INTRODUCTION TO DIGITAL SIGNAL PROCESSING AND FILTER DESIGN

B. A. Shenoi

A JOHN WILEY & SONS, INC., PUBLICATION

INTRODUCTION TO DIGITAL SIGNAL PROCESSING AND FILTER DESIGN

INTRODUCTION TO DIGITAL SIGNAL PROCESSING AND FILTER DESIGN

B. A. Shenoi

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

ISBN-13 978-0-471-46482-2 (cloth)
ISBN-10 0-471- 46482-1 (cloth)

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

CONTENTS

PREFACE

This preface is addressed to instructors as well as students at the junior–senior level for the following reasons. I have been teaching courses on digital signal processing, including its applications and digital filter design, at the undergraduate and the graduate levels for more than 25 years. One common complaint I have heard from undergraduate students in recent years is that there are not enough numerical problems worked out in the chapters of the book prescribed for the course. But some of the very well known textbooks on digital signal processing have more problems than do a few of the books published in earlier years. However, these books are written for students in the senior and graduate levels, and hence the junior-level students find that there is too much of mathematical theory in these books. They also have concerns about the advanced level of problems found at the end of chapters. I have not found a textbook on digital signal processing that meets these complaints and concerns from junior-level students. So here is a book that I have written to meet the junior students' needs and written with a student-oriented approach, based on many years of teaching courses at the junior level.

Network Analysis is an undergraduate textbook authored by my Ph.D. thesis advisor Professor M. E. Van Valkenburg (published by Prentice-Hall in 1964), which became a world-famous classic, not because it contained an abundance of all topics in network analysis discussed with the rigor and beauty of mathematical theory, but because it helped the students understand the basic ideas in their simplest form when they took the first course on network analysis. I have been highly influenced by that book, while writing this textbook for the first course on digital signal processing that the students take. But I also have had to remember that the generation of undergraduate students is different; the curriculum and the topic of digital signal processing is also different. This textbook does not contain many of the topics that are found in the senior–graduate-level textbooks mentioned above. One of its main features is that it uses a very large number of numerical problems as well as problems using functions from MATLAB® (MATLAB is a registered trademark of The MathWorks, Inc.) and Signal Processing Toolbox, worked out in every chapter, in order to highlight the fundamental concepts. These problems are solved as examples after the theory is discussed or are worked out first and the theory is then presented. Either way, the thrust of the approach is that the students should understand the basic ideas, using the worked, out problems as an instrument to achieve that goal. In some cases, the presentation is more informal than in other cases. The students will find statements beginning with "Note that. . .," "Remember. . .," or "It is pointed out," and so on; they are meant

to emphasize the important concepts and the results stated in those sentences. Many of the important results are mentioned more than once or summarized in order to emphasize their significance.

The other attractive feature of this book is that all the problems given at the end of the chapters are problems that can be solved by using only the material discussed in the chapters, so that students would feel confident that they have an understanding of the material covered in the course when they succeed in solving the problems. Because of such considerations mentioned above, the author claims that the book is written with a student-oriented approach. Yet, the students should know that the ability to understand the solution to the problems is important but understanding the theory behind them is far more important.

The following paragraphs are addressed to the instructors teaching a junior-level course on digital signal processing. The first seven chapters cover well-defined topics: (1) an introduction, (2) time-domain analysis and z-transform, (3) frequency-domain analysis, (4) infinite impulse response filters, (5) finite impulse response filters, (6) realization of structures, and (7) quantization filter analysis. Chapter 8 discusses hardware design, and Chapter 9 covers MATLAB. The book treats the mainstream topics in digital signal processing with a well-defined focus on the fundamental concepts.

Most of the senior–graduate-level textbooks treat the theory of finite wordlength in great detail, but the students get no help in analyzing the effect of finite word-length on the frequency response of a filter or designing a filter that meets a set of frequency response specifications with a given wordlength and quantization format. In Chapter 7, we discuss the use of a MATLAB tool known as the "FDA Tool" to thoroughly investigate the effect of finite wordlength and different formats of quantization. This is another attractive feature of the textbook, and the material included in this chapter is not found in any other textbook published so far.

When the students have taken a course on digital signal processing, and join an industry that designs digital signal processing (DSP) systems using commercially available DSP chips, they have very little guidance on what they need to learn. It is with that concern that additional material in Chapter 8 has been added, leading them to the material that they have to learn in order to succeed in their professional development. It is very brief but important material presented to guide them in the right direction. The textbooks that are written on DSP hardly provide any guidance on this matter, although there are quite a few books on the hardware implementation of digital systems using commercially available DSP chips. Only a few schools offer laboratory-oriented courses on the design and testing of digital systems using such chips. Even the minimal amount of information in Chapter 8 is not found in any other textbook that contains "digital signal processing" in its title. However, Chapter 8 is not an exhaustive treatment of hardware implementation but only as an introduction to what the students have to learn when they begin a career in the industry.

Chapter 1 is devoted to discrete-time signals. It describes some applications of digital signal processing and defines and, suggests several ways of describing discrete-time signals. Examples of a few discrete-time signals and some basic

operations applied with them is followed by their properties. In particular, the properties of complex exponential and sinusoidal discrete-time signals are described. A brief history of analog and digital filter design is given. Then the advantages of digital signal processing over continuous-time (analog) signal processing is discussed in this chapter.

Chapter 2 is devoted to discrete-time systems. Several ways of modeling them and four methods for obtaining the response of discrete-time systems when excited by discrete-time signals are discussed in detail. The four methods are (1) recursive algorithm, (2) convolution sum, (3) classical method, and (4) z-transform method to find the total response in the time domain. The use of z-transform theory to find the zero state response, zero input response, natural and forced responses, and transient and steady-state responses is discussed in great detail and illustrated with many numerical examples as well as the application of MATLAB functions. Properties of discrete-time systems, unit pulse response and transfer functions, stability theory, and the Jury–Marden test are treated in this chapter. The amount of material on the time-domain analysis of discrete-time systems is a lot more than that included in many other textbooks.

Chapter 3 concentrates on frequency-domain analysis. Derivation of sampling theorem is followed by the derivation of the discrete-time Fourier transform (DTFT) along with its importance in filter design. Several properties of DTFT and examples of deriving the DTFT of typical discrete-time signals are included with many numerical examples worked out to explain them. A large number of problems solved by MATLAB functions are also added. This chapter devoted to frequency-domain analysis is very different from those found in other textbooks in many respects.

The design of infinite impulse response (IIR) filters is the main topic of Chapter 4. The theory of approximation of analog filter functions, design of analog filters that approximate specified frequency response, the use of impulse-invariant transformation, and bilinear transformation are discussed in this chapter. Plenty of numerical examples are worked out, and the use of MATLAB functions to design many more filters are included, to provide a hands-on experience to the students.

Chapter 5 is concerned with the theory and design of finite impulse response (FIR) filters. Properties of FIR filters with linear phase, and design of such filters by the Fourier series method modified by window functions, is a major part of this chapter. The design of equiripple FIR filters using the Remez exchange algorithm is also discussed in this chapter. Many numerical examples and MATLAB functions are used in this chapter to illustrate the design procedures.

After learning several methods for designing IIR and FIR filters from Chapters 4 and 5, the students need to obtain as many realization structures as possible, to enable them to investigate the effects of finite wordlength on the frequency response of these structures and to select the best structure. In Chapter 6, we describe methods for deriving several structures for realizing FIR filters and IIR filters. The structures for FIR filters describe the direct, cascade, and polyphase forms and the lattice structure along with their transpose forms. The structures for

IIR filters include direct-form and cascade and parallel structures, lattice–ladder structures with autoregressive (AR), moving-average (MA), and allpass structures as special cases, and lattice-coupled allpass structures. Again, this chapter contains a large number of examples worked out numerically and using the functions from MATLAB and Signal Processing Toolbox; the material is more than what is found in many other textbooks.

The effect of finite wordlength on the frequency response of filters realized by the many structures discussed in Chapter 6 is treated in Chapter 7, and the treatment is significantly different from that found in all other textbooks. There is no theoretical analysis of finite wordlength effect in this chapter, because it is beyond the scope of a junior-level course. I have chosen to illustrate the use of a MATLAB tool called the "FDA Tool" for investigating these effects on the different structures, different transfer functions, and different formats for quantizing the values of filter coefficients. The additional choices such as truncation, rounding, saturation, and scaling to find the optimum filter structure, besides the alternative choices for the many structures, transfer functions, and so on, makes this a more powerful tool than the theoretical results. Students would find experience in using this tool far more useful than the theory in practical hardware implementation.

Chapters 1–7 cover the core topics of digital signal processing. Chapter 8, on hardware implementation of digital filters, briefly describes the simulation of digital filters on Simulink®, and the generation of C code from Simulink using Real-Time Workshop® (Simulink and Real-Time Workshop are registered trademarks of The MathWorks, Inc.), generating assembly language code from the C code, linking the separate sections of the assembly language code to generate an executable object code under the Code Composer Studio from Texas Instruments is outlined. Information on DSP Development Starter kits and simulator and emulator boards is also included. Chapter 9, on MATLAB and Signal Processing Toolbox, concludes the book.

The author suggests that the first three chapters, which discuss the basics of digital signal processing, can be taught at the junior level in one quarter. The prerequisite for taking this course is a junior-level course on linear, continuous-time signals and systems that covers Laplace transform, Fourier transform, and Fourier series in particular. Chapters 4–7, which discuss the design and implementation of digital filters, can be taught in the next quarter or in the senior year as an elective course depending on the curriculum of the department. Instructors must use discretion in choosing the worked-out problems for discussion in the class, noting that the real purpose of these problems is to help the students understand the theory. There are a few topics that are either too advanced for a junior-level course or take too much of class time. Examples of such topics are the derivation of the objective function that is minimized by the Remez exchange algorithm, the formulas for deriving the lattice–ladder realization, and the derivation of the fast Fourier transform algorithm. It is my experience that students are interested only in the use of MATLAB functions that implement these algorithms, and hence I have deleted a theoretical exposition of the last two topics and also a description

of the optimization technique in the Remez exchange algorithm. However, I have included many examples using the MATLAB functions to explain the subject matter.

Solutions to the problems given at the end of chapters can be obtained by the instructors from the Website `http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471464821.html`. They have to access the solutions by clicking "Download the software solutions manual link" displayed on the Webpage. The author plans to add more problems and their solutions, posting them on the Website frequently after the book is published.

As mentioned at the beginning of this preface, the book is written from my own experience in teaching a junior-level course on digital signal processing. I wish to thank Dr. M. D. Srinath, Southern Methodist University, Dallas, for making a thorough review and constructive suggestions to improve the material of this book. I also wish to thank my colleague Dr. A. K. Shaw, Wright State University, Dayton. And I am most grateful to my wife Suman, who has spent hundreds of lonely hours while I was writing this book. Without her patience and support, I would not have even started on this project, let alone complete it. So I dedicate this book to her and also to our family.

B. A. SHENOI

May 2005

CHAPTER 1

Introduction

1.1 INTRODUCTION

We are living in an age of information technology. Most of this technology is based on the theory of digital signal processing (DSP) and implementation of the theory by devices embedded in what are known as *digital signal processors* (DSPs). Of course, the theory of digital signal processing and its applications is supported by other disciplines such as computer science and engineering, and advances in technologies such as the design and manufacturing of very large scale integration (VLSI) chips. The number of devices, systems, and applications of digital signal processing currently affecting our lives is very large and there is no end to the list of new devices, systems, and applications expected to be introduced into the market in the coming years. Hence it is difficult to forecast the future of digital signal processing and the impact of information technology. Some of the current applications are described below.

1.2 APPLICATIONS OF DSP

Digital signal processing is used in several areas, including the following:

1. *Telecommunications*. Wireless or mobile phones are rapidly replacing wired (landline) telephones, both of which are connected to a large-scale telecommunications network. They are used for voice communication as well as data communications. So also are the computers connected to a different network that is used for data and information processing. Computers are used to generate, transmit, and receive an enormous amount of information through the Internet and will be used more extensively over the same network, in the coming years for voice communications also. This technology is known as *voice over Internet protocol* (VoIP) or *Internet telephony*. At present we can transmit and receive a limited amount of text, graphics, pictures, and video images from

Introduction to Digital Signal Processing and Filter Design, by B. A. Shenoi

mobile phones, besides voice, music, and other audio signals—all of which are classified as multimedia—because of limited hardware in the mobile phones and not the software that has already been developed. However, the computers can be used to carry out the same functions more efficiently with greater memory and large bandwidth. We see a seamless integration of wireless telephones and computers already developing in the market at present. The new technologies being used in the abovementioned applications are known by such terms as CDMA, TDMA,[1] spread spectrum, echo cancellation, channel coding, adaptive equalization, ADPCM coding, and data encryption and decryption, some of which are used in the software to be introduced in the third-generation (G3) mobile phones.

2. *Speech Processing*. The quality of speech transmission in real time over telecommunications networks from wired (landline) telephones or wireless (cellular) telephones is very high. Speech recognition, speech synthesis, speaker verification, speech enhancement, text-to-speech translation, and speech-to-text dictation are some of the other applications of speech processing.

3. *Consumer Electronics*. We have already mentioned cellular or mobile phones. Then we have HDTV, digital cameras, digital phones, answering machines, fax and modems, music synthesizers, recording and mixing of music signals to produce CD and DVDs. Surround-sound entertainment systems including CD and DVD players, laser printers, copying machines, and scanners are found in many homes. But the TV set, PC, telephones, CD-DVD players, and scanners are present in our homes as separate systems. However, the TV set can be used to read email and access the Internet just like the PC; the PC can be used to tune and view TV channels, and record and play music as well as data on CD-DVD in addition to their use to make telephone calls on VoIP. This trend toward the development of fewer systems with multiple applications is expected to accelerate in the near future.

4. *Biomedical Systems*. The variety of machines used in hospitals and biomedical applications is staggering. Included are X-ray machines, MRI, PET scanning, bone scanning, CT scanning, ultrasound imaging, fetal monitoring, patient monitoring, and ECG and EEC mapping. Another example of advanced digital signal processing is found in hearing aids and cardiac pacemakers.

5. *Image Processing*. Image enhancement, image restoration, image understanding, computer vision, radar and sonar processing, geophysical and seismic data processing, remote sensing, and weather monitoring are some of the applications of image processing. Reconstruction of two-dimensional (2D) images from several pictures taken at different angles and three-dimensional (3D) images from several contiguous slices has been used in many applications.

6. *Military Electronics*. The applications of digital signal processing in military and defense electronics systems use very advanced techniques. Some of the applications are GPS and navigation, radar and sonar image processing, detection

[1]Code- and time-division multiple access. In the following sections we will mention several technical terms and well-known acronyms without any explanation or definition. A few of them will be described in detail in the remaining part of this book.

and tracking of targets, missile guidance, secure communications, jamming and countermeasures, remote control of surveillance aircraft, and electronic warfare.

7. *Aerospace and Automotive Electronics*. Applications include control of aircraft and automotive engines, monitoring and control of flying performance of aircraft, navigation and communications, vibration analysis and antiskid control of cars, control of brakes in aircrafts, control of suspension, and riding comfort of cars.

8. *Industrial Applications*. Numerical control, robotics, control of engines and motors, manufacturing automation, security access, and videoconferencing are a few of the industrial applications.

Obviously there is some overlap among these applications in different devices and systems. It is also true that a few basic operations are common in all the applications and systems, and these basic operations will be discussed in the following chapters. The list of applications given above is not exhaustive. A few applications are described in further detail in [1]. Needless to say, the number of new applications and improvements to the existing applications will continue to grow at a very rapid rate in the near future.

1.3 DISCRETE-TIME SIGNALS

A signal defines the variation of some physical quantity as a function of one or more independent variables, and this variation contains information that is of interest to us. For example, a continuous-time signal that is periodic contains the values of its fundamental frequency and the harmonics contained in it, as well as the amplitudes and phase angles of the individual harmonics. The purpose of signal processing is to modify the given signal such that the quality of information is improved in some well-defined meaning. For example, in mixing consoles for recording music, the frequency responses of different filters are adjusted so that the overall quality of the audio signal (music) offers as high fidelity as possible. Note that the contents of a telephone directory or the encyclopedia downloaded from an Internet site contains a lot of useful information but the contents do not constitute a signal according to the definition above. It is the functional relationship between the function and the independent variable that allows us to derive methods for modeling the signals and find the output of the systems when they are excited by the input signals. This also leads us to develop methods for designing these systems such that the information contained in the input signals is improved.

We define a *continuous-time signal* as a function of an independent variable that is continuous. A one-dimensional continuous-time signal $f(t)$ is expressed as a function of time that varies continuously from $-\infty$ to ∞. But it may be a function of other variables such as temperature, pressure, or elevation; yet we will denote them as continuous-time signals, in which time is continuous but the signal may have discontinuities at some values of time. The signal may be a

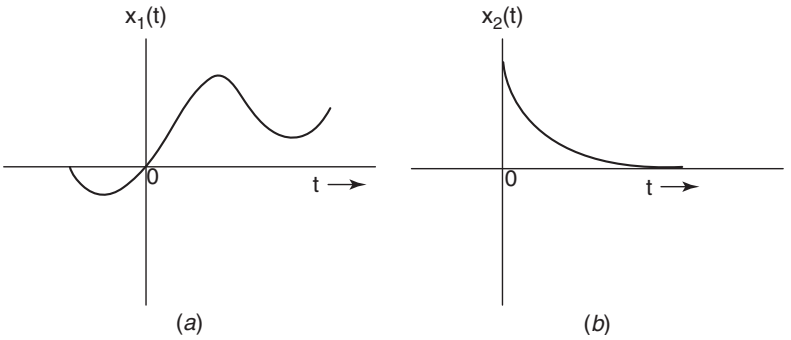


Figure 1.1 Two samples of continuous-time signals.

real- or complex-valued function of time. We can also define a continuous-time signal as a mapping of the set of all values of time to a set of corresponding values of the functions that are subject to certain properties. Since the function is well defined for all values of time in $-\infty$ to ∞, it is differentiable at all values of the independent variable t (except perhaps at a finite number of values). Two examples of continuous-time functions are shown in Figure 1.1.

A *discrete-time signal* is a function that is defined only at discrete instants of time and undefined at all other values of time. Although a discrete-time function may be defined at arbitrary values of time in the interval $-\infty$ to ∞, we will consider only a function defined at equal intervals of time and defined at $t = nT$, where T is a fixed interval in seconds known as the *sampling period* and n is an integer variable defined over $-\infty$ to ∞. If we choose to sample $f(t)$ at equal intervals of T seconds, we generate $f(nT) = f(t)|_{t=nT}$ as a sequence of numbers. Since T is fixed, $f(nT)$ is a function of only the integer variable n and hence can be considered as a function of n or expressed as $f(n)$. The continuous-time function $f(t)$ and the discrete-time function $f(n)$ are plotted in Figure 1.2.

In this book, we will denote a discrete-time (DT) function as a DT sequence, DT signal, or a DT series. So a DT function is a mapping of a set of all integers to a set of values of the functions that may be real-valued or complex-valued. Values of both $f(t)$ and $f(n)$ are assumed to be continuous, taking any value in a continuous range; hence can have a value even with an infinite number of digits, for example, $f(3) = 0.4\sqrt{2}$ in Figure 1.2.

A zero-order hold (ZOH) circuit is used to sample a continuous signal $f(t)$ with a sampling period T and hold the sampled values for one period before the next sampling takes place. The DT signal so generated by the ZOH is shown in Figure 1.3, in which the value of the sample value during each period of sampling is a constant; the sample can assume any continuous value. The signals of this type are known as *sampled-data signals*, and they are used extensively in sampled-data control systems and switched-capacitor filters. However, the duration of time over which the samples are held constant may be a very small fraction of the sampling period in these systems. When the value of a sample

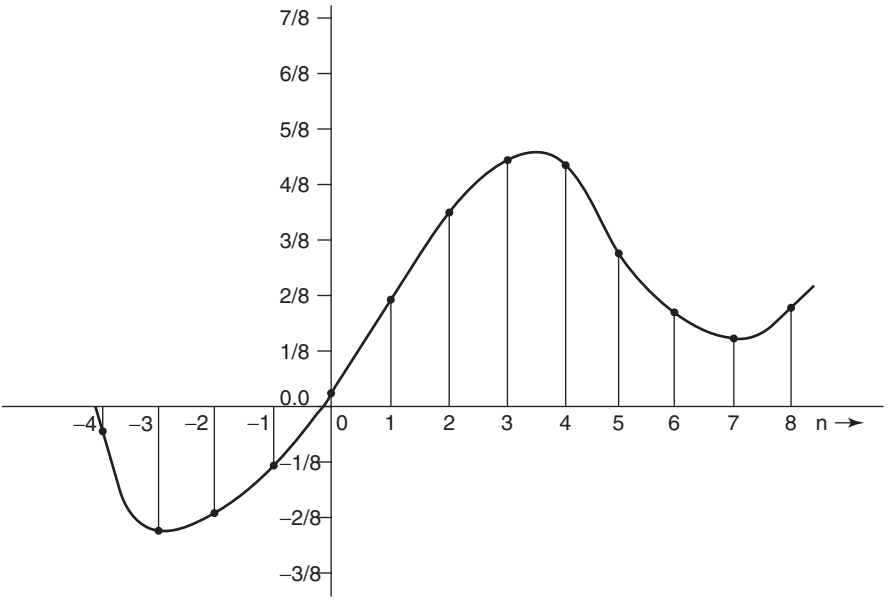


Figure 1.2 The continuous-time function $f(t)$ and the discrete-time function $f(n)$.

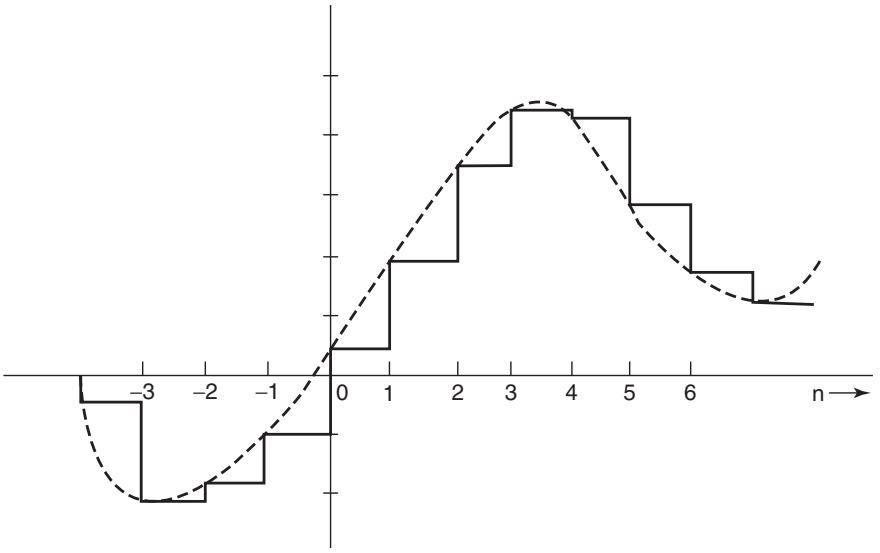


Figure 1.3 Sampled data signal.

is held constant during a period T (or a fraction of T) by the ZOH circuit as its output, that signal can be converted to a value by a quantizer circuit, with finite levels of value as determined by the binary form of representation. Such a process is called *binary coding* or *quantization*. A This process is discussed in full detail in Chapter 7. The precision with which the values are represented is determined by the number of bits (binary digits) used to represent each value. If, for example, we select 3 bits, to express their values using a method known as "signed magnitude fixed-point binary number representation" and one more bit to denote positive or negative values, we have the finite number of values, represented in binary form and in their equivalent decimal form. Note that a 4-bit binary form can represent values between $-\frac{7}{8}$ and $\frac{7}{8}$ at 15 distinct levels as shown in Table 1.1. So a value of $f(n)$ at the output of the ZOH, which lies between these distinct levels, is rounded or truncated by the quantizer according to some rules and the output of the quantizer when coded to its equivalent binary representation, is called the *digital signal*. Although there is a difference between the discrete-time signal and digital signal, in the next few chapters we assume that the signals are discrete-time signals and in Chapter 7, we consider the effect of quantizing the signals to their binary form, on the frequency response of the

TABLE 1.1 4 Bit Binary Numbers and their Decimal Equivalents

Binary Form	Decimal Value
$0_{\triangle}111$	$\frac{7}{8} = 0.875$
$0_{\triangle}110$	$\frac{6}{8} = 0.750$
$0_{\triangle}101$	$\frac{5}{8} = 0.625$
$0_{\triangle}100$	$\frac{4}{8} = 0.500$
$0_{\triangle}011$	$\frac{3}{8} = 0.375$
$0_{\triangle}010$	$\frac{2}{8} = 0.250$
$0_{\triangle}001$	$\frac{1}{8} = 0.125$
$0_{\triangle}000$	$0.0 = 0.000$
$1_{\triangle}000$	$-0.0 = -0.000$
$1_{\triangle}001$	$-\frac{1}{8} = -0.125$
$1_{\triangle}010$	$-\frac{2}{8} = -0.250$
$1_{\triangle}011$	$-\frac{3}{8} = -0.375$
$1_{\triangle}100$	$-\frac{4}{8} = -0.500$
$1_{\triangle}101$	$-\frac{5}{8} = -0.625$
$1_{\triangle}110$	$-\frac{6}{8} = -0.750$
$1_{\triangle}111$	$-\frac{7}{8} = -0.875$

filters. However, we use the terms *digital filter* and *discrete-time system* interchangeably in this book. Continuous-time signals and systems are also called *analog signals* and *analog systems*, respectively. A system that contains both the ZOH circuit and the quantizer is called an *analog-to digital converter* (ADC), which will be discussed in more detail in Chapter 7.

Consider an analog signal as shown by the solid line in Figure 1.2. When it is sampled, let us assume that the discrete-time sequence has values as listed in the second column of Table 1.2. They are expressed in only six significant decimal digits and their values, when truncated to four digits, are shown in the third column. When these values are quantized by the quantizer with four binary digits (bits), the decimal values are truncated to the values at the finite discrete levels. In decimal number notation, the values are listed in the fourth column, and in binary number notation, they are listed in the fifth column of Table 1.2. The binary values of $f(n)$ listed in the third column of Table 1.2 are plotted in Figure 1.4.

A continuous-time signal $f(t)$ or a discrete-time signal $f(n)$ expresses the variation of a physical quantity as a function of one variable. A black-and-white photograph can be considered as a two-dimensional signal $f(m, r)$, when the intensity of the dots making up the picture is measured along the horizontal axis (x axis; abscissa) and the vertical axis (y axis; ordinate) of the picture plane and are expressed as a function of two integer variables m and r, respectively. We can consider the signal $f(m, r)$ as the discretized form of a two-dimensional signal $f(x, y)$, where x and y are the continuous spatial variables for the horizontal and vertical coordinates of the picture and T_1 and T_2 are the sampling

TABLE 1.2 Numbers in Decimal and Binary Forms

n	Decimal Values of $f(n)$	Values of $f(n)$ Truncated to Four Digits	Quantized Values of $f(n)$	Binary Number Form
−4	−0.054307	−0.0543	0.000	$1_{\triangle}000$
−3	−0.253287	−0.2532	−0.250	$1_{\triangle}010$
−2	−0.236654	−0.2366	−0.125	$1_{\triangle}001$
−1	−0.125101	−0.1251	−0.125	$1_{\triangle}001$
0	0.522312	0.5223	0.000	$0_{\triangle}000$
1	0.246210	0.2462	0.125	$0_{\triangle}001$
2	0.387508	0.3875	0.375	$0_{\triangle}011$
3	0.554090	0.5540	0.500	$0_{\triangle}100$
4	0.521112	0.5211	0.500	$0_{\triangle}100$
5	0.275432	0.2754	0.250	$0_{\triangle}010$
6	0.194501	0.1945	0.125	$0_{\triangle}001$
7	0.168887	0.1687	0.125	$0_{\triangle}001$
8	0.217588	0.2175	0.125	$0_{\triangle}001$

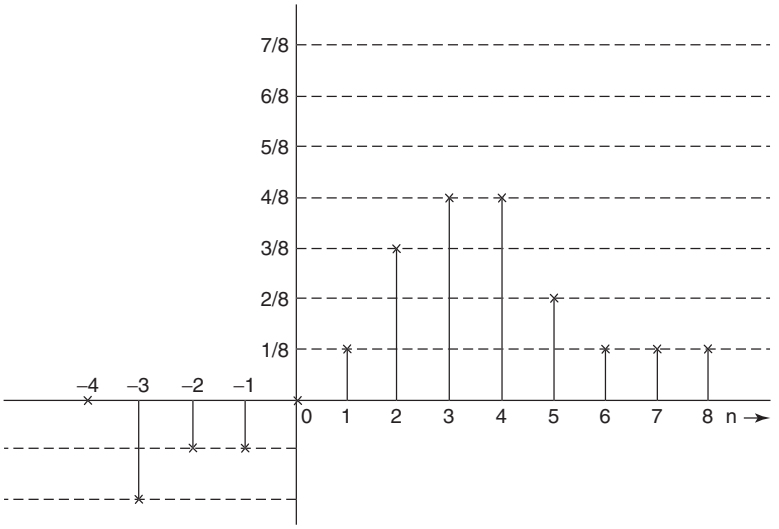


Figure 1.4 Binary values in Table 1.2, after truncation of $f(n)$ to 4 bits.

periods (measured in meters) along the x and y axes, respectively. In other words, $f(x, y)|_{x=mT_1, y=rT_2} = f(m, r)$.

A black-and-white video signal $f(x, y, t)$ is a 3D function of two spatial coordinates x and y and one temporal coordinate t. When it is discretized, we have a 3D discrete signal $f(m, p, n)$. When a color video signal is to be modeled, it is expressed by a vector of three 3D signals, each representing one of the three primary colors—red, green, and blue—or their equivalent forms of two luminance and one chrominance. So this is an example of multivariable function or a multichannel signal:

$$\mathbf{F}(m, r, n) = \begin{bmatrix} f_r(m, p, n) \\ f_g(m, p, n) \\ f_b(m, p, n) \end{bmatrix} \tag{1.1}$$

1.3.1 Modeling and Properties of Discrete-Time Signals

There are several ways of describing the functional relationship between the integer variable n and the value of the discrete-time signal $f(n)$: (1) to plot the values of $f(n)$ versus n as shown in Figure 1.2, (2) to tabulate their values as shown in Table 1.2, and (3) to define the sequence by expressing the sample values as elements of a set, when the sequence has a finite number of samples.

For example, in a sequence $x_1(n)$ as shown below, the arrow indicates the value of the sample when $n = 0$:

$$x_1(n) = \left\{2 \quad 3 \quad 1.5 \quad \underset{\uparrow}{0.5} \quad -1 \quad 4\right\} \tag{1.2}$$

We denote the DT sequence by $x(n)$ and also the value of a sample of the sequence at a particular value of n by $x(n)$. If a sequence has zero values for $n < 0$, then it is called a *causal sequence*. It is misleading to state that the causal function is a sequence defined for $n \geq 0$, because, strictly speaking, a DT sequence has to be defined for all values of n. Hence it is understood that a causal sequence has zero-valued samples for $-\infty < n < 0$. Similarly, when a function is defined for $N_1 \leq n \leq N_2$, it is understood that the function has zero values for $-\infty < n < N_1$ and $N_2 < n < \infty$. So the sequence $x_1(n)$ in Equation (1.2) has zero values for $2 < n < \infty$ and for $-\infty < n < -3$. The discrete-time sequence $x_2(n)$ given below is a causal sequence. In this form for representing $x_2(n)$, it is implied that $x_2(n) = 0$ for $-\infty < n < 0$ and also for $4 < n < \infty$:

$$x_2(n) = \left\{ \underset{\uparrow}{1} \quad -2 \quad 0.4 \quad 0.3 \quad 0.4 \quad 0 \quad 0 \quad 0 \right\} \tag{1.3}$$

The length of a finite sequence is often defined by other authors as the number of samples, which becomes a little ambiguous in the case of a sequence like $x_2(n)$ given above. The function $x_2(n)$ is the same as $x_3(n)$ given below:

$$x_3(n) = \left\{ \underset{\uparrow}{1} \quad -2 \quad 0.4 \quad 0.3 \quad 0.4 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right\} \tag{1.4}$$

But does it have more samples? So the length of the sequence $x_3(n)$ would be different from the length of $x_2(n)$ according to the definition above. When a sequence such as $x_4(n)$ given below is considered, the definition again gives an ambiguous answer:

$$x_4(n) = \left\{ \underset{\uparrow}{0} \quad 0 \quad 0.4 \quad 0.3 \quad 0.4 \right\} \tag{1.5}$$

The definition for the length of a DT sequence would be refined when we define the degree (or order) of a polynomial in z^{-1} to express the z transform of a DT sequence, in the next chapter.

To model the discrete-time signals mathematically, instead of listing their values as shown above or plotting as shown in Figure 1.2, we introduce some basic DT functions as follows.

1.3.2 Unit Pulse Function

The unit pulse function $\delta(n)$ is defined by

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{1.6}$$

and it is plotted in Figure 1.5a. It is often called the *unit sample function* and also the *unit impulse function*. But note that the function $\delta(n)$ has a finite numerical

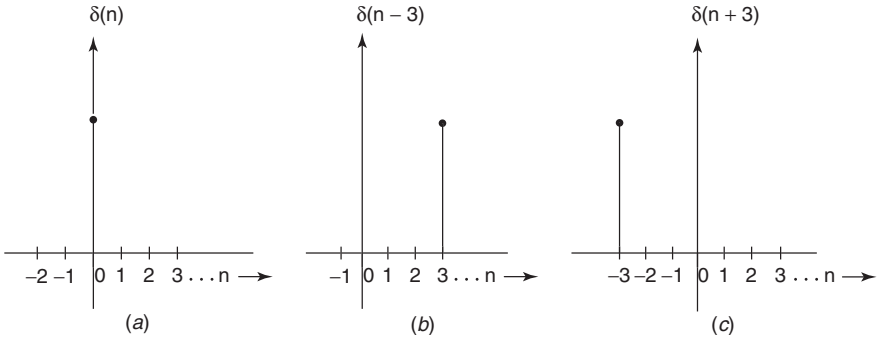


Figure 1.5 Unit pulse functions $\delta(n)$, $\delta(n-3)$, and $\delta(n+3)$.

value of one at $n = 0$ and zero at all other values of integer n, whereas the unit impulse function $\delta(t)$ is defined entirely in a different way.

When the unit pulse function is delayed by k samples, it is described by

$$\delta(n-k) = \begin{cases} 1 & n = k \\ 0 & n \neq k \end{cases} \tag{1.7}$$

and it is plotted in Figure 1.5b for $k = 3$. When $\delta(n)$ is advanced by $k = 3$, we get $\delta(n+k)$, and it is plotted in Figure 1.5c.

1.3.3 Constant Sequence

This sequence $x(n)$ has a constant value for all n and is therefore defined by $x(n) = K; -\infty < n < \infty$.

1.3.4 Unit Step Function

The unit step function $u(n)$ is defined by

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases} \tag{1.8}$$

and it is plotted in Figure 1.6a.

When the unit step function is delayed by k samples, where k is a positive integer, we have

$$u(n-k) = \begin{cases} 1 & n \geq k \\ 0 & n < k \end{cases} \tag{1.9}$$

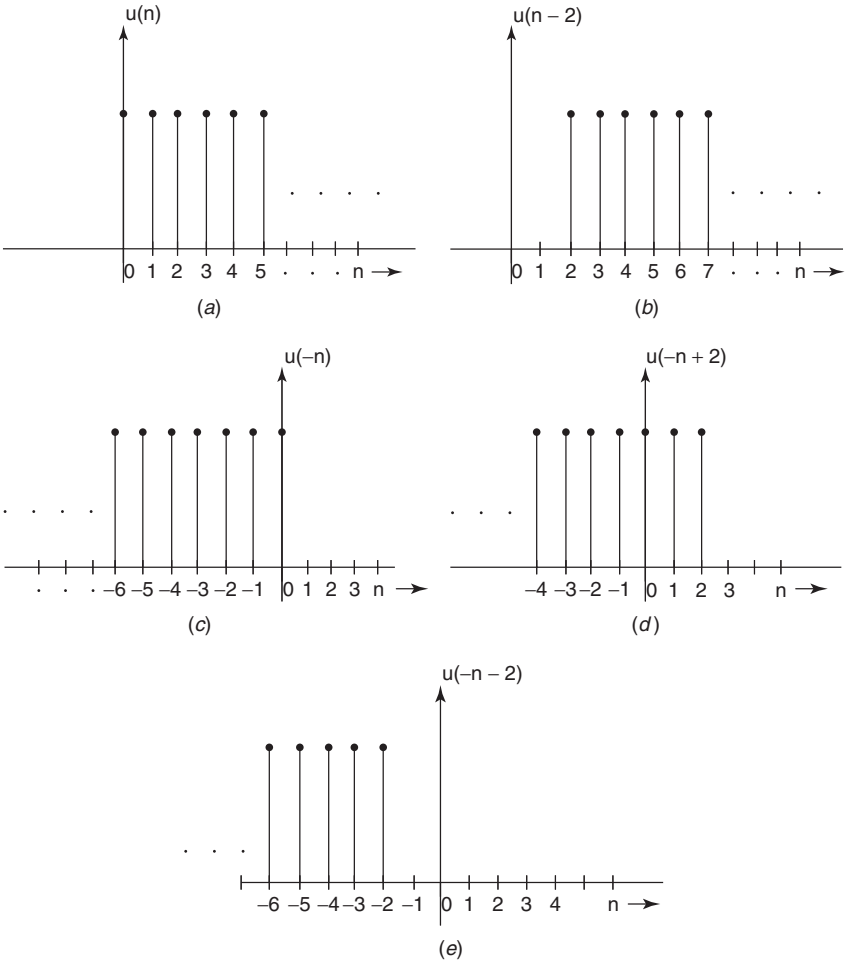


Figure 1.6 Unit step functions.

The sequence $u(n+k)$ is obtained when $u(n)$ is advanced by k samples. It is defined by

$$u(n+k) = \begin{cases} 1 & n \geq -k \\ 0 & n < -k \end{cases} \tag{1.10}$$

We also define the function $u(-n)$, obtained from the time reversal of $u(n)$, as a sequence that is zero for $n > 0$. The sequences $u(-n+k)$ and $u(-n-k)$, where k is a positive integer, are obtained when $u(-n)$ is delayed by k samples and advanced by k samples, respectively. In other words, $u(-n+k)$ is obtained by

delaying $u(-n)$ when k is positive and obtained by advancing $u(-n)$ when k is a negative integer. Note that the effect on $u(-n-k)$ is opposite that on $u(n-k)$, when k is assumed to take positive and negative values. These functions are shown in Figure 1.6, where $k = 2$. In a strict sense, all of these functions are defined implicitly for $-\infty < n < \infty$.

1.3.5 Real Exponential Function

The real exponential function is defined by

$$x(n) = a^n; \qquad -\infty < n < \infty \tag{1.11}$$

where a is real constant. If a is a complex constant, it becomes the complex exponential sequence. The real exponential sequence or the complex exponential sequence may also be defined by a more general relationship of the form

$$x(n) = \begin{cases} a^n & k \le n < \infty \\ b^n & -\infty < n < k \end{cases} \tag{1.12}$$

A special discrete-time sequence that we often use is the function defined for $n \ge 0$:

$$x(n) = a^n u(n) \tag{1.13}$$

An example of $x_1(n) = (0.8)^n u(n)$ is plotted in Figure 1.7a. The function $x_2(n) = x_1(n-3) = (0.8)^{(n-3)} u(n-3)$ is obtained when $x_1(n)$ is delayed by three samples. It is plotted in Figure 1.7b. But the function $x_3(n) = (0.8)^n u(n-3)$ is obtained by chopping off the first three samples of $x_1(n) = (0.8)^n u(n)$, and as shown in Figure 1.7c, it is different from $x_2(n)$.

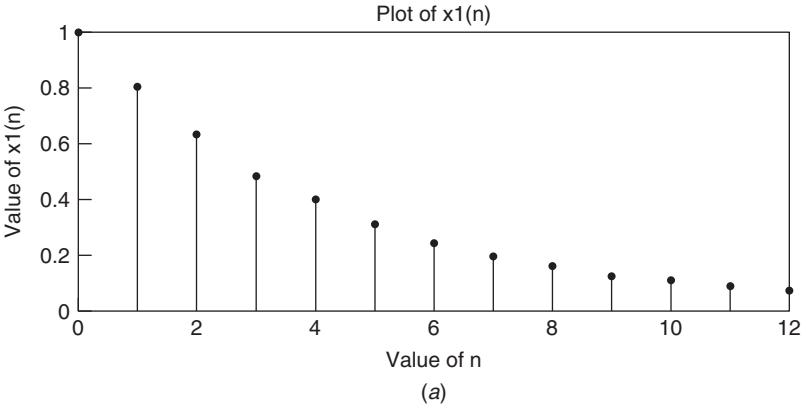
1.3.6 Complex Exponential Function

The complex exponential sequence is a function that is complex-valued as a function of n. The most general form of such a function is given by

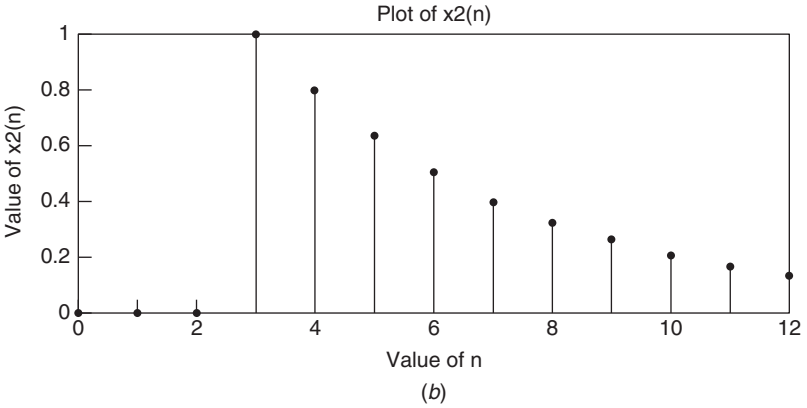
$$x(n) = A\alpha^n, \qquad -\infty < n < \infty \tag{1.14}$$

where both A and α are complex numbers. If we let $A = |A|\, e^{j\phi}$ and $\alpha = e^{(\sigma_0 + j\omega_0)}$, where σ_0, ω_0, and ϕ are real numbers, the sequence can be expanded to the form

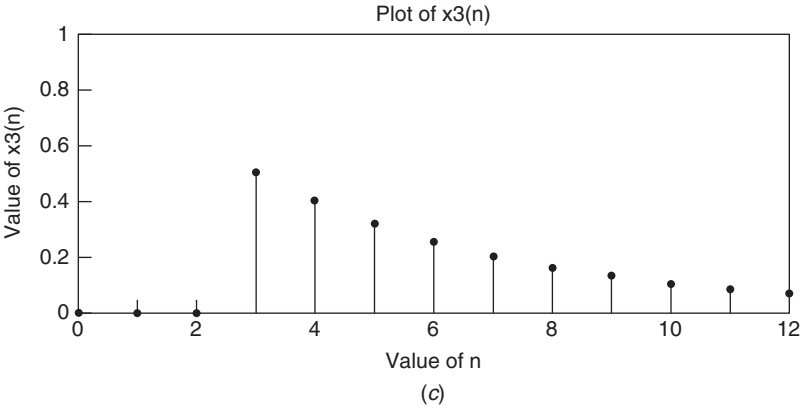
$$\begin{aligned} x(n) &= |A|\, e^{j\phi} e^{(\sigma_0 + j\omega_0)n} \\ &= |A|\, e^{\sigma_0 n} e^{j(\omega_0 n + \phi)} \\ &= |A|\, e^{\sigma_0 n} \cos(\omega_0 n + \phi) + j\, |A|\, e^{\sigma_0 n} \sin(\omega_0 n + \phi) \\ &= x_{\text{re}}(n) + j x_{\text{im}}(n) \end{aligned} \tag{1.15}$$



(*a*)



(*b*)



(*c*)

Figure 1.7 Plots of $x_1(n)$, $x_2(n)$, and $x_3(n)$.

When $\sigma_0 = 0$, the real and imaginary parts of this complex exponential sequence are $|A|\cos(\omega_0 n + \phi)$ and $|A|\sin(\omega_0 n + \phi)$, respectively, and are real sinusoidal sequences with an amplitude equal to $|A|$. When $\sigma_0 > 0$, the two sequences increase as $n \to \infty$ and decrease when $\sigma_0 < 0$ as $n \to \infty$. When $\omega_0 = \phi = 0$, the sequence reduces to the real exponential sequence $|A|\, e^{\sigma_0 n}$.

1.3.7 Properties of $\cos(\omega_0 n)$

When $A = 1$, and $\sigma_0 = \phi = 0$, we get $x(n) = e^{j\omega_0 n} = \cos(\omega_0 n) + j\sin(\omega_0 n)$. This function has some interesting properties, when compared with the continuous-time function $e^{j\omega_0' t}$ and they are described below.

First we point out that ω_0 in $x(n) = e^{j\omega_0 n}$ is a frequency normalized by $f_s = 1/T$, where f_s is the sampling frequency in hertz and T is the sampling period in seconds, specifically, $\omega_0 = 2\pi f_0'/f_s = \omega_0' T$, where $\omega_0' = 2\pi f_0'$ is the actual real frequency in radians per second and f_0' is the actual frequency in hertz. Therefore the unit of the normalized frequency ω_0 is radians. It is common practice in the literature on discrete-time systems to choose ω as the normalized frequency variable, and we follow that notation in the following chapters; here we denote ω_0 as a constant in radians. We will discuss this normalized frequency again in a later chapter.

Property 1.1 In the complex exponential function $x(n) = e^{j\omega_0 n}$, two frequencies separated by an integer multiple of 2π are indistinguishable from each other. In other words, it is easily seen that $e^{j\omega_0 n} = e^{j(\omega_0 n + 2\pi r)}$. The real part and the imaginary part of the function $x(n) = e^{j\omega_0 n}$, which are sinusoidal functions, also exhibit this property. As an example, we have plotted $x_1(n) = \cos(0.3\pi n)$ and $x_2(n) = \cos(0.3\pi + 4\pi)n$ in Figure 1.8. In contrast, we know that two continuous-time functions $x_1(t) = e^{j\omega_1 t}$ and $x_2(t) = e^{j\omega_2 t}$or their real and imaginary parts are different if ω_1 and ω_2 are different. They are different even if they are separated by integer multiples of 2π. From the property $e^{j\omega_0 n} = e^{j(\omega_0 n + 2\pi r)}$ above, we arrive at another important result, namely, that the output of a discrete-time system has the same value when these two functions are excited by the complex exponential functions $e^{j\omega_0 n}$ or $e^{\,j(\omega_0 n + 2\pi r)}$. We will show in Chapter 3 that this is true for all frequencies separated by integer multiples of 2π, and therefore the frequency response of a DT system is periodic in ω.

Property 1.2 Another important property of the sequence $e^{j\omega_0 n}$ is that it is periodic in n. A discrete-time function $x(n)$ is defined to be periodic if there exists an integer N such that $x(n + rN) = x(n)$, where r is any arbitrary integer and N is the period of the periodic sequence. To find the value for N such that $e^{j\omega_0 n}$ is periodic, we equate $e^{j\omega_0 n}$ to $e^{j\omega_0 (n + rN)}$. Therefore $e^{j\omega_0 n} = e^{j\omega_0 n} e^{j\omega_0 rN}$, which condition is satisfied when $e^{j\omega_0 rN} = 1$, that is, when $\omega_0 N = 2\pi K$, where

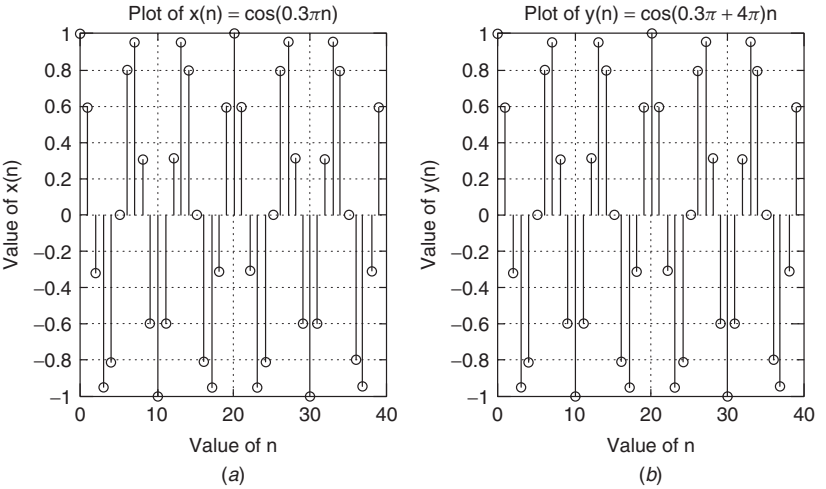


Figure 1.8 Plots of $\cos(0.3\pi n)$ and $\cos(0.3\pi + 4\pi)n$.

K is any arbitrary integer. This condition is satisfied by the following equation:

$$\frac{\omega_0}{2\pi} = \frac{K}{N} \tag{1.16}$$

In words, this means that the ratio of the given normalized frequency ω_0 and 2π must be a rational number. The period of the sequence N is given by

$$N = \frac{2\pi K}{\omega_0} \tag{1.17}$$

When this condition is satisfied by the smallest integer K, the corresponding value of N gives the fundamental period of the periodic sequence, and integer multiples of this frequency are the harmonic frequencies.

Example 1.1

Consider a sequence $x(n) = \cos(0.3\pi n)$. In this case $\omega_0 = 0.3\pi$ and $\omega_0/2\pi = 0.3\pi/2\pi = \frac{3}{20}$. Therefore the sequence is periodic and its period N is 20 samples. This periodicity is noticed in Figure 1.8a and also in Figure 1.8b.

Consider another sequence $x(n) = \cos(0.5n)$, in which case $\omega_0 = 0.5$. Therefore $\omega_0/2\pi = 0.5/2\pi = 1/4\pi$, which is not a rational number. Hence this is not a periodic sequence.

When the given sequence is the sum of several complex exponential functions, each of which is periodic with different periods, it is still periodic. We consider an example to illustrate the method to find the fundamental period in this case.

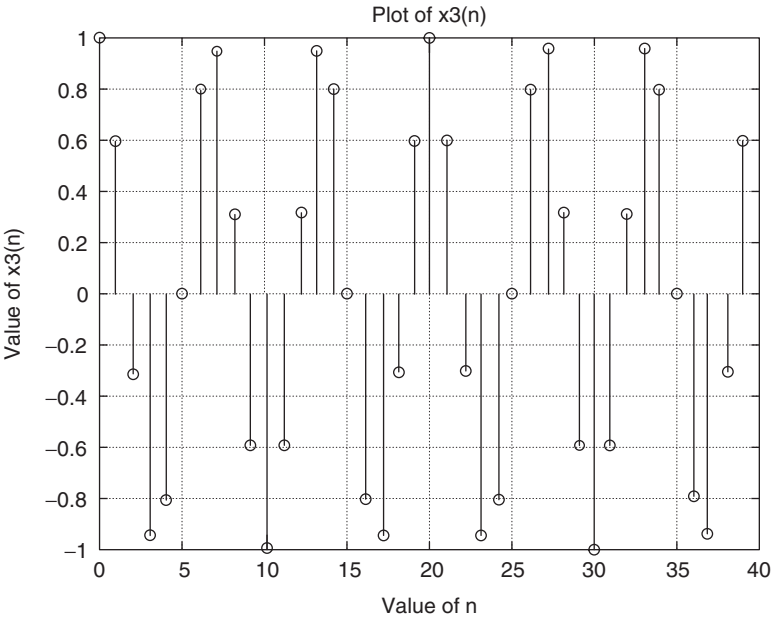


Figure 1.9 Plot of $x_3(n)$.

Suppose $x_3(n) = \cos(0.2\pi n) + \cos(0.5\pi n) + \cos(0.6\pi n)$. Its fundamental period N must satisfy the condition

$$N = \frac{2\pi K_1}{0.2\pi} = \frac{2\pi K_2}{0.5\pi} = \frac{2\pi K_3}{0.6\pi} \tag{1.18}$$

$$= 10K_1 = 4K_2 = \frac{10K_3}{3} \tag{1.19}$$

where K_1, K_2, and K_3 and N are integers. The value of N that satisfies this condition is 20 when $K_1 = 2$, $K_2 = 5$, and $K_3 = 6$. So $N = 20$ is the fundamental period of $x_3(n)$. The sequence $x_3(n)$ plotted in Figure 1.9 for $0 \leq n \leq 40$ shows that it is periodic with a period of 20 samples.

Property 1.3 We have already observed that the frequencies at ω_0 and at $\omega_0 + 2\pi$ are the same, and hence the frequency of oscillation are the same. But consider the frequency of oscillation as ω_0 changes between 0 and 2π. It is found that the frequency of oscillation of the sinusoidal sequence $\cos(\omega_0 n)$ increases as ω_0 increases from 0 to π and the frequency of oscillation decreases as ω_0 increases from π to 2π. Therefore the highest frequency of oscillation of a discrete-time sequence $\cos(\omega_0 n)$ occurs when $\omega_0 = \pm\pi$. When the normalized frequency $\omega_0 = 2\pi f_0'/f_s$ attains the value of π, the value of $f_0' = f_s/2$. So the highest frequency of oscillation occurs when it is equal to half the sampling

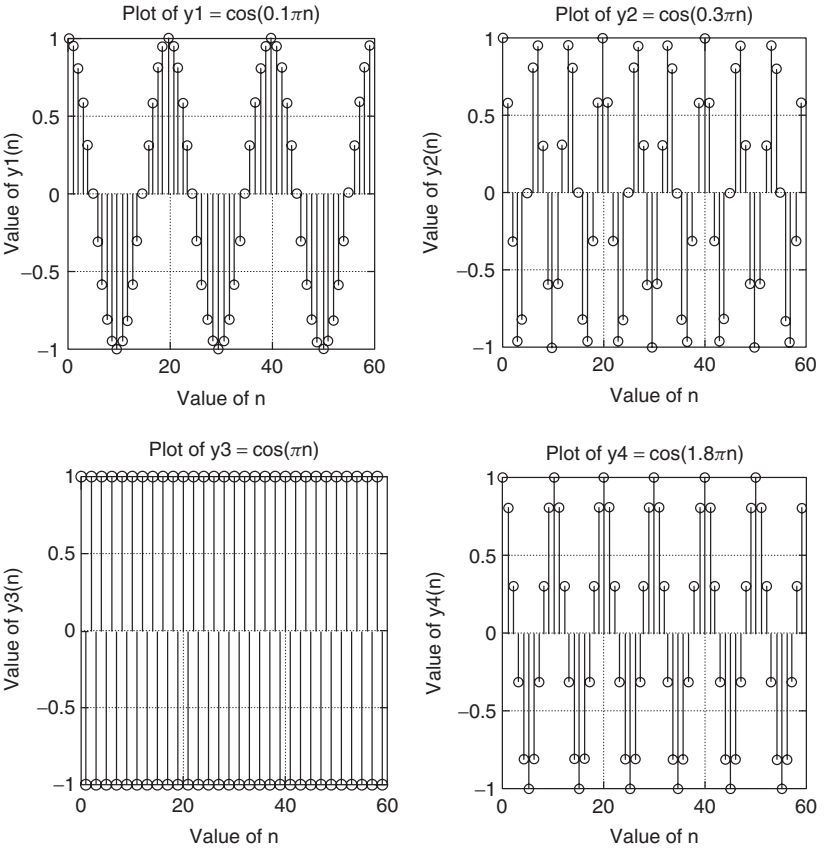


Figure 1.10 Plot of $\cos(\omega_0 n)$ for different values of ω_0 between 0 and 2π.

frequency. In Figure 1.10 we have plotted the DT sequences as ω_0 attains a few values between 0 and 2π, to illustrate this property. We will elaborate on this property in later chapters of the book.

Since frequencies separated by 2π are the same, as ω_0 increases from 2π to 3π, the frequency of oscillation increases in the same manner as the frequency of oscillation when it increases from 0 to π. As an example, we see that the frequency of $v_0(n) = \cos(0.1\pi n)$ is the same as that of $v_1(n) = \cos(2.1\pi n)$. It is interesting to note that $v_2(n) = \cos(1.9\pi n)$ also has the same frequency of oscillation as $v_1(n)$ because

$$v_2(n) = \cos(1.9\pi n) = \cos(2\pi - 0.1\pi n) \tag{1.20}$$

$$= \cos(2\pi n)\cos(0.1\pi n) + \sin(2\pi n)\sin(0.1\pi n) \tag{1.21}$$

$$= \cos(0.1\pi n)$$

$$= v_0(n)$$

$$v_1(n) = \cos(2.1\pi n) = \cos(2\pi n + 0.1\pi n) \tag{1.22}$$

$$= \cos(2\pi n)\cos(0.1\pi n) - \sin(2\pi n)\sin(0.1\pi n) \tag{1.23}$$

$$= \cos(0.1n)$$

$$= v_0(n)$$

We have plotted the sequences $v_1(n)$ and $v_2(n)$ in Figure 1.11, to verify this property.

Remember that in Chapter 3, we will use the term "folding" to describe new implications of this property. We will also show in Chapter 3 that a large class of discrete-time signals can be expressed as the weighted sum of exponential sequences of the form $e^{j\omega_0 n}$, and such a model leads us to derive some powerful analytical techniques of digital signal processing.

We have described several ways of characterizing the DT sequences in this chapter. Using the unit sample function and the unit step function, we can express the DT sequences in other ways as shown below.

For example, $\delta(n) = u(n) - u(n-1)$ and $u(n) = \sum_{m=-\infty}^{m=n} \delta(m)$. A mathematical way of modeling a sequence

$$x(n) = \left\{2 \quad 3 \quad 1.5 \quad \underset{\uparrow}{0.5} \quad -1 \quad 4\right\} \tag{1.24}$$

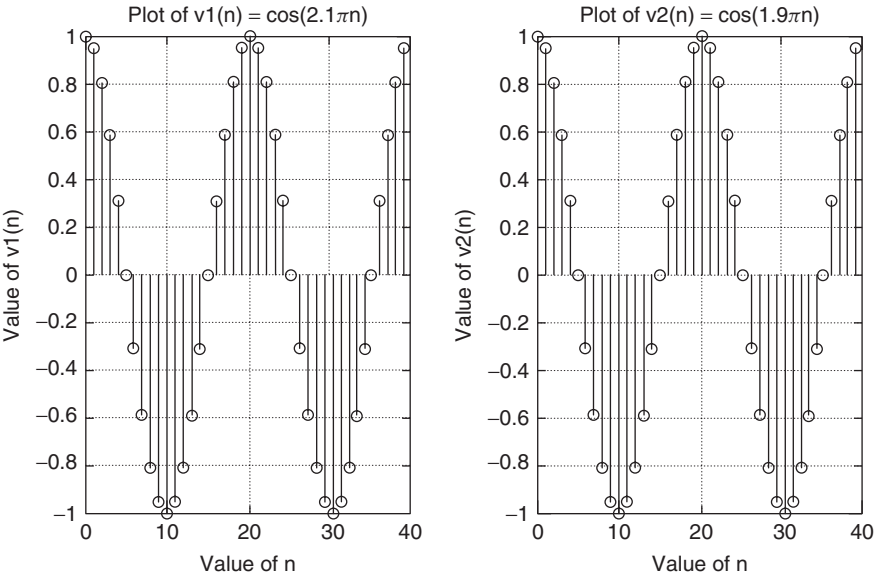


Figure 1.11 Plots of $\cos(2.1\pi n)$ and $\cos(1.9\pi n)$.

is the weighted sum of shifted unit sample functions, as given by

$$x(n) = 2\delta(n+3) + 3\delta(n+2) + 1.5\delta(n+1) + 0.5\delta(n) - \delta(n-1) + 4\delta(n-2) \tag{1.25}$$

If the sequence is given in an analytic form $x(n) = a^n u(n)$, it can also be expressed as the weighted sum of impulse functions:

$$x(n) = \sum_{m=0}^{\infty} x(m)\delta(n-m) = \sum_{m=0}^{\infty} a^m \delta(n-m) \tag{1.26}$$

In the next chapter, we will introduce a transform known as the *z transform*, which will be used to model the DT sequences in additional forms. We will show that this model given by (1.26) is very useful in deriving the z transform and in analyzing the performance of discrete-time systems.

1.4 HISTORY OF FILTER DESIGN

Filtering is the most common form of signal processing used in all the applications mentioned in Section 1.2, to remove the frequencies in certain parts and to improve the magnitude, phase, or group delay in some other part(s) of the spectrum of a signal. The vast literature on filters consists of two parts: (1) the theory of approximation to derive the transfer function of the filter such that the magnitude, phase, or group delay approximates the given frequency response specifications and (2) procedures to design the filters using the hardware components. Originally filters were designed using inductors, capacitors, and transformers and were terminated by resistors representing the load and the internal resistance of the source. These were called the *LC* (inductance × capacitance) filters that admirably met the filtering requirements in the telephone networks for many decades of the nineteenth and twentieth centuries. When the vacuum tubes and bipolar junction transistors were developed, the design procedure had to be changed in order to integrate the models for these active devices into the filter circuits, but the mathematical theory of filter approximation was being advanced independently of these devices. In the second half of the twentieth century, operational amplifiers using bipolar transistors were introduced and filters were designed without inductors to realize the transfer functions. The design procedure was much simpler, and device technology also was improved to fabricate resistors in the form of thick-film and later thin-film depositions on ceramic substrates instead of using printed circuit boards. These filters did not use inductors and transformers and were known as *active-RC* (resistance × capacitance) filters. In the second half of the century, switched-capacitor filters were developed, and they are the most common type of filters being used at present for audio applications. These filters contained only capacitors and operational amplifiers using complementary metal oxide semiconductor (CMOS) transistors. They used no resistors and inductors, and the whole circuit was fabricated by the

very large scale integration (VLSI) technology. The analog signals were converted to sampled data signals by these filters and the signal processing was treated as analog signal processing. But later, the signals were transitioned as discrete-time signals, and the theory of discrete-time systems is currently used to analyze and design these filters. Examples of an *LC* filter, an active-*RC* filter, and a switched-capacitor filter that realize a third-order lowpass filter function are shown in Figures 1.12–1.14.

The evolution of digital signal processing has a different history. At the beginning, the development of discrete-time system theory was motivated by a search for numerical techniques to perform integration and interpolation and to solve differential equations. When computers became available, the solution of physical systems modeled by differential equations was implemented by the digital

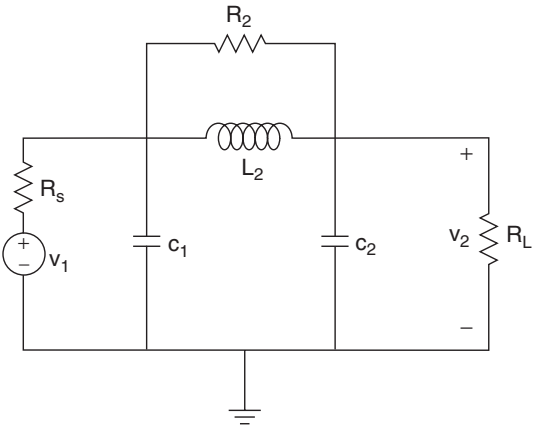


Figure 1.12 A lowpass analog *LC* filter.

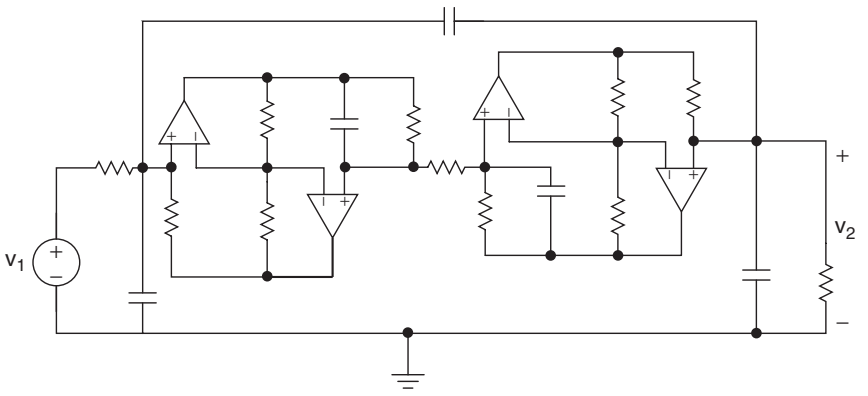


Figure 1.13 An active-*RC* lowpass analog filter.

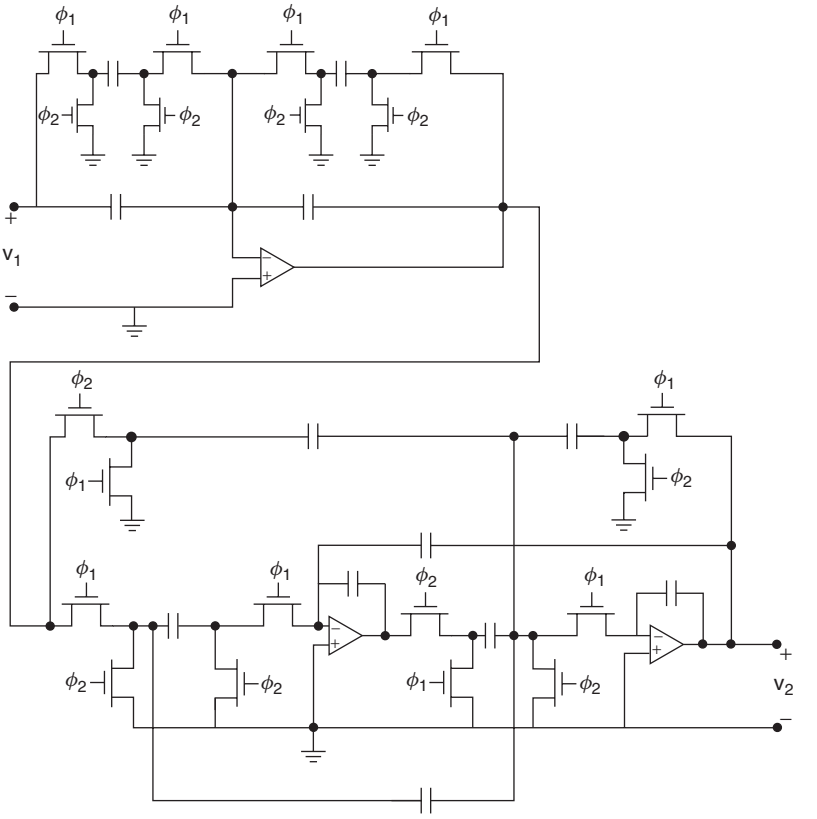


Figure 1.14 A switched-capacitor lowpass (analog) filter.

computers. As the digital computers became more powerful in their computational power, they were heavily used by the oil industry for geologic signal processing and by the telecommunications industry for speech processing. The theory of digital filters matured, and with the advent of more powerful computers built on integrated circuit technology, the theory and applications of digital signal processing has explosively advanced in the last few decades. The two revolutionary results that have formed the foundations of digital signal processing are the Shannon's sampling theorem and the Cooley–Tukey algorithm for fast Fourier transform technique. Both of them will be discussed in great detail in the following chapters. The Shannon's sampling theorem proved that if a continuous-time signal is bandlimited (i.e., if its Fourier transform is zero for frequencies above a maximum frequency f_m) and it is sampled at a rate that is more than twice the maximum frequency f_m in the signal, then no information contained in the analog signal is lost in the sense that the continuous-time signal can be exactly reconstructed from the samples of the discrete-time signal. In practical applications, most of the analog signals are first fed to an analog lowpass filter—known

as the *preconditioning filter* or *antialiasing filter*—such that the output of the lowpass filter attenuates the frequencies considerably beyond a well-chosen frequency so that it can be considered a bandlimited signal. It is this signal that is sampled and converted to a discrete-time signal and coded to a digital signal by the analog-to-digital converter (ADC) that was briefly discussed earlier in this chapter. We consider the discrete-time signal as the input to the digital filter designed in such a way that it improves the information contained in the original analog signal or its equivalent discrete-time signal generated by sampling it. A typical example of a digital lowpass filter is shown in Figure 1.15.

The output of the digital filter is next fed to a digital-to-analog converter (DAC) as shown in Figure 1.17 that also uses a lowpass analog filter that smooths the sampled-data signal from the DAC and is known as the "smoothing filter." Thus we obtain an analog signal $y_d(t)$ at the output of the smoothing filter as shown. It is obvious that compared to the analog filter shown in Figure 1.16, the circuit shown in Figure 1.17 requires considerably more hardware or involves a lot more signal processing in order to filter out the undesirable frequencies from the analog signal $x(t)$ and deliver an output signal $y_d(t)$. It is appropriate to compare these two circuit configurations and determine whether it is possible to get the output $y_d(t)$ that is the same or nearly the same as the output $y(t)$ shown in Figure 1.16; if so, what are the advantages of digital signal processing instead of analog signal processing, even though digital signal processing requires more circuits compared to analog signal processing?

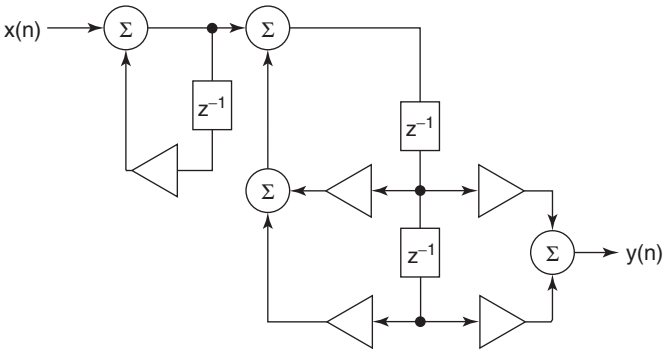


Figure 1.15 A lowpass third-order digital filter.

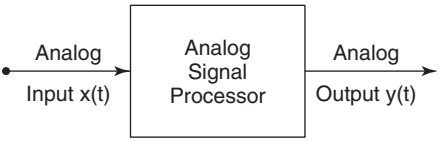


Figure 1.16 Example of an analog signal processing system.