{S}_{k+1}$ matrix may on rare occasions arise. To safeguard against this possibility and ensure that a descent direction is achieved in every iteration, the quantity $\boldsymbol{\delta}_k^T\boldsymbol{\gamma}_k$ is checked in step 8b and if it is found to be negative or zero, the unity matrix $\mathbf{I}_n$ is assigned to $\mathbf{S}_{k+1}$.

The DFP and BFGS updating formulas are very similar, and there are no clear theoretical advantages that apply to one and not the other. Indeed, the two formulas are interrelated in terms of a mathematical principle known as *duality*, which allows each of the two formulas to be derived from the other by simple algebraic manipulation. Nevertheless, extensive experimental results reported by Fletcher [6] show that the use of the BFGS formula tends to yield algorithms that are somewhat more tolerant to line-search imprecision. As a consequence, algorithms based on the BFGS formula are somewhat more efficient.

Example 16.2 In an application, the piecewise-continuous function

$$D(\omega) = \begin{cases} 2\omega & \text{for } \ 0 \le \omega < 6 \\ 12 & \text{for } \ 6 \le \omega < 12 \\ -\omega + 24 & \text{for } 12 \le \omega < 16 \\ 8 & \text{for } 16 \le \omega < 22 \end{cases}$$

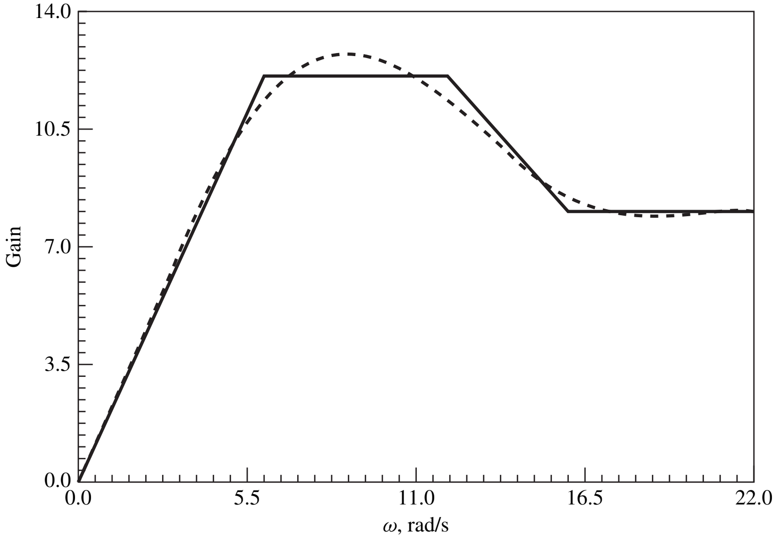


Figure 16.3 Plots of $D(\omega)$ and $P(\omega)$ (Example 16.2).
———— $D(\omega)$; – – – – – $P(\omega)$.

(see Fig. 16.3) has to be approximated by a polynomial of the form

$$P(\omega) = \sum_{k=0}^{5} a_k \omega^k$$

Using Algorithm 5, obtain a set of coefficients a_k for $k = 0, 1, \ldots, 5$ that minimizes the difference between $D(\omega)$ and $P(\omega)$ in the range $0 \leq \omega \leq 22$ in a least-squares sense.

■ **Solution**

A suitable objective function can be constructed as

$$\Psi(\mathbf{x}) = \frac{1}{2}L_2^2 = \frac{1}{2}\sum_{i=1}^{12}[D(\omega_i) - P(\omega_i)]^2$$

where $\omega_i = 2i - 2$ and

$$\mathbf{x} = [x_1\ x_2\ \cdots\ x_6]^T = [a_0\ a_1\ \cdots\ a_5]^T$$

by sampling the error $D(\omega) - P(\omega)$ at 12 points. The first partial derivatives of $\Psi(\mathbf{x})$ can be readily determined as

$$\frac{\partial\Psi(\mathbf{x})}{\partial x_k} = -\sum_{i=1}^{12}[D(\omega_i) - P(\omega_i)]\omega_i^k$$

for $k = 1, 2, \ldots, 6$. Using Algorithm 5 with an initial point $\mathbf{x}_0 = [0\ 0\ \cdots\ 0]^T$ and a termination tolerance $\varepsilon_1 = 10^{-6}$, the coefficients in Table 16.1 were obtained. The progress of the algorithm is illustrated in Table 16.2. The number of function evaluations is equal to the number of evaluations of the objective function $\Psi(\mathbf{x})$ plus the number of evaluations of the partial derivative function $\partial\Psi(\mathbf{x})/\partial x_k$. The polynomial $P(\omega)$ is compared with $D(\omega)$ in Fig. 16.3 and the error between the two is plotted versus frequency

Table 16.1 Coefficients of $P(\omega)$ (Example 16.2)

Coefficient	Value
a_0	$-7.626758E-2$
a_1	1.801233
a_2	$2.389372E-1$
a_3	$-5.286809E-2$
a_4	$2.829081E-3$
a_5	$-4.791669E-5$

Table 16.2 Progress of algorithm (Example 16.2)

k	Funct. evals.	$\Psi(\mathbf{x})$
0	7	5.060000E+2
5	44	3.104894
10	87	1.017671
13	114	1.016952

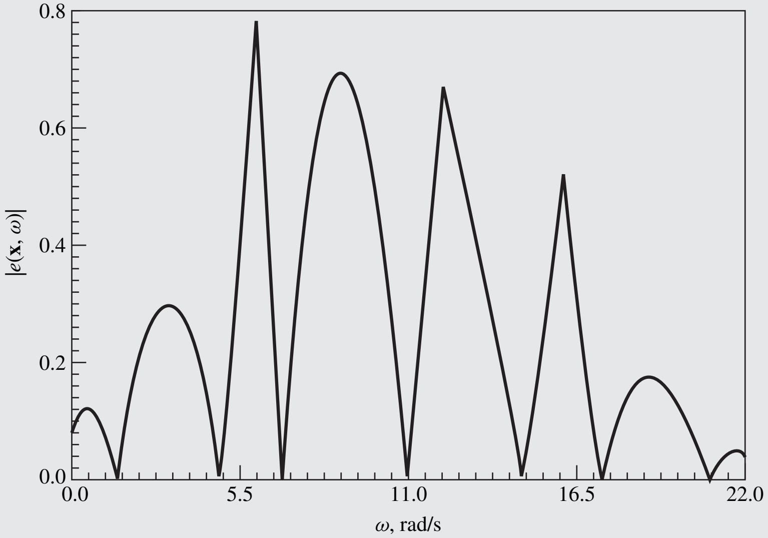


Figure 16.4 Error $|e(\mathbf{x}, \omega)|$ versus ω (Example 16.2).

in Fig. 16.4. Note that the error is unevenly distributed with respect to the frequency. This is a common feature of least-squares solutions and is sometimes of concern.

16.5 MINIMAX ALGORITHMS

The design of digital filters can be accomplished by minimizing one of the norms described in Sec. 16.2. If the L_1 or L_2^2 norm is minimized, then the sum of the magnitudes or the sum of the squares of the elemental errors is minimized. The minimum error achieved usually turns out to be unevenly distributed with respect to the frequency and may exhibit large peaks (e.g., see the error achieved for Example 16.2 depicted in Fig. 16.4) which are often objectionable. If prescribed amplitude response specifications are to be met, the magnitude of the largest elemental error should be minimized and, therefore, the L_∞ norm of the error function should be used. Algorithms developed specifically for the minimization of the L_∞ norm are known as *minimax* algorithms and lead to designs in which the error is uniformly distributed with respect to frequency. The solutions obtained tend to be equiripple, much like the solutions obtained by using the elliptic approximation of Chap. 10, which is, in effect, the minimax solution for filters with piecewise-constant amplitude responses.

The most fundamental minimax algorithm is the so-called *least-pth* algorithm, which involves minimizing an objective function of the type given in Eq. (16.5) for increasing values of p, say $p = 2, 4, 8, \ldots$, and is as follows [12].

Algorithm 6: Least-pth minimax algorithm

1. Input $\breve{\mathbf{x}}_0$ and ε_1. Set $k = 1$, $p = 2$, $\mu = 2$, $\widehat{E}_0 = 10^{99}$.

2. Initialize frequencies $\omega_1, \omega_2, \ldots, \omega_K$.

3. Using $\breve{\mathbf{x}}_{k-1}$ as initial value, minimize

$$\Psi_k(\mathbf{x}) = \widehat{E}(\mathbf{x}) \left\{ \sum_{i=1}^{K} \left[\frac{|e_i(\mathbf{x})|}{\widehat{E}(\mathbf{x})} \right]^p \right\}^{1/p} \tag{16.29}$$

where

$$\widehat{E}(\mathbf{x}) = \max_{1 \le i \le K} |e_i(\mathbf{x})|$$

with respect to $\mathbf{x}$, to obtain $\breve{\mathbf{x}}_k$. Set $\widehat{E}_k = \widehat{E}(\breve{\mathbf{x}})$.

4. If $|\widehat{E}_{k-1} - \widehat{E}_k| < \varepsilon_1$, then output $\breve{\mathbf{x}}_k$ and $\widehat{E}_k$, and stop. Otherwise, set $p = \mu p$, $k = k + 1$ and go to step 3. ●

The underlying principle for Algorithm 6 is that the minimax problem is solved by solving a sequence of closely related problems whereby the solution of one renders the solution of the next one more tractable. Parameter μ in step 1, which must obviously be an integer, should not be too large in order to avoid numerical ill-conditioning. A value of 2 was found to give good results.

The minimization in step 3 can be carried out by using any unconstrained optimization algorithm, for example, Algorithm 5 described in the previous section. The gradient of $\Psi_k(\mathbf{x})$ is given by [12]

$$\nabla \Psi_k(\mathbf{x}) = \left\{ \sum_{i=1}^{K} \left[\frac{|e_i(\mathbf{x})|}{\widehat{E}(\mathbf{x})} \right]^p \right\}^{(1/p)-1} \sum_{i=1}^{K} \left[\frac{|e_i(\mathbf{x})|}{\widehat{E}(\mathbf{x})} \right]^{p-1} \boldsymbol{\nabla} |e_i(\mathbf{x})| \tag{16.30}$$

The preceding algorithm works very well, except that it requires a considerable amount of computation. An alternative and much more efficient minimax algorithm is one described in [13, 14]. This algorithm is based on principles developed by Charalambous [15] and involves the minimization of the objective function

$$\Psi(\mathbf{x}, \boldsymbol{\lambda}, \xi) = \sum_{i \in I_1} \frac{1}{2} \lambda_i [\phi_i(\mathbf{x}, \xi)]^2 + \sum_{i \in I_2} \frac{1}{2} [\phi_i(\mathbf{x}, \xi)]^2 \tag{16.31}$$

where ξ and λ_i for $i = 1, 2, \ldots, K$ are constants

$$\phi_i(\mathbf{x}, \xi) = |e_i(\mathbf{x})| - \xi$$

$$I_1 = \{i : \phi_i(\mathbf{x}, \xi) > 0 \text{ and } \lambda_i > 0\} \tag{16.32}$$

and

$$I_2 = \{i : \phi_i(\mathbf{x}, \xi) > 0 \text{ and } \lambda_i = 0\} \tag{16.33}$$

The halves in Eq. (16.31) are included for the purpose of simplifying the gradient (see Eq. (16.34)).

If

(*a*) the second-order sufficiency conditions for a minimum of $\widehat{E}(\mathbf{x})$ hold at $\breve{\mathbf{x}}$,

(*b*) $\lambda_i = \breve{\lambda}_i$ for $i = 1, 2, \ldots, K$ where $\breve{\lambda}_i$ are the minimax multipliers corresponding to the minimum point $\breve{\mathbf{x}}$ of $\widehat{E}(\mathbf{x})$, and

(*c*) $\widehat{E}(\breve{\mathbf{x}}) - \xi$ is sufficiently small

then it can be proved that $\breve{\mathbf{x}}$ is a *strong* local minimum point of function $\Psi(\mathbf{x}, \boldsymbol{\lambda}, \xi)$ given by Eq. (16.31) (see [15] for details). In practice, the conditions in (*a*) are satisfied for most practical problems. Consequently, if multipliers λ_i are forced to approach the minimax multipliers $\breve{\lambda}_i$ and ξ is forced to approach $\widehat{E}(\breve{\mathbf{x}})$, then the minimization of $\widehat{E}(\mathbf{x})$ can be accomplished by minimizing $\Psi(\mathbf{x}, \boldsymbol{\lambda}, \xi)$ with respect to $\mathbf{x}$. A minimax algorithm based on these principles is as follows:

Algorithm 7: Charalambous minimax algorithm

1. Input $\breve{\mathbf{x}}_0$ and ε_1. Set $k = 1, \xi_1 = 0, \lambda_{11} = \lambda_{12} = \ldots = \lambda_{1K} = 1, \widehat{E}_0 = 10^{99}$.
2. Initialize frequencies $\omega_1, \omega_2, \ldots, \omega_K$.
3. Using $\breve{\mathbf{x}}_{k-1}$ as initial value, minimize $\Psi(\mathbf{x}, \boldsymbol{\lambda}_k, \xi_k)$ with respect to $\mathbf{x}$ to obtain $\breve{\mathbf{x}}_k$. Set

$$\widehat{E}_k = \widehat{E}(\breve{\mathbf{x}}_k) = \max_{1 \le i \le K} |e_i(\breve{\mathbf{x}}_k)|$$

4. Compute

$$\Phi_k = \sum_{i \in I_1} \lambda_{ki} \phi_i(\breve{\mathbf{x}}_k, \xi_k) + \sum_{i \in I_2} \phi_i(\breve{\mathbf{x}}_k, \xi_k)$$

and update

$$\lambda_{(k+1)i} = \begin{cases} \lambda_{ki} \phi_i(\breve{\mathbf{x}}_k, \xi_k)/\Phi_k & \text{for } i \in I_1 \\ \phi_i(\breve{\mathbf{x}}_k, \xi_k)/\Phi_k & \text{for } i \in I_2 \\ 0 & \text{for } i \in I_3 \end{cases}$$

for $i = 1, 2, \ldots, K$ where

$$I_1 = \{i : \phi_i(\breve{\mathbf{x}}_k, \xi_k) > 0 \text{ and } \lambda_{ki} > 0\}$$

$$I_2 = \{i : \phi_i(\breve{\mathbf{x}}_k, \xi_k) > 0 \text{ and } \lambda_{ki} = 0\}$$

and

$$I_3 = \{i : \phi_i(\breve{\mathbf{x}}_k, \xi_k) \le 0\}$$

5. Compute

$$\xi_{k+1} = \sum_{i=1}^{K} \lambda_{(k+1)i} |e_i(\breve{\mathbf{x}})|$$

6. If $|\widehat{E}_{k-1} - \widehat{E}_k| < \varepsilon_1$, then output $\breve{\mathbf{x}}_k$ and $\widehat{E}_k$, and stop. Otherwise, set $k = k + 1$ and go to step 3. ●

The gradient of $\Psi(\mathbf{x}, \boldsymbol{\lambda}_k, \xi_k)$, which is required in step 3 of the algorithm, is given by

$$\begin{aligned}\boldsymbol{\nabla}\Psi(\mathbf{x}, \boldsymbol{\lambda}_k, \xi_k) &= \sum_{i\in I_1} \lambda_{ki}\phi_i(\mathbf{x}, \xi_k)\boldsymbol{\nabla}|e_i(\mathbf{x})| \\ &\quad + \sum_{i\in I_2} \phi_i(\mathbf{x}, \xi_k)\boldsymbol{\nabla}|e_i(\mathbf{x})| \qquad (16.34)\end{aligned}$$

Constant ξ is a lower bound of the minimum of $\widehat{E}(\mathbf{x})$ and as the algorithm progresses, it approaches $\widehat{E}(\breve{\mathbf{x}})$ from below. Consequently, the number of functions $\phi_i(\mathbf{x}, \xi)$ that do not satisfy either Eq. (16.32) or Eq. (16.33) increases rapidly with the number of iterations. Since the derivatives of these functions are unnecessary in the minimization of $\Psi(\mathbf{x}, \boldsymbol{\lambda}, \xi)$, they need not be evaluated. This increases the efficiency of the algorithm quite significantly.

As in Algorithm 6, the minimization in step 3 of Algorithm 7 can be carried out by using Algorithm 5.

16.6 IMPROVED MINIMAX ALGORITHMS

To achieve good results in the above minimax algorithms, the sampling of $e(\mathbf{x},\omega)$ with respect to ω must be dense; otherwise, the error function may develop spikes in the intervals between sampling points during the minimization. This problem is usually overcome by using a fairly large value of K of the order of three to six times the number of variables, e.g., if an eighth-order digital filter is to be designed, a value as high as 100 may be required. In such a case, each function evaluation in the minimization of the objective function would involve computing the gain of the filter as many as 100 times. A single optimization may sometimes necessitate 300 to 600 function evaluations, and a minimax algorithm like Algorithm 6 or 7 may require 5 to 10 unconstrained optimizations to converge. Consequently, the amount of computation required to complete a design is considerable.

A technique will now be described that can be used to suppress spikes in the error function without using a large value of K [16]. The technique entails the application of *nonuniform variable sampling* and involves the following steps:

1. Evaluate the error function in Eq. (16.3) with respect to a dense set of uniformly spaced frequencies that span the frequency band of interest, say $\bar{\omega}_1, \bar{\omega}_2, \ldots, \bar{\omega}_L$, where L is fairly large, of the order of $10 \times K$.
2. Segment the frequency band of interest into K intervals.
3. For each of the K intervals, find the frequency that yields maximum error. Let these frequencies be $\widehat{\omega}_i$ for $i = 1, 2, \ldots, K$.
4. Use frequencies $\widehat{\omega}_i$ as sample frequencies in the evaluation of the objective function, i.e., set $\omega_i = \widehat{\omega}_i$ for $i = 1, 2, \ldots, K$.

By applying the above nonuniform sampling technique before the start of the second and subsequent optimizations, *frequency points at which spikes are beginning to form are located and are used as sample points in the next optimization.* In this way, the error at these frequencies is reduced and the formation of spikes is suppressed.

Assume that a digital filter is required to have a specified amplitude response with respect to a frequency band B that extends from $\bar{\omega}_1$ to $\bar{\omega}_L$, and let $\bar{\omega}_1, \bar{\omega}_2, \ldots, \bar{\omega}_L$ be uniformly-spaced frequencies such that

$$\bar{\omega}_i = \bar{\omega}_{i-1} + \Delta\omega$$

for $i = 2, 3, \ldots, L$ where

$$\Delta\omega = \frac{\bar{\omega}_L - \bar{\omega}_1}{L - 1} \tag{16.35}$$

These frequency points may be referred to as *virtual sample points*. Band B can be segmented into K intervals, say Ω_1 to Ω_K such that Ω_1 and Ω_K are of width $\Delta\omega/2$, Ω_2 and Ω_{K-1} are of width $l\Delta\omega$, and Ω_i for $i = 3, 4, \ldots, K-2$ are of width $2l\Delta\omega$ where l is an integer. These requirements can be satisfied by letting

$$\Omega_1 = \left\{\omega : \bar{\omega}_1 \le \omega < \bar{\omega}_1 + \frac{1}{2}\Delta\omega\right\}$$

$$\Omega_2 = \left\{\omega : \bar{\omega}_1 + \frac{1}{2}\Delta\omega \le \omega < \bar{\omega}_1 + \left(l + \frac{1}{2}\right)\Delta\omega\right\}$$

$$\Omega_i = \left\{\omega : \bar{\omega}_1 + \left[(2i-5)l + \frac{1}{2}\right]\Delta\omega \le \omega < \bar{\omega}_1 + \left[(2i-3)l + \frac{1}{2}\right]\Delta\omega\right\}$$

for $i = 3, 4, \ldots, K-2$

$$\Omega_{K-1} = \left\{\omega : \bar{\omega}_1 + \left[(2K-7)l + \frac{1}{2}\right]\Delta\omega \le \omega < \bar{\omega}_1 + \left[(2K-6)l + \frac{1}{2}\right]\Delta\omega\right\}$$

and

$$\Omega_K = \left\{\omega : \bar{\omega}_1 + \left[(2K-6)l + \frac{1}{2}\right]\Delta\omega \le \omega \le \bar{\omega}_L\right\}$$

where

$$\bar{\omega}_L = \bar{\omega}_1 + [(2K-6)l + 1]\Delta\omega \tag{16.36}$$

The scheme is feasible if

$$L = (2K-6)l + 2 \tag{16.37}$$

according to Eqs. (16.35) and (16.36), and is illustrated in Fig. 16.5 for the case where $K = 8$ and $l = 5$.

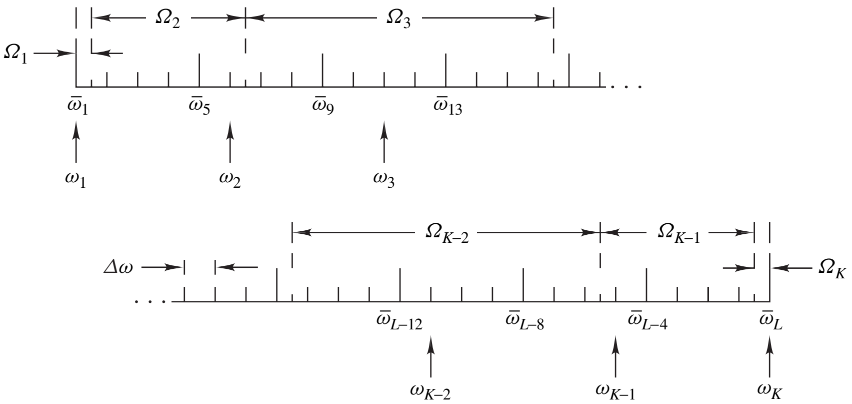


Figure 16.5 Segmentation of frequency axis.

In the above segmentation scheme, there is only one sample in each of intervals Ω_1 and Ω_K, l samples in each of intervals Ω_2 and Ω_{K-1}, and $2l$ samples in each of intervals $\Omega_3, \Omega_4, \ldots, \Omega_{K-2}$, as can be seen in Fig. 16.5. Thus step 3 of the technique will yield $\widehat{\omega}_1 = \bar{\omega}_1$ and $\widehat{\omega}_K = \bar{\omega}_L$, that is, the lower and upper band edges are forced to remain sample frequencies throughout the optimization. This strategy leads to two advantages: (*a*) the error at the band edges is always minimized, and (*b*) a somewhat higher sampling density is maintained near the band edges where spikes are more likely to occur.

In the above technique, the required amplitude response needs to be specified with respect to a *dense* set of frequency points. This problem can be overcome through the use of interpolation. Let us assume that the amplitude response is specified at frequencies $\tilde{\omega}_1$ to $\tilde{\omega}_S$, where $\tilde{\omega}_1 = \bar{\omega}_1$ and $\tilde{\omega}_S = \bar{\omega}_L$. The required amplitude response for any frequency interval spanned by four successive specification points, say $\tilde{\omega}_j \le \omega \le \tilde{\omega}_{j+3}$, can be represented by a third-order polynomial of ω of the form

$$M_0(\omega) = a_{0j} + a_{1j}\omega + a_{2j}\omega^2 + a_{3j}\omega^3 \tag{16.38}$$

and by varying j from 1 to $S-3$, a set of $S-3$ third-order polynomials can be obtained which can be used to interpolate the amplitude response to any desired degree of resolution. To achieve maximum interpolation accuracy, each of these polynomials should as far as possible be used only in the center of its frequency range of validity. Hence, the first and last polynomials should be used for frequency ranges $\tilde{\omega}_1 \le \omega < \tilde{\omega}_3$ and $\tilde{\omega}_{S-2} \le \omega \le \tilde{\omega}_S$, respectively, and the jth polynomial for $2 \le j \le S-4$ should be used for the frequency range $\tilde{\omega}_{j+1} \le \omega < \tilde{\omega}_{j+2}$.

Coefficients a_{ij} for $i = 0, 1, \ldots, 3$ and $j = 1$ to $S-3$ can be determined by computing $\tilde{\omega}_m$, $(\tilde{\omega}_m)^2$, and $(\tilde{\omega}_m)^3$ for $m = j, j+1, \ldots, j+3$, and then constructing the system of simultaneous equations

$$\tilde{\mathbf{\Omega}}_j \mathbf{a}_j = \mathbf{M}_{0j} \tag{16.39}$$

where

$$\mathbf{a}_j = \begin{bmatrix} a_{0j} & \cdots & a_{3j} \end{bmatrix} \qquad \text{and} \qquad \mathbf{M}_{0j} = \begin{bmatrix} M_0(\tilde{\omega}_j) & \cdots & M_0(\tilde{\omega}_{j+3}) \end{bmatrix}^T$$

are column vectors and $\boldsymbol{\Omega}_j$ is the 4 × 4 matrix given by

$$\tilde{\boldsymbol{\Omega}}_j = \begin{bmatrix} 1 & \tilde{\omega}_j & (\tilde{\omega}_j)^2 & (\tilde{\omega}_j)^3 \\ 1 & \tilde{\omega}_{j+1} & (\tilde{\omega}_{j+1})^2 & (\tilde{\omega}_{j+1})^3 \\ 1 & \tilde{\omega}_{j+2} & (\tilde{\omega}_{j+2})^2 & (\tilde{\omega}_{j+2})^3 \\ 1 & \tilde{\omega}_{j+3} & (\tilde{\omega}_{j+3})^2 & (\tilde{\omega}_{j+3})^3 \end{bmatrix}$$

Therefore, from Eq. (16.39) we have

$$\mathbf{a}_j = \tilde{\boldsymbol{\Omega}}_j^{-1}\mathbf{M}_{0j} \tag{16.40}$$

The above nonuniform sampling technique can be incorporated in Algorithm 6 by replacing steps 1, 2, and 4 by the modified steps 1*A*, 2*A*, and 4*A* listed below. The filter to be designed is assumed to be a single-band filter, for the sake of simplicity, although the technique is applicable to filters with an arbitrary number of bands.

1*A*. *a*. Input $\breve{\mathbf{x}}_0$ and ε_1. Set $k = 1$, $p = 2$, $\mu = 2$, $\widehat{E}_0 = 10^{99}$. Initialize K.
b. Input the required amplitude response $M_0(\tilde{\omega}_m)$ for $m = 1, 2, \ldots, S$.
c. Compute L and $\Delta\omega$ using Eqs. (16.37) and (16.35), respectively.
d. Compute coefficients a_{ij} for $i = 0, 1, \ldots, 3$ and $j = 1$ to $S - 3$ using Eq. (16.40).
e. Compute the required amplitude response for $\bar{\omega}_1, \bar{\omega}_2, \ldots, \bar{\omega}_L$ using Eq. (16.38).

2*A*. Set $\omega_1 = \bar{\omega}_1$, $\omega_2 = \bar{\omega}_{1+l}$, $\omega_i = \bar{\omega}_{2(i-2)l+1}$ for $i = 3, 4, \ldots, K-2$, $\omega_{K-1} = \bar{\omega}_{L-l}$, and $\omega_K = \bar{\omega}_L$.

4*A*. *a*. Compute $|e_i(\breve{\mathbf{x}}_k)|$ for $i = 1, 2, \ldots, L$ using Eqs. (16.3) and (16.4).
b. Determine frequencies $\widehat{\omega}_i$ for $i = 1, 2, \ldots, K$ and

$$\widehat{P}_k = \widehat{P}(\breve{\mathbf{x}}_k) = \max_{1 \le i \le L} |e_i(\breve{\mathbf{x}}_k)|$$

c. Set $\omega_i = \widehat{\omega}_i$ for $i = 1, 2, \ldots, K$.
d. If $|\widehat{E}_{k-1} - \widehat{E}_k| < \varepsilon_1$ and $|\widehat{P}_k - \widehat{E}_k| < \varepsilon_1$, then output $\breve{\mathbf{x}}_k$ and $\widehat{E}_k$, and stop. Otherwise, set $p = \mu p$, $k = k + 1$ and go to step 3.

Similarly, the technique can be applied to Algorithm 7, by replacing steps 1, 2, and 6 by the following modified steps:

1*A*. *a*. Input $\breve{\mathbf{x}}_0$ and ε_1. Set $k = 1$, $\xi_1 = 0$, $\lambda_{11} = \lambda_{12} = \ldots = \lambda_{1K} = 1$, $\widehat{E}_0 = 10^{99}$. Initialize K.
b. Input the required amplitude response $M_0(\tilde{\omega}_m)$ for $m = 1, 2, \ldots, S$.
c. Compute L and $\Delta\omega$ using Eqs. (16.37) and (16.35), respectively.
d. Compute coefficients a_{ij} for $i = 0, 1, \ldots, 3$ and $j = 1$ to $S - 3$ using Eq. (16.40).
e. Compute the required amplitude response for $\bar{\omega}_1, \bar{\omega}_2, \ldots, \bar{\omega}_L$ using Eq. (16.38).

2*A*. Set $\omega_1 = \bar{\omega}_1$, $\omega_2 = \bar{\omega}_{1+l}$, $\omega_i = \bar{\omega}_{2(i-2)l+1}$ for $i = 3, 4, \ldots, K-2$, $\omega_{K-1} = \bar{\omega}_{L-l}$, and $\omega_K = \bar{\omega}_L$.

6*A*. *a*. Compute $|e_i(\breve{\mathbf{x}}_k)|$ for $i = 1, 2, \ldots, L$ using Eqs. (16.3) and (16.4).
b. Determine frequencies $\widehat{\omega}_i$ for $i = 1, 2, \ldots, K$ and

$$\widehat{P}_k = \widehat{P}(\breve{\mathbf{x}}_k) = \max_{1 \le i \le L} |e_i(\breve{\mathbf{x}}_k)|$$

c. Set $\omega_i = \widehat{\omega}_i$ for $i = 1, 2, \ldots, K$.

d. If $|\widehat{E}_{k-1} - \widehat{E}_k| < \varepsilon_1$ and $|\widehat{P}_k - \widehat{E}_k| < \varepsilon_1$, then output $\breve{\mathbf{x}}_k$ and $\widehat{E}_k$, and stop. Otherwise, set $k = k + 1$ and go to step 3.

In step 2*A*, the initial sample frequencies ω_1 and ω_K are assumed to be at the left-hand and right-hand band edges, respectively; ω_2 and ω_{K-1} are taken to be the last and first frequencies in intervals Ω_2 and Ω_{K-1}, respectively; and each of frequencies $\omega_3, \omega_4, \ldots, \omega_{K-2}$ is set near the center of each of intervals $\Omega_3, \Omega_4, \ldots, \Omega_{K-2}$. This assignment is illustrated in Fig. 16.5 for the case where $K = 8$ and $l = 5$.

Without the nonuniform sampling technique, the number of samples K should be chosen to be of the order of three to six times the number of variables, depending on the selectivity of the filter. While a value of 50 may be entirely satisfactory for an eighth-order lowpass filter with a wide transition band, a value of 100 may not be adequate for a highly selective narrow-band bandpass filter of the same order. With the technique, the number of virtual samples is approximately equal to $2l \times K$, according to Eq. (16.37). As l is increased above unity, the frequencies of maximum error $\widehat{\omega}_i$ become progressively more precise, owing to the increased resolution; however, the amount of computation required in step 4*A* of Algorithm 6 or step 6*A* of Algorithm 7 is proportionally increased. Eventually, a situation of diminishing returns is reached whereby further increases in l bring about only slight improvements in the precision of the $\widehat{\omega}_i$'s. The values $K = 35$ and $l = 5$, which correspond to 35 actual and 322 virtual sample points, were found to give excellent results for a diverse range of designs, including some complex 28th-order phase-equalizer designs (see Sec. 16.8).

16.7 DESIGN OF RECURSIVE FILTERS

The application of Algorithms 6 and 7 for the design of recursive digital filters can be readily accomplished by obtaining expressions for the objective functions $\Psi_k(\mathbf{x})$ and $\Psi(\mathbf{x}, \boldsymbol{\lambda}_k, \xi_k)$ and their gradients.

16.7.1 Objective Function

The amplitude response of an Nth-order filter is given by Eqs. (16.1) and (16.2) as

$$M(\mathbf{x}, \omega) = H_0 \prod_{j=1}^{J} \frac{N_j(\omega)}{D_j(\omega)}$$

where

$$N_j(\omega) = \left[1 + a_{0j}^2 + a_{1j}^2 + 2a_{1j}(1 + a_{0j}) \cos \omega T + 2a_{0j} \cos 2\omega T\right]^{\frac{1}{2}}$$

and

$$D_j(\omega) = \left[1 + b_{0j}^2 + b_{1j}^2 + 2b_{1j}(1 + b_{0j}) \cos \omega T + 2b_{0j} \cos 2\omega T\right]^{\frac{1}{2}}$$

for $j = 1, 2, \ldots, J$. Hence, Eqs. (16.3) and (16.4) yield

$$e_i(\mathbf{x}) = M(\mathbf{x}, \omega_i) - M_0(\omega_i)$$

and from Eqs. (16.29) and (16.31), $\Psi_k(\mathbf{x})$ and $\Psi(\mathbf{x}, \boldsymbol{\lambda}_k, \xi_k)$ can be formed.

16.7.2 Gradient Information

Since $M_0(\omega_i)$ in the formula for the error function is a constant, we obtain

$$\frac{\partial e_i(\mathbf{x})}{\partial a_{0l}} = \frac{a_{0l} + a_{1l}\cos\omega_i T + \cos 2\omega_i T}{[N_l(\omega_i)]^2} \cdot M(\mathbf{x}, \omega_i)$$
$$\frac{\partial e_i(\mathbf{x})}{\partial a_{1l}} = \frac{a_{1l} + (1 + a_{0l})\cos\omega_i T}{[N_l(\omega_i)]^2} \cdot M(\mathbf{x}, \omega_i)$$
$$\frac{\partial e_i(\mathbf{x})}{\partial b_{0l}} = -\frac{b_{0l} + b_{1l}\cos\omega_i T + \cos 2\omega_i T}{[D_l(\omega_i)]^2} \cdot M(\mathbf{x}, \omega_i)$$
$$\frac{\partial e_i(\mathbf{x})}{\partial b_{1l}} = -\frac{b_{1l} + (1 + b_{0l})\cos\omega_i T}{[D_l(\omega_i)]^2} \cdot M(\mathbf{x}, \omega_i)$$
$$\frac{\partial e_i(\mathbf{x})}{\partial H_0} = \frac{1}{H_0} \cdot M(\mathbf{x}, \omega_i)$$

for $l = 1, 2, \ldots, J$ and $i = 1, 2, \ldots, K$. Hence the gradient of $e_i(\mathbf{x})$, namely, $\boldsymbol{\nabla} e_i(\mathbf{x})$, can be formed, and since

$$\boldsymbol{\nabla}|e_i(\mathbf{x})| = \operatorname{sgn} e_i(\mathbf{x})\boldsymbol{\nabla} e_i(\mathbf{x})$$

where

$$\operatorname{sgn} e_i(\mathbf{x}) = \begin{cases} 1 & \text{if } e_i(\mathbf{x}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

$\boldsymbol{\nabla}\Psi_k(\mathbf{x})$ and $\boldsymbol{\nabla}\Psi(\mathbf{x}, \boldsymbol{\lambda}_k, \xi_k)$ can be evaluated using Eqs. (16.30) and (16.34), respectively.

16.7.3 Stability

The minimax algorithms considered will yield filters which may or may not be stable since the transfer function obtained may have poles outside the unit circle of the z plane. However, the problem can be easily eliminated by replacing the offending poles by their reciprocals and simultaneously adjusting the multiplier constant H_0 so as to compensate for the change in gain. This stabilization technique is described in Sec. 11.4.

16.7.4 Minimum Filter Order

A problem associated with the design of filters with arbitrary amplitude and/or phase responses is that there are no known methods for the prediction of the filter order that will limit the approximation error to within prescribed bounds. However, satisfactory results can often be achieved on a cut-and-try basis by designing filters of increasing orders until the error is sufficiently small to satisfy the requirements.

16.7.5 Use of Weighting

If $\mathbf{x} = \breve{\mathbf{x}}$ is a solution in the design of a recursive M-band filter, then the error at convergence, namely,

$$e(\breve{\mathbf{x}}, \omega) = M(\breve{\mathbf{x}}, \omega) - M_0(\omega)$$

would tend to be uniformly distributed in the passband(s) and stopband(s) such that $-\delta \leq e(\breve{\mathbf{x}}, \omega) \leq \delta$ in each and every band, where δ is some positive constant. In such a design, the maximum passband ripple and minimum stopband attenuation would be given by

$$A_p = 20 \log \frac{1 + \delta}{1 - \delta} \text{ dB} \qquad \text{and} \qquad A_a = -20 \log \delta \text{ dB}$$

respectively (see Sec. 9.4.6). In effect, the passband ripple would be correlated to the minimum stopband attenuation and a small or large passband ripple would be associated with a small or large minimum stopband attenuation. If the required specifications call for a passband ripple that is different from the stopband ripple, then by using a sufficiently large filter order one would be able to obtain a filter that just satisfies the required specifications with respect to the most critical band specification and oversatisfies the specifications in all the other bands. Such a design would, of course, be suboptimal with respect to the required specifications.

The above problem can be circumvented through the use of weighting as was done in Chap. 15 for the case of equiripple nonrecursive filters. The discretized error can be formulated as

$$e_i(\mathbf{x}) = w_m[M(\mathbf{x}, \omega_i) - M_0(\omega_i)]$$

where $m = 1, 2, \ldots, M$ and from Eq. (16.29) or Eq. (16.31) a weighted objective function can be obtained. Minimization of the weighted objective function will result in a uniformly distributed weighted error such that $-\delta \leq e(\breve{\mathbf{x}}, \omega) \leq \delta$ and, therefore, the actual error in the various bands will be

$$[M(\breve{\mathbf{x}}, \omega_i) - M_0(\omega_i)] = \frac{\delta}{w_m}$$

for $m = 1, 2, \ldots, M$. Thus if a band weighting constant w_m is larger or smaller than unity, the actual band error will be reduced or increased relative to the value achieved without weighting.

The required filter specifications can be readily used to calculate the required band errors $\delta_1, \delta_2, \ldots, \delta_M$ and if we assume that an equiripple solution exists such that $-\delta \leq e(\breve{\mathbf{x}}, \omega) \leq \delta$, then at convergence we would have

$$\delta_1 = \frac{\delta}{w_1} \qquad \delta_2 = \frac{\delta}{w_2} \qquad \cdots \qquad \delta_M = \frac{\delta}{w_M}$$

If we assume that a solution exists that would satisfy the required specification in the first band with a weighting constant $w_1 = 1$, then the required weighting constants for the remaining bands can be deduced as

$$w_2 = \frac{\delta_1}{\delta_2} \qquad w_3 = \frac{\delta_1}{\delta_3} \qquad \cdots \qquad w_M = \frac{\delta_1}{\delta_M}$$

The use of this weighting scheme will result in a filter in which the band errors are in the correct proportion with respect to the specifications, and by using a sufficiently high filter order, all the specifications will be uniformly satisfied. In this way, it may be possible to find a lower-order approximation that would satisfy the required specifications, which would translate into a more economical design.

Example 16.3 A lowpass digital filter is to be used in cascade with a D/A converter. The overall amplitude response from the input of the filter to the output of the D/A converter is required to be

$$M(\omega) = \begin{cases} 1.0 & \text{for } 0 \le \omega \le 4 \times 10^4 \text{rad/s} \\ 0.01 & \text{for } 4.5 \times 10^4 \le \omega \le 10^5 \end{cases}$$

and the amplitude response of the D/A converter is given by

$$\phi(\omega) = \left| \frac{\sin(\omega\tau/2)}{\omega\tau/2} \right|$$

where τ is the pulse duration at the output of the D/A converter (see Sec. 6.10). Design the lowpass filter using Algorithm 7 first without and then with the nonuniform sampling technique of Sec. 16.6 and compare the results obtained. Use an eighth-order transfer function and assume that $\omega_s = 2 \times 10^5$ rad/s and $\tau = T$.

■ **Solution**

The amplitude response of the filter must be modified as [17]

$$\tilde{M}(\omega) = \begin{cases} 1.0/\phi(\omega) & \text{for } 0 \le \omega \le 4 \times 10^4 \text{ rad/s} \\ 0.01/\phi(\omega) & \text{for } 4.5 \times 10^4 \le \omega \le 10^5 \end{cases}$$

to achieve the required amplitude response between the input of the filter and the output of the D/A converter.

The amount of computation required by optimization methods in general and the quality of the solution obtained tend to depend heavily on the initial solution assumed. If the initial point is close to the actual solution, the amount of computation tends to be low and the precision of the solution tends to be high. In this example, a good initial estimate of the solution can be obtained by designing an eighth-order lowpass filter with passband ripple $A_p = 0.1$ dB, minimum stopband attenuation $A_a = 59.5$ dB, passband edge $\omega_p = 4.0 \times 10^4$ rad/s, and stopband edge $\omega_a = 4.5 \times 10^4$ rad/s. A lowpass filter that would satisfy these specifications can be readily designed using the method of Chap. 12. The transfer-function coefficients of a design based on the elliptic approximation are given in Table 16.3.

Using Algorithm 7 with $K = 40$ (25 sample points in the passband and 15 in the stopband) first without and then with the technique of Sec. 16.6, designs A and B of Table 16.3 were obtained. The progress of the algorithm is illustrated in Table 16.4.

Table 16.3 Coefficients of $H(z)$ (Example 16.3)

	j	a_{0j}	a_{1j}	b_{0j}	b_{1j}
Initial filter	1	1.0	1.663591	$2.964920E-1$	$-9.685886E-1$
	2	1.0	$4.646911E-1$	$5.578139E-1$	$-7.881536E-1$
	3	1.0	$-1.082131E-1$	$7.999954E-1$	$-6.314471E-1$
	4	1.0	$-2.936755E-1$	$9.452177E-1$	$-5.696838E-1$
	$H_0 = 1.375814E-2$				
Design A	1	1.201422	1.802335	$1.826366E-1$	$-7.094977E-1$
	2	1.023690	$5.173944E-1$	$4.754965E-1$	$-6.411708E-1$
	3	$9.871557E-1$	$-9.208725E-2$	$7.562144E-1$	$-5.689563E-1$
	4	$9.970934E-1$	$-2.981699E-1$	$9.334971E-1$	$-5.428448E-1$
	$H_0 = 1.987973E-2$				
Design B	1	1.255164	1.822823	$1.916731E-1$	$-7.257967E-1$
	2	1.048624	$4.620755E-1$	$4.905558E-1$	$-6.448195E-1$
	3	1.003053	$-1.109377E-1$	$7.657275E-1$	$-5.639906E-1$
	4	1.000126	$-2.939426E-1$	$9.349386E-1$	$-5.365697E-1$
	$H_0 = 2.000669E-2$				

The magnitude of the error function for each of the two designs is plotted in Fig. 16.6*a* and *b*. As can be seen, spikes are present in the error function of design A but are entirely eliminated in design B through the use of the technique in Sec. 16.6. The amplitude response achieved in design B is illustrated in Fig. 16.7.

Table 16.4 Progress of algorithm (Example 16.3)

	Design A		**Design B**	
k	ξ	$\Psi(\mathbf{x}, \lambda_k, \xi_k)$	ξ	$\Psi(\mathbf{x}, \lambda_k, \xi_k)$
1	0.0	$7.509080E-6$	0.0	$7.509080E-6$
2	$7.510177E-4$	$5.098063E-9$	$7.510177E-4$	$6.274917E-8$
3	$8.854903E-4$	$6.874848E-11$	$1.158158E-3$	$3.401966E-9$
4	$9.013783E-4$	$2.732611E-13$	$1.250634E-3$	$3.311865E-11$
5	$9.023167E-4$	$2.371856E-15$	$1.260096E-3$	$4.468298E-14$

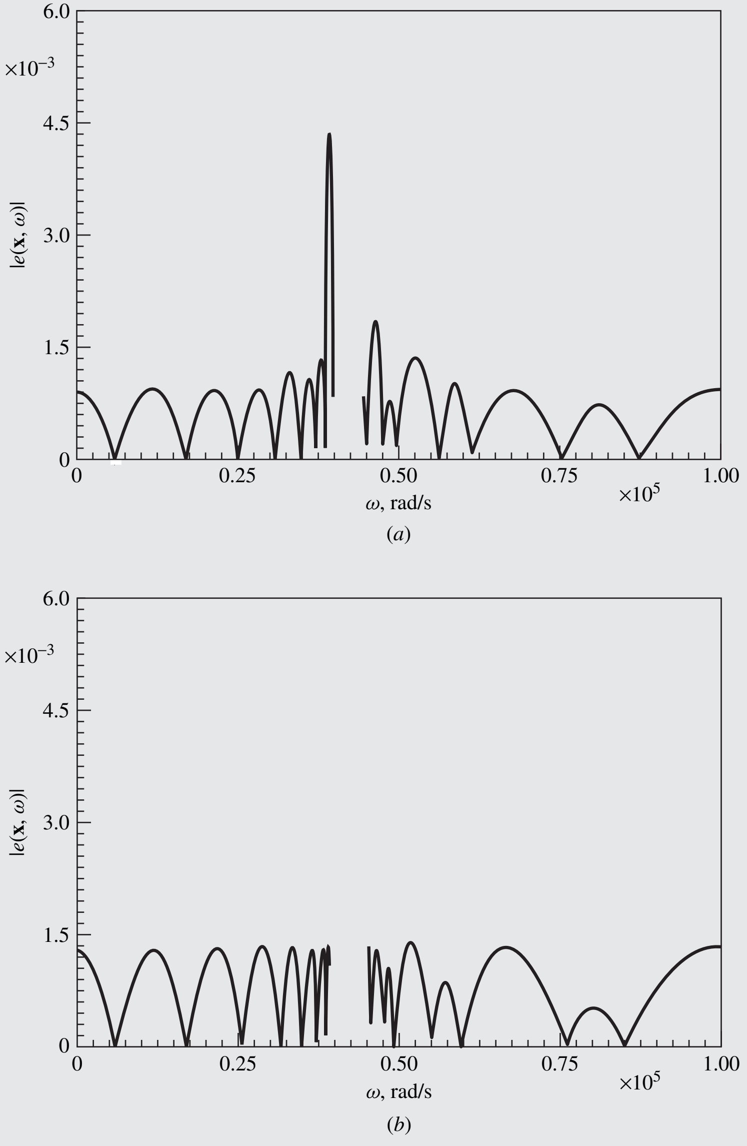
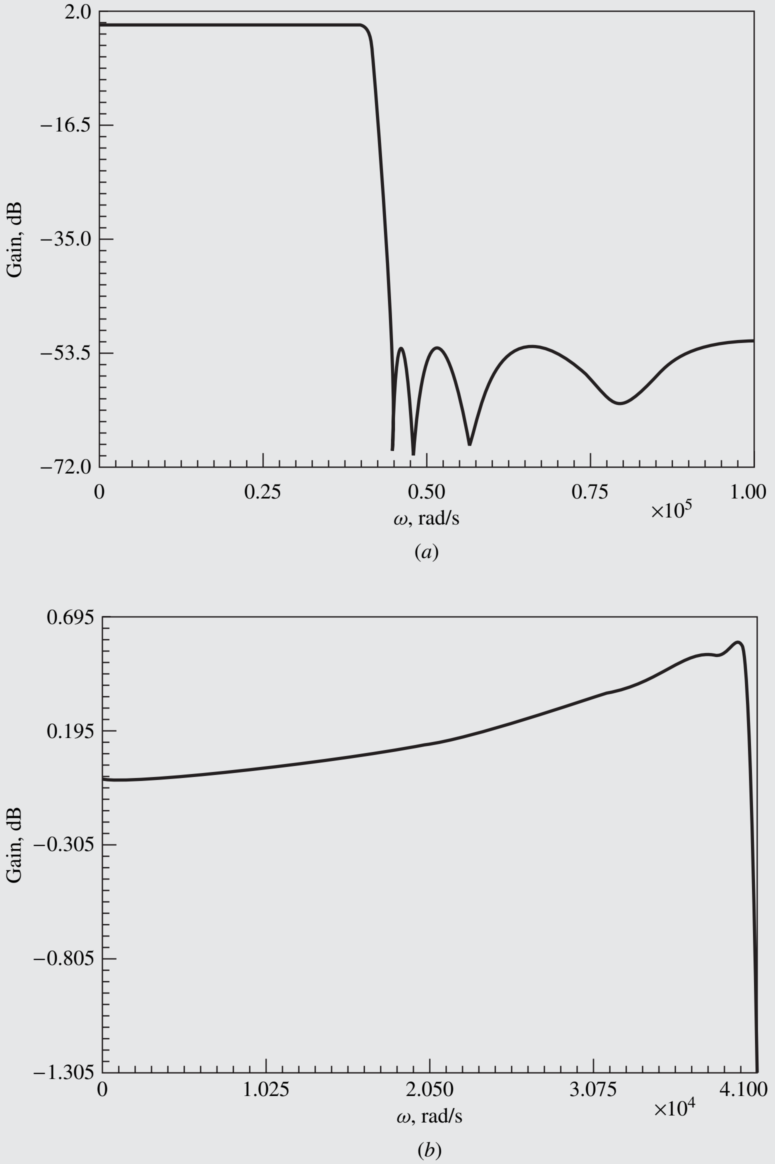


Figure 16.6 Error $|e(\mathbf{x}, \omega)|$ versus ω (Example 16.3): (*a*) Without the technique of Sec. 16.6, (*b*) with the technique of Sec. 16.6.



(*a*)

(*b*)

Figure 16.7 Amplitude response of lowpass filter (Example 16.3): (*a*) For $0 \leq \omega \leq 10^5$, (*b*) for $0 \leq \omega \leq 4.1 \times 10^4$.

Example 16.4 Through the application of the singular-value decomposition, the problem of designing two-dimensional digital filters (see Sec. 18.6) can be broken down into a problem of designing a set of one-dimensional digital filters [18]. The amplitude responses of the one-dimensional filters so obtained turn out to be quite irregular and, consequently, their design can be accomplished only through the use of optimization methods. The amplitude response of such a filter is specified at 21 frequency points, as in Table 16.5, and $\omega_s = 2$ rad/s. Obtain eighth-order designs using Algorithms 6 and 7 in conjunction with the nonuniform sampling technique of Sec. 16.6 in each case, and compare the results obtained. Assume that $K = 35$.

Table 16.5 Specified amplitude response (Example 16.4)

ω	Gain	ω	Gain	ω	Gain
0.00	1.0770	0.35	0.0304	0.70	0.7950
0.05	0.9863	0.40	0.1665	0.75	0.7950
0.10	0.9866	0.45	0.4402	0.80	0.7950
0.15	0.8428	0.50	0.6231	0.85	0.7950
0.20	0.8436	0.55	0.7471	0.90	0.7950
0.25	0.6466	0.60	0.7950	0.95	0.7950
0.30	0.3955	0.65	0.7950	1.00	0.7950

■ **Solution**

Using an initial point

$$\mathbf{x} = [1\ \ 1\ \ 0.75\ \ 1\ \ 1\ \ 1\ \ 0.75\ \ 1\ \ 1\ \ -1\ \ 0.75\ \ -1\ \ 1\ \ -1\ \ 0.75\ \ -1\ \ 1]^T$$

designs A and B of Table 16.6 were obtained. The progress of each algorithm is illustrated in Table 16.7. The maximum amplitude-response errors in designs A and B were 3.2675×10^{-2} and 3.5292×10^{-2}. Evidently, Algorithm 6 gave a somewhat better design although the amount of computation time was nearly twice that required by Algorithm 7 in terms of function evaluations. The amplitude response achieved in design A is illustrated in Fig. 16.8.

Table 16.6 Coefficients of $H(z)$ (Example 16.4)

	j	a_{0j}	a_{1j}	b_{0j}	b_{1j}
Design A	1	1.002238	2.482808	$-4.716961E-2$	$-9.493371E-1$
	2	$-1.973023E+1$	$1.880026E+1$	$-2.123562E-1$	$3.655407E-1$
	3	1.000000	$-8.468213E-1$	$1.496466E-1$	$1.873191E-2$
	4	1.830361	-2.033032	$6.498825E-1$	-1.155793
	$H_0 = 8.425338E-3$				
Design B	1	$-1.260454E+1$	$3.977791E+1$	$4.318101E-1$	-1.055599
	2	2.377913	-2.490881	$-1.831163E-2$	$-5.216264E-1$
	3	$9.849419E-1$	$-8.325620E-1$	$3.616646E-1$	$-2.230790E-1$
	4	$5.511632E-1$	$-9.021266E-1$	$6.733342E-1$	-1.088983
	$H_0 = 6.418782E-3$				

Table 16.7 Progress of algorithms (Example 16.4)

	Design A		Design B	
k	p	$\Psi(\mathbf{x})$	ξ	$\Psi(\mathbf{x}, \lambda_k, \xi_k)$
1	2	$7.106816E-2$	0.0	$2.893164E-4$
2	4	$3.726389E-2$	$2.229626E-2$	$4.092217E-5$
3	8	$3.329217E-2$	$3.397612E-2$	$5.915443E-7$
4	16	$3.757264E-2$	$3.527249E-2$	$6.184436E-20$
5	32	$3.472619E-2$	$3.174503E-2$	$4.251311E-5$
6	64	$3.359927E-2$	–	–
7	128	$3.304717E-2$	–	–

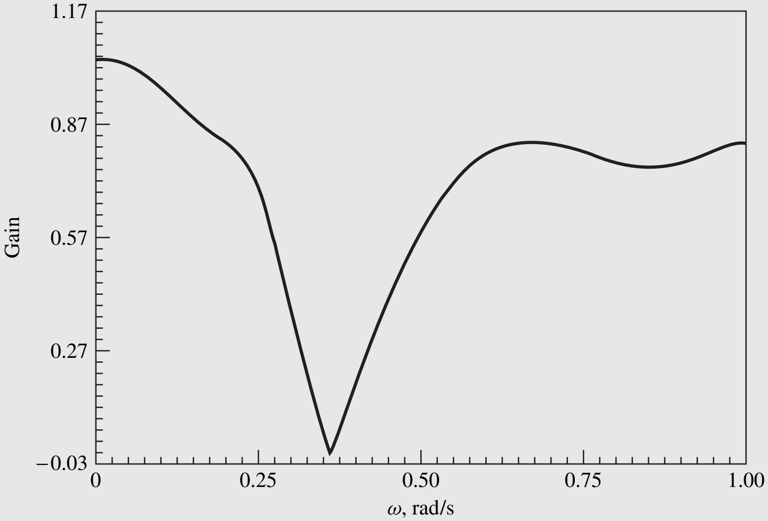


Figure 16.8 Amplitude response of one-dimensional digital filter (Design A, Example 16.4).

16.8 DESIGN OF RECURSIVE DELAY EQUALIZERS

The minimax algorithms described can also be applied for the design of recursive delay equalizers, as will now be demonstrated. Consider a filter characterized by the transfer function

$$H_F(z) = H_0 \prod_{j=1}^{J} \frac{a_{0j} + a_{1j}z + a_{2j}z^2}{b_{0j} + b_{1j}z + b_{2j}z^2} \tag{16.41}$$

The group delay of the filter is given by

$$\tau_F(\omega) = -\frac{d\theta_F(\omega)}{d\omega} \tag{16.42}$$

where

$$\theta_F(\omega) = \arg H_F(e^{j\omega T}) \tag{16.43}$$

From Eqs. (16.41) and (16.42), we can show that

$$\tau_F(\omega) = -T\sum_{j=1}^{J}\frac{\tilde{N}_j(\omega)}{N_j(\omega)} + T\sum_{j=1}^{J}\frac{\tilde{D}_j(\omega)}{D_j(\omega)} \tag{16.44}$$

where

$$\tilde{N}_j(\omega) = a_{2j}^2 - a_{0j}^2 + a_{1j}(a_{2j} - a_{0j})\cos\omega T$$

$$N_j(\omega) = (a_{2j} - a_{0j})^2 + a_{1j}^2 + 2a_{1j}(a_{2j} + a_{0j})\cos\omega T + 4a_{0j}a_{2j}\cos^2\omega T$$

$$\tilde{D}_j(\omega) = b_{2j}^2 - b_{0j}^2 + b_{1j}(b_{2j} - b_{0j})\cos\omega T$$

$$D_j(\omega) = (b_{2j} - b_{0j})^2 + b_{1j}^2 + 2b_{1j}(b_{2j} + b_{0j})\cos\omega T + 4b_{0j}b_{2j}\cos^2\omega T$$

The group delay of the filter can be equalized with respect to a frequency range $\omega_1 \le \omega \le \omega_L$ by connecting an *allpass* delay equalizer in cascade with the filter, as described in Sec. 12.5.1. Let the transfer function of the equalizer be

$$H_E(z) = \prod_{j=1}^{M}\frac{1 + c_{1j}z + c_{0j}z^2}{c_{0j} + c_{1j}z + z^2}$$

The group delay of the equalizer can be obtained as

$$\tau_E(\mathbf{c}, \omega) = -\frac{d\theta_E(\omega)}{d\omega}$$

where

$$\theta_E(\mathbf{c}, \omega) = \arg H_E(e^{j\omega T})$$

Hence

$$\tau_E(\mathbf{c}, \omega) = 2T\sum_{j=1}^{M}\frac{\tilde{C}_j(\omega)}{C_j(\omega)} \tag{16.45}$$

where

$$\tilde{C}_j(\omega) = 1 - c_{0j}^2 + c_{1j}(1 - c_{0j})\cos\omega T$$

$$C_j(\omega) = (1 - c_{0j})^2 + c_{1j}^2 + 2c_{1j}(1 + c_{0j})\cos\omega T + 4c_{0j}\cos^2\omega T$$

and

$$\mathbf{c} = [c_{01}\ c_{11}\ c_{02}\ c_{12}\ \ldots\ c_{1M}]^T$$

The equalizer is stable if and only if the transfer function coefficients satisfy the relations

$$c_{0j} < 1 \qquad c_{1j} - c_{0j} < 1 \qquad c_{1j} + c_{0j} > -1$$

for $j = 1, 2, \ldots, M$ as can be shown by using the Jury-Marden stability criterion (see Sec. 5.3.7). The region of stability in the (c_0, c_1) plane is illustrated in Fig. 16.9. This may be referred to as the feasible region of the parameter space.

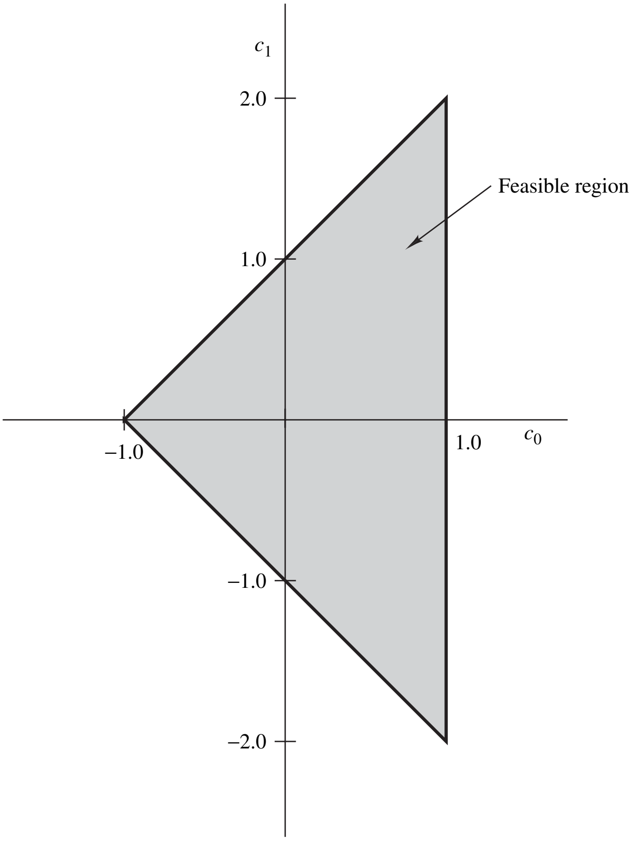


Figure 16.9 Feasible region of (c_0, c_1) plane.

The group delay of the filter-equalizer combination can be expressed as

$$\tau_{FE}(\mathbf{c}, \omega) = \tau_F(\omega) + \tau_E(\mathbf{c}, \omega)$$

where $\tau_F(\omega)$ and $\tau_E(\mathbf{c}, \omega)$ are given by Eqs. (16.44) and (16.45), respectively.

The required equalizer can be designed by solving the optimization problem [13]

$$\underset{\mathbf{x}}{\text{minimize}} \ \widehat{E}(\mathbf{x})$$

where

$$\widehat{E}(\mathbf{x}) = \max_{1 \le i \le K} |e_i(\mathbf{x})|$$

$$e_i(\mathbf{x}) = \frac{1}{T}\tau_{FE}(\mathbf{x}, \omega_i) - \tau_0$$

$$\mathbf{x} = \left[\mathbf{c}^T \ \tau_0\right]^T \qquad \tau_0 = \frac{\tau}{T}$$

and

$$\omega_1 \le \omega \le \omega_L$$

The problem can be readily solved by using Algorithm 6 or 7. As the solution is approached, variable τ_0 approaches the average of τ_{FE}/T with respect to the frequency band of interest, i.e., τ approaches the average of τ_{FE}.

The gradient of $|e_i(\mathbf{x})|$, which is required for the evaluation of $\nabla\Psi(\mathbf{x}, \boldsymbol{\lambda}_k, \xi_k)$, can be obtained, as in Sec. 16.7.2, by using the derivatives of $e_i(\mathbf{x})$, namely,

$$\frac{\partial e_i(\mathbf{x})}{\partial c_{0l}} = \frac{U_{0l} + U_{1l}\cos\omega_i T + U_{2l}\cos^2\omega_i T + U_{3l}\cos^3\omega_i T}{[C_l(\omega_i)]^2}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial c_{1l}} = \frac{V_{0l} + V_{1l}\cos\omega_i T + V_{2l}\cos^2\omega_i T + V_{3l}\cos^3\omega_i T}{[C_l(\omega_i)]^2}$$

$$\frac{\partial e_i(\mathbf{x})}{\partial \tau_0} = -1$$

for $l = 1, 2, \ldots, M$ and $i = 1, 2, \ldots, K$, where

$$U_{0l} = 4\left[(1 - c_{0l})^2 - c_{0l}c_{1l}^2\right] \qquad U_{1l} = -2c_{1l}\left(1 + 6c_{0l} + c_{0l}^2 + c_{1l}^2\right)$$

$$U_{2l} = -8\left(1 + c_{0l}^2 + c_{1l}^2\right) \qquad U_{3l} = -8c_{1l}$$

$$V_{0l} = -4c_{1l}(1 - c_{0l})(1 + c_{0l}) \qquad V_{1l} = -2(1 - c_{0l})\left(1 + 6c_{0l} + c_{0l}^2 + c_{1l}^2\right)$$

$$V_{2l} = 0 \qquad V_{3l} = 8(1 - c_{0l})c_{0l}$$

The quality of an equalizer is inversely related to the maximum variation of τ_{FE} over the frequency band of interest. A measure that can be used to assess the quality of an equalizer design can, therefore, be defined as

$$Q = \frac{100(\widehat{\tau}_{FE} - \breve{\tau}_{FE})}{2\tilde{\tau}_{FE}} \tag{16.46}$$

where

$$\widehat{\tau}_{FE} = \max_{\omega_1 \leq \omega \leq \omega_L} \tau_{FE}$$

$$\breve{\tau}_{FE} = \min_{\omega_1 \leq \omega \leq \omega_L} \tau_{FE}$$

and

$$\tilde{\tau}_{FE} = \frac{1}{2}(\widehat{\tau}_{FE} + \breve{\tau}_{FE}) \tag{16.47}$$

Alternatively, from Eqs. (16.46) and (16.47))

$$Q = \frac{100(\widehat{\tau}_{FE} - \breve{\tau}_{FE})}{(\widehat{\tau}_{FE} + \breve{\tau}_{FE})} \tag{16.48}$$

As in the design of recursive filters, the application of Algorithm 6 or 7 for the design of equalizers may yield an unstable design. While it is possible to restore stability in such a design by replacing poles that are outside the unit circle of the z plane by their reciprocals, the group-delay characteristic of the equalizer will be changed and the resulting design will not be useful. A brute force approach to overcome this problem is to carry out several designs using different starting points, and then select the best design from the set of stable designs. An alternative and more methodical approach, which was found to give good results, is based on the following algorithm:

Algorithm 8: Design of equalizers

1. Compute $\tilde{\tau}_F = (\widehat{\tau}_F + \breve{\tau}_F)/2$, where $\widehat{\tau}_F$ and $\breve{\tau}_F$ are the maximum and minimum of the filter group delay, respectively. Assume a 1-section equalizer, and set $j = 1$ and $\tau_{01} = (1 + k_1)\tilde{\tau}_F/T$, where k_1 is a constant in the range $0 \leq k_1 \leq 0.5$. Carry out designs using points 1 to 8 in Table 16.8 for the initialization of the equalizer coefficients until a stable design is obtained; let the coefficients of the stable design be $\bar{c}_{01}$ and $\bar{c}_{11}$. Compute $\tilde{\tau}_{FE1}$ using Eq. (16.47).

Table 16.8 Initialization points in the feasible region of the (c_0, c_1) plane

No.	Point	No.	Point	No.	Point
1	(0.3, 0.3)	1A	(0.25, 0.50)	1B	(0.50, 0.25)
2	(0.7, 0.7)	2A	(0.50, 0.75)	2B	(0.75, 0.50)
3	(0.7, 1.3)	3A	(0.50, 1.25)	3B	(0.75, 1.50)
4	(−0.3, 0.3)	4A	(−0.25, 0.50)	4B	(−0.50, 0.25)
5	(0.3, −0.3)	5A	(0.25, −0.50)	5B	(0.50, −0.25)
6	(0.7, −0.7)	6A	(0.50, −0.75)	6B	(0.75, −0.50)
7	(0.7, −1.3)	7A	(0.50, −1.25)	7B	(0.75, −1.50)
8	(−0.3, −0.3)	8A	(−0.25, −0.50)	8B	(−0.50, −0.25)

2. *a.* Increase the number of equalizer sections to two; set $j = j + 1$ and $\tau_{02} = \tilde{\tau}_{FE1}/T$.[5]

b. Carry out designs using point $(\bar{c}_{01}, \bar{c}_{11})$ for the initialization of the first section and each of the points

$$P_{12} = [(1 - \varepsilon_1)\bar{c}_{01}, (1 - \varepsilon_1)\bar{c}_{11}]$$

$$P_{22} = [(1 + \varepsilon_1)\bar{c}_{01}, (1 - \varepsilon_1)\bar{c}_{11}]$$

$$P_{32} = [(1 + \varepsilon_1)\bar{c}_{01}, (1 + \varepsilon_1)\bar{c}_{11}]$$

$$P_{42} = [(1 - \varepsilon_1)\bar{c}_{01}, (1 + \varepsilon_1)\bar{c}_{11}]$$

in turn for the initialization of the second section (ε_1 is a small positive constant).

c. Compute parameter Q using Eq. (16.48).

d. If the design obtained is successful, i.e., it is stable and has a Q which is significantly lower than that of the 1-section design, compute $\tilde{\tau}_{FE2}$ and continue with step 3; otherwise, change ε_1 and repeat from step 2*b*.

3. *a.* Increase the number of equalizer sections by one. Set $j = j + 1$ and $\tau_{0j} = \tilde{\tau}_{FE(j-1)}/T$, and carry out designs using the most recent successful design for the initialization of sections $1, 2, \ldots, j-1$ and point

$$P_{0j} = \left[\frac{1}{2}(\widehat{c}_{0(j-1)} + \breve{c}_{0(j-1)}), \frac{1}{2}(\widehat{c}_{1(j-1)} + \breve{c}_{1(j-1)})\right]$$

for the initialization of the jth section where $\widehat{c}_{0(j-1)}$ and $\breve{c}_{0(j-1)}$ are the largest and smallest c_0 coefficients and $\widehat{c}_{1(j-1)}$ and $\breve{c}_{1(j-1)}$ are the largest and smallest c_1 coefficients in the most recent successful design.

b. If the design obtained in step 3*a* is unsuccessful, carry out designs using the most recent successful design for the initialization of sections $1, 2, \ldots, j - 1$, and each of the points

$$P_{1j} = (\breve{c}_{0(j-1)}, \breve{c}_{1(j-1)})$$

$$P_{2j} = (\widehat{c}_{0(j-1)}, \breve{c}_{1(j-1)})$$

$$P_{3j} = (\widehat{c}_{0(j-1)}, \widehat{c}_{1(j-1)})$$

$$P_{4j} = (\breve{c}_{0(j-1)}, \widehat{c}_{1(j-1)})$$

in turn for the initialization of the jth section. If a successful design is obtained, compute τ_{FEj} and proceed to step 4; otherwise, stop.

4. Compute Q; if $Q \le Q_{\max}$, stop; otherwise, go to step 3*a*. ●

Extensive experimentation with Algorithm 8 has shown that for a given filter the solution points (c_{0j}, c_{1j}) tend to form a cluster in the (c_0, c_1) plane. Hence, once a stable 1-section design is obtained in step 1, the general domain of a multisection stable design is located. Consequently, as new sections are added in steps 2 and 3 one by one, a sequence of progressively improved stable designs are obtained. The logarithm of Q tends to decrease almost linearly with the number of equalizer

[5] *The amount of computation can be reduced by using $\tilde{\tau}_{FEj}/T$ instead of $\tilde{\tau}_{FE(j-1)}/T$ for τ_{0j} in steps 2 and 3; this modification can be readily incorporated in the algorithm by including the jth equalizer section in the calculation of $\tilde{\tau}_{FE}$ using the initial coefficient values for the jth section.*

sections at a rate that depends on the selectivity and passband width of the filter. In some examples, Q was found to reach a lower bound at some value less than 5 percent but the cause has not been identified.

The optimizations required in steps 1 to 3 can, in principle, be carried out by using either Algorithm 6 or Algorithm 7. As in the design of recursive filters, Algorithm 7 tends to be much more efficient, while Algorithm 6 tends to yield better local minima (see Example 16.4). The advantages of the two algorithms can be combined by using Algorithm 6 in step 1, where a better design is highly desirable, and Algorithm 7 in steps 2 and 3, where computational efficiency is more important. Should Algorithm 7 fail to give a successful design in step 2 or 3, Algorithm 6 can be tried as an alternative.

At the solution, parameter τ_0 tends to approach the average of τ_{FE}/T. A fairly good estimate of this quantity for the 1-section design, which can be used to initialize τ_{01}, is obtained by letting $k_1 = 0.50$ in step 1. This value of k_1 was found to give good results.

For lowpass and highpass filters, points (c_{0j}, c_{1j}) tend to form clusters in the fourth and first quadrant of the feasible region, respectively. Hence, only points 5 to 8 of Table 16.8 need be tried for lowpass filters and only points 1 to 4 need be tried for highpass filters. In the unlikely situation where none of these points gives a solution, points 1A to 8A and 1B to 8B of Table 16.8 may be tried.

For filters with moderate or high selectivity, the value of ε_1 should be of the order of 0.01 or less; on the other hand, if the selectivity of the filter is low, a value as high as 0.1 may be necessary.

In steps $2b$ and $3b$, a rectangular domain is established in the parameter space, which encloses points (c_{0i}, c_{1i}) for $i = 1, 2, \ldots j - 1$, and each of the corner points P_{1j} to P_{4j} is used for the initialization of the jth section. Occasionally, one or two of these points may be located outside the feasible region of the parameter space and should not be used.

$Q_{\max}$ in step 4 is the maximum allowable value of Q for the application at hand. If the number of sections is sufficient to reduce Q below $Q_{\max}$, the algorithm is terminated.

Example 16.5 The coefficients in Table 16.9 represent an elliptic highpass filter satisfying the following specifications:

- Passband ripple A_p: 0.5 dB
- Minimum stopband attenuation A_a: 50 dB
- Passband edge ω_p: 0.75 rad/s
- Stopband edge ω_a: 0.64 rad/s
- Sampling frequency ω_s: 2.0 rad/s

Table 16.9 Coefficients of $H_F(z)$ (Example 16.5)

j	a_{0j}	a_{1j}	a_{2j}	b_{0j}	b_{1j}	b_{2j}
1	-1.0	1.0	0.0	$7.022673E-1$	1.0	0.0
2	1.0	$1.765666E-2$	1.0	$6.452156E-1$	1.351877	1.0
3	1.0	$7.880299E-1$	1.0	$8.893343E-1$	1.320853	1.0

$H_0 = 1.033262E-2$

Design a delay equalizer that will reduce the Q of the filter-equalizer combination to a value less than 1.0 percent.

■ **Solution**

The design was carried out using Algorithm 6 for step 1 and Algorithm 7 for steps 2 and 3, along with the nonuniform variable sampling technique of Sec. 16.6 in each case. In order to achieve the desired degree of flatness in the delay characteristic, it was found necessary to increase the number of equalizer sections to five. The progress of the design is illustrated in Table 16.10. The transfer-function coefficients for the successive equalizers are given in Table 16.11. The delay characteristics of the filter-equalizer combination with no equalizer, a 2-section equalizer, and a 5-section equalizer are illustrated in Fig. 16.10.

Table 16.10 Progress of design (Example 16.5)

j	$(c_{0j},\ c_{1j})$	$\tilde{\tau}_{FEj}/T$	Q
0	—	11.76	66.38
1	(0.3, 0.3)	16.21	34.41
2	(0.6097, 1.482)	20.16	19.79
3	(0.7582, 1.610)	26.47	8.05
4	(0.7690, 1.579)	32.85	2.72
5	(0.7803, 1.567)	39.08	0.83

Table 16.11 Coefficients of $H_E(z)$ (Example 16.5)

Sections	j	c_{0j}	c_{1j}
1	1	$6.158622E-1$	1.496936
2	1	$7.549257E-1$	1.715040
	2	$7.614137E-1$	1.504945
3	1	$7.552047E-1$	1.726392
	2	$7.826521E-1$	1.431156
	3	$7.668681E-1$	1.634637
4	1	$7.703755E-1$	1.681226
	2	$7.671458E-1$	1.551901
	3	$7.945904E-1$	1.391108
	4	$7.659007E-1$	1.741710
5	1	$7.593030E-1$	1.692920
	2	$7.602346E-1$	1.483221
	3	$7.985100E-1$	1.365123
	4	$7.551977E-1$	1.732325
	5	$7.607868E-1$	1.610131

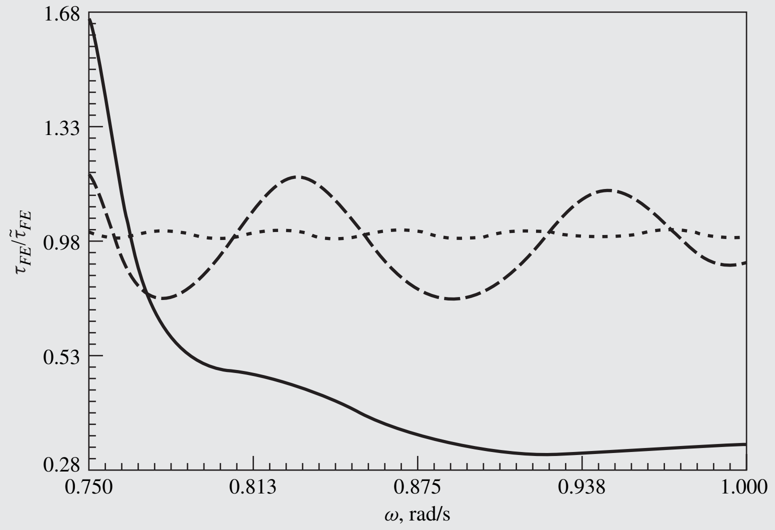


Figure 16.10 Delay characteristics of filter-equalizer combination (Example 16.5): ——- no equalizer, — — — 2-section equalizer, · · · · · · 5-section equalizer.

Example 16.6 The coefficients in Table 16.12 represent an elliptic bandpass filter satisfying the following specifications:

- Maximum Passband ripple A_p: 1.0 dB
- Minimum stopband attenuation A_a: 40 dB
- Low passband edge ω_{p1}: 0.3 rad/s
- High passband edge ω_{p2}: 0.5 rad/s
- Low stopband edge ω_{a1}: 0.2 rad/s
- High stopband edge ω_{a2}: 0.7 rad/s
- Sampling frequency ω_s: 2.0 rad/s

Design a delay equalizer that will reduce the Q of the filter-equalizer combination to a value less than 2.0 percent.

Table 16.12 Coefficients of $H_F(z)$ (Example 16.6)

j	a_{0j}	a_{1j}	a_{2j}	b_{0j}	b_{1j}	b_{2j}
1	−1.0	0.0	1.0	$7.105797E-1$	$-5.558010E-1$	1.0
2	1.0	−1.676442	1.0	$8.610875E-1$	$-1.312559E-2$	1.0
3	1.0	$9.873155E-1$	1.0	$8.856595E-1$	−1.099622	1.0
$H_0 = 2.602536E-2$						

■ **Solution**

The design was carried out as in Example 16.5. In order to achieve the desired degree of flatness in the delay characteristic, it was found necessary to increase the number of equalizer sections to four. The progress of the design is illustrated in Table 16.13. The transfer-function coefficients for the successive equalizers are given in Table 16.14. The delay characteristics of the filter-equalizer combination with no equalizer, a 2-section equalizer, and a 4-section equalizer are illustrated in Fig. 16.11.

Table 16.13 Progress of design (Example 16.6)

j	(c_{0j}, c_{1j})	$\tilde{\tau}_{FE_j}/T$	Q
0	–	11.95	52.22
1	(0.3, 0.3)	16.38	27.78
2	(0.7332, −0.5297)	23.19	9.36
3	(0.7783, −0.7739)	29.13	3.31
4	(0.7469, −0.1775)	32.44	1.96

Table 16.14 Coefficients of $H_E(z)$ (Example 16.6)

Sections	j	c_{0j}	c_{1j}
1	1	$7.405829E-1$	$-5.245374E-1$
2	1	$7.814228E-1$	$-7.738830E-1$
	2	$7.783450E-1$	$-2.892557E-1$
3	1	$7.621367E-1$	$-1.775013E-1$
	2	$7.468771E-1$	$-4.845243E-1$
	3	$7.925267E-1$	$-8.497135E-1$
4	1	$7.748554E-1$	$-2.501149E-1$
	2	$7.393927E-1$	$-5.571269E-1$
	3	$7.930800E-1$	$-8.709362E-1$
	4	$5.866017E-1$	$1.566691E-1$

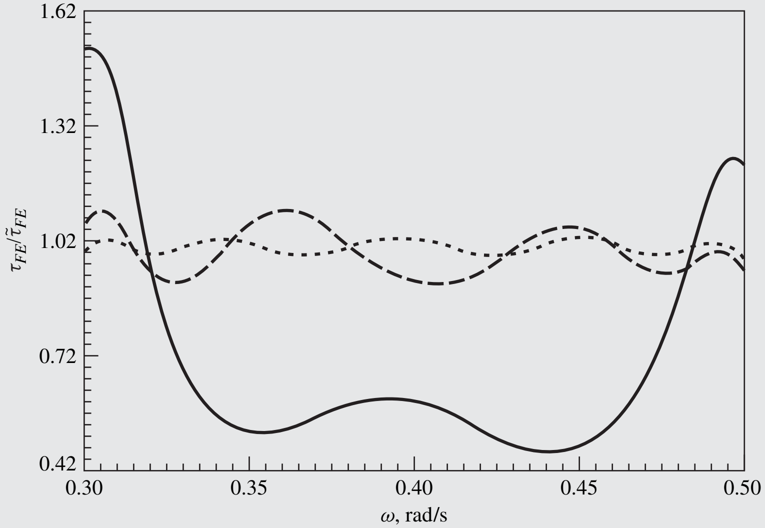


Figure 16.11 Delay characteristics of filter-equalizer combination (Example 16.6): ——- no equalizer,— — — 2-section equalizer, · · · · · · 4-section equalizer.

The mechanism by which Algorithm 8 leads to a series of progressively improved *stable* designs is illustrated in Figs. 16.12 and 16.13. As can be seen in Figs. 16.12*a* and 16.13*a*, the error surface for the 1-section equalizer has a well-defined depression in the feasible region of the parameter space which tends to be maintained as the number of equalizer sections is increased; see, for example, the error surface for the 4-section equalizer illustrated in Figs. 16.12*b* and 16.13*b* In effect, a natural barrier is formed around the solution that assures the stability of successive equalizer sections.

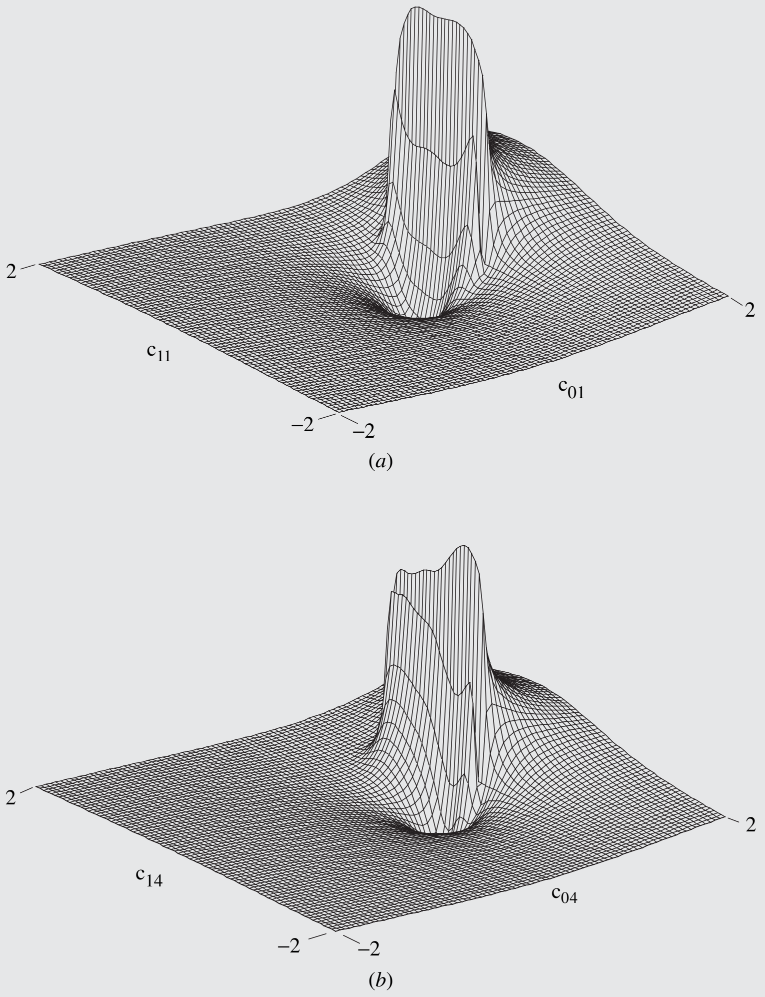
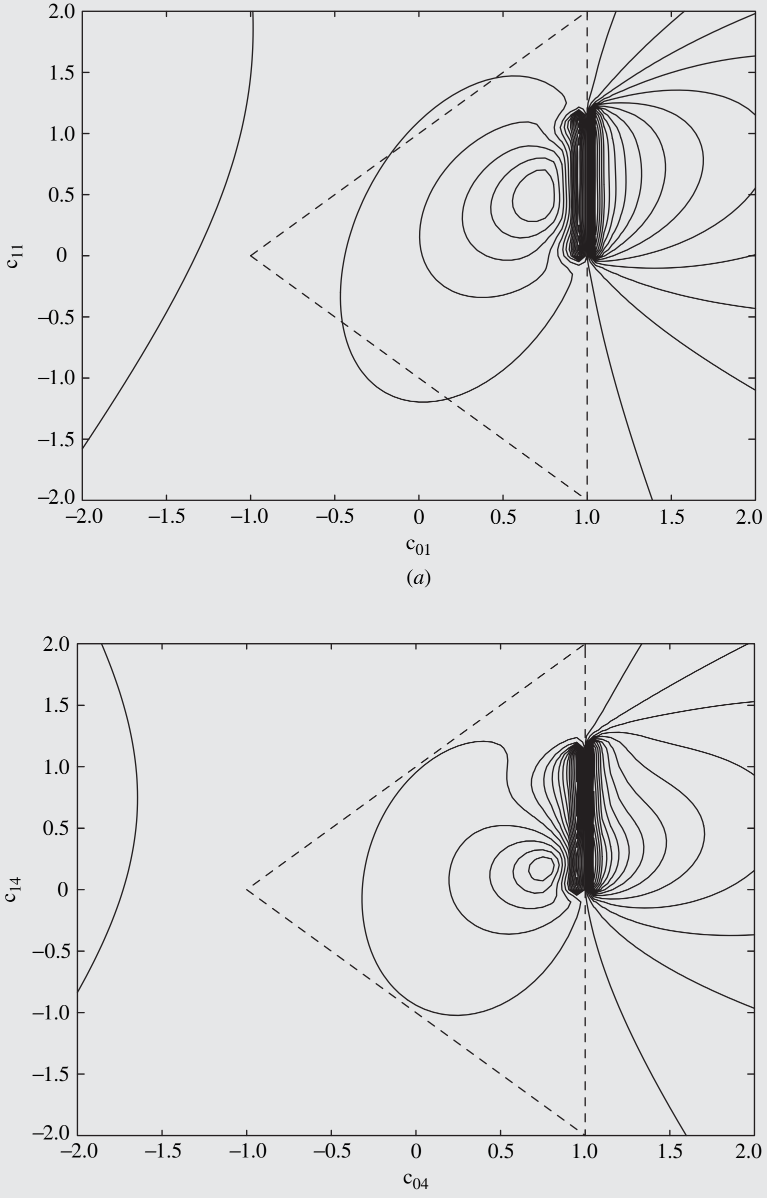


Figure 16.12 3-D plots of error function (Example 16.6): (*a*) 1-section equalizer, (*b*) 4-section equalizer (the coefficients of the first three sections have been assumed to have the optimized values achieved in the 3-section equalizer).



(*a*)

(*b*)

Figure 16.13 Contour plots of error function (Example 16.6): (*a*) 1-section equalizer, (*b*) 4-section equalizer (the coefficients of the first three sections have been assumed to have the optimized values achieved in the 3-section equalizer).

REFERENCES

[1] K. Steiglitz, "Computer-aided design of recursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. 18, pp. 123–129, June 1970.

[2] A. G. Deczky, "Synthesis of recursive digital filters using the minimum p-error criterion," *IEEE Trans. Audio Electroacoust.*, vol. 20, pp. 257–263, Oct. 1972.

[3] J. W. Bandler and B. L. Bardakjian, "Least pth optimization of recursive digital filters," *IEEE Trans. Audio Electroacoust.*, vol. 21, pp. 460–470, Oct. 1973.

[4] C. Charalambous, "Minimax design of recursive digital filters," *Computer Aided Design*, vol. 6, pp. 73–81, Apr. 1974.

[5] C. Charalambous, "Minimax optimization of recursive digital filters using recent minimax results," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 23, pp. 333–345, Aug. 1975.

[6] R. Fletcher, *Practical Methods of Optimization, Unconstrained Optimization*, vol. 1, New York: Wiley, 1980. (See also R. Fletcher, *Practical Methods of Optimization*, 2nd ed., New York: Wiley, 1990.)

[7] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed., Reading, MA: Addison-Wesley, 1984.

[8] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, New York: Academic, 1981.

[9] D. M. Himmelblau, *Applied Nonlinear Programming*, New York: McGraw-Hill, 1972.

[10] B. D. Bunday, *Basic Optimisation Methods*, London: Edward Arnold, 1984.

[11] M. Al-Baali and R. Fletcher, "An efficient line search for nonlinear least squares," *J. Opt. Theo. Applns.*, vol. 48, pp. 359–378, 1986.

[12] C. Charalambous, "A unified review of optimization," *IEEE Trans. Microwave Theory and Techniques*, vol. MTT-22, pp. 289–300, Mar. 1974.

[13] C. Charalambous and A. Antoniou, "Equalisation of recursive digital filters," *Proc. Inst. Elect. Eng.*, Part G, vol. 127, pp. 219–225, Oct. 1980.

[14] C. Charalambous, "Design of 2-dimensional circularly-symmetric digital filters," *Proc. Inst. Elect. Eng.*, part G, vol. 129, pp. 47–54, Apr. 1982.

[15] C. Charalambous, "Acceleration of the least pth algorithm for minimax optimization with engineering applications," *Mathematical Programming*, vol. 17, pp. 270–297, 1979.

[16] A. Antoniou, "Improved minimax optimisation algorithms and their application in the design of recursive digital filters," *Proc. Inst. Elect. Eng.*, part G, vol. 138, pp. 724–730, Dec. 1991.

[17] A. Antoniou, M. Degano, and C. Charalambous, "Compensation for the effects of the D/A convertor in recursive digital filters," *Proc. Inst. Elect. Eng.*, part G, vol. 129, pp. 273–279, Dec. 1982.

[18] A. Antoniou and W.-S. Lu, "Design of two-dimensional digital filters by using the singular value decomposition," *IEEE Trans. Circuits Syst.*, vol. 34, pp. 1191–1198, Oct. 1987.

ADDITIONAL REFERENCES

Charalambous, C., "A new approach to multicriterion optimization problem and its application to the design of 1-D digital filters," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 773–784, June 1989.

Chottera A. T. and G. A. Jullien, "A linear programming approach to recursive digital filter design with linear phase," *IEEE Trans. Circuits Syst.*, vol. 29, pp. 139–149, Mar. 1982.

Lang, M.C., "Least-squares design of IIR filters with prescribed magnitude and phase response and a pole radius constraint," *IEEE Trans. Signal Processing*, vol. 48, pp. 3109–3126, Nov. 2001.

Lim, Y. C., J. H. Lee, C. K. Chen, and R.-H. Yang, "A weighted least-squares approximation for quasi-equiripple FIR and IIR digital filter design," *IEEE Trans. Signal Processing*, vol. 40, pp. 551–558, Mar. 1992.

Lu, W.-S., S.-C. Pei, and C.-C. Tseng, "A weighted least-squares method for the design of 1-D and 2-D IIR digital filters," *IEEE Trans. Signal Processing*, vol. 46, pp. 1–10, Jan. 1998.

Lu, W.-S. and A. Antoniou, "Design of digital filters and filter banks by optimization: A state of the art review," in *Proc. 2000 European Signal Processing Conference*, vol. 1, pp. 351–354, Tampere, Finland, Sept. 2000.

W.-S. Lu and T. Hinamoto, "Optimal design of IIR digital filters with robust stability using conic-quadratic-programming updates," IEEE Trans. Signal Processing, vol. 51, pp. 1581–1592, June 2003.

PROBLEMS

16.1. The step response $y(t)$ of a digital filter is required to approximate the ideal step response

$$y_0(t) = \begin{cases} t & \text{for } 0 \le t < 2 \\ 2 & \text{for } 2 \le t < 3 \\ -t+5 & \text{for } 3 \le t < 4 \\ 1 & \text{for } 4 \le t < 5 \end{cases}$$

where $t = nT$. Formulate a least-squares objective function for the solution of the problem.

16.2. The quantity y should in theory be related to parameters x_1 and x_2 by a formula of the form

$$y = \frac{a_0 x_1}{1 + a_1 x_1 + a_2 x_2}$$

In a specific experiment, the data in Table P16.2 were collected. Construct an objective function that can be used to find coefficients a_0 to a_2 such that the maximum difference between y and y_0 is minimized.

Table P16.2

x_1	x_2	y_0
1	1	0.1265
2	1	0.2193
1	2	0.0075
2	2	0.1262
0.1	0	0.1859

16.3. Obtain an objective function that can be used to find approximate values of x_1 and x_2 that satisfy the relations

$$x_1 = x_2^2 - 3\log x_1$$
$$x_2 = \left(2x_1^2 - 5x_1 + 1\right)/x_1$$

16.4. The so-called *Rosenbrock function*

$$f(\mathbf{x}) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$$

represents a highly nonlinear surface in the shape of a narrow curved falling valley. It is often used to test the ability of algorithms to maneuver around curved valleys. Show that $f(\mathbf{x})$ has a minimum at point $[1\ 1]^T$.

16.5. Find and classify the stationary points of function

$$f(\mathbf{x}) = x_1^2 - x_2^2 + x_3^2 - 2x_1x_3 - x_2x_3 + 4x_1 + 12$$

16.6. (*a*) Show that the function

$$f(\mathbf{x}) = 2x_1^2 - 2x_1x_2 + x_2^2 + 2x_1 - 2x_2$$

has a minimum.

(*b*) Find the minimum of the function using Algorithm 1 with $\mathbf{x}_0 = [0\ 0]^T$ as initial point.

16.7. Repeat Prob. 16.6 for the function

$$f(\mathbf{x}) = x_1^2 + 2x_2^2 + 4x_1 + 4x_2$$

16.8. Show that at point $[1 + \varepsilon_1, 1 + \varepsilon_2]$, where $|\varepsilon_1| \ll 1$ and $|\varepsilon_2| \ll 1$, the Rosenbrock function given in Prob. 16.4 can be approximated by a quadratic function.

16.9. Derive Eq. (16.27).

16.10. Derive Eq. (16.28).

16.11. (*a*) Show that the estimate of α_0 used in step 2*d* of Algorithm 5 (practical quasi-Newton algorithm), can be derived by using Eq. (16.27).

(*b*) Justify the use of $\alpha_0 = 1$, if the estimate in part (*a*) is unreasonable.

16.12. (*a*) Show that the extrapolation in step 6 of Algorithm 5 can yield a negative $\breve{\alpha}_0$.

(*b*) How is the problem in part (*a*) avoided.

(*c*) In the interpolation as well as extrapolation routines of Algorithm 5, the search is aborted if the number of function evaluations exceeds a certain maximum. Why is it advisable to include such a termination criterion in optimization algorithms?

(*d*) Explain the purpose of steps 4*c*, 4*d*, and 6*c* in Algorithm 5.

16.13. Write a computer program for Algorithm 5 using the BFGS updating formula.

(*a*) Use the program to obtain a least-squares solution of the problem in Prob. 16.1.

(*b*) Repeat part (*a*) for the problem in Prob. 16.2.

(*c*) Repeat part (*a*) for the problem in Prob. 16.3.

16.14. (*a*) Use the program in Prob. 16.13 to minimize Rosenbrock's function given in Prob. 16.4.

(*b*) Repeat part (*a*) for the function

$$f(\mathbf{x}) = 100[(x_3 - 10\theta)^2 + (r - 1)^2] + x_3^2$$

where

$$\theta = \begin{cases} \dfrac{1}{2\pi}\tan^{-1}\left(\dfrac{x_2}{x_1}\right) & \text{for } x_1 > 0 \\ 0.25 & \text{for } x_1 = 0 \\ 0.5 + \dfrac{1}{2\pi}\tan^{-1}\left(\dfrac{x_2}{x_1}\right) & \text{for } x_1 < 0 \end{cases}$$

and

$$r = \sqrt{(x_1^2 + x_2^2)}$$

16.15. Use the computer program of Prob. 16.13 to minimize the following functions

(*a*)

$$f(\mathbf{x}) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 100(x_1 - x_4)^4$$

(*b*)

$$f(\mathbf{x}) = \sum_{i=2}^{5} \left[100\left(x_i - x_{i-1}^2\right)^2 + (1 - x_i)^2\right]$$

16.16. Write a computer program for Algorithm 5 using the DFP updating formula.

(*a*) Use the program to obtain a least-squares solution of the problem in Prob. 16.1.

(*b*) Repeat part (*a*) for the problem in Prob. 16.2.

(*c*) Repeat part (*a*) for the problem in Prob. 16.3.

16.17. (*a*) Run the computer program in Prob. 16.13 (BFGS version) with the function of Prob. 16.4 using 10 different initial points and find the average number of function evaluations. Count one function evaluation for each evaluation of $f(\mathbf{x})$ and one for each partial derivative of $f(\mathbf{x})$. Repeat this process using the program in Prob. 16.16 (DFP version). Compare the results obtained.

(*b*) Repeat part (*a*) with the function in part (*b*) of Prob. 16.15.

16.18. Replace the line search in the computer program of Prob. 16.13 (BFGS version) by the line search described in Ref. [11] (see also second edition of Fletcher [6], pp. 34–35).

(*a*) Use the program to obtain a least-squares solution of the problem in Prob. 16.1.

(*b*) Repeat part (*a*) for the problem in Prob. 16.2.

(*c*) Repeat part (*a*) for the problem in Prob. 16.3.

16.19. (*a*) Run the computer program in Prob. 16.18 (different line search) with the function in Prob. 16.4 using 10 different initial points and find the average number of function evaluations. Count one function evaluation for each evaluation of $f(\mathbf{x})$ and one for each partial derivative of $f(\mathbf{x})$. Repeat this process using the program in Prob. 16.13. Compare the results obtained.

(*b*) Repeat part (*a*) with the function in part (*b*) of Prob. 16.15.

16.20. Write a computer program for Algorithm 6 (least-pth minimax algorithm) using the BFGS updating formula.

(*a*) Use the program to obtain a minimax solution of the problem in Prob. 16.1.

(*b*) Repeat part (*a*) for the problem in Prob. 16.2.

(*c*) Repeat part (*a*) for the problem in Prob. 16.3.

(*d*) Compare the minimax solutions obtained in parts (*a*) to (*c*) with the corresponding least-squares solutions obtained in Prob. 16.13 with respect to the minimum error achieved and the amount of computation required.

16.21. Write a computer program for Algorithm 7 (Charalambous minimax algorithm) using the BFGS updating formula.

(*a*) Use the program to obtain a minimax solution of the problem in Prob. 16.1.

(*b*) Repeat part (*a*) for the problem in Prob. 16.2.

(*c*) Repeat part (*a*) for the problem in Prob. 16.3.

(*d*) Compare the minimax solutions obtained with Algorithm 7 with the corresponding solutions obtained with Algorithm 6 in Prob. 16.20.

16.22. (*a*) Using the computer program in Prob. 16.20, design a fourth-order highpass digital filter with passband and stopband edges of 3.5 and 1.5 rad/s, respectively. The sampling frequency is required to be 10 rad/s.

(*b*) Repeat part (*a*) with the program in Prob. 16.21.

(*c*) Design an elliptic highpass filter that would satisfy the same specifications as the filter obtained in part (*a*) (i.e., same maximum passband ripple, minimum stopband attenuation, passband and stopband edges) using the closed-form method of Chap. 12.

(*d*) Compare the designs obtained in parts (*a*) and (*b*), and (*a*) and (*c*).

16.23. (*a*) Using the computer program in Prob. 16.20, design a sixth-order bandpass digital filter with passband edges 900 and 1100 rad/s, and stopband edges 800 and 1200 rad/s. The sampling frequency is required to be 6000 rad/s.

(*b*) Repeat part (*a*) with the program in Prob. 16.21.

(*c*) Design an elliptic bandpass filter that would satisfy the same specifications as the filter obtained in part (*a*) (i.e., same maximum passband ripple, minimum stopband attenuation, passband and stopband edges) using the closed-form method of Chap. 12.

(*d*) Compare the designs obtained in parts (*a*) and (*b*), and (*a*) and (*c*).

16.24. Modify the computer program in Prob. 16.20 using the nonuniform sampling technique in Sec. 16.6. Then use the program obtained to design a sixth-order lowpass filter with the idealized piecewise-continuous amplitude response

$$M_0(\omega) = \begin{cases} 0.545455\omega + 1.0 & \text{for } 0 \le \omega \le 0.55 \\ 0.75\omega - 0.45 & \text{for } 0.60 \le \omega \le 1.0 \end{cases}$$

The frequency range 0.55 to 0.60 rad/s represents a transition band in which the amplitude response is undefined. The sampling frequency is 2.0 rad/s.

16.25. Modify the computer program in Prob. 16.21 using the nonuniform sampling technique in Sec. 16.6.

(*a*) Use the program obtained to design the lowpass filter described in Prob. 16.24.

(*b*) Compare the results with those obtained in Prob. 16.24.

16.26. (*a*) Using the method in Chap. 12, design an elliptic bandpass filter that would satisfy the following specifications:

- Maximum passband ripple A_p : 1.0 dB
- Minimum stopband attenuation A_a: 30 dB
- Low passband edge ω_{p1}: 0.15 rad/s
- High passband edge ω_{p2}: 0.20 rad/s
- Low stopband edge ω_{a1}: 0.40 rad/s
- High stopband edge ω_{a2}: 0.50 rad/s
- Sampling frequency ω_s: 2.0 rad/s

(*b*) Design a bandpass filter with the same order and band edges as for the filter in part (*a*) by using the program in Prob. 16.24. The ideal passband and stopband gains can be assumed to be 1.0 and 0.01, respectively.

(*c*) The bandpass filter in part (*b*) is to be used in cascade with a D/A converter. The amplitude response of the D/A converter $\phi(\omega)$ is of the form given in Example 16.3 with $\tau = T$. Using the program in Prob. 16.24, redesign the filter taking into consideration the amplitude response of the D/A converter. Use the coefficients of the filter in part (*a*) for the initialization of the algorithm.

(*d*) Repeat part (*c*) with the program in Prob. 16.25.

16.27. (*a*) The amplitude response of a recursive filter is specified at 21 frequency points as in Table P16.27 and $\omega_s = 2$ rad/s. Obtain an eighth-order design using the program in Prob. 16.24.

Table P16.27 Specified amplitude response

ω	Gain	ω	Gain	ω	Gain
0.00	0.9135	0.35	0.6232	0.70	0.5681
0.05	0.7080	0.40	0.7986	0.75	0.5725
0.10	0.6939	0.45	0.8186	0.80	0.5758
0.15	0.4062	0.50	0.7808	0.85	0.5780
0.20	0.3872	0.55	0.6598	0.90	0.5794
0.25	0.0070	0.60	0.5544	0.95	0.5802
0.30	0.2888	0.65	0.5623	1.00	0.5805

(*b*) Design the filter in part (*a*) using the program in Prob. 16.25.

16.28. (*a*) Using the method in Chap. 12, design an elliptic lowpass filter that would satisfy the following specifications:

- Passband ripple A_p: 1.0 dB
- Minimum stopband attenuation A_a: 40 dB
- Passband edge ω_p: 0.10 rad/s
- Stopband edge ω_a: 0.15 rad/s
- Sampling frequency ω_s: 2.0 rad/s

(*b*) Design a delay equalizer that will reduce the Q of the filter-equalizer combination to a value less than 5.0 percent using the computer program in Prob. 16.24.

(*c*) Repeat part (*b*) using the program in Prob. 16.25.

16.29. (*a*) Using the method in Chap. 12, design an elliptic highpass filter that would satisfy the following specifications:

- Passband ripple A_p: 1.0 dB
- Minimum stopband attenuation A_a: 60 dB
- Passband edge ω_p: 0.90 rad/s
- Stopband edge ω_a: 0.85 rad/s
- Sampling frequency ω_s: 2.0 rad/s

(*b*) Design a delay equalizer that will reduce the Q of the filter-equalizer combination to a value less than 5.0 percent using the computer program in Prob. 16.24.

(*c*) Repeat part (*b*) using the program in Prob. 16.25.

16.30. (*a*) Design a delay equalizer for the filter in Prob. 16.26*a* that will reduce the Q of the filter-equalizer combination to a value less than 5.0 percent using the computer program in Prob. 16.24.

(*b*) Repeat part (*a*) using the program in Prob. 16.25.

This page intentionally left blank

CHAPTER 17

WAVE DIGITAL FILTERS

17.1 INTRODUCTION

The effects of coefficient quantization in digital filters can be kept small by realizing the transfer function in terms of a cascade or parallel arrangement of second-order filter sections of the type described in Sec. 14.4.

Alternative *low-sensitivity* structures can be obtained by using a synthesis methodology advanced by Fettweis [1, 3] and developed further by Fettweis, Sedlmeyer, and others [4, 6]. In this approach, an *equally terminated LC* filter that would satisfy prescribed specifications is first designed. Then by replacing analog elements by appropriate digital realizations, the LC filter is transformed into a topologically equivalent digital filter. The synthesis is based on the wave network characterization, and for this reason the resulting structures are referred to collectively as *wave digital filters*. The low sensitivity comes about because equally terminated LC filters are inherently low-sensitivity structures.

The chapter begins with a qualitative justification of the low-sensitivity attribute of equally terminated LC filters. It then proceeds to the derivation of digital realizations for the various analog elements and then to the design and analysis details of structures based on LC lattice and ladder filters. An important advantage of wave digital filters is that they can be designed to be free of zero-input and overflow limit-cycle oscillations. This property of wave digital filters is demonstrated in Sec. 17.10. Later, in Sec. 17.12, an alternative cascade realization is developed by using the concept of the generalized-immittance converter. This approach yields filters with improved in-band signal-to-noise ratio and leads to digital biquadratic multiple-output realizations that are amenable to very-large-scale integrated (VLSI) circuit implementation.

17.2 SENSITIVITY CONSIDERATIONS

An equally terminated LC filter like that in Fig. 17.1a can be characterized in terms of its *insertion loss* which is defined as

$$L(\omega) = 10 \log \frac{P_m(\omega)}{P(\omega)}$$

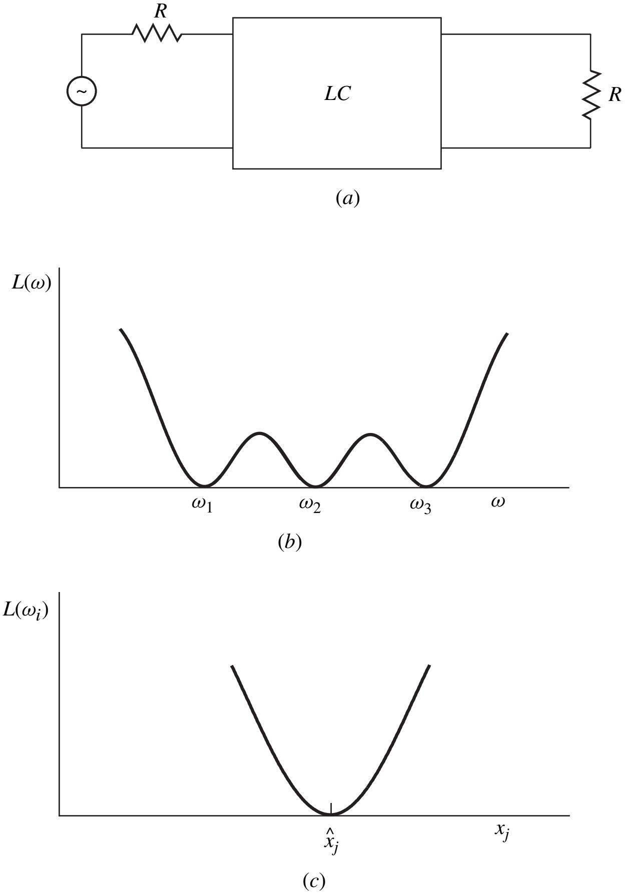


Figure 17.1 (a) Equally terminated LC filter, (b) equiripple loss characteristic, (c) $L(\omega_i)$ versus x_j.

$P(\omega)$ is the actual output power and $P_m(\omega)$ is the maximum output power under perfect matching conditions. Since the LC 2-port network (or 2-port for short) is a passive lossless network, $P(\omega) \leq P_m(\omega)$ and thus $L(\omega) \geq 0$. Now let us assume that $L(\omega_i) = 0$ for $i = 1, 2, \ldots,$ as depicted in Fig. 17.1*b*, as in the case of an elliptic characteristic. At frequency ω_i the filter delivers the maximum available power, and if any lossless element x_j is increased above or decreased below its nominal value $\hat{x}_j$, $L(\omega_i)$ must necessarily increase above zero as illustrated in Fig. 17.1*c*. Clearly

$$\lim_{x_j \to \hat{x}_j} \frac{\Delta L(\omega_i)}{\Delta x_j} = \frac{dL(\omega_i)}{dx_j} = 0$$

for $i = 1, 2, \ldots$ and $j = 1, 2, \ldots$ independently of the order of the filter [7]. Consequently, the sensitivity of the passband loss to element variations in equally terminated LC elliptic filters is inherently low. Therefore, by simulating filters of this type digitally, low-sensitivity digital-filter structures can be obtained.

In Sec. 11.6, we have shown that a discrete-time transfer function $H_D(z)$ can be readily obtained by applying the bilinear transformation

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right)$$

to a continuous-time transfer function $H_A(s)$. It would, therefore, appear that one should be able to obtain low-sensitivity digital structures by simply applying the bilinear transformation to signal flow graphs of equally terminated LC filters. Unfortunately, this approach leads to flow graphs with *delay-free loops* which are not realizable (see Sec. 4.8.1). The problem is due to the fact that any realization of the bilinear transformation has a direct delay-free path between input and output.

The problem of delay-free loops can be avoided in a somewhat circuitous manner by using the wave network characterization, as will be demonstrated in the next and subsequent sections.

17.3 WAVE NETWORK CHARACTERIZATION

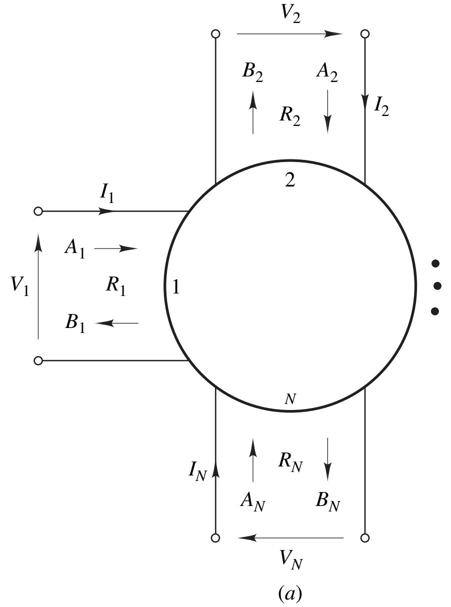
An analog N-port network of the form shown in Fig. 17.2*a* can be represented by the set of equations

$$\left.\begin{aligned} A_k &= V_k + I_k R_k \\ B_k &= V_k - I_k R_k \end{aligned}\right\} \quad \text{for } k = 1, 2, \ldots, N \tag{17.1}$$

The parameters A_k and B_k are referred to as the *incident* and *reflected wave quantities*, respectively, and R_k is the *port resistance*. The representation can be either in the time or frequency domain except that lower-case symbols are usually used for the time-domain representation.

If two N-ports are cascaded as in Fig. 17.2*b*, it is necessary to assign

$$R_j = R_k$$



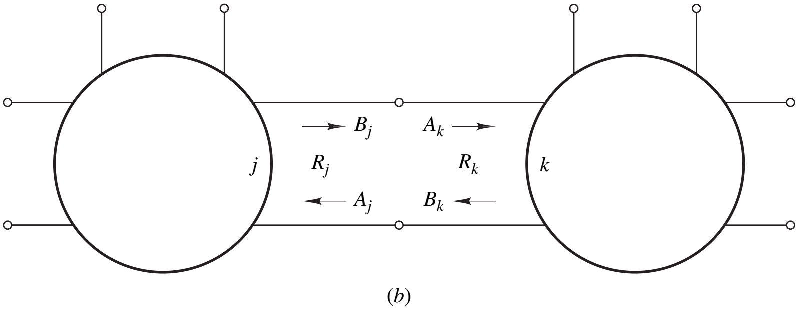


Figure 17.2 (*a*) Analog N-port network, (*b*) interconnected N-ports.

so that

$$A_k = B_j \qquad \text{and} \qquad A_j = B_k$$

i.e., a common resistance must be assigned to two interconnected ports to maintain continuity in the wave flow. Otherwise, R_k can be assigned on an arbitrary basis.

An LC filter can be regarded as a conglomerate of a number of impedances (R, sL, or $1/sC$), a source (voltage or current), and a number of 3-port series and parallel wire interconnections as illustrated in Fig. 7.3. By realizing these elements digitally and subsequently replacing analog elements in LC filters by their digital realizations, wave digital filters can be synthesized.

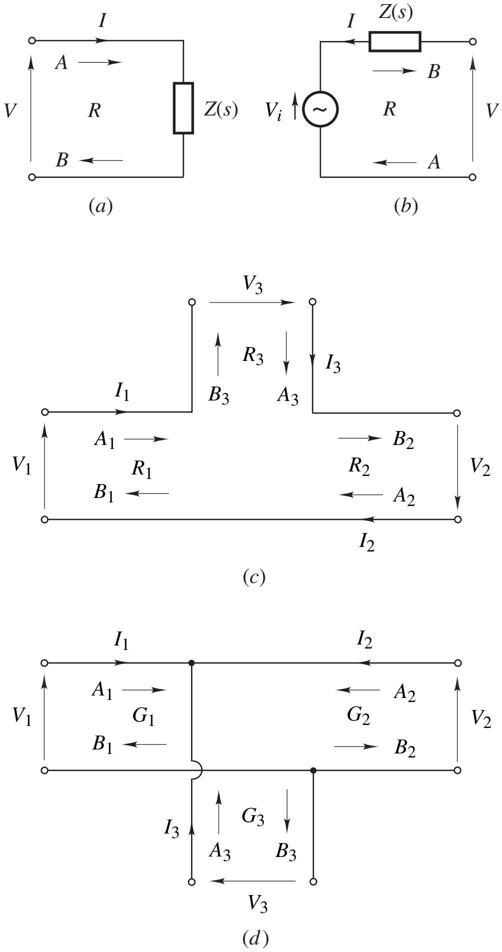


Figure 17.3 (*a*) Impedance, (*b*) voltage source, (*c*) series wire interconnection, (*d*) parallel wire interconnection.

17.4 ELEMENT REALIZATIONS

Digital realizations for analog elements can be derived by using the following procedure:

1. Represent the element in terms of the wave characterization.
2. Eliminate variables V_k, I_k, and s, using the loop and node equations and the bilinear transformation

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right)$$

3. Express the reflected wave quantities as functions of the incident wave quantities.
4. Realize the resulting set of equations using unit delays, adders, inverters, and multipliers.

17.4.1 Impedances

Consider an *impedance*

$$Z(s) = s^{\lambda} R_x \tag{17.2}$$

where R_x is a positive constant and $\lambda = -1$ for a capacitance, $\lambda = 0$ for a resistance, and $\lambda = 1$ for an inductance. From Eq. (17.1) and Fig. 17.3*a*

$$A = V + IR \tag{17.3a}$$

$$B = V - IR \tag{17.3b}$$

where

$$V = IZ(s)$$

and if the bilinear transformation is applied to continuous-time variables A, B, V, and I, that is,

$$Q\Big|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)} \rightarrow Q \tag{17.4}$$

for $Q = A$, B, V, and I, we obtain

$$B = f(z)A \tag{17.5}$$

where

$$f(z) = \left.\frac{Z(s) - R}{Z(s) + R}\right|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)} \tag{17.6}$$

Now on choosing

$$R = \left(\frac{2}{T}\right)^{\lambda} R_x \tag{17.7}$$

and then using Eqs. (17.2) and (17.6) we have

$$f(z) = \begin{cases} z^{-1} & \text{for } \lambda = -1 \\ 0 & \text{for } \lambda = 0 \\ -z^{-1} & \text{for } \lambda = 1 \end{cases}$$

Hence Eq. (17.5) results in the element realizations of Fig. 17.4; that is, a resistance translates into a digital sink, a capacitance into a unit delay, and an inductance into a unit delay in cascade with an inverter.

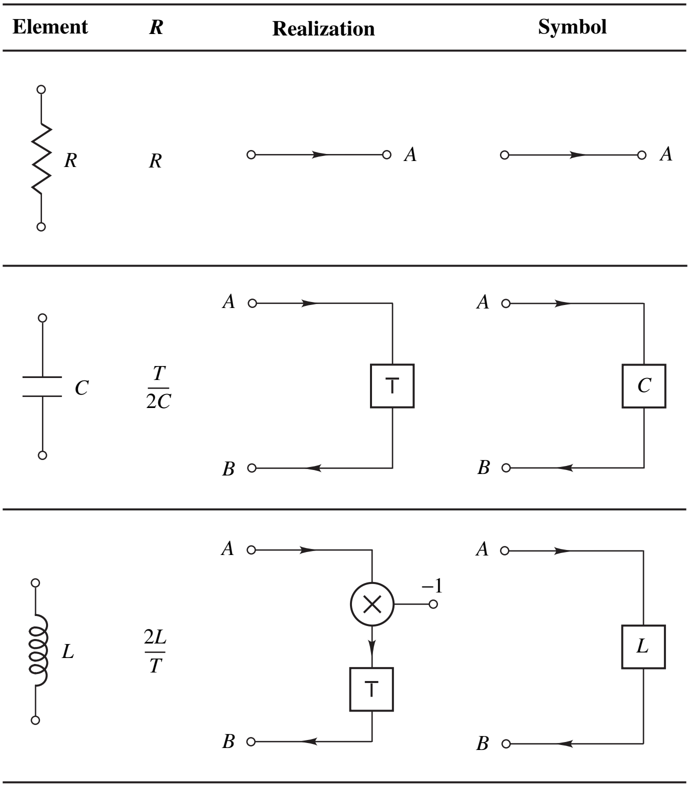


Figure 17.4 Digital realization of impedances.

17.4.2 Voltage Sources

For the *voltage source* of Fig. 17.3*b*, where

$$Z(s) = s^{\lambda} R_x$$

we can write

$$A = V + IR \qquad B = V - IR \qquad V = IZ(s) + V_i$$

and on eliminating V, I, and s we deduce

$$B = f_1(z)V_i + f_2(z)A \tag{17.8}$$

where $f_1(z) = \left.\dfrac{2R}{R + Z(s)}\right|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}$ and $f_2(z) = \left.\dfrac{Z(s) - R}{Z(s) + R}\right|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}$

With

$$R = \left(\frac{2}{T}\right)^{\lambda} R_x$$

$f_1(z)$ and $f_2(z)$ simplify to

$$f_1(z) = \begin{cases} 1 - z^{-1} & \text{for } \lambda = -1 \\ 1 & \text{for } \lambda = 0 \\ 1 + z^{-1} & \text{for } \lambda = 1 \end{cases} \qquad \text{and} \qquad f_2(z) = \begin{cases} z^{-1} & \text{for } \lambda = -1 \\ 0 & \text{for } \lambda = 0 \\ -z^{-1} & \text{for } \lambda = 1 \end{cases}$$

Hence Eq. (17.8) yields realizations for capacitive, resistive, and inductive sources, as depicted in Fig. 17.5.

17.4.3 Series Wire Interconnection

The preceding approach can be readily extended to the realization of *wire interconnections*. For the *series* interconnection of Fig. 17.3*c*

$$I_1 = I_2 = I_3 \qquad V_1 + V_2 + V_3 = 0$$

and on eliminating voltages and currents in Eq. (17.1), we can show that

$$\mathbf{B} = (\mathbf{I} - \mathbf{M_s})\mathbf{A} \tag{17.9}$$

where **I** is the 3 × 3 unity matrix, **A** and **B** are column vectors

$$\mathbf{M_s} = \begin{bmatrix} m_{s1} & m_{s1} & m_{s1} \\ m_{s2} & m_{s2} & m_{s2} \\ m_{s3} & m_{s3} & m_{s3} \end{bmatrix} \qquad \text{where } m_{s3} = 2 - m_{s1} - m_{s2}$$

and

$$m_{sk} = \frac{2R_k}{R_1 + R_2 + R_3} \qquad \text{for } k = 1, 2 \tag{17.10}$$

A realization of Eq. (17.9) is shown in Fig. 17.6*a*. This can be referred to as type S2 adaptor, i.e., series 2-multiplier adaptor.

With R_2 unspecified, one can choose

$$R_2 = R_1 + R_3$$

so that

$$m_{s1} = \frac{R_1}{R_2} \qquad m_{s2} = 1$$

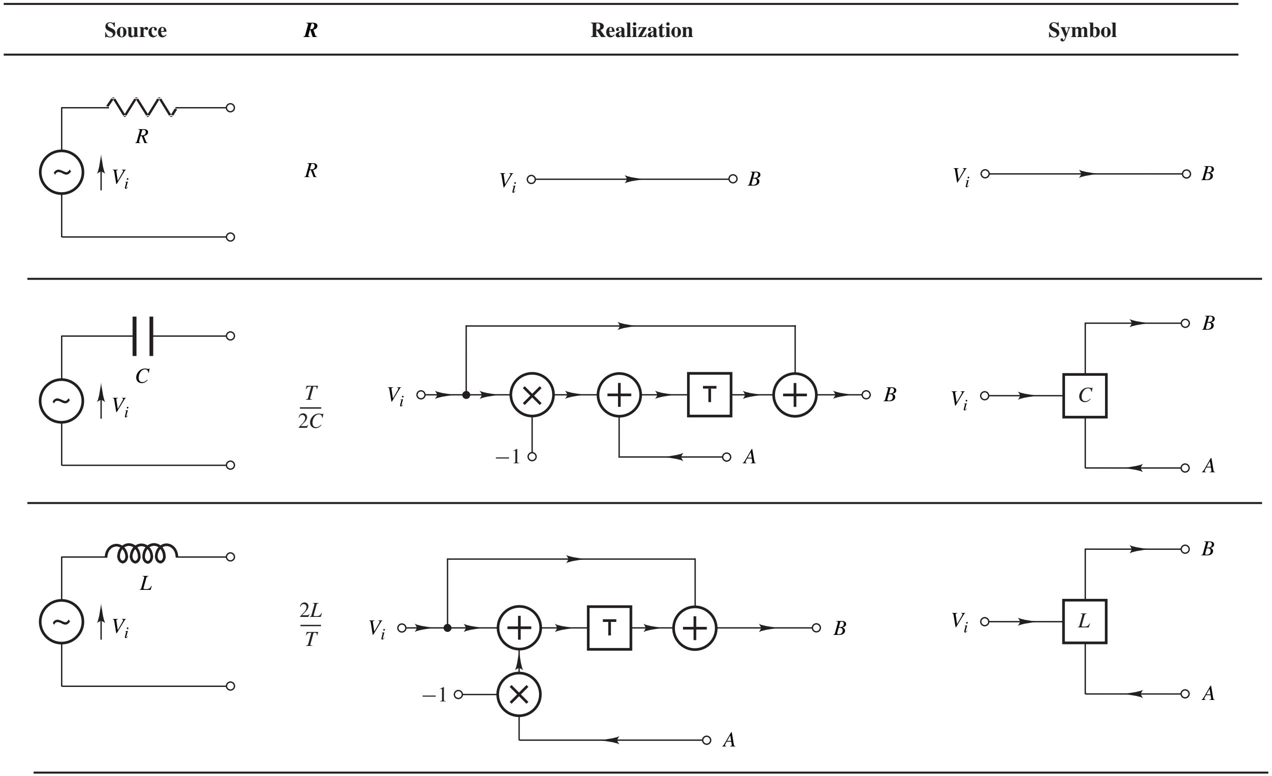


Figure 17.5 Digital realization of voltage sources.

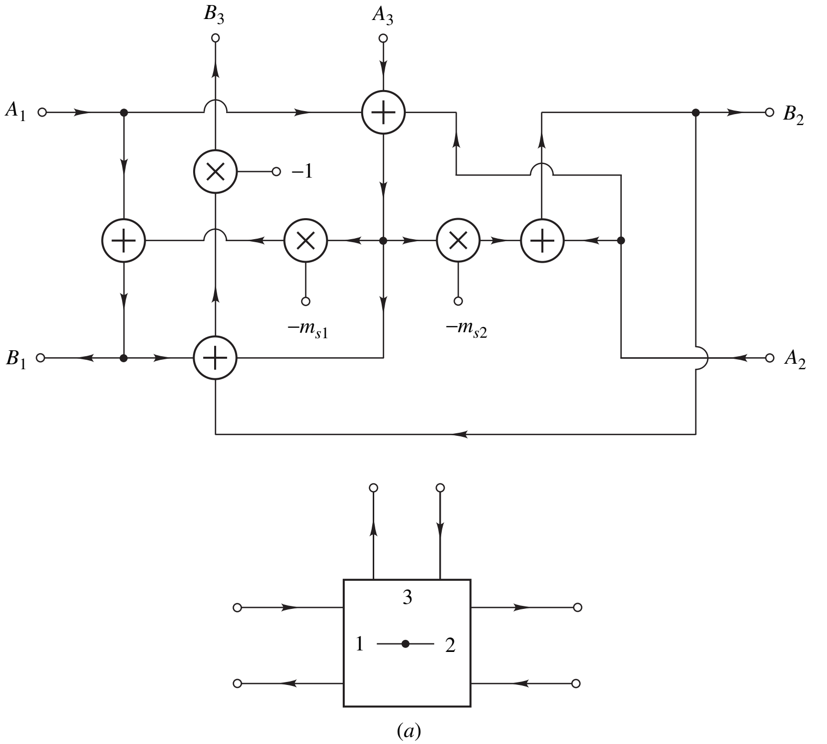


Figure 17.6 Series adaptors: (*a*) Type S2 adaptor.

according to Eq. (17.10). As a consequence, the above adaptor can be simplified to the series 1-multiplier adaptor (type S1) of Fig. 17.6*b*.

17.4.4 Parallel Wire Interconnection

Similarly, for the *parallel* wire interconnection of Fig. 17.3*d*, we have

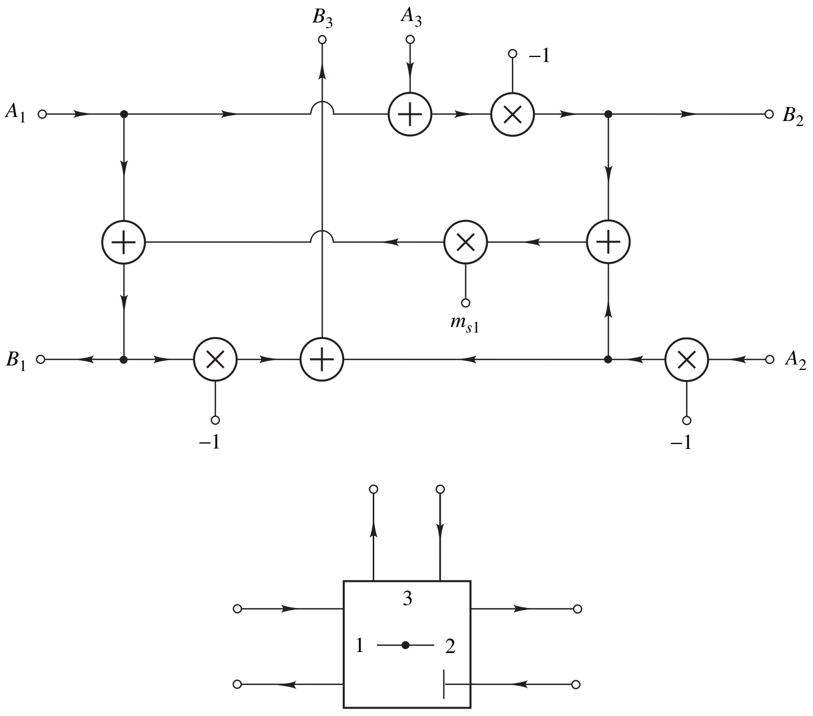
$$V_1 = V_2 = V_3 \qquad \text{and} \qquad I_1 + I_2 + I_3 = 0$$

and from Eq. (17.1)

$$\mathbf{B} = (\mathbf{M}_p - \mathbf{I})A \tag{17.11}$$

where
$$\mathbf{M}_p = \begin{bmatrix} m_{p1} & m_{p2} & m_{p3} \\ m_{p1} & m_{p2} & m_{p3} \\ m_{p1} & m_{p2} & m_{p3} \end{bmatrix} \qquad \text{with } m_{p3} = 2 - m_{p1} - m_{p2}$$

$$m_{pk} = \frac{2G_k}{G_1 + G_2 + G_3} \qquad \text{for } k = 1, 2 \tag{17.12}$$



(*b*)

Figure 17.6 Cont'd Series adaptors: (*b*) Type S1 adaptor.

and G_k is the *port conductance*. A realization of Eq. (17.11), referred to here as type P2 adaptor, is shown in Fig. 17.7*a*. The corresponding 1-multiplier realization (type P1 adaptor) shown in Fig. 17.7*b* is obtained by choosing the conductance at port 2 as

$$G_2 = G_1 + G_3$$

so that

$$m_{p1} = \frac{G_1}{G_2} \qquad m_{p2} = 1$$

Adaptors S2 and P2 are said to be *unconstrained* since their port resistances can be assigned arbitrary values.

17.4.5 2-Port Adaptors

Unconstrained 2-port adaptors can be obtained by modifying series or parallel 3-port adaptors. By letting $A_3 = G_3 = 0$ and deleting the terminal for B_3 in the parallel adaptor of Fig. 17.7*a*, the 2-port

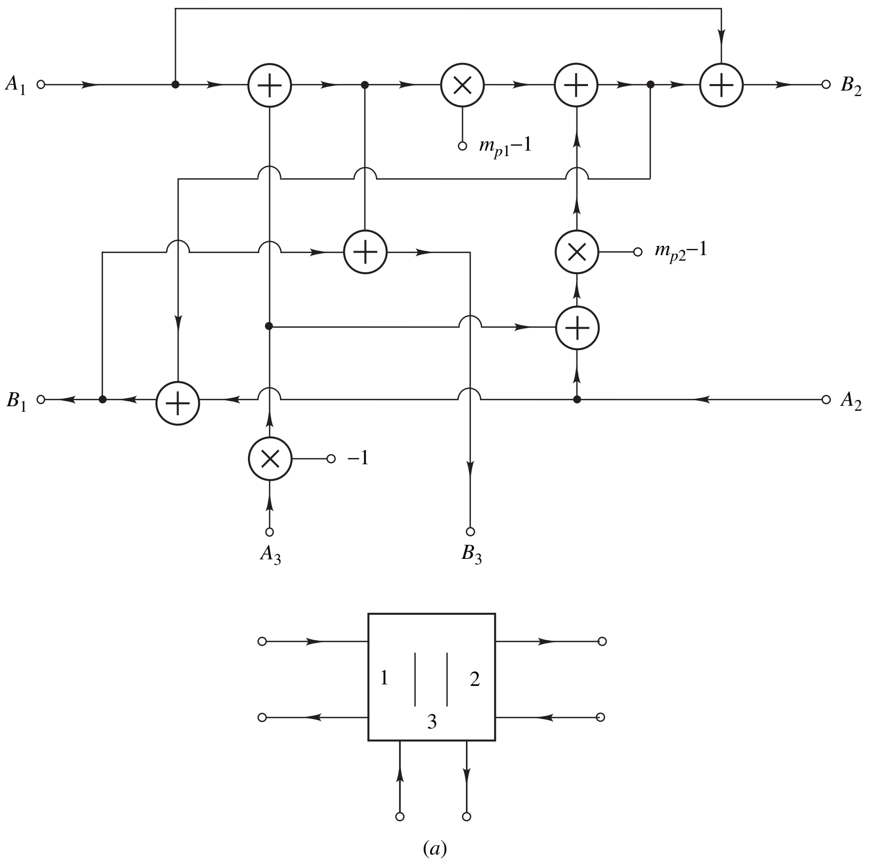


Figure 17.7 Parallel adaptors: (a) Type P2 adaptor.

adaptor depicted in Fig. 17.8a can be obtained. From Eq. (17.12)

$$m_{p1} - 1 = -(m_{p2} - 1)$$

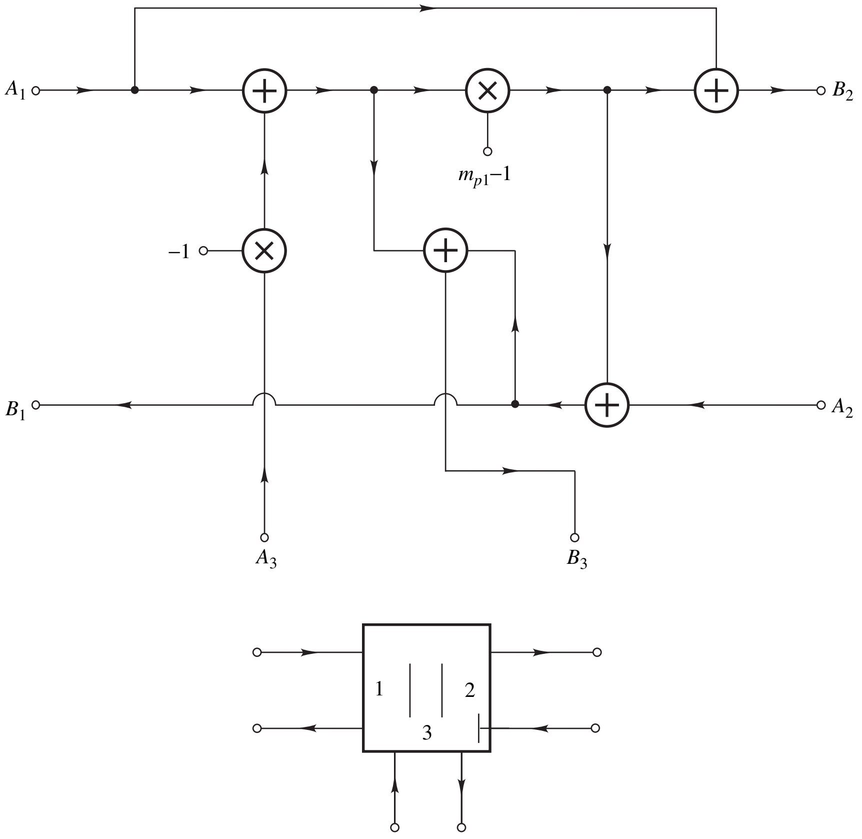
and hence the number of multipliers in the adaptor can be reduced to one, as shown in Fig. 17.8b. The value of multiplier constant μ is given by

$$\mu = m_{p2} - 1 = \frac{G_2 - G_1}{G_2 + G_1}$$

Alternative adaptor configurations can be found in [8, 9].

17.4.6 Transformers

The above principles can be used for the derivation of digital equivalent networks for 2-port devices such as transformers, gyrators, and circulators. For example, the *ideal transformer* of Fig. 17.9a can



(*b*)

Figure 17.7 Cont'd Parallel adaptors: (*b*) Type P1 adaptor.

be represented by the relations

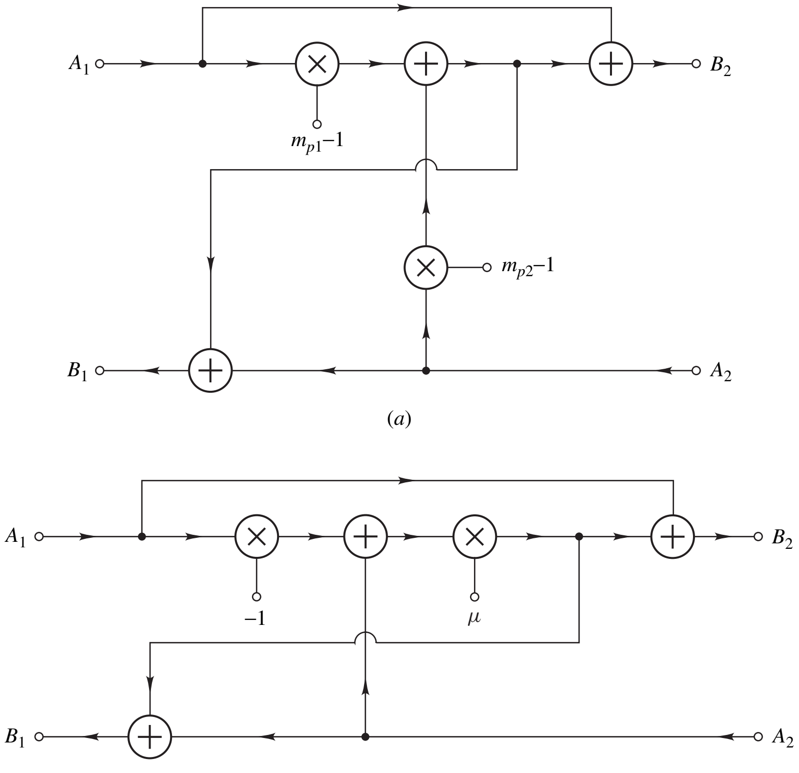
$$V_2 = kV_1 \qquad I_2 = -I_1/k$$

where k is the turns ratio. On assigning resistances R_1 and R_2 to ports 1 and 2, respectively, we can show that

$$\mathbf{B} = \mathbf{M}_T\mathbf{A} \tag{17.13}$$

where

$$\mathbf{M}_T = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$



(*a*)

(*b*)

Figure 17.8 (*a*) 2-port adaptor, (*b*) 1-multiplier version.

with

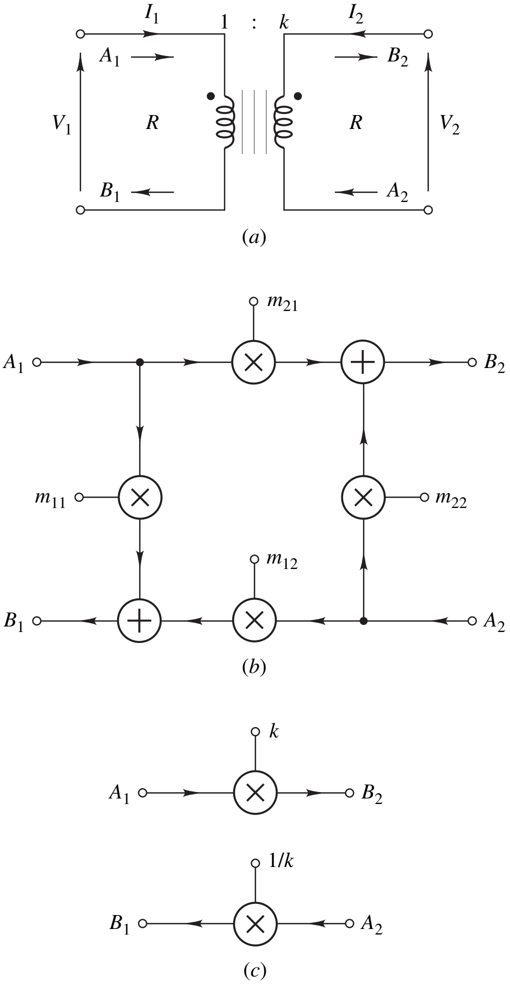
$$m_{11} = \frac{R_2 - k^2 R_1}{R_2 + k^2 R_1} \qquad m_{12} = \frac{2kR_1}{R_2 + k^2 R_1}$$

$$m_{21} = \frac{2kR_2}{R_2 + k^2 R_1} \qquad m_{22} = -\frac{R_2 - k^2 R_1}{R_2 + k^2 R_1}$$

A realization of Eq. (17.13) is shown in Fig. 17.9*b*. If we assign $R_2 = k^2 R_1$, then $m_{11} = 0$, $m_{12} = 1/k$, $m_{21} = k$, and $m_{22} = 0$ and the simplified realization of Fig. 17.9*c* is obtained.

17.4.7 Unit Elements

Another 2-port element which is often quite useful is the so-called *unit element*. This device simulates a transmission line of delay $T/2$ and *characteristic impedance* R, and it has been used extensively in the design of microwave filters. It can be represented by the symbol of Fig. 17.10*a* and is characterized



(*a*)

(*b*)

(*c*)

Figure 17.9 (*a*) Ideal transformer, (*b*) digital realization, (*c*) simplified version.

by the equations

$$V_1 = k_{11}V_2 - k_{12}I_2 \tag{17.14a}$$

$$I_1 = k_{21}V_2 - k_{22}I_2 \tag{17.14b}$$

where

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} = \frac{1}{\sqrt{1-\hat{s}^{\,2}}} \begin{bmatrix} 1 & \hat{s}R \\ \hat{s}/R & 1 \end{bmatrix} \tag{17.15}$$

with $\hat{s} = sT/2$. **K** is said to be the *chain matrix* of the 2-port. On assigning port resistances

$$R_1 = R_2 = R$$

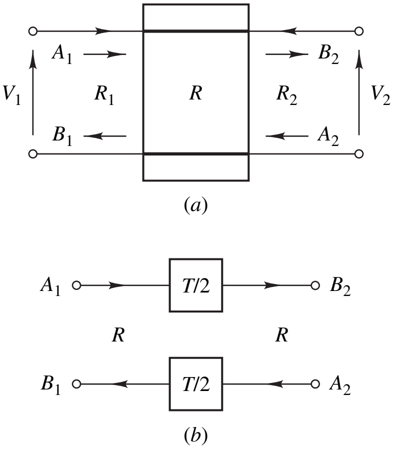


Figure 17.10 (*a*) Unit element, (*b*) digital realization.

and then expressing B_1 and B_2 in terms of A_1 and A_2 using Eqs. (17.1), (17.4), and (17.14), we obtain

$$B_1 = z^{-\frac{1}{2}} A_2$$
$$B_2 = z^{-\frac{1}{2}} A_1$$

Therefore, a digital realization of the unit element can be obtained as shown in Fig. 17.10*b*.

17.4.8 Circulators

Circulators are N-port devices with $N \geq 3$ in which the reflected wave at a given port is equal to the incident wave at the adjacent port, say, in the counterclockwise direction, i.e.,

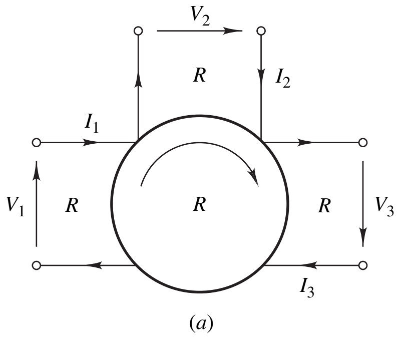
$$B_1 = A_N \qquad B_2 = A_1 \qquad \cdots \qquad B_N = A_{N-1}$$

A 3-port circulator and its digital realization are illustrated in Fig. 17.11*a* and 17.11*b*. Circulators, like unit elements, are used in the design of microwave circuits and filters but, as will be shown in Sec. 17.5, they can also be used in the design of an important class of wave digital filters known as *lattice wave digital filters*.

17.4.9 Resonant Circuits

Wave digital filters are sometimes designed as simple interconnections of *series* or *parallel resonant circuits*. Digital realizations for circuits of this type can be readily obtained by using the principles described so far. For example, a series resonant circuit comprising a capacitor C and an inductor L can be drawn as shown in Fig. 17.12*a* and on assigning the port resistances

$$R_1 = R \qquad R_2 = T/2C \qquad \text{and} \qquad R_3 = 2L/T$$



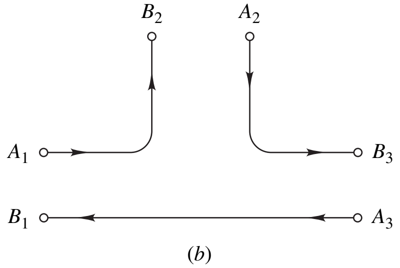
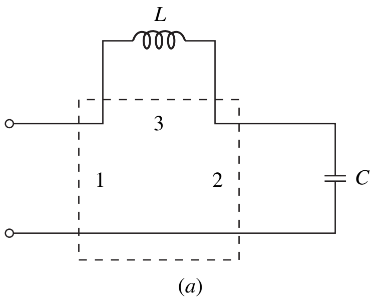


Figure 17.11 (*a*) 3-port circulator, (*b*) digital realization.



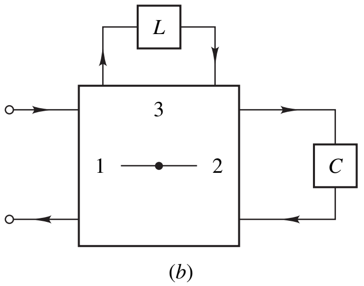


Figure 17.12 (*a*) Series resonant circuit, (*b*) digital realization using a 3-port adaptor.

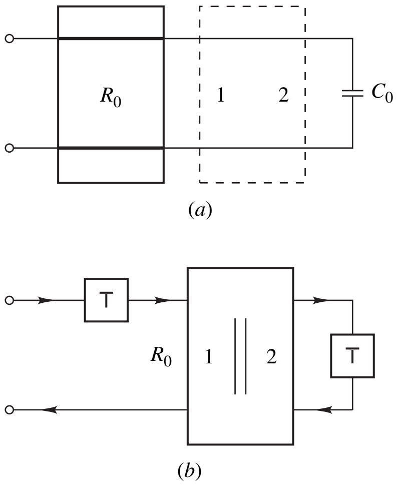


Figure 17.13 (*a*) Series resonant circuit using a unit element and a capacitor, (*b*) digital realization.

the digital realization of Fig. 17.12*b* is obtained.

An alternative realization of a series resonant circuit can be derived by using a unit element terminated by a capacitor, as depicted in Fig. 17.13*a*. From Eqs. (17.14) and (17.15), the input impedance of the circuit can be deduced as

$$Z_i = \frac{1}{K_3}\left(sK_1 + \frac{1}{sK_2}\right) \tag{17.16}$$

where

$$K_1 = \frac{1}{2}R_0T \qquad K_2 = C_0 \qquad \text{and} \qquad K_3 = 1 + \frac{T}{2R_0C_0}$$

Now if we assign

$$R_0 = \frac{4LC + T^2}{2TC} \qquad \text{and} \qquad C_0 = \frac{4LC^2}{4LC + T^2}$$

Eq. (17.16) assumes the form

$$Z_i = sL + \frac{1}{sC}$$

i.e., the network realizes a series resonant circuit. On assigning conductances $G_1 = 1/R_0$ and $G_2 = 2C_0/T$ to ports 1 and 2 of the wire interconnection and then replacing the analog elements by their digital counterparts, the digital realization of Fig. 17.13*b* is obtained; the multiplier constant of the 2-port adaptor can be expressed as

$$\mu = \frac{G_2 - G_1}{G_2 + G_1} = \frac{4LC - T^2}{4LC + T^2}$$

17.4.10 Realizability Constraint

Digital networks containing delay-free loops are said to be *unrealizable* because certain node signals in such networks cannot be computed (see Sec. 4.8.1). The networks derived so far do not contain delay-free loops. However, such can arise if adaptor ports with direct paths are interconnected. The only adaptor port without direct paths is port 2 in adaptors S1 and P1, as can be seen in Figs. 17.6*b* and 17.7*b*. Therefore, for the sake of realizability, every direct connection between adaptor ports *must necessarily involve port 2 of either an S1 or a P1 adaptor.*

17.5 LATTICE WAVE DIGITAL FILTERS

With digital realizations available for the various analog elements, several families of wave digital filters can be obtained by converting classical lattice, ladder, microwave, active, and other types of analog filters into digital filters [1], [4–6], [9–12]. In this and the next section, we examine the realization of *lattice* and *ladder* digital filters.

The class of wave lattice digital filters is based on the lattice network shown in Fig. 17.14*a* where Z_A and Z_B are usually canonic, lossless, LC impedances (see Refs. [1–3] of Chap. 10).

17.5.1 Analysis

Like any other 2-port network, the lattice network of Fig. 17.14*a* can be represented by the wave characterization of Eq. (17.1). Applying Kirchhoff's laws to the network yields

$$I_1 = \frac{1}{Z_B}(V_1 + V_2) - I_2$$

$$I_2 = \frac{2V_2 + (Z_A - Z_B)I_1}{Z_A + Z_B}$$

and on assigning port resistances $R_1 = R_2 = R$ and then eliminating I_1, I_2, V_1, and V_2 in Eq. (17.1), we obtain

$$\mathbf{B} = \mathbf{SA} \tag{17.17}$$

where

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \tag{17.18}$$

with

$$S_{11} = S_{22} = (S_B + S_A)/2 \tag{17.19a}$$

$$S_{12} = S_{21} = (S_B - S_A)/2 \tag{17.19b}$$

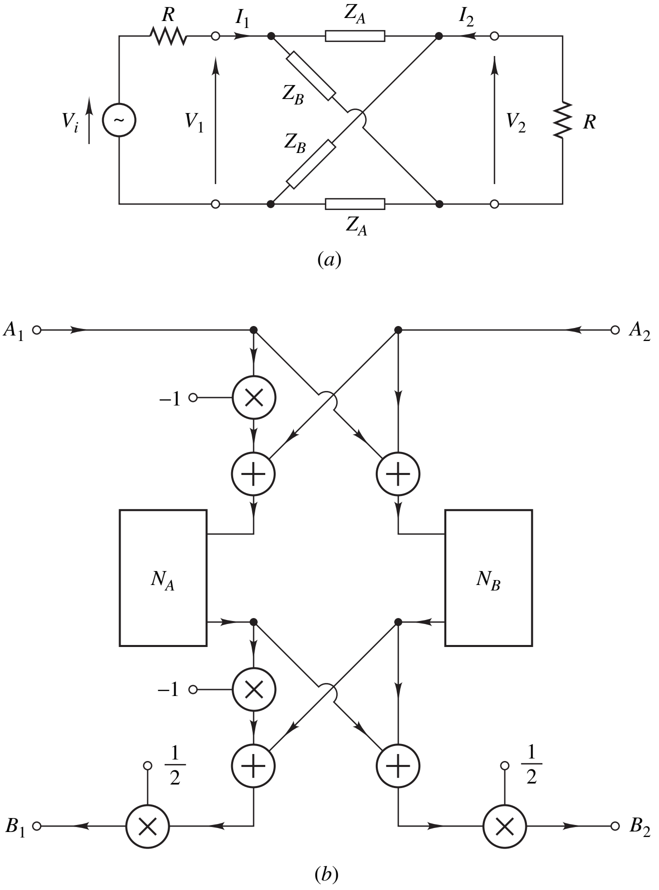


Figure 17.14 (*a*) Analog lattice network, (*b*) alternative realization based on wave characterization.

$$S_A = \frac{Z_A - R}{Z_A + R} \tag{17.20a}$$

$$S_B = \frac{Z_B - R}{Z_B + R} \tag{17.20b}$$

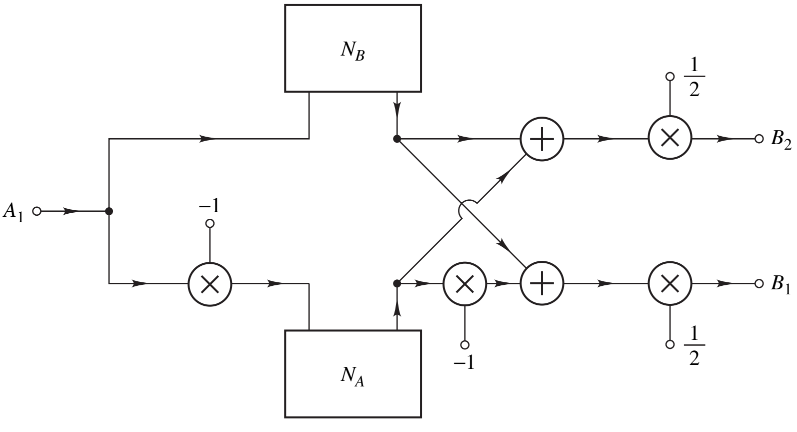
17.5.2 Alternative Lattice Configuration

From Eqs. (17.17)–(17.19), we can write

$$B_1 = \frac{1}{2}[S_A(A_1 - A_2) + S_B(A_1 + A_2)]$$

$$B_2 = \frac{1}{2}[S_A(A_2 - A_1) + S_B(A_1 + A_2)]$$

Thus an alternative analog realization for the lattice network can be derived, as depicted in Fig. 17.14*b*, where networks N_A and N_B realize S_A and S_B, respectively. With port 2 of the network in Fig. 17.14*a*



(*c*)

Figure 17.14 Cont'd (*c*) Simplified configuration.

terminated by a resistance R, we have $A_2 = 0$ and hence, the simplified configuration of Fig. 17.14*c* is obtained. The transfer functions of the configuration from input to outputs B_1 and B_2 are given by

$$\tilde{H}_A(s) = \frac{B_1}{A_1} = S_{11} = \frac{1}{2}(S_B + S_A) \tag{17.21a}$$

$$H_A(s) = \frac{B_2}{A_1} = S_{21} = \frac{1}{2}(S_B - S_A) \tag{17.21b}$$

If Z_A and Z_B are assumed to be lossless LC impedances in order to achieve the low-sensitivity property described in Sec. 17.2, then for $s = j\omega$ they assume imaginary values and, therefore, from Eq. (17.20)

$$|S_A| = |S_B| = 1 \qquad \text{for all } \omega$$

i.e., S_A and S_B are *allpass* transfer functions.

A lattice analog filter can be designed by expressing its transfer function as the difference of allpass transfer functions S_A and S_B, as in Eq. (17.21b), and then realizing these transfer functions. We shall now consider these tasks in the reverse order.

An allpass transfer function of the form

$$S = \frac{B}{A} = \frac{Z(s) - R}{Z(s) + R}$$

can be realized either directly as an allpass network or indirectly by realizing impedance $Z(s)$ and then applying the wave characterization, as can be seen from Eq. (17.3). An arbitrary LC impedance can be realized in terms of the classical *Foster* or *Cauer forms* (see Refs. [1–3] of Chap. 10) or a combination of the two or by a cascade arrangement of unit elements [9]. Consequently, a large variety of realizations are possible for S_A and S_B.

The design effort can be kept to a minimum by realizing S_A and S_B in terms of cascade arrangements of first- and second-order allpass sections which can be realized by simple reactances or resonant circuits. Thus, an arbitrary allpass network represented by

$$S = \prod_{i=1}^{K} S_i$$

can be realized by the cascade arrangement illustrated in Fig. 17.15*a*. Now if the return path from A_2 to B_1 is included as shown and the port resistances are assumed to be R throughout, an LC network realizing S can be obtained as illustrated in Fig. 17.15*b*.

Let us now consider the decomposition of the required transfer function into a difference of two allpass transfer functions, as in Eq. (17.21b). If we let

$$S_A = -\frac{d_A(-s)}{d_A(s)} \tag{17.22a}$$

$$S_B = \frac{d_B(-s)}{d_B(s)} \tag{17.22b}$$

$$S_{11} = \frac{1}{2}(S_B + S_A) = \frac{M(s)}{D(s)} \tag{17.23a}$$

$$S_{21} = \frac{1}{2}(S_B - S_A) = \frac{N(s)}{D(s)} \tag{17.23b}$$

where $d_A(s)$ and $d_B(s)$ are *Hurwitz polynomials* (see Refs. [1–3] of Chap. 10) of degree N_1 and N_2, respectively, with N_1 odd and N_2 even, then Eqs. (17.22) and (17.23) give

$$D(s) = d_A(s)d_B(s)$$

$$\frac{1}{2}[d_A(s)d_B(-s) - d_A(-s)d_B(s)] = M(s)$$

and

$$\frac{1}{2}[d_A(s)d_B(-s) + d_A(-s)d_B(s)] = N(s)$$

where $D(s)$ is of degree N with N odd. Hence

$$d_A(s)d_B(-s) = M(s) + N(s) \tag{17.24}$$

For Butterworth, Chebyshev, inverse-Chebyshev, and elliptic lowpass filters, the factorization of Eq. (17.24) exists. Furthermore, it can be easily obtained by using a technique due to Rhodes [13], as will now be shown. If the denominator of the transfer function is expressed as

$$D(s) = (s + p_1) \prod_{i=2}^{(N+1)/2} (s + p_i)(s + p_i^*)$$

where p_1 is real and $\Im m\ p_i > \Im m\ p_{i-1} > 0$ for $i = 2, 3, \ldots, (N + 1)/2$, then the factorization in Eq. (17.24) can be carried out by assigning poles with odd index to $d_A(s)$ and poles with even index

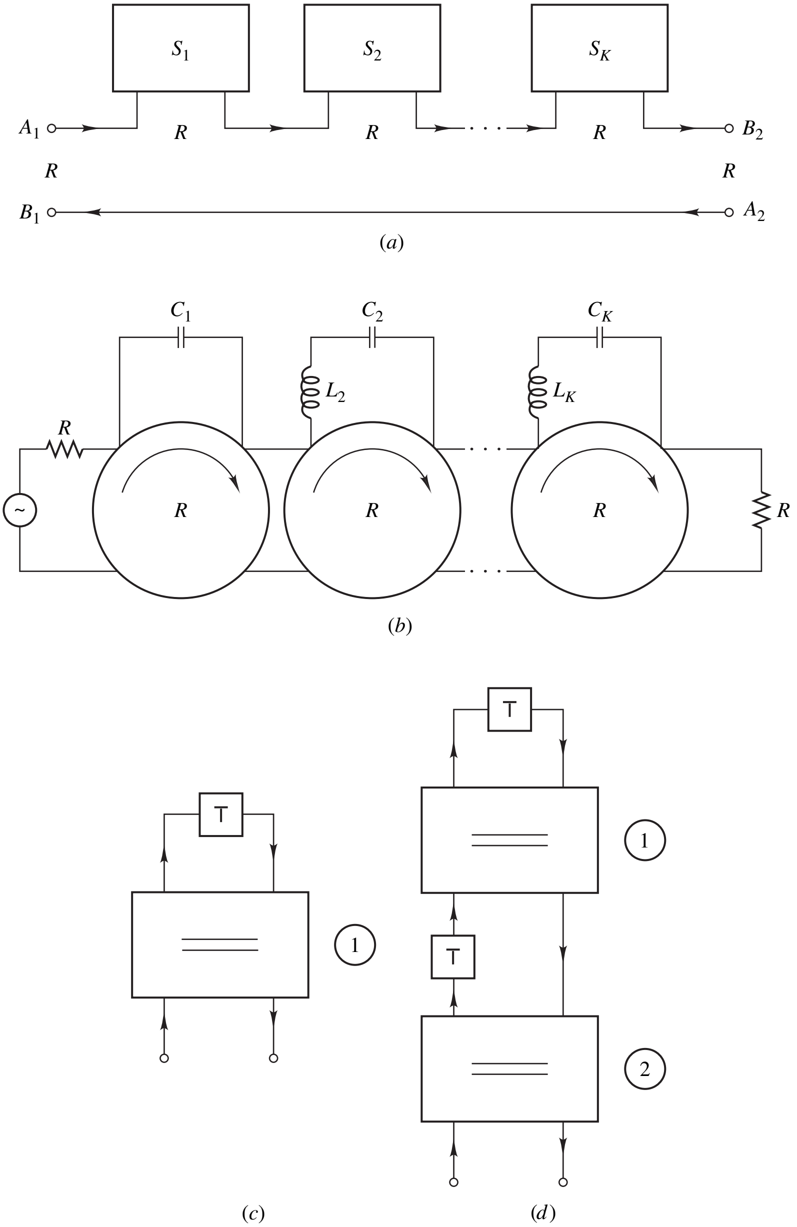


Figure 17.15 (*a*) Realization of transfer function S, (*b*) corresponding LC network, (*c*) digital realization of first-order allpass network, (*d*) digital realization of second-order allpass network.

to $d_B(s)$, that is,

$$d_A(s) = (s + p_1)(s + p_3)(s + p_3^*) \cdots \tag{17.25a}$$

$$d_B(s) = (s + p_2)(s + p_2^*)(s + p_4)(s + p_4^*) \cdots \tag{17.25b}$$

For the aforementioned types of filters, the transfer functions $\tilde{H}_A(s)$ and $H_A(s)$ satisfy the so-called *Feldkeller equation* [10] given by

$$|\tilde{H}_A(j\omega)|^2 + |H_A(j\omega)|^2 = 1$$

Consequently, if $\tilde{H}_A(s)$ represents a lowpass filter such that

$$\tilde{H}_A(e^{j\omega T}) \approx \begin{cases} 1 & \text{for } 0 < |\omega| < \omega_p \\ 0 & \text{otherwise} \end{cases}$$

then

$$H_A(e^{j\omega T}) \approx \begin{cases} 1 & \text{for } \omega_p < |\omega| < \omega_s/2 \\ 0 & \text{otherwise} \end{cases} \tag{17.26}$$

and vice versa. In effect, the configuration of Fig. 17.14*c* realizes simultaneously a lowpass and a highpass filter whose frequency responses are complementary. The structure finds applications in the design of quadrature mirror-image filter banks (see Sec. 18.3).

17.5.3 Digital Realization

Given a lowpass transfer function of the form

$$H_A(s) = \frac{N(s)}{(s + b_{01}) \prod_{i=2}^{(N+1)/2} (s^2 + b_{1i}s + b_{0i})} \tag{17.27}$$

allpass sections characterized by

$$H_1(s) = \frac{-s + b_{01}}{s + b_{01}} \qquad \text{and} \qquad H_i(s) = \frac{s^2 - b_{1i}s + b_{0i}}{s^2 + b_{1i}s + b_{0i}}$$

are required. $H_1(s)$ can be expressed as

$$H_1(s) = \frac{Z(s) - R}{Z(s) + R} \tag{17.28}$$

where $Z(s) = Rb_{01}/s$ is the impedance of a capacitor. It can, therefore, be realized by a 2-port adaptor with a multiplier constant

$$\mu_{11} = \frac{2 - Tb_{01}}{2 + Tb_{01}} \tag{17.29}$$

terminated by a unit delay, as depicted in Fig. 17.15*c*. Similarly, $H_i(s)$ can be put in the form of Eq. (17.28) with

$$Z(s) = \frac{Rs}{b_{1i}} + \frac{Rb_{0i}}{sb_{1i}}$$

In this case, $Z(s)$ represents a series resonant circuit with $L = R/b_{1i}$ and $C = b_{1i}/Rb_{0i}$. A realization of $H_i(s)$ can, therefore, be obtained as shown in Fig. 17.15*d* by using the structure of Fig. 17.13*b*. The bottom adaptor is used to match the port resistance of the resonant circuit to that of the circulator. The multiplier constants of the two adaptors are given by

$$\mu_{1i} = \frac{4 - T^2 b_{0i}}{4 + T^2 b_{0i}} \tag{17.30a}$$

and

$$\mu_{2i} = \frac{2Tb_{1i} - T^2 b_{0i} - 4}{2Tb_{1i} + T^2 b_{0i} + 4} \tag{17.30b}$$

for $i = 2, 3, \ldots, (N+1)/2$.

A transfer function of the form given by Eq. (17.27) can be realized in terms of a wave lattice structure by using the following procedure:

1. Carry out the decomposition in Eq. (17.24) as in Eq. (17.25) using the technique described.
2. Form the allpass transfer functions S_A and S_B as in Eq. (17.22).
3. Realize the allpass sections obtained in (2) using the structure in Fig. 17.15*c* for the first-order section and the structure of Fig. 17.15*d* for second-order sections.
4. Form the digital realizations of networks N_A and N_B and connect them as in Fig. 17.14*c*.

The transfer function of the digital filter obtained is given by Eq. (17.21b) as

$$H_D(z) = H_A(s)\Big|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}$$

Example 17.1 Obtain a lattice realization for the fifth-order Butterworth lowpass transfer function

$$H_A(s) = \frac{1}{(s+1)(s^2 + 0.618034s + 1)(s^2 + 1.618034s + 1)}$$

Assume that $\omega_s = 2\pi$.

■ **Solution**

The polynomials $d_A(s)$ and $d_B(s)$ are given by

$$d_A(s) = (s+1)(s^2 + 0.618034s + 1)$$
$$d_B(s) = s^2 + 1.618034s + 1$$

Hence

$$S_A(s) = -\frac{(-s+1)(s^2 - 0.618034s + 1)}{(s+1)(s^2 + 0.618034s + 1)}$$
$$S_B(s) = \frac{s^2 - 1.618034s + 1}{s^2 + 1.618034s + 1}$$

With $\omega_s = 2\pi$, we have $T = 1$s; on using Eqs. (17.28)–(17.30), the multiplier constants of the adaptors can be computed as $\mu_{11} = 0.333333$, $\mu_{12} = 0.60$, $\mu_{22} = -0.214172$, $\mu_{13} = 0.60$, and $\mu_{23} = -0.603575$. The realization obtained is shown in Fig. 17.16. ■

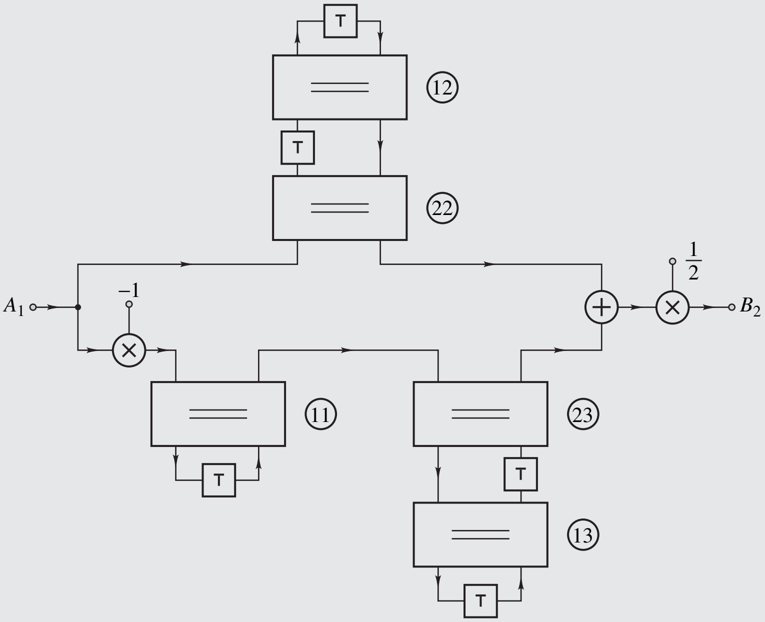


Figure 17.16 Lattice realization of Butterworth lowpass filter (Example 17.1).

17.6 LADDER WAVE DIGITAL FILTERS

Although lattice filters are easy to design and have low sensitivity to coefficient quantization at passband frequencies, the sensitivity at stopband frequencies can be quite high. This is due to the fact that transmission zeros (frequencies of zero gain) are achieved by the exact cancellation of the signals through networks N_A and N_B. A class of wave digital filters in which the sensitivity is low at passband as well as stopband frequencies can be obtained by realizing the 2-port in Fig. 17.1*a* in terms of an *LC ladder* network [13–17] and then applying the wave characterization. The following design procedure can be employed.

1. Identify the various series and parallel wire interconnections in the LC filter and number the ports such that every direct connection between wire-interconnection ports involves a port 2.
2. Assign port resistances to the wire-interconnection ports. For a port terminated by an impedance $s^\lambda R_x$ or by a voltage source with an internal impedance $s^\lambda R_x$ assign a port resistance $(2/T)^\lambda R_x$. Then choose the unspecified port resistances to give as far as possible type S1 and P1 adaptors, ensuring that a common resistance is assigned to any two interconnected ports.
3. Calculate the multiplier constants for the various adaptors.
4. Replace each analog element in the LC filter by its digital realization.

The transfer function of the filter obtained is given by

$$H_D(z) = \frac{B_2}{A_1} = \frac{A_o}{B_i} \tag{17.31}$$

where A_o is the incident wave quantity for the output resistance and B_i is the reflected wave quantity for the input source. From Eqs. (17.3)–(17.6), and (17.8), we obtain $B_i = V_i$ and $A_o = 2V_o$ and so Eq. (17.31) yields

$$H_D(z) = 2H_A(s)\Big|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}$$

where $H_A(s)$ is the transfer function of the analog filter.

Example 17.2 Figure 17.17*a* represents an elliptic lowpass filter satisfying the following specifications:

- Maximum passband loss: 1.0 dB
- Minimum stopband loss: 34.5 dB
- Passband edge: $\sqrt{0.5}$ rad/s
- Stopband edge: $1/\sqrt{0.5}$ rad/s

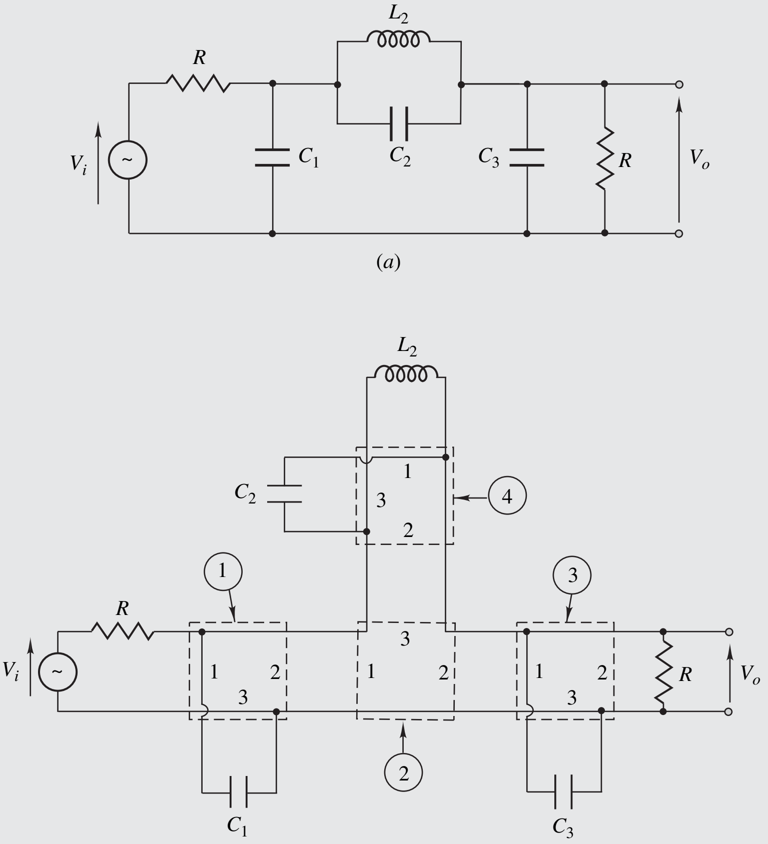
The element values of the filter are

$$C_1 = C_3 = 2.6189 \text{ F} \qquad C_2 = 0.31946 \text{ F} \qquad L_2 = 1.2149 \text{ H} \qquad R = 1\ \Omega$$

Derive a corresponding wave digital filter using a sampling frequency of 10 rad/s.

■ **Solution**

The wire interconnections can be identified as illustrated in Fig. 17.17*b*. Let G_{jk} (R_{jk}) represent the port conductance (resistance) assigned to the jth port of the kth wire interconnection. From step 2 of the above procedure the following assignments can be made:



(*a*)

(*b*)

Figure 17.17 (*a*) Elliptic lowpass filter (Example 17.2), (*b*) identification of wire interconnections.

Interconnection 1:

$$G_{11} = \frac{1}{R} \qquad G_{31} = \frac{2C_1}{T} \qquad G_{21} = G_{11} + G_{31} \qquad m_{p1} = 0.107110$$

Interconnection 4:

$$G_{14} = \frac{T}{2L_2} \qquad G_{34} = \frac{2C_2}{T} \qquad G_{24} = G_{14} + G_{34} \qquad m_{p1} = 0.202741$$

Interconnection 2:

$$R_{12} = \frac{1}{G_{21}} \qquad R_{32} = \frac{1}{G_{24}} \qquad R_{22} = R_{12} + R_{32} \qquad m_{s1} = 0.120194$$

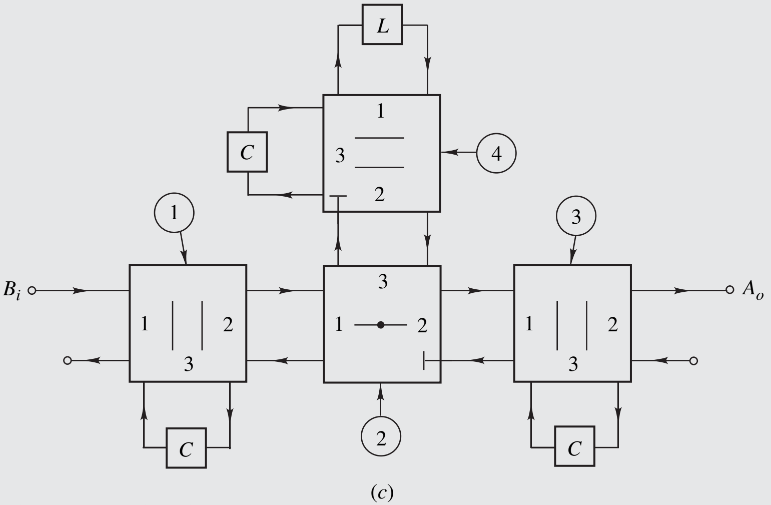


Figure 17.17 Cont'd (*c*) Wave digital filter.

Interconnection 3:

$$G_{13} = \frac{1}{R_{22}} \qquad G_{23} = \frac{1}{R} \qquad G_{33} = \frac{2C_3}{T}$$

$$m_{p1} = 0.214595 \qquad m_{p2} = 0.191234$$

Interconnections 1, 2, 3, and 4 result in P1, S1, P2, and P1 adaptors, respectively, as depicted in Fig. 17.17*c*. The multiplier coefficients can be computed by using Eqs. (17.10)–(17.12). ■

The realization of Fig. 17.17*c* is one of a large number of possibilities since the assignment of port resistances to wire interconnections is not unique and one may realize the resonant circuits using unit elements and capacitors or inductors instead of 3-port adaptors, capacitors, and inductors (see Figs. 17.12 and 17.13). Furthermore, alternative realizations can be obtained by viewing the 2-port ladder network as a cascade connection of elemental 2-ports which can be realized individually [18, 19]. While many of these structures may appear to be equivalent in terms of computational complexity, frequently there are important practical considerations that may favor one or the other type of structure, depending on the application at hand. Topological constraints necessitate that one of the adaptors in a ladder digital filter be of the unconstrained type (i.e., of type S2 or P2). Increased speed of operation and signal-to-noise ratio can often be achieved by assigning port resistances such that an unconstrained adaptor is obtained at the center of the structure, e.g., for interconnection 2 of the filter in Fig. 17.17*b*. There are many other issues involved in the design of wave digital filters, like their VLSI implementation and the compensation for parasitic delays that arise in practical circuits. These issues and many others are considered in some detail by Fettweis [9].[1]

It should be mentioned that the numbers of delays and adders in ladder wave digital filters can sometimes be reduced somewhat by employing a pair of impedance transformations first used by Bruton [20] in the domain of active filters. In these transformations, each impedance $Z(s)$ in the analog filter is replaced by $sZ(s)$ or $Z(s)/s$. In the first case, impedances R_x, R_x/s, and sR_x are replaced by impedances sR_x, R_x, and s^2R_x, that is, resistances translate into inductances, capacitances into resistances, and inductances into s^2-impedance elements. With $s=j\omega$, we have $s^2=-\omega^2$, that is, an s^2-impedance element behaves as a frequency-dependent negative resistance (FDNR) and for this reason the transformed network is often referred to as an *FDNR network*. With this transformation, filters with a large number of capacitances and a small number of inductances, e.g., minimum inductance lowpass and highpass filters, translate into filters with a large number of resistances and a small number of s^2-impedance elements. Since the digital equivalents of resistances are simple sinks, the FDNR filter leads to a somewhat more economical digital design [21].

17.7 FILTERS SATISFYING PRESCRIBED SPECIFICATIONS

In lattice as well as ladder digital filters, the wave quantities are transformed using the bilinear transformation (see Eq. (17.4)) and, therefore, the filters obtained, like other filters based on the bilinear transformation, are subject to the warping effect discussed in Sec. 11.6.3.

Wave digital filters that would satisfy prescribed specifications can be designed by using the prewarping techniques of Chap. 12. A detailed design procedure is as follows:

1. Using the specifications, derive an appropriate normalized lowpass transfer function according to steps 1 to 3 in Sec. 12.4.
2. Realize the transfer function derived in step 1 as an equally terminated LC lattice or ladder filter.
3. Transform the lowpass filter realized in step 2 using the appropriate formula in Table 12.1.
4. Form the desired digital filter using the procedure in Sec. 17.5 or 17.6.

For the case of ladder filters, step 2 can be carried out by using filter-design tables like those found in Skwirzynski [14], Saal [15], and Zverev [16] or by using one of several available filter-design software.[2]

Example 17.3 Design a wave bandpass digital filter that would satisfy the following specifications:

- Maximum passband loss: 1.0 dB
- Minimum stopband loss: 35.0 dB
- Lower and upper passband edges: 2.0 and 3.0 rad/s
- Lower and upper stopband edges: 1.5 and 3.5 rad/s
- Sampling frequency: 10 rad/s

[1] *This paper includes,in addition, a fairly long list of references on the subject.*

[2] *See Circuit Sage at http://www.circuitsage.com/filter.html.*

■ **Solution**

On choosing an elliptic approximation and then using the procedure in Sec. 12.4, we obtain

$$n = 3 \qquad k = 0.4472136 \qquad \omega_0 = 3.183099 \qquad B = 3.093133$$

where n and k are the order and selectivity factor of the normalized lowpass filter, respectively, and ω_0 and B are the parameters in the transformation

$$s = \frac{1}{B}\left(\bar{s} + \frac{\omega_0^2}{\bar{s}}\right)$$

A normalized lowpass LC filter with $n = 3$ and $k = 0.45$ can be obtained from Ref. [14] as depicted in Fig. 17.17*a*, where we now have

$$C_1 = C_3 = 2.8130\text{ F} \qquad C_2 = 0.26242\text{ F} \qquad L_2 = 1.3217\text{ H} \qquad R = 1\ \Omega$$

On applying the above lowpass-to-bandpass transformation, the bandpass filter of Fig. 17.18*a* can be formed, where

$$\begin{gathered} L_1' = L_4' = 0.108525\text{ H} \qquad C_1' = C_4' = 0.909434\text{ F} \\ L_2' = 0.427301\text{ H} \qquad C_2' = 0.230975\text{ F} \\ L_3' = 1.16333\text{ H} \qquad C_3' = 0.0848396\text{ F} \\ R = 1\ \Omega \end{gathered}$$

Table 17.1 Multiplier constants (Example 17.3)

Adaptor	Type	k	m_{pk} or m_{sk}
1	P1	1	0.256751
2	P1	1	0.573642
3	S1	1	0.117926
4	P1	1	0.216662
5	P2	1	0.973738
		2	0.263494
6	P1	1	0.702492
7	P1	1	0.576495
8	S1	1	0.500000

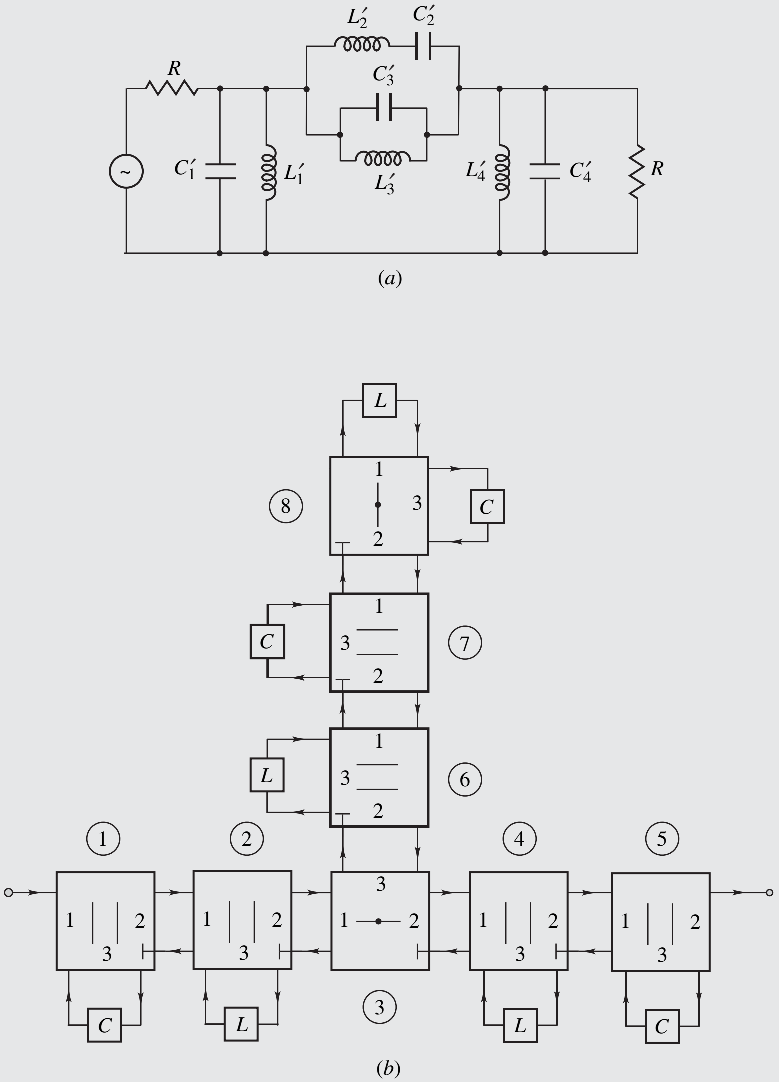


Figure 17.18 (*a*) Bandpass elliptic filter (Example 17.3), (*b*) wave digital filter.

Subsequently, on using the procedure of Sec. 17.6, the wave digital filter of Fig. 17.18*b* can be derived. The resulting multiplier constants are given in Table 17.1. The loss characteristic achieved is plotted in Fig. 17.19. ■

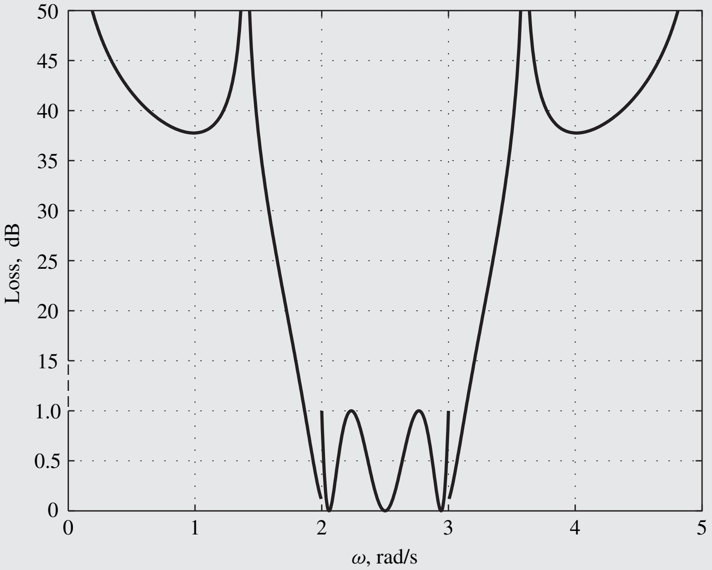


Figure 17.19 Loss characteristic of a bandpass filter (Example 17.3).

17.8 FREQUENCY-DOMAIN ANALYSIS

Once a wave digital filter is designed, a frequency-domain analysis is often necessary to study quantization effects or simply to verify the design. Such an analysis will now be described.

Consider the network in Fig. 17.20*a*, where adaptor q is terminated by subnetworks N_p, N_r, and N_s. Adaptor q can be characterized by

$$H_q(z) = \frac{B_{2q}}{A_{1q}} \qquad F_{1q} = \frac{B_{1q}}{A_{1q}} \qquad F_{2q} = \frac{B_{2q}}{A_{2q}}$$

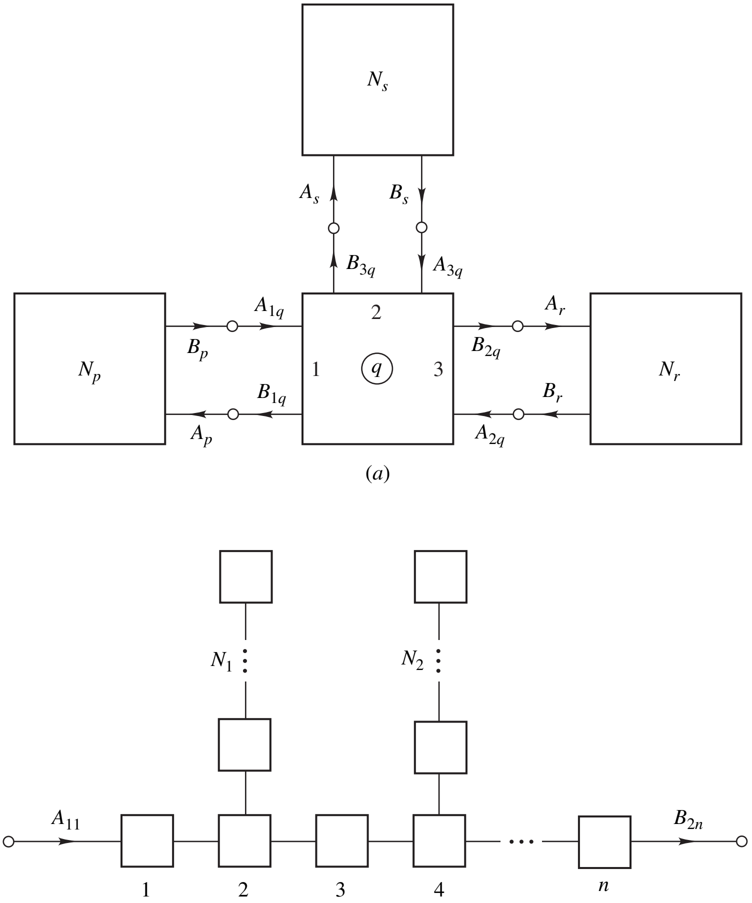
$H_q(z)$ is the transfer function of the terminated adaptor, and F_{1q} and F_{2q} are its *input functions* at ports 1 and 2, respectively. Similarly, subnetworks N_p, N_r, and N_s can be characterized by the input functions

$$F_p = \frac{B_p}{A_p} = \frac{A_{1q}}{B_{1q}} \qquad F_r = \frac{B_r}{A_r} = \frac{A_{2q}}{B_{2q}} \qquad F_s = \frac{B_s}{A_s} = \frac{A_{3q}}{B_{3q}}$$

Expressions for $H_q(z)$, F_{1q}, and F_{2q} in terms of F_p, F_r, and F_s for series and parallel adaptors can be derived from Eqs. (17.9) and (17.11). For the S2 adaptor we have

$$H_q(z) = \frac{m_{s2}(F_s - 1)}{D_1} \qquad F_{1q} = \frac{-C_1 + C_3F_r - C_2F_s - F_rF_s}{D_1}$$

$$F_{2q} = \frac{-C_2 + C_3F_p - C_1F_s - F_pF_s}{D_2}$$



(*a*)

(*b*)

Figure 17.20 Analysis of wave digital filters: (*a*) Embedded adaptor, (*b*) block diagram.

where

$$D_1 = 1 + C_2F_r - C_3F_s + C_1F_rF_s \qquad D_2 = 1 + C_1F_p - C_3F_s + C_2F_pF_s$$

$$C_1 = m_{s1} - 1 \qquad C_2 = m_{s2} - 1 \qquad C_3 = m_{s1} + m_{s2} - 1$$

Similarly, for the P2 adaptor

$$H_q(z) = \frac{m_{p1}(1 + F_s)}{D_1} \qquad F_{1q} = \frac{C_1 + C_3F_r - C_2F_s + F_rF_s}{D_1}$$

$$F_{2q} = \frac{C_2 + C_3F_p - C_1F_s + F_pF_s}{D_2}$$

where

$$D_1 = 1 - C_2 F_r + C_3 F_s + C_1 F_r F_s \qquad D_2 = 1 - C_1 F_p + C_3 F_s + C_2 F_p F_s$$

$$C_1 = m_{p1} - 1 \qquad C_2 = m_{p2} - 1 \qquad C_3 = m_{p1} + m_{p2} - 1$$

These relations apply to S1 and P1 adaptors except that $m_{s2} = 1$ and $m_{p2} = 1$.

Now consider the filter of Fig. 17.20*b* and assume that the adaptors of the main path are numbered consecutively from input to output. The overall transfer function of the filter is

$$H(z) = \frac{B_{2n}}{A_{11}}$$

Since the reflected and incident wave quantities at the output of adaptor q become the incident and reflected wave quantities at the input of adaptor $q + 1$, respectively, we can write

$$\begin{aligned} H(z) &= \frac{B_{21}}{A_{11}} \frac{B_{22}}{B_{21}} \cdots \frac{B_{2n}}{B_{2(n-1)}} \\ &= \frac{B_{21}}{A_{11}} \frac{B_{22}}{A_{12}} \cdots \frac{B_{2n}}{A_{1n}} \end{aligned}$$

Therefore

$$H(z) = \prod_{q=1}^{n} H_q(z) \tag{17.32}$$

For the connection of Fig. 17.20*a*, $H_q(z)$ and F_{1q} depend on F_s and F_r, as was shown earlier. If N_s (N_r) comprises a cascade of adaptors, F_s (F_r) will depend on the input function of the second adaptor in the cascade, which will in turn depend on the input function of the third adaptor, and so on. Consequently, for the filter in Fig. 17.20*b*, the input functions of branches N_1, N_2, ... must be evaluated first, starting with the last adaptor and proceeding to the branch input in each case. Subsequently, the main-path adaptors should be analyzed, starting with the output adaptor and proceeding to the filter input. With the frequency responses of the individual main-path adaptors known, the overall response of the filter can be evaluated by using Eq. (17.32).

17.9 SCALING

The signal-to-noise ratio in wave digital filters can be improved by applying signal scaling, as in other types of digital filters. This can be achieved by scaling the incident and reflected wave quantities at port 1 of the ith adaptor by factors λ_i and $1/\lambda_i$, respectively, as depicted in Fig. 17.21. The first multiplier scales down the inputs of adaptor multipliers in order to avoid overflow, whereas the second one ensures that the input functions at ports 1 and 2 of the adaptor remain unchanged after the application of signal scaling. Note that the two multipliers form a 2-port that is equivalent to an ideal transformer in the analog network with a turns ratio of λ, as can be seen in Fig. 17.9*c*.

If the top and bottom multiplier constants are not exactly the reciprocal of each other, then frequency-response errors similar to coefficient-quantization errors will occur. This problem can be

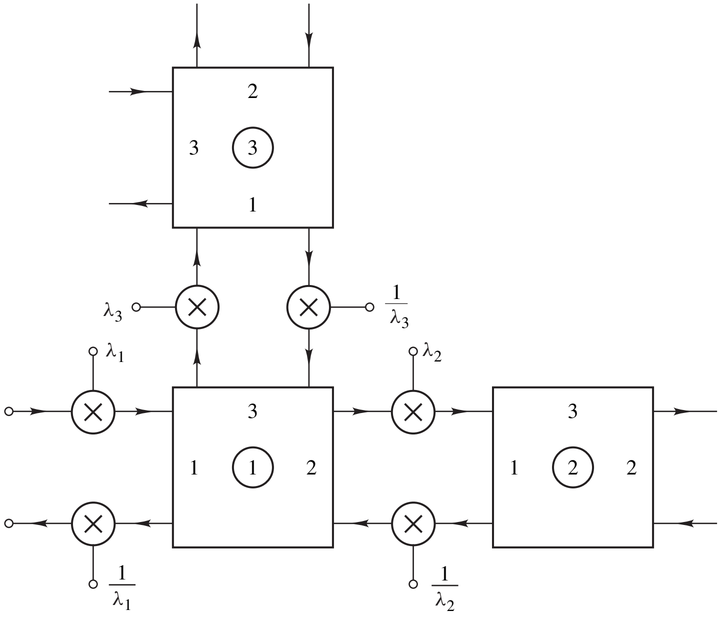


Figure 17.21 Application of signal scaling to wave digital filters.

avoided in practice by choosing λ to be a power of 2. This choice of constant has the additional advantage that both scaling multiplications become simple data shifts which are easy to implement. Unfortunately, however, with this choice of λ the signal-to-noise ratio cannot be optimal.

17.10 ELIMINATION OF LIMIT-CYCLE OSCILLATIONS

In Sec. 14.9.3, it was shown that in certain second-order structures zero-input limit-cycle oscillations can be eliminated by carrying out the quantization of signals in terms of magnitude truncation. A similar approach is applicable for the class of wave digital filters, as was demonstrated by Fettweis and Meerkötter [22]. The details of this approach are as follows.

Consider the wave digital filter of Fig. 17.22, where block B is a linear subnetwork containing adders, multipliers, and interconnections but no unit delays or delay-free loops. Further, assume that signal quantization is carried out by using quantizers Q_k for $k = 3, 4, \ldots, N$, as shown, and let the block enclosed by dashed lines be referred to as block $\tilde{B}$. The quantity

$$p_k(n) = G_k a_k^2(n)$$

where G_k is the conductance assigned to the kth port, represents the power[3] stored in the kth unit delay at instant nT. Since port conductances can be assigned on an arbitrary basis as long as pairs of interconnected ports are assigned the same conductances (see Sec. 17.3), G_k for $k = 3, 4, \ldots, N$ can be assumed to be positive without any loss of generality. Hence the *total power stored* in all the

[3] *Fettweis refers to this quantity as pseudopower to distinguish it from its analog counterpart which is actual power.*

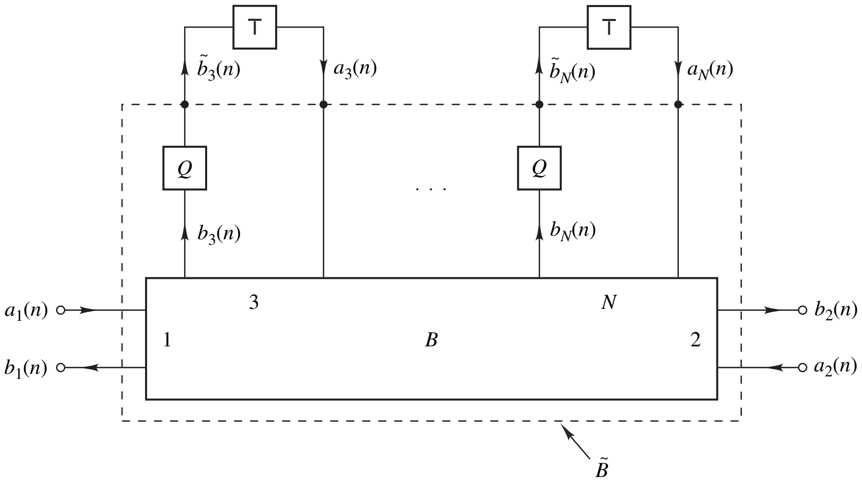


Figure 17.22 Elimination of zero-input limit-cycle oscillations in wave digital filters.

unit delays at instant nT can be obtained as

$$p_D(n) = \sum_{k=3}^{N} G_k a_k^2(n)$$

The increase in $p_D(n)$ after one filter cycle is given by

$$\begin{aligned}\Delta p_D(n) &= p_D(n+1) - p_D(n)\\ &= \sum_{k=3}^{N} G_k \left[a_k^2(n+1) - a_k^2(n)\right]\end{aligned}$$

and since

$$a_k(n+1) = \tilde{b}_k(n)$$

we have

$$\Delta p_D(n) = \sum_{k=3}^{N} G_k \left[\tilde{b}_k^2(n) - a_k^2(n)\right] \tag{17.33}$$

The total power absorbed by block $\tilde{B}$ at instant nT is given by

$$\tilde{p}_N(n) = \sum_{k=1}^{2} G_k \left[a_k^2(n) - b_k^2(n)\right] + \sum_{k=3}^{N} G_k \left[a_k^2(n) - \tilde{b}_k^2(n)\right] \tag{17.34}$$

(see Ref. [3]). Hence, Eqs. (17.33) and (17.34) give

$$\Delta p_D(n) = -\tilde{p}_N(n) + \sum_{k=1}^{2} G_k\left[a_k^2(n) - b_k^2(n)\right]$$

and under zero-input conditions such that $a_1(n) = a_2(n) = 0$, we have

$$\Delta p_D(n) = -\tilde{p}_N(n) - G_1 b_1^2(n) - G_2 b_2^2(n) \tag{17.35}$$

If the reflected quantities at ports 3 to N are quantized such that

$$|\tilde{b}_k(n)| \le |b_k(n)| \qquad \text{for } k = 3, 4, \ldots, N \tag{17.36}$$

then from Eqs. (17.35) and (17.36), the power absorbed by block $\tilde{B}$ can be expressed as

$$\begin{aligned}\tilde{p}_N(n) &= \sum_{k=1}^{2} G_k\left[a_k^2(n) - b_k^2(n)\right] + \sum_{k=3}^{N} G_k\left[a_k^2(n) - \tilde{b}_k^2(n)\right]\\ &\ge \sum_{k=1}^{N} G_k\left[a_k^2(n) - b_k^2(n)\right] = p_N(n)\end{aligned} \tag{17.37}$$

where $p_N(n)$ is the power absorbed by block B. Now if block B represents a wave digital filter derived from a passive network, we have [3]

$$p_N(n) \ge 0 \tag{17.38}$$

and from Eqs. (17.37) and (17.38), we conclude that

$$\tilde{p}_N(n) \ge 0$$

Hence Eq. (17.35) yields

$$\Delta p_D(n) \le 0$$

Under these circumstances, the total power stored in the unit delays cannot increase and, for the reasons stated immediately after Eq. (14.88), the wave digital filter cannot sustain zero-input limit-cycle oscillations. Therefore, wave digital filters obtained from passive networks support the elimination of zero-input limit cycles.

Overflow oscillations can be eliminated as in other realizations that support the elimination of zero-input limit cycles, as described in Sec. 14.9.4 (see Ref. [36] of Chap. 14).

17.11 RELATED SYNTHESIS METHODS

An alternative but closely related methodology for the design of low-sensitivity digital filters has been developed by Vaidyanathan and Mitra and others [23, 24]. This methodology encompasses concepts that are analogous to those found in classical network synthesis (such as, passivity, positive real functions, 2-port networks, and extraction of elements) and provides a framework for the realization to be carried out entirely in the z domain without recourse to LC prototype filters. The methodology can be used for the realization of a variety of types of filters, including wave ladder and lattice filters as well as low-sensitivity nonrecursive filters.

Yet another class of low-sensitivity filters is the class of *lossless-discrete-integrator (LDI) ladder* filters, proposed by Bruton [25] and developed further by Bruton and Vaughan-Pope, and others [26–28]. As in wave digital filters, low sensitivity is achieved by emulating analog LC filters. In this approach, the problem of delay-free loops is avoided by replacing the bilinear transformation by the transformation

$$s = \frac{1}{T}\left(\frac{z-1}{z^{\frac{1}{2}}}\right)$$

In this way, the required digital filter is obtained directly from the analog filter without recourse to the wave characterization. The penalty paid is that the one-to-one correspondence between the imaginary axis of the s plane and the unit circle of the z plane is lost and, as a consequence, various techniques must be used to correct the distortion introduced in the amplitude response.

17.12 A CASCADE SYNTHESIS BASED ON THE WAVE CHARACTERIZATION

The wave characterization along with the concept of the *generalized-immittance converter* (GIC) [29] can be used to develop an alternative to the cascade realization of Sec. 8.2.5 [11, 30]. The details of this approach are as follows.

17.12.1 Generalized-Immittance Converters

A GIC is a 2-port whose input admittance Y_i is related to the load admittance Y_L by

$$Y_i = h(s)Y_L$$

where $h(s)$ is the *admittance conversion function* of the device. Two specific types of GIC can be identified, namely, voltage- and current-conversion GICs. The current-conversion GIC (CGIC) is characterized by the terminal relations

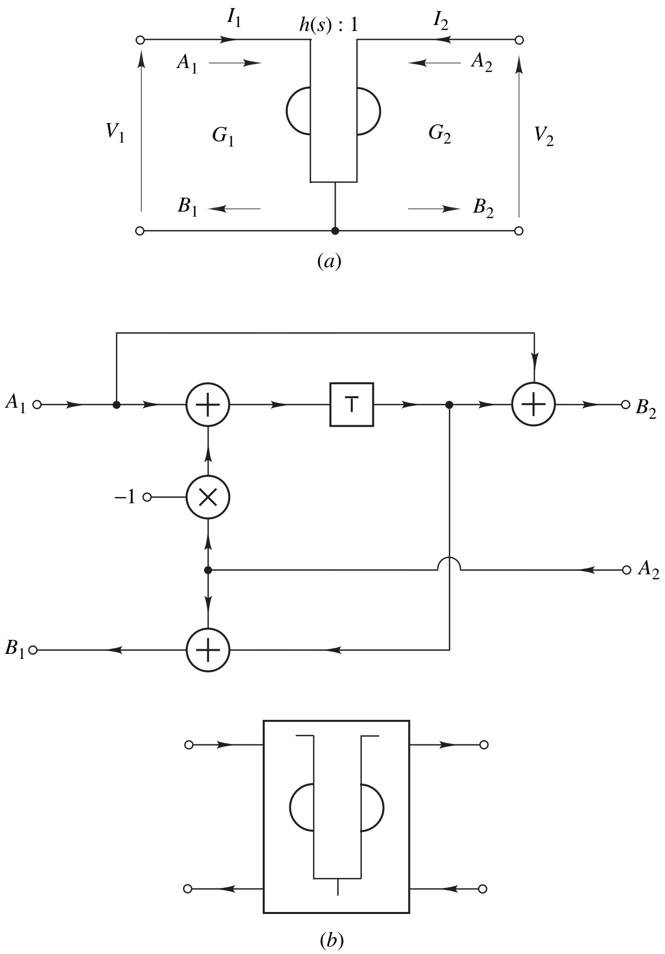
$$V_1 = V_2 \qquad I_1 = -h(s)I_2 \tag{17.39}$$

This is usually represented by the symbol of Fig. 17.23*a*.

17.12.2 Analog G-CGIC Configuration

By interconnecting three conductances and two CGICs, we can construct the G-CGIC configuration of Fig. 17.24*a* [31]. If each CGIC is assumed to have a conversion function $h(s) = s$, straightforward analysis yields

$$\frac{V_o}{V_i} = \frac{k_0G_0 + k_1G_1s + k_2G_2s^2}{G_0 + G_1s + G_2s^2}$$



(*a*)

(*b*)

Figure 17.23 (*a*) Current-conversion generalized-immittance converter, (*b*) digital realization.

and if $G_r = b_r$ and $k_r = a_r/b_r$ for $r = 0, 1, 2$, the network realizes the transfer function

$$H(s) = \frac{a_0 + a_1 s + a_2 s^2}{b_0 + b_1 s + b_2 s^2} \tag{17.40}$$

By cascading a number of sections like the above any stable continuous-time transfer function can be realized.

17.12.3 Digital G-CGIC Configuration

Like an LC network, the G-CGIC network of Fig. 17.24*a* can be readily simulated by digital elements. We need only develop a digital realization for the CGIC by using the procedure outlined in Sec. 17.4.

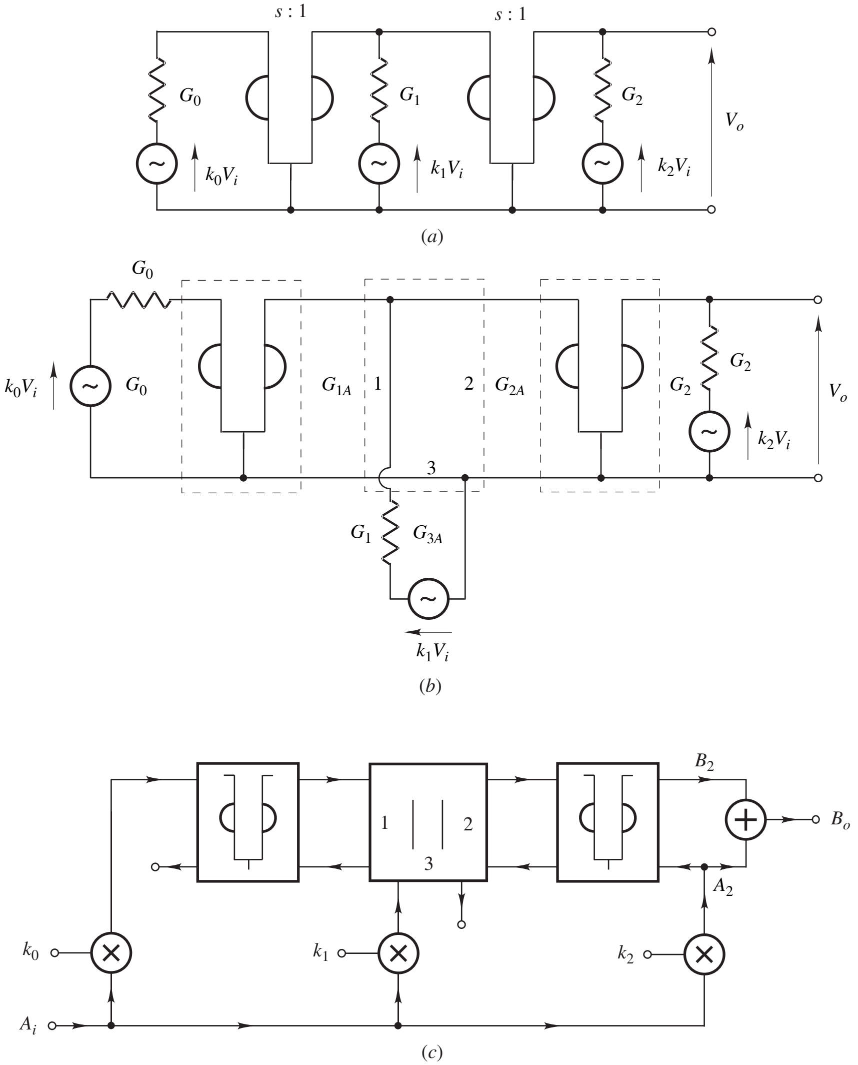


Figure 17.24 (*a*) Analog G-CGIC configuration, (*b*) identification of *N*-ports, (*c*) digital realization.

On assigning wave quantities and conductances to the CGIC ports as illustrated in Fig. 17.23*a*, and then using Eqs. (17.1), (17.4), and (17.39), we can show that

$$B_1 = A_2 + (A_1 - A_2)F(z) \qquad B_2 = A_1 + (A_1 - A_2)F(z)$$

where

$$F(z) = \frac{G_1 - G_2h(z)}{G_1 + G_2h(z)} \tag{17.41}$$

$$h(z) = h(s)\Big|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}$$

Hence with $h(s) = s$ and $G_1 = 2G_2/T$, Eq. (17.41) reduces to

$$F(z) = z^{-1}$$

Therefore, a digital realization for the CGIC can be obtained, as depicted in Fig. 17.23*b*.

The individual N-ports of the G-CGIC configuration can now be identified, as indicated in Fig. 17.24*b*. On assigning the port conductances

$$G_{1A} = \frac{TG_0}{2} \qquad G_{2A} = \frac{2G_2}{T} \qquad G_{3A} = G_1$$

the general second-order digital section of Fig. 17.24*c* can be derived. An output proportional to V_o can be formed by using an adder at the input or output of any one of the CGICs, as in Fig. 17.24*c*, or at port 3 of the adaptor. This is permissible by virtue of Eq. (17.39).

The transfer function of the derived structure can be obtained from Eqs. (17.1) and (17.4) as

$$H_D(z) = \frac{B_o}{A_i} = \frac{B_2 + A_2}{A_i} = \frac{2V_o}{V_i} = 2H(s)\Big|_{s=\frac{2}{T}\left(\frac{z-1}{z+1}\right)}$$

17.12.4 Cascade Synthesis

Almost invariably recursive filters are designed by using Butterworth, Chebyshev, Bessel-Thomson, or elliptic transfer functions which have zeros at the origin of the s plane, on the imaginary axis, or at infinity (see Chap. 10). Hence the continuous-time transfer function can be realized as a cascade connection of second-order sections characterized by transfer functions of the type

$$H_A(s) = \frac{N_A(s)}{b_0 + b_1 s + s^2}$$

where $N_A(s)$ can take the form b_0, s^2, $b_1 s$, or $a_0 + s^2$ for a lowpass (LP), highpass (HP), bandpass (BP), or notch (N) section, respectively. On the other hand, delay equalizers are designed by using allpass (AP) sections, in which

$$H_A(s) = \frac{b_0 - b_1 s + s^2}{b_0 + b_1 s + s^2}$$

(see Sec. 5.6.2).

Evidently, the above transfer functions are special cases of the transfer function in Eq. (17.40), and, therefore, they can all be readily realized by using the digital structure of Fig. 17.24*c*. The resulting structures are shown in Fig. 17.25, where

$$k_0 = \frac{a_0}{b_0} \tag{17.42}$$

$$m_1 = \frac{b_0 - (2/T)b_1 - (2/T)^2}{b_0 + (2/T)b_1 + (2/T)^2} \tag{17.43}$$

$$m_2 = -\frac{b_0 + (2/T)b_1 - (2/T)^2}{b_0 + (2/T)b_1 + (2/T)^2} \tag{17.44}$$

in each case.

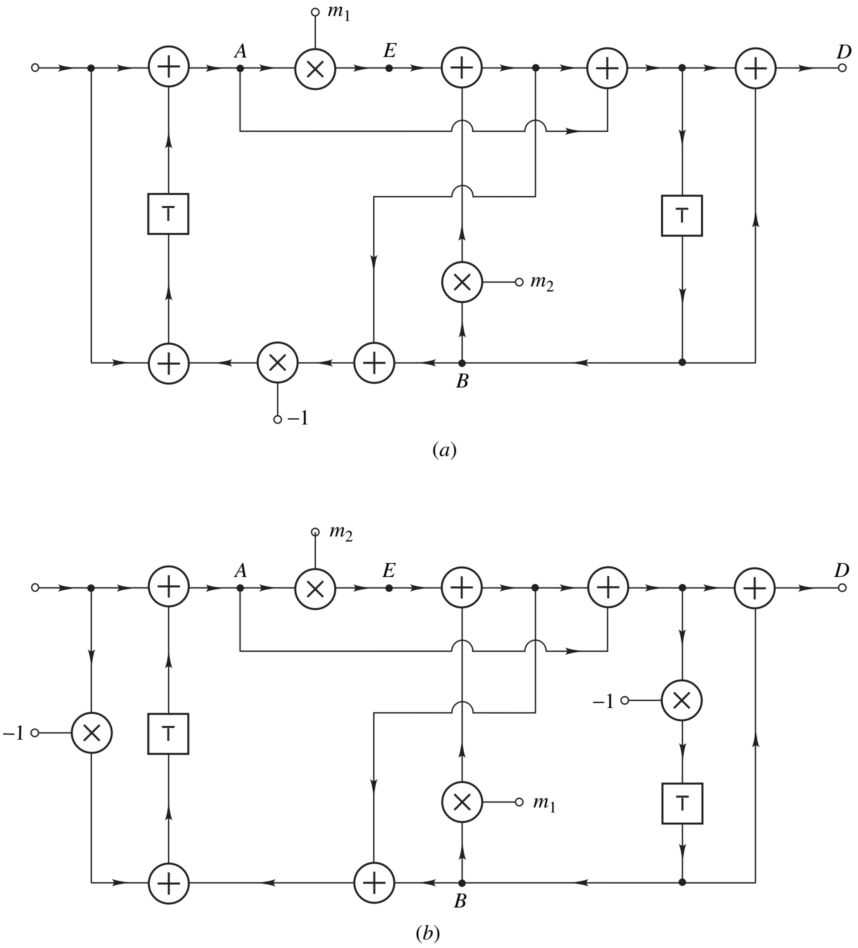


Figure 17.25 Universal second-order CGIC sections: (*a*) Lowpass, (*b*) highpass.

With a set of universal sections available, any Butterworth, Chebyshev, Bessel-Thomson, or elliptic digital filter satisfying prescribed specifications can be designed by using the following procedure:

1. Using the specifications, derive the appropriate normalized lowpass transfer function according to steps 1 to 3 in Sec. 12.4.
2. Apply the transformation in Eq. (12.1).
3. Select suitable sections from Fig. 17.25.
4. Calculate the multiplier constants using Eqs. (17.42)–(17.44).
5. Connect the various sections in cascade.

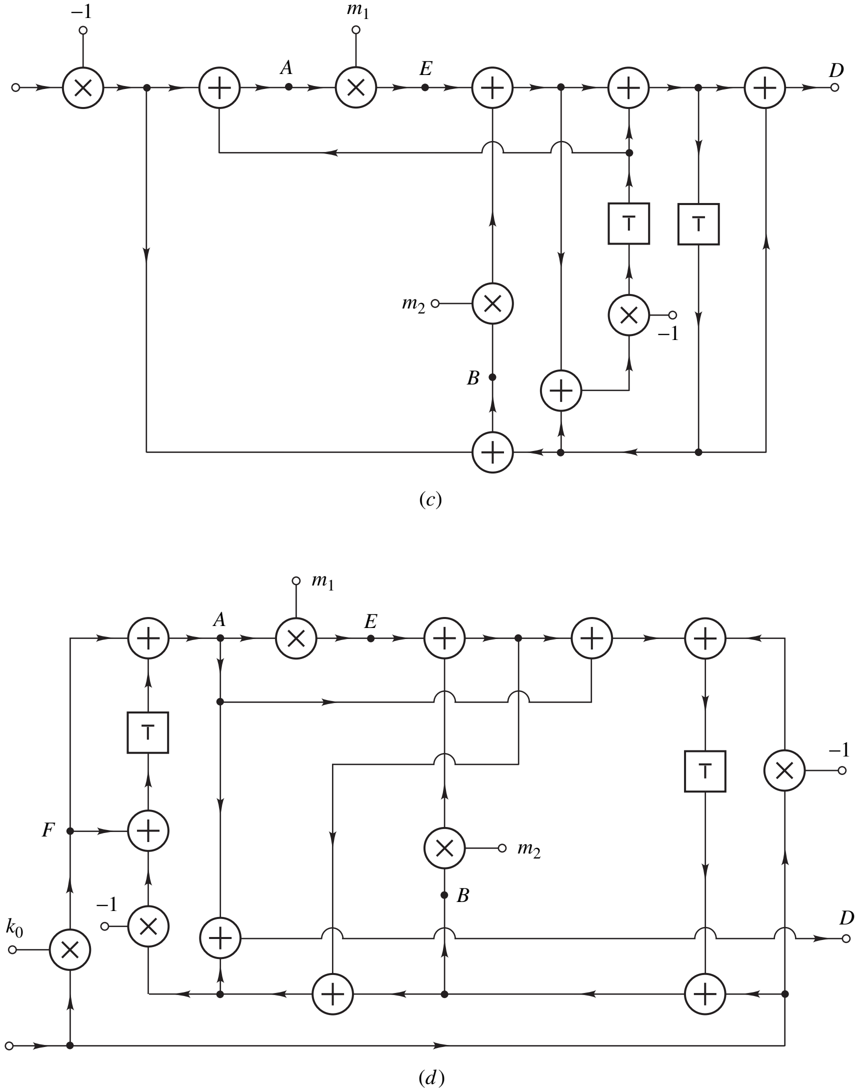


Figure 17.25 Cont'd Universal second-order CGIC sections: (*c*) Bandpass, (*d*) notch.

Example 17.4 A Butterworth lowpass filter is characterized by

$$H(s) = \prod_{j=1}^{3} \frac{b_{0j}}{b_{0j} + b_{1j}s + s^2}$$

where coefficients b_{ij} are given in Table 17.2. Design a corresponding digital filter by using the CGIC cascade synthesis. The sampling frequency is 10^4 rad/s.

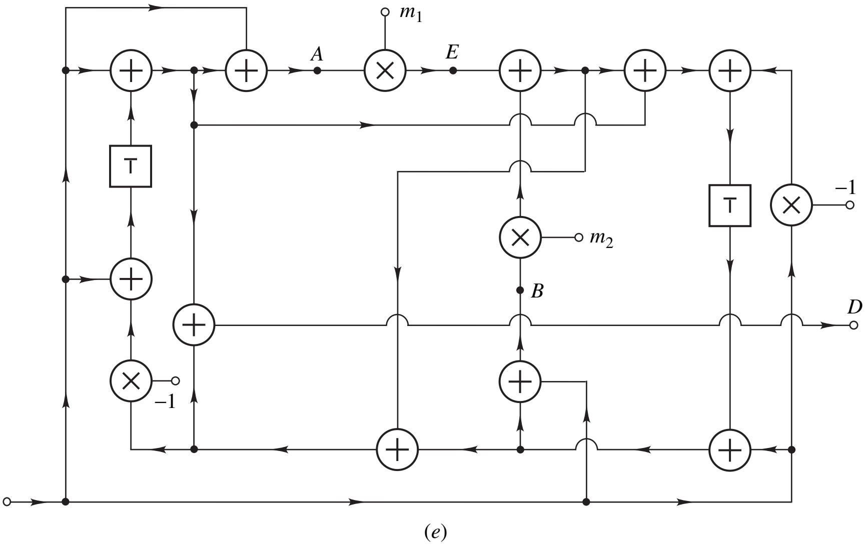


Figure 17.25 Cont'd Universal second-order CGIC sections: (*e*) Allpass.

■ **Solution**

The filter can be designed by cascading three LP sections of the type shown in Fig. 17.25*a*. The values of the multiplier constants can be readily evaluated as in columns 4 and 5 of Table 17.2. ■

Table 17.2 Lowpass-filter parameters (Example 17.4)

j	b_{0j}	b_{1j}	m_{1j}	m_{2j}
1	$1.069676E+6$	$5.353680E+2$	$-8.342350E-1$	$5.701500E-1$
2	$1.069676E+6$	$1.462653E+3$	$-8.650900E-1$	$2.778910E-1$
3	$1.069676E+6$	$1.998021E+3$	$-8.781810E-1$	$1.538890E-1$

17.12.5 Signal Scaling

Assuming a fixed-point implementation, the CGIC sections of Fig. 17.25 can be scaled by using Jackson's technique (see Sec. 14.6.2). For this purpose each of the five sections can be represented by the signal flow graph of Fig. 17.26*a*, where

$$H_A(z) = \frac{N_A(z)}{D(z)} \qquad H_B(z) = \frac{N_B(z)}{D(z)} \qquad \text{and} \qquad H_D(z) = \frac{N_D(z)}{D(z)}$$

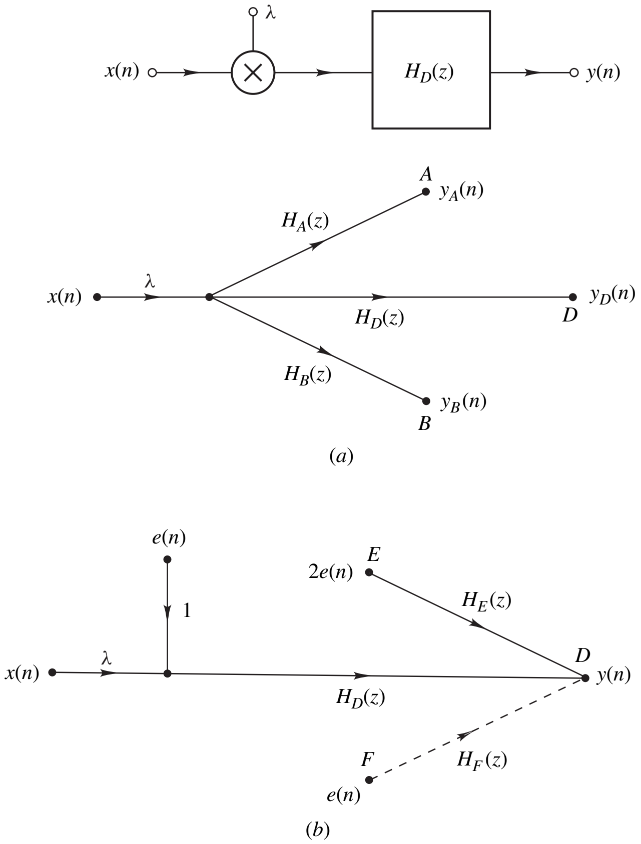


Figure 17.26 Universal CGIC sections: (*a*) Scaling model, (*b*) noise model.

are the transfer functions between section input and nodes A, B, and D, respectively. The above polynomials are given in Table 17.3. The optimum value of λ, for L_∞ scaling, is given by

$$\lambda = \frac{1}{\max\left[\|H_A(e^{j\omega T})\|_\infty, \|H_B(e^{j\omega T})\|_\infty, \|H_D(e^{j\omega T})\|_\infty\right]}$$

17.12.6 Output Noise

For the purpose of noise analysis, the five CGIC sections can be represented by the model of Fig. 17.26*b*, where $e(n)$ is the noise component generated by one multiplier and

$$H_E(z) = \frac{(z+1)(z-1)}{D(z)} \qquad H_F(z) = \frac{(1+m_1)(z+1)^2}{D(z)}$$

are the transfer functions between nodes E and F and section output. The dotted line in Fig. 17.26*b* applies to the N section only.

Table 17.3 Polynomials in CGIC sections

Type	$N_A(s)$	$N_D(z)$	$N_A(z)$	$N_B(z)$
LP	b_0	$(1+m_1)(z+1)^2$	$(z-m_2)(z+1)$	$(1+m_1)(z+1)$
HP	s^2	$(1+m_2)(z-1)^2$	$(z+m_1)(z-1)$	$-(1+m_2)(z-1)$
BP	$b_1 s$	$(m_1+m_2)(1-z^2)$	$-(z^2-2m_2z+1)$	$-(z^2+2m_1z+1)$
N	a_0+s^2	$k_0(1+m_1)(z+1)^2$ $+(1+m_2)(z-1)^2$	$k_0(z-m_2)(z+1)$ $-(1+m_2)(z-1)$	$k_0(1+m_1)(z+1)$ $+(z+m_1)(z-1)$
AP	$b_0-b_1s+s^2$	$2[(1+m_1+m_2)z^2$ $+(m_1-m_2)z+1]$	$2(z^2-2m_2z+1)$	$2(z^2+2m_1z+1)$
$D(z)=z^2+(m_1-m_2)z+(1+m_1+m_2)$				

On using the approach of Sec. 14.5 the *power spectral density* (*PSD*) of the output noise can be deduced as

$$S_o(e^{j\omega T}) = [|H_D(e^{j\omega T})|^2 + 2|H_E(e^{j\omega T})|^2 + |H_F(e^{j\omega T})|^2]S_e(e^{j\omega T})$$

where
$$S_e(e^{j\omega T}) = \frac{q^2}{12} \qquad \text{and} \qquad H_F(e^{j\omega T}) = 0$$

in all sections except for the N section in which

$$H_F(e^{j\omega T}) \neq 0$$

A useful property of the CGIC sections can be identified at this point. $H_E(z)$ is a bandpass transfer function in each of the five sections. As a consequence, noise generated by multipliers m_1 and m_2 will be attenuated at low as well as high frequencies, becoming zero at $\omega = 0$ as well as at $\omega = \omega_s/2$. By contrast, in the conventional canonic sections noise due to the multipliers is subjected to the same transfer function as the signal; e.g., in a lowpass section, the quantization noise is subjected to a lowpass transfer function. Because of this property, the CGIC synthesis tends to yield lowpass, highpass, and bandstop filters and also equalizers with improved in-band signal-to-noise ratio [11, 30]. In addition, like wave structures derived from passive networks, CGIC structures can be designed to be free of zero-input and overflow limit-cycle oscillations, as was demonstrated by Eswaran and Ganapathy [32].

17.13 CHOICE OF STRUCTURE

This chapter, like Chap. 8, has demonstrated that many distinct structures are possible for a given set of filter specifications. Hence one of the initial tasks of the filter designer is to choose a structure. The principal factors in this task are the sensitivity of the structure to coefficient quantization, the level of output roundoff noise, and the computational efficiency of the structure. As may be expected, these factors tend to depend to a large extent on the desired specifications; on the type of filter, i.e., lowpass, bandpass, etc.; on the type of approximation used, i.e., Butterworth, elliptic, etc.; on the type of arithmetic, i.e., fixed-point or floating-point; on the number system used, i.e., two's-complement, signed-magnitude, etc.; on the scaling norm used, i.e., L_2 or L_∞; and, in cascade structures, on the pairing of zeros and poles into second-order transfer functions, on the ordering of sections, and so on. Consequently, categorical statements about one or the other type of structure are difficult to make. Nevertheless, certain tendencies have been noted by researchers in the field, as follows:

1. High-order direct structures are very sensitive to coefficient quantization and also generate a high level of roundoff noise [33–35].
2. Cascade, parallel, and wave structures tend to have similar sensitivities for fixed-point arithmetic [30], [33–35].
3. Parallel structures tend to generate a lower level of roundoff noise than cascade structures [34, 38, 39]. However, they tend to be more sensitive to coefficient quantization at stopband frequencies because the zeros can wander off the unit circle.
4. For filters with zeros on the unit circle of the z plane, cascade canonic structures involve the lowest number of arithmetic operations.
5. The direct second-order structures described in Sec. 14.4 lead to low sensitivity and are, in addition, suitable for the application of error-spectrum shaping which can reduce the level of output roundoff noise quite significantly. However, signals must be scaled before and after each multiplication (see Ref. [9] of Chap. 14).
6. State-space structures can be designed to have minimum roundoff noise but require a large number of multiplications (see Refs. [15, 17] of Chap. 14).
7. Wave structures are significantly less sensitive than cascade structures for floating-point arithmetic [36, 37]. Similar results are expected for fixed-point arithmetic provided that scaling is applied before and after each multiplication. However, if the signal scaling constants are not powers of two, mismatch can arise at the inputs of adaptors, which can increase the sensitivity.
8. Lattice wave structures tend to be more sensitive to coefficient quantization at stopband frequencies than ladder wave structures.
9. CGIC cascade structures tend to yield improved in-band signal-to-noise ratio [11, 30].
10. State-space, wave, and CGIC structures can be designed to be free of limit-cycle oscillations.

It should be mentioned that the choice of structure involves many other issues besides the above, e.g., the suitability of the structure to the application at hand, the amenability of the structure to VLSI implementation (see Sec. 8.3), and the cost of the hardware. Also, in applications where very high sampling rates are employed, the degree of parallelism inherent in the various structures should be considered. In canonic structures, all multiplications can be performed simultaneously and, as a consequence, the time taken to complete the processing for one filter cycle can be nearly as short as the time taken to perform one multiplication. In wave structures, multiplications must be performed in sequence according to a certain hierarchy because of topological constraints, and hence the minimum time required to do the processing for one filter cycle can be much longer [30, 33].

REFERENCES

[1] A. Fettweis, "Digital filter structures related to classical filter networks," *Arch. Elektron. Übertrag.*, vol. 25, pp. 79–89, 1971.

[2] A. Fettweis, "Some principles of designing digital filters imitating classical filter structures," *IEEE Trans. Circuit Theory*, vol. 18, pp. 314–316, Mar. 1971.

[3] A. Fettweis, "Pseudopassivity, sensitivity, and stability of wave-digital filters," *IEEE Trans. Circuit Theory*, vol. 19, pp. 668–673, Nov. 1972.

[4] A. Sedlmeyer and A. Fettweis, "Digital filters with true ladder configuration," *Int. J. Circuit Theory Appl.*, vol. 1, pp. 5–10, Mar. 1973.

[5] R. Nouta, "The Jaumann structure in wave-digital filters," *Int. J. Circuit Theory Appl.*, vol. 2, pp. 163–174, June 1974.

[6] A. Fettweis, H. Levin, and A. Sedlmeyer, "Wave digital lattice filters," *Int. J. Circuit Theory Appl.*, vol. 2, pp. 203–211, June 1974.

[7] H. J. Orchard, "Inductorless filters," *Electron. Lett.*, vol. 2, pp. 224–225, June 1966.

[8] A. Fettweis and K. Meerkötter, "On adaptors for wave digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 23, pp. 516–525, Dec. 1975.

[9] A. Fettweis, "Wave digital filters: Theory and practice," *Proc. IEEE*, vol. 74, pp. 270–327, Feb. 1986.

[10] L. Gazsi, "Explicit formulas for lattice wave digital filters," *IEEE Trans. Circuits Syst.*, vol. 32, pp. 68–88, Jan. 1985.

[11] A. Antoniou and M. G. Rezk, "Digital-filter synthesis using concept of generalized-immittance convertor," *IEE J. Electron. Circuits Syst.*, vol. 1, pp. 207–216, Nov. 1977 (see vol. 2, p. 88, May 1978 for errata).

[12] C. Eswaran, V. Ganapathy, and A. Antoniou, "Wave digital biquads derived from RC-active configurations," *IEEE Trans. Circuits Syst.*, vol. 31, pp. 779–787, Sept. 1984.

[13] J. D. Rhodes, *Theory of Electrical Filters*, New York: Wiley, 1976.

[14] J. K. Skwirzynski, *Design Theory and Data for Electrical Filters*, London: Van Nostrand, 1965.

[15] R. Saal, *Handbook of Filter Design*, Backnang: AEG Telefunken, 1979.

[16] A. I. Zverev, *Handbook of Filter Synthesis*, New York: Wiley, 1967.

[17] E. Chirlian, *LC Filters: Design, Testing, and Manufacturing*, New York: Wiley, 1983.

[18] A. G. Constantinides, "Alternative approach to design of wave digital filters," *Electron. Lett.*, vol. 10, pp. 59–60, Mar. 1974. (See also *Proc. Inst. Elect. Eng.*, vol. 123, pp. 1307–1312, Dec. 1976.)

[19] M. N. S. Swamy and K. S. Thyagarajan, "A new type of wave digital filter," *J. Franklin Inst.*, vol. 300, pp. 41–58, July 1975.

[20] L. T. Bruton, "Network transfer functions using concept of frequency-dependent negative resistance," *IEEE Trans. Circuit Theory*, vol. 16, pp. 406–408, Aug. 1969.

[21] A. Fettweis, "Wave digital filters with reduced number of delays," *Int. J. Circuit Theory Appl.*, vol. 2, pp. 319–330, Dec. 1974.

[22] A. Fettweis and K. Meerkötter, "Suppression of parasitic oscillations in wave digital filters," *IEEE Trans. Circuits Syst.*, vol. 22, pp. 239–246, Mar. 1975.

[23] P. P. Vaidyanathan and S. K. Mitra, "Low passband sensitivity digital filters: A generalized viewpoint and synthesis procedures," *Proc. IEEE*, vol. 72, pp. 404–423, Apr. 1984.

[24] P. P. Vaidyanathan, "A unified approach to orthogonal digital filters and wave digital filters, based on LBR two-pair extraction," *IEEE Trans. Circuits Syst.*, vol. 32, pp. 673–686, July 1985.

[25] L. T. Bruton, "Low-sensitivity digital ladder filters," *IEEE Trans. Circuits Syst.*, vol. 22, pp. 168–176, Mar. 1975.

[26] L. T. Bruton and D. A. Vaughan-Pope, "Synthesis of digital ladder filters from LC filters," *IEEE Trans. Circuits Syst.*, vol. 23, pp. 395–402, June 1976.

[27] E. S. K. Liu, L. E. Turner, and L. T. Bruton, "Exact synthesis of LDI and LDD ladder filters," *IEEE Trans. Circuits Syst.*, vol. 31, pp. 369–381, Apr. 1984.

[28] B. D. Green and L. E. Turner, "Digital LDI filters using lattice equivalents and wave concepts," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 133–135, Jan. 1990.

[29] A. Antoniou, "Realisation of gyrators using operational amplifiers, and their use in RC-active-network synthesis," *Proc. Inst. Elect. Eng.*, vol. 116, pp. 1838–1850, Nov. 1969.

[30] A. Antoniou and M. G. Rezk, "A comparison of cascade and wave fixed-point digital-filter structures," *IEEE Trans. Circuits Syst.*, vol. 27, pp. 1184–1194, Dec. 1980.

[31] A. Antoniou, "Novel RC-active-network synthesis using generalized-immittance converters," *IEEE Trans. Circuit Theory*, vol. 17, pp. 212–217, May 1970.

[32] C. Eswaran and V. Ganapathy, "On the stability of digital filters designed using the concept of generalized-immittance convertor," *IEEE Trans. Circuits Syst.*, vol. 28, pp. 745–747, July 1981.

[33] R. E. Crochiere and A. V. Oppenheim, "Analysis of linear digital networks," *Proc. IEEE*, vol. 63, pp. 581–595, Apr. 1975.

[34] W. K. Jenkins and B. J. Leon, "An analysis of quantization error in digital filters based on interval algebras," *IEEE Trans. Circuits Syst.*, vol. 22, pp. 223–232, Mar. 1975.

[35] J. L. Long and T. N. Trick, "Sensitivity and noise comparison of some fixed-point recursive digital filter structures," *in Proc. IEEE Int. Symp. Circuits and Systems,* pp. 56–59, 1975.

[36] R. E. Crochiere, "Digital ladder structures and coefficient sensitivity," *IEEE Trans. Audio Electroacoust.*, vol. 20, pp. 240–246, Oct. 1972.

[37] W. H. Ku and S.-M. Ng, "Floating-point coefficient sensitivity and roundoff noise of recursive digital filters realized in ladder structures," *IEEE Trans. Circuits Syst.*, vol. 22, pp. 927–936, Dec. 1975.

[38] L. B. Jackson, "Roundoff-noise analysis for fixed-point digital filters realized in cascade or parallel form," *IEEE Trans. Audio Electroacoust.*, vol. 18, pp. 107–122, June 1970.

[39] L. B. Jackson, "Roundoff noise bounds derived from coefficient sensitivities for digital filters," *IEEE Trans. Circuits Syst.*, vol. 23, pp. 481–485, Aug. 1976.

PROBLEMS

17.1. Figure P17.1 depicts an equally terminated LC filter.

(*a*) Obtain a signal flow graph for the filter by applying Kirchhoff's voltage and current laws.

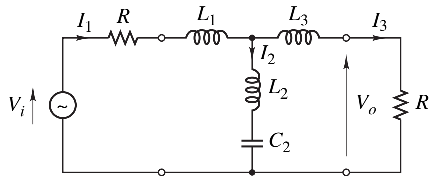


Figure P17.1

(*b*) Show that the application of the bilinear transformation to the signal flow graph of part (*a*) leads to at least one delay-free loop.

17.2. Figure P17.2 represents an independent current source with an internal impedance $Z(s) = s^{\lambda}R_x$. Obtain digital realizations for $\lambda = -1, 0$, and 1 if $R = (2/T)^{\lambda}R_x$.

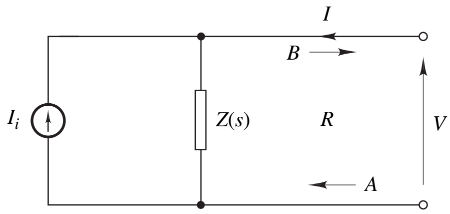


Figure P17.2

17.3. Analyze the series adaptors of Fig. 17.6*a* and *b*.

17.4. Analyze the parallel adaptors of Fig. 17.7*a* and *b*.

17.5. Show that a 2-port series adaptor and a 2-port parallel adaptor are interrelated by the equivalence of Fig. P17.5.

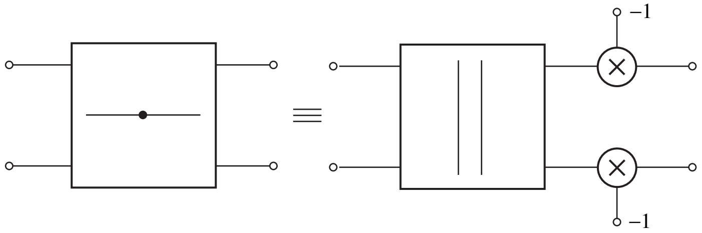


Figure P17.5

17.6. (*a*) Analyze the 2-port parallel adaptor of Fig. 17.8*b*.

(*b*) Obtain an alternative 1-multiplier 2-port parallel adaptor.

17.7. Show that a 2-port parallel adaptor is equivalent to the realization of a transformer with a turns ratio $k = 1$.

17.8. The 2-port of Fig. P17.8, where $V_1 = -RI_2$ and $V_2 = RI_1$, represents a gyrator circuit. Obtain a corresponding digital realization.

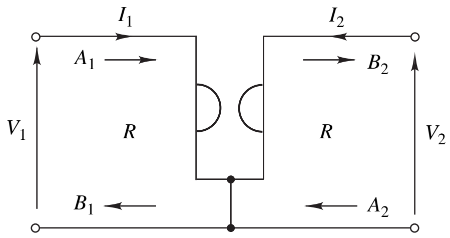


Figure P17.8

17.9. A 2-port in which the input impedance Z_i is related to the load impedance Z_L by $Z_i = -kZ_L$ is said to be a *negative-impedance converter* (NIC). The parameter k is referred to as the *impedance-conversion* *factor* of the device. Two types of NICs can be identified, namely, voltage-conversion NICs, in which

$$V_1 = -kV_2 \qquad I_1 = -I_2$$

and current-conversion NICs, in which

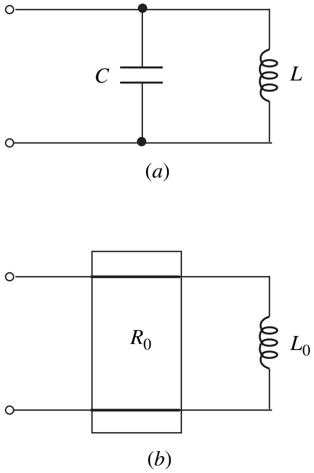
$$V_1 = V_2 \qquad I_1 = kI_2$$

Derive digital realizations for each case if port resistances R_1 and R_2 are assigned to the input and output ports, respectively.

17.10. (*a*) Obtain a digital realization of the parallel resonant circuit of Fig. P17.10*a* using a parallel adaptor.

(*b*) Show that the resonant circuit of Fig. P17.10*a* can be realized by a unit element terminated by an inductance L_0 as depicted in Fig. P17.10*b*. Obtain expressions for R_0 and L_0.

(*c*) Obtain a digital realization of the resonant circuit in part (*b*).



(*a*)

(*b*)

Figure P17.10

17.11. Obtain a digital realization for the transfer function

$$H(s) = \frac{24s^4 - 24s^3 + 20s^2 - 2s + 1}{24s^4 + 24s^3 + 20s^2 + 2s + 1}$$

17.12. Derive the wave characterization of the lattice network given in Fig. 17.14*a*.

17.13. A fifth-order Chebyshev lowpass filter with passband edge $\omega_p = 1$ rad/s and passband ripple 1 dB is characterized by the transfer function

$$H(s) = \frac{0.1228}{(s + 0.2895)(s^2 + 0.4684s + 0.4293)(s^2 + 0.1789s + 0.9883)}$$

Assuming a sampling frequency of 8 rad/s, obtain a lattice realization for the filter.

17.14. An application calls for a highpass digital filter that would satisfy the following specifications:

$$A_p = 1.0 \text{ dB} \qquad A_a = 45.0 \text{ dB} \qquad \tilde{\Omega}_p = 3.5 \text{ rad/s} \qquad \tilde{\Omega}_a = 1.5 \text{ rad/s}$$

The sampling frequency ω_s is to be 10 rad/s.

(*a*) Obtain the required transfer function using a Butterworth approximation.

(*b*) Realize the transfer function obtained in part (*a*) using a wave lattice structure.

17.15. Repeat parts (*a*) and (*b*) of Prob. 17.14 using an elliptic approximation.

17.16. Figure P17.16 shows an elliptic lowpass filter. Obtain a corresponding wave structure, assuming a sampling frequency $\omega_s = 10$ rad/s.

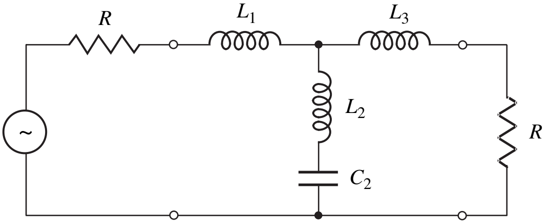


Figure P17.16 $L_1 = L_3 = 3.0316$ H $L_2 = 0.21286$ H $C_2 = 1.4396$ F $R = 1\ \Omega$

17.17. Figure P17.17 shows an elliptic highpass filter satisfying the following specifications:

$$A_p = 0.5 \text{ dB} \qquad A_a = 31.2 \text{ dB} \qquad \omega_p = 1/\sqrt{0.5} \text{ rad/s} \qquad \omega_a = \sqrt{0.5} \text{ rad/s}$$

(*a*) Obtain a corresponding wave digital filter, assuming that $\omega_s = 10$ rad/s.

(*b*) Determine the resulting passband and stopband edges.

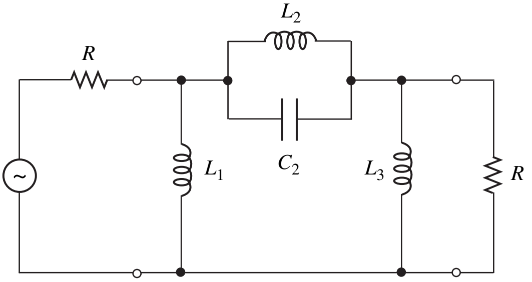


Figure P17.17 $L_1 = L_3 = 0.48948$ H $L_2 = 3.4132$ H $C_2 = 0.75489$ F $R = 1\ \Omega$

17.18. An analog bandpass filter can be obtained by applying the lowpass-to-bandpass transformation

$$s = \frac{1}{10}\left(\bar{s} + \frac{625}{\bar{s}}\right)$$

to the lowpass filter of Fig. P17.16. Derive a corresponding wave digital filter if $\omega_s = 250$ rad/s.

17.19. Design a lowpass digital filter that would satisfy the specifications

$$A_p = 1.0 \text{ dB} \qquad A_a \geq 60.0 \text{ dB} \qquad \tilde{\Omega}_p \approx 100 \text{ rad/s} \qquad \tilde{\Omega}_a \approx 200 \text{ rad/s}$$

assuming a sampling frequency of 1000 rad/s. Use an elliptic approximation and a wave ladder realization. (Hint: Use the tables in Ref. [14] or one of the available filter-design software packages.)

17.20. By applying the impedance transformation $Z(s) \to Z(s)/s$ to the filter of Fig. P17.16, derive a corresponding FDNR wave digital filter.

17.21. Repeat Prob. 17.20 using the highpass filter of Fig. P17.17 as a prototype.

17.22. The multiplier constants for the filter of Fig. 17.17*c* are given in Table P17.22. Compute the amplitude response of the filter if $\omega_s = 10$ rad/s.

Table P17.22

Adaptor	Multiplier constants
1	$m_{p1} = 1.341381E - 1$ $m_{p2} = 1.0$
2	$m_{s1} = 9.615504E - 2$ $m_{s2} = 1.0$
3	$m_{p1} = 1.720167E - 1$ $m_{p2} = 2.399664E - 1$
4	$m_{p1} = 2.436145E - 1$ $m_{p2} = 1.0$

17.23. Compute the amplitude response of the digital filter depicted in Fig. P17.23, assuming that $\omega_s = 10$ rad/s. The values of the multiplier constants are given in Table P17.23.

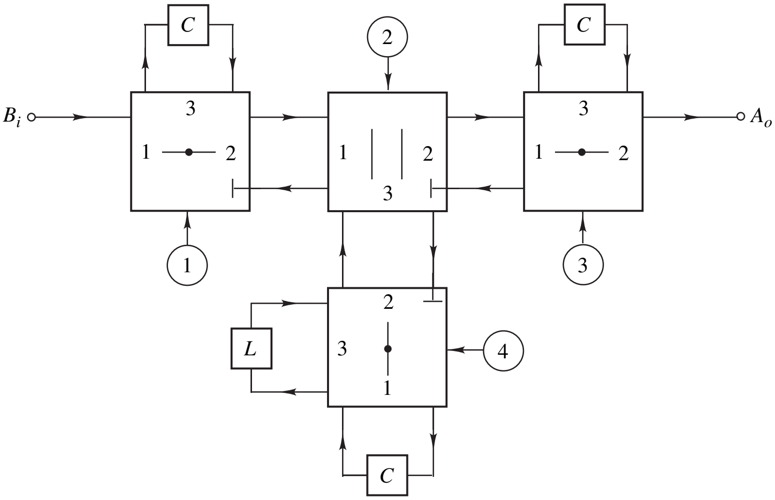


Figure P17.23

Table P17.23

Adaptor	Multiplier constants
1	$m_{s1} = 5.846557E - 1$ $m_{s2} = 1.0$
2	$m_{p1} = 4.307685E - 1$ $m_{p2} = 1.0$
3	$m_{s1} = 6.021498E - 1$ $m_{s2} = 8.172611E - 1$
4	$m_{s1} = 1.466151E - 2$ $m_{s2} = 1.0$

17.24. Obtain a digital realization for a general 2-port represented by Eqs. (17.14a) and (17.14b).

17.25. (*a*) Obtain a digital realization for the 2-port of Fig. P17.25*a*, assuming that $R_1 = R_2 + L$ and $\omega_s = \pi$ rad/s.

(*b*) Repeat part (*a*) for the 2-port of Fig. P17.25*b* if $G_1 = G_2 + C$ and $\omega_s = \pi$ rad/s.

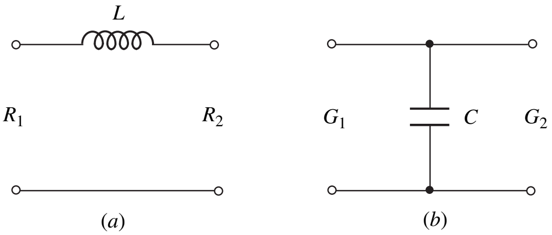


Figure P17.25

17.26. (*a*) Show that the analog filter of Fig. P17.26, where $R = 1\ \Omega$, $C = 1$ F, and $L = 2$ H, represents a 3rd-order Butterworth filter.

(*b*) Derive a corresponding digital filter using the realizations of the inductor and capacitor obtained in Prob. 17.25.

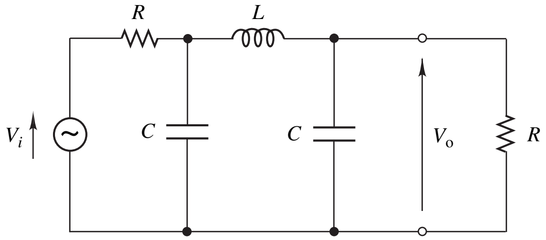


Figure P17.26

17.27. Analyze the configuration of Fig. 17.24*a*.

17.28. (*a*) Derive the lowpass section of Fig. 17.25*a*.

(*b*) Derive the highpass section of Fig. 17.25*b*.

17.29. An analog highpass filter is characterized by

$$H(s) = \prod_{j=1}^{3} \frac{s^2}{b_{0j} + b_{1j}s + s^2}$$

where $b_{01} = b_{02} = b_{03} = 31.15762 \qquad b_{11} = 10.78340$

$b_{12} = 7.8940 \qquad b_{13} = 2.889405$

Obtain a corresponding digital filter by using the CGIC synthesis, assuming that $\omega_s = 10$ rad/s.

17.30. Design a CGIC digital lowpass filter satisfying the following specifications:

$$A_p = 0.5\text{ dB} \qquad A_a = 65\text{ dB} \qquad \tilde{\Omega}_p = 200\text{ rad/s}$$

$$\tilde{\Omega}_a = 300\text{ rad/s} \qquad \omega_s = 1000\text{ rad/s}$$

Use an elliptic approximation.

This page intentionally left blank

CHAPTER 18

DIGITAL SIGNAL PROCESSING APPLICATIONS

18.1 INTRODUCTION

In the past several chapters, some sophisticated design methods have been described in detail. In this chapter, we consider a number of extensions of these methods and some of their numerous applications to digital signal processing.

The chapter begins with the underlying principles involved when the sampling frequency is changed from one value to another, and devices that can be used for the conversion, known as *decimators* and *interpolators*, are described [1]. The application of decimators and interpolators in the design of *filter banks* [2–4] is then considered. These subsystems find widespread applications in communications systems, spectrum analyzers, and speech synthesis. The principles involved are examined and a specific type of filter bank, the so-called *quadrature-mirror-image filter bank*, is examined in some detail. Another useful device for communications applications is the *Hilbert transformer*. Its theoretical basis, design principles, and applications to the sampling of bandpassed signals and to single-sideband modulation are described in Sec. 18.4.

Another topic that is receiving considerable attention these days is the design and applications of *adaptive filters*. These are filters that have an adaptation mechanism which allows them to change their characteristics with time in order to satisfy some performance requirement. The topic is a field in its own right, and several textbooks have been written on the theory and design of these filters [5–8]. In Sec. 18.5, we examine some of the fundamentals involved and consider some of the algorithms that are being used as adaptation algorithms.

In many applications, the signal to be processed is a function of two discrete variables. Signals of this type can be processed by *two-dimensional* (2-D) digital filters. The methods involved are usually extensions of their one-dimensional (1-D) counterparts, but once in a while there are marked differences. As in the case of adaptive filters, the topic of 2-D digital filters is emerging as an important field of study, and some textbooks have already been written on the subject [9–11]. For the sake of completeness, we conclude this chapter by examining some of the basic principles involved in the analysis and design of these filters.

18.2 SAMPLING-FREQUENCY CONVERSION

For various technical reasons, it may be preferable to store or record a discrete-time signal using one sampling frequency but process or transmit it using another sampling frequency. The conversion of the sampling frequency from one value to another can be accomplished in an indirect way by converting a discrete-time signal back to a continuous-time using a D/A converter then sampling the latter at a different rate using an A/D converter. However, each such conversion introduces quantization noise and $\sin\omega/\omega$ distortion, as well as other problems. An alternative and more satisfactory approach is to carry out the conversion in the discrete-time domain using a class of digital filters known as *decimators* and *interpolators*.

Decimators can be used to reduce the sampling frequency, whereas interpolators can be used to increase it. This section deals with the underlying principles involved in connection with sampling-frequency conversion in the discrete-time domain and highlights some design aspects concerning decimators and interpolators.

18.2.1 Decimators

Let us first examine the situation where the sampling frequency of a discrete-time signal $x(nT)$ is to be reduced.

If $x(nT)$ is deemed to have been obtained by sampling a continuous-time signal $x(t)$ using a sampling frequency ω_s, then the frequency spectrum of $x(nT)$ can be readily obtained from Eq. (6.45a) as[1]

$$X(e^{j\omega T}) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X_c[j(\omega - n\omega_s)] \tag{18.1}$$

where

$$X_c(j\omega) = \mathcal{F}x(t) \qquad \text{and} \qquad X(z) = \mathcal{Z}x(nT)$$

Let the new sampling frequency be $\omega_s' = \omega_s/M$ where M is an integer greater than unity. Since the sampling period is increased from T to $T' = MT$, a signal $x_d(nT')$ is obtained which is related to

[1] *The notation has been changed somewhat for the sake of convenience.*

$x(nT)$ by the equation

$$x_d(nT') \equiv x(nT') = x(nMT)$$

That is, $x_d(nT')$ is obtained by retaining only samples $\ldots, -2M, -M, 0, M, 2M, \ldots$ of discrete-time signal $x(nT)$ as shown in Fig. 18.1 for the case where $M = 3$. This process has been referred to in the past as *downsampling, sampling-frequency compression*, or *discrete-time sampling* [1]. A device that can perform this operation is said to be a *downsampler, sampling-frequency compressor*, or simply *compressor* and is often represented by the block diagram depicted in Fig. 18.2a.

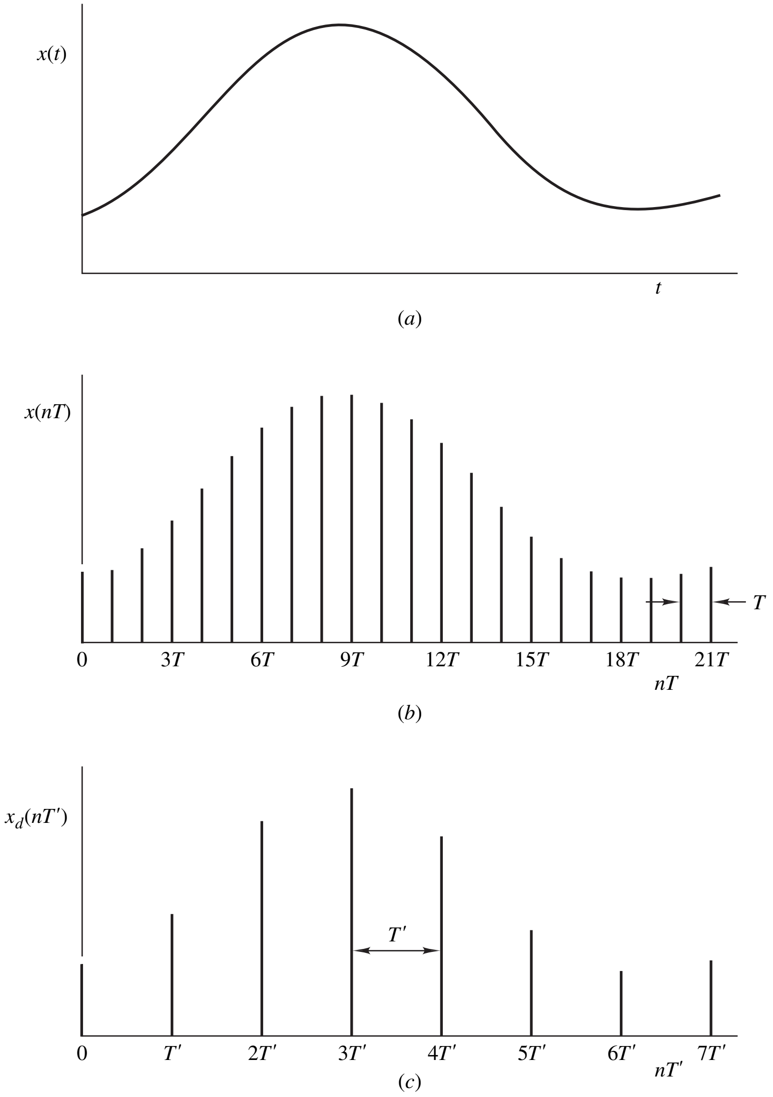


Figure 18.1 The process of downsampling.

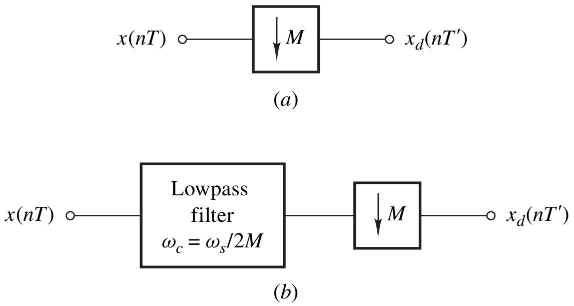


Figure 18.2 *(a)* Downsampler, *(b)* decimator.

By analogy with Eq. (18.1), the spectrum of signal $x_d(nT')$ can be readily expressed as

$$\begin{aligned} X_d(e^{j\omega T'}) &= \frac{1}{T'} \sum_{n=-\infty}^{\infty} X_c[j(\omega - n\omega_s')] \\ &= \frac{1}{MT} \sum_{n=-\infty}^{\infty} X_c \left[j \left(\omega - n\frac{\omega_s}{M} \right) \right] \end{aligned} \tag{18.2}$$

i.e., *the spectrum of $x_d(nT')$ can be regarded as an infinite sum of shifted copies of the spectrum of $x(t)$ divided by MT*, where each copy is shifted by a multiple of ω_s/M. Alternatively, Eq. (18.2) can be expressed as

$$X_d(e^{j\omega T'}) = \frac{1}{MT} \sum_{n=-\infty}^{\infty} \sum_{m=0}^{M-1} X_c \left[j \left(\omega - n\omega_s - m\frac{\omega_s}{M} \right) \right]$$

since $X_d(e^{j\omega T'})$ can be considered to be periodic with period ω_s, and hence

$$\begin{aligned} X_d(e^{j\omega T'}) &= \frac{1}{M} \sum_{m=0}^{M-1} \left\{ \frac{1}{T} \sum_{n=-\infty}^{\infty} X_c \left[j \left(\omega - n\omega_s - m\frac{\omega_s}{M} \right) \right] \right\} \\ &= \frac{1}{M} \sum_{m=0}^{M-1} X(e^{j(\omega - m\omega_s/M)T}) \end{aligned} \tag{18.3}$$

According to this representation, *the spectrum of $x_d(nT)$ may be deemed to be the sum of M copies of the spectrum of $x(nT)$ divided by M*, where each copy is shifted by a multiple of ω_s/M.

The above conversion process is illustrated in Fig. 18.3 for the case where $M = 3$ and $X_c(j\omega)$ is real. As can be seen, if $X_c(j\omega) = 0$ for $|\omega| \geq \omega_s/2M$, then the copies of the spectrum of $x(nT)$ produced by downsampling do not overlap and, as a consequence, $x(t)$ can be recovered by using an ideal lowpass filter with cutoff frequency $\omega_s/2M$. If the aforementioned requirement is violated, then aliasing will be introduced, as detailed in Sec. 6.7. To avoid this possibility, a bandlimiting filter with cutoff frequency $\omega_c = \omega_s/2M$ is usually used at the input of the downsampler, as depicted in Fig. 18.2*b*. The configuration obtained is said to be a *decimator*.

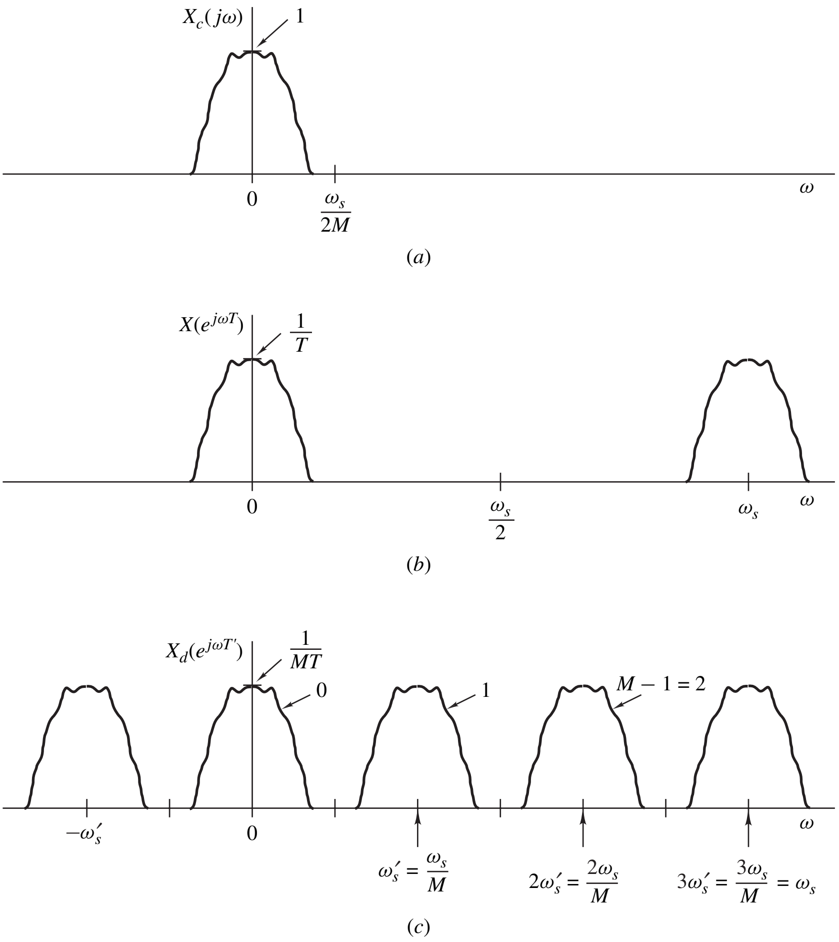


Figure 18.3 Operation of downsampler.

18.2.2 Interpolators

The preceding section has shown that sampling-frequency reduction can be achieved by applying downsampling which is analogous to the sampling of a continuous-time signal. We now show that a sampling-frequency increase can be achieved through a process which is analogous to converting a discrete-time signal into a continuous-time signal.

A sampling-frequency increase from ω_s to $L\omega_s$, where L is an integer, can be achieved by using a device known as an *upsampler*, *sampling-frequency expander*, or simply *expander* in conjunction with a lowpass filter [1]. An upsampler is usually represented by the block diagram depicted in Fig. 18.4a, and its response to an excitation $x(nT)$, designated as $x_u(nT)$, can be expressed as

$$x_u(nT') = \begin{cases} x(nT/L) & \text{for } n = 0, \pm L, \pm 2L, \ldots \\ 0 & \text{otherwise} \end{cases} \tag{18.4}$$

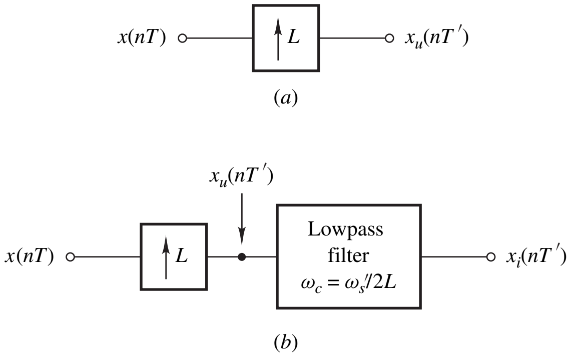


Figure 18.4 (*a*) Upsampler, (*b*) interpolator.

where $T' = T/L$. This process is illustrated in Fig. 18.5 for the case where $L = 3$. Equation (18.4) can be written as

$$x_u(nT') = \sum_{k=-\infty}^{\infty} x(kT)\delta(nT - kLT)$$

and by applying the z transform, we obtain

$$\begin{aligned} X_u(e^{j\omega T'}) &= \sum_{n=-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} x(kT)\delta(nT - kLT) \right] e^{-j\omega nT'} \\ &= \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(kT)[\delta(nT - kLT)e^{-j\omega nT/L}] \\ &= \sum_{k=-\infty}^{\infty} x(kT)e^{-j\omega kT} = X(e^{j\omega T}) \end{aligned} \tag{18.5}$$

In effect, *the frequency spectrums of* $x_u(nT')$ *and* $x(nT)$ *are identical*, as illustrated in Fig. 18.6*a* and *b* for the case where $L = 3$. Since $T' = T/L$, or $\omega_s' = L\omega_s$, there are L *images* of the signal in the baseband $-\omega_s'/2 \le \omega \le \omega_s'/2$, that is, upsampling will simply change the location of the sampling frequency as shown.

A discrete-time version of $x(nT)$, sampled at the increased rate of $\omega_s' = L\omega_s$, can now be generated by filtering signal $x_u(nT')$ using an ideal lowpass digital filter with a frequency response

$$H(e^{j\omega T'}) = \begin{cases} L & \text{for } 0 < |\omega| < \omega_c \\ 0 & \text{otherwise} \end{cases} \tag{18.6a}$$

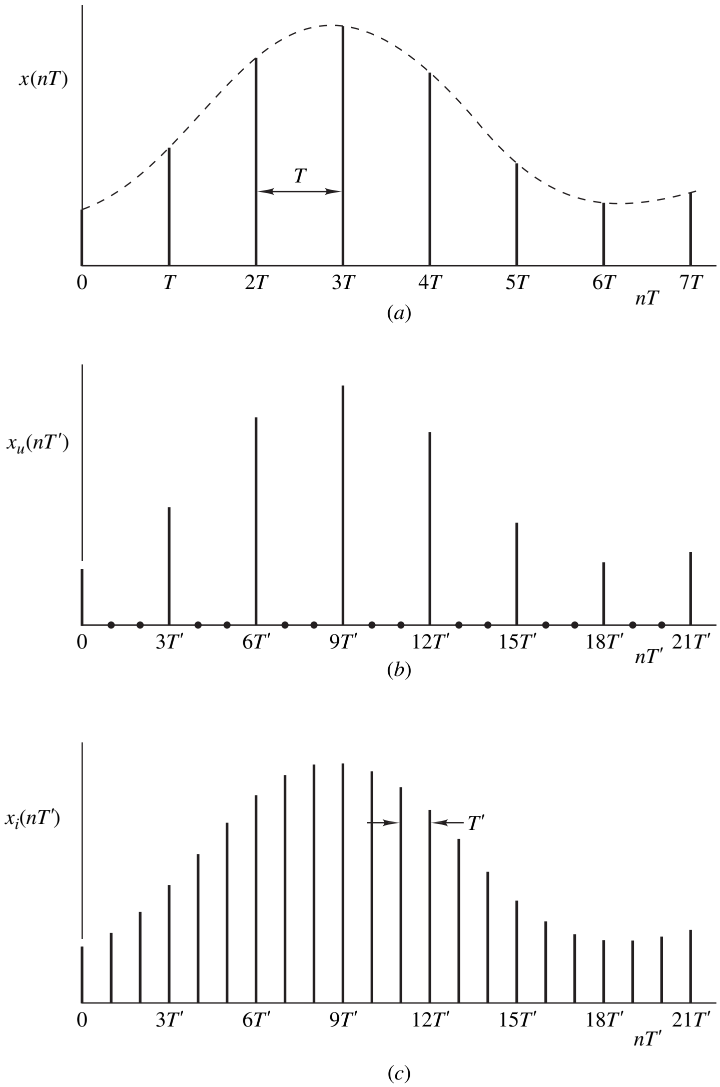
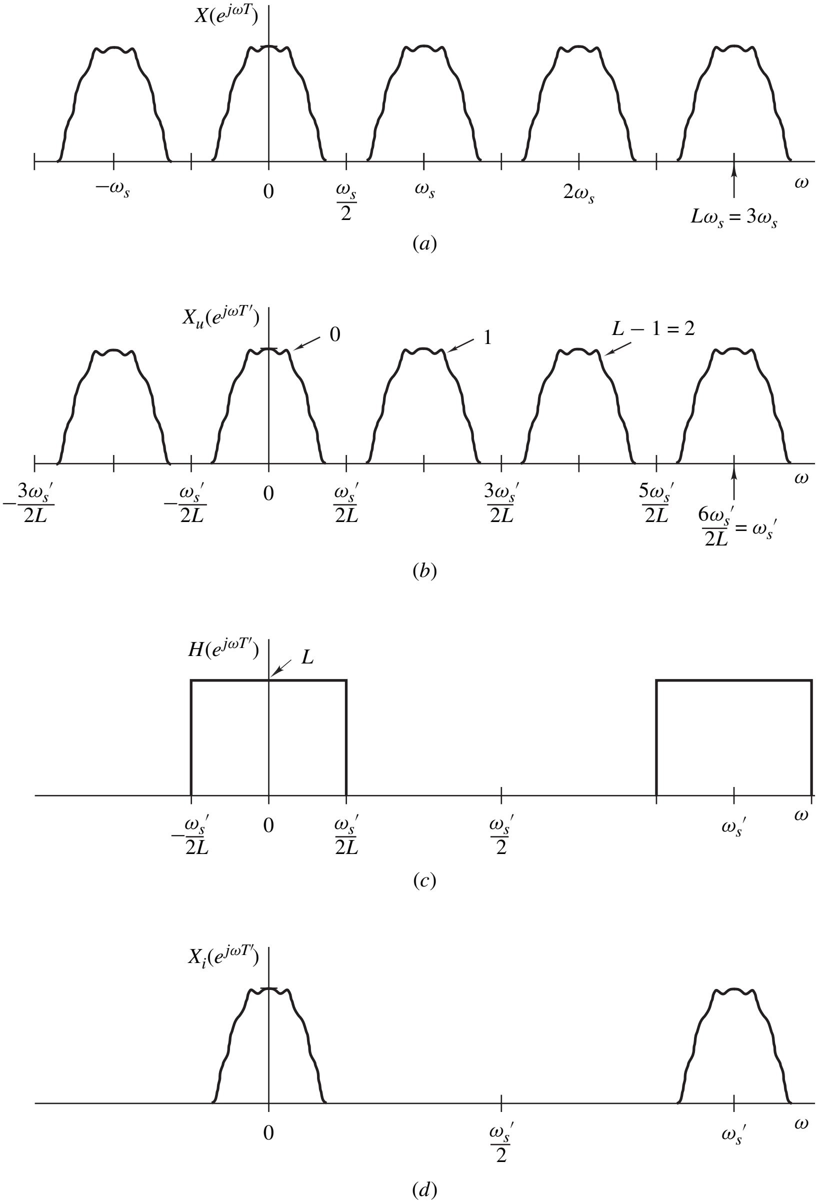


Figure 18.5 The process of upsampling.

where $\omega_c = \omega_s'/2L$, as illustrated in Fig. 18.4*b*. This filter will reject unnecessary images and yield a discrete-time signal $x_i(nT') = x(nT')$, as illustrated in Fig. 18.6*c* and *d*. The configuration of Fig. 18.4*b* is said to be an *interpolator*.

The output of the lowpass filter in Fig. 18.4*b* can be shown to be an interpolated version of $x(nT)$ by finding the time-domain response of the ideal lowpass filter. We can write,

$$X_i(z) = H(z)X_u(z) \tag{18.6b}$$



(*a*)

(*b*)

(*c*)

(*d*)

Figure 18.6 Operation of upsampler and interpolator.

where $H(z)$ is the transfer function of the lowpass filter and since $T = LT'$. From Eq. (18.5), we have

$$\begin{aligned}
X_u(z) &= \mathcal{Z}x_u(nT') \\
&= \sum_{k=-\infty}^{\infty} x(kT)e^{-j\omega kLT'}\Bigg|_{e^{j\omega T'}=z} \\
&= \sum_{k=-\infty}^{\infty} x(kT)z^{-kL}
\end{aligned} \tag{18.6c}$$

and hence Eq. (18.6b) yields

$$\begin{aligned}
x_i(nT') &= \mathcal{Z}^{-1}[H(z)X_u(z)] \\
&= \mathcal{Z}^{-1}\left[H(z)\sum_{k=-\infty}^{\infty} x(kT)z^{-kL}\right] \\
&= \sum_{k=-\infty}^{\infty} x(kT)\mathcal{Z}^{-1}[H(z)z^{-kL}]
\end{aligned} \tag{18.7}$$

Now a digital filter with a frequency response $H(e^{j\omega T'})$ as defined in Eq. (18.6a) has an impulse response

$$\begin{aligned}
h(nT') &= \frac{L}{n\pi}\sin\omega_c nT' = \frac{L}{n\pi}\sin\frac{\omega_s' nT'}{2L} \\
&= \frac{\sin \pi n/2L}{\pi n/2L}
\end{aligned}$$

(see Example 9.1) and from Eq. (18.7) and the complex-convolution theorem (Theorem 3.10), $x_i(nT')$ can be expressed as

$$\begin{aligned}
x_i(nT') &= \sum_{k=-\infty}^{\infty} x(kT)h[(n-kL)T'] \\
&= \sum_{k=-\infty}^{\infty} x(kT)\frac{\sin[\pi(n-kL)/L]}{\pi(n-kL)/L}
\end{aligned} \tag{18.8}$$

Therefore,

$$x_i(nT') = \begin{cases} x(nT) & \text{if } k = n \\ 0 & \text{otherwise} \end{cases}$$

since

$$\frac{\sin[\pi(n-kL)/L]}{\pi(n-kL)/L} = \begin{cases} 1 & \text{if } k = n \\ 0 & \text{otherwise} \end{cases}$$

The above analysis is analogous to the use of a lowpass filter to recover a continuous-time signal from an impulse-modulated signal as discussed in Sec. 6.6, that is, the relation in Eq. (18.8) is analogous to the interpolation formula in Eq. (6.51).

Example 18.1 Show that the use of a digital filter characterized by the impulse response

$$h(nT') = \begin{cases} 1 - |n|/L & \text{for } |n| < L \\ 0 & \text{otherwise} \end{cases} \tag{18.9}$$

in the scheme of Fig. 18.4*b* will result in a system that can perform linear interpolation.

■ **Solution**

The response of the filter at nT' to an excitation $x_u(nT')$ is given by the convolution summation as

$$x_i(n) = \sum_{k=-\infty}^{\infty} x_u(k)h(n-k)$$

where period T' is dropped for the sake of brevity. The response at $(mL+\lambda)T'$, where λ is an integer in the range $0 \le \lambda \le L$, is given by

$$\begin{aligned} x_i(mL+\lambda) &= \sum_{k=-\infty}^{\infty} x_u(k)h(mL+\lambda-k) \\ &= \cdots + x_u(mL-1)h(1+\lambda) + x_u(mL)h(\lambda) \\ &\quad + x_u(mL+1)h(\lambda-1) + \cdots \\ &\quad + x_u[(m+1)L-1]h[\lambda-(L-1)] \\ &\quad + x_u[(m+1)L]h(\lambda-L) \\ &\quad + x_u[(m+1)L+1]h[\lambda-(L+1)] + \cdots \end{aligned} \tag{18.10}$$

From Eq. (18.4), $x_u(n) = 0$ if n is not a multiple of L; therefore, for $0 < \lambda < L$, Eqs. (18.9) and (18.10) yield

$$\begin{aligned} x_i(mL + \lambda) &= x_u(mL)h(\lambda) + x_u[(m+1)L]h(\lambda - L) \\ &= \left(1 - \frac{\lambda}{L}\right) x_u(mL) + \frac{\lambda}{L} x_u[(m+1)L] \end{aligned}$$

which is the linear interpolation between samples mL and $(m+1)L$.

18.2.3 Sampling-Frequency Conversion by a Noninteger Factor

Through the use of an interpolator, the sampling frequency can be increased from ω_s to $\omega_s' = L\omega_s$. On the other hand, through the use of a decimator, the sampling frequency can be decreased from ω_s' to $\omega_s'' = \omega_s'/M$. By cascading an interpolator and a decimator, as depicted in Fig. 18.7, a sampling frequency $\omega_s'' = L\omega_s/M$ can be obtained, where L/M is a ratio of integers. By this means, arbitrary conversion factors can be achieved. In Fig. 18.7, the interpolator necessitates a lowpass filter with a cutoff frequency $\omega_s'/2L$, while in the decimator a lowpass filter with a cutoff frequency $\omega_s'/2M$ is needed. One of the two filters, namely, the one with the higher cutoff frequency, is obviously redundant and can be eliminated.

18.2.4 Design Considerations

The design of interpolators and decimators is straightforward, and perhaps the most complicated part of the design involves the design of the required filters. Depending on the system requirements, a recursive or nonrecursive filter may be more appropriate, and any one of the methods described in Chaps. 9, 12, 15, and 16 can be used. Furthermore, a large variety of filter structures can be employed. A comparison between recursive and nonrecursive designs can be found in Sec. 11.8. The merits and demerits of the various types of structures can be found in Sec. 17.13.

18.3 QUADRATURE-MIRROR-IMAGE FILTER BANKS

Filter banks find applications in many areas of science and engineering and are used for time-division to frequency-division multiplex translation, subband speech coding, bandwidth compression, and many other types of signal processing [1, 2, 4]. In these systems, a given signal is decomposed

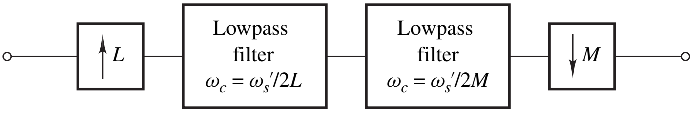


Figure 18.7 Sampling-frequency conversion by noninteger factor.

into consecutive subbands that are processed independently. The subbands obtained are then used to synthesize a processed version of the signal.

The motivation behind this roundabout way of processing a signal is usually to achieve some economical advantage, e.g., increased utilization of equipment or improved signal-to-noise ratio, but on occasion a filter bank is used to carry out some type of processing that cannot be carried out by any other means. Time-division to frequency-division multiplex translation is used quite extensively in communications systems in order to achieve increased channel capacity [12]. In subband speech coding, the signal is decomposed into several bands that are coded individually, taking advantage of certain perceptive properties of the human ear. In this way, improved signal quality can be achieved without increasing the bit rate [13].

In its most general form, a filter bank consists of a *decomposition* or *analysis* section and a *reconstruction* or *synthesis* section, as depicted in Fig. 18.8. Depending on the application, (1) the spectrums of $x_0(nT)$, $x_1(nT)$, ... , $x_K(nT)$ should not overlap, may overlap somewhat, or can overlap extensively; (2) the processed signal $y(nT)$ must be a faithful reproduction of $x(nT)$ or it may be a transformed version that bears little resemblance; and (3) the subband widths may be uniform, irregular, or fixed by some logarithmic relation.

In this section, we consider a specific type of filter bank, the so-called *quadrature-mirror-image filter* (QMF) *bank*.

18.3.1 Operation

A 2-band QMF bank is illustrated in Fig. 18.9. The analysis section of the bank consists of a lowpass and a highpass filter which decompose the input signal $x(nT)$ into two components $x_0(nT)$ and

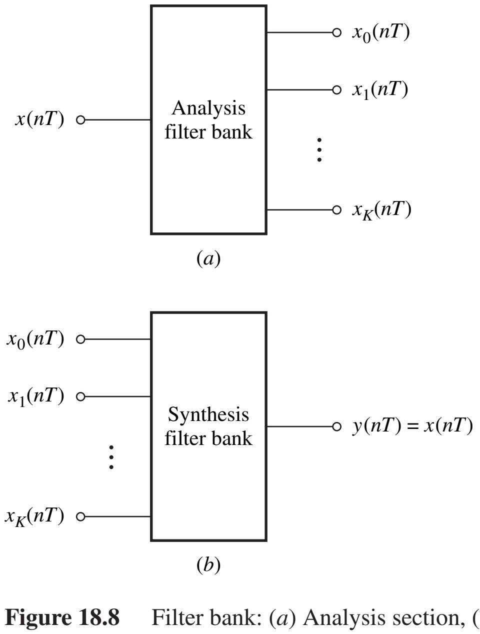


Figure 18.8 Filter bank: (*a*) Analysis section, (*b*) synthesis section.

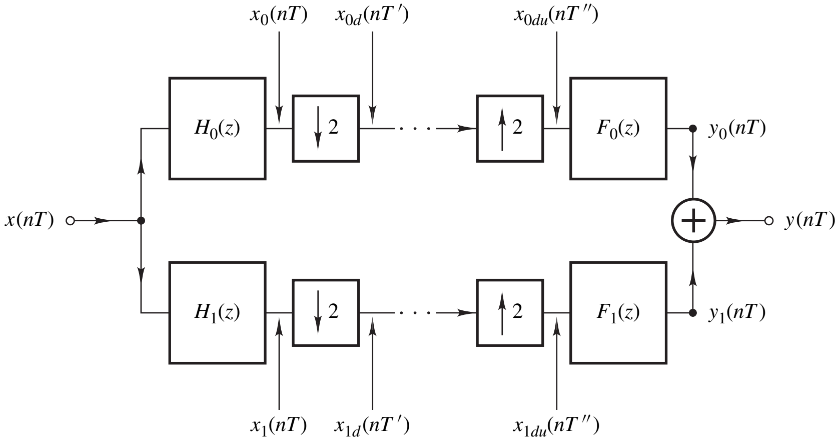


Figure 18.9 2-band QMF bank.

$x_1(nT)$. Ideally, the lowpass and highpass filters should have the frequency responses

$$H_0(e^{j\omega T}) = \begin{cases} 1 & \text{for } 0 < |\omega| < \omega_s/4 \\ 0 & \text{otherwise} \end{cases} \tag{18.11}$$

and

$$H_1(e^{j\omega T}) = \begin{cases} 1 & \text{for } \omega_s/4 < |\omega| < \omega_s/2 \\ 0 & \text{otherwise} \end{cases} \tag{18.12}$$

respectively. The operation of the filter bank is illustrated in Fig. 18.10 and 18.11. Downsampling by a factor $M = 2$ will produce shifted copies of the lowpassed and highpassed spectrums, as depicted in Fig. 18.10g and h and upsampling by a factor $L = 2$ will change the location of the sampling frequency, as shown in Fig. 18.11a and b (see Sec. 18.2.2); that is, one image of each of the lowpassed and highpassed spectrums will be introduced in the baseband, as in Fig. 18.11a and b by analogy with Fig. 18.3. At the output, the synthesis lowpass and highpass filters, which should in principle be identical to the corresponding analysis filters, will regenerate $x_0(nT)$ and $x_1(nT)$, respectively, as shown in Fig. 18.11e and f. Hence the output of the adder will be the required signal, as depicted in Fig. 18.11g.

In practical filters, the passband gain is not unity, the stopband loss is not infinite, and transitions between passbands and stopbands are gradual. Hence amplitude and phase distortion as well as aliasing will be introduced, as can be seen in Fig. 18.12. While the effects of these imperfections can be reduced by designing filters of better quality, these problems can largely be eliminated in a much simpler way by using lowpass and highpass filters whose frequency responses have certain symmetry properties. This possibility will now be examined.

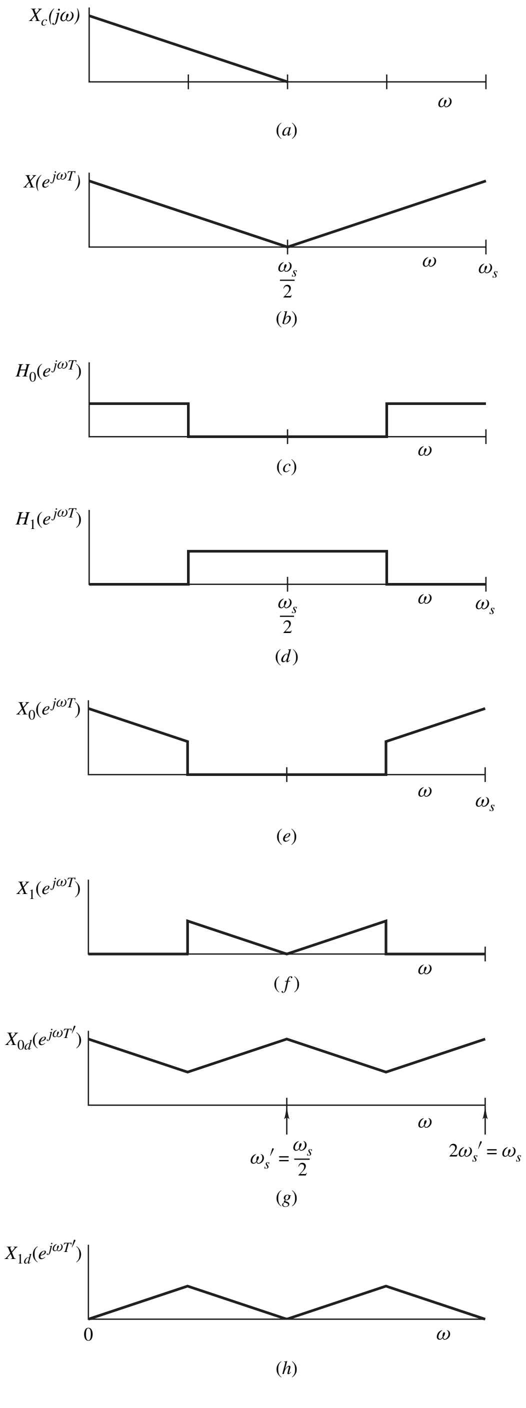


Figure 18.10 Operation of analysis section of 2-band QMF bank.

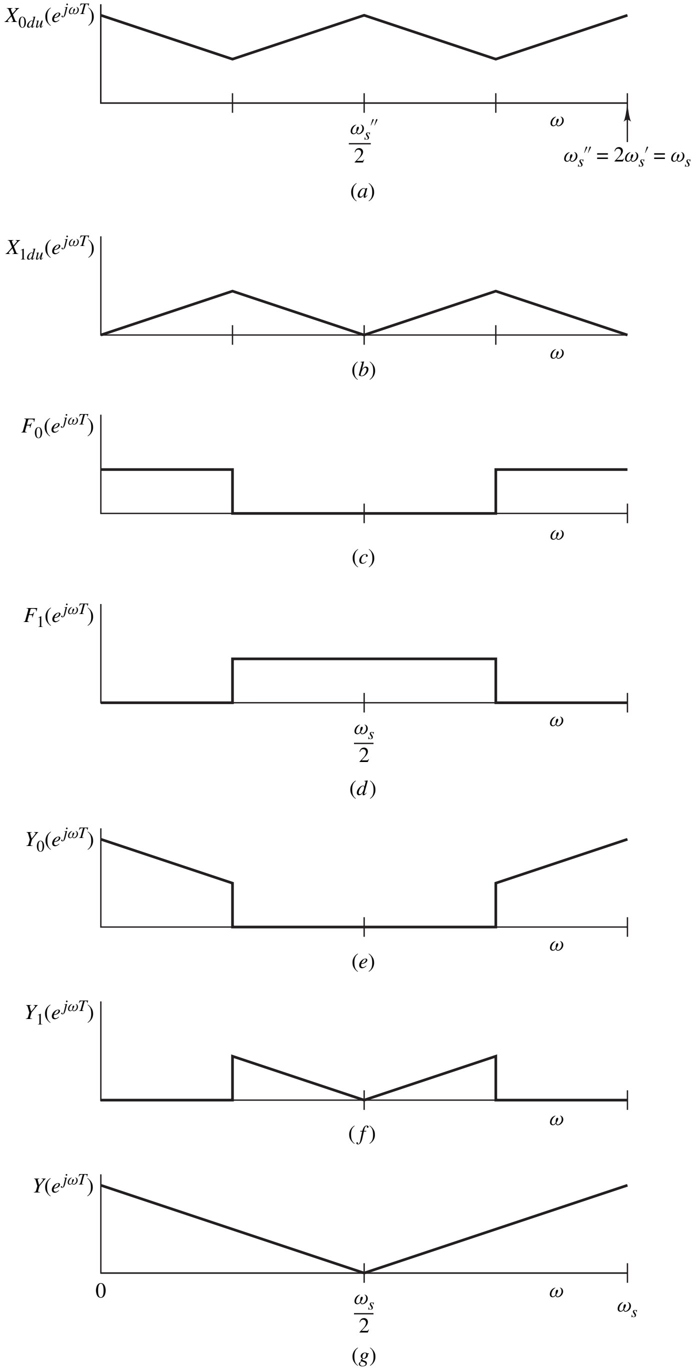


Figure 18.11 Operation of synthesis section of 2-band QMF bank.

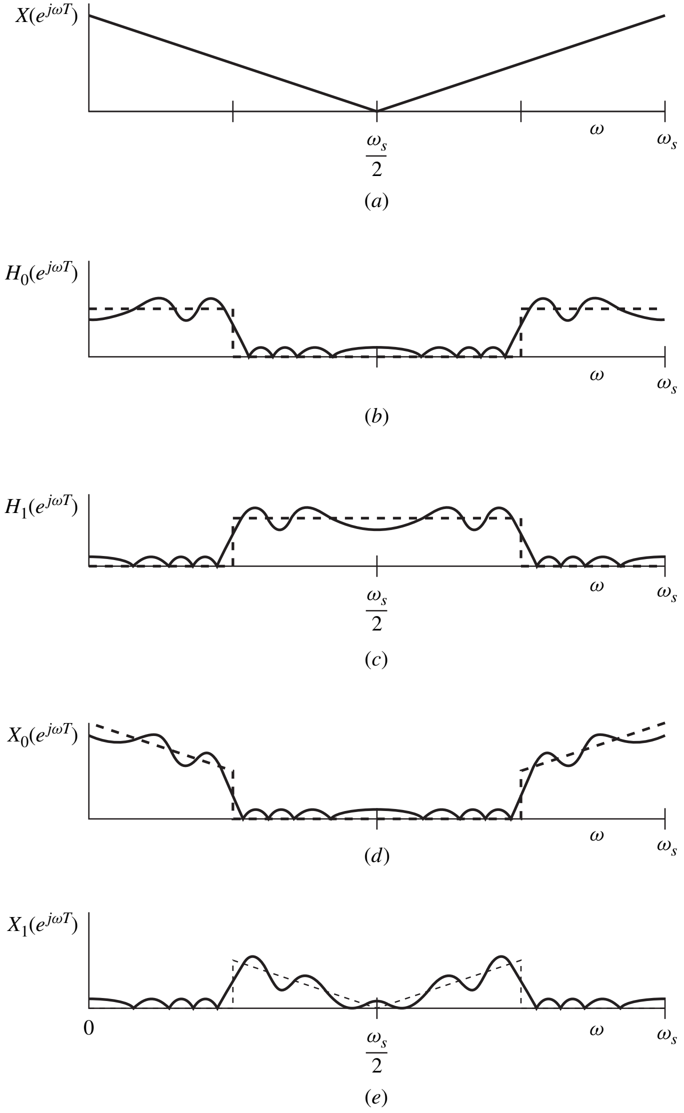


Figure 18.12 Effects of filter imperfections in 2-band QMF bank.

18.3.2 Elimination of Aliasing Errors

Let $T' = 2T$ and $T'' = T'/2 = T$. From Eq. (18.3), the frequency spectrum at the output of the top down-sampler in Fig. 18.9 can be expressed as

$$X_{0d}(e^{j\omega T'}) = \tfrac{1}{2}[X(e^{j\omega T'/2})H_0(e^{j\omega T'/2}) + X(e^{j(\omega T'/2-\pi)})H_0(e^{j(\omega T'/2-\pi)})] \tag{18.13}$$

On the other hand, the frequency spectrum at the output of the top upsampler is given by Eq. (18.5) as

$$X_{0du}(e^{j\omega T''}) = X_{0du}(e^{j\omega T'/2}) = X_{0d}(e^{j\omega T'}) \tag{18.14}$$

and from Fig. 18.9

$$Y_0(e^{j\omega T''}) = X_{0du}(e^{j\omega T''})F_0(e^{j\omega T''}) \tag{18.15}$$

Now Eqs. (18.13)–(18.15) give

$$\begin{aligned} Y_0(e^{j\omega T''}) = \tfrac{1}{2}[&X(e^{j\omega T'/2})H_0(e^{j\omega T'/2}) \\ &+X(e^{j(\omega T'/2-\pi)})H_0(e^{j(\omega T'/2-\pi)})]F_0(e^{j\omega T''}) \end{aligned}$$

and on eliminating T' and T'', we have

$$Y_0(e^{j\omega T}) = \tfrac{1}{2}[X(e^{j\omega T})H_0(e^{j\omega T}) + X(e^{j(\omega T-\pi)})H_0(e^{j(\omega T-\pi)})]F_0(e^{j\omega T})$$

Similarly,

$$Y_1(e^{j\omega T}) = \tfrac{1}{2}[X(e^{j\omega T})H_1(e^{j\omega T}) + X(e^{j(\omega T-\pi)})H_1(e^{j(\omega T-\pi)})]F_1(e^{j\omega T})$$

and, therefore,

$$\begin{aligned} Y(e^{j\omega T}) &= Y_0(e^{j\omega T}) + Y_1(e^{j\omega T}) \\ &= \tfrac{1}{2}[H_0(e^{j\omega T})F_0(e^{j\omega T}) + H_1(e^{j\omega T})F_1(e^{j\omega T})]X(e^{j\omega T}) \\ &\quad +\tfrac{1}{2}[H_0(e^{j(\omega T-\pi)})F_0(e^{j\omega T}) \\ &\quad +H_1(e^{j(\omega T-\pi)})F_1(e^{j\omega T})]X(e^{j(\omega T-\pi)}) \end{aligned} \tag{18.16}$$

The first term in the above relation represents the required signal, whereas the second term represents spurious components due to aliasing. If the filters are designed such that

$$H_0(e^{j(\omega T-\pi)})F_0(e^{j\omega T}) + H_1(e^{j(\omega T-\pi)})F_1(e^{j\omega T}) = 0 \tag{18.17}$$

then the aliasing produced by the lowpass filters will be canceled exactly by the aliasing produced by the highpass filters even if the quality of these filters is not particularly good. This relation can be satisfied by designing the analysis and synthesis filters such that

$$F_0(e^{j\omega T}) = 2H_1(e^{j(\omega T-\pi)})$$

and

$$F_1(e^{j\omega T}) = -2H_0(e^{j(\omega T-\pi)})$$

The factor 2 is required to compensate for the factor 1/2 introduced by the downsampling (see Fig. 18.3*c*).

With Eq. (18.17) satisfied, the output of a 2-band QMF bank is given by Eq. (18.16) as

$$Y(e^{j\omega T}) = T(e^{j\omega T})X(e^{j\omega T})$$

where

$$T(e^{j\omega T}) = H_0(e^{j\omega T})H_1(e^{j(\omega T-\pi)}) - H_1(e^{j\omega T})H_0(e^{j(\omega T-\pi)}) \tag{18.18}$$

is the overall frequency response of the QMF bank. If we assume that the lowpass and highpass filters have frequency responses that are mirror-image symmetric with respect to frequency $\omega = \omega_s/4$, then

$$H_1(e^{j\omega T}) = H_0(e^{j(\omega T-\pi)}) \tag{18.19a}$$

or

$$H_1(e^{j(\omega T-\pi)}) = H_0(e^{j\omega T}) \tag{18.19b}$$

Hence Eq. (18.18) can be expressed as

$$\begin{aligned} T(e^{j\omega T}) &= H_0^2(e^{j\omega T}) - H_0^2(e^{j(\omega T-\pi)}) \\ &= M(\omega)e^{j\theta(\omega)} \end{aligned} \tag{18.20}$$

where $M(\omega)$ and $\theta(\omega)$ are the overall amplitude and phase responses of the QMF bank. In practice, $M(\omega)$ should as far as possible be constant and $\theta(\omega)$ should be a linear function of ω to avoid amplitude and phase distortion in the reconstruction process (see Sec. 5.7).

A transfer function that characterizes the QMF bank can be formed as

$$T(z) = H_0(z)H_1(-z) - H_1(z)H_0(-z) \tag{18.21}$$

by letting $e^{j\omega T} = z$ in Eq. (18.18).

18.3.3 Design Considerations

QMF banks can be designed in terms of nonrecursive or recursive filters. Let us consider the first case.

On assuming that the impulse response of the analysis lowpass filter is symmetrical, its frequency response can be expressed as

$$H_0(e^{j\omega T}) = e^{j\omega(N-1)T/2}|H_0(e^{j\omega T})| \tag{18.22}$$

(see Table 9.1), where N is the filter length. From Eqs. (18.20) and (18.22), we can write

$$T(e^{j\omega T}) = [|H_0(e^{j\omega T})|^2 \pm |H_0(e^{j(\omega T-\pi)})|^2]e^{j\omega(N-1)T}$$

where the plus sign applies in the case where N is even and the minus sign applies in the case where N is odd. In effect, *linear phase response is achieved*, i.e., the design has zero phase distortion. If the required mirror-image symmetry is assumed, the overall frequency response becomes zero at $\omega = \omega_s/4$ if N is odd. Since $T(e^{j\omega T})$ is required to be an allpass function, we conclude that *N must be even*.

A nonrecursive filter can at this point be designed on the basis of the idealized frequency response in Eq. (18.11) such that the constraint

$$|H_0(e^{j\omega T})|^2 + |H_0(e^{j(\omega T-\pi)})|^2 = 1 \qquad \text{for } 0 \le \omega \le \omega_s/2$$

is satisfied. Designs of this type can be obtained by using the window technique [14].

Nonrecursive QMF banks can also be designed by using optimization methods, as described in [15, 16]. One of many possibilities is to construct an error function of the form

$$E(\mathbf{x}) = E_1(\mathbf{x}) + \alpha E_2(\mathbf{x})$$

where **x** is a column vector whose elements are the transfer function coefficients and

$$E_1(\mathbf{x}) = \int_{\omega_s/4}^{\omega_s/2} |H_0(e^{j\omega T})|^2 \, d\omega$$

$$E_2(\mathbf{x}) = \int_{\omega_s/4}^{\omega_s/2} [|H_0(e^{j\omega T})|^2 + |H_0(e^{j(\omega T-\pi)})|^2 - 1]^2 \, d\omega$$

Minimizing $E(\mathbf{x})$ with respect to the transfer-function coefficients using one of the quasi-Newton algorithms described in Chap. 16 will minimize the stopband error in the lowpass filter and force the overall amplitude response to approach unity in a least-squares sense. Parameter α can be used to emphasize or de-emphasize the error in the overall amplitude response relative to the stopband error of the lowpass filter.

In certain types of recursive filters, it is quite easy to design a pair of complementary filters whose amplitude responses assume the form

$$H_0(e^{j\omega T}) = \tfrac{1}{2}[A_0(e^{j\omega T}) + A_1(e^{j\omega T})] \tag{18.23}$$

and

$$H_1(e^{j\omega T}) = \tfrac{1}{2}[A_0(e^{j\omega T}) - A_1(e^{j\omega T})] \tag{18.24}$$

where $A_0(e^{j\omega T})$ and $A_1(e^{j\omega T})$ are allpass functions. With these methods, arbitrary amplitude response specifications can be obtained and the required symmetry about the frequency $\omega_s/4$ can be easily achieved. From Eqs. (18.18) and (18.19), the overall frequency response can be put in the form

$$T(e^{j\omega T}) = H_0^2(e^{j\omega T}) - H_1^2(e^{j\omega T})$$

and from Eqs. (18.23) and (18.24)

$$T(e^{j\omega T}) = A_0(e^{j\omega T})A_1(e^{j\omega T})$$

In effect, *the overall frequency response of the QMF bank is an allpass function*; that is, the design obtained has zero amplitude distortion. However, a certain amount of phase distortion will be present, which may or may not be objectionable depending on the application. Such filters can be designed as wave lattice filters, as described by Gazsi (see Sec. 17.5 and Ref. [10] of Chap. 17).

Example 18.2 Design a 2-band QMF bank as a wave lattice filter using the method described in Sec. 17.5 along with an elliptic approximation.[2] The specifications of the lowpass filter are as follows:

$$A_p = 1.0 \text{ dB} \qquad A_a = 48.3 \text{ dB} \qquad \tilde{\Omega}_p = 4{,}500 \text{ rad/s}$$

$$\tilde{\Omega}_a = 5{,}500 \text{ rad/s} \qquad \omega_s = 2.0 \times 10^4 \text{ rad/s}$$

■ **Solution**

Using the formulas in Tables 12.2 and 12.6, the required selectivity factor is found to be $k = 0.729454$. The minimum filter order that will satisfy the required specifications is $n = 5$, according to Eqs. (10.86)–(10.90). The value of λ in the lowpass-to-lowpass transformation (see Table 12.1) that will compensate for the warping effect introduced by the bilinear transformation can be determined as $\lambda = 1.570796 \times 10^{-4}$. On using the formulas in Eqs. (10.91)–(10.99) and then applying the lowpass-to-lowpass transformation, the denormalized transfer function is obtained as

$$H_A(s) = \frac{1}{s + b_{01}} \prod_{j=2}^{3} \frac{s^2 + a_{0j}}{s^2 + b_{1j}s + b_{0j}}$$

where coefficients a_{ij} and b_{ij} are given in Table 18.1.

Now from Eqs. (17.25a) and (17.25b), polynomials $d_A(s)$ and $d_B(s)$ can be deduced as

$$d_A(s) = (s + b_{01})(s^2 + b_{13}s + b_{03}) \tag{18.25}$$

and

$$d_B(s) = s^2 + b_{12}s + b_{02} \tag{18.26}$$

Table 18.1 Coefficients of $H_A(s)$ (Example 18.2)

j	a_{0j}	b_{0j}	b_{1j}
1	–	$1.918343E + 3$	–
2	$1.279580E + 8$	$1.605501E + 7$	$2.459931E + 3$
3	$5.951444E + 7$	$2.945786E + 7$	$6.607256E + 2$
$H_0 = 1.191374E + 2$			

[2] *See Sec. VI of Ref. [17] for a related design example.*

Digital realizations for networks N_A and N_B shown in Fig. 17.14*b* can be obtained as depicted in Fig. 18.13 by using the approach described in Sec. 17.5. The multiplier constants of the adaptors can be calculated from Eqs. (17.29) and (17.30) as follows: $\mu_{11} = 0.536886$, $\mu_{12} = 0.432520$, $\mu_{22} = -0.566457$, $\mu_{13} = 0.158183$, and $\mu_{23} = -0.886611$.

The configuration obtained will operate as a halfband lowpass filter with respect to output B_2 and as a halfband highpass filter with respect to output B_1 by virtue of the Feldkeller equation (see Sec. 17.5.2).

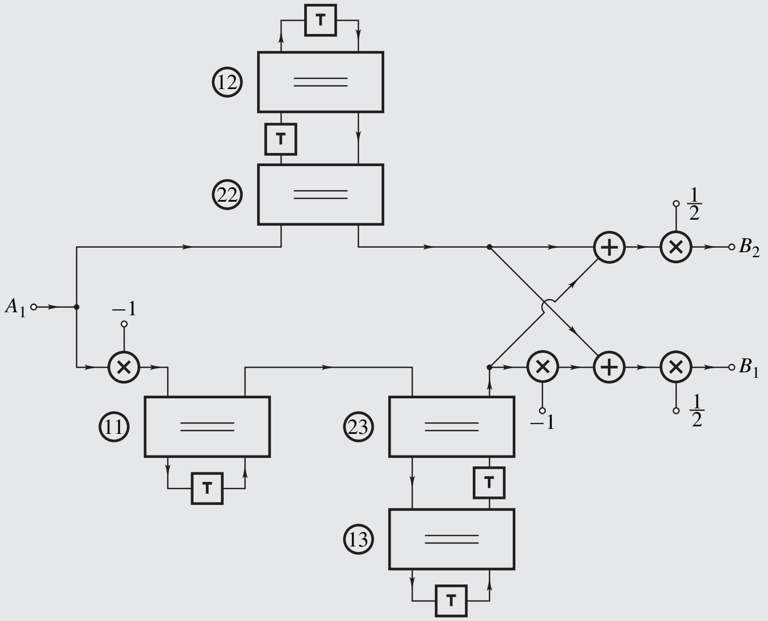


Figure 18.13 Lattice wave digital filter realizing 2-band QMF bank (Example 18.2).

18.3.4 Perfect Reconstruction

In the preceding section, it has been shown that designs with zero phase distortion can be easily obtained by using linear-phase nonrecursive filters whereas zero amplitude distortion can be achieved by using recursive wave digital filters. In the case of nonrecursive filters, amplitude distortion can be rendered insignificant by using a high-order filter to achieve small passband and stopband ripples. On the other hand, in the case of recursive filters, phase distortion can be rendered insignificant through the use of phase equalization (see Sec. 16.8). In this section, a scheme proposed by Smith and Barnwell [18] is described by which zero phase distortion as well as zero amplitude distortion can be achieved simultaneously independently of the quality of the filters used.

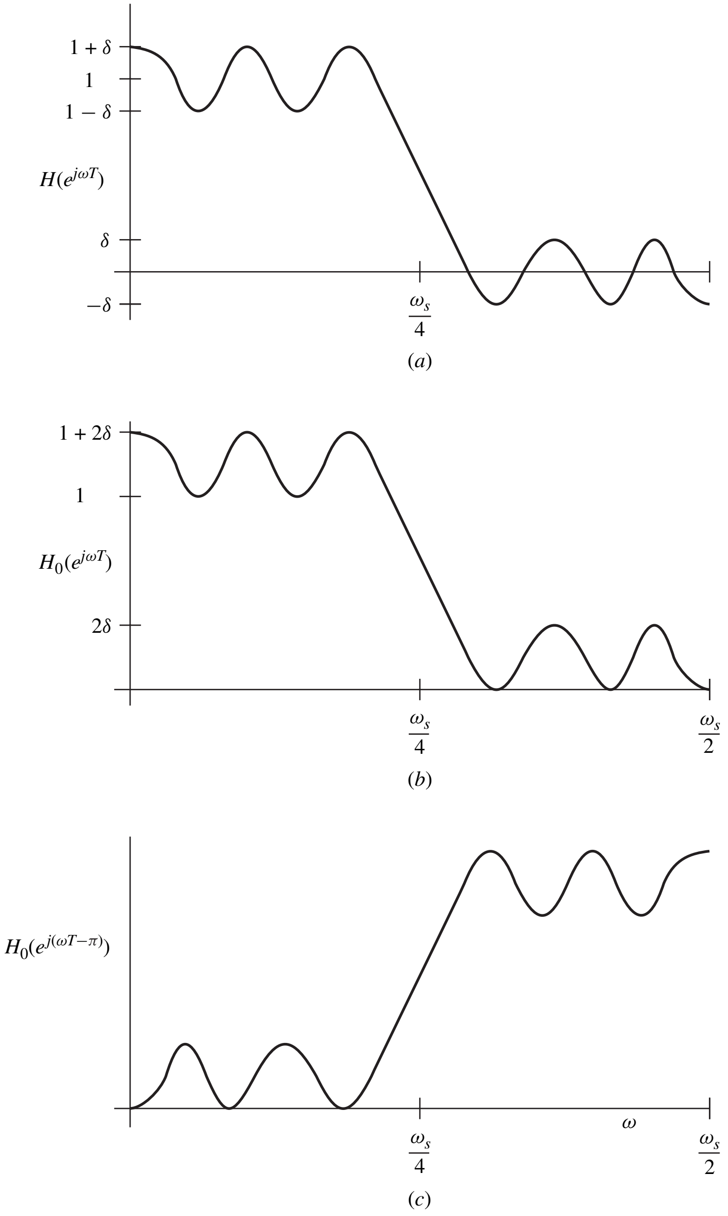


Figure 18.14 Required amplitude-response symmetry for perfect reconstruction.

Let $H(z)$ be a linear-phase, lowpass, nonrecursive transfer function and assume that the passband and stopband errors satisfy the symmetry property

$$||H(e^{j\omega T})| - 1| = |H(e^{j(\omega_s/2-\omega)T})| \leq \delta \tag{18.27}$$

as illustrated in Fig. 18.14*a*. Using $H(z)$, a modified transfer function

$$H_0(z) = H(z) + \delta z^{-(N-1)/2}$$

can be formed, where N is the length of the filter. From the above symmetry property, we can write

$$|H_0(e^{j\omega T})|^2 + |H_0(e^{j(\omega T-\pi)})|^2 = G = 1 + 2\delta \tag{18.28}$$

where G is a constant, i.e., changes in the two functions cancel each other out exactly, as demonstrated in Fig. 18.14*a* to *c*.

A linear-phase transfer function with the amplitude response given by Eq. (18.28) can now be constructed as

$$T(z) = -H_0(z)H_0(z^{-1})z^{-(N-1)} - [-H_0(-z^{-1})(-z)^{-(N-1)}]H_0(-z) \tag{18.29}$$

(see Prob. 18.6), where N is even. On comparing Eqs. (18.21) and (18.29), the assignment

$$H_1(z) = z^{-(N-1)}H_0(-z^{-1})$$

can be made and if we let

$$F_0(z) = 2H_1(-z) \qquad \text{and} \qquad F_1(z) = -2H_0(-z)$$

a realization of the QMF bank is obtained in which both the amplitude and phase distortions are zero at the same time. Realizations of this type are said to have the *perfect reconstruction property*.

Halfband filters with the symmetry property of Eq. (18.27) can be designed by using the methods described in Chap. 15.

The above principles can be extended to the design of QMF banks with multiple bands.

18.4 HILBERT TRANSFORMERS

In certain digital signal processing applications, it is necessary to form a special version of a given signal $x(nT)$, designated as $\tilde{x}(nT)$, with the special property that its frequency spectrum is equal to that of $x(nT)$ for the positive Nyquist interval and zero for the negative Nyquist interval, i.e.,[3]

$$\tilde{X}(e^{j\omega T}) = \begin{cases} X(e^{j\omega T}) & \text{for } 0 < \omega < \omega_s/2 \\ 0 & \text{for } -\omega_s/2 \leq \omega < 0 \end{cases} \tag{18.30}$$

[3] *Alternatively, the spectrum of $\tilde{x}(nT)$ may be taken to be equal to that of $x(nT)$ for the negative Nyquist interval and zero for the positive Nyquist interval.*

Signals with this property have been referred to as *analytic signals* in the past [19, 20] and are useful in a number of applications, for example, in single-sideband modulation in frequency-division multiplex systems. The motivation for eliminating the spectrum of a signal over the negative Nyquist interval is that it is the mirror image of the spectrum over the positive Nyquist interval, i.e., it contains the same information, and its elimination reduces the required bandwidth for the processing and/or transmission of the signal by half.

An essential property of real continuous-time signals is that their amplitude spectrums are even, and their phase spectrums are odd functions of ω (see Prob. 2.16). Since analytic signals violate these requirements, they must necessarily be complex of the form

$$\tilde{x}(nT) = x_r(nT) + jx_i(nT) \tag{18.31}$$

where $x_r(nT)$ and $x_i(nT)$ are real sequences. If the spectrum of an analytic signal $\tilde{x}(nT)$ is known, the spectrums of $x_r(nT)$ and $x_i(nT)$ can be readily deduced as

$$X_r(e^{j\omega T}) = \tfrac{1}{2}[\tilde{X}(e^{j\omega T}) + \tilde{X}^*(e^{-j\omega T})] \tag{18.32}$$

and

$$jX_i(e^{j\omega T}) = \tfrac{1}{2}[\tilde{X}(e^{j\omega T}) - \tilde{X}^*(e^{-j\omega T})] \tag{18.33}$$

where $\tilde{X}^*$ is the complex conjugate of $\tilde{X}$. These relations are illustrated in Fig. 18.15.

From Eqs. (18.32) and (18.33), we obtain

$$\tilde{X}(e^{j\omega T}) = 2X_r(e^{j\omega T}) - \tilde{X}^*(e^{-j\omega T}) \tag{18.34}$$

and

$$\tilde{X}(e^{j\omega T}) = 2jX_i(e^{j\omega T}) + \tilde{X}^*(e^{-j\omega T}) \tag{18.35}$$

and since $\tilde{X}^*(e^{-j\omega T}) = 0$ for $0 < \omega < \omega_s/2$ (see Fig. 18.15*b*), Eqs. (18.30), (18.34), and (18.35) give

$$\tilde{X}(e^{j\omega T}) = \begin{cases} 2X_r(e^{j\omega T}) & \text{for } 0 < \omega < \omega_s/2 \\ 0 & \text{for } -\omega_s/2 \le \omega < 0 \end{cases}$$

and

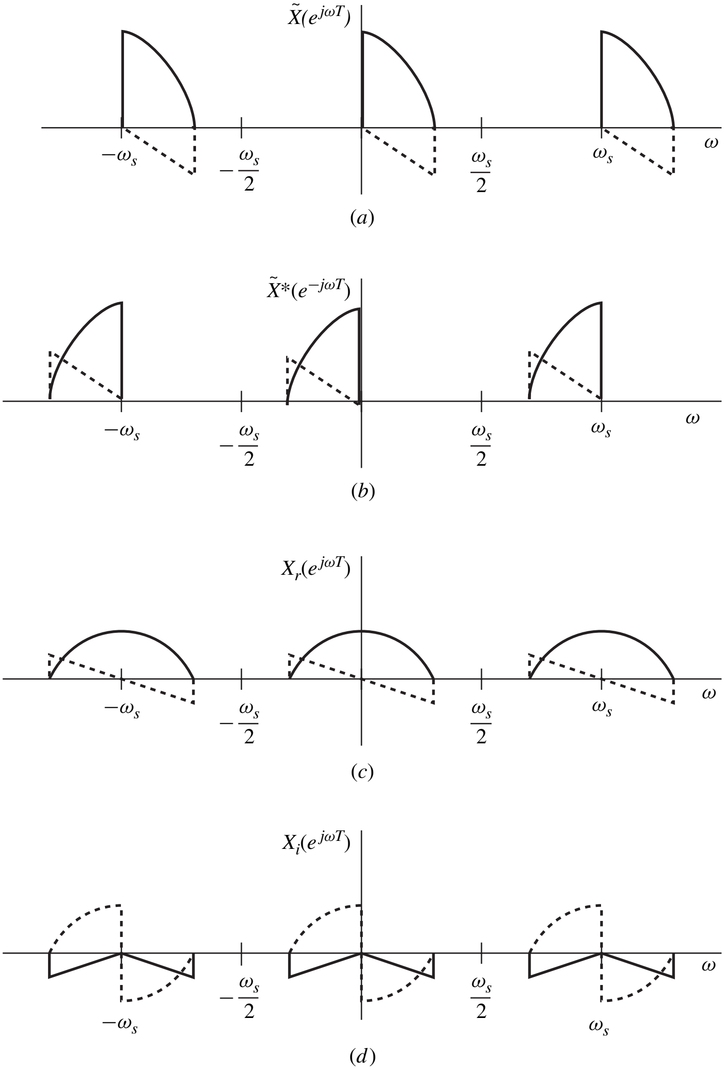
$$\tilde{X}(e^{j\omega T}) = \begin{cases} 2jX_i(e^{j\omega T}) & \text{for } 0 < \omega < \omega_s/2 \\ 0 & \text{for } -\omega_s/2 \le \omega < 0 \end{cases}$$

Thus

$$X_i(e^{j\omega T}) = -jX_r(e^{j\omega T}) \qquad \text{for } 0 < \omega < \omega_s/2 \tag{18.36}$$

On the other hand, from Eq. (18.31)

$$\tilde{X}(e^{j\omega T}) = X_r(e^{j\omega T}) + jX_i(e^{j\omega T})$$



(*a*)

(*b*)

(*c*)

(*d*)

Figure 18.15 Derivation of $x_r(nT)$ and $x_i(nT)$ from $\tilde{x}(nT)$ (solid curves represent real parts and dashed curves represent imaginary parts).

and since $\tilde{X}(e^{j\omega T}) = 0$ for $-\omega_s/2 \le \omega < 0$, we have

$$X_i(e^{j\omega T}) = jX_r(e^{j\omega T}) \qquad \text{for } -\omega_s/2 \le \omega < 0 \tag{18.37}$$

Therefore, Eqs. (18.36) and (18.37) can be expressed as

$$X_i(e^{j\omega T}) = \begin{cases} -jX_r(e^{j\omega T}) & \text{for } 0 < \omega < \omega_s/2 \\ jX_r(e^{j\omega T}) & \text{for } -\omega_s/2 \le \omega < 0 \end{cases}$$

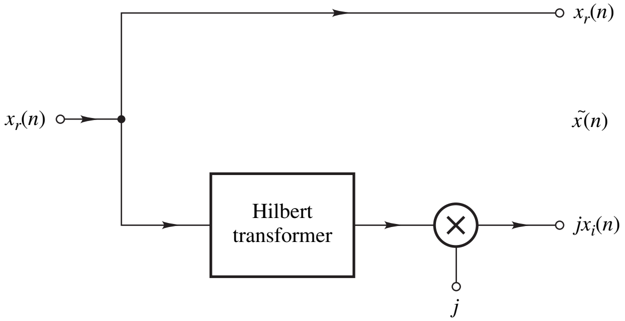


Figure 18.16 Synthesis of an analytic signal using a Hilbert transformer.

or

$$X_i(e^{j\omega T}) = H(e^{j\omega T})X_r(e^{j\omega T}) \tag{18.38}$$

where

$$H(e^{j\omega T}) = \begin{cases} -j & \text{for } 0 < \omega < \omega_s/2 \\ j & \text{for } -\omega_s/2 < \omega < 0 \end{cases} \tag{18.39}$$

These results show that the real and imaginary parts of an analytic signal are interrelated and the imaginary part can be obtained from the real part by using Eq. (18.38). This relation may be deemed to represent a filter with input $x_r(nT)$, output $x_i(nT)$, and frequency response $H(e^{j\omega T})$. A filter of this type is commonly referred to as a *Hilbert transformer*. Its output $x_i(nT)$ is said to be the *Hilbert transform* of $x_r(nT)$.

On the basis of these principles, given a real sequence $x_r(nT)$, a corresponding analytic signal $\tilde{x}(nT)$ can be synthesized by using the configuration depicted in Fig. 18.16. The operation of this scheme is illustrated in Fig. 18.17. As can be seen, if $x_r(nT)$ is a real sequence and $x_i(nT)$ is generated by using a Hilbert transformer, then $x_r(nT) + jx_i(nT)$ is an analytic signal, as shown in Fig. 18.17*d*.

18.4.1 Design of Hilbert Transformers

Hilbert transformers can be designed either in terms of nonrecursive or recursive filters. In the former case, either the Fourier series method of Chap. 9 or the weighted-Chebyshev method of Chap. 15 can be used.

Using the Fourier series method of Sec. 9.3, the impulse response of a Hilbert transformer can be obtained as

$$\begin{aligned} h(nT) &= \frac{T}{2\pi}\left(\int_{-\omega_s/2}^{0} je^{j\omega nT}d\omega - \int_{0}^{-\omega_s/2} je^{j\omega nT}\,d\omega\right) \\ &= \begin{cases} \dfrac{2}{n\pi}\sin^2\dfrac{n\pi}{2} & \text{for } n \neq 0 \\ 0 & \text{for } n = 0 \end{cases} \end{aligned} \tag{18.40}$$

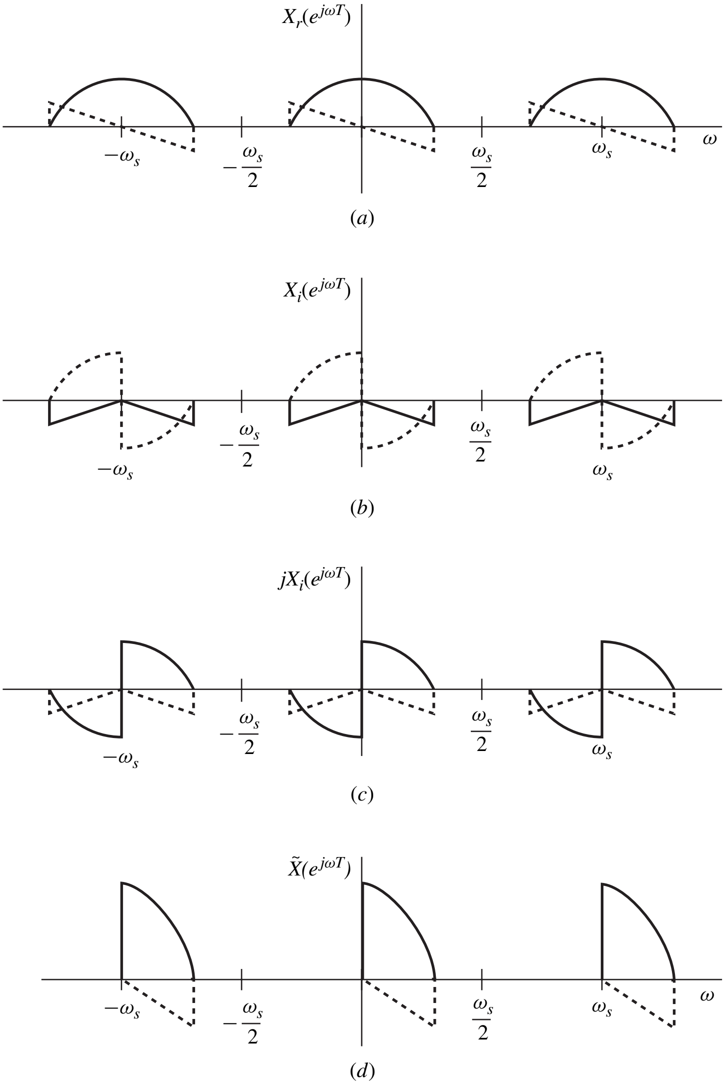


Figure 18.17 Operation of configuration in Fig. 18.16 (solid curves represent real parts and dashed curves represent imaginary parts).

Evidently, a Hilbert transformer, like a digital differentiator, has an antisymmetrical impulse response and can be designed either with an odd or even filter length N using the window technique described in Sec. 9.4. From Table 9.1, we note that for odd N, the amplitude response is zero for $\omega = 0$ and $\omega = \omega_s/2$. Hence the useful bandwidth that can be achieved is restricted to some range $0 < \omega_L \leq \omega \leq \omega_H < \omega_s/2$ where ω_L and ω_H can be made to approach 0 and $\omega_s/2$, respectively, as closely as desired by increasing N. On the other hand, for even N, the amplitude response need not be zero at $\omega = \omega_s/2$, as can be seen in Table 9.1, and, in this case, Hilbert transformers can be designed in which $\omega_H = \omega_s/2$.

The ideal response of a causal Hilbert transformer can be expressed in the form $e^{-j\omega c}jD(\omega)$ where

$$D(\omega) = \begin{cases} -1 & \text{for } 0 < \omega < \omega_s/2 \\ 1 & \text{for } -\omega_s/2 < \omega < 0 \end{cases}$$

and $c = (N-1)/2$. From Table 15.7, we conclude that a Hilbert transformer can be designed using the weighted-Chebyshev algorithm. The problem formulation is similar to that used for the design of digital differentiators in Sec. 15.9 and is left as an exercise to the reader (see Prob. 18.19).

Example 18.3 (*a*) Design a Hilbert transformer of length $N = 21$ using the Kaiser window with $\alpha = 3.0$, assuming a sampling frequency of 10 rad/s. (*b*) Repeat part (*a*) with $N = 22$ and compare the results obtained in the two cases.

■ **Solution**

(*a*) Using Eq. (18.40), the impulse response of the Hilbert transformer, $h(n)$, can be computed as shown in column 2 of Table 18.2*a*. The modified impulse response assumes the form

$$h_w(n) = w_K(n)h(n) \tag{18.41}$$

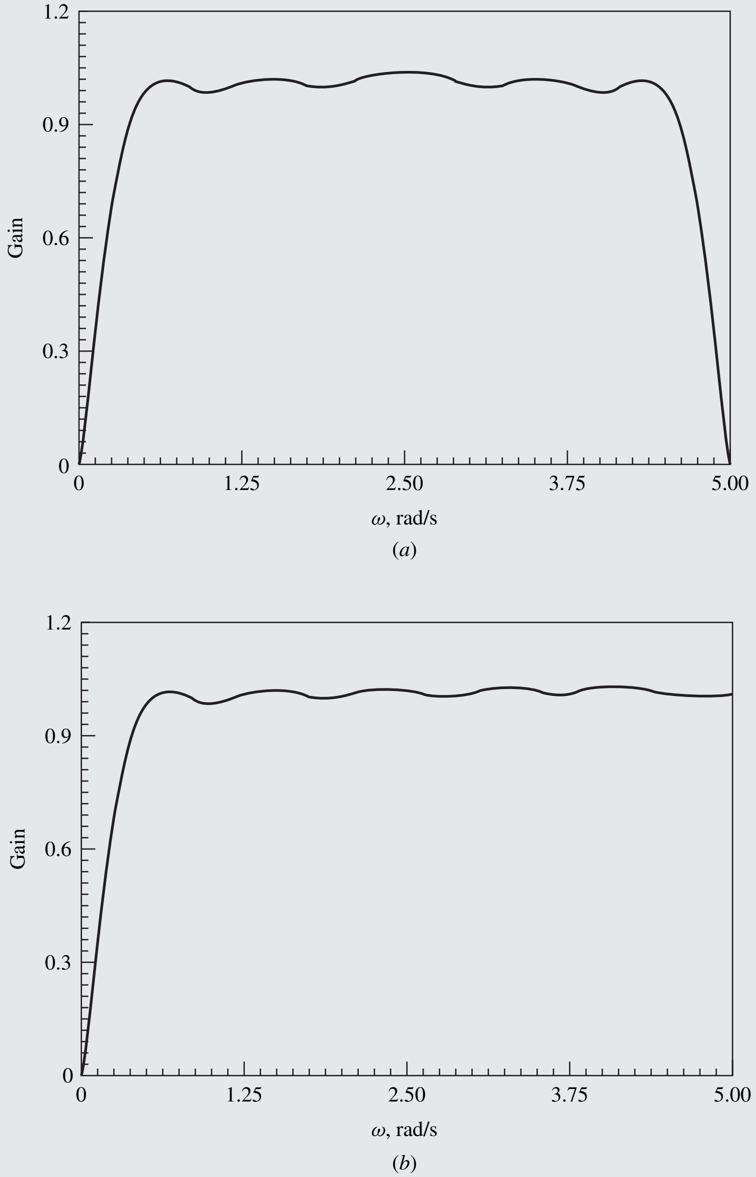
where $w_K(n)$ is the Kaiser window. With $\alpha = 3.0$, Eqs. (9.25) and (18.41) give the impulse response in column 3 of Table 18.2*a*. The amplitude response of the Hilbert transformer is depicted in Fig. 18.18*a*; the maximum passband error is 1.44 percent of the passband gain.

(*b*) For even N, the impulse response must be antisymmetrical about the midpoint between samples $(N-2)/2$ and $N/2$, that is, samples 10 and 11 for $N = 22$ (see Sec. 9.2.1).

Table 18.2*a* Impulse response of Hilbert transformer (Example 18.3, part (*a*))

n	$h(n) = -h(-n)$	$h_w(n)$
0	0.0	0.0
1	$6.366198E-1$	$6.289178E-1$
2	0.0	0.0
3	$2.122066E-1$	$1.898748E-1$
4	0.0	0.0
5	$1.273240E-1$	$9.255503E-2$
6	0.0	0.0
7	$9.094568E-2$	$4.698992E-2$
8	0.0	0.0
9	$7.073553E-2$	$2.138270E-2$
10	0.0	0.0

Consequently, in a noncausal Hilbert transformer, the impulse response is defined at $n' = \pm(n - 0.5)$ with $n = 1, 2, \ldots, N/2$. Such a design can be obtained by first converting the impulse response in Eq. (18.40) into a continuous-time function by letting $nT = t$ in $h(nT)$ and then resampling $h(t)$ at $t = (n - 0.5)T$ for $-NT/2 \leq t \leq NT/2$ where



(*a*)

(*b*)

Figure 18.18 Amplitude response of Hilbert transformer (Example 18.3): (*a*) $N = 21$, (*b*) $N = 22$.

Table 18.2*b* Impulse response of Hilbert transformer (Example 18.3, part (*b*))

n	$h(n) = -h(-n)$	$h_w(n)$
0.5	$6.366198E-1$	$6.302499E-1$
1.5	$2.122066E-1$	$2.038026E-1$
2.5	$1.273240E-1$	$1.161765E-1$
3.5	$9.094568E-2$	$7.713458E-2$
4.5	$7.073553E-2$	$5.447434E-2$
5.5	$5.787452E-2$	$3.944281E-2$
6.5	$4.897075E-2$	$2.869059E-2$
7.5	$4.244132E-2$	$2.066188E-2$
8.5	$3.744822E-2$	$1.453254E-2$
9.5	$3.350630E-2$	$9.818996E-3$
10.5	$3.031523E-2$	$6.211128E-3$

N is the required even integer. On applying this simple transformation to Eq. (18.40), we obtain the required impulse response as

$$h(n-0.5) = -h(-n+0.5) = \frac{2}{(n-0.5)\pi}\sin^2\left[\frac{(n-0.5)\pi}{2}\right] \qquad \text{for } n = 1, 2, \ldots, N/2 \qquad (18.42)$$

Similarly, by sampling the continuous-time Kaiser window at $t = (n-0.5)T$ for $-NT/2 \le t \le NT/2$ with $N = 22$ using Eqs. (7.24) and (7.25), the values of $w_K(n-0.5) = w_K(-n+0.5)$ can be computed, and from Eqs. (18.41) and (18.42) the design in Table 18.2*b* can be readily obtained. The amplitude response for this case is depicted in Fig. 18.18*b*; the maximum passband error in this case is 0.36 percent of the passband gain. We note that for odd N, several values of the impulse response are zero, which renders the design more economical. However, the design for even N provides a wider bandwidth and the approximation error is much smaller.

A feature of the latter design, which may be a problem in certain applications, is that the impulse response must be shifted by a noninteger multiple of the sampling period in order to achieve a causal design.

The frequency response of a Hilbert transformer can also be expressed as

$$H(e^{j\omega T}) = M(\omega)e^{j\theta(\omega)}$$

where

$$M(\omega) = 1 \qquad \text{and} \qquad \theta(\omega) = \begin{cases} -\pi/2 & \text{for } 0 < \omega < \omega_s/2 \\ \pi/2 & \text{for } -\omega_s/2 < \omega < 0 \end{cases}$$

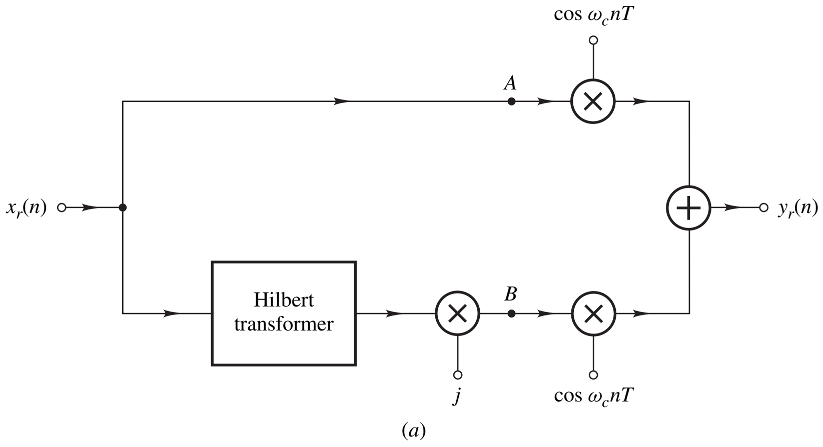
according to Eq. (18.39). Hence Hilbert transformers can also be designed as recursive filters by assuming a set of cascaded allpass sections, as in Sec. 16.8, and then forcing the overall phase response to approach -90° for the range $0 < \omega < \omega_s/2$ and 90° for the range $-\omega_s/2 < \omega < 0$ to within a prescribed tolerance. This can be done by using the optimization methods described in Chap. 16.

18.4.2 Single-Sideband Modulation

One of the important applications of Hilbert transformers is concerned with *single-sideband modulation*. This process involves two steps: first, an analytic version of a real signal is generated; second, it is used to modulate a sinusoidal carrier of frequency ω_c. Single-sideband modulation can be carried out by using the scheme depicted in Fig. 18.19, as will now be demonstrated.

The signals at nodes A and B in Fig. 18.19*a* constitute an analytic signal given by

$$\begin{aligned}\tilde{x}(nT) &= x_r(nT) + jx_i(nT)\\ &= A(nT)e^{j\phi(nT)} \qquad (18.43)\end{aligned}$$



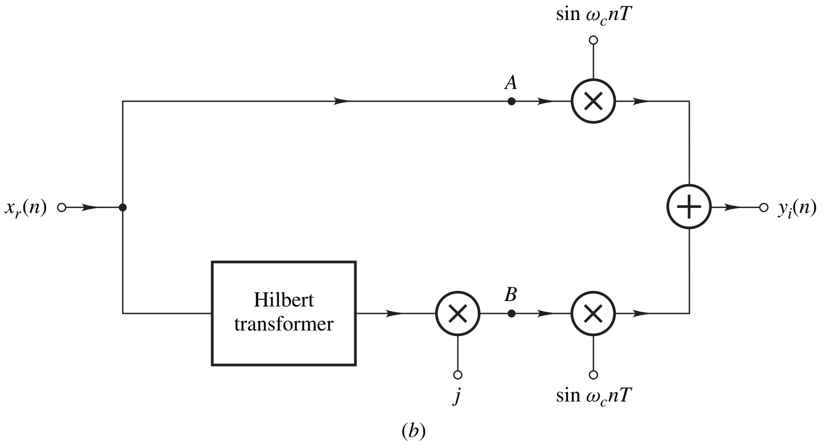


Figure 18.19 Single-sideband modulation: (*a*) Generation of real sequence, (*b*) generation of imaginary sequence.

where

$$A(nT) = \left[x_r^2(nT) + x_i^2(nT)\right]^{1/2} \qquad \text{and} \qquad \phi(nT) = \tan^{-1}\frac{x_i(nT)}{x_r(nT)}$$

Hence the outputs of the top and bottom structures are given by

$$y_r(n) = A(nT)\cos[\omega_c nT + \phi(nT)]$$

and

$$y_i(n) = A(nT)\sin[\omega_c nT + \phi(nT)]$$

respectively. Therefore,

$$\begin{aligned}\tilde{y}(nT) &= y_r(n) + jy_i(n)\\ &= A(nT)e^{j[\omega_c nT + \phi(nT)]}\end{aligned}$$

and, if the spectrum of $\tilde{x}(nT)$ (see Eq. (18.30)) is assumed to be zero for $\omega_m \le \omega \le \omega_s/2$ and $\omega_c + \omega_m < \omega_s/2$, then the spectrum of $\tilde{y}(nT)$ is given by

$$\tilde{Y}(e^{j\omega T}) = \tilde{X}(e^{j(\omega-\omega_c)T}) \qquad \text{for } 0 < \omega < \omega_s/2 \tag{18.44}$$

(see Prob. 18.20). In effect, $\tilde{y}(nT)$ represents a carrier modulated by the upper sideband of signal $x_r(nT)$. The operation of the modulator is illustrated in Fig. 18.20. A simplified realization of the modulator can be easily obtained from Fig. 18.19, as depicted in Fig. 18.21.

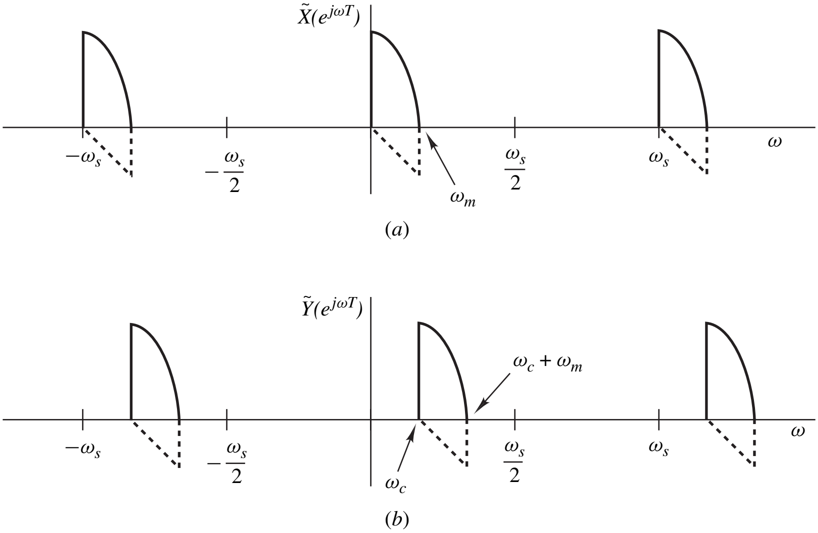


Figure 18.20 Operation of single-sideband modulator (solid curves represent real parts and dashed curves represent imaginary parts).

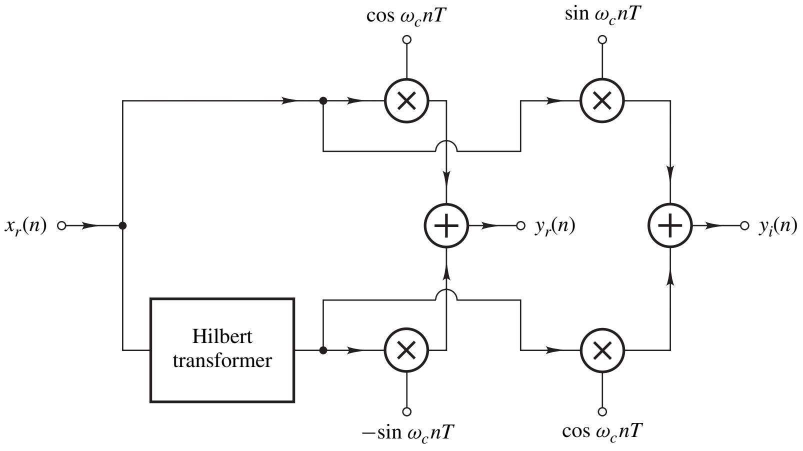


Figure 18.21 Simplified single-sideband modulator.

18.4.3 Sampling of Bandpassed Signals

In certain applications, it is necessary to sample a signal $x(t)$ whose spectrum occupies a frequency interval $\omega_L \leq \omega \leq \omega_H$. Signals of this type are often generated through the use of bandpass filters and can be referred to as *bandpassed*[4] signals. The processing of such signals would necessitate a minimum sampling frequency $\omega_s = 2\omega_H$, according to the sampling theorem. However, through the use of a Hilbert transformer and a pair of decimators, it is possible to generate a downsampled version of the signal that can be processed at a much lower sampling frequency. Such a scheme is shown in Fig. 18.22a, and its mode of operation is illustrated by the frequency spectrums of Fig. 18.23.

The signal components at nodes A and B in Fig. 18.22a represent an analytic signal $\tilde{x}(nT)$ with the frequency spectrum depicted in Fig. 18.23b, as demonstrated earlier. Now if $M = \text{Int}(\omega_s/B)$, where $B = (\omega_H - \omega_L)$, the downsamplers will produce exactly M copies of the spectrum of $\tilde{x}(nT)$ in the interval $-\omega_s/2 \leq \omega \leq \omega_s/2$, as illustrated in Fig. 18.23c for the case $M = 8$. As can be seen, the information content of the signal now occupies the interval $-B/2 \leq \omega \leq B/2$ and the signal can be processed using a sampling frequency B. For a narrowband signal, we have $B \ll 2\omega_H$ and, therefore, the necessary speed of operation of the hardware is significantly reduced.

The processed bandpassed signal can be recovered by reversing the above procedure using the configuration of Fig. 18.22b, where the bandpass filter is a complex bandpass filter with a frequency response

$$H(e^{j\omega T}) = \begin{cases} M & \text{for } \omega_L \leq \omega \leq \omega_H \\ 0 & \text{otherwise} \end{cases}$$

[4] *Some authors refer to such signals as* bandpass *signals but "bandpass" is normally used to qualify filters or systems.*

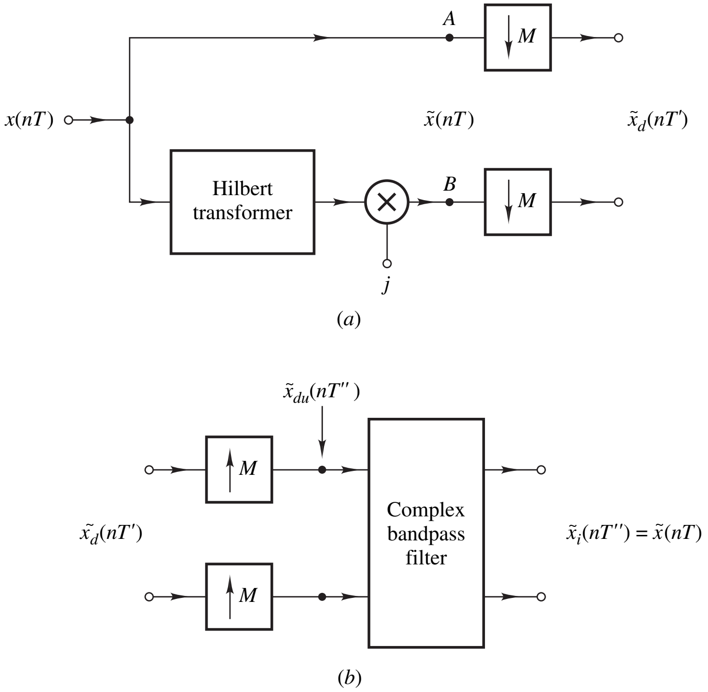


Figure 18.22 Processing of bandpassed signals: (*a*) Sampling, (*b*) reconstruction.

The operation of this scheme is illustrated by the frequency spectrums in Fig. 18.24. Upsampling will produce $M - 1$ images of the spectrum of $\tilde{x}(nT)$, as shown in Fig. 18.24*b*, and the desired image can be selected by the bandpass filter, as depicted in Fig. 18.24*d*.

18.5 ADAPTIVE DIGITAL FILTERS

In many applications, time-variable filters whose characteristics can be varied with time are required. Filters of this type can be obtained by using multipliers with time-variable coefficients. A time-variable filter that incorporates some adaptation mechanism by which the multiplier coefficients can be adjusted on line so as to optimize some performance criterion is said to be an *adaptive filter* [5–8]. The adaptation mechanism usually incorporates an optimization algorithm that evaluates the instantaneous values of the multiplier coefficients such that some norm of an error function of the form

$$e(n) = d(n) - y(n) \tag{18.45}$$

is minimized, where $d(n)$ is some desired reference signal and $y(n)$ is the filter output. A typical adaptive-filter configuration is illustrated in Fig. 18.25.

The design of adaptive filters involves the choice of filter structure, the specific error norm to be used as objective function, and the type of adaptation algorithm [21–27]. The structure can be nonrecursive or recursive; the objective function may involve the expected amplitude or square of the

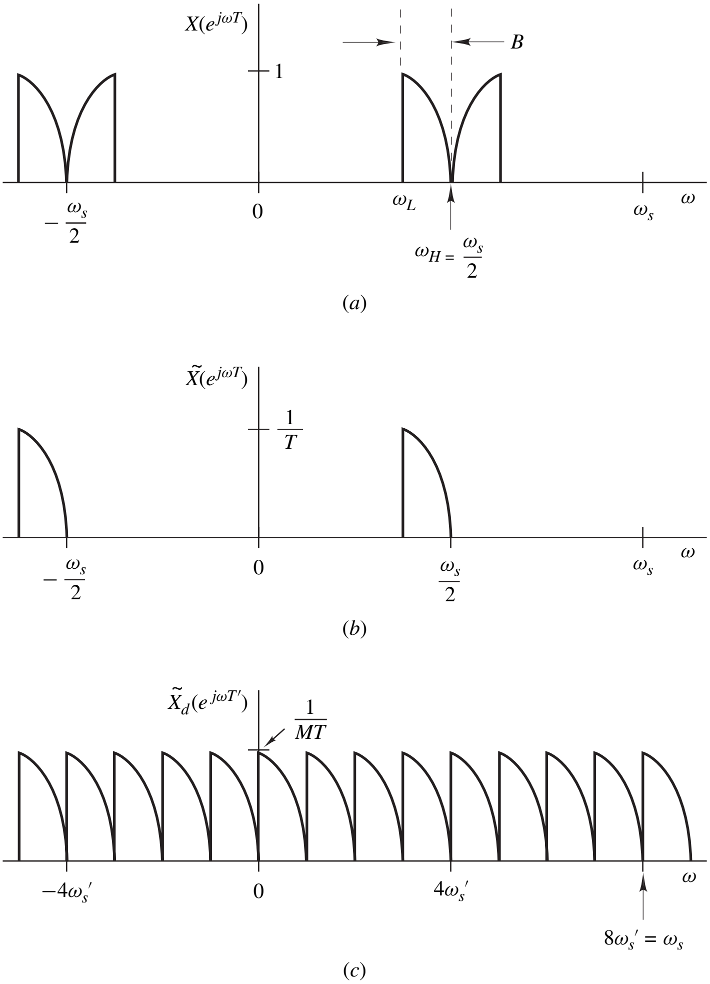


Figure 18.23 Operation of scheme in Fig. 18.22*a*.

error and possibly the expected value of some higher power or the maximum of the error; similarly, the adaptation algorithm can be one of several possibilities. Like other filters, adaptive filters are required to be economical, fast, and insensitive to finite word-length effects; in addition, they should adapt in a short period of time, and the residual error after adaptation should be as small as possible. Hence, the performance criteria for these filters are the simplicity and properties of the structure, and the flexibility, reliability, computational complexity, and convergence properties of the adaptation algorithm employed.

Usually, the most well behaved of the possible objective functions involves the square of the error (e.g., the mean-square error), which can be minimized very quickly using some relatively simple optimization algorithms. As a consequence, this objective function is preferred, although a mean-square solution may not be the most appropriate in certain applications.

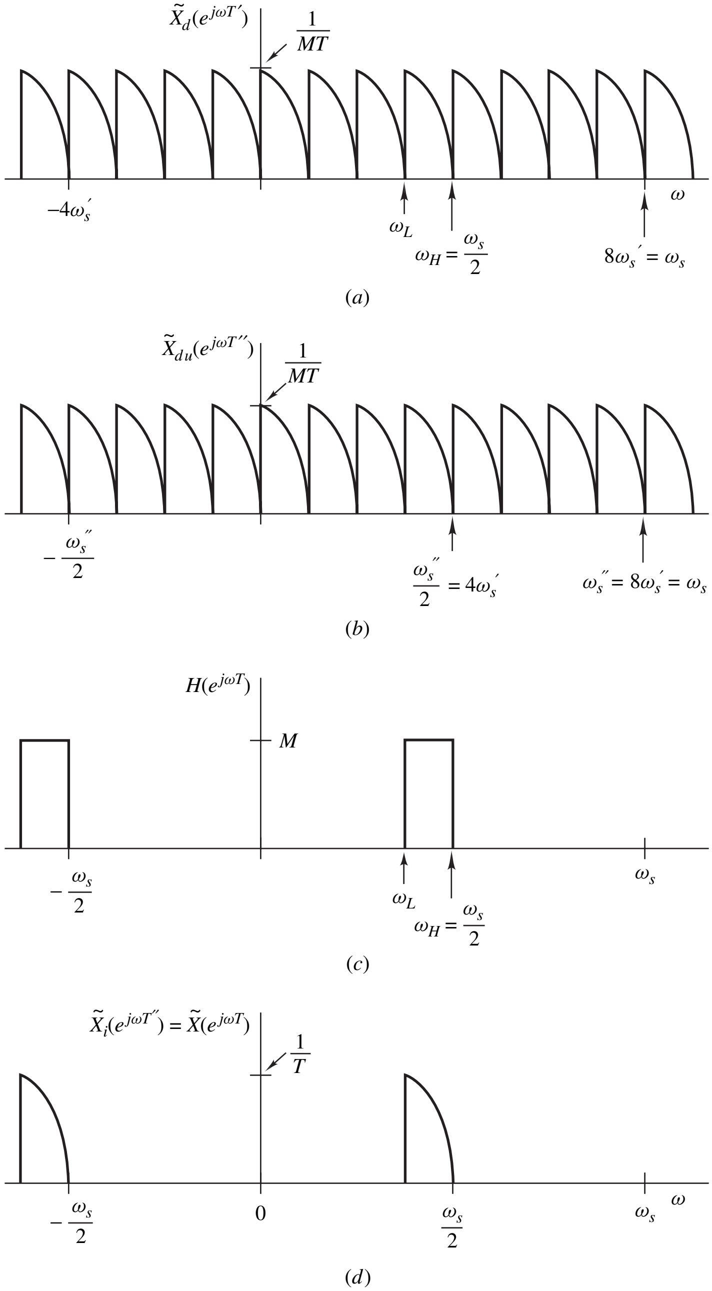


Figure 18.24 Operation of scheme in Fig. 18.22*b*.

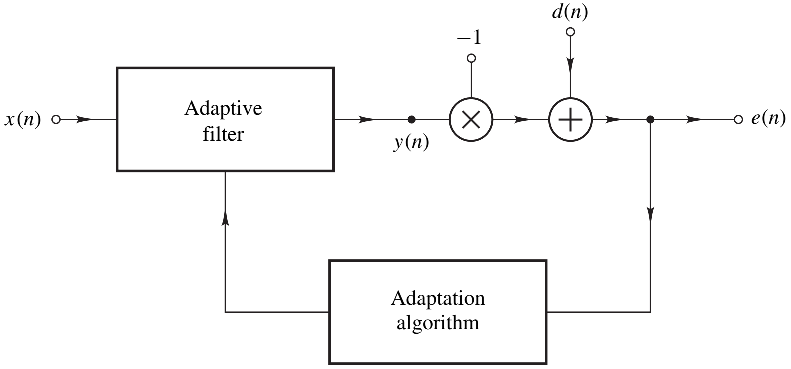


Figure 18.25 Typical adaptive-filter configuration.

In the next section we examine an optimal class of digital filters that are designed by choosing the transfer-function coefficients such that the mean-square value of an error function of the type given in Eq. (18.45) is minimized for some desired response $d(n)$. Filters so designed are commonly referred to as *Wiener filters*, and optimization algorithms that can be used for their design can often be implemented as adaptation algorithms.

18.5.1 Wiener Filters

The simplest structure that can be used for adaptive filters is the direct nonrecursive structure of Fig. 18.26. The output of this configuration is given by

$$y(n) = \sum_{i=0}^{N-1} a_i(n)x(n-i) = \mathbf{a}_n^T \mathbf{x}_n \tag{18.46}$$

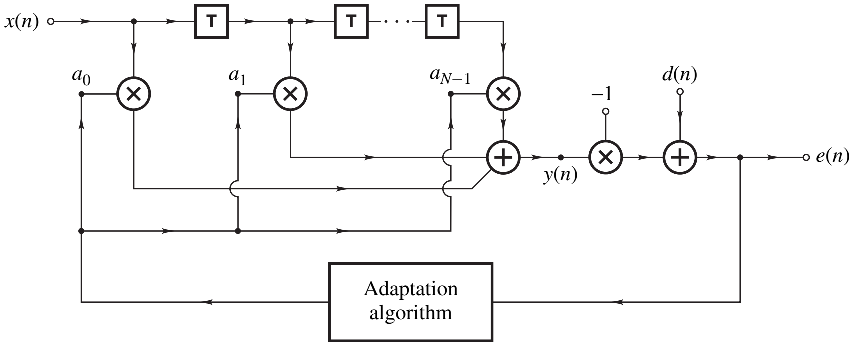


Figure 18.26 Nonrecursive adaptive-filter configuration.

where

$$\mathbf{x}_n = [x(n)\; x(n-1)\; \cdots\; x[n-(N-1)]]^T$$

and

$$\mathbf{a}_n = [a_0(n)\; a_1(n)\; \cdots\; a_{N-1}(n)]^T$$

are the input signal and coefficient vectors, respectively, at instant nT.

The *mean-square error* (MSE) is defined as

$$\mathbf{\Psi}(\mathbf{a}_n) = E[e^2(n)] \tag{18.47}$$

where $E[\cdot]$ is the *expected* value of $[\cdot]$. From Eqs. (18.47), (18.45), and (18.46), we can write

$$\begin{aligned}\mathbf{\Psi}(\mathbf{a}_n) &= E[d^2(n) - 2d(n)y(n) + y^2(n)]\\ &= E\left[d^2(n) - 2d(n)\mathbf{a}_n^T\mathbf{x}_n + \mathbf{a}_n^T\mathbf{x}_n\mathbf{x}_n^T\mathbf{a}_n\right]\\ &= E\left[d^2(n)\right] - 2E\left[d(n)\mathbf{a}_n^T\mathbf{x}_n\right] + E\left[\mathbf{a}_n^T\mathbf{x}_n\mathbf{x}_n^T\mathbf{a}_n\right]\end{aligned}$$

For a filter with fixed coefficients, the MSE function is given by

$$\mathbf{\Psi}(\mathbf{a}_n) = E[d^2(n)] - 2\mathbf{a}_n^T\mathbf{p}_n + \mathbf{a}_n^T\mathbf{R}_n\mathbf{a}_n \tag{18.48}$$

where

$$\mathbf{p}_n = E[d(n)\mathbf{x}_n]$$

and

$$\mathbf{R}_n = E\left[\mathbf{x}_n\mathbf{x}_n^T\right]$$

are the *cross correlation* between the desired and input signals and the correlation matrix of the input signal, respectively, at instant nT. In effect, *the objective function in Eq. (18.47) is a quadratic function of the filter coefficients $a_0(n), a_1(n), \ldots, a_{N-1}(n)$.*

The *gradient* vector of the MSE function can be readily expressed as

$$\mathbf{g}_n = \mathbf{\nabla\Psi}(\mathbf{a}_n) = \left[\frac{\partial\mathbf{\Psi}(\mathbf{a}_n)}{\partial a_0(n)}\; \frac{\partial\mathbf{\Psi}(\mathbf{a}_n)}{\partial a_1(n)}\; \cdots\; \frac{\partial\mathbf{\Psi}(\mathbf{a}_n)}{\partial a_{N-1}(n)}\right]^T \tag{18.49}$$

and from Eqs. (18.48) and (18.49)

$$\mathbf{g}_n = -2\mathbf{p}_n + 2\mathbf{R}_n\mathbf{a}_n \tag{18.50}$$

Now on equating the elements of the gradient vector to zero, the coefficient vector that minimizes the MSE function, say $\breve{\mathbf{a}}$, can be deduced as

$$\breve{\mathbf{a}} = \mathbf{R}_n^{-1}\mathbf{p}_n$$

Evidently, if $\mathbf{p}_n$ and $\mathbf{R}_n$ are known the Wiener solution can be readily obtained. In practice, accurate estimates of $\mathbf{p}_n$ and $\mathbf{R}_n$ are not always available, but *time averages* may be used for their estimation if $d(n)$ and $x(n)$ are stationary and ergodic signals. In such applications, $\mathbf{p}_n$ and $\mathbf{R}_n$ represent a constant vector and a constant matrix, respectively, and the subscript n can be dropped.

Many of the available adaptation algorithms are practical algorithms that lead to the Wiener solution and are borrowed from the field of optimization. Commonly used algorithms are based on the Newton and steepest-descent algorithms.

18.5.2 Newton Algorithm

The Hessian matrix of $\Psi(\mathbf{a}_n)$ can be obtained from Eq. (18.50) as $\mathbf{H}_n = 2\mathbf{R}_n$ and, therefore, the Newton direction (see Sec. 16.3) can be determined as $\mathbf{R}_n^{-1}\mathbf{g}_n/2$. The Wiener solution can, therefore, be approached by obtaining estimates of $\mathbf{a}_{n+1}$ such that

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \alpha\,\mathbf{R}_n^{-1}\tilde{\mathbf{g}}_n \tag{18.51}$$

for $n = 0, 1, \ldots,$ where α is a constant, $\tilde{\mathbf{g}}_n$ is an estimate of $\mathbf{g}_n$, and $\mathbf{a}_0$ is an initial estimate of the transfer-function coefficients. The algorithm is terminated when some convergence criterion is satisfied. This algorithm is essentially the basic Newton algorithm described in detail in Sec. 16.3 except that no line search is used. The constant α, which is sometimes called the *convergence factor*, is chosen to achieve fast convergence for the type of application under consideration. Line searches have not been used in the past owing to the additional amount of computation required, but certain related techniques are likely to be employed in the future.

If $\tilde{\mathbf{g}}_n$ in Eq. (18.51) is the exact gradient, matrix $\mathbf{R}_n$ is a well-behaved positive-definite matrix, and $\alpha = 1/2$, then the Newton algorithm gives the required solution in one iteration, as was demonstrated in Sec. 16.3. However, if $\tilde{\mathbf{g}}_n$ is an approximate estimate of the gradient, a number of iterations is required.

In practice, the Newton algorithm is characterized by a very small number of iterations, but the amount of computation required per iteration is quite large since the inversion of matrix $\mathbf{R}_n$ is required. If $\mathbf{R}_n$ is nearly nonsingular or ill-conditioned, the algorithm can become quite inefficient.

18.5.3 Steepest-Descent Algorithm

If matrix $\mathbf{R}_n$ is assumed to be the $N \times N$ unity matrix, then the updating formula in Eq. (18.51) assumes the form

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \alpha\,\tilde{\mathbf{g}}_n \tag{18.52}$$

If the negative of the gradient vector, namely, $-\mathbf{g}_n$, is drawn through point $\mathbf{a}_n$, it points in the direction of steepest descent, as can be easily shown, and for this reason the use of Eq. (18.52) as updating formula leads to the so-called *steepest-descent* algorithm.

This algorithm is very simple to implement since the Hessian matrix and its inversion are not required. At the start of the adaptation, the algorithm leads to a large reduction in the error function per iteration. However, as the solution is approached, the elements of the gradient become smaller and smaller and progress in the adaptation process tends to slow down considerably; in particular, if the minimum point is located in the middle of a relatively flat valley. Overall, the steepest-descent algorithm is usually less efficient than the Newton algorithm, but the amount of computation per iteration is much smaller. This makes the steepest-descent algorithm more suitable for real-time applications.

If the ratio of the largest to the smallest eigenvalue of matrix $\mathbf{R}_n$ is large, then the solution point tends to follow a zig-zag trajectory in the parameter space and the performance of the algorithm tends to deteriorate. Under certain circumstances, the algorithm can actually become unstable, as demonstrated by the following example.

Example 18.4 (*a*) Show that the steepest-descent algorithm can be treated as an N-input, N-output, first-order digital filter. (*b*) Using the filter obtained, find a necessary and sufficient condition for the stability of the algorithm.

■ **Solution**

(*a*) From Eqs. (18.52) and (18.50)

$$\mathbf{a}_{n+1} = \tilde{\mathbf{R}}_n \mathbf{a}_n + 2\alpha\, \mathbf{p}_n \tag{18.53}$$

where

$$\tilde{\mathbf{R}}_n = \mathbf{I} - 2\alpha\, \mathbf{R}_n$$

Matrix $\tilde{\mathbf{R}}_n$ can be expressed as

$$\tilde{\mathbf{R}}_n = \mathbf{Q}\tilde{\mathbf{\Lambda}}\mathbf{Q}^T$$

where $\mathbf{Q}$ is a unitary matrix whose columns comprise an orthogonal set of eigenvectors associated with the eigenvalues of $\tilde{\mathbf{R}}_n$ and $\tilde{\mathbf{\Lambda}}$ is a diagonal matrix whose diagonal elements are the eigenvalues of $\tilde{\mathbf{R}}_n$. Hence Eq. (18.53) can be put in the form

$$\mathbf{a}_{n+1} = \mathbf{Q}\tilde{\mathbf{\Lambda}}\mathbf{Q}^T \mathbf{a}_n + 2\alpha\, \mathbf{p}_n$$

and on premultiplying both sides by $\mathbf{Q}^T$ and letting

$$\mathbf{a}'_{n+1} = \mathbf{Q}^T \mathbf{a}_{n+1} \qquad \text{and} \qquad \mathbf{p}'_n = 2\alpha\, \mathbf{Q}^T \mathbf{p}_n$$

we obtain

$$\mathbf{a}'_{n+1} = \tilde{\mathbf{\Lambda}}\mathbf{a}'_n + \mathbf{p}'_n \tag{18.54}$$

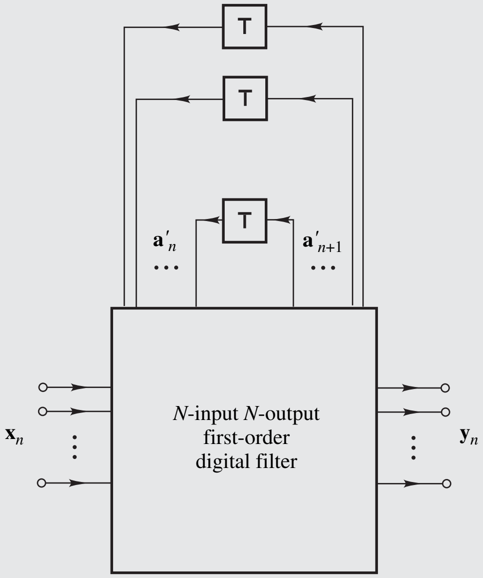


Figure 18.27 N-input, N-output, first-order digital filter (Example 18.4).

This equation represents an N-input, N-output, first-order digital filter with input $\mathbf{x}_n = \mathbf{p}'_n$ and output $\mathbf{y}_n = \mathbf{a}'_{n+1}$, as shown in Fig. 18.27.

(*b*) The algorithm is stable if and only if each of the elements of the impulse response vector $\mathbf{h}_n$ of the filter in Fig. 18.27 is absolutely summable (see Sec. 4.7). From Eq. (18.54)

$$\begin{aligned}
\mathbf{h}_0 &= \mathbf{a}'_1 = \tilde{\mathbf{\Lambda}}\mathbf{a}'_0 + \boldsymbol{\delta}_0 \\
\mathbf{h}_1 &= \mathbf{a}'_2 = \tilde{\mathbf{\Lambda}}\mathbf{a}'_1 + \mathbf{0} = \tilde{\mathbf{\Lambda}}^2\mathbf{a}'_0 + \tilde{\mathbf{\Lambda}}\boldsymbol{\delta}_0 \\
\mathbf{h}_2 &= \mathbf{a}'_3 = \tilde{\mathbf{\Lambda}}\mathbf{a}'_2 + \mathbf{0} = \tilde{\mathbf{\Lambda}}^3\mathbf{a}'_0 + \tilde{\mathbf{\Lambda}}^2\boldsymbol{\delta}_0 \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
\mathbf{h}_n &= \mathbf{a}'_{n+1} = \tilde{\mathbf{\Lambda}}^{n+1}\mathbf{a}'_0 + \tilde{\mathbf{\Lambda}}^n\boldsymbol{\delta}_0
\end{aligned}$$

Without loss of generality, we can assume that $\mathbf{a}_0 = 0$ or $\mathbf{a}'_0 = 0$ and hence

$$\mathbf{h}_n = \tilde{\mathbf{\Lambda}}^n\boldsymbol{\delta}_0$$

Thus, the ith element of the impulse response vector $\mathbf{h}_n$ is obtained as

$$h_i(n) = \tilde{\lambda}_i^n \tag{18.55}$$

The filter is stable if and only if

$$\sum_{n=0}^{\infty} |h_i(n)| < \infty \qquad \text{for } i = 1, 2, \ldots, N \tag{18.56}$$

and, therefore, a necessary and sufficient condition for stability can be obtained from Eqs. (18.55) and (18.56) as

$$|\tilde{\lambda}_i| < 1 \qquad \text{for } i = 1, 2, \ldots, N \tag{18.57}$$

Now the eigenvalues of $\tilde{\mathbf{R}}_n$ are related to those of $\mathbf{R}_n$ by the equation

$$\tilde{\lambda}_i = 1 - 2\alpha\,\lambda_i$$

(see Prob. 18.21) and hence the inequality in Eq. (18.57) can be expressed as

$$|1 - 2\alpha\lambda_i| < 1 \qquad \text{for } i = 1, 2, \ldots, N$$

Since the eigenvalues of $\mathbf{R}_n$ are real and positive, the steepest-descent algorithm is stable if and only if

$$0 < \alpha < 1/\hat{\lambda}$$

where $\hat{\lambda}$ is the largest eigenvalue of $\mathbf{R}_n$.

Improved performance can often be achieved, in practice, by starting the adaptation process with the steepest-descent algorithm and, when certain progress has been achieved, switching over to the Newton algorithm, which is more efficient at points in the neighborhood of the solution point.

18.5.4 Least-Mean-Square Algorithm

As can be seen in Eq. (18.50), the gradient depends on vector $\mathbf{p}_n$ and matrix $\mathbf{R}_n$, which are not in general available. Nevertheless, an estimate of the gradient $\tilde{\mathbf{g}}_n$ can be deduced by letting

$$\tilde{\mathbf{p}}_n = d(n)\mathbf{x}_n \tag{18.58}$$

and

$$\tilde{\mathbf{R}}_n = \mathbf{x}_n\mathbf{x}_n^T \tag{18.59}$$

be estimates of **p** and **R**, respectively. From Eqs (18.50), (18.58), (18.59), (18.45), and (18.46), we can write

$$\begin{aligned}\tilde{\mathbf{g}}_n &= -2d(n)\mathbf{x}_n + 2\mathbf{x}_n\mathbf{x}_n^T\mathbf{a}_n \\ &= -2\left[d(n) - \mathbf{x}_n^T\mathbf{a}_n\right]\mathbf{x}_n \\ &= -2e(n)\mathbf{x}_n \qquad (18.60)\end{aligned}$$

On using this estimate of the gradient in the updating formula of the steepest-descent algorithm given in Eq. (18.52), we obtain

$$\mathbf{a}_{n+1} = \mathbf{a}_n + 2\alpha\, e(n)\mathbf{x}_n \qquad (18.61)$$

The use of this formula yields the so-called *least-mean-square* (LMS) algorithm. The convergence factor α is chosen in the range

$$0 < \alpha < \frac{1}{NE[x^2(n)]} \qquad (18.62)$$

in order to guarantee the convergence of the algorithm (see Prob. 18.23). The quantity $E[x^2(n)]$ represents the average input power and is usually easy to estimate.

Note that

$$\boldsymbol{\nabla} e^2(n) = 2e(n)\left[\frac{\partial e(n)}{\partial a_0(n)}\ \frac{\partial e(n)}{\partial a_1(n)}\ \cdots\ \frac{\partial e(n)}{\partial a_{N-1}(n)}\right]^T$$

and from Eqs. (18.45), (18.46), and (18.60)

$$\boldsymbol{\nabla} e^2(n) = -2e(n)\mathbf{x}_n = \tilde{\mathbf{g}}_n$$

that is, $\tilde{\mathbf{g}}_n$ is the exact gradient of $e^2(n)$ and thus the LMS algorithm minimizes the instantaneous power of the error signal.

Further details on the above algorithms as well as some others can be found in [5–8].

18.5.5 Recursive Filters

In applications where high selectivity is required, a nonrecursive design would necessitate a transfer function of high order, which may entail a high computational complexity. For such applications, a recursive design may be the only possible solution. Unfortunately, however, the use of recursive structures introduces several new problems. First, the coefficients of the denominator polynomial of the transfer function may assume values in the unstable region of the coefficient space and, if this happens, the adaptive filter will become unstable. Second, the objective function becomes highly nonlinear and may have several local minima, some of them quite shallow. Hence the adaptation algorithm may easily converge to some unsatisfactory solution.

The problem of instability can be overcome by incorporating checks in the adaptation algorithm that can detect an unstable solution and restore stability by making a suitable adjustment to the current coefficient values. For example, the coefficients can be adjusted in such a way as to replace any poles outside the unit circle of the z plane by their reciprocals (see Sec. 11.4). The problem of several local minima, which is quite difficult to solve, can be eased to some extent by selecting a well-behaved objective function and by using a good estimate of the solution for the initialization of $\mathbf{a}_0$. For most practical problems, the objective function is well behaved in the neighborhood of the optimum solution, and an initial point in this domain will cause the adaptation algorithm to converge to the optimum solution.

The design of adaptive filters based on recursive structures has been studied by a number of researchers in recent years [26–27] and more work is anticipated in the future.

18.5.6 Applications

The applications of adaptive filters are numerous and include system identification, channel equalization, signal enhancement, and signal prediction.

In a *system identification* application, a broadband signal, usually white noise, is applied simultaneously at the inputs of an unknown system and an adaptive filter, and an error signal is formed by subtracting the output of the adaptive filter from that of the unknown system, as depicted in Fig. 18.28. If the error signal obtained is minimized, the adaptive filter becomes a model for the unknown system.

The transmission of a signal through an imperfect channel entails amplitude and phase distortion, as was shown in Sec. 5.7. If the frequency response of the channel is known, it can be equalized using a fixed filter, as described in Secs. 12.5.1 and 12.6. However, if the channel response is variable, equalization by means of an adaptive filter is more appropriate. A variable channel response can occur in telephony where the physical wire path between any two subscribers depends not only on the locations of the two subscribers but also on the time of the call. *Channel equalization* can

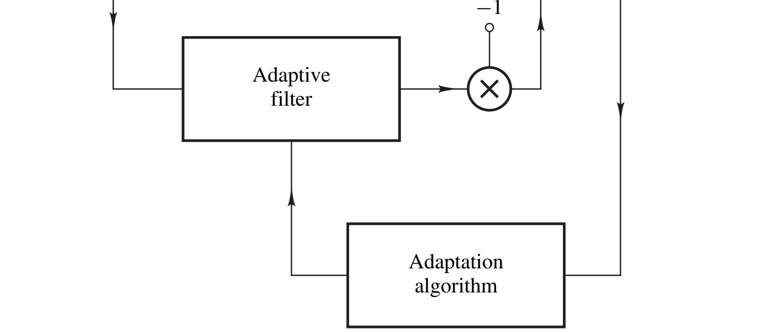


Figure 18.28 Use of adaptive filters for system identification.

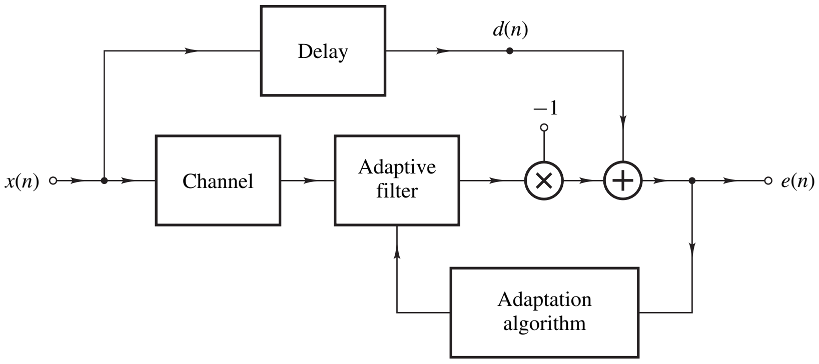


Figure 18.29 Use of adaptive filters for channel equalization.

be achieved by connecting an adaptive filter in cascade with the channel and comparing the output of the cascade arrangement with a delayed version of the input signal, as illustrated in Fig. 18.29. In this application, the reference signal is a training signal and is known at the receiving end, i.e., it need not be transmitted.

In *signal enhancement* applications, a signal $x(n)$ corrupted by a noise component $n_1(n)$, namely,

$$d(n) = x(n) + n_1(n)$$

is used as a reference signal, and a signal $n_2(n)$ that is correlated to the noise component $n_1(n)$ is applied to the input of an adaptive filter, as depicted in Fig. 18.30. After adaptation, the error signal $e(n)$ will represent an enhanced version of the signal $x(n)$ in which a significant amount of noise has been removed.

In *signal prediction* applications, the signal of interest is used as the reference signal and a delayed version is applied to the input of an adaptive filter. The error signal is generated by subtracting the output of the adaptive filter from the reference signal, as shown in Fig. 18.31. After convergence,

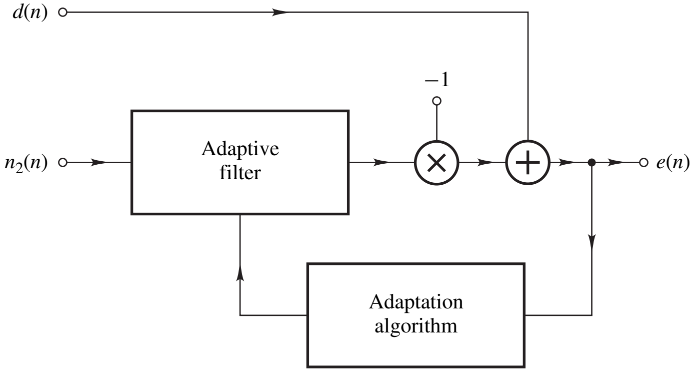


Figure 18.30 Use of adaptive filters for signal enhancement.

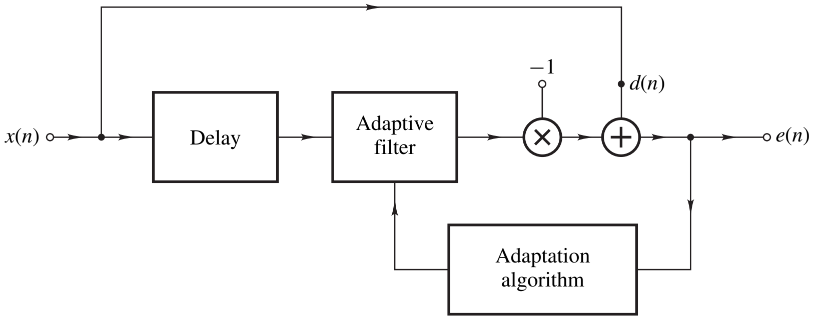


Figure 18.31 Use of adaptive filters for signal prediction.

the filter coefficients are adjusted in response to the past signal values and can, therefore, be used to reconstruct or extrapolate the input signal. Signal prediction is used in speech coding, where a model of the signal rather than the signal itself is encoded and transmitted.

18.6 TWO-DIMENSIONAL DIGITAL FILTERS

In many applications, continuous signals are encountered that are functions of two independent variables, say, t_1 and t_2. Two-dimensional signals of this type can be represented by functions of the form $x(t_1, t_2)$. Each of the two variables can represent an arbitrary physical quantity such as time, length, velocity, acceleration, and temperature. An example of a 2-D continuous signal is the light intensity in an image as a function of the x and y coordinates.

Two-dimensional continuous signals, like their 1-D counterparts, can be represented by frequency spectrums. Furthermore, they can be sampled at discrete points $(t_1, t_2) = (n_1T_1, n_2T_2)$ to yield discrete[5] signals $x(n_1T_1, n_2T_2)$. Signals of this type have frequency spectrums that are periodic with respect to frequencies ω_1 and ω_2 with a 2-D period $(\omega_{s1}, \omega_{s2})$, where $\omega_{s1} = 2\pi/T_1$ and $\omega_{s2} = 2\pi/T_2$.

Two-dimensional discrete signals can be processed by *2-D digital filters* whose operation and properties are analogous to those of their 1-D counterparts, that is, they can be time invariant or time dependent, causal or noncausal, and linear or nonlinear. Since neither of the two independent variables needs to be time, causality does not have the usual physical interpretation.

The theory, analysis, and design methods, and applications of 2-D digital filters are quite extensive but are largely beyond the scope of this book. Below, we present some of the basic principles involved and some of the straightforward extensions of 1-D methods and techniques for the sake of completeness. The reader with more than a casual interest in the analysis and design of 2-D digital filters is referred to the more specialized books on the subject cited earlier, namely, Refs. [9–11].

A 2-D causal recursive (or IIR) digital filter with excitation $x(n_1, n_2)$ and response $y(n_1, n_2)$ can be represented by a difference equation in two variables of the form

$$y(n_1, n_2) = \sum_{i=0}^{N_1}\sum_{j=0}^{N_2} a_{ij}x(n_1 - i, n_2 - j) - \sum_{i=0}^{N_1}\sum_{j=0}^{N_2} b_{ij}y(n_1 - i, n_2 - j)$$

[5] *Discrete-time is changed to discrete since neither of the two variables needs to be time.*

where $b_{00} = 0$. If $b_{ij} = 0$ for $0 \leq i \leq N_1$ and $0 \leq j \leq N_2$, the representation of a 2-D nonrecursive (or FIR) filter is obtained. The pair (N_1, N_2) is the *order* of the filter.

18.6.1 Two-Dimensional Convolution

If the impulse response of a 2-D filter $h(n_{1,}n_2)$ is known, then its response $y(n_1, n_2)$ to an arbitrary excitation $x(n_1, n_2)$ can be determined by using the *2-D convolution*. If $h(n_1, n_2) = x(n_1, n_2) = 0$ for $n_1 < 0$ or $n_2 < 0$, then

$$\begin{aligned} y(n_1, n_2) &= \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} x(i, j)h(n_1 - i, n_2 - j) \\ &= \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} h(i, j)x(n_1 - i, n_2 - j) \end{aligned}$$

This formula can be derived by following the approach of Sec. 4.6.

18.6.2 Two-Dimensional z Transform

The most important mathematical tool for the analysis and design of 2-D digital filters is the *2-D z transform*, which is a straightforward extension of its 1-D counterpart. The 2-D z transform of a function $f(n_1, n_2)$ is defined as

$$F(z_1, z_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f(n_1, n_2) z_1^{-n_1} z_2^{-n_2}$$

for all (z_1, z_2) for which the double summation converges. Function $f(n_1, n_2)$ is the 2-D inverse z transform of $F(z_1, z_2)$ and is given by

$$f(n_1, n_2) = \frac{1}{(2\pi j)^2} \oint_{\Gamma_2} \oint_{\Gamma_1} F(z_1, z_2) z_1^{n_1-1} z_2^{n_2-1} \, dz_1 \, dz_2 \tag{18.63}$$

where the two integrals are evaluated in the counterclockwise sense over contours Γ_1 and Γ_2 that are in the region of convergence of $F(z_1, z_2)$.

18.6.3 Two-Dimensional Transfer Function

The transfer function of a 2-D digital filter is the z transform of the impulse response, as can be shown by applying the z transform to the convolution summation. It can be expressed as

$$H(z_1, z_2) = \frac{N(z_1, z_2)}{D(z_1, z_2)} = \frac{\sum_{i=0}^{N_1} \sum_{j=0}^{N_2} a_{ij} z_1^{N_1-i} z_2^{N_2-j}}{z_1^{N_1} z_2^{N_2} + \sum_{i=0}^{N_1} \sum_{j=0}^{N_2} b_{ij} z_1^{N_1-i} z_2^{N_2-j}}$$

where $b_{00} = 0$. The transfer function can be used to find the response of the filter to an arbitrary excitation and its frequency-domain response; furthermore, it contains all the necessary information to determine whether the filter is stable or unstable.

18.6.4 Stability

The *stability* of a 2-D digital filter is closely linked with the singularities of the transfer function, as in 1-D filters. Unfortunately, in 2-D digital filters the singularities are not in general isolated and, as a result, stability analysis is much more complicated. A sufficient condition for the stability of a 2-D filter due to Shanks [28, 29] is that all the singularities of the transfer function are located on the open unit bidisc defined by the set

$$U^2 = \{(z_1, z_2) : |z_1| < 1, |z_2| < 1\}$$

i.e.,

$$D(z_1, z_2) \neq 0 \quad \text{for} \;\; (z_1, z_2) \notin U^2 \tag{18.64}$$

where

$$D(z_1, z_2) = z_1^{N_1} z_2^{N_2} + \sum_{i=0}^{N_1} \sum_{j=0}^{N_2} b_{ij} z_1^{N_1 - i} z_2^{N_2 - j}$$

with $b_{00} = 0$.

Example 18.5 A 2-D digital filter is characterized by the transfer function

$$H(z_1, z_2) = \frac{N(z_1, z_2)}{D(z_1, z_2)}$$

where

$$N(z_1, z_2) = 512(z_1 + 1)^2 (z_2 + 1)^2$$

and

$$D(z_1, z_2) = 512 z_1^2 z_2^2 - 128 z_1 z_2^2 + 256 z_1^2 z_2 - 192 z_2^2 - 40 z_1^2 - 64 z_1 z_2$$
$$+10 z_1 - 96 z_2 + 15$$

Check the stability of the filter.

■ **Solution**

The transfer function can be expressed as

$$H(z_1, z_2) = \frac{N'(z_1, z_2)}{D'(z_1, z_2)}$$

where

$$N'(z_1, z_2) = (z_1 + 1)^2 (z_2 + 1)^2$$

and

$$D'(z_1, z_2) = z_1^2 z_2^2 - \tfrac{1}{4} z_1 z_2^2 + \tfrac{1}{2} z_1^2 z_2 - \tfrac{3}{8} z_2^2 - \tfrac{5}{64} z_1^2 - \tfrac{1}{8} z_1 z_2 + \tfrac{5}{256} z_1 - \tfrac{3}{16} z_2 + \tfrac{15}{512}$$

The denominator polynomial $D'(z_1, z_2)$ can now be put in the form

$$\begin{aligned} D'(z_1, z_2) &= \left(z_2^2 + \tfrac{1}{2} z_2 - \tfrac{5}{64}\right) z_1^2 - \left(\tfrac{1}{4} z_2^2 + \tfrac{1}{8} z_2 - \tfrac{5}{256}\right) z_1 - \left(\tfrac{3}{8} z_2^2 + \tfrac{3}{16} z_2 - \tfrac{15}{512}\right) \\ &= \left(z_1^2 - \tfrac{1}{4} z_1 - \tfrac{3}{8}\right)\left(z_2^2 + \tfrac{1}{2} z_2 - \tfrac{5}{64}\right) \\ &= \left(z_1 + \tfrac{1}{2}\right)\left(z_1 - \tfrac{3}{4}\right)\left(z_2 + \tfrac{5}{8}\right)\left(z_2 - \tfrac{1}{8}\right) \end{aligned}$$

Hence the transfer function $H(z_1, z_2)$ is singular only at points

$$(z_1, z_2) = \begin{cases} \left(-\tfrac{1}{2}, z_2\right) & \left(z_1, -\tfrac{5}{8}\right) \\ \left(\tfrac{3}{4}, z_2\right) & \left(z_1, \tfrac{1}{8}\right) \end{cases}$$

Therefore, $D(z_1, z_2)$ satisfies Eq. (18.64) and as a consequence the filter is stable.

If the denominator of the transfer function can be factorized into a product of polynomials of the form $D_1(z_1)D_2(z_2)$ where $D_1(z_1)$ and $D_2(z_2)$ are polynomials in z_1 and z_2, respectively, as in the above example, the stability of the filter can be easily checked by using the stability criteria of Sec. 5.3.3, for example, by applying the Jury-Marden stability criterion. However, if $D(z_1, z_2)$ cannot be factorized, the stability analysis can be quite involved (see Chap. 5 of Ref. [11]).

18.6.5 Frequency-Domain Analysis

The *frequency response* of a 2-D filter is given by

$$H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2}) = M(\omega_1, \omega_2) e^{j\theta(\omega_1, \omega_2)}$$

where

$$M(\omega_1, \omega_2) = |H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2})| \tag{18.65}$$

and

$$\theta(\omega_1, \omega_2) = \arg H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2}) \tag{18.66}$$

are the amplitude and phase response, respectively. A pair of parameters that are sometimes of interest in 2-D digital filters are the *group delays*. These are defined as

$$\tau_1 = -\frac{\partial\theta(\omega_1, \omega_2)}{\partial\omega_1} \qquad \text{and} \qquad \tau_2 = -\frac{\partial\theta(\omega_1, \omega_2)}{\partial\omega_2} \tag{18.67}$$

Example 18.6 A 2-D nonrecursive digital filter designed by using the method in Ref. [30] has the transfer function

$$H(z_1, z_2) = \sum_{i=1}^{4} (-1)^{i+1} H_i(z_1) H_i(z_2)$$

where

$$H_i(z_k) = \sum_{j=0}^{24} a_{ij} z_k^{-j} \qquad \text{for } k = 1, 2$$

and

$$a_{ij} = a_{i(24-j)} \qquad \text{for } i = 1, 2, 3, 4 \tag{18.68}$$

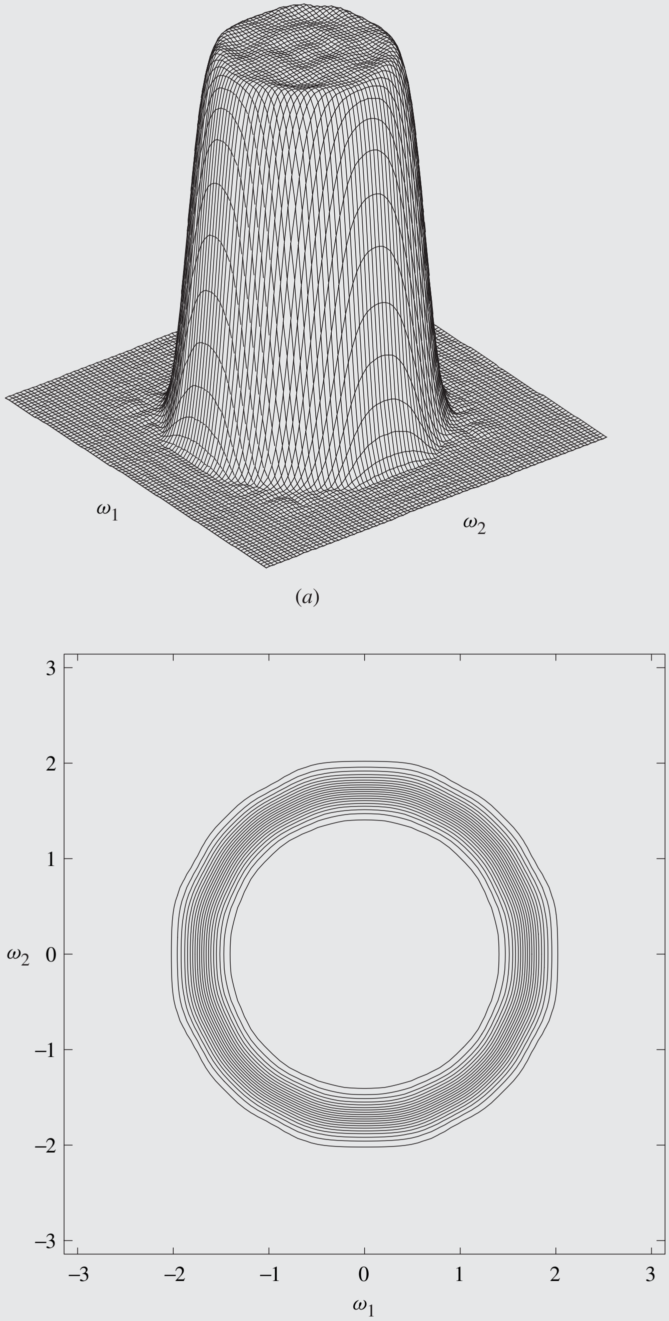
The coefficients a_{ij} are given in Table 18.3. Obtain the amplitude response of the filter.

Table 18.3 Coefficients of 2-D transfer function (Example 18.6)

j	a_{1j}	a_{2j}	a_{3j}	a_{4j}
0	0.0007	0.0012	0.0031	0.0027
1	−0.0004	−0.0009	0.0006	0.0034
2	−0.0013	−0.0033	−0.0065	−0.0063
3	0.0022	0.0043	−0.0009	−0.0099
4	0.0025	0.0105	0.0173	0.0155
5	−0.0086	−0.0119	−0.0019	0.0301
6	−0.0036	−0.0206	−0.0474	−0.0109
7	0.0241	0.0374	−0.0154	−0.0460
8	−0.0007	0.0549	0.0746	−0.0214
9	−0.0688	−0.0713	0.0744	−0.0007
10	0.0189	−0.1635	0.0015	−0.0137
11	0.3145	−0.0519	−0.0028	−0.0049
12	0.4935	0.0544	0.0250	0.0124

■ **Solution**

The amplitude response can be obtained as shown in Fig. 18.32, by using Eq. (18.65). As can be seen, the given transfer function represents a lowpass filter with a circular passband. Equation (18.68) amounts to symmetrical impulse responses in the 1-D filters, represented by $H_1(z_1)$ and $H_2(z_2)$, that is, these filters have a linear phase response (see Sec. 9.2). Therefore, the phase response of the 2-D filter, given by Eq. (18.66), is linear with respect to both ω_1 and ω_2, and the group delays in Eq. (18.67) are constant throughout the baseband. It should be mentioned in passing that a linear phase response is highly desirable in image processing applications.



(*a*)

(*b*)

Figure 18.32 Amplitude response of 2-D digital filter (Example 18.6): (*a*) 3-D plot, (*b*) contour plot.

18.6.6 Types of 2-D Filters

As in the case of 1-D filters, different types of 2-D filters can be identified on the basis of their amplitude responses, e.g., lowpass, highpass, bandpass, and bandstop. Passbands and stopbands are now subareas of the $(\omega_{1,}\omega_2)$ plane and can be rectangular or circular. A 2-D lowpass filter has an amplitude response of the form

$$M(\omega_1,\omega_2) \approx \begin{cases} 1 & \text{for } (\omega_1,\omega_2) \in R_1 \\ 0 & \text{for } (\omega_1,\omega_2) \in R_2 \end{cases}$$

where

$$R_1 = \{(\omega_1,\omega_2)\colon\ |\omega_1| \le \omega_{p1} \text{ and } |\omega_2| \le \omega_{p2}\}$$

and

$$R_2 = \{(\omega_1,\omega_2)\colon\ |\omega_1| \ge \omega_{a1} \text{ or } |\omega_2| \ge \omega_{a2}\}$$

in a filter with rectangular band boundaries or

$$R_1 = \left\{(\omega_1,\omega_2)\colon\ \sqrt{\omega_1^2+\omega_2^2} < \omega_p\right\}$$

and

$$R_2 = \left\{(\omega_1,\omega_2)\colon\ \sqrt{\omega_1^2+\omega_2^2} > \omega_a\right\}$$

in a filter with circular band boundaries. A 2-D highpass filter, on the other hand, has an amplitude response

$$M(\omega_1,\omega_2) \approx \begin{cases} 0 & \text{for } (\omega_1,\omega_2) \in R_1 \\ 1 & \text{for } (\omega_1,\omega_2) \in R_2 \end{cases}$$

where R_1 and R_2 are as above.

Another type of filter that has no counterpart in the 1-D domain is the so called *fan filter*. A fan filter has an amplitude response

$$M_1(\omega_1,\omega_2) \approx \begin{cases} 1 & \text{for } (\omega_1,\omega_2) \in S_1 \\ 0 & \text{for } (\omega_1,\omega_2) \in S_2 \end{cases}$$

where

$$S_1 = \left\{(\omega_1,\omega_2)\colon\ \left|\tan^{-1}\frac{\omega_2}{\omega_1}\right| < \theta_1 \text{ or } \left|\tan^{-1}\frac{\omega_2}{\omega_1}\right| > \pi - \theta_1\right\}$$

and

$$S_2 = \left\{(\omega_1, \omega_2)\colon \left|\tan^{-1}\frac{\omega_2}{\omega_1}\right| > \theta_2 \text{ or } \left|\tan^{-1}\frac{\omega_2}{\omega_1}\right| < \pi - \theta_2\right\}$$

with $\theta_2 > \theta_1$.

18.6.7 Approximations

The most difficult task in the design of 2-D digital filters is the solution of the *approximation* problem, which entails the derivation of a stable transfer function such that prescribed amplitude and/or phase response specifications are achieved. As in 1-D filters, the approximation problem can be solved by using direct or indirect methods in terms of closed-form or iterative solutions.

Nonrecursive filters can be designed by using the 2-D Fourier series in conjunction with 2-D window functions [31] (see Secs. 9.3 and 9.4) or by using a transformation due to McClellan [32, 33]. Recursive filters, on the other hand, can be designed by applying transformations to 1-D filters [34, 35]. Nonrecursive as well as recursive filters can be designed by using the singular-value decomposition [30, 36] or through the application of optimization methods [37–40].

If the numerator and denominator of the transfer function can be factorized into products $N_1(z_1)N_2(z_2)$ and $D_1(z_1)D_2(z_2)$, then the transfer function is said to be *separable* and can be expressed as

$$H(z_1, z_2) = H_1(z_1, z_2)H_2(z_1, z_2)$$

where

$$H_1(z_1, z_2) = \frac{N_1(z_1, z_2)}{D_1(z_1, z_2)} \qquad \text{and} \qquad H_2(z_1, z_2) = \frac{N_2(z_1, z_2)}{D_2(z_1, z_2)}$$

Filters of this class can be readily designed using the approximation techniques for 1-D digital filters described in the previous chapters, and they are suitable for applications where rectangular band boundaries are acceptable. However, if the transfer function is not separable, as may be the case in filters with circular band boundaries, the design is much more involved.

18.6.8 Applications

Two-dimensional digital filters are useful in several areas. Lowpass filters can be used for the *reduction of noise* in images for the same reasons as their 1-D counterparts. Use is made of the fact that the information content of the 2-D signal is often concentrated at low frequencies, whereas noise tends to be distributed throughout the baseband. Highpass filters are sometimes used for the *enhancement of edges in images*; their application is based on the fact that abrupt changes in an image tend to increase the high-frequency content of an image, and its amplification by a highpass filter tends to exaggerate edges or outlines. Edge enhancement finds applications in pattern recognition, surveying, and computer vision. Fan filters have been found very useful for the *processing of geophysical signals*; for example, they can enhance the quality of seismic signals by eliminating signal components that are not associated with the subsurface ground formations. Seismic signals are indispensable for oil prospecting and other geological applications [11].

REFERENCES

[1] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1983.

[2] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[3] T. Nguyen, "Digital filter bank design quadratic-constrained formulation," *IEEE Trans. Signal Processing*, vol. 43, pp. 2103–2108, Sept. 1995.

[4] P. Heller, T. Karp, and T. Nguyen, "A general formulation of modulated filter banks," *IEEE Trans. Signal Processing*, vol. 47, pp. 986–1002, Apr. 1999.

[5] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.

[6] P. A. Regalia, *Adaptive IIR Filtering for Signal Processing Control*, New York: Marcel Dekker, 1995.

[7] S. Haykin, *Adaptive Filter Theory*, 4th ed., Englewood Cliffs, NJ: Prentice-Hall, 2002.

[8] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 2th ed., Boston: Kluwer Academic Publishers, 2002.

[9] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[10] J. S. Lim, *Two-Dimensional Signal and Image Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1990.

[11] W.-S. Lu and A. Antoniou, *Two-Dimensional Digital Filters*, New York: Marcel Dekker, 1992.

[12] H. Scheuermann and H. Gockler, "A comprehensive survey of digital transmultiplexing methods," *Proc. IEEE*, vol. 69, pp. 1419–1450, Nov. 1981.

[13] N. S. Jayant and P. Noll, *Digital Coding of Waveforms,* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[14] J. D. Johnson and R. E. Crochiere, "An all-digital commentary grade sub-band coder," *J. Audio Eng. Soc.*, vol. 27, pp. 855–865, Nov. 1979.

[15] J. D. Johnson, "A filter family designed for use in quadrature mirror filter banks," *in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 1984*, pp. 291–294, Apr. 1980.

[16] V. K. Jain and R. E. Crochiere, "Quadrature mirror filter design in the time domain," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, pp. 353–361, Apr. 1984.

[17] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial," *Proc. IEEE*, vol. 78, pp. 56–93, Jan. 1990.

[18] M. J. T. Smith and T. P. Barnwell, III, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, pp. 434–441, June 1986.

[19] B. Gold, A. V. Oppenheim, and C. M. Rader, "Theory and implementation of the discrete Hilbert transform," *Proc. Symp. Computer Process. in Comm.*, vol. 19, pp. 235–250, New York: Polytechnic Press, 1970. (See also *Digital Signal Processing*, edited by L. R. Rabiner and C. M. Rader, IEEE Press, pp. 94–109, 1972.)

[20] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[21] D. T. M. Slock and T. Kailath, “Numerically stable fast transversal filters for recursive least squares adaptive filtering,” *IEEE Trans. Signal Processing*, vol. 39, pp. 92–114, Jan. 1991.

[22] P. A. Regalia and M. G. Bellanger, “On the duality between fast QR methods and lattice methods in least squares adaptive filtering,” *IEEE Trans. Signal Processing*, vol. 39, pp. 879–891, Apr. 1991.

[23] G. Carayannis, D. G. Manolakis, and N. Kalouptsidis, “A fast sequential algorithm for least-squares filtering and prediction,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, pp. 1394–1402, Dec. 1983.

[24] P. A. Regalia, “Stable and efficient lattice algorithms for adaptive IIR filtering,” *IEEE Trans. Signal Processing*, vol. 40, pp. 375–388, Feb. 1992.

[25] M. G. Bellanger, “FLS-QR algorithm for adaptive filtering,” *Signal Processing*, vol. 17, pp. 291–304, 1989.

[26] J. J. Shynk, “Adaptive IIR filtering,” *IEEE ASSP Magazine*, vol. 6, pp. 4–21, Apr. 1989.

[27] M. Nayeri and W. K. Jenkins, “Alternate realizations to adaptive IIR filters and properties of their performance surfaces,” *IEEE Trans. Circuits Syst.*, vol. 36, pp. 485–496, Apr. 1989.

[28] J. L. Shanks, “Two-dimensional recursive filters,” *SWIEECO Rec.*, pp. 19E1–19E8, 1969.

[29] J. L. Shanks, S. Treitel, and J. H. Justice, “Stability and synthesis of two-dimensional recursive filters,” *IEEE Trans. Audio Electroacoust.*, vol. 20, pp. 115–128, June 1972.

[30] A. Antoniou and W.-S. Lu, “Design of two-dimensional digital filters by using the singular value decomposition,” *IEEE Trans. Circuits Syst.*, vol. 34, pp. 1191–1198, Oct. 1987.

[31] T. S. Huang, “Two-dimensional windows,” *IEEE Trans. Audio Electroacoust.*vol. 20, pp. 88–89, Mar. 1972.

[32] J. H. McClellan, “The design of two-dimensional digital filters by transformations,” *Proc. 7th Annual Princeton Conf. Information Sciences and Systems*, pp. 247–251, 1973.

[33] R. M. Mersereau, W. F. G. Mecklenbräuker, and T. F. Quatieri, Jr., “McClellan transformations for two-dimensional digital filtering: I—Design,” *IEEE Trans. Circuits Syst.*, vol. 23, pp. 405–413, July 1976.

[34] J. M. Costa and A. N. Venetsanopoulos, “Design of circularly symmetric two-dimensional recursive filters,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 22, pp. 432–443, Dec. 1974.

[35] D. M. Goodman, “A design technique for circularly symmetric low-pass filters,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, pp. 290–304, Aug. 1978.

[36] W.-S. Lu, H.-P. Wang, and A. Antoniou, “Design of two-dimensional FIR digital filters by using the singular value decomposition,” *IEEE Trans. Circuits Syst.*, vol. 37, pp. 35–46, Jan. 1990.

[37] G. A. Maria and M. M. Fahmy, “An l_p design technique for two-dimensional digital recursive filters,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 22, pp. 15–21, Feb. 1974.

[38] P. A. Ramamoorthy and L. T. Bruton, “Design of stable two-dimensional analogue and digital filters with applications in image processing,” *Int. J. Circuit Theory Appl.*, vol. 7, pp. 229–245, 1979.

[39] C. Charalambous, “The performance of an algorithm for minimax design of two-dimensional linear phase FIR digital filters,” *IEEE Trans. Circuits Syst.*, vol. 32, pp. 1016–1028, Oct. 1985.

[40] C. Charalambous, “Design of 2-dimensional circularly-symmetric digital Filters,” *Proc. Inst. Elect. Eng.*, Part G, vol. 129, pp. 47–54, Apr. 1982.

ADDITIONAL REFERENCES

Friedlander, B, "Lattice filters for adaptive processing," *Proc. IEEE*, vol. 70, pp. 829–867, Aug. 1982.

Gilloire, A. and M. Vetterli, "Adaptive filtering in subbands with critical sampling: Analysis, experiments, and applications to acoustic echo cancellation," *IEEE Trans. Signal Processing*, vol. 40, pp. 1862–1875, Aug. 1992.

Glentis, G. O., K. Berberidis, and S. Theodoridis, "Efficient least-squares adaptive algorithms for FIR transversal filtering," *IEEE Signal Processing Magazine*, vol. 16, pp. 13–41, July 1999.

Johns, D. A., W. M. Snelgrove, and A. S. Sedra, "Adaptive recursive state-space filters using a gradient-based algorithm," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 673–683, June 1990.

Johnson, Jr., C. R, "On the interaction of adaptive filtering, identification, and control," *IEEE Signal Processing Magazine*, vol. 12, pp. 22–37, Mar. 1995.

Koilpillai, R. D. and P. P. Vaidyanathan, "Cosine-modulated FIR filter banks satisfying perfect reconstruction," *IEEE Trans. Signal Processing*, vol. 40, pp. 770–783, Apr. 1992.

Lin, Y.-P. and P. P. Vaidyanathan,"Linear phase cosine modulated maximally decimated filter banks with perfect reconstruction," *IEEE Trans. Signal Processing*, vol. 42, pp. 2525–2539, Nov. 1995.

Marshall, D. F., W. K. Jenkins, and J. J. Murphy, "The use of orthogonal transforms for improving performance of adaptive filters," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 474–483, Apr. 1989.

Mathews, V. J., "Adaptive polynomial filters," *IEEE Signal Processing Magazine*, vol. 8, pp. 10–26, July 1991.

Shynk, J. J., "Adaptive IIR filtering using parallel-form realizations," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, pp. 519–533, Apr. 1989.

Shynk, J. J., "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, pp. 14–37, Jan. 1992.

PROBLEMS

18.1. The input signal $x(nT)$ in the downsampler of Fig. 18.2*a* has the real frequency spectrum depicted in Fig. P18.1 and $\omega_s = 20$ rad/s.

(*a*) Sketch the frequency spectrum of $x_d(nT')$ if $M = 2$.

(*b*) Repeat part (*a*) if $M = 4$.

(*c*) Comment on the answers obtained in parts (*a*) and (*b*).

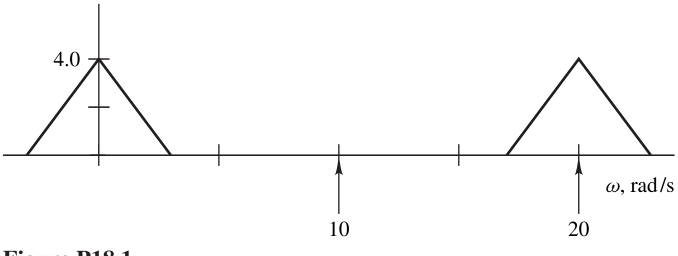


Figure P18.1

18.2. Repeat Prob. 18.1 if the spectrum of $x(nT)$ is given by

$$X(e^{j\omega T}) = \Re e\, X(e^{j\omega T}) + j\, \Im m\, X(e^{j\omega T})$$

where

$$\Re e\, X(e^{j\omega T}) = \begin{cases} 1 - |\omega| & \text{for } -1 < \omega < 1 \\ 0 & \text{for } 1 \le |\omega| \le 10 \end{cases}$$

and

$$\Im m\, X(e^{j\omega T}) = \begin{cases} -\omega & \text{for } -1 < \omega < 1 \\ \ \ 0 & \text{for } 1 \le |\omega| \le 10 \end{cases}$$

The sampling frequency is the same as in Prob. 18.1.

18.3. The spectrum of signal $x(nT)$ in the downsampler of Fig. 18.2*a* is given by

$$X(e^{j\omega T}) = e^{-|\omega|} \qquad \text{for } 0 \le |\omega| < 12$$

and $\omega_s = 24$ rad/s. Find the maximum value of M that will limit the aliasing error to a value less than 1 percent relative to the spectrum of the signal at $\omega = 1$ rad/s.

18.4. In an application, the sampling frequency needs to be increased by a factor of 10.

(*a*) Design a nonrecursive filter that can be used along with an upsampler to construct an interpolator. Linear interpolation is acceptable.

(*b*) Plot the amplitude response of the filter.

18.5. A signal $x(nT)$ is applied at the input of the configuration depicted in Fig. P18.5*a*. The frequency spectrum of $x_c(t)$, namely, $X_c(j\omega)$, is zero for $|\omega| \ge \omega_c$, as illustrated in Fig. P18.5*b*. The filter shown is a nonrecursive filter of length N with a frequency response

$$H(j\omega) = M(\omega)e^{j\theta(\omega)}$$

where

$$M(\omega) = \begin{cases} 3 & \text{for } |\omega| < \omega_c \\ 0 & \text{for } \ \omega_c \le |\omega| \le \omega_s'/2 \end{cases}$$

and

$$\theta(\omega) = (N-1)\omega T'/2$$

(*a*) Sketch the frequency spectrums at points A, B, C, and D.

(*b*) Write expressions for the signals and their frequency spectrums at points A, B, C, and D.

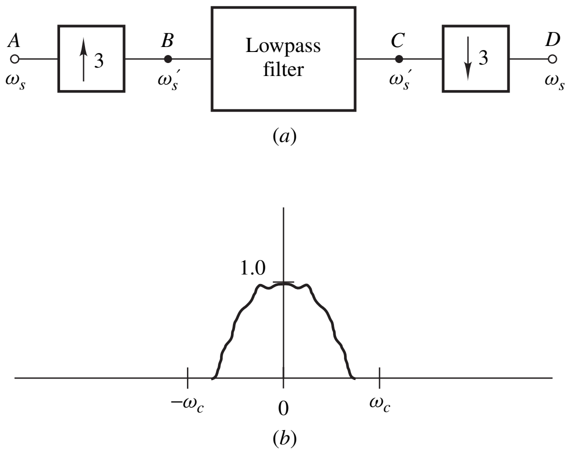


Figure P18.5

18.6. Demonstrate the validity of Eq. (18.29).

18.7. The signal $x(nT)$ in a 4-band QMF bank has the triangular frequency spectrum shown in Fig. 18.10*b*.

(*a*) Sketch the frequency spectrums at the various nodes of the analysis section.

(*b*) Repeat part (*a*) for the synthesis section.

18.8. Time-division to frequency-division multiplex translation can be carried out by using the scheme depicted in Fig. P18.8. Signals $x_{ck}(t)$ for $k = 0, 1, \ldots, K-1$ are bandlimited such that $X_{ck}(j\omega) = 0$ for $|\omega| \geq \omega_m$. The lowpass filters shown are identical and each has a cutoff frequency $\omega_c = \omega_m$. On the other hand, the highpass filters have distinct cutoff frequencies $\omega_0, \omega_1, \ldots, \omega_{K-1}$. For correct operation, $\omega_s \geq 2\omega_m$, $\omega_s' > 2(\omega_{LO} + K\omega_m)$, and $\omega_k \geq \omega_{k-1} + \omega_m$ for $k = 1, 2, \ldots, K-1$.

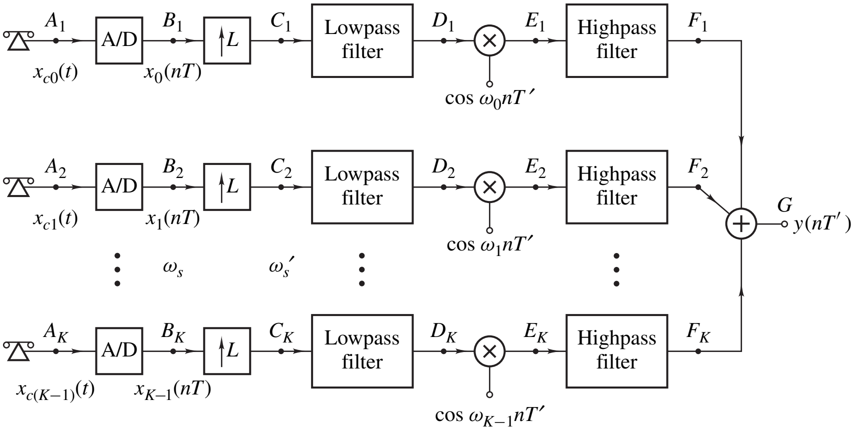


Figure P18.8

(*a*) Sketch the frequency spectrums at points $A_{k,} B_k, \ldots, F_k$, and G for the case where $K = 3$.

(*b*) Explain the role of the lowpass and highpass filters.

18.9. Find the maximum number of channels in the scheme of Fig. P18.8 if $\omega_m = 4$ kHz, $\omega_{LO} = 60$ kHz, $\omega_s = 8$ kHz, and $\omega_s' = 216$ kHz.

18.10. Frequency-division to time-division multiplex translation can be carried out by using the scheme depicted in Fig. P18.10 where the bandpass filters have passbands $\omega_k \le \omega \le \omega_k + \omega_m$ for $k = 1, 2, \ldots, K-1$ and each of the lowpass filters has a cutoff frequency $\omega_c \ge \omega_m$. Sketch the frequency spectrums of the signals at nodes A, B_k, C_k, D_k, and E_k for the case where $K = 3$.

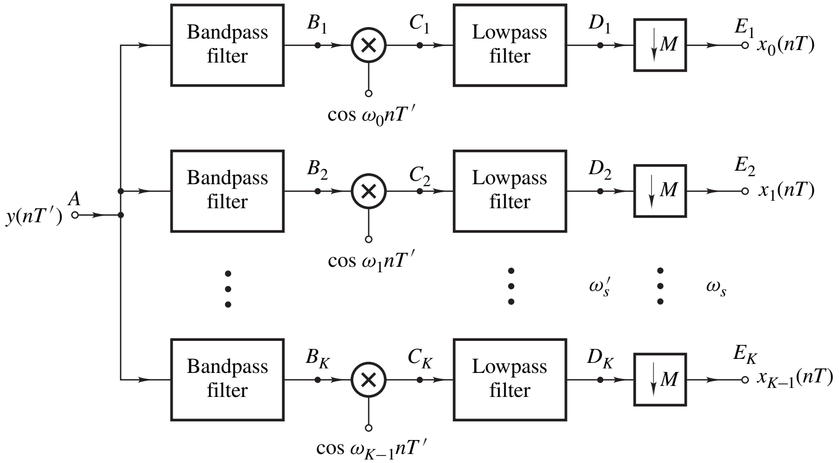


Figure P18.10

18.11. Chapter 9 describes the Fourier series method for the design of nonrecursive filters for the case where the filter length N is odd. Derive the impulse response for a lowpass filter with cutoff frequency ω_c for the case where N is even.

18.12. (*a*) Using the formula for the impulse response obtained in Prob. 18.11 along with the von Hann window design a halfband lowpass filter. Assume that $N = 32$ and $\omega_s = 16$ rad/s.

(*b*) Design a corresponding halfband highpass filter.

(*c*) The filters in parts (*a*) and (*b*) are used in a QMF bank. Plot the amplitude response of the QMF bank.

18.13. Redesign the filters in Prob. 18.12 using the Kaiser window with $\alpha = 3.0$. Compare the results with those obtained using the von Hann window.

18.14. Let the numerator polynomial of transfer function $H_A(s)$ in Example 18.2 be $N(s)$. Demonstrate that $N(s)$ and polynomials $d_A(s)$ and $d_B(s)$ in Eqs. (18.25) and (18.26) satisfy the relation

$$\tfrac{1}{2}[d_A(s)d_B(-s) + d_A(-s)d_B(s)] = N(s)$$

(see Sec. 17.5).

18.15. (*a*) Redesign the filter in Example 18.2 using a fifth-order Butterworth approximation.

(*b*) Demonstrate that the formula in Prob. 18.14 applies.

(*c*) Two copies of the filter obtained will be used as the analysis and synthesis banks in a transmission system. Plot the overall group delay characteristic of the system.

18.16. (*a*) Redesign the filter of Example 18.2 using a fifth-order Chebyshev approximation.

(*b*) Determine the amplitude response of the lowpass filter by applying the bilinear transformation to the analog transfer function.

(*c*) Determine the amplitude response of the lowpass filter by analyzing the lattice structure obtained (see Sec. 17.8).

18.17. The filter obtained in Prob. 18.16 is to be used both for the analysis and synthesis banks in the scheme of Fig. 18.9. Find the overall phase response of the system.

18.18. (*a*) Design a Hilbert transformer of length $N = 31$ using the Kaiser window with $\alpha = 4.0$, assuming a sampling frequency of 100 rad/s.

(*b*) Repeat part (*a*) with $N = 32$.

(*c*) Compare the results obtained in the two cases.

18.19. Formulate the error function and obtain the necessary derivatives to enable the design of Hilbert transformers using the Remez exchange algorithm (say Algorithm 4 in Chap. 15).

18.20. Demonstrate the validity of Eq. (18.44).

18.21. The eigenvalues of an $N \times N$ matrix $\mathbf{R}_n$ are $\lambda_1, \lambda_2, \ldots, \lambda_N$. Show that the eigenvalues of $\tilde{\mathbf{R}}_n = \mathbf{I}_n - 2\alpha\mathbf{R}_n$ are given by $\tilde{\lambda}_i = 1 - 2\alpha\lambda_i$.

18.22. The input and desired signals in an adaptive filter are given by

$$x(n) = e^{-j\omega n/N}$$

and

$$d(n) = e^{-j(\omega n/N+\phi)} + n_1(n)$$

respectively, where $n_1(n)$ is a white noise source with variance σ_n^2.

(*a*) Calculate $\mathbf{p}_n$ and $\mathbf{R}_n$ for the case where a nonrecursive filter of length $N = 2$ is employed.

(*b*) Obtain the Wiener solution as well as the minimum MSE at the output.

18.23. Show that the inequality in Eq. (18.62) is a sufficient condition for the stability of the LMS algorithm.

18.24. Three variations of the standard LMS updating formula given in Eq. (18.61) are

$$\mathbf{a}_{n+1} = \mathbf{a}_n + 2\alpha\,\mathrm{sgn}[e(n)]\mathbf{x}_n$$
$$\mathbf{a}_{n+1} = \mathbf{a}_n + 2\alpha e(n)\,\mathrm{sgn}(\mathbf{x}_n)$$

and

$$\mathbf{a}_{n+1} = \mathbf{a}_n + 2\alpha\,\mathrm{sgn}[e(n)]\,\mathrm{sgn}(\mathbf{x}_n)$$

where

$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ -1 & \text{for } x < 0 \end{cases}$$

and

$$\mathrm{sgn}(\mathbf{x}) = [\mathrm{sgn}(x_1)\ \mathrm{sgn}(x_2)\ \ldots\ \mathrm{sgn}(x_N)]^T$$

Constant 2α is usually chosen to be a power of two for the sake of computational efficiency. Discuss the effects of these simplifications on the gradient direction, convergence, and the residual error.

18.25. Apply the LMS algorithm and each of the variations described in Prob. 18.24 for the identification of a system characterized by

$$H(z) = \sum_{i=0}^{4} z^{-i}$$

using the initial coefficient vector $\mathbf{a}_0 = [0\ 0\ 0\ 0\ 0]^T$. Discuss the results obtained.

18.26. If matrix $\mathbf{R}_n$ is approximated by a diagonal matrix whose diagonal elements are all equal to

$$\|\mathbf{x}_n\|^2 = \mathbf{x}_n^T\mathbf{x}_n$$

the so-called *normalized-LMS* algorithm is obtained.

(*a*) Show that in this algorithm, the updating formula assumes the form

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \frac{2\alpha\, e(n)\mathbf{x}_n}{\gamma + \mathbf{x}_n^T \mathbf{x}_n}$$

where γ is a small constant.

(*b*) Explain the purpose of constant γ.

18.27. A transmission channel can be represented by the transfer function

$$H(z) = \sum_{i=0}^{8} (i-4) z^{-i}$$

Identify the channel by using first the LMS algorithm and then the normalized-LMS algorithm, and compare the results obtained.

18.28. In real-time applications an estimate for $\mathbf{R}_n$, designated by $\tilde{\mathbf{R}}_n$, can be generated as

$$\tilde{\mathbf{R}}_n = (1-\mu)\tilde{\mathbf{R}}_{n-1} + \mu \mathbf{x}_n \mathbf{x}_n^T$$

where μ is a constant. On the other hand, if $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are matrices of appropriate dimensions, then they are interrelated in terms of the so-called *matrix inversion lemma* which states that

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}$$

Using the above formulas, derive a recursive formula for $\tilde{\mathbf{R}}_n^{-1}$.

18.29. Algorithms using the gradient estimate given in Eq. (18.60) along with some estimate for $\tilde{\mathbf{R}}_n^{-1}$ are referred to as *LMS-Newton* adaptation algorithms.

(*a*) Construct such an algorithm using the estimate for $\tilde{\mathbf{R}}_n^{-1}$ obtained in Prob. 18.28.

(*b*) Apply this algorithm to the system identification problem described in Prob. 18.25.

18.30. A 2-D digital filter has the transfer function

$$H(z_1, z_2) = \frac{N(z_1, z_2)}{D(z_1, z_2)} = \frac{2z_1 z_2}{z_1 z_2 - 0.5z_1 - 0.5z_2 + 0.25}$$

Find its impulse response. The 2-D impulse function is defined as

$$\delta(n_1, n_2) = \begin{cases} 1 & \text{for } n_1 = n_2 = 0 \\ 0 & \text{otherwise} \end{cases}$$

18.31. Repeat Prob. 18.30 if the transfer function is given by

$$H(z_1, z_2) = \frac{N(z_1, z_2)}{D(z_1, z_2)} = \frac{z_1 z_2}{2z_1 z_2 - 1}$$

18.32. Plot the amplitude and phase response of the filter described in Prob. 18.30.

18.33. Check the stability of the filters described in Probs. 18.30 and 18.31.

18.34. A 2-D digital filter is characterized by the transfer function

$$H(z_1, z_2) = \frac{N(z_1, z_2)}{D(z_1, z_2)}$$

where

$$N(z_1, z_2) = 64(z_1 - 1)^2(z_2 - 1)^2$$

and

$$\begin{aligned} D(z_1, z_2) = {} & 64z_1^2z_2^2 - 32z_1z_2^2 + 48z_1^2z_2 + 8z_2^2 + 8z_1^2 \\ & -24z_1z_2 - 4z_1 + 6z_2 + 1 \end{aligned}$$

Check its stability.

18.35. A 2-D lowpass digital filter comprises two cascaded 1-D lowpass filters with passband edges ω_{pi} rad/s, stopband edges ω_{ai} rad/s, passband ripples A_{pi} dB, and minimum stopband attenuations A_{ai} rad/s for $i = 1$ and 2. Find the passband and stopband edges, passband ripple, and minimum stopband attenuation of the 2-D filter.

18.36. Using the formulas obtained in Prob. 18.35 design a 2-D lowpass filter satisfying the following specifications:

$$\omega_{p1} = 2.0 \text{ rad/s} \qquad \omega_{p2} = 3.0 \text{ rad/s}$$
$$\omega_{a1} = 2.4 \text{ rad/s} \qquad \omega_{a2} = 3.6 \text{ rad/s} \qquad \omega_{s1} = \omega_{s2} = 10 \text{ rad/s}$$
$$A_p = 1.0 \text{ dB} \qquad A_a \geq 40.0 \text{ dB}$$

APPENDIX A

COMPLEX ANALYSIS

A.1 INTRODUCTION

Digital signal processing (DSP) relies heavily on transform theory which, in turn, necessitates a fairly good understanding of complex analysis. In many universities, a course is available on this branch of mathematics, which is usually a prerequisite for courses on system theory, linear circuits, and DSP. Often no such course is offered and the instructor of DSP is obliged to deal with the relevant parts of complex analysis on the fly along with the standard DSP material.

This appendix deals with the fundamentals of complex analysis and the basic objective is to enable an instructor to teach DSP at a university where a suitable prerequisite on complex analysis is not available. It can also serve as a quick reference to the basic principles. The topics to be discussed are selected on the basis of their relevance to DSP and the exposition is intended for the practitioner rather than the mathematician, i.e., principles, definitions, and theorems are presented with minimal rigor or proof. For a more mathematical treatment of the subject, the reader is referred to one of the standard textbooks on complex analysis [1–3].

The subjects considered include complex arithmetic, complex variables, differentiability, and analyticity of functions of a complex variable and their representation in terms of power series like the Laurent series. The appendix also includes brief biographical notes on some of the great mathematicians who developed the subject in the first place. Some of this material originates from the Biographies Index of the The MacTutor History of Mathematics Archive, School of Mathematics and Statistics, University of St. Andrews, Scotland [4].

A.2 COMPLEX NUMBERS

The first reference to what we know today as complex numbers occurred during the fifteenth century. According to the record, the first person to carry out a calculation involving complex numbers was an Italian by the name of Cardano who was a qualified medical doctor turned mathematician by circumstances.[1] Cardano's quote in Fn. 1 makes it quite clear that he did not grasp the enormity of what he had stumbled upon but another Italian by the name of Bombelli was able to put everything into perspective.[2]

The term *complex number* was introduced by Gauss who also paved the way for the development of complex numbers as an organized branch of mathematics.[3] The correct meaning of the term is of course *composite number*, not *complicated number* as perceived by students more or less everywhere.

The roots of a quadratic equation

$$az^2 + bz + c = 0$$

are given by

$$z = -\frac{b}{2a} \pm \frac{\sqrt{b^2 - 4ac}}{2a} \tag{A.1}$$

and if $b^2 < 4ac$, we can write

$$\begin{aligned} z &= -\frac{b}{2a} \pm \frac{\sqrt{b^2 - 4ac}}{2a} \\ &= -\frac{b}{2a} \pm \sqrt{-1} \cdot \frac{\sqrt{4ac - b^2}}{2a} \\ &= x + jy \end{aligned}$$

where $x = -b/2a$, $y = \sqrt{(4ac - b^2)}/(2a)$, and $j = \sqrt{-1}$. The components of a complex number, x and y, are called the *real and imaginary parts* and can be represented by the notation

$$x = \Re e\, z \qquad \text{and} \qquad y = \Im m\, z$$

[1] *Girolano Cardano (Cardan in Latin) (1501–1576) is known for his work on the solution of the cubic and quartic equations. In his mathematical treatise Ars Magna, which also deals with his methods for the solution of cubic and quartic equations, Cardano states "Dismissing mental tortures, and multiplying 5 +* $\sqrt{-15}$ *by* $5 - \sqrt{-15}$, *we obtain* $25 - (-15)$. *Therefore the product is 40, ... , and thus far does arithmetical subtlety go, of which this, the extreme, is, as I have said, so subtle that it is useless."* [4].

[2] *Rafael Bombelli (1525–1572) was the first person to work out the rules of complex arithmetic. He also published an algebra book that dealt with the state of the art on the subject and included his own contributions to complex arithmetic. The historical record shows that Bombelli had studied Cardano's work and, no doubt, he was influenced quite substantially by it.*

[3] *Carl Friedrich Gauss (1777–1855) made many contributions to mathematics in the areas of differential equations, complex analysis, numerical analysis, and number theory. He also made important contributions to the theory of magnetism and, apparently, Gauss and Weber built a primitive telegraph device that could send messages over a distance of 5000 ft.*

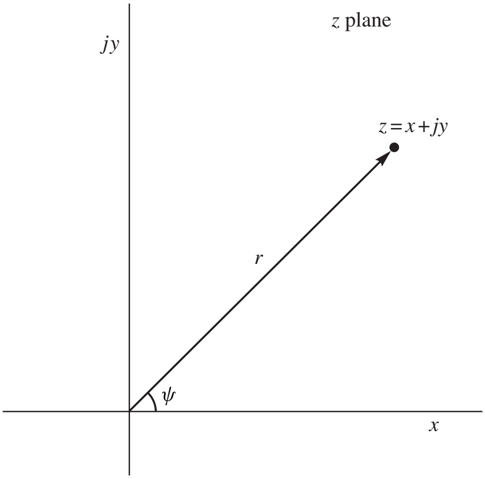


Figure A.1 Complex z plane (Argand diagram).

and j is called the *imaginary unit* [3].[4] If coefficients a, b, and c are variables, then z in Eq. (A.1) becomes a complex variable, in general, which can assume real values.

A complex number is deemed to be equal to zero if and only if its real and imaginary parts are both zero and two complex numbers z_1 and z_2 are deemed to be equal to one another if and only if[5] the real and imaginary parts of z_1 are equal to the real and imaginary parts of z_2, respectively, i.e.,

$$z_1 = z_2 \qquad \text{iff} \qquad x_1 = x_2 \qquad \text{and} \qquad y_1 = y_2$$

A complex number can be depicted graphically in an $\{x,\ y\}$ rectangular coordinate system such as that in Fig. A.1. A coordinate system of this type is known as a complex plane or Argand diagram.[6] The representation of a complex number in terms of its real and imaginary parts is known as the *Cartesian representation.*[7]

From Fig. A.1, we note that

$$x = r\cos\psi \qquad \text{and} \qquad y = r\sin\psi$$

where

$$r = |z| = \sqrt{x^2 + y^2} \qquad \text{and} \qquad \psi = \arg z = \tan^{-1}\frac{y}{x} \tag{A.2}$$

[4] *Mathematicians tend to use the symbol i for the imaginary unit.*

[5] *"If and only if" is often denoted as iff in mathematical language.*

[6] *Jean-Robert Argand (1768–1822) was an accountant and bookkeeper by profession but delved into mathematics in his spare time. He made other important contributions to mathematics in addition to his geometrical representations of complex numbers. For example, on the fundamental theorem of algebra that states that an nth-order polynomial has n roots and on combinations whereby r distinct objects are taken at a time from a set of s objects.*

[7] *After René Descartes (1596–1650), the inventor of analytic geometry.*

are the magnitude (radius) and angle (or argument) of z, respectively. Therefore,

$$z = x + jy = r(\cos\psi + j\sin\psi) \tag{A.3}$$

Evidently, the radius and angle completely specify complex number z and the set $\{r,\ \psi\}$ is said to be its *polar representation*.

A.2.1 Complex Arithmetic

Complex numbers and variables can be added, subtracted, or multiplied according to the usual laws of algebra, such as, the commutative, associative, and distributive laws (see Sec. 4.4.1). Therefore, complex arithmetic need not present problems.

A complex arithmetic operation that has no counterpart in real arithmetic is complex conjugation. The complex conjugate (or simply conjugate) of $z = x + jy$ is defined as

$$z^* = (x + jy)^* = x - jy$$

Addition or subtraction of two complex numbers $z_1 = x_1 + jy_1$ and $z_2 = x_2 + jy_2$ is carried out by adding or subtracting their respective real and imaginary parts, i.e.,

$$z_1 + z_2 = x_1 + jy_1 + x_2 + jy_2 = (x_1 + x_2) + j(y_1 + y_2) \tag{A.4a}$$

and

$$z_1 - z_2 = (x_1 + jy_1) - (x_2 + jy_2) = (x_1 - x_2) + j(y_1 - y_2) \tag{A.4b}$$

Multiplication is carried out by multiplying the two complex numbers term by term treating j just like a real number. Powers of j are simplified by noting that $j^2 = -1$, $j^3 = -j$, $j^4 = 1$, $j^5 = j$, and so on. Thus

$$\begin{aligned} z_1 z_2 &= (x_1 + jy_1)(x_2 + jy_2) = x_1x_2 + j(x_2y_1 + x_1y_2) + j^2y_1y_2 \\ &= (x_1x_2 - y_1y_2) + j(x_2y_1 + x_1y_2) \end{aligned} \tag{A.4c}$$

Division can be carried out by multiplying the dividend and divisor by the conjugate of the divisor, i.e.,

$$\begin{aligned} \frac{z_1}{z_2} &= \frac{x_1 + jy_1}{x_2 + jy_2} = \frac{(x_1 + jy_1)(x_2 - jy_2)}{(x_2 + jy_2)(x_2 - jy_2)} \\ &= \frac{x_1x_2 + y_1y_2 + j(x_2y_1 - x_1y_2)}{x_2^2 + y_2^2} \end{aligned} \tag{A.4d}$$

A.2.2 De Moivre’s Theorem

If $z_1 = x_1 + jy_1 = r_1(\cos\psi_1 + j\sin\psi_1)$ and $z_2 = x_2 + jy_2 = r_2(\cos\psi_2 + j\sin\psi_2)$, it can be easily shown that

$$z_1 z_2 = r_1 r_2[\cos(\psi_1 + \psi_2) + j\sin(\psi_1 + \psi_2)] \tag{A.5a}$$

and

$$\frac{z_1}{z_2} = \frac{r_1}{r_2}[\cos(\psi_1 - \psi_2) + j\sin(\psi_1 - \psi_2)] \tag{A.5b}$$

The formula in Eq. (A.5a) can be readily extended to a product of n complex numbers as

$$z_1 z_2 \cdots z_n = r_1 r_2 \cdots r_n[\cos(\psi_1 + \psi_2 + \cdots \psi_n) + j\sin(\psi_1 + \psi_2 + \cdots + \psi_n)]$$

and if $z_1 = z_2 = \cdots = z = r(\cos\psi + j\sin\psi)$, we get

$$z^n = [r(\cos\psi + j\sin\psi)]^n = r^n(\cos n\psi + j\sin n\psi) \tag{A.6}$$

This relation is known as *De Moivre's theorem.*

If $w^n = z$, then $w = z^{1/n}$ is said to be an nth *root* of z. Using De Moivre's relation in Eq. (A.6), it can be shown that a complex number has n nth roots, given by

$$\begin{aligned} w_k = z^{1/n} &= r^{1/n}(\cos\psi + j\sin\psi)^{1/n} \\ &= r^{1/n}\left(\cos\frac{\psi + 2k\pi}{n} + j\sin\frac{\psi + 2k\pi}{n}\right) \end{aligned} \tag{A.7}$$

for $k = 0, 1, \ldots, n-1$.

A.2.3 Euler's Formula

An alternative representation for a complex number z, referred to in this textbook as the *exponential form*, can be deduced from the following well-known series

$$\sin\psi = \psi - \frac{\psi^3}{3!} + \frac{\psi^5}{5!} + \cdots \tag{A.8}$$

$$\cos\psi = 1 - \frac{\psi^2}{2!} + \frac{\psi^4}{4!} - \cdots \tag{A.9}$$

$$\tan\psi = \psi + \frac{\psi^3}{3} + \frac{2\psi^5}{15} + \cdots \tag{A.10}$$

$$e^{\psi} = 1 + \psi + \frac{\psi^2}{2!} + \frac{\psi^3}{3!} + \frac{\psi^4}{4!} + \frac{\psi^5}{5!}\cdots \tag{A.11a}$$

If we replace ψ by $j\psi$ in Eq. (A.10), we get

$$\begin{aligned} e^{j\psi} &= 1 + j\psi + \frac{j^2\psi^2}{2!} + \frac{j^3\psi^3}{3!} + \frac{j^4\psi^4}{4!} + \frac{j^5\psi^5}{5!}\cdots \\ &= 1 + j\psi - \frac{\psi^2}{2!} - \frac{j\psi^3}{3!} + \frac{\psi^4}{4!} + \frac{j\psi^5}{5!}\cdots \\ &= \left(1 - \frac{\psi^2}{2!} + \frac{\psi^4}{4!}\cdots\right) + j\left(\psi - \frac{\psi^3}{3!} + \frac{\psi^5}{5!}\cdots\right) \end{aligned} \tag{A.11b}$$

and from Eqs. (A.8), (A.9), and (A.11b), we obtain the relation

$$e^{j\psi} = \cos\psi + j\sin\psi \tag{A.12}$$

which is known as *Euler's formula*.

A.2.4 Exponential Form

An arbitrary complex number z with polar representation $\{r, \psi\}$ can be expressed as

$$z = r\cos\psi + jr\sin\psi = r(\cos\psi + j\sin\psi) \tag{A.13a}$$

and from Euler's formula in Eq. (A.12), z can be expressed in terms of the *exponential form*

$$z = re^{j\psi} \tag{A.13b}$$

where $r = |z|$ and $\psi = \arg z$.

Complex numbers like their real counterparts obey the law of exponents and thus the product of two complex numbers $z_1 = r_1e^{j\psi_1}$ and $z_2 = r_2e^{j\psi_2}$ can be obtained as

$$z_1z_2 = r_1e^{j\psi_1}r_2e^{j\psi_2} = r_1r_2e^{j(\psi_1+\psi_2)} \tag{A.14a}$$

Hence

$$|z_1z_2| = r_1r_2 \qquad \text{and} \qquad \arg(z_1z_2) = \psi_1 + \psi_2 \tag{A.14b}$$

Division is just as easy. We can write

$$\frac{z_1}{z_2} = \frac{r_1e^{j\psi_1}}{r_2e^{j\psi_2}} = \frac{r_1}{r_2}e^{j(\psi_1-\psi_2)} \tag{A.15a}$$

and hence

$$\left|\frac{z_1}{z_2}\right| = \frac{r_1}{r_2} \qquad \text{and} \qquad \arg\frac{z_1}{z_2} = \psi_1 - \psi_2 \tag{A.15b}$$

In general, an arbitrary ratio of products can be expressed as

$$\frac{\prod_{i=1}^{M} z_{mi}}{\prod_{i=1}^{N} z_{ni}} = re^{j\psi} \tag{A.16a}$$

where

$$r = \frac{\prod_{i=1}^{M} |z_{mi}|}{\prod_{i=1}^{N} |z_{ni}|} \tag{A.16b}$$

and

$$\psi = \sum_{i=1}^{M} \arg z_{mi} - \sum_{i=1}^{N} \arg z_{ni} \tag{A.16c}$$

Similarly, the nth power of z can be expressed as

$$z^n = (re^{j\psi})^n = r^n e^{jn\psi} = r^n(\cos n\psi + j \sin n\psi) \tag{A.17}$$

which is an alternative form of De Moivre's relation. Note that Euler's formula in Eq. (A.12) is actually De Moivre's relation for the special case where $r = 1$ and $n = 1$.

A.2.5 Vector Representation

Complex numbers may be deemed to be two-dimensional vectors. Hence vector methodology can be used. Thus two complex numbers z_1 and z_2 can be added by using the parallelogram law illustrated in Fig. A.2*a*. Extending this principle, an arbitrary number of complex numbers can be added by aligning them end to end. For example, three complex numbers $z_1 = -1 + j1$, $z_2 = 2 + j2$, and $z_3 = 2 - j1$ can be added by using the construction illustrated in Fig. A.2*b*.

The sum of the magnitudes of N complex numbers is equal to or greater than the magnitude of their sum, i.e.,

$$\sum_{i=1}^{N} |z_i| \geq \left| \sum_{i=1}^{N} z_i \right| \tag{A.18}$$

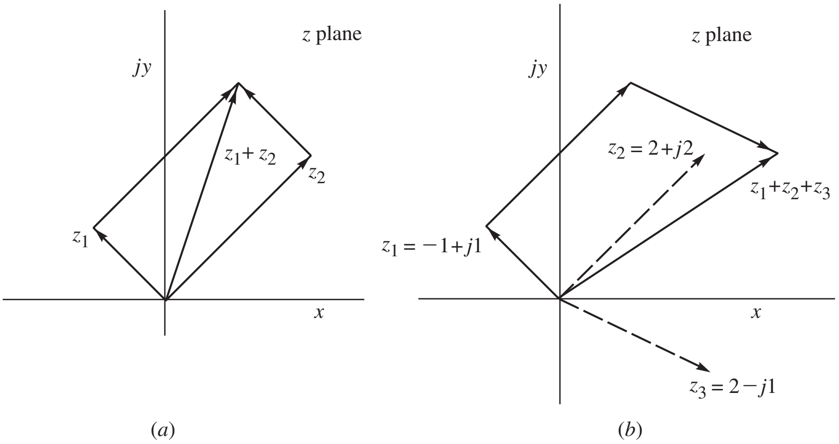


Figure A.2 Vector representation of complex numbers: (*a*) Addition of two complex numbers using the parallelogram law, (*b*) addition of three complex numbers.

where the equal sign applies only if the complex numbers have the same angle. For example, if $z_1 = -1 + j1$, $z_2 = 2 + j2$, and $z_3 = 2 - j1$, we have

$$\sum_{i=1}^{3} |z_i| = |(-1 + j1)| + |(2 + j2)| + |(2 - j1)| = \sqrt{2} + \sqrt{8} + \sqrt{5} = 6.479$$

whereas

$$\left| \sum_{i=1}^{3} c_i \right| = |(-1 + j1) + (2 + j2) + (2 - j1)| = |3 + j2| = \sqrt{13} = 3.606$$

Clearly,

$$\sum_{i=1}^{3} |z_i| > \left| \sum_{i=1}^{3} z_i \right|$$

This simple, yet important, inequality is illustrated in Fig. A.2*b*.

A.2.6 Spherical Representation

It is sometimes convenient to represent the complex z plane in terms of the surface of a sphere of unit radius, as depicted in Fig. A.3 where the line passing through the north and south poles of the sphere passes through the origin of the complex z plane and is perpendicular to it. In this

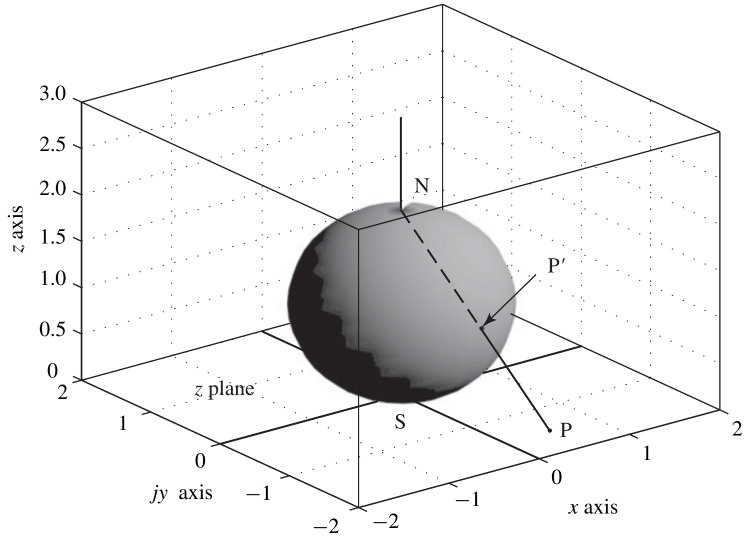


Figure A.3 Riemann sphere.

geometrical construction, which is known as a *Riemann sphere*,[8] given an arbitrary point P in the z plane, a line can be drawn joining point P with the north pole N of the sphere, as depicted in Fig. A.3, and the point of intersection of line PN with the surface of the sphere, namely, P′, bares a one-to-one correspondence with point P. Evidently, each and every point in the complex z plane can be mapped onto a corresponding point on the surface of the sphere. The most significant feature of this stereographic projection is that any point situated at a very large distance from the origin will map in the neighborhood of the north pole and thus a point at infinity will map at the north pole. The Riemann sphere renders the abstract concept of infinity easier to understand.

A.3 FUNCTIONS OF A COMPLEX VARIABLE

A complex variable W may be a function of another complex variable $z = x + jy$. Such a relation can be expressed as

$$W = F(z)$$

and if U and V are the real and imaginary parts of W, we have

$$W = F(z) = U(x, y) + jV(x, y)$$

Functions of a complex variable appear frequently in DSP and several types are available, e.g.,

- polynomials,
- rational algebraic functions,
- inverse algebraic functions,
- exponential and logarithmic functions,
- trigonometric and their inverse functions,
- hyperbolic and their inverse functions, etc.

These functions are generalizations of their real counterparts.

A.3.1 Polynomials

A polynomial in z assumes the form

$$P(z) = a_0 + a_1 z + a_2 z^2 + \cdots + a_N z^N = \sum_{i=0}^{N} a_i z^i$$

where z is a complex variable and coefficients a_i for $0, 1, \ldots, N$ are typically real in DSP although they could be complex in certain applications. Integer N is the degree or order of the polynomial.

[8] *Friedrich Bernhard Riemann (1826–1866) was born in Breselenz, Hanover (now in Germany). He took lessons from Gauss and Dirichlet. He contributed greatly to the theory of complex analysis and built upon the theories of Cauchy. He produced original work on conformal transformations (see Sec. A.9) and introduced topological methods into complex analysis.*

Values of z that yield $P(z) = 0$ are said to be the *roots* of the polynomial and from the so-called *fundamental theorem of algebra*, an Nth-order polynomial has N roots. If the coefficients are real, the roots are either real or occur in complex-conjugate pairs.

A.3.2 Inverse Algebraic Functions

Given a function

$$s = F(z)$$

where s and z are both complex variables, a new function

$$z = G(s)$$

can sometimes be obtained. Such a function is said to be the *inverse* of $F(z)$ and can be expressed as

$$z = G(s) = F^{-1}\{s\}$$

For example, if

$$s = \frac{z+1}{z-1}$$

then

$$z = F^{-1}\{s\} = \frac{s+1}{s-1}$$

A.3.3 Trigonometric Functions and Their Inverses

On replacing ψ in Eqs. (A.8)–(A.10) first by z and then by $-z$, we can readily conclude that the sine and tangent functions are odd functions and the cosine is an even function of z, i.e.,

$$\sin(-z) = -\sin z \qquad \cos(-z) = \cos z \qquad \tan(-z) = -\tan z \tag{A.19}$$

Now on replacing ψ first by z and then by $-z$ in $e^{j\psi}$ in Eq. (A.11b), the basic trigonometric functions of a complex variable z can be obtained as

$$\sin z = \frac{1}{2j}(e^{jz} - e^{-jz}) \tag{A.20a}$$

$$\cos z = \frac{1}{2}(e^{jz} + e^{-jz}) \tag{A.20b}$$

$$\tan z = \frac{\sin z}{\cos z} = \frac{e^{jz} - e^{-jz}}{j(e^{jz} + e^{-jz})} \tag{A.20c}$$

The following identities follow their real counterparts:

$$\sin(z_1 \pm z_2) = \sin z_1 \cos z_2 \pm \cos z_1 \sin z_2 \tag{A.21a}$$

$$\cos(z_1 \pm z_2) = \cos z_1 \cos z_2 \mp \sin z_1 \sin z_2 \tag{A.21b}$$

$$\tan(z_1 \pm z_2) = \frac{\tan z_1 \pm \tan z_2}{1 \mp \tan z_1 \tan z_2} \tag{A.21c}$$

$$\sin^2 z + \cos^2 z = 1 \tag{A.21d}$$

The standard inverse trigonometric functions are given by

$$\sin^{-1} z = \frac{1}{j} \ln\left(jz + \sqrt{1 - z^2}\right) \tag{A.22a}$$

$$\cos^{-1} z = \frac{1}{j} \ln\left(z + \sqrt{z^2 - 1}\right) \tag{A.22b}$$

$$\tan^{-1} z = \frac{1}{2j} \ln\left(\frac{1 + jz}{1 - jz}\right) \tag{A.22c}$$

where $\ln z \equiv \log_e z$ is the natural logarithm of z.

A.3.4 Hyperbolic Functions and Their Inverses

Like their trigonometric counterparts, the hyperbolic sine and tangent are odd functions and the hyperbolic cosine is an even function of z, i.e.,

$$\sinh(-z) = -\sinh z \qquad \cosh(-z) = \cosh z \qquad \tanh(-z) = -\tanh z \tag{A.23}$$

and by analogy with Eqns. (A.20) and (A.21), we have

$$\sinh z = \frac{1}{2}(e^z - e^{-z}) = -j \sin jz \tag{A.24a}$$

$$\cosh z = \frac{1}{2}(e^z + e^{-z}) = \cos jz \tag{A.24b}$$

$$\tanh z = \frac{\sinh z}{\cosh z} = \frac{e^z - e^{-z}}{e^z + e^{-z}} = -j \tan jz \tag{A.24c}$$

and

$$\sinh(z_1 \pm z_2) = \sinh z_1 \cosh z_2 \pm \cosh z_1 \sinh z_2 \tag{A.25a}$$

$$\cosh(z_1 \pm z_2) = \cosh z_1 \cosh z_2 \pm \sinh z_1 \sinh z_2 \tag{A.25b}$$

$$\tanh(z_1 \pm z_2) = \frac{\tanh z_1 \pm \tanh z_2}{1 \pm \tanh z_1 \tanh z_2} \tag{A.25c}$$

$$\cosh^2 z - \sinh^2 z = 1 \tag{A.25d}$$

On the other hand, the inverse hyperbolic functions are given by

$$\sinh^{-1} z = \ln\left(z + \sqrt{z^2+1}\right) \tag{A.26a}$$

$$\cosh^{-1} z = \ln\left(z \pm \sqrt{z^2-1}\right) \tag{A.26b}$$

$$\tanh^{-1} z = \frac{1}{2}\ln\left(\frac{1+z}{1-z}\right) \tag{A.26c}$$

A.3.5 Multi-Valued Functions

In a functional relation of the form

$$w = F(z)$$

z can assume arbitrary complex values in the z plane and for each value of z there is one or more values of w that can be plotted in the w plane. We can say that the relation maps points of the z plane onto points of the w plane.

Consider the functional relation

$$w = z^{1/2} \tag{A.27}$$

and let $z = re^{j\psi}$ be an arbitrary complex number which can be drawn as shown in Fig. A.4*a*. Solving Eq. (A.27) for w, we get

$$w_1 = u_1 + jv_1 = r^{1/2}e^{j\psi/2}$$

and thus point z in Fig. A.4*a* maps onto point w_1 in Fig. A.4*b*. Since angles ψ and $\psi + 2\pi$ are essentially one and the same angle, complex number z can also be written as

$$z = re^{j(\psi+2\pi)}$$

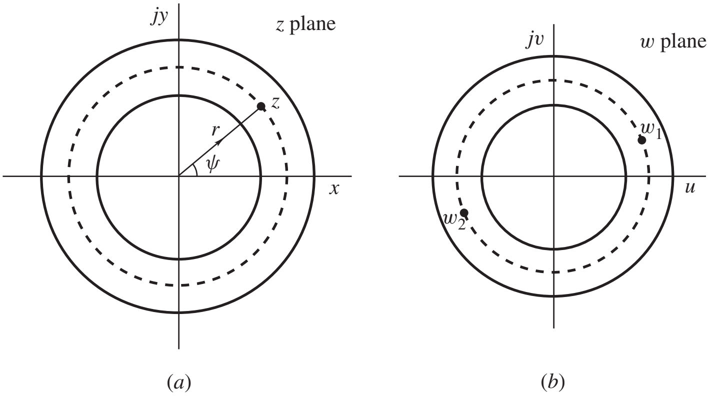


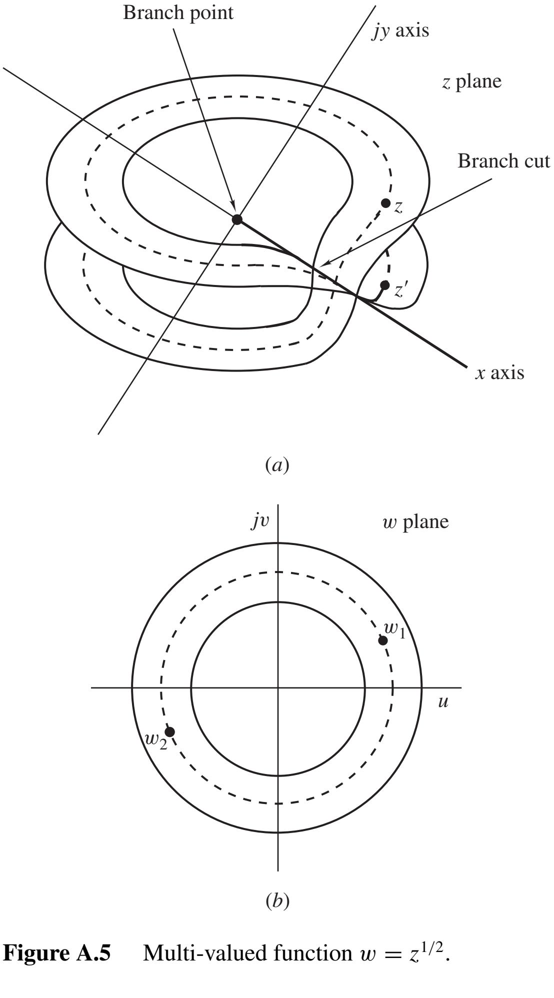
Figure A.4 Multi-valued function $w = z^{1/2}$.

and if we solve Eq. (A.27) again for w, we get

$$w_2 = u_2 + jv_2 = r^{1/2}e^{j\psi/2+\pi}$$

Thus, one and the same point in the z plane maps onto two points in the w plane, which means that for each value of z, function w assumes two distinct values, as depicted in Fig. A.4*b*. Such a function is, in effect, a two-valued function. Generalizing this principle, a function $w = F(z)$ that can assume more than one value in the w plane for each value of z is said to be a *multi-valued* function.

Many of the theorems of complex analysis are applicable only to single-valued functions and it would appear that such theorems would not be applicable to multi-valued functions such as the one in Eq. (A.27). However, through a geometrical interpretation due to Riemann it is possible to treat multi-valued functions as if they were single-valued. In this interpretation, the z plane is deemed to be made up of overlapping sheets and points like $z = re^{j\psi}$ and $z' = re^{j\psi+2\pi}$ are considered to be unique points on different overlapping sheets. To illustrate this idea, let us reconsider the function



(*a*)

(*b*)

Figure A.5 Multi-valued function $w = z^{1/2}$.

in Eq. (A.27). By imagining the z plane to be made of two overlapping sheets such that the bottom sheet is joined to the top sheet along the positive real axis through a four-way seam, as depicted in Fig. A.5*a*, then points $z = re^{j\psi}$ and $z' = re^{j\psi+2\pi}$ can be considered to be distinct thereby causing the mapping to become one-to-one, i.e., each and every point in the z plane corresponds to a unique point in the w plane, as depicted in Fig. A.5*b*. Under these circumstances, the function in Eq. (A.27) can be considered as if it were single-valued and, consequently, any theorems that apply to single-valued functions also apply to the function in Eq. (A.27).

Surfaces such as that in Fig. A.5*a* are said to be Riemann surfaces after their inventor. The four-way seam in Fig. A.5*a* (solid line), which extends from $x = 0$ to infinity, is commonly referred to as a *branch cut* and the origin of the Riemann surface is called a *branch point*.

Another example of a multi-valued function is the n root of z, that is,

$$w = z^{1/n}$$

As in the previous example, the origin of the z plane is a branch point and the positive real axis is a branch cut. The Riemann surface comprises n sheets in this case.

In certain multi-valued functions, the Riemann surface has an infinite number of sheets and such functions are, therefore, said to be infinite-valued. Consider the natural logarithm of z given by

$$w = \ln z \tag{A.28}$$

where

$$z = re^{j\psi}$$

For any integer k, the identity $1 \equiv (\cos 2k\pi + j \sin 2k\pi) \equiv e^{j2k\pi}$ holds and thus we can write

$$z = re^{j\psi} \cdot 1 = re^{j\psi} \cdot e^{j2k\pi} = re^{j(\psi+2k\pi)} \tag{A.29}$$

Hence Eqs. (A.28) and (A.29) give

$$\begin{aligned} w &= \ln z = \ln(re^{j(\psi+2k\pi)}) = \ln r + \ln e^{j(\psi+2k\pi)} \\ &= \ln r + j(\psi + 2k\pi) \end{aligned}$$

We conclude, therefore, the natural logarithm of z is an infinite-valued function.

Just like the other multi-valued functions considered, the natural logarithm of z can also be treated as if it were a single-valued function by representing the z plane in terms of a Riemann surface comprising an infinite number of sheets connected in the form of a spiral as that illustrated in Fig. A.6. The distance between overlapping sheets is, of course, zero, in theory. The range $-\pi < \psi \leq \pi$ is said to be the *principal angle* of z.

A.3.6 Periodic Functions

In DSP, certain functions of a complex variable such as the frequency spectrum of a signal or the frequency response of a discrete-time system are periodic.

A function $H(e^{j\omega T})$ is a periodic function of ω with period ω_s, if

$$H(e^{j(\omega+k\omega_s)T}) = H(e^{j\omega T}) \tag{A.30}$$

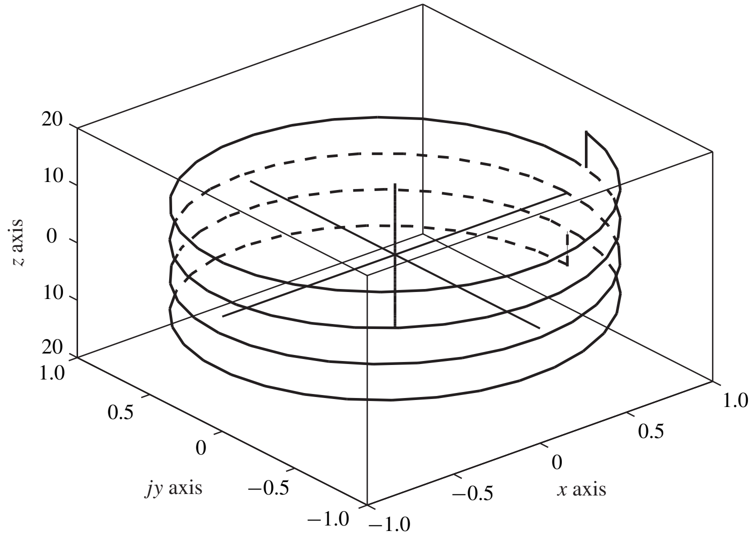


Figure A.6 Riemann surface of a periodic function.

As in the case of multi-valued functions, the nature of periodic functions can be elucidated by representing the z plane in terms of a Riemann surface. For the periodic function of Eq. (A.30), the Riemann surface would assume the form of a spiral ramp such as those found in car parkades, as illustrated in Fig. A.6. The parkade would have an infinite number of floors above as well as below ground level but the height between floors would be zero. For a given ω, points $\ldots e^{j(\omega-\omega_s)}$, $e^{j\omega}$, $e^{j(\omega+\omega_s)} \ldots$ would map at the same coordinates but on distinct sheets one above the other in Fig. A.6.

Note that there is an important difference between the Riemann surface of the periodic function in Eq. (A.30) and that of the multi-valued function in Eq. (A.27). The latter has a branch cut on the positive real axis as depicted in Fig. A.5*a* but the former does not.

A.3.7 Rational Algebraic Functions

A rational algebraic function is a ratio of polynomials of the form

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{A} a_i z^i}{\sum_{i=0}^{B} b_i z^i} \tag{A.31}$$

Rational functions arise frequently both in analog and digital filters in the form of continuous- or discrete-time transfer functions. The frequency response of these filters is determined by evaluating the transfer function with respect to some domain of a complex plane, for example, the frequency response of a digital filter is obtained by letting $z = e^{j\omega T}$ in the discrete-time transfer function $H(z)$, that is, $H(e^{j\omega T})$, whereas for an analog filter, we evaluate the continuous-time transfer function on the $j\omega$ axis. The amplitude and phase responses of a digital filter are simply the magnitude and angle of the frequency response (see Chap. 5) and can be obtained as

$$M(\omega) = |H(e^{j\omega T})| \qquad \text{and} \qquad \theta(\omega) = \arg H(e^{j\omega T}) \tag{A.32a}$$

and as in Eqs. (A.16a)–(A.16c), Eq. (A.31) gives

$$\begin{aligned} M(\omega) &= \left| \frac{N(e^{j\omega T})}{D(e^{j\omega T})} \right| \\ &= \left\{ \frac{[\Re e\, N(e^{j\omega T})]^2 + [\Im m\, N(e^{j\omega T})]^2}{[\Re e\, D(e^{j\omega T})]^2 + [\Im m\, D(e^{j\omega T})]^2} \right\}^{1/2} \end{aligned} \tag{A.32b}$$

and

$$\begin{aligned} \theta(\omega) &= \arg H(e^{j\omega T}) = \arg N(e^{j\omega T}) - \arg D(e^{j\omega T}) \\ &= \tan^{-1} \frac{\Im m\, N(e^{j\omega T})}{\Re e\, N(e^{j\omega T})} - \tan^{-1} \frac{\Im m\, D(e^{j\omega T})}{\Re e\, D(e^{j\omega T})} \end{aligned} \tag{A.32c}$$

The determination of angle $\theta(\omega)$ needs special attention because the inverse tangent is a multivalued function. To start with, one should not divide each imaginary part by the corresponding real part before calculating the inverse tangents, otherwise, an erroneous result may be obtained through loss of information. If, for example, the real and imaginary parts are both negative, then the inverse tangent would give an angle in the third quadrant but if the real part were divided by the imaginary part to start with, a positive number would be obtained, which would give an angle in the first quadrant.[9]

Another issue to be resolved has to do with the fact that computers in general will evaluate $\theta(\omega)$ in the range $-\pi \le \theta(\omega) \le \pi$ although the phase response of a digital filter can be smaller than $-\pi$ or larger than π. This problem can be resolved on the basis of the continuity of the phase response. If the phase angle changes in an anticlockwise direction from $\pi - \vartheta_1$ to $\pi + \vartheta_2$, where $0 < \vartheta_1 < \pi$ and $0 < \vartheta_2 < \pi$, the new phase angle will be evaluated as $-\pi + \vartheta_2$. Thus if the complex value of the frequency response moves from the second to the third quadrant of the z plane, an angle of 2π must be added to the computed phase response in order to get the correct phase angle. On the other hand, if the phase angle changes in a clockwise direction from $-(\pi - \vartheta_1)$ to $-(\pi + \vartheta_2)$, the phase angle would be computed as $\pi - \vartheta_2$, i.e., if the complex value of the frequency response moves from the third to the second quadrant, an angle of 2π must be subtracted from the computed phase angle. In other words, if the complex value of the frequency response crosses the negative real axis in an anticlockwise or clockwise direction an angle of 2π must be added to or subtracted from the computed value, as appropriate.

A.4 BASIC PRINCIPLES OF COMPLEX ANALYSIS

Below some of the key basic principles of complex analysis are highlighted.

A.4.1 Limit

A function $F(z)$ is said to have a *limit* F_0 as z approaches z_0, if (a) $F(z)$ is defined in a neighborhood of z_0 (except perhaps at point z_0) and (b) for every positive real number ϵ there exists a positive real

[9] *In the MATLAB environment, one should use the four-quadrant inverse tangent function* atan2.

number δ such that $|F(z) - F_0| < \epsilon$ for all values of $z \neq z_0$ in the disk $|z - z_0| < \delta$. Limit F_0 can be expressed as

$$F_0 = \lim_{z \to z_0} F(z)$$

A function $F(z)$ is said to be *continuous* at point $z = z_0$ if $F(z_0)$ is defined and is given by

$$F(z_0) = \lim_{z \to z_0} F(z) = F_0$$

Extending this concept somewhat, a *continuous function* is one that is continuous at all the points where it is defined.

A.4.2 Differentiability

The concept of limit leads readily to the definition of *differentiability* of a complex function.

Definition A.1 Differentiability *A function $F(z)$ is said to be differentiable at a point $z = z_0$ if the limit*

$$F'(z_0) = \lim_{\Delta z \to 0} \frac{F(z_0 + \Delta z) - F(z_0)}{\Delta z} \tag{A.33}$$

exists. This limit is called the derivative of $F(z)$ at point $z = z_0$. ▲

If we let $z_0 + \Delta z = z$ in Eq. (A.33), we obtain

$$F'(z_0) = \lim_{z \to z_0} \frac{F(z) - F(z_0)}{z - z_0} \tag{A.34}$$

Hence the derivative exists if and only if the quotient in Eq. (A.34) approaches a unique value independent of the path z may take to approach z_0.

A.4.3 Analyticity

A closely related property to differentiability is the *analyticity* of a complex function.

Definition A.2 Analyticity *A function $F(z)$ is said to be* analytic *at a point $z = z_0$ if it is defined and has a derivative at every point in some neighborhood of z_0. A function $F(z)$ is said to be analytic (also referred to as holomorphic or regular) in a domain D if it is analytic at every point* in D. ▲

Differentiability is a crucial requirement in practice and, consequently, the importance of analyticity cannot be overstated. Indeed, complex analysis is concerned exclusively with analytic functions. Two important equations that pertain to the analyticity of a function are the *Cauchy-Riemann equations* which are given by

$$\frac{\partial U}{\partial x} = \frac{\partial V}{\partial y} \qquad \text{and} \qquad \frac{\partial U}{\partial y} = -\frac{\partial V}{\partial x}$$

These equations are necessary and sufficient for a function to be analytic; that is, if the real and imaginary parts of a function satisfy the Cauchy-Riemann equations in domain D, then the function is analytic in D, and conversely.[10]

A.4.4 Zeros

If a function $F(z)$ is analytic in a domain D and is zero at a point z_0, then the function is said to have a *zero* at z_0. If in addition to $F(z)$, the derivatives

$$\frac{dF(z)}{dz} \quad \cdots \quad \frac{d^{(n-1)}F(z)}{dz^{n-1}}$$

are also zero and

$$\frac{d^n F(z)}{dz^n} \neq 0 \qquad \text{at } z = z_0$$

then the function is said to have a zero of *order n* at point z_0. A function $F(z)$ that has an nth-order zero can be expressed as

$$F(z) = (z - z_0)^n G(z) \tag{A.35}$$

where $G(z_0) \neq 0$. A first-order zero is usually referred to as a *simple zero*.

An analytic function $F(z)$ is said to have an nth-order zero at infinity if $F(1/z)$ has an nth-order zero at $z = 0$.

A.4.5 Singularities

A point z_∞ at which a function $F(z)$ ceases to be analytic is referred to as a *singular point* of the function; alternatively, the function is said to have a *singularity* at $z = z_\infty$.

There are several types of singularities, e.g.,

- poles,
- essential singularities,
- branch points, etc.

(see Ref. [1]) but the most significant ones for DSP are the poles; the other types show up only rarely.

POLES. A function

$$F(z) = \frac{G(z)}{(z - z_\infty)^n}$$

[10] *Augustin-Louis Cauchy (1789–1857) grew up in Paris during the difficult times of the French revolution. In 1810 Cauchy took up his first job to work on port facilities for Napoleon's English invasion fleet. Laplace and Lagrange were family friends, Legendre was an acquaintance, and Ampere was his tutor.*

is said to have an nth-order pole at $z = z_\infty$ if

$$\lim_{z \to z_\infty} (z - z_\infty)^n F(z) = G(z_\infty) \neq 0 \tag{A.36}$$

As in the case of zeros, a pole is said to be *simple* if $n = 1$. A function $F(z)$ has a pole at infinity if $F(1/z)$ has a pole at the origin.

Some functions with poles are as follows:

$$F_A(z) = \frac{z-1}{z+1} \quad \text{has a simple zero at } z = 1$$

$$F_B(z) = \frac{z^2}{z^2 - 2z + 1} \quad \text{has a second-order zero at } z = 0$$

$$F_C(z) = (z^2 + 9)^3 \quad \text{has a third-order zero at } z = \pm j3$$

$$F_D(z) = \frac{1}{z^5} \quad \text{has a fifth-order zero at } z = \infty$$

BRANCH POINTS. Branch points occur in multi-valued functions. As was shown in Sec. A.3.5, $w = z^{1/2}$ is a multi-valued function with a branch point at the origin of the z plane. Since

$$\frac{dw}{dz} = \frac{1}{2z^{1/2}}$$

the derivative of w does not exist at $z = 0$ and, therefore, w has a singularity at the origin.

ESSENTIAL SINGULARITIES. Essential singularities typically arise in functions that can be expressed in terms of infinite series (see Laurent Theorem in Sec. A.6). The following two functions have essential singularities at the origin of the z plane:

$$F_E(z) = e^{1/z} = 1 + \frac{1}{z} + \frac{1}{2!z^2} + \frac{1}{3!z^3} + \cdots$$

$$F_F(z) = \tan\frac{1}{z} = \frac{1}{z} + \frac{1}{3z^3} + \frac{2}{15z^5} + \cdots$$

ISOLATED AND NONISOLATED SINGULARITIES. Singularities can also be classified as *isolated* or *nonisolated.* An *isolated singularity* has a neighborhood that contains no other singular points. If no such neighborhood can be found, the singularity is said to be *nonisolated.*

Poles are always isolated singularities. Essential singularities can be either isolated or nonisolated. The function

$$F_E(z) = e^{1/z} = 1 + \frac{1}{z} + \frac{1}{2!z^2} + \frac{1}{3!z^3} + \cdots$$

has an isolated essential singularity at the origin, since $F(z)$ does not have a singularity at $z = 0 + \epsilon$. On the other hand, function $F_F(z) = \tan 1/z$ has a nonisolated singularity at $z = 0$ since the function is not analytic at an infinite number of points clustered in any neighborhood of $z = 0$. To demonstrate this fact, we note that the tangent function assumes an infinite value if its argument is $\pm\pi/2, \pm 3\pi/2, \pm 5\pi/2, \ldots$. Hence $F_F(z)$ is not analytic at

$$z = \pm\frac{2}{\pi},\ \pm\frac{2}{3\pi},\ \pm\frac{2}{5\pi},\ldots$$

and, therefore, it is not analytic at an infinite number of points in the range $-\epsilon \le \Re e\, z \le \epsilon$ for any positive ϵ.

A.4.6 Zero-Pole Plots

An arbitrary rational function can be expressed as

$$F(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{M} a_i z^{M-i}}{z^N + \sum_{i=1}^{N} b_i z^{N-i}} \tag{A.37a}$$

and by finding the roots of the numerator and denominator polynomials $N(z)$ and $D(z)$, $F(z)$ can be put in the form

$$F(z) = \frac{N(z)}{D(z)} = H_0 \frac{\prod_{i=1}^{Z} (z - z_i)^{m_i}}{\prod_{i=1}^{P} (z - p_i)^{n_i}} \tag{A.37b}$$

where $z_1, z_2, \ldots, z_Z$ and $p_1, p_2, \ldots, p_P$ are the zeros and poles of $F(z)$, respectively, m_i and n_i are the orders of the ith zero and ith pole, respectively, and H_0 is a multiplier constant. The order of the numerator and denominator polynomials in $F(z)$ are given by

$$M = \sum_{i=1}^{Z} m_i \qquad \text{and} \qquad N = \sum_{i=1}^{P} n_i \tag{A.37c}$$

respectively.

A plot of the zeros and poles of a rational function is said to be the *zero-pole plot* of the function. Such a plot along with the corresponding orders of the zeros and poles and the multiplier constant H_0 completely represent the function.

As an example, the function

$$F(z) = \frac{(z^2 - 4)}{(z^2 - 1)(z^2 + 4)} \tag{A.38a}$$

can be expressed as

$$F(z) = \frac{(z-2)(z+2)}{(z-1)(z+1)(z-j2)(z+j2)} \tag{A.38b}$$

and by using small circles and crosses for the zeros and poles, respectively, the zero-pole plot of Fig. A.7 can be constructed for $F(z)$.

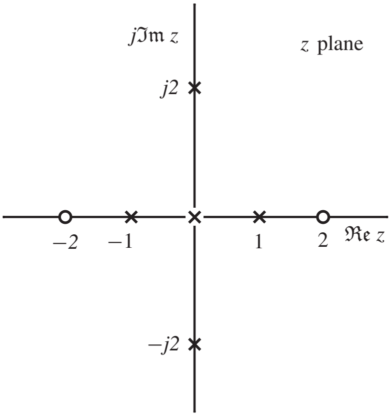


Figure A.7 Zero-pole plot.

Functions of z that are analytic in the entire finite z plane, e.g., a polynomial in z such as $F(z) = 1 + 2z + 3z^2 + z^4$, are called *entire functions*. Functions whose singularities in the finite z plane (i.e., for all $z \neq \infty$) are poles, e.g., rational functions, are called *meromorphic functions*.

A.5 SERIES

Given a sequence of numbers $w_0, w_1, \ldots, w_i \ldots$, which may be real or complex, the infinite series

$$\sum_{i=0}^{\infty} w_i \tag{A.39}$$

can be formed where w_i is said to be the *ith term* of the series.

The sum

$$S_n = \sum_{i=0}^{n} w_i$$

is said to be the *nth partial sum* and

$$R_n = \sum_{i=n+1}^{\infty} w_i$$

is said to be the *nth remainder* of the series.

If

$$S = \lim_{n\to\infty} S_n = \sum_{i=0}^{\infty} w_i$$

and a number N can be found such that

$$|S - S_n| < \epsilon \qquad \text{for all } n > N \tag{A.40}$$

then the series *converges* and S is said to be the *limit* of the sum.

Series arise quite frequently in DSP. Some of their properties can be summarized in terms of a number of theorems, as follows [1].

Theorem A.1 ***If a series $w_0 + w_1 + \cdots + w_N + \cdots$ converges then***

$$\lim_{N\to\infty} w_N \to 0 \quad \blacktriangle \tag{A.41}$$

Theorem A.1 is stating, in effect, that a series diverges if Eq. (A.41) is not satisfied.

A series $w_0 + w_1 + \cdots + w_N + \cdots$ is said to be *absolutely convergent* if the series

$$\sum_{i=0}^{\infty} |w_i|$$

converges.

Theorem A.2 Absolute Convergence ***If a series $w_0 + w_1 + \cdots + w_N + \cdots$ is absolutely convergent, i.e., $|w_0| + |w_1| + \cdots + |w_N| + \cdots$ is finite, then the series converges.*** $\blacktriangle$

Theorem A.2 follows from the fact that sum of the magnitudes of a series of complex numbers is equal to or greater than the magnitude of the sum of the same series of complex numbers (see Eq. (A.18)).

A number of tests that can be used to check the convergence of a series are available such as the ratio and root tests. The ratio test can be stated in terms of the following theorem.

Theorem A.3 Ratio Test ***If $w_i \neq 0$ for $i = 0, 1, 2, \ldots$ and in addition***

$$\left|\frac{w_{n+1}}{w_n}\right| \leq q \qquad \text{for } i > N$$

where q is a fixed number less than 1, then the series in Eq. (A.39) converges. On the other hand, if

$$\left|\frac{w_{n+1}}{w_n}\right| \geq 1 \qquad \text{for } i > N$$

then the series diverges. $\blacktriangle$

If w_i is replaced by $c_i z^i$ in the series of Eq. (A.39) where z is a complex variable, a series of the form

$$\sum_{i=0}^{\infty} c_i z^i \tag{A.42}$$

is obtained, which is usually referred to as a *power series*. The sum of a power series and its nth partial sum are given by

$$S(z) = \sum_{i=0}^{\infty} c_i z^i$$

and

$$S_n(z) = \sum_{i=0}^{n} c_i z^i$$

respectively. If for any given $\epsilon > 0$, a number N can be found such that

$$|S(z) - S_n(z)| < \epsilon \qquad \text{for all } n > N \tag{A.43}$$

where N may depend on ϵ and z, then the power series converges. If a number N can be found that is independent of z, then the power series is said to converge *uniformly*.

A power series may converge for some values of z and diverge for others. Regions of the z plane over which a power series converges or diverges are said to be *regions of convergence* or *divergence*.

If $c_i = 1$, the series in Eq. (A.42) assumes the form

$$\sum_{i=0}^{\infty} z^i \tag{A.44}$$

Such a series is said to be a *geometric series* with a common ratio

$$\frac{w_{N+1}}{w_N} = z \tag{A.45}$$

In order to check the convergence of a geometric series, let

$$S = \sum_{i=0}^{N} z^i \tag{A.46a}$$

be the sum of a finite geometric series. We can write

$$\begin{aligned} S - zS &= (1 - z)S \\ &= (1 + z + z^2 + \cdots + z^N) - z(1 + z + z^2 + \cdots + z^N) \\ &= 1 - z^{(N+1)} \end{aligned}$$

and hence

$$S = \frac{1 - z^{(N+1)}}{1 - z} \tag{A.46b}$$

Now if $|z| < 1$, say $z = re^{j\theta}$ with $r < 1$, we have

$$\begin{aligned} \sum_{i=0}^{\infty} z^i &= \lim_{N\to\infty} S \\ &= \lim_{N\to\infty} \frac{1 - r^{(N+1)}e^{j\theta(N+1)}}{1 - re^{j\theta}} = \frac{1}{1 - z} \end{aligned}$$

since $\lim_{N\to\infty} r^{(N+1)} \to 0$ for $r < 1$. Therefore, the series converges for $|z| < 1$.

For $|z| > 1$, say $z = re^{j\theta}$ with $r > 1$, we can now write

$$\lim_{N\to\infty} S = \lim_{N\to\infty} \sum_{i=0}^{N} r^N e^{j\theta N} \to \infty$$

since $\lim_{N\to\infty} r^{(N+1)} \to \infty$ for $r > 1$.

For $|z| = 1$, say $z = e^{j\theta}$, the Nth term of the series assumes the form

$$w_N = e^{jN\theta} = \cos N\theta + j \sin N\theta$$

and since

$$\lim_{N\to\infty} w_N \neq 0$$

then on the basis of Theorem A.1, we conclude that the series does not converge for $|z| = 1$.

If a power series converges for values of z such that $|z| > \rho$ and diverges for $|z| < \rho$ or the other way around, then the circle $|z| = \rho$ is said to be the *circle of convergence* and ρ is the *radius of convergence*. For the geometric infinite series of Eq. (A.44), $\rho = 1$.

A power series that occurs frequently in DSP is the binomial series which is given by

$$(1+b)^r = 1 + \binom{r}{1} b + \binom{r}{2} b^2 + \cdots + \binom{r}{s} b^s + \cdots \tag{A.47}$$

where

$$\binom{r}{s} = \frac{r(r-1)\cdots(r-s+1)}{s!} \tag{A.48}$$

and $0! = 1$. For a positive integer r, the coefficients of the polynomial obtained are the entries of the $(r+1)$th row in the so-called *Pascal triangle*, which is as follows:

$$\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1 \\
\vdots & & & & & \vdots & & & & & \vdots
\end{array}$$

e.g., for $r = 3$, we have $(1+b)^3 = 1 + 3b + 3b^2 + b^3$.

A.6 LAURENT THEOREM

One of the most important theorems in complex analysis is the *Laurent theorem*[11] which defines the Laurent series and deals with some of its properties. The Laurent series happens to be particularly important for DSP because, as shown in Chap. 3, the z transform is actually a Laurent series.

Theorem A.4 Laurent Theorem ***(a) If $F(z)$ is an analytic and single-valued function[12] on two concentric circles C_1 and C_2 with center a and in the annulus between them, as illustrated in Fig. A.8a, then it can be represented by the Laurent series***

$$F(z) = \sum_{n=-\infty}^{\infty} a_n(z-a)^{-n} \tag{A.49}$$

where

$$a_n = \frac{1}{2\pi j} \oint_{\Gamma} F(z)(z-a)^{n-1}\, dz \tag{A.50}$$

The contour of integration Γ is a closed contour in the counterclockwise sense lying in the annulus between circles C_1 and C_2 and encircling the inner circle.

(b) The Laurent series converges and represents $F(z)$ in the open annulus obtained by continuously increasing the radius of C_2 and decreasing the radius of C_1 until each of C_1 and C_2 reaches a point where $F(z)$ is singular, as depicted in Fig. A.8b.

(c) A function $F(z)$ can have several, possibly many, annuli of convergence about a given point $z = a$, as shown in Fig. A.8c, and for each one a Laurent series can be obtained.

(d) The Laurent series for a given annulus of convergence is unique. ▲

The Laurent series can expressed as a sum of two series as

$$\begin{aligned} F(z) &= \sum_{n=-\infty}^{\infty} a_n(z-a)^{-n} \\ &= \sum_{n=-\infty}^{0} a_n(z-a)^{-n} + \sum_{n=1}^{\infty} a_n(z-a)^{-n} \end{aligned}$$

[11] *Pierre Laurent (1813–1854) was born in Paris. He served in the engineering corps and spent six years directing the operations for the enlargement of the port at Le Havre (north-west of Paris). Laurent submitted his famous work on the* Laurent series *for the Grand Prize of 1842 of the Academie des Sciences but, unfortunately, he missed the official deadline. Cauchy, who was 24 years his senior, reported on the work and argued that the submission should be approved but it was not accepted.*

[12] *The Laurent theorem is also applicable to multi-valued functions provided that the z plane is treated as a Riemann surface (see Sec. A.3.5).*

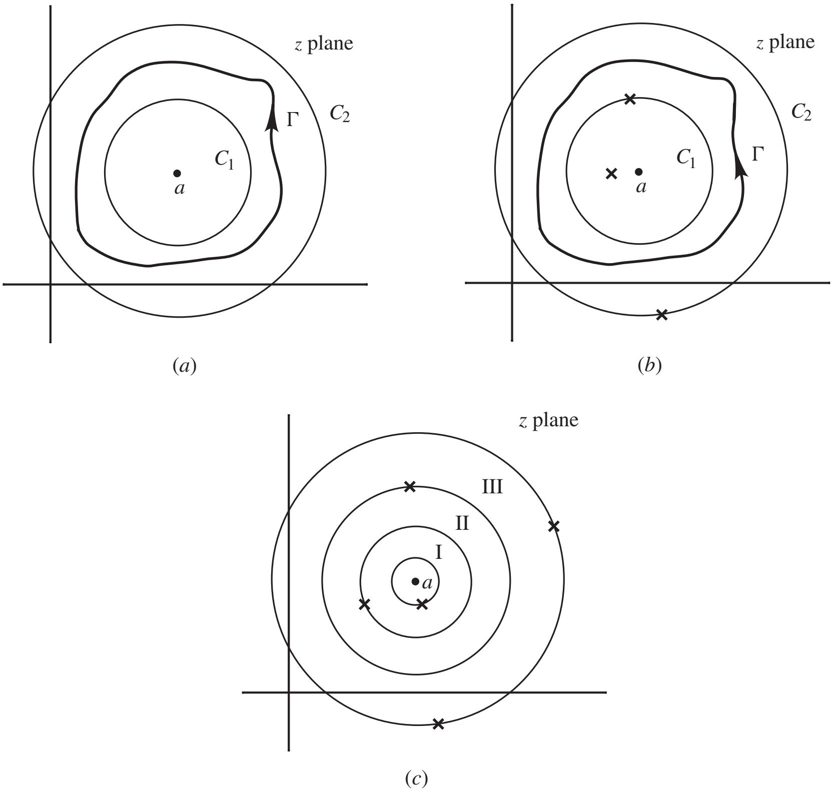


Figure A.8 Laurent theorem.

and if we let $a_n = b_{-n}$ and then replace n by $-n$ in the first part, we obtain

$$\begin{aligned} F(z) &= \sum_{n=-\infty}^{0} b_{-n}(z-a)^{-n} + \sum_{n=1}^{\infty} \frac{c_n}{(z-a)^n} \\ &= \sum_{n=0}^{\infty} b_n(z-a)^n + \sum_{n=1}^{\infty} \frac{c_n}{(z-a)^n} \end{aligned} \tag{A.51}$$

The left- and right-hand parts of the Laurent series are known as the *analytic* and *principal parts*, respectively, and if the Laurent series has a principal part, then function $F(z)$ has a singularity at $z = a$. The type of singularity depends on the number of terms in the principal part as follows:

- If the principal part has just one term, i.e., $c_1 \neq 0$ and $c_n = 0$ for $n > 1$, then $F(z)$ has a simple pole at $z = a$.
- If the principal part has just m terms, i.e., $c_m \neq 0$ and $c_n = 0$ for $n > m$, then $F(z)$ has an mth-order pole at $z = a$.
- If the principal part has an infinite number of terms, then $F(z)$ has an *essential singularity* at $z = a$.

Coefficient c_1, that is, the first coefficient in the principal part, is of crucial importance in complex analysis and for this reason it has a special name. It is called the *residue* of function $F(z)$ at the singular point $z = a$ and it will surface again in the next section in the so-called *residue theorem*.

Some typical Laurent series are as follows:

$$F_A(z) = z^2 + 2z + 1 + \frac{1}{z-3} \quad \text{has a simple pole at } z = 3$$

$$F_B(z) = z^2 + 2z + 1 + \frac{1}{(z-1)} + \frac{2}{(z-1)^2} + \frac{3}{(z-1)^3} \quad \text{has a third-order pole at } z = 1$$

$$F_C(z) = 4z^4 + 3z^3 + 2z^2 + z + 1 \quad \text{is analytic in the finite } z \text{ plane}$$

$$F_D(z) = 1 + \frac{1}{z^7} \quad \text{has a seventh-order pole at } z = 0$$

$$F_E(z) = e^{1/z} = 1 + \frac{1}{z} + \frac{1}{2!z^2} + \frac{1}{3!z^3} + \cdots \quad \text{has an isolated essential singularity at } z = 0$$

$$F_F(z) = \tan\frac{1}{z} = \frac{1}{z} + \frac{1}{3z^3} + \frac{2}{15z^5} + \cdots \quad \text{has an nonisolated essential singularity at } z = 0$$

According to part (c) of the Laurent theorem, a function $F(z)$ can have one and only one Laurent series in a given annulus of convergence. However, a function can have several annuli of convergence and each will have its unique Laurent series. For example, function $F(z)$ in Eq. (A.38b) has three annuli of convergence about point $z = a$, as depicted in Fig. A.9a and b, where the radius of the inner circle in annulus I can be infinitesimally small and that of the radius of the outer circle in annulus III can be infinitely large.

If $c_n = 0$ for all $n \geq 1$, the Laurent series assumes the form

$$F(z) = \sum_{n=0}^{\infty} b_n(z-a)^n \tag{A.52a}$$

and if we let $z - a = h$ or $z = a + h$, we get

$$F(a+h) = \sum_{n=0}^{\infty} b_n h^n \tag{A.52b}$$

Straightforward analysis will show that

$$b_0 = F(a) \qquad \text{and} \qquad b_n = \left.\frac{1}{n!}\frac{d^n F(a+h)}{dh^n}\right|_{h=0} \tag{A.52c}$$

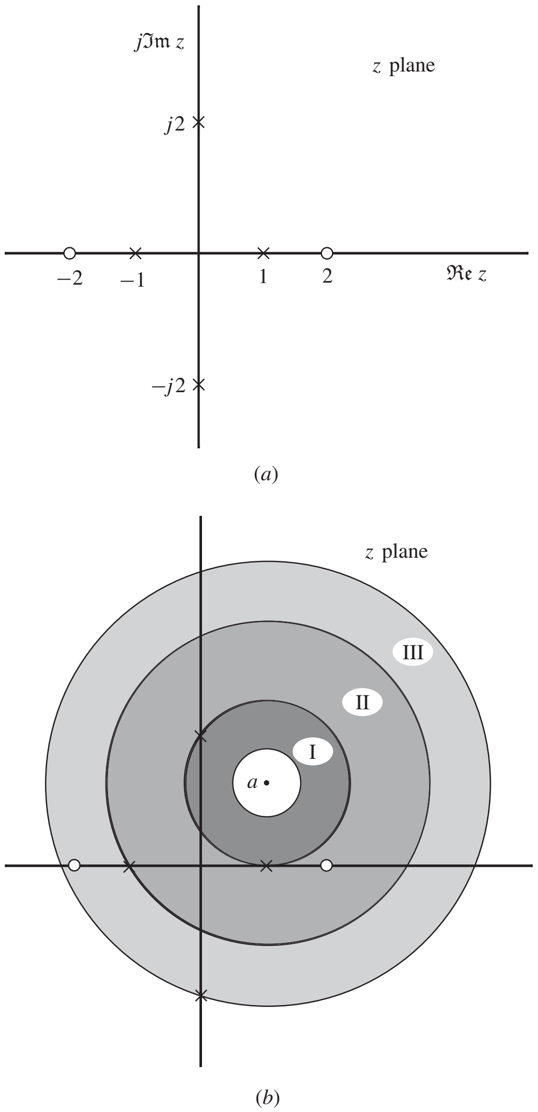


Figure A.9 Annuli of convergence for function $F(z)$ in Eq. (A.38b) for point $z = a$.

and from Eqs. (A.52b) and (A.52c), we get

$$F(a+h) = F(a) + \sum_{n=1}^{\infty} \frac{h^n}{n!} \frac{d^n F(a+h)}{dh^n}\bigg|_{h=0} \tag{A.52d}$$

If z is assumed to be a real variable and $a \equiv x$, Eq. (A.52d) assumes the form of the familiar Taylor series of a function of a real variable, namely,

$$F(x+h) = F(x) + \sum_{n=1}^{\infty} \frac{h^n}{n!} \frac{d^n F(x)}{dh^n} \tag{A.52e}$$

In effect, Eq. (A.52a) is the *Taylor series* for $F(z)$ about point $z = a$. If, in addition, $a = 0$, the Taylor series about the origin of the z plane is obtained, i.e.,

$$F(z) = \sum_{n=0}^{\infty} b_n z^n \tag{A.52f}$$

which is commonly referred to as the *Maclaurin series* of $F(z)$.

A.7 RESIDUE THEOREM

A Laurent series for a function $F(z)$ can be obtained by evaluating coefficients a_n for $-\infty < n < \infty$ using the contour integral in Eq. (A.50). This appears to be a formidable task but for the class of meromorphic functions, the evaluation of the contour integral in Eq. (A.50) becomes a matter of simple algebra by virtue of the *residue theorem.*

Theorem A.5 Residue Theorem ***If $G(z)$ is an analytic function on a simple contour Γ and inside Γ, except for a finite number of singular points $p_1, p_2, \ldots, p_P$, then***

$$\frac{1}{2\pi j} \oint_{\Gamma} G(z)\,dz = \sum_{i=1}^{P} \underset{z\to p_i}{\mathfrak{Res}}\, G(z) \tag{A.53}$$

where the integral is taken in the counterclockwise sense and $\mathfrak{Res}_{z\to p_i} G(z)$ is the residue of $G(z)$ at singular point $z = p_i$. ▲

For a rational function of the form

$$G(z) = \frac{N(z)}{D(z)} = H_0 \frac{\prod_{i=1}^{Z} (z - z_i)^{m_i}}{\prod_{i=1}^{P} (z - p_i)^{n_i}} \tag{A.54}$$

the residue at a pole $z = p_i$ of order n_i is given by the general formula

$$\underset{z=p_i}{\mathfrak{Res}}\, G(z) = \frac{1}{(n_i - 1)!} \lim_{z\to p_i} \frac{d^{n_i - 1}}{dz^{n_i - 1}} [(z - p_i)^{n_i} G(z)] \tag{A.55}$$

For a simple pole, i.e., if $n_i = 1$, we differentiate zero times and since 0!=1 by definition, we get the simplified formula

$$\underset{z=p_i}{\mathfrak{Res}}\, G(z) = \lim_{z\to p_i} [(z - p_i) G(z)]$$

A.8 ANALYTIC CONTINUATION

On many occasions in the design of analog and digital filters, a function of a complex variable z, say, $F(z)$, is known to be analytic in some specified region of the z plane, say, inside circle C_1 shown in Fig. A.10, but is otherwise unknown. Such a function can be represented by a Taylor series of the form

$$S_1 = \sum_{n=0}^{\infty} b_n (z-a)^{-n} \tag{A.56a}$$

Series S_1 represents function $F(z)$ everywhere inside circle C_1 and hence the function value and its derivatives at point $\tilde{a}$ can be determined. If function $F(z)$ is analytic in circle C_2, then a new Taylor series can obtained for $F(z)$ given by

$$S_2 = \sum_{n=0}^{\infty} \tilde{b}_n (z-\tilde{a})^{-n} \tag{A.56b}$$

where coefficients $\tilde{b}_n$ can be obtained from the derivatives of $F(z)$ at point $\tilde{a}$ using Eq. (A.56a). Series S_2 can now be used to obtain the function value and its derivatives at point $\hat{a}$. If $F(z)$ is analytic in circle C_3, a new Taylor series can obtained for $F(z)$ given by

$$S_3 = \sum_{n=0}^{\infty} \hat{b}_n (z-\hat{a})^{-n} \tag{A.56c}$$

where coefficients $\hat{b}$ can be determined from the derivatives of $F(z)$ at point $\hat{a}$. Series S_3 represents $F(z)$ everywhere in circle C_3. Through this process, the domain of validity of $F(z)$ can be extended to include the areas of circles C_2 and C_3. Proceeding in the same way, the domain of validity of $F(z)$

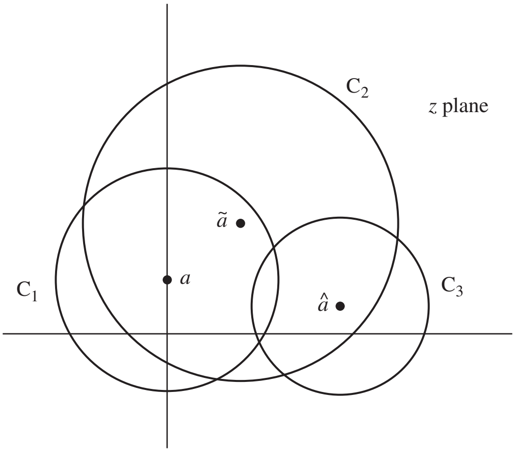


Figure A.10 Analytic continuation.

can be extended to cover all the areas of the z plane over which the function is analytic. This process is known as *analytic continuation* [5] and it has a number of applications in DSP.

The frequency response of a stable analog filter, $H_A(j\omega)$, is analytic on the $j\omega$ axis of the s plane. Through analytic continuation, the domain of the function can be extended to points off the imaginary axis and hence $j\omega$ can be replaced by $s = \sigma + j\omega$. The function obtained, namely, $H_A(s)$ is the transfer function of the analog filter and represents the filter at all points where $H_A(s)$ is not singular. Similarly, the frequency response $H(e^{j\omega T})$ of a stable digital filter is analytic on the unit circle $|z| = 1$ and on the basis of analytic continuation, $e^{j\omega T}$ can be replaced by z to obtain the transfer function $H(z)$ of the digital filter, which is valid everywhere in the z plane except at points where $H(z)$ is not singular.

A.9 CONFORMAL TRANSFORMATIONS

An equation of the form

$$w = u + jv = F(z) \tag{A.57}$$

where $z = x + jy$ is, in effect, a *transformation* that will *map* points in the z plane to corresponding points in the w plane. If each and every point in the z plane maps to one and only one point in the w plane, and conversely, then the transformation is said to be one-to-one.

An important class of transformations is the so-called class of *conformal transformations*. These are transformations that have the important property that intersecting curves in the z plane map into intersecting curves in the w plane such that the angles between the z plane curves at the point of intersection are equal to the corresponding angles in the w plane both in magnitude as well as sense. A conformal transformation is illustrated in Fig. A.11 where θ_1 and θ_2 are equal to θ_1' and θ_2', respectively. If the angles at intersection points are equal in magnitude but opposite in sense, then the transformation is said to be *isogonal*.

Theorem A.6 ***If $f(z)$ is analytic in a region $\mathcal{R}$, then the transformation in Eq. (A.57) is conformal for all points in $\mathcal{R}$ except at points where $f'(z) = 0$ (see Ref. [1]).*** ▲

Some standard conformal transformations are as follows:

1. **Translation**

$$w = z + \sigma + j\omega \tag{A.58a}$$

It translates a point $x + jy$ in the z plane to point $x + \sigma + j(y + j\omega)$ in the w plane.

2. **Rotation**

$$w = e^{j\theta}z \tag{A.58b}$$

It rotates a point $z = re^{j\phi}$ in the z plane to a point $w = re^{j(\phi+\theta)}$ in the w plane.

3. **Scaling**

$$w = \lambda z \tag{A.58c}$$

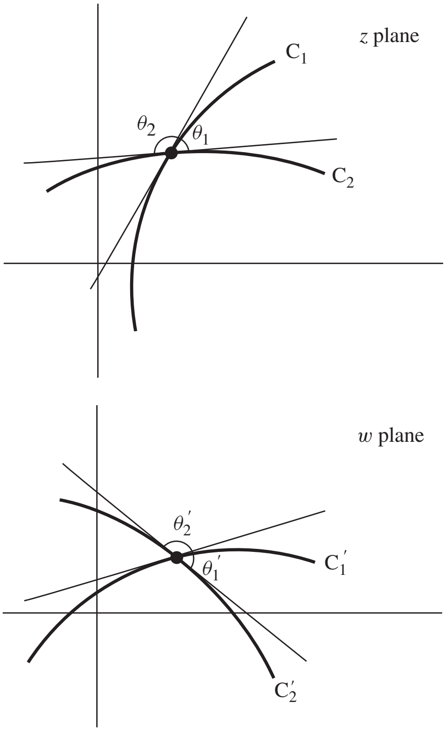


Figure A.11 Conformal transformation.

It scales a point $z = re^{j\phi}$ in the z plane to a point $w = \lambda re^{j(\phi)}$ in the w plane. If $\lambda > 1$ the magnitude of z is scaled up, and if $\lambda < 1$ it is scaled down.

4. **Rotation and scaling**

$$w = \lambda e^{j\theta} z \tag{A.58d}$$

It combines rotation as in item (2) and scaling as in item (3).

5. **Inversion**

$$w = \frac{1}{z} \tag{A.58e}$$

It inverts a point $z = re^{j\phi}$ in the z plane to a point $w = \frac{1}{r}e^{-j\phi}$ in the w plane.

5. **Inversion and scaling**

$$w = \frac{\lambda}{z} \tag{A.58f}$$

It combines inversion with scaling.

6. **Linear transformation**

$$w = \lambda z + \sigma + j\omega \tag{A.58g}$$

It combines translation and scaling as in items (1) and (3).

7. **Bilinear transformation**

$$w = \frac{\alpha z + \beta}{\gamma z + \delta} \qquad \text{where } \alpha\delta - \beta\gamma \neq 0 \tag{A.58h}$$

Through straightforward algebraic manipulation, the transformation can be expressed as

$$w = \frac{\alpha}{\gamma} + \frac{\beta\gamma - \alpha\delta}{\gamma(\gamma z + \delta)} = \varepsilon + \frac{\zeta}{z + \eta} \tag{A.58i}$$

where

$$\varepsilon = \frac{\alpha}{\gamma} \qquad \zeta = \frac{\beta\gamma - \alpha\delta}{\gamma^2} \qquad \text{and} \qquad \eta = \frac{\delta}{\gamma}$$

are constants. Now Eq. (A.58i) can be viewed as a series of transformations, namely, translation

$$w_1 = z + \eta$$

followed by inversion

$$w_2 = \frac{1}{w_1} = \frac{1}{z + \eta}$$

followed by scaling

$$w_3 = \zeta w_2 = \frac{\zeta}{z + \eta}$$

followed by translation

$$w = \varepsilon + w_3 = \varepsilon + \frac{\zeta}{z + \eta}$$

The bilinear transformation maps circles in the z plane into circles in the w plane whose relative sizes and locations depend on constants α, β, γ, and δ but by choosing $\alpha = \gamma = \delta = 1$ and $\beta = -1$ the transformation would map the $j\omega$ axis of the z plane onto the unit circle of the w plane.

An interesting feature of conformal transformations is that small figures in the z plane map into similar figures in the w plane. However, this property does not extend to large figures.

Conformal transformations are used in Chap. 10 for obtaining denormalized lowpass, highpass, bandpass, or bandstop analog filters from normalized lowpass analog filters, and in Chap. 11 for deriving digital filters of the standard types from a given lowpass digital filter.

REFERENCES

[1] E. Kreyszig, *Advanced Engineering Mathematics*, New York: Wiley, 1972.

[2] R. V. Churchill, *Complex Variables and Applications*, New York: McGraw-Hill, 1960.

[3] M. R. Spiegel, *Complex Variables*, New York: McGraw-Hill, 1964.

[4] Biographies Index of the The MacTutor History of Mathematics Archive, School of Mathematics and Statistics, University of St. Andrews, Scotland: http://www-groups.dcs.st-and.ac.uk/ history/BiogIndex.html

[5] W. R. LePage, *Complex Variables and the Laplace Transform for Engineers*, New York: McGraw-Hill, 1961.

APPENDIX B

ELLIPTIC FUNCTIONS

B.1 INTRODUCTION

The Jacobian elliptic functions are derived by employing the Legendre elliptic integral of the first kind. Their theory is quite extensive and is discussed in detail by Bowman [1] and Hancock [2, 3]. We provide here a brief but adequate treatment of this theory to facilitate the understanding of the derivation of the elliptic approximation in Chap. 10 [4].

B.2 ELLIPTIC INTEGRAL OF THE FIRST KIND

The *elliptic integral of the first kind* can be expressed as

$$u \equiv u(\phi, k) = \int_0^{\phi} \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}} \tag{B.1}$$

where $0 \le k < 1$. The parameter k is called the *modulus* and the upper limit of integration ϕ is called the *amplitude* of the integral. Evidently, for a real value of ϕ, $u(\phi, k)$ is real and represents the area bounded by the curve

$$I = \frac{1}{\sqrt{1 - k^2 \sin^2 \theta}}$$

and the vertical lines $\theta = 0$ and $\theta = \phi$. Plots of I and $u(\phi, k)$ for $k = 0.995$ are shown in Fig. B.1. The integrand I has minima equal to unity at $\theta = 0, \pi, 2\pi \ldots$ and maxima equal to $1/\sqrt{1 - k^2}$ at

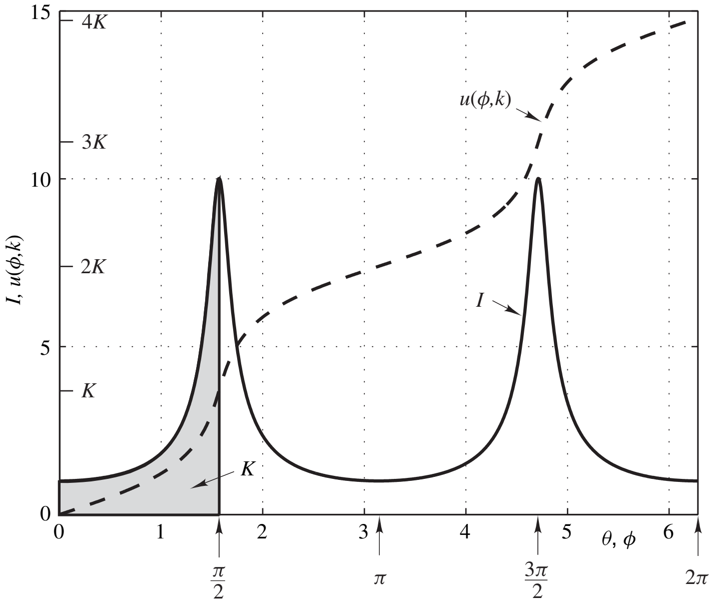


Figure B.1 Plots of I versus θ and $u(\phi, k)$ versus ϕ for $k = 0.995$.

$\theta = \pi/2, 3\pi/2, \ldots$. In effect, I is a periodic function of θ with a period π. The area bounded by lines $\theta = n\pi/2$ and $\theta = (n+1)\pi/2$ is constant for any n because of the symmetry of I and is equal to the area bounded by lines $\theta = 0$ and $\theta = \pi/2$. This area is referred to as the *complete* elliptic integral of the first kind and is given by

$$u\left(\frac{\pi}{2}, k\right) = K = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}} \tag{B.2}$$

(see Fig. B.1).

As a consequence of the periodicity and symmetry of I, we can write

$$u(n\pi + \phi_1, k) = 2nK + u(\phi_1, k) \qquad \text{and} \qquad u\left(\frac{\pi}{2} + \phi_1, k\right) = 2K - u\left(\frac{\pi}{2} - \phi_1, k\right)$$

where $0 \le \phi_1 < \pi/2$. That is, the elliptic integral for a given k and any real ϕ can be determined from a table giving the values of the integral in the interval $0 \le \phi < \pi/2$.

If $k = 0$, Eq. (B.1) gives

$$u(\phi, 0) = \int_0^{\phi} d\theta = \phi$$

and if $k = 1$,

$$u(\phi, 1) = \int_0^{\phi} \frac{d\theta}{\cos\theta} = \ln\left[\tan\left(\frac{\pi}{4} + \frac{\phi}{2}\right)\right]$$

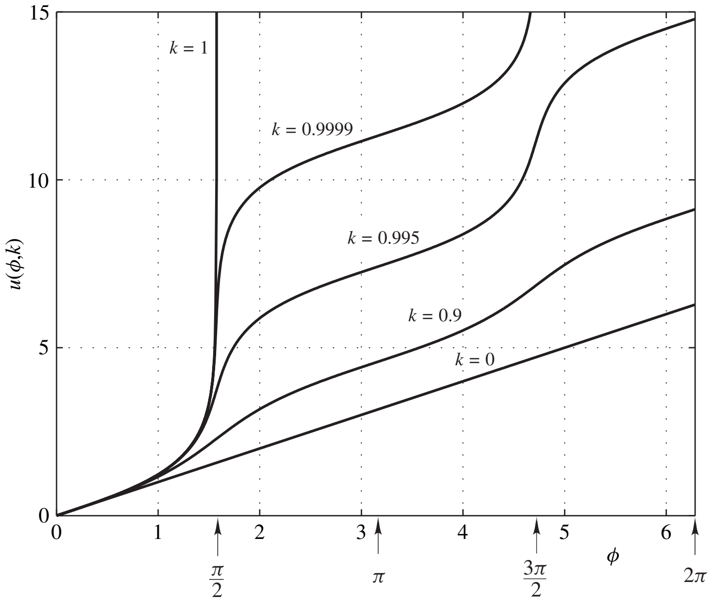


Figure B.2 Plots of u versus ϕ for various values of k.

according to standard integral tables. Hence $u(\phi, 0)$ increases linearly with ϕ, whereas $u(\phi, 1)$ is discontinuous at $\phi = \pi/2$. For $0 \leq \phi < \pi/2$

$$u(\phi, 0) \leq u(\phi, k) \leq u(\phi, 1)$$

as can be seen in Fig. B.2.

B.3 ELLIPTIC FUNCTIONS

Figure B.2 demonstrates a one-to-one correspondence between u and ϕ. Thus for a given pair of values (u, k) there corresponds a unique amplitude ϕ such that

$$\phi = f(u, k)$$

The *Jacobian elliptic functions* are defined as

$$\text{sn}(u, k) = \sin\phi \tag{B.3}$$

$$\text{cn}(u, k) = \cos\phi \tag{B.4}$$

$$\text{dn}(u, k) = \sqrt{1 - k^2 \sin^2\phi} \tag{B.5}$$

Many of the properties of elliptic functions follow directly from the properties of trigonometric functions. For example, we can write

$$\text{sn}^2(u, k) + \text{cn}^2(u, k) = 1 \tag{B.6}$$

and

$$k^2\,\text{sn}^2(u, k) + \text{dn}^2(u, k) = 1 \tag{B.7}$$

and so forth.

Plots of the elliptic functions versus u can be constructed as in Fig. B.3. As can be seen, $\mathrm{sn}(u, k)$, $\mathrm{cn}(u, k)$, and $\mathrm{dn}(u, k)$ are periodic functions u with periods $4K$, $4K$, and $2K$, respectively, i.e.,

$$\mathrm{sn}(u + 4mK, k) = \mathrm{sn}(u, k) \tag{B.8}$$

$$\mathrm{cn}(u + 4mK, k) = \mathrm{cn}(u, k) \tag{B.9}$$

$$\mathrm{dn}(u + 2mK, k) = \mathrm{dn}(u, k) \tag{B.10}$$

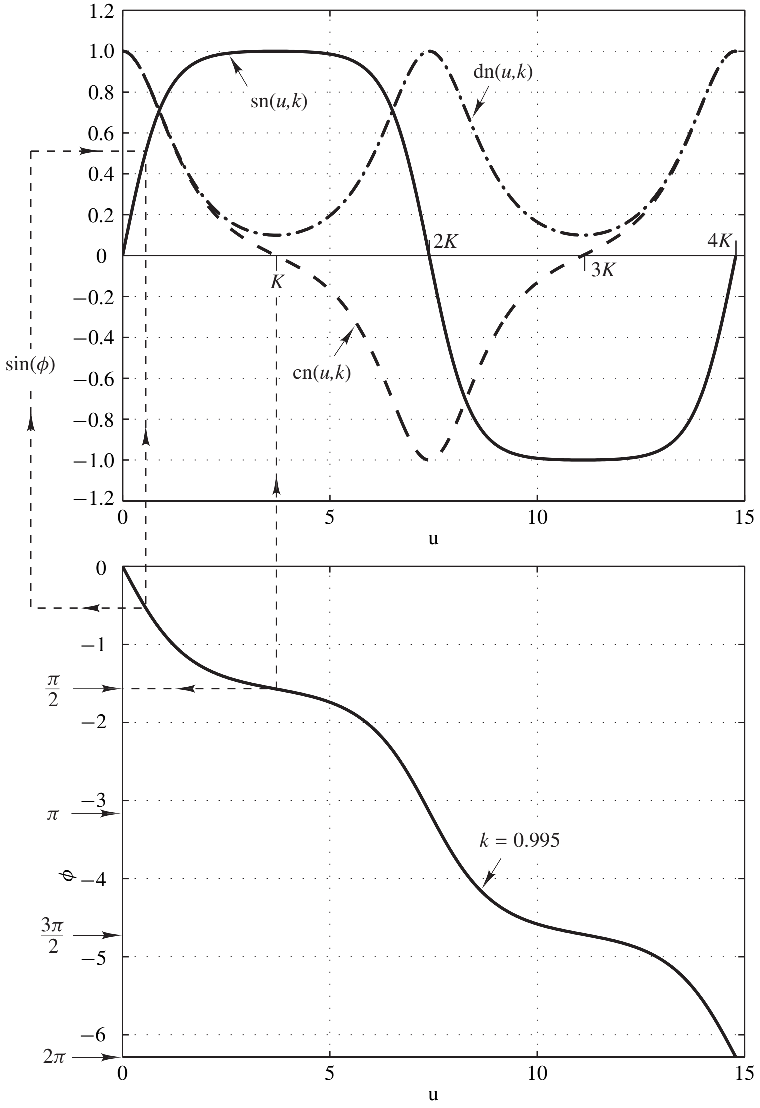


Figure B.3 Plots of $\mathrm{sn}(u, k)$, $\mathrm{cn}(u, k)$, and $\mathrm{dn}(u, k)$ versus u.

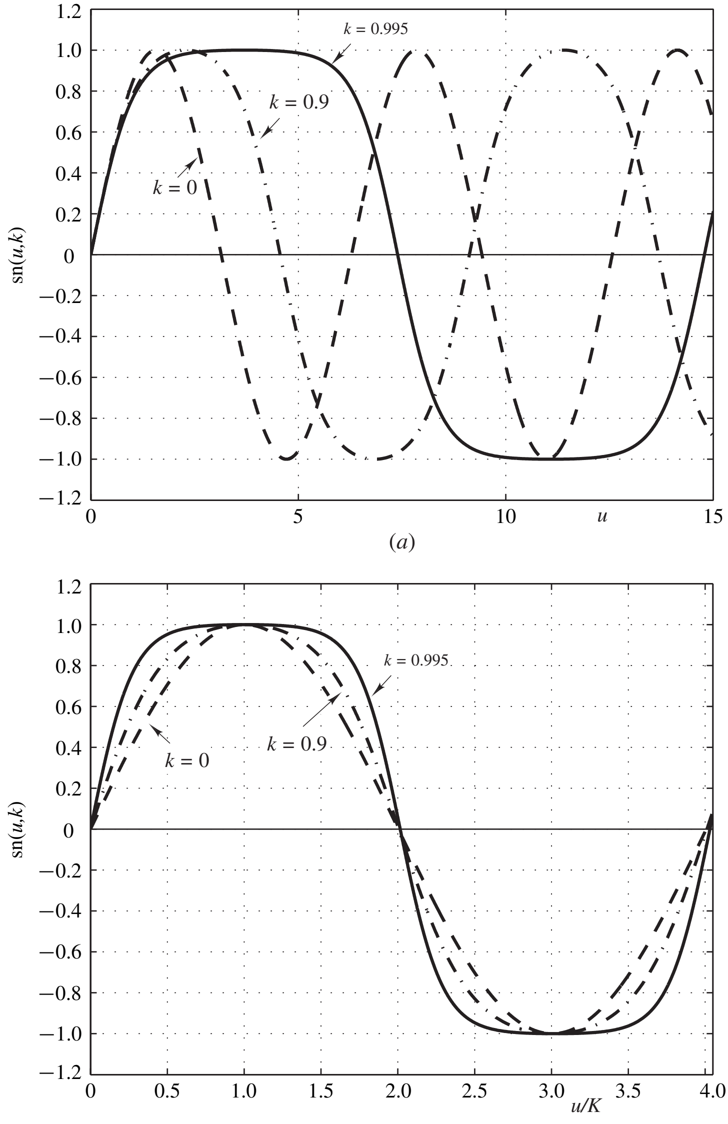


Figure B.4 Effect of variations in k on the elliptic sine (a) sn(n, k) versus u, (b) sn(n, k) versus u/K.

Variations in k tend to change the shape and period of the elliptic functions, as illustrated in Fig. B.4. i.e., *the elliptic sine and cosine are generalizations of the conventional sine and cosine*, respectively.

If $k = 0$, we have

$$u(\phi, 0) = \phi$$

and so $$\text{sn}(u, 0) = \text{sn}(\phi, 0) = \sin\phi \qquad \text{cn}(u, 0) = \text{cn}(\phi, 0) = \cos\phi$$

that is, sn(n, k) and cn(n, k) become the usual sine and cosine functions of ϕ.

B.4 IMAGINARY ARGUMENT

Thus far the argument of the elliptic functions, namely, u, has been assumed to be a real quantity. By performing the integration of Eq. (B.1) over an appropriate path in a complex plane, the elliptic integral can assume complex values. Let us consider the case of imaginary value whereby

$$jv = \int_0^{\psi} \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}} \tag{B.11}$$

As in Sec. B.3, we can define

$$\operatorname{sn}(jv, k) = \sin \psi \tag{B.12}$$

$$\operatorname{cn}(jv, k) = \cos \psi \tag{B.13}$$

$$\operatorname{dn}(jv, k) = \sqrt{1 - k^2 \sin^2 \psi} \tag{B.14}$$

These functions can be expressed in terms of elliptic functions that have real arguments, as we will now show.

By applying the transformations

$$\sin \theta = j \tan \theta' \qquad \sin \psi = j \tan \psi' \tag{B.15}$$

in Eq. (B.11), we have

$$jv = \int_0^{\psi'} \frac{j\, d\theta'}{\sqrt{1 - \sin^2 \theta' + k^2 \sin^2 \theta'}}$$

Alternatively,

$$v = \int_0^{\psi'} \frac{d\theta'}{\sqrt{1 - (k')^2 \sin^2 \theta'}}$$

where k', given by

$$k' = \sqrt{1 - k^2}$$

is called the *complementary* modulus. Now, from Sec. B.3

$$\operatorname{sn}(v, k') = \sin \psi' \tag{B.16}$$

$$\operatorname{cn}(v, k') = \cos \psi' \tag{B.17}$$

$$\operatorname{dn}(v, k') = \sqrt{1 - (k')^2 \sin^2 \psi'} \tag{B.18}$$

and, therefore, from Eqs. (B.12)–(B.18),

$$\operatorname{sn}(jv, k) = j\tan\psi' = j\frac{\sin\psi'}{\cos\psi'} = \frac{j\operatorname{sn}(v, k')}{\operatorname{cn}(v, k')} \tag{B.19}$$

$$\operatorname{cn}(jv, k) = \frac{1}{\operatorname{cn}(v, k')} \tag{B.20}$$

$$\operatorname{dn}(jv, k) = \frac{\operatorname{dn}(v, k')}{\operatorname{cn}(v, k')} \tag{B.21}$$

By analogy with Eq. (B.2), the *complementary* complete integral of the first kind is given by

$$K' = \int_0^{\pi/2} \frac{d\theta'}{\sqrt{1-(k')^2\sin^2\theta'}}$$

This has a similar interpretation as K; that is, it is the quarter period of $\operatorname{sn}(v, k')$ and $\operatorname{cn}(v, k')$ or the half period of $\operatorname{dn}(v, k')$.

The functions $\operatorname{sn}(jv, k)$, $\operatorname{cn}(jv, k)$, and $\operatorname{dn}(jv, k)$ are periodic functions of jv, as can be seen in Fig. B.5, with periods $j2K'$, $j4K'$, and $j4K'$, respectively, i.e.,

$$\operatorname{sn}(jv + j2nK', k) = \operatorname{sn}(jv, k)$$

$$\operatorname{cn}(jv + j4nK', k) = \operatorname{cn}(jv, k)$$

$$\operatorname{dn}(jv + j4nK', k) = \operatorname{dn}(jv, k)$$

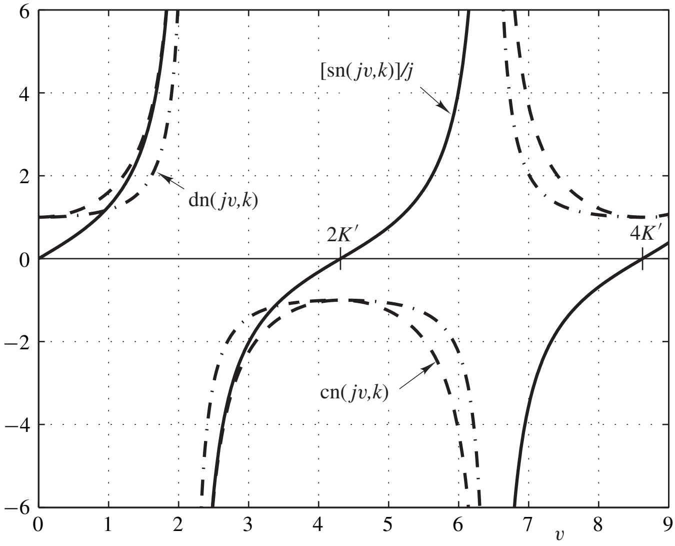


Figure B.5 Plots of $[\operatorname{sn}(jv, k)]/j$, $\operatorname{cn}(jv, k)$, and $\operatorname{dn}(jv, k)$ versus v.

B.5 FORMULAS

Elliptic functions, like trigonometric functions, are interrelated by many useful formulas. The most basic one is the *addition formula*, which is of the form

$$\operatorname{sn}(z_1+z_2,k)=\frac{\operatorname{sn}(z_1,k)\operatorname{cn}(z_2,k)\operatorname{dn}(z_2,k)+\operatorname{cn}(z_1,k)\operatorname{sn}(z_2,k)\operatorname{dn}(z_1,k)}{D} \tag{B.22}$$

where $$D=1-k^2\operatorname{sn}^2(z_1,k)\operatorname{sn}^2(z_2,k)$$

The variables z_1 and z_2 can assume real or complex values. By using the above formula and Eqs. (B.6) and (B.7) we can show that

$$\operatorname{cn}(z_1+z_2,k)=\frac{\operatorname{cn}(z_1,k)\operatorname{cn}(z_2,k)-\operatorname{sn}(z_1,k)\operatorname{sn}(z_2,k)\operatorname{dn}(z_1,k)\operatorname{dn}(z_2,k)}{D} \tag{B.23}$$

$$\operatorname{dn}(z_1+z_2,k)=\frac{\operatorname{dn}(z_1,k)\operatorname{dn}(z_2,k)-k^2\operatorname{sn}(z_1,k)\operatorname{sn}(z_2,k)\operatorname{cn}(z_1,k)\operatorname{cn}(z_2,k)}{D} \tag{B.24}$$

Another formula of interest is

$$\operatorname{dn}^2\left(\frac{z}{2},k\right)=\frac{\operatorname{dn}(z,k)+\operatorname{cn}(z,k)}{1+\operatorname{cn}(z,k)} \tag{B.25}$$

B.6 PERIODICITY

In the preceding sections we have demonstrated that $\operatorname{sn}(z,k)$, where $z=u+jv$, has a real period of $4K$ if $v=0$ and an imaginary period of $2K'$ if $u=0$. In fact these are general properties for any value of v or u as can be easily shown. From the addition formula

$$\operatorname{sn}(z+4mK,k)=\frac{\operatorname{sn}(z,k)\operatorname{cn}(4mK,k)\operatorname{dn}(4mK,k)+\operatorname{cn}(z,k)\operatorname{sn}(4mK,k)\operatorname{dn}(z,k)}{1-k^2\operatorname{sn}^2(z,k)\operatorname{sn}^2(4mK,k)}$$

and since

$$\operatorname{sn}(4mK,k)=\operatorname{sn}(0,k)=0$$
$$\operatorname{cn}(4mK,k)=\operatorname{cn}(0,k)=1$$
$$\operatorname{dn}(4mK,k)=\operatorname{dn}(0,k)=1$$

according to Eqs. (B.8)–(B.10), it follows that

$$\operatorname{sn}(z+4mK,k)=\operatorname{sn}(z,k) \tag{B.26}$$

Similarly,

$$\operatorname{sn}(z+j2nK',k)=\frac{\operatorname{sn}(z,k)\operatorname{cn}(j2nK',k)\operatorname{dn}(j2nK',k)+\operatorname{cn}(z,k)\operatorname{sn}(j2nK',k)\operatorname{dn}(z,k)}{1-k^2\operatorname{sn}^2(z,k)\operatorname{sn}^2(j2nK',k)}$$

and from Eqs. (B.19)–(B.21)

$$\begin{aligned}
\text{sn}(j2nK', k) &= \frac{j\,\text{sn}(2nK', k')}{\text{cn}(2nK', k')} = 0\\
\text{cn}(j2nK', k) &= \frac{1}{\text{cn}(2nK', k')} = (-1)^n\\
\text{dn}(j2nK', k) &= \frac{\text{dn}(2nK', k')}{\text{cn}(2nK', k')} = (-1)^n
\end{aligned}$$

Hence we have

$$\text{sn}(z + j2nK', k) = \text{sn}(z, k) \tag{B.27}$$

Therefore, by combining Eqs. (B.26) and (B.27) we obtain

$$\text{sn}(z + 4mK + j2nK', k) = \text{sn}(z, k)$$

that is, sn(z, k) is a *doubly periodic* function of z with a real period of $4K$ and an imaginary period of $2K'$.

The z plane can be subdivided into *period parallelograms* by means of lines

$$u = 4mK \qquad \text{and} \qquad jv = j2nK'$$

as illustrated in Fig. B.6. The specific parallelogram defined by vertices (0, 0), $(4K, 0)$, $(4K, j2K')$, and $(0, j2K')$ is called the *fundamental period parallelogram.* If the value of sn(z, k) is known for each and every value of z within this parallelogram and along any two adjacent sides, the function is known over the entire z plane.

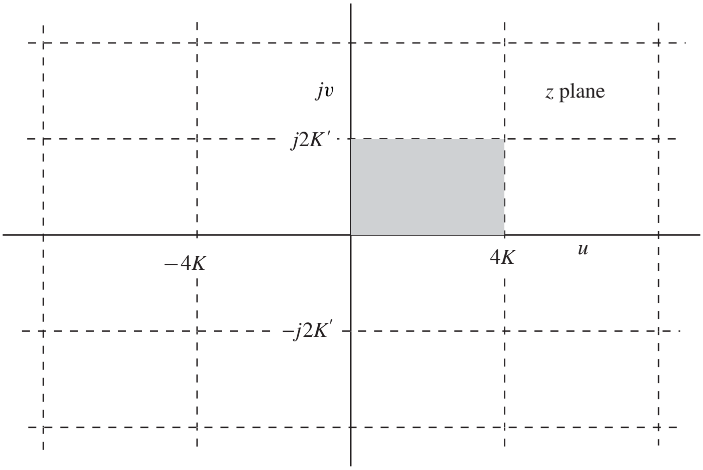


Figure B.6 Period parallelograms of sn(z, k).

Similarly, the functions $\mathrm{cn}(z, k)$ and $\mathrm{dn}(z, k)$ can be shown to be doubly periodic. The first has a real period of $4K$ and an imaginary period of $4K'$, whereas the second has a real period of $2K$ and an imaginary period of $4K'$.

B.7 TRANSFORMATION

The equation

$$\omega = \sqrt{k}\,\mathrm{sn}(z, k) \tag{B.28}$$

is essentially a variable transformation that maps points in the z plane onto corresponding points in the ω plane. Let us examine the mapping properties of this transformation. These are required in the derivation of $F(\omega)$ in Sec. 10.6.

A point z_p as well as all points

$$z = z_p + 4mK + j2nK'$$

map onto a single point in the ω plane by virtue of the periodicity of $\mathrm{sn}(z, k)$. Hence, only points in the fundamental period parallelogram need be considered. Three domains of $\sqrt{k}\,\mathrm{sn}(z, k)$ are of interest as follows:

- Domain 1: $z = u$ with $0 \le u \le K$
- Domain 2: $z = K + jv$ with $0 \le v \le K'$
- Domain 3: $z = u + jK'$ with $0 \le u \le K$

In domain 1, we have

$$\omega = \sqrt{k}\,\mathrm{sn}(u, k)$$

If $u = 0$, then

$$\omega = \sqrt{k}\,\mathrm{sn}(0, k) = 0$$

and if $u = K$, we obtain

$$\omega = \sqrt{k}\,\mathrm{sn}(K, k) = \sqrt{k}$$

that is, Eq. (B.28) maps points on the real axis of the z plane between 0 and K onto points on the real axis of the ω plane between 0 and $\sqrt{k}$.

In domain 2, we have

$$\omega = \sqrt{k}\,\mathrm{sn}(K + jv, k)$$

From the addition formula

$$\omega = \frac{\sqrt{k}\,\mathrm{cn}(jv, k)\,\mathrm{dn}(jv, k)}{1 - k^2\,\mathrm{sn}^2(jv, k)} \tag{B.29}$$

since $\mathrm{cn}(K, k) = 0$, and from Eqs. (B.19)–(B.21)

$$\omega = \frac{\sqrt{k}\,\mathrm{dn}(v, k')}{\mathrm{cn}^2(v, k') + k^2\,\mathrm{sn}^2(v, k')}$$

Now from Eqs. (B.6) and (B.7)

$$\begin{aligned}\mathrm{cn}^2(v, k') + k^2\,\mathrm{sn}^2(v, k') &= 1 - \mathrm{sn}^2(v, k') + k^2\,\mathrm{sn}^2(v, k')\\ &= 1 - (k')^2\,\mathrm{sn}^2(v, k') = \mathrm{dn}^2(v, k')\end{aligned}$$

Therefore, Eq. (B.29) simplifies to

$$\omega = \frac{\sqrt{k}}{\mathrm{dn}(v, k')}$$

If $v = 0$, then

$$\omega = \frac{\sqrt{k}}{\mathrm{dn}(0, k')} = \sqrt{k}$$

and if $v = K'$, we have

$$\omega = \frac{\sqrt{k}}{\mathrm{dn}(K', k')} = \frac{1}{\sqrt{k}}$$

For $v = K'/2$, the use of Eq. (B.25) yields

$$\omega = \frac{\sqrt{k}}{\mathrm{dn}(K'/2, k')} = \sqrt{k}\left[\frac{1 + \mathrm{cn}(K', k')}{\mathrm{dn}(K', k') + \mathrm{cn}(K', k')}\right]^{1/2} = 1$$

Thus Eq. (B.28) maps points on the line $z = K + jv$ for v between 0 and K' onto points on the real axis of the ω plane between $\sqrt{k}$ and $1/\sqrt{k}$; in particular, point $z = K + jK'/2$ maps onto point $\omega = 1$.

In domain 3, Eq. (B.28) assumes the form

$$\omega = \sqrt{k}\,\mathrm{sn}(u + jK', k)$$

and, as above, Eq. (B.22) yields

$$\omega = \frac{1}{\sqrt{k}\,\mathrm{sn}(u, k)}$$

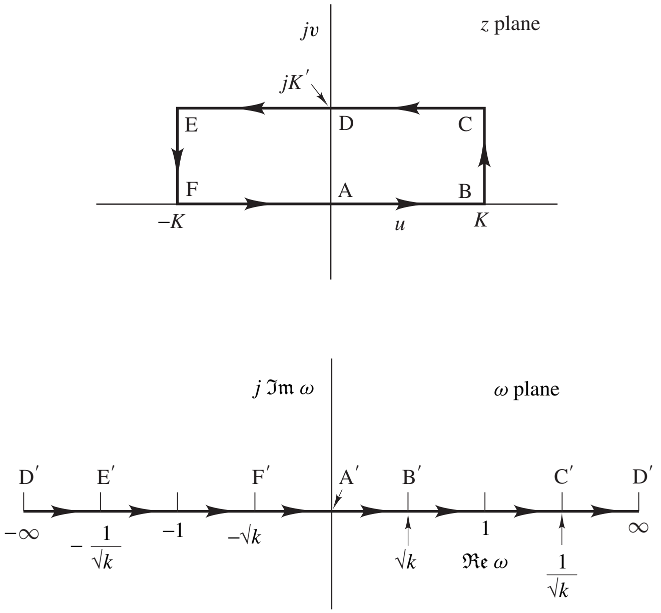


Figure B.7 Mapping properties of transformation $\omega = \sqrt{k}\,\mathrm{sn}(z, k)$.

If $u = 0$, then

$$\omega = \frac{1}{\sqrt{k}\,\mathrm{sn}(0, k)} = \infty$$

and if $u = K$, we get

$$\omega = \frac{1}{\sqrt{k}\,\mathrm{sn}(K, k)} = \frac{1}{\sqrt{k}}$$

i.e., points on line $z = u + jK'$ with u between 0 and K map onto the real axis of the ω plane between ∞ and $1/\sqrt{k}$.

By considering mirror-image points to those considered so far, the mapping depicted in Fig. B.7 can be completed, where points A, B, ... map onto points A', B', ...

B.8 SERIES REPRESENTATION

Elliptic functions, like trigonometric functions, can be represented in terms of series. From Ref. [3] or [4],

$$\mathrm{sn}(z, k) = \frac{1}{\sqrt{k}} \frac{\theta_1(z/2K, q)}{\theta_0(z/2K, q)} \tag{B.30}$$

$$\mathrm{cn}(z, k) = \sqrt{\frac{k'}{k}} \frac{\theta_2(z/2K, q)}{\theta_0(z/2K, q)} \tag{B.31}$$

$$\mathrm{dn}(z, k) = \sqrt{k'} \frac{\theta_3(z/2K, q)}{\theta_0(z/2K, q)} \tag{B.32}$$

The parameter q is known as the *modular constant* and is given by

$$q = e^{-\pi K'/K}$$

The functions $\theta_0(z/2K, q)$ to $\theta_3(z/2K, q)$ are called *theta functions* and are given by

$$\theta_0\left(\frac{z}{2K}, q\right) = 1 + 2\sum_{m=1}^{\infty}(-1)^m q^{m^2} \cos\left(2m\frac{\pi z}{2K}\right)$$

$$\theta_1\left(\frac{z}{2K}, q\right) = 2q^{1/4}\sum_{m=0}^{\infty}(-1)^m q^{m(m+1)} \sin\left[(2m+1)\frac{\pi z}{2K}\right]$$

$$\theta_2\left(\frac{z}{2K}, q\right) = 2q^{1/4}\sum_{m=0}^{\infty} q^{m(m+1)} \cos\left[(2m+1)\frac{\pi z}{2K}\right]$$

$$\theta_3\left(\frac{z}{2K}, q\right) = 1 + 2\sum_{m=1}^{\infty} q^{m^2} \cos\left(2m\frac{\pi z}{2K}\right)$$

The above series converge rapidly and can be used to evaluate the elliptic functions to any desired degree of accuracy.

REFERENCES

[1] F. Bowman, *Introduction to Elliptic Functions with Applications*, New York: Dover, 1961.

[2] H. Hancock, *Elliptic Integrals*, New York: Dover, 1958.

[3] H. Hancock, *Lectures on the Theory of Elliptic Functions*, New York: Dover, 1958.

[4] A. J. Grossman, “Synthesis of Tchebyscheff Parameter Symmetrical Filters,” *Proc. IRE*, vol. 45, pp. 454–473, Apr. 1957.

This page intentionally left blank

INDEX

In index entries with more than one page number, the bold page number designates the more significant citation.