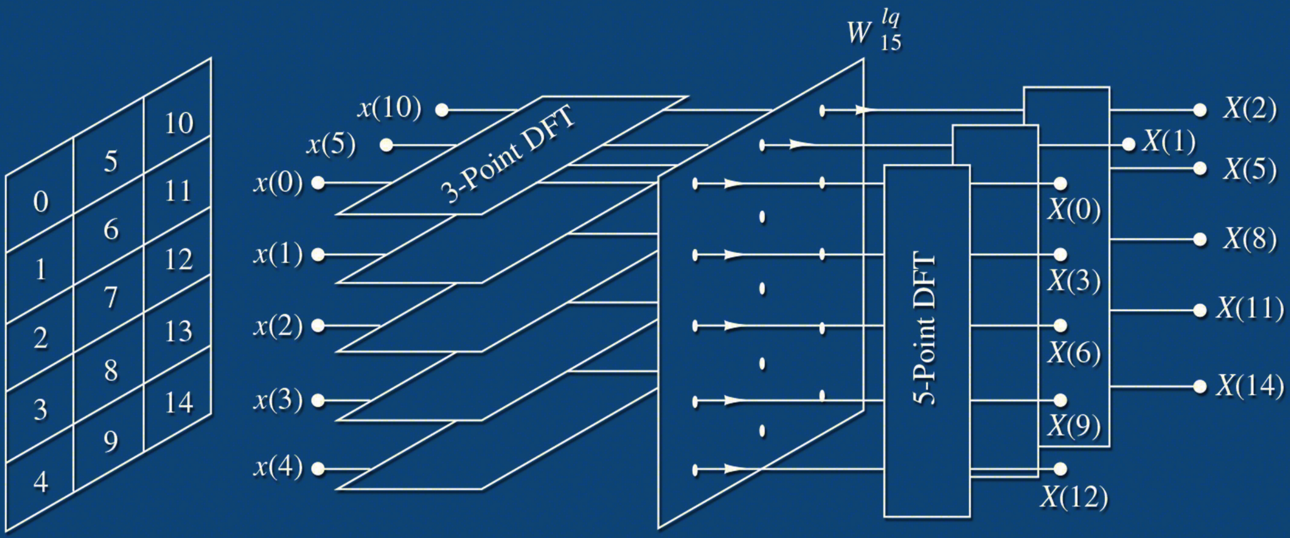
DIGITAL SIGNAL PROCESSING

PRINCIPLES, ALGORITHMS, AND APPLICATIONS

FIFTH EDITION



JOHN G. PROAKIS
DIMITRIS G. MANOLAKIS

This page intentionally left blank

Digital Signal Processing

Fifth Edition

John G. Proakis

Department of Electrical and Computer Engineering
Northeastern University
Boston, Massachusetts

Dimitris G. Manolakis

MIT Lincoln Laboratory
Lexington, Massachusetts

Pearson

Content Management: Dawn Murrin, Erin Ault
Content Production: Ishan Chaudhary
Product Management: Holly Stark
Product Marketing: Wayne Stevens
Rights and Permissions: Rimpy Sharma and Tanvi Bhatia

Please contact https://support.pearson.com/getsupport/s/ with any queries on this content.

Print Offer
ISBN-10: 0-13-734861-4
ISBN-13: 978-0-13-734861-9

Rental
ISBN-10: 0-13-734824-X
ISBN-13: 978-0-13-734824-4

Library of Congress Cataloging-in-Publication Data

Names: Proakis, John G., author. | Manolakis, Dimitris G., author.
Title: Digital signal processing / John G. Proakis, Department of Electrical and Computer Engineering, Northeastern University, Boston, Massachusetts, Dimitris G. Manolakis, MIT Lincoln Laboratory, Lexington, Massachusetts.
Identifiers: LCCN 2020052822 | ISBN 9780137348619 (paperback) | ISBN 9780137348244 | ISBN 9780136961598 (epub)
Subjects: LCSH: Signal processing–Digital techniques.
Classification: LCC TK5102.9 .P757 2021 | DDC 621.382/2–dc23
LC record available at https://lccn.loc.gov/2020052822

ScoutAutomatedPrintCode

Pearson

To Felia, George, and Elena
—John G. Proakis

To Anna
—Dimitris Manolakis

This page intentionally left blank

Contents

This page intentionally left blank

Preface

This book was developed based on our teaching of undergraduate- and graduate-level courses in digital signal processing over the past several years. In this book, we present the fundamentals of discrete-time signals, systems, and modern digital processing as well as applications for students in electrical engineering, computer engineering, and computer science. The book is suitable for either a one-semester or a two-semester undergraduate-level course in discrete systems and digital signal processing. It is also intended for use in a one-semester first-year graduate-level course in digital signal processing.

We have assumed that students have taken undergraduate courses in advanced calculus (including ordinary differential equations) and linear systems for continuous-time signals, including an introduction to the Laplace transform. Although the Fourier series and Fourier transforms of periodic and aperiodic signals are described in Chapter 4, we expect that many students may have had this material in a prior course. In Chapters 13 to 15, some prior exposure to probability and random processes would be helpful.

We have provided balanced coverage of both theory and practical applications. A large number of well-designed problems are given to help the student in mastering the subject. A solutions manual is available for download for instructors only. Additionally, Microsoft PowerPoint slides of text figures are available for instructors at www.pearsonhighered.com/irc. If you are in need of a login and password for this site, please contact your local Pearson representative.

In the fifth edition of the book, we have added a new chapter on multirate digital filter banks and wavelets, and made modifications to existing chapters. Several new topics have been added, including the short-time Fourier Transform, the sparse FFT algorithm, ARMA model parameter estimation, and reverberation filters.

In Chapter 1, we present the basic elements of a digital signal processing system and the advantages of digital over analog signal processing. We also describe the classification of several different types of signals.

Chapter 2 is devoted entirely to the characterization and analysis of linear time-invariant (shift-invariant) discrete-time systems and discrete-time signals in the time domain. The convolution sum is derived and systems are categorized according to the duration of their impulse response as a finite-duration impulse response (FIR) and as an infinite-duration impulse response (IIR). Linear time-invariant (LTI) systems characterized by constant-coefficient difference equations are described and an application of LTI systems for signal smoothing is also discussed. The chapter concludes with a treatment of discrete-time correlation.

The z-transform is introduced in Chapter 3. Both the bilateral and the unilateral z-transforms are presented, and methods for determining the inverse z-transform are described. Use of the z-transform in the analysis of linear time-invariant systems is illustrated, and important properties of systems, such as causality and stability, are related to z-domain characteristics.

Chapter 4 treats the analysis of signals in the frequency domain. We begin by introducing the concept of frequency in continuous-time and discrete-time signals, and relate the two frequency variables by means of the sampling theorem. Fourier series and the Fourier transform are presented for both continuous-time and discrete-time signals.

In Chapter 5, linear time-invariant (LTI) discrete systems are characterized in the frequency domain by their frequency response function and their response to periodic and aperiodic signals is determined. A number of important types of discrete-time systems are described, including resonators, notch filters, comb filters, all-pass filters, and oscillators. The design of a number of simple FIR and IIR filters is also considered. In addition, the student is introduced to the concepts of minimum-phase, mixed- phase, and maximum-phase systems, and to the problem of deconvolution.

Chapter 6 provides a thorough treatment of sampling of continuous-time signals and the reconstruction of the signals from their samples. Our coverage includes the sampling and reconstruction of bandpass signals, the sampling of discrete-time signals, and A/D and D/A conversion. The chapter concludes with the treatment of oversampling A/D and D/A converters.

The DFT, its properties and its applications, are covered in Chapter 7. Two methods are described for using the DFT to perform linear filtering. The use of the DFT to perform frequency analysis of signals is also described. In addition, the short-time Fourier Transform is introduced. The final topic treated in this chapter is the discrete cosine transform.

Chapter 8 covers the efficient computation of the DFT. Included in this chapter are descriptions of radix-2, radix-4, and split-radix fast Fourier transform (FFT) algorithms, and applications of the FFT algorithms to the computation of convolution and correlation. The Goertzel algorithm and the chirp-z transform are introduced as two methods for computing the DFT using linear filtering. In addition, we introduce the sparse FFT algorithm.

Chapter 9 treats the realization of IIR and FIR systems. This treatment includes direct-form, cascade, parallel, lattice, and lattice-ladder realizations. The chapter also examines quantization effects in a digital implementation of FIR and IIR systems.

Techniques for design of digital FIR and IIR filters are presented in Chapter 10. The design techniques include both direct methods in discrete time and methods in volving the conversion of analog filters into digital filters by various transformations.

Chapter 11 treats sampling-rate conversion and its applications to multirate digital signal processing. In addition to describing decimation and interpolation by integer and rational factors, we describe methods for sampling-rate conversion by an arbitrary factor and implementations by polyphase filter structures.

Multirate digital filter banks and wavelets are covered in Chapter 12. We explain the two-channel quadrature mirror filter (QMF) banks and multichannel filter banks that eliminate aliasing and provide perfect reconstruction of signals. We also treat the design of FIR filters for both two-channel and multichannel filter banks. The second part of the chapter is focused on wavelets and the discrete wavelet transform. We describe the construction of the discrete wavelet transform and the connections between wavelets and filter banks.

Linear prediction and optimum linear (Wiener) filters are treated in Chapter 13. Also included in this chapter is descriptions of the Levinson–Durbin algorithm for solving normal equations, as well as the AR lattice and ARMA lattice-ladder filters.

Chapter 14 treats single-channel adaptive filters based on the LMS algorithm and on recursive least squares (RLS) algorithms. Both direct form FIR and lattice RLS algorithms and filter structures are described.

Power spectrum estimation is the main topic of Chapter 15. Our coverage includes a description of nonparametric and model-based (parametric) methods. Also described are eigen-decomposition-based methods, including MUSIC and ESPRIT. A one-semester senior-level course for students who have had prior exposure to discrete systems can use the material in Chapters 1 through 5 for a quick review and then proceed to cover Chapters 6 to 10.

In a first-year graduate-level course in digital signal processing, the first six chapters provide the student with a good review of discrete-time systems. The instructor can move quickly through most of this material and then cover Chapters 7 through 11, followed by selected topics from Chapters 12 to 15.

Numerous examples are included throughout the book. Over 500 homework problems, including computer problems, are provided to help the student. Answers to selected problems appear in the back of the book. The computer problems can be solved numerically by using a software package such as MATLAB or Python. The *Student Manual for Digital Signal Processing with MATLAB®*, which contains complete solutions to numerous computer problems, is also available for students at www.pearsonhighered.com/engineering-resources.

We are indebted to many faculty colleagues who provided valuable suggestions through reviews of previous editions of this book. These include W. E. Alexander, G. Arslan, Y. Bresler, J. Deller, F. DePiero, V. Ingle, J.S. Kang, C. Keller, H. Lev-Ari, L. Merakos, W. Mikhael, P. Monticciolo, C. Nikias, M. Schetzen, E. Serpedin, T. M. Sullivan, H. Trussell, S. Wilson, and M. Zoltowski. In addition, we acknowledge the following faculty members who reviewed the 4th edition of the book and made recommendations for its revision: D. Bukofzer, A. Dogandzic, E. Doering, E. Greco, R. Jordan, D. Krusienski, H. Lev-Ari, S. Nelatury, and M. Azimi-Sadjadi. Special thanks are given to H. Lev-Ari and T.Q. Nguyen for their invaluable suggestions in the preparation and review of the new material on filter banks and wavelets and to C. Nikias for the numerical results in Section 15.6.6. Finally, we wish to thank the former and current graduate students, A.L. Kok, J. Lin, E. Sozer, S. Srinidhi, Z. Li, and Y. Xiang, for their assistance in the preparation of several illustrations and the solutions manual.

JOHN G. PROAKIS
DIMITRIS G. MANOLAKIS

This page intentionally left blank

CHAPTER

1

Introduction

Digital signal processing is an area of science and engineering that has developed rapidly over the past 50 years. This rapid development is a result of the significant advances in digital computer technology and integrated-circuit fabrication. The digital computers and associated digital hardware of five decades ago were relatively large and expensive and, as a consequence, their use was limited to general-purpose non-real-time (off-line) scientific computations and business applications. The rapid developments in integrated-circuit technology, starting with medium-scale integration (MSI) and progressing to large-scale integration (LSI), and now, very-large-scale integration (VLSI) of electronic circuits has spurred the development of powerful, smaller, faster, and cheaper digital computers and special-purpose digital hardware. These inexpensive and relatively fast digital circuits have made it possible to construct highly sophisticated digital systems capable of performing complex digital signal processing functions and tasks, which are usually too difficult and/or too expensive to be performed by analog circuitry or analog signal processing systems. Hence many of the signal processing tasks that were conventionally performed by analog means are realized today by less expensive and often more reliable digital hardware.

We do not wish to imply that digital signal processing is the proper solution for all signal processing problems. Indeed, for many signals with extremely wide bandwidths, real-time processing is a requirement. For such signals, analog or, perhaps, optical signal processing is the only possible solution. However, where digital circuits are available and have sufficient speed to perform the signal processing, they are usually preferable.

Not only do digital circuits yield cheaper and more reliable systems for signal processing, they have other advantages as well. In particular, digital processing hardware allows programmable operations. Through software, one can more easily modify the signal processing functions to be performed by the hardware. Thus digital hardware and associated software provide a greater degree of flexibility in system design. Also, there is often a higher order of precision achievable with digital hardware and software compared with analog circuits and analog signal processing systems. For all these reasons, there has been an explosive growth in digital signal processing theory and applications over the past five decades.

In this book our objective is to present an introduction of the basic analysis tools and techniques for digital processing of signals. The emphasis is on the analysis and design of digital signal processing systems and computational techniques.

1.1 Signals, Systems, and Signal Processing

A *signal* is defined as any physical quantity that varies with time, space, or any other independent variable or variables. Mathematically, we describe a signal as a function of one or more independent variables. For example, the functions

$$\begin{aligned} s_1(t) &= 5t \\ s_2(t) &= 20t^2 \end{aligned} \tag{1.1.1}$$

describe two signals, one that varies linearly with the independent variable t (time) and a second that varies quadratically with t. As another example, consider the function

$$s(x, y) = 3x + 2xy + 10y^2 \tag{1.1.2}$$

This function describes a signal of two independent variables x and y that could represent the two spatial coordinates in a plane.

The signals described by (1.1.1) and (1.1.2) belong to a class of signals that are precisely defined by specifying the functional dependence on the independent variable. However, there are cases where such a functional relationship is unknown or too highly complicated to be of any practical use.

For example, a speech signal (see Fig. 1.1.1) cannot be described functionally by expressions such as (1.1.1). In general, a segment of speech may be represented to a high degree of accuracy as a sum of several sinusoids of different amplitudes and frequencies, that is, as

$$\sum_{i=1}^{N} A_i(t) \sin[2\pi F_i(t)t + \theta_i(t)] \tag{1.1.3}$$

where $\{A_i(t)\}$, $\{F_i(t)\}$, and $\{\theta_i(t)\}$ are the sets of (possibly time-varying) amplitudes, frequencies, and phases, respectively, of the sinusoids. In fact, one way to interpret the information content or message conveyed by any short time segment of the

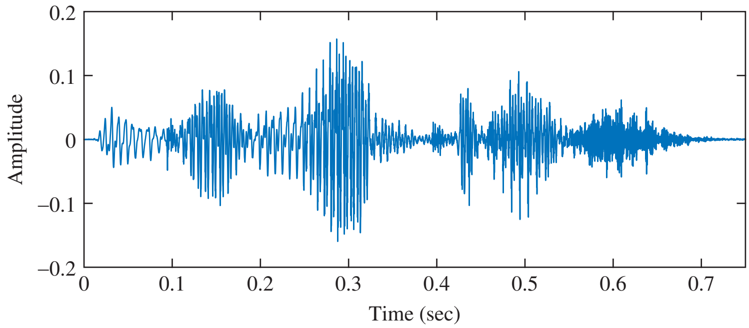


Figure 1.1.1 Example of a speech signal.

speech signal is to measure the amplitudes, frequencies, and phases contained in the short time segment of the signal.

Another example of a natural signal is an electrocardiogram (ECG). Such a signal provides a doctor with information about the condition of the patient's heart. Similarly, an electroencephalogram (EEG) signal provides information about the activity of the brain.

Speech, electrocardiogram, and electroencephalogram signals are examples of information-bearing signals that evolve as functions of a single independent variable, namely, time. An example of a signal that is a function of two independent variables is an image signal. The independent variables in this case are the spatial coordinates. These are but a few examples of the countless number of natural signals encountered in practice.

Associated with natural signals are the means by which such signals are generated. For example, speech signals are generated by forcing air through the vocal cords. Images are obtained by exposing a photographic film to a scene or an object. Thus signal generation is usually associated with a *system* that responds to a stimulus or force. In a speech signal, the system consists of the vocal cords and the vocal tract, also called the vocal cavity. The stimulus in combination with the system is called a *signal source*. Thus we have speech sources, images sources, and various other types of signal sources.

A *system* may also be defined as a physical device that performs an operation on a signal. For example, a filter used to reduce the noise and interference corrupting a desired information-bearing signal is called a system. In this case the filter performs some operation(s) on the signal, which has the effect of reducing (filtering) the noise and interference from the desired information-bearing signal.

When we pass a signal through a system, as in filtering, we say that we have processed the signal. In this case the processing of the signal involves filtering the noise and interference from the desired signal. In general, the system is characterized by the type of operation that it performs on the signal. For example, if the operation is linear, the system is called linear. If the operation on the signal is nonlinear, the system is said to be nonlinear, and so forth. Such operations are usually referred to as *signal processing*.

For our purposes, it is convenient to broaden the definition of a system to include not only physical devices, but also software realizations of operations on a signal. In digital processing of signals on a digital computer, the operations performed on a signal consist of a number of mathematical operations as specified by a software program. In this case, the program represents an implementation of the system in *software*. Thus we have a system that is realized on a digital computer by means of a sequence of mathematical operations; that is, we have a digital signal processing system realized in software. For example, a digital computer can be programmed to perform digital filtering. Alternatively, the digital processing on the signal may be performed by digital *hardware* (logic circuits) configured to perform the desired specified operations. In such a realization, we have a physical device that performs the specified operations. In a broader sense, a digital system can be implemented as a combination of digital hardware and software, each of which performs its own set of specified operations.

This book deals with the processing of signals by digital means, either in software or in hardware. Since many of the signals encountered in practice are analog, we will also consider the problem of converting an analog signal into a digital signal for processing. Thus we will be dealing primarily with digital systems. The operations performed by such a system can usually be specified mathematically. The method or set of rules for implementing the system by a program that performs the corresponding mathematical operations is called an *algorithm*. Usually, there are many ways or algorithms by which a system can be implemented, either in software or in hardware, to perform the desired operations and computations. In practice, we have an interest in devising algorithms that are computationally efficient, fast, and easily implemented. Thus a major topic in our study of digital signal processing is the discussion of efficient algorithms for performing such operations as filtering, correlation, and spectral analysis.

1.1.1 Basic Elements of a Digital Signal Processing System

Many of the signals encountered in science and engineering are analog in nature. That is, the signals are functions of a continuous variable, such as time or space, and usually take on values in a continuous range. Such signals may be processed directly by appropriate analog systems (such as filters, frequency analyzers, or frequency multipliers) for the purpose of changing their characteristics or extracting some desired information. In such a case we say that the signal has been processed directly in its analog form, as illustrated in Fig. 1.1.2. Both the input signal and the output signal are in analog form.

Digital signal processing provides an alternative method for processing the analog signal, as illustrated in Fig. 1.1.3. To perform the processing digitally, there

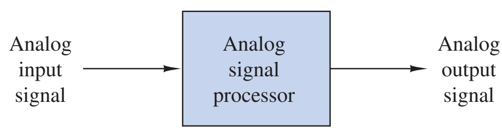


Figure 1.1.2
Analog signal processing.

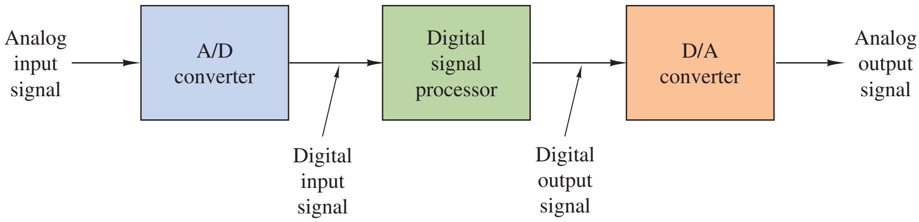


Figure 1.1.3 Block diagram of a digital signal processing system.

is a need for an interface between the analog signal and the digital processor. This interface is called an *analog-to-digital (A/D) converter*. The output of the A/D converter is a digital signal that is appropriate as an input to the digital processor.

The digital signal processor may be a large programmable digital computer or a small microprocessor programmed to perform the desired operations on the input signal. It may also be a hardwired digital processor configured to perform a specified set of operations on the input signal. Programmable machines provide the flexibility to change the signal processing operations through a change in the software, whereas hardwired machines are difficult to reconfigure. Consequently, programmable signal processors are in very common use. On the other hand, when signal processing operations are well defined, a hardwired implementation of the operations can be optimized, resulting in a cheaper signal processor and, usually, one that runs faster than its programmable counterpart. In applications where the digital output from the digital signal processor is to be given to the user in analog form, such as in speech communications, we must provide another interface from the digital domain to the analog domain. Such an interface is called a *digital-to-analog (D/A) converter*. Thus the signal is provided to the user in analog form, as illustrated in the block diagram of Fig. 1.1.3. However, there are other practical applications involving signal analysis, where the desired information is conveyed in digital form and no D/A converter is required. For example, in the digital processing of radar signals, the information extracted from the radar signal, such as the position of the aircraft and its speed, may simply be displayed on a computer terminal. There is no need for a D/A converter in this case.

1.1.2 Advantages of Digital over Analog Signal Processing

There are many reasons why digital signal processing of an analog signal may be preferable to processing the signal directly in the analog domain, as mentioned briefly earlier. First, a digital programmable system allows flexibility in reconfiguring the digital signal processing operations simply by changing the program. Reconfiguration of an analog system usually implies a redesign of the hardware followed by testing and verification to see that it operates properly.

Accuracy considerations also play an important role in determining the form of the signal processor. Tolerances in analog circuit components make it extremely difficult for the system designer to control the accuracy of an analog signal processing system. On the other hand, a digital system provides much better control of accuracy requirements. Such requirements, in turn, result in specifying the accuracy requirements in the A/D converter and the digital signal processor, in terms of word length, floating-point versus fixed-point arithmetic, and similar factors.

Digital signals are easily stored on magnetic media (tape or disk) without deterioration or loss of signal fidelity beyond that introduced in the A/D conversion. As a consequence, the signals become transportable and can be processed off-line in a remote laboratory. The digital signal processing method also allows for the implementation of more sophisticated signal processing algorithms. It is usually very difficult to perform precise mathematical operations on signals in analog form but these same operations can be routinely implemented on a digital computer using software.

In some cases a digital implementation of the signal processing system is cheaper than its analog counterpart. The lower cost may be due to the fact that the digital hardware is cheaper, or perhaps it is a result of the flexibility for modifications provided by the digital implementation.

As a consequence of these advantages, digital signal processing has been applied in practical systems covering a broad range of disciplines. We cite, for example, the application of digital signal processing techniques in speech processing and signal transmission on telephone channels, in image processing and transmission, in seismology and geophysics, in oil exploration, in the detection of nuclear explosions, in the processing of signals received from outer space, and in a vast variety of other applications. Some of these applications are cited in subsequent chapters.

As already indicated, however, digital implementation has its limitations. One practical limitation is the speed of operation of A/D converters and digital signal processors. We shall see that signals having extremely wide bandwidths require fast-sampling-rate A/D converters and fast digital signal processors. Hence there are analog signals with large bandwidths for which a digital processing approach is beyond the state of the art of digital hardware.

1.2 Classification of Signals

The methods we use in processing a signal or in analyzing the response of a system to a signal depend heavily on the characteristic attributes of the specific signal. There are techniques that apply only to specific families of signals. Consequently, any investigation in signal processing should start with a classification of the signals involved in the specific application.

1.2.1 Multichannel and Multidimensional Signals

As explained in Section 1.1, a signal is described by a function of one or more independent variables. The value of the function (i.e., the dependent variable) can be a real-valued scalar quantity, a complex-valued quantity, or perhaps a vector. For example, the signal

$$s_1(t) = A \sin 3\pi t$$

is a real-valued signal. However, the signal

$$s_2(t) = Ae^{j3\pi t} = A\cos 3\pi t + jA\sin 3\pi t$$

is complex valued.

In some applications, signals are generated by multiple sources or multiple sensors. Such signals, in turn, can be represented in vector form. Figure 1.2.1 shows the three components of a vector signal that represents the ground acceleration due to an earthquake. This acceleration is the result of three basic types of elastic waves. The primary (P) waves and the secondary (S) waves propagate within the body of rock and are longitudinal and transversal, respectively. The third type of elastic wave is called the surface wave, because it propagates near the ground surface. If $s_k(t)$,

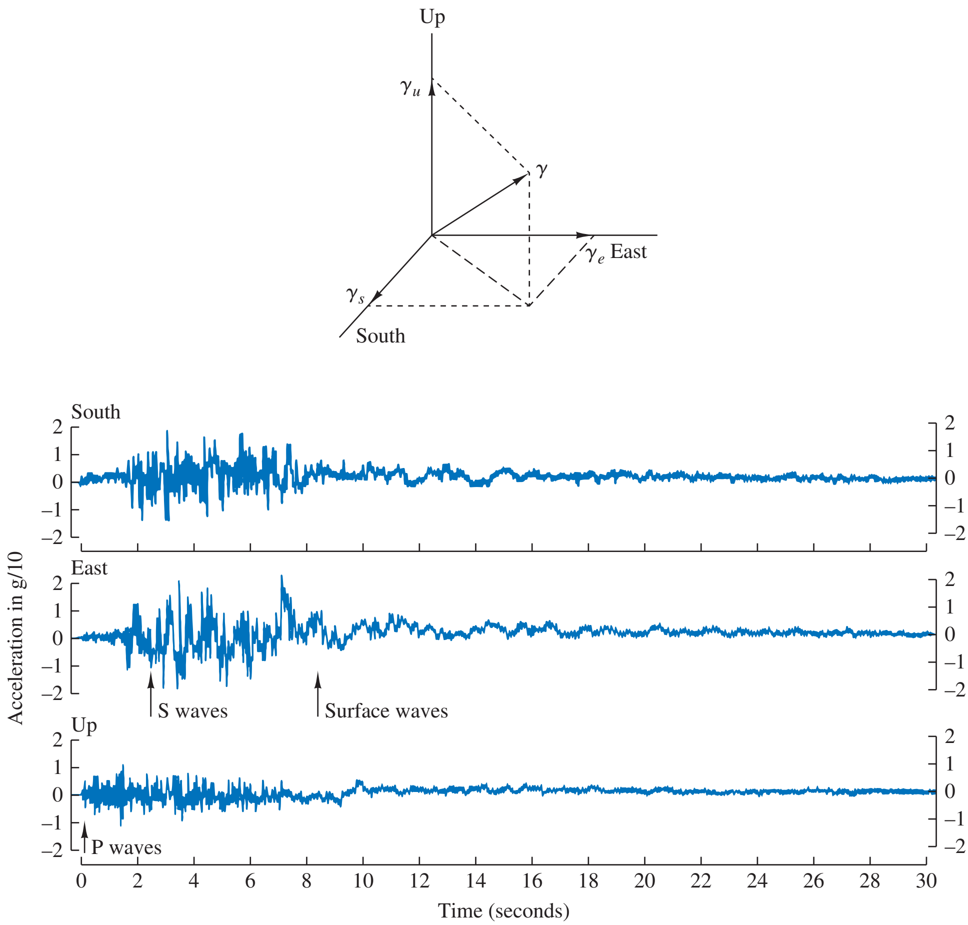


Figure 1.2.1 Three components of ground acceleration measured a few kilometers from the epicenter of an earthquake. (From *Earthquakes*, by B. A. Bold, ©1988 by W. H. Freeman and Company. Reprinted with permission of the publisher.)

$k = 1, 2, 3$, denotes the electrical signal from the kth sensor as a function of time, the set of $p = 3$ signals can be represented by a vector $\mathbf{S}_3(t)$, where

$$\mathbf{S}_3(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix}$$

We refer to such a vector of signals as a *multichannel signal*. In electrocardiography, for example, 3-lead and 12-lead electrocardiograms (ECG) are often used in practice, which result in 3-channel and 12-channel signals.

Let us now turn our attention to the independent variable(s). If the signal is a function of a single independent variable, the signal is called a *one-dimensional* signal. On the other hand, a signal is called *M-dimensional* if its value is a function of M independent variables.

The picture shown in Fig. 1.2.2 is an example of a two-dimensional signal, since the intensity or brightness $I(x, y)$ at each point is a function of two independent variables. On the other hand, a black-and-white television picture may be represented as $I(x, y, t)$ since the brightness is a function of time. Hence the TV picture may be treated as a three-dimensional signal. In contrast, a color TV picture may be described by three intensity functions of the form $I_r(x, y, t)$, $I_g(x, y, t)$, and $I_b(x, y, t)$, corresponding to the brightness of the three principal colors (red, green, blue) as functions of time. Hence the color TV picture is a three-channel, three-dimensional signal, which can be represented by the vector

$$\mathbf{I}(x, y, t) = \begin{bmatrix} I_r(x, y, t) \\ I_g(x, y, t) \\ I_b(x, y, t) \end{bmatrix}$$

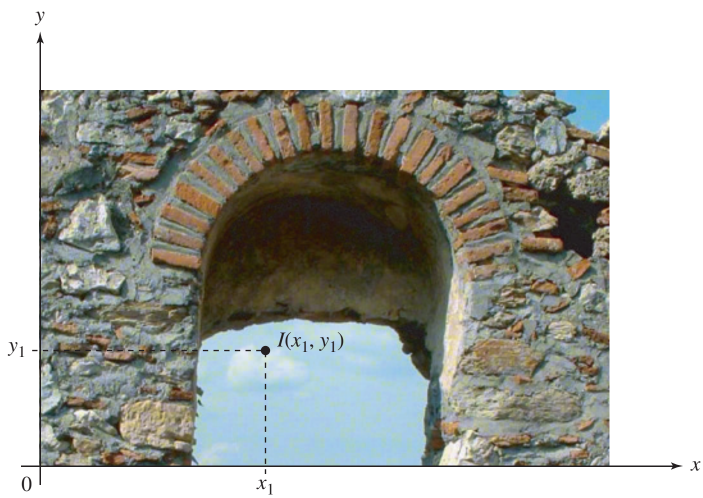


Figure 1.2.2
Example of a two-dimensional signal.

In this book we deal mainly with single-channel, one-dimensional real- or complex-valued signals and we refer to them simply as signals. In mathematical terms these signals are described by a function of a single independent variable. Although the independent variable need not be time, it is common practice to use t as the independent variable. In many cases the signal processing operations and algorithms developed in this text for one-dimensional, single-channel signals can be extended to multichannel and multidimensional signals.

1.2.2 Continuous-Time Versus Discrete-Time Signals

Signals can be further classified into four different categories depending on the characteristics of the time (independent) variable and the values they take. *Continuous-time signals* or *analog signals* are defined for every value of time and they take on values in the continuous interval (a, b), where a can be $-\infty$ and b can be ∞. Mathematically, these signals can be described by functions of a continuous variable. The speech waveform in Fig. 1.1.1 and the signals $x_1(t) = \cos \pi t$, $x_2(t) = e^{-|t|}$, $-\infty < t < \infty$ are examples of analog signals. *Discrete-time signals* are defined only at certain specific values of time. These time instants need not be equidistant, but in practice they are usually taken at equally spaced intervals for computational convenience and mathematical tractability. The signal $x(t_n) = e^{-|t_n|}$, $n = 0, \pm 1, \pm 2, \ldots$ provides an example of a discrete-time signal. If we use the index n of the discrete-time instants as the independent variable, the signal value becomes a function of an integer variable (i.e., a sequence of numbers). Thus a discrete-time signal can be represented mathematically by a sequence of real or complex numbers. To emphasize the discrete-time nature of a signal, we shall denote such a signal as $x(n)$ instead of $x(t)$. If the time instants t_n are equally spaced (i.e., $t_n = nT$), the notation $x(nT)$ is also used. For example, the sequence

$$x(n) = \begin{cases} 0.8^n, & \text{if } n \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{1.2.1}$$

is a discrete-time signal, which is represented graphically as in Fig. 1.2.3.

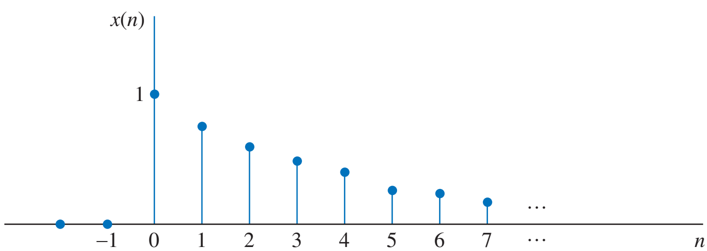


Figure 1.2.3 Graphical representation of the discrete time signal $x(n) = 0.8^n$ for $n > 0$ and $x(n) = 0$ for $n < 0$.

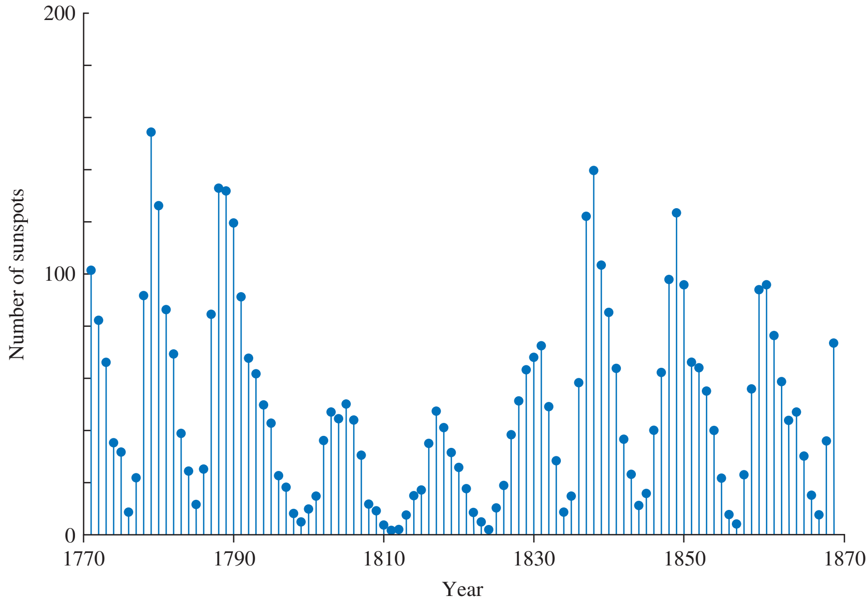


Figure 1.2.4 Wölfer annual sunspot numbers (1770–1869).

In applications, discrete-time signals may arise in two ways:

1. By selecting values of an analog signal at discrete-time instants. This process is called *sampling* and is discussed in more detail in Chapter 6. All measuring instruments that take measurements at a regular interval of time provide discrete-time signals. For example, the signal $x(n)$ in Fig. 1.2.3 can be obtained by sampling the analog signal $x(t) = 0.8^t$, $t \geq 0$ and $x(t) = 0$, $t < 0$ once every second.

2. By accumulating a variable over a period of time. For example, counting the number of cars using a given street every hour, or recording the value of gold every day, results in discrete-time signals. Figure 1.2.4 shows a graph of the Wölfer sunspot numbers. Each sample of this discrete-time signal provides the number of sunspots observed during an interval of 1 year.

1.2.3 Continuous-Valued Versus Discrete-Valued Signals

The values of a continuous-time or discrete-time signal can be continuous or discrete. If a signal takes on all possible values on a finite or an infinite range, it is said to be a continuous-valued signal. Alternatively, if the signal takes on values from a finite set of possible values, it is said to be a discrete-valued signal. Usually, these values are equidistant and hence can be expressed as an integer multiple of the distance between two successive values. A discrete-time signal having a set of discrete values is called a *digital signal*. Figure 1.2.5 shows a digital signal that takes on one of four possible values.

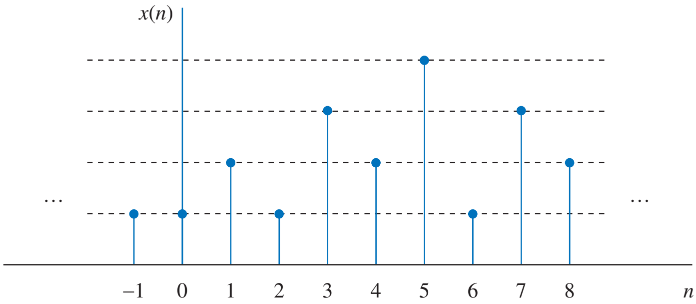


Figure 1.2.5 Digital signal with four different amplitude values.

In order for a signal to be processed digitally, it must be discrete in time and its values must be discrete (i.e., it must be a digital signal). If the signal to be processed is in analog form, it is converted to a digital signal by sampling the analog signal at discrete instants in time, obtaining a discrete-time signal, and then by *quantizing* its values to a set of discrete values, as described in Chapter 6. The process of converting a continuous-valued signal into a discrete-valued signal, called *quantization*, is basically an approximation process. It may be accomplished simply by rounding or truncation. For example, if the allowable signal values in the digital signal are integers, say 0 through 15, the continuous-value signal is quantized into these integer values. Thus the signal value 8.58 will be approximated by the value 8 if the quantization process is performed by truncation or by 9 if the quantization process is performed by rounding to the nearest integer.

1.2.4 Deterministic Versus Random Signals

The mathematical analysis and processing of signals requires the availability of a mathematical description for the signal itself. This mathematical description, often referred to as the *signal model*, leads to another important classification of signals. Any signal that can be uniquely described by an explicit mathematical expression, a table of data, or a well-defined rule is called *deterministic*. This term is used to emphasize the fact that all past, present, and future values of the signal are known precisely, without any uncertainty.

In many practical applications, however, there are signals that either cannot be described to any reasonable degree of accuracy by explicit mathematical formulas, or such a description is too complicated to be of any practical use. The lack of such a relationship implies that such signals evolve in time in an unpredictable manner. We refer to these signals as *random*. The output of a noise generator, the seismic signal of Fig. 1.2.1, and the speech signal in Fig. 1.1.1 are examples of random signals.

The mathematical framework for the theoretical analysis of random signals is provided by the theory of probability and stochastic processes. Some basic elements of this approach, adapted to the needs of this book, are presented in Section 13.1.

It should be emphasized at this point that the classification of a *real-world* signal as deterministic or random is not always clear. Sometimes, both approaches lead to meaningful results that provide more insight into signal behavior. At other times, the wrong classification may lead to erroneous results, since some mathematical tools may apply only to deterministic signals while others may apply only to random signals. This will become clearer as we examine specific mathematical tools.

1.3 Summary

In this introductory chapter we have attempted to provide the motivation for digital signal processing as an alternative to analog signal processing. We presented the basic elements of a digital signal processing system and defined the operations needed to convert an analog signal into a digital signal ready for processing.

There are numerous practical applications of digital signal processing. The book edited by Oppenheim (1978) treats applications to speech processing, image processing, radar signal processing, sonar signal processing, and geophysical signal processing.

Problems

1.1 Classify the following signals according to whether they are (1) one- or multi-dimensional, (2) single or multichannel, (3) continuous time or discrete time, and (4) analog or digital (in amplitude). Give a brief explanation.

(a) Closing prices of utility stocks on the New York Stock Exchange.

(b) A color movie.

(c) Position of the steering wheel of a car in motion relative to car's reference frame.

(d) Position of the steering wheel of a car in motion relative to ground reference frame.

(e) Weight and height measurements of a child taken every month.

CHAPTER

2

Discrete-Time Signals and Systems

In Chapter 1 we introduced the reader to a number of important types of signals. In this chapter we present several elementary signals that are important in our treatment of signal processing. These discrete-time signals are used as basis functions or building blocks to describe more complex signals.

The major emphasis in this chapter is the characterization of discrete-time systems in general and the class of linear time-invariant (LTI) systems in particular. A number of important time-domain properties of LTI systems are defined and developed, and an important formula, called the convolution formula, is derived which allows us to determine the output of an LTI system to any given arbitrary input signal. In addition to the convolution formula, difference equations are introduced as an alternative method for describing the input–output relationship of an LTI system, and in addition, recursive and nonrecursive realizations of LTI systems are treated.

Our motivation for the emphasis on the study of LTI systems is twofold. First, there is a large collection of mathematical techniques that can be applied to the analysis of LTI systems. Second, many practical systems are either LTI systems or can be approximated by LTI systems. Because of its importance in digital signal processing applications and its close resemblance to the convolution formula, we also introduce the correlation between two signals. The autocorrelation and crosscorrelation of signals are defined and their properties are presented.

2.1 Discrete-Time Signals

As we discussed in Chapter 1, a discrete-time signal $x(n)$ is a function of an independent variable that is an integer. It is graphically represented as in Fig. 2.1.1. It is important to note that a discrete-time signal is *not defined* at instants between two successive samples. Also, it is incorrect to think that $x(n)$ is equal to zero if n is not an integer. Simply, the signal $x(n)$ is not defined for noninteger values of n.

In the sequel we will assume that a discrete-time signal is defined for every integer value n for $-\infty < n < \infty$. By tradition, we refer to $x(n)$ as the "nth sample" of the signal even if the signal $x(n)$ is inherently discrete time (i.e., not obtained by sampling an analog signal). If, indeed, $x(n)$ was obtained from sampling an analog signal $x_a(t)$, then $x(n) \equiv x_a(nT)$, where T is the sampling period (i.e., the time between successive samples).

Besides the graphical representation of a discrete-time signal or sequence as illustrated in Fig. 2.1.1, there are some alternative representations that are often more convenient to use. These are:

1. Functional representation, such as

$$x(n) = \begin{cases} 1, & \text{for } n = 1, 3 \\ 4, & \text{for } n = 2 \\ 0, & \text{elsewhere} \end{cases} \tag{2.1.1}$$

2. Tabular representation, such as

n	$\cdots$	-2	-1	0	1	2	3	4	5	$\ldots$
$x(n)$	$\cdots$	0	0	0	1	4	1	0	0	$\ldots$

3. Sequence representation

An infinite-duration signal or sequence with the time origin ($n = 0$) indicated by the symbol $\uparrow$ is represented as

$$x(n) = \{\ldots 0, \underset{\uparrow}{0}, 1, 4, 1, 0, 0, \ldots\} \tag{2.1.2}$$

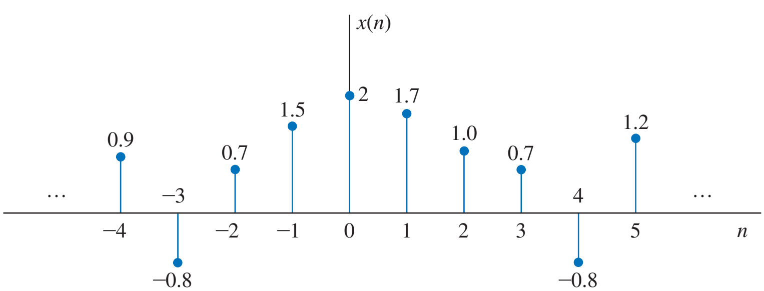


Figure 2.1.1 Graphical representation of a discrete-time signal.

A sequence $x(n)$, which is zero for $n < 0$, can be represented as

$$x(n) = \{\underset{\uparrow}{0}, 1, 4, 1, 0, 0, \ldots\} \tag{2.1.3}$$

The time origin for a sequence $x(n)$, which is zero for $n < 0$, is understood to be the first (leftmost) point in the sequence.

A finite-duration sequence can be represented as

$$x(n) = \{3, -1, \underset{\uparrow}{-2}, 5, 0, 4, -1\} \tag{2.1.4}$$

whereas a finite-duration sequence that satisfies the condition $x(n) = 0$ for $n < 0$ can be represented as

$$x(n) = \{\underset{\uparrow}{0}, 1, 4, 1\} \tag{2.1.5}$$

The signal in (2.1.4) consists of seven samples or points (in time), so it is called or identified as a seven-point sequence. Similarly, the sequence given by (2.1.5) is a four-point sequence.

2.1.1 Some Elementary Discrete-Time Signals

In our study of discrete-time signals and systems there are a number of basic signals that appear often and play an important role. These signals are defined below.

1. The *unit sample sequence* is denoted as $\delta(n)$ and is defined as

$$\delta(n) \equiv \begin{cases} 1, & \text{for } n = 0 \\ 0, & \text{for } n \neq 0 \end{cases} \tag{2.1.6}$$

In words, the unit sample sequence is a signal that is zero everywhere, except at $n = 0$ where its value is unity. This signal is sometimes referred to as a *unit impulse*. In contrast to the analog signal $\delta(t)$, which is also called a unit impulse and is defined to be zero everywhere except at $t = 0$, and has unit area, the unit sample sequence is much less mathematically complicated. The graphical representation of $\delta(n)$ is shown in Fig. 2.1.2.

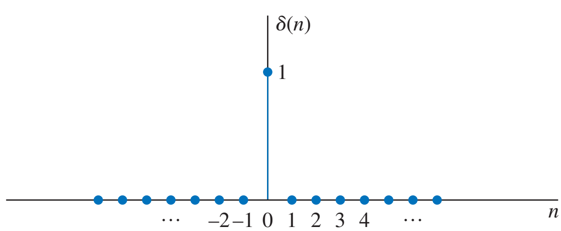


Figure 2.1.2
Graphical representation of the unit sample signal.

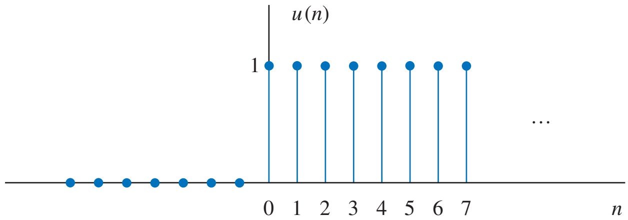


Figure 2.1.3
Graphical representation of the unit step signal.

2. The *unit step signal* is denoted as $u(n)$ and is defined as

$$u(n) \equiv \begin{cases} 1, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases} \tag{2.1.7}$$

Figure 2.1.3 illustrates the unit step signal.

3. The *unit ramp signal* is denoted as $u_r(n)$ and is defined as

$$u_r(n) \equiv \begin{cases} n, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases} \tag{2.1.8}$$

This signal is illustrated in Fig. 2.1.4.

4. The *exponential signal* is a sequence of the form

$$x(n) = a^n \qquad \text{for all } n \tag{2.1.9}$$

If the parameter a is real, then $x(n)$ is a real signal. Figure 2.1.5 illustrates $x(n)$ for various values of the parameter a.

When the parameter a is complex valued, it can be expressed as

$$a \equiv re^{j\theta}$$

where r and θ are now the parameters. Hence we can express $x(n)$ as

$$\begin{aligned} x(n) &= r^n e^{j\theta n} \\ &= r^n(\cos\theta n + j\sin\theta n) \end{aligned} \tag{2.1.10}$$

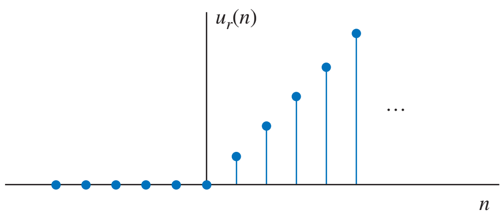


Figure 2.1.4
Graphical representation of the unit ramp signal.

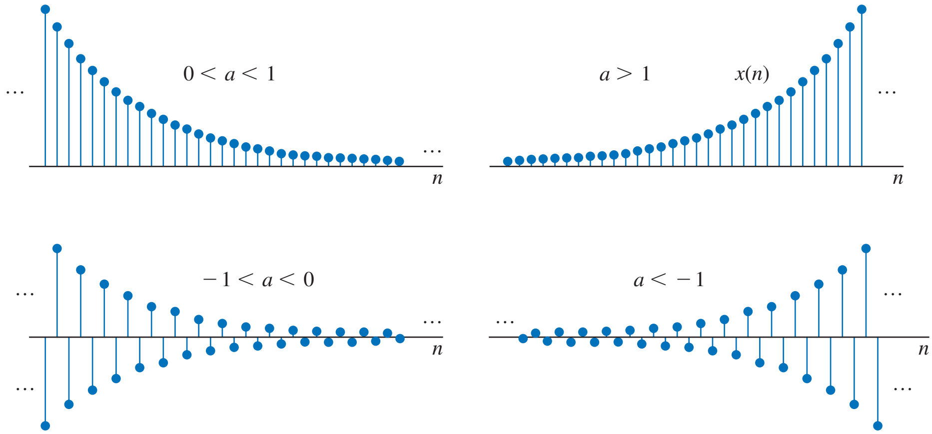


Figure 2.1.5 Graphical representation of exponential signals.

Since $x(n)$ is now complex valued, it can be represented graphically by plotting the real part

$$x_R(n) \equiv r^n \cos \theta n \tag{2.1.11}$$

as a function of n, and separately plotting the imaginary part

$$x_I(n) \equiv r^n \sin \theta n \tag{2.1.12}$$

as a function of n. Figure 2.1.6 illustrates the graphs of $x_R(n)$ and $x_I(n)$ for $r = 0.9$ and $\theta = \pi/10$. We observe that the signals $x_R(n)$ and $x_I(n)$ are a damped (decaying exponential) cosine function and a damped sine function. The angle variable θ is simply the frequency of the sinusoid. Clearly, if $r = 1$, the damping disappears and $x_R(n)$, $x_I(n)$, and $x(n)$ have a fixed amplitude, which is unity.

Alternatively, the signal $x(n)$ given by (2.1.10) can be represented graphically by the amplitude function

$$|x(n)| = A(n) \equiv r^n \tag{2.1.13}$$

and the phase function

$$\angle x(n) = \phi(n) \equiv \theta n \tag{2.1.14}$$

Figure 2.1.7 illustrates $A(n)$ and $\phi(n)$ for $r = 0.9$ and $\theta = \pi/10$. We observe that the phase function is linear with n. However, the phase is defined only over the interval $-\pi < \theta \leq \pi$ or, equivalently, over the interval $0 \leq \theta < 2\pi$. Consequently, by convention $\phi(n)$ is plotted over the finite interval $-\pi < \theta \leq \pi$ or $0 \leq \theta < 2\pi$. In other words, we subtract multiples of 2π from $\phi(n)$ before plotting. The subtraction of multiples of 2π from $\phi(n)$ is equivalent to interpreting the function $\phi(n)$ as $\phi(n)$, modulo 2π.

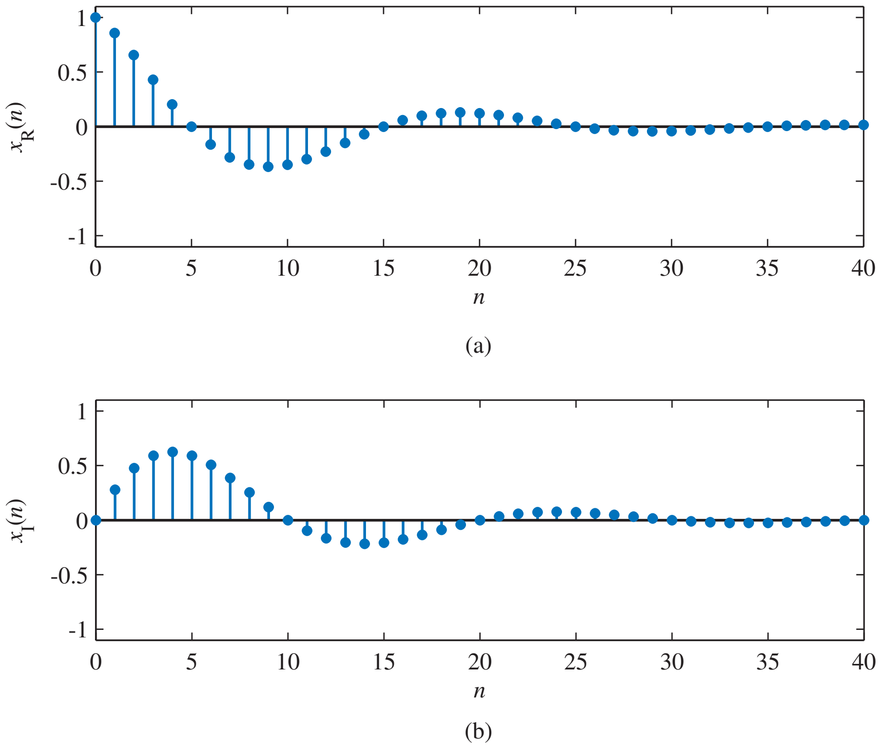


Figure 2.1.6 Graph of the real and imaginary components of a complex-valued exponential signal.

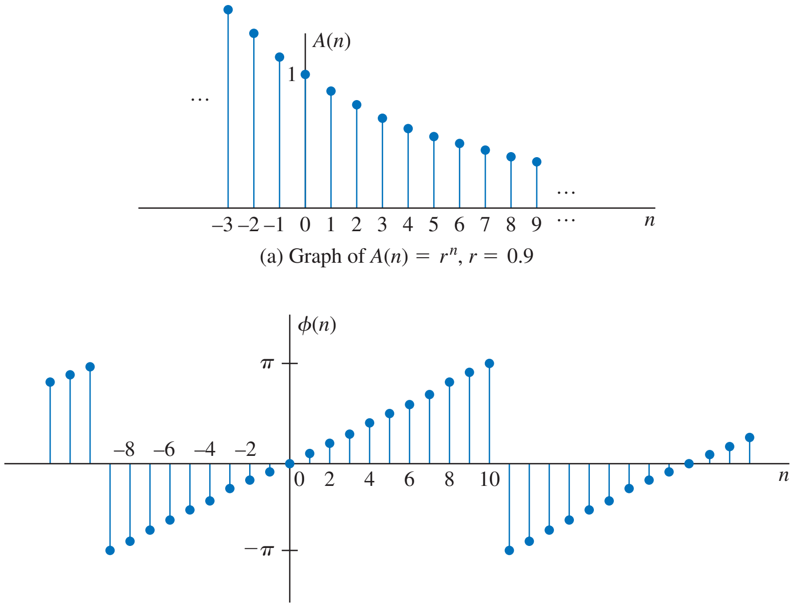
2.1.2 Classification of Discrete-Time Signals

The mathematical methods employed in the analysis of discrete-time signals and systems depend on the characteristics of the signals. In this section we classify discrete-time signals according to a number of different characteristics.

Energy signals and power signals. The energy E of a signal $x(n)$ is defined as

$$E \equiv \sum_{n=-\infty}^{\infty} |x(n)|^2 \tag{2.1.15}$$

We have used the magnitude-squared values of $x(n)$, so that our definition applies to complex-valued signals as well as real-valued signals. The energy of a signal can be finite or infinite. If E is finite (i.e., $0 < E < \infty$), then $x(n)$ is called an *energy signal*. Sometimes we add a subscript x to E and write E_x to emphasize that E_x is the energy of the signal $x(n)$.



(a) Graph of $A(n) = r^n$, $r = 0.9$

(b) Graph of $\phi(n) = \frac{\pi}{10}n$, modulo 2π plotted in the range $(-\pi, \pi)$

Figure 2.1.7 Graph of amplitude and phase function of a complex-valued exponential signal: (a) graph of $A(n) = r^n$, $r = 0.9$; (b) graph of $\phi(n) = (\pi/10)n$, modulo 2π plotted in the range $(-\pi, \pi]$.

Many signals that possess infinite energy have a finite average power. The average power of a discrete-time signal $x(n)$ is defined as

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2 \tag{2.1.16}$$

If we define the signal energy of $x(n)$ over the finite interval $-N \le n \le N$ as

$$E_N \equiv \sum_{n=-N}^{N} |x(n)|^2 \tag{2.1.17}$$

then we can express the signal energy E as

$$E \equiv \lim_{N\to\infty} E_N \tag{2.1.18}$$

and the average power of the signal $x(n)$ as

$$P \equiv \lim_{N\to\infty} \frac{1}{2N+1} E_N \tag{2.1.19}$$

Clearly, if E is finite, $P = 0$. On the other hand, if E is infinite, the average power P may be either finite or infinite. If P is finite (and nonzero), the signal is called a *power signal*. The following example illustrates such a signal.

EXAMPLE 2.1.1

Determine the power and energy of the unit step sequence. The average power of the unit step signal is

$$\begin{aligned} P &= \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=0}^{N} u^2(n) \\ &= \lim_{N\to\infty} \frac{N+1}{2N+1} = \lim_{N\to\infty} \frac{1+1/N}{2+1/N} = \frac{1}{2} \end{aligned}$$

Consequently, the unit step sequence is a power signal. Its energy is infinite.

Similarly, it can be shown that the complex exponential sequence $x(n) = Ae^{j\omega_0 n}$ has average power A^2, so it is a power signal. On the other hand, the unit ramp sequence is neither a power signal nor an energy signal.

Periodic signals and aperiodic signals. A signal $x(n)$ is periodic with period $N (N > 0)$ if and only if

$$x(n+N) = x(n) \text{ for all } n \tag{2.1.20}$$

The smallest value of N for which (2.1.20) holds is called the (fundamental) period. If there is no value of N that satisfies (2.1.20), the signal is called *nonperiodic* or *aperiodic*.

We observe that the sinusoidal signal of the form

$$x(n) = A \sin 2\pi f_0 n \tag{2.1.21}$$

is periodic when f_0 is a rational number, that is, if f_0 can be expressed as

$$f_0 = \frac{k}{N} \tag{2.1.22}$$

where k and N are integers.

The energy of a periodic signal $x(n)$ over a single period, say, over the interval $0 \leq n \leq N-1$, is finite if $x(n)$ takes on finite values over the period. However, the energy of the periodic signal for $-\infty \leq n \leq \infty$ is infinite. On the other hand, the average power of the periodic signal is finite and it is equal to the average power over a single period. Thus if $x(n)$ is a periodic signal with fundamental period N and takes on finite values, its power is given by

$$P = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 \tag{2.1.23}$$

Consequently, periodic signals are power signals.

Symmetric (even) and antisymmetric (odd) signals. A real-valued signal $x(n)$ is called symmetric (even) if

$$x(-n) = x(n) \tag{2.1.24}$$

On the other hand, a signal $x(n)$ is called antisymmetric (odd) if

$$x(-n) = -x(n) \tag{2.1.25}$$

We note that if $x(n)$ is odd, then $x(0) = 0$. Examples of signals with even and odd symmetry are illustrated in Fig. 2.1.8.

We wish to illustrate that any arbitrary signal can be expressed as the sum of two signal components, one of which is even and the other odd. The even signal component is formed by adding $x(n)$ to $x(-n)$ and dividing by 2, that is,

$$x_e(n) = \frac{1}{2}[x(n) + x(-n)] \tag{2.1.26}$$

Clearly, $x_e(n)$ satisfies the symmetry condition (2.1.24). Similarly, we form an odd signal component $x_o(n)$ according to the relation

$$x_o(n) = \frac{1}{2}[x(n) - x(-n)] \tag{2.1.27}$$

Again, it is clear that $x_o(n)$ satisfies (2.1.25); hence it is indeed odd. Now, if we add the two signal components, defined by (2.1.26) and (2.1.27), we obtain $x(n)$, that is,

$$x(n) = x_e(n) + x_o(n) \tag{2.1.28}$$

Thus any arbitrary signal can be expressed as in (2.1.28).

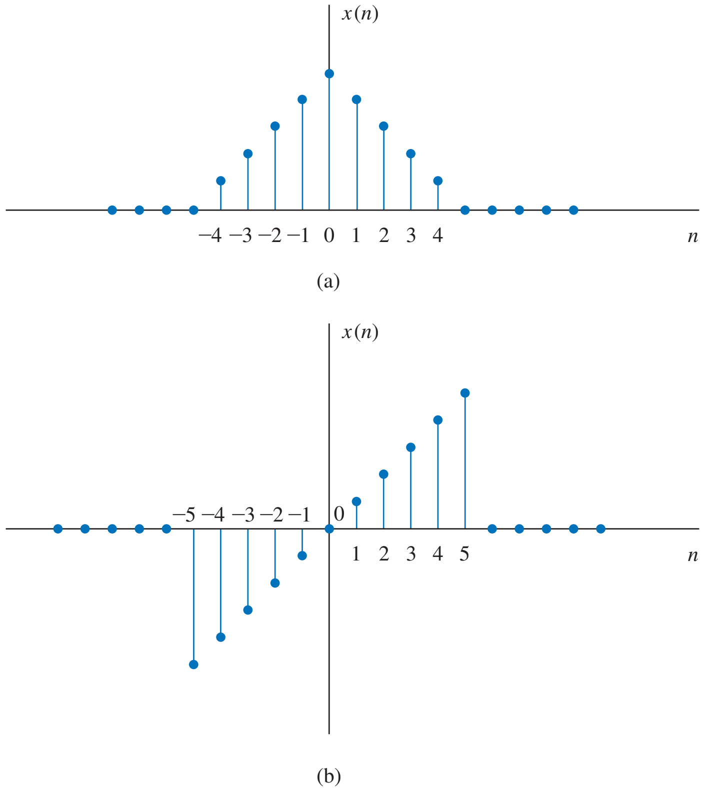


Figure 2.1.8 Example of even (a) and odd (b) signals.

2.1.3 Simple Manipulations of Discrete-Time Signals

In this section we consider some simple modifications or manipulations involving the independent variable and the signal amplitude (dependent variable).

Transformation of the independent variable (time). A signal $x(n)$ may be shifted in time by replacing the independent variable n by $n - k$, where k is an integer. If k is a positive integer, the time shift results in a delay of the signal by k units of time. If k is a negative integer, the time shift results in an advance of the signal by $|k|$ units in time.

EXAMPLE 2.1.2

A signal $x(n)$ is graphically illustrated in Fig. 2.1.9(a). Show a graphical representation of the signals $x(n - 3)$ and $x(n + 2)$.

Solution. The signal $x(n-3)$ is obtained by delaying $x(n)$ by three units in time. The result is illustrated in Fig. 2.1.9(b). On the other hand, the signal $x(n + 2)$ is obtained by advancing $x(n)$ by two units in time. The result is illustrated in Fig. 2.1.9(c). Note that delay corresponds

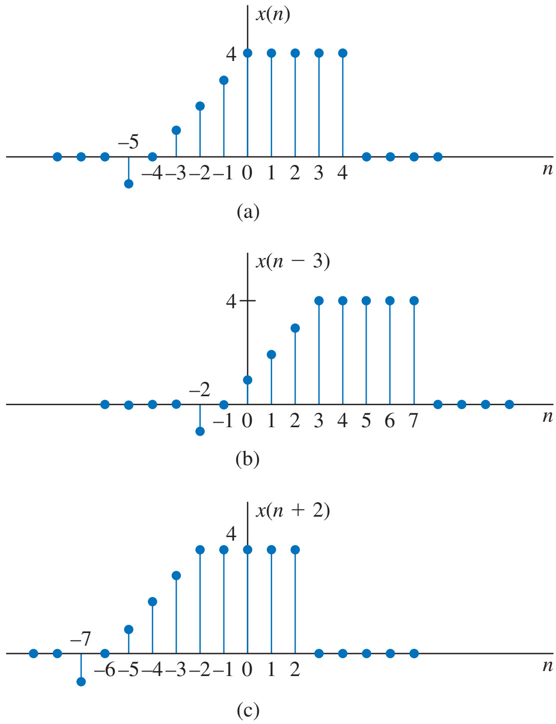


Figure 2.1.9
Graphical representation of a signal, and its delayed and advanced versions.

to shifting a signal to the right, whereas advance implies shifting the signal to the left on the time axis.

If the signal $x(n)$ is stored on magnetic tape or on a disk or, perhaps, in the memory of a computer, it is a relatively simple operation to modify the base by introducing a delay or an advance. On the other hand, if the signal is not stored but is being generated by some physical phenomenon in real time, it is not possible to advance the signal in time, since such an operation involves signal samples that have not yet been generated. Whereas it is always possible to insert a delay into signal samples that have already been generated, it is physically impossible to view the future signal samples. Consequently, in real-time signal processing applications, the operation of advancing the time base of the signal is physically unrealizable.

Another useful modification of the time base is to replace the independent variable n by $-n$. The result of this operation is a *folding* or a *reflection* of the signal about the time origin $n = 0$.

EXAMPLE 2.1.3

Show the graphical representation of the signals $x(-n)$ and $x(-n+2)$, where $x(n)$ is the signal illustrated in Fig. 2.1.10(a).

Solution. The new signal $y(n) = x(-n)$ is shown in Fig. 2.1.10(b). Note that $y(0) = x(0)$, $y(1) = x(-1)$, $y(2) = x(-2)$, and so on. Also, $y(-1) = x(1)$, $y(-2) = x(2)$, and so on.

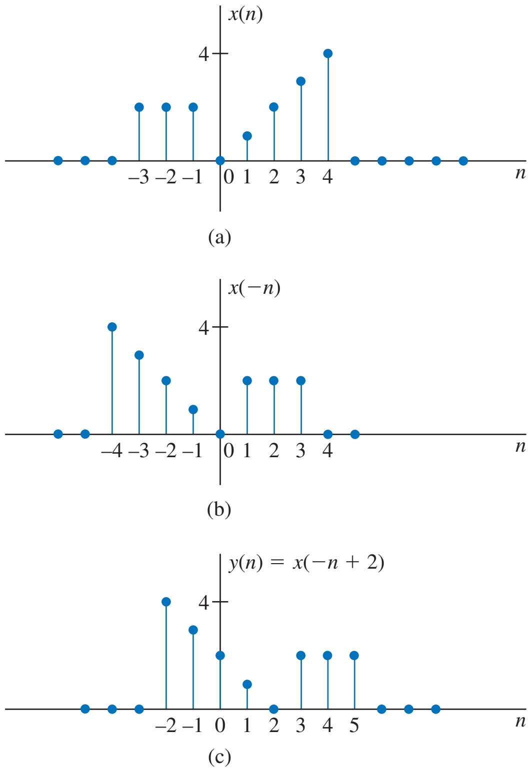


Figure 2.1.10
Graphical illustration of the folding and shifting operations.

Therefore, $y(n)$ is simply $x(n)$ reflected or folded about the time origin $n = 0$. The signal $y(n) = x(-n + 2)$ is simply $x(-n)$ delayed by two units in time. The resulting signal is illustrated in Fig. 2.1.10(c). A simple way to verify that the result in Fig. 2.1.10(c) is correct is to compute samples, such as $y(0) = x(2)$, $y(1) = x(1)$, $y(2) = x(0)$, $y(-1) = x(3)$, and so on.

It is important to note that the operations of folding and time delaying (or advancing) a signal are not commutative. If we denote the time-delay operation by TD and the folding operation by FD, we can write

$$\begin{aligned} \mathrm{TD}_k[x(n)] &= x(n-k), \qquad k > 0 \\ \mathrm{FD}[x(n)] &= x(-n) \end{aligned} \tag{2.1.29}$$

Now

$$\mathrm{TD}_k\{\mathrm{FD}[x(n)]\} = \mathrm{TD}_k[x(-n)] = x(-n+k) \tag{2.1.30}$$

whereas

$$\mathrm{FD}\{\mathrm{TD}_k[x(n)]\} = \mathrm{FD}[x(n-k)] = x(-n-k) \tag{2.1.31}$$

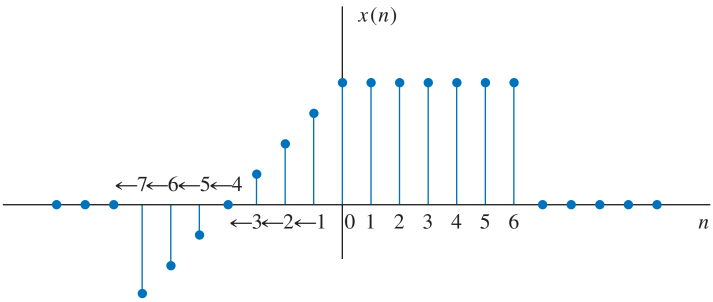
Note that because the signs of n and k in $x(n-k)$ and $x(-n+k)$ are different, the result is a shift of the signals $x(n)$ and $x(-n)$ to the right by k samples, corresponding to a time delay.

A third modification of the independent variable involves replacing n by μn, where μ is an integer. We refer to this time-base modification as *time scaling* or *down-sampling*.

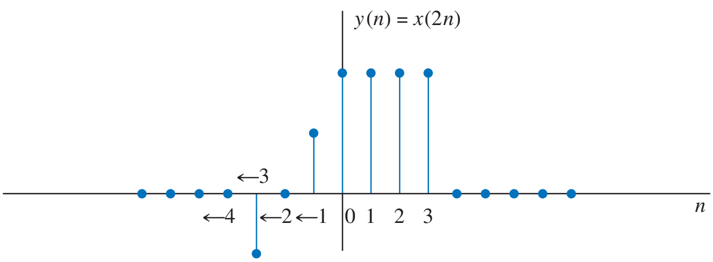
EXAMPLE 2.1.4

Show the graphical representation of the signal $y(n) = x(2n)$, where $x(n)$ is the signal illustrated in Fig. 2.1.11(a).

Solution. We note that the signal $y(n)$ is obtained from $x(n)$ by taking every other sample from $x(n)$, starting with $x(0)$. Thus $y(0) = x(0)$, $y(1) = x(2)$, $y(2) = x(4)$, ... and $y(-1) = x(-2)$, $y(-2) = x(-4)$, and so on. In other words, we have skipped the odd-numbered samples in $x(n)$ and retained the even-numbered samples. The resulting signal is illustrated in Fig. 2.1.11(b).



(a)



(b)

Figure 2.1.11 Graphical illustration of down-sampling operation.

If the signal $x(n)$ was originally obtained by sampling an analog signal $x_a(t)$, then $x(n) = x_a(nT)$, where T is the sampling interval. Now, $y(n) = x(2n) = x_a(2Tn)$. Hence the time-scaling operation described in Example 2.1.4 is equivalent to changing the sampling rate from $1/T$ to $1/2T$, that is, to decreasing the rate by a factor of 2. This is a *down-sampling* operation.

Addition, multiplication, and scaling of sequences. Amplitude modifications include *addition*, *multiplication*, and *scaling* of discrete-time signals.

Amplitude scaling of a signal by a constant A is accomplished by multiplying the value of every signal sample by A. Consequently, we obtain

$$y(n) = Ax(n), \qquad -\infty < n < \infty$$

The *sum* of two signals $x_1(n)$ and $x_2(n)$ is a signal $y(n)$, whose value at any instant is equal to the sum of the values of these two signals at that instant, that is,

$$y(n) = x_1(n) + x_2(n), \qquad -\infty < n < \infty$$

The *product* of two signals is similarly defined on a sample-to-sample basis as

$$y(n) = x_1(n)x_2(n), \qquad -\infty < n < \infty$$

2.2 Discrete-Time Systems

In many applications of digital signal processing we wish to design a device or an algorithm that performs some prescribed operation on a discrete-time signal. Such a device or algorithm is called a discrete-time system. More specifically, a *discrete-time system* is a device or algorithm that operates on a discrete-time signal, called the *input* or *excitation*, according to some well-defined rule, to produce another discrete-time signal called the *output* or *response* of the system. In general, we view a system as an operation or a set of operations performed on the input signal $x(n)$ to produce the output signal $y(n)$. We say that the input signal $x(n)$ is *transformed* by the system into a signal $y(n)$, and express the general relationship between $x(n)$ and $y(n)$ as

$$y(n) \equiv \mathcal{T}[x(n)] \tag{2.2.1}$$

where the symbol $\mathcal{T}$ denotes the transformation (also called an operator) or processing performed by the system on $x(n)$ to produce $y(n)$. The mathematical relationship in (2.2.1) is depicted graphically in Fig. 2.2.1.

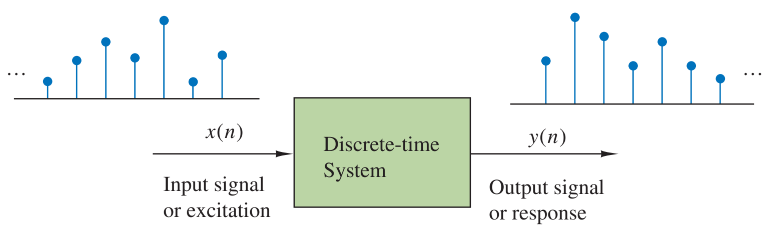


Figure 2.2.1 Block diagram representation of a discrete-time system.

There are various ways to describe the characteristics of the system and the operation it performs on $x(n)$ to produce $y(n)$. In this chapter we shall be concerned with the time-domain characterization of systems. We shall begin with an input–output description of the system. The input–output description focuses on the behavior at the terminals of the system and ignores the detailed internal construction or realization of the system. Later, in Section 2.5 and in Chapter 9, we consider the implementation of discrete-time systems and describe the different structures for their realization.

2.2.1 Input–Output Description of Systems

The input–output description of a discrete-time system consists of a mathematical expression or a rule, which explicitly defines the relation between the input and output signals (*input–output relationship*). The exact internal structure of the system is either unknown or ignored. Thus the only way to interact with the system is by using its input and output terminals (i.e., the system is assumed to be a "black box" to the user). To reflect this philosophy, we use the graphical representation depicted in Fig. 2.2.1, and the general input–output relationship in (2.2.1) or, alternatively, the notation

$$x(n) \xrightarrow{\mathcal{T}} y(n) \tag{2.2.2}$$

which simply means that $y(n)$ is the response of the system $\mathcal{T}$ to the excitation $x(n)$. The following examples illustrate several different systems.

EXAMPLE 2.2.1

Determine the response of the following sytems to the input signal

$$x(n) = \begin{cases} |n|, & -3 \le n \le 3 \\ 0, & \text{otherwise} \end{cases}$$

(a) $y(n) = x(n)$ (identity system)

(b) $y(n) = x(n-1)$ (unit delay system)

(c) $y(n) = x(n+1)$ (unit advance system)

(d) $y(n) = \frac{1}{3}[x(n+1) + x(n) + x(n-1)]$ (moving average filter)

(e) $y(n) = \text{median}\{x(n+1), x(n), x(n-1)\}$ (median filter)

(f) $y(n) = \sum_{k=-\infty}^{n} x(k) = x(n) + x(n-1) + x(n-2) + \cdots$ (accumulator) (2.2.3)

Solution. First, we determine explicitly the sample values of the input signal

$$x(n) = \{\ldots, 0, 3, 2, 1, \underset{\uparrow}{0}, 1, 2, 3, 0, \ldots\}$$

Next, we determine the output of each system using its input–output relationship.

(a) In this case the output is exactly the same as the input signal. Such a system is known as the *identity* system.

(b) This system simply delays the input by one sample. Thus its output is given by

$$y(n) = \{\ldots, 0, 3, 2, \underset{\uparrow}{1}, 0, 1, 2, 3, 0, \ldots\}$$

(c) In this case the system "advances" the input one sample into the future. For example, the value of the output at time $n = 0$ is $y(0) = x(1)$. The response of this system to the given input is

$$y(n) = \{\ldots, 0, 3, 2, 1, 0, \underset{\uparrow}{1}, 2, 3, 0, \ldots\}$$

(d) The output of this system at any time is the mean value of the present, the immediate past, and the immediate future samples. For example, the output at time $n = 0$ is

$$y(0) = \frac{1}{3}[x(-1) + x(0) + x(1)] = \frac{1}{3}[1 + 0 + 1] = \frac{2}{3}$$

Repeating this computation for every value of n, we obtain the output signal

$$y(n) = \{\ldots, 0, 1, \frac{5}{3}, 2, 1, \underset{\uparrow}{\frac{2}{3}}, 1, 2, \frac{5}{3}, 1, 0, \ldots\}$$

(e) This system selects as its output at time n the median value of the three input samples $x(n-1)$, $x(n)$, and $x(n+1)$. Thus the response of this system to the input signal $x(n)$ is

$$y(n) = \{0, 2, 2, 1, 1, \underset{\uparrow}{1}, 2, 2, 0, 0, 0, \ldots\}$$

(f) This system is basically an *accumulator* that computes the running sum of all the past input values up to present time. The response of this system to the given input is

$$y(n) = \{\ldots, 0, 3, 5, 6, \underset{\uparrow}{6}, 7, 9, 12, 12, \ldots\}$$

We observe that for several of the systems considered in Example 2.2.1 the output at time $n = n_0$ depends not only on the value of the input at $n = n_0$ [i.e., $x(n_0)$], but also on the values of the input applied to the system before and after $n = n_0$. Consider, for instance, the accumulator in the example. We see that the output at time $n = n_0$ depends not only on the input at time $n = n_0$, but also on $x(n)$ at times $n = n_0 - 1, n_0 - 2$, and so on. By a simple algebraic manipulation the input–output relation of the accumulator can be written as

$$
\begin{aligned}
y(n) &= \sum_{k=-\infty}^{n} x(k) = \sum_{k=-\infty}^{n-1} x(k) + x(n) \\
&= y(n-1) + x(n)
\end{aligned}
\tag{2.2.4}
$$

which justifies the term *accumulator*. Indeed, the system computes the current value of the output by adding (accumulating) the current value of the input to the previous output value.

There are some interesting conclusions that can be drawn by taking a close look into this apparently simple system. Suppose that we are given the input signal $x(n)$ for $n \geq n_0$, and we wish to determine the output $y(n)$ of this system for $n \geq n_0$. For $n = n_0, n_0 + 1, \ldots$, (2.2.4) gives

$$
\begin{aligned}
y(n_0) &= y(n_0 - 1) + x(n_0) \\
y(n_0 + 1) &= y(n_0) + x(n_0 + 1)
\end{aligned}
$$

and so on. Note that we have a problem in computing $y(n_0)$, since it depends on $y(n_0 - 1)$. However,

$$
y(n_0 - 1) = \sum_{k=-\infty}^{n_0 - 1} x(k)
$$

that is, $y(n_0 - 1)$ "summarizes" the effect on the system from all the inputs which had been applied to the system before time n_0. Thus the response of the system for $n \geq n_0$ to the input $x(n)$ that is applied at time n_0 is the combined result of this input and all inputs that had been applied previously to the system. Consequently, $y(n)$, $n \geq n_0$ is not uniquely determined by the input $x(n)$ for $n \geq n_0$.

The additional information required to determine $y(n)$ for $n \geq n_0$ is the *initial condition* $y(n_0 - 1)$. This value summarizes the effect of all previous inputs to the system. Thus the initial condition $y(n_0 - 1)$ together with the input sequence $x(n)$ for $n \geq n_0$ uniquely determine the output sequence $y(n)$ for $n \geq n_0$.

If the accumulator had no excitation prior to n_0, the initial condition is $y(n_0 - 1) = 0$. In such a case we say that the system is *initially relaxed*. Since $y(n_0 - 1) = 0$, the output sequence $y(n)$ depends only on the input sequence $x(n)$ for $n \geq n_0$.

It is customary to assume that every system is relaxed at $n = -\infty$. In this case, if an input $x(n)$ is applied at $n = -\infty$, the corresponding output $y(n)$ is *solely* and *uniquely* determined by the given input.

EXAMPLE 2.2.2

The accumulator described by (2.2.30) is excited by the sequence $x(n) = nu(n)$. Determine its output under the condition that:

(a) It is initially relaxed [i.e., $y(-1) = 0$].

(b) Initially, $y(-1) = 1$.

Solution. The output of the system is defined as

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{n} x(k) = \sum_{k=-\infty}^{-1} x(k) + \sum_{k=0}^{n} x(k) \\ &= y(-1) + \sum_{k=0}^{n} x(k) \\ &= y(-1) + \frac{n(n+1)}{2} \end{aligned}$$

(a) If the system is initially relaxed, $y(-1) = 0$ and hence

$$y(n) = \frac{n(n+1)}{2}, \qquad n \geq 0$$

(b) On the other hand, if the initial condition is $y(-1) = 1$, then

$$y(n) = 1 + \frac{n(n+1)}{2} = \frac{n^2+n+2}{2}, \qquad n \geq 0$$

2.2.2 Block Diagram Representation of Discrete-Time Systems

It is useful at this point to introduce a block diagram representation of discrete-time systems. For this purpose we need to define some basic building blocks that can be interconnected to form complex systems.

An adder. Figure 2.2.2 illustrates a system (adder) that performs the addition of two signal sequences to form another (the sum) sequence, which we denote as $y(n)$. Note that it is not necessary to store either one of the sequences in order to perform the addition. In other words, the addition operation is *memoryless*.

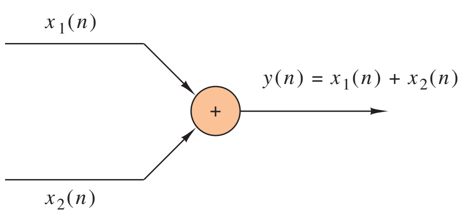


Figure 2.2.2
Graphical representation of an adder.

A constant multiplier. This operation is depicted by Fig. 2.2.3, and simply represents applying a scale factor on the input $x(n)$. Note that this operation is also memoryless.

Figure 2.2.3
Graphical representation of a constant multiplier.

A signal multiplier. Figure 2.2.4 illustrates the multiplication of two signal sequences to form another (the product) sequence, denoted in the figure as $y(n)$. As in the preceding two cases, we can view the multiplication operation as memoryless.

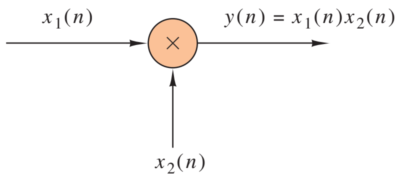


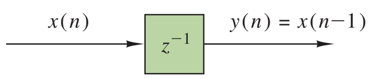
Figure 2.2.4
Graphical representation of a signal multiplier.

A unit delay element. The unit delay is a special system that simply delays the signal passing through it by one sample. Figure 2.2.5 illustrates such a system. If the input signal is $x(n)$, the output is $x(n-1)$. In fact, the sample $x(n-1)$ is stored in memory at time $n-1$ and it is recalled from memory at time n to form

$$y(n) = x(n-1)$$

Thus this basic building block requires memory. The use of the symbol z^{-1} to denote the unit of delay will become apparent when we discuss the z-transform in Chapter 3.

Figure 2.2.5
Graphical representation of the unit delay element.



A unit advance element. In contrast to the unit delay, a unit advance moves the input $x(n)$ ahead by one sample in time to yield $x(n+1)$. Figure 2.2.6 illustrates this operation, with the operator z being used to denote the unit advance. We observe that any such advance is physically impossible in real time, since, in fact, it involves looking into the future of the signal. On the other hand, if we store the signal in the memory of the computer, we can recall any sample at any time. In such a non-real-time application, it is possible to advance the signal $x(n)$ in time.

Figure 2.2.6
Graphical representation of the unit advance element.