PART 1
Background Material

Part I consists of Chapters 1 through 4. In this part of the book we present background material on discrete-time signals and systems and thereby lay a foundation for the rest of the book, as summarized here:

- Chapter 1 reviews fundamentals of discrete-time signal processing, with emphasis on the *z*-transform, the discrete Fourier transform, and the discrete cosine transform.
- Chapter 2 covers the time-domain characteristics of discrete-time stochastic processes.
- Chapter 3 covers the frequency-domain characteristics of discrete-time stochastic processes, with particular emphasis on the notion of a power spectrum or power spectral density. Higher-order statistics and cyclostationary properties of stochastic processes are also discussed here.
- In Chapter 4 we study the eigenvalue problem, which is central to a detailed mathematical description of discrete-time wide-sense stationary processes.

CHAPTER

1

Discrete-time Signal Processing

Typically, a signal of interest is described as a function of time. The transformation of a signal from the time domain into the frequency domain plays a key role in the study of signal processing. The particular transformation used in practice depends on the type of signal being considered. Given the pervasive nature of digital processing and the benefits (flexibility and accuracy of computation) offered by its use, our interest in this book is confined to discrete-time signals. Specifically, the signal is described as a *time series*, consisting of a sequence of uniformly spaced samples whose varying amplitudes carry the useful information content of the signal. In such a situation, the transforms that immediately come to mind are two closely related transforms, namely, the z-transform and the Fourier transform. The Fourier transform is defined in terms of a real variable (frequency), whereas the z-transform is defined in terms of a complex variable.

In this chapter we present a brief review of discrete-time signal processing,[1] beginning with a definition of the z-transform and its properties.

1.1 *z*-TRANSFORM

Consider a time series (sequence) denoted by the samples $u(n)$, $u(n-1)$, $u(n-2)$, . . . , where n denotes *discrete time*. For convenience of presentation, it is assumed that the

[1]For a detailed treatment of the many facets of discrete-time signal processing, see Oppenheim and Schafer (1989).

spacing between adjacent samples of the sequence is unity. The sequence is written as $\{u(n)\}$ or simply $u(n)$. The two-sided *z-transform* of $u(n)$ is defined as

$$\begin{aligned} U(z) &= z[u(n)] \\ &= \sum_{n=-\infty}^{\infty} u(n) z^{-n} \end{aligned} \tag{1.1}$$

where z is a *complex variable.* The first line of Eq. (1.1) describes the z-transform as an "operator," and the second line defines it as an infinite power series in z. The sequence $u(n)$ and its z-transform form a *z-transform pair,* described by

$$u(n) \rightleftharpoons U(z) \tag{1.2}$$

The power series defined in Eq. (1.1) is a *Laurent series,* which features prominently in the functional theory of complex variables; a brief review of complex variable theory is presented in Appendix A. The important point to note here is that for the z-transform $U(z)$ to be meaningful, the power series defined in Eq. (1.1) must be absolutely summable; that is $U(z)$ is uniformly convergent. For any given time series $u(n)$, the set of values of the complex variable z for which the z-transform $U(z)$ is uniformly convergent is referred to as the *region of convergence* (ROC).

Let the region of convergence of the z-transform $U(z)$ be denoted by the annular domain $R_1 < |z| < R_2$. Let $\mathscr{C}$ be a closed contour that encloses the origin and is contained in this region of convergence. Then, given the z-transform $U(z)$, the original time series $u(n)$ may be uniquely recovered using the *z-transform inversion integral formula* (see Appendix A)

$$u(n) = \frac{1}{2\pi j} \oint_{\mathscr{C}} U(z) z^n \frac{dz}{z} \tag{1.3}$$

where the contour integration is performed by transversing the contour $\mathscr{C}$ in the counterclockwise direction.

Properties of the z-Transform

The z-transform is a *linear transform* in that it satisfies the principle of superposition. Specifically, given two sequences $u_1(n)$ and $u_2(n)$ whose z-transforms are denoted by $U_1(z)$ and $U_2(z)$, respectively, we may write

$$a\, u_1(n) + b\, u_2(n) \rightleftharpoons a\, U_1(z) + b\, U_2(z) \tag{1.4}$$

where a and b are scaling factors. The region of convergence, for which Eq. (1.4) holds, contains the intersection of the regions of convergence of $U_1(z)$ and $U_2(z)$.

Another important property of the z-transform is the *time-shifting property.* Let $U(z)$ denote the z-transform of the sequence $u(n)$. The z-transform of $u(n - n_0)$ is described by the relation

$$u(n - n_0) \rightleftharpoons z^{-n_0} U(z) \tag{1.5}$$

where n_0 is an integer. Equation (1.5) holds for the same region of convergence as the original time series $u(n)$, except for a possible addition or deletion of $z = 0$ or $z = \infty$. For the special case of $n_0 = 1$, we see that such a time shift has the effect of multiplying the z-transform $U(z)$ by the factor z^{-1}. It is for this reason that z^{-1} is commonly referred to as a *unit-delay element*.

One other property of the z-transform of particular interest to us is the *convolution theorem*. Let $U_1(z)$ and $U_2(z)$ denote the z-transforms of the time series $u_1(n)$ and $u_2(n)$, respectively. According to the convolution theorem, we have

$$\sum_{i=-\infty}^{\infty} u_1(i)u_2(n-i) \rightleftharpoons U_1(z)U_2(z) \tag{1.6}$$

where the region of convergence includes the intersection of the regions of convergence of $U_1(z)$ and $U_2(z)$. The proof of Eq. (1.6) follows directly from the defining equation (1.1). In other words, convolution of two sequences in the time domain is transformed into multiplication of their z-transforms in the frequency domain.

1.2 LINEAR TIME-INVARIANT FILTERS

The z-transform plays a key role in the study of a particular class of filters known as *linear time-invariant filters*, which are characterized by the following two properties: linearity and time invariance. The *linearity* property means that the filter satisfies the principle of superposition. Specifically, if $v_1(n)$ and $v_2(n)$ are two different *excitations* applied to the filter and $u_1(n)$ and $u_2(n)$ are the *responses* produced by the filter, respectively, then the response of the filter to the composite excitation $a\,v_1(n) + b\,v_2(n)$ is equal to $a\,u_1(n) + b\,u_2(n)$, where a and b are arbitrary constants. The *time-invariance* property means that if $u(n)$ is the response of the filter due to the excitation $v(n)$, then the response of the filter to the new excitation $v(n-k)$ is equal to $u(n-k)$, where k is an arbitrary time shift.

One useful way of describing a linear time-invariant filter is in terms of its *impulse response*, defined as the response of the filter to a unit impulse or delta function applied to the filter at zero time. Let $h(n)$ denote the impulse response of the filter. The response $u(n)$ of the filter produced by an arbitrary excitation $v(n)$ is defined by the *convolution sum*

$$u(n) = \sum_{i=-\infty}^{\infty} h(i)v(n-i) \tag{1.7}$$

Applying the z-transform to both sides of Eq. (1.7) and invoking the convolution theorem, we may write

$$U(z) = H(z)\,V(z) \tag{1.8}$$

where $U(z)$, $V(z)$, and $H(z)$ are the z-transforms of $u(n)$, $v(n)$, and $h(n)$, respectively.

The z-transform $H(z)$ [i.e., the z-transform of the impulse response $h(n)$] is called the *transfer function* of the filter; it provides the basis of another way of describing a linear time-invariant filter. According to Eq. (1.8), we have

$$H(z) = \frac{U(z)}{V(z)} \tag{1.9}$$

Thus, the transfer function $H(z)$ is equal to the ratio of the z-transform of the filter's response to the z-transform of the excitation applied to the filter.

In an important subclass of linear time-invariant filters, the input sequence (excitation) $v(n)$ and the output sequence (response) $u(n)$ are related by a difference equation of order N as follows:

$$\sum_{j=0}^{N} a_j u(n-j) = \sum_{j=0}^{N} b_j v(n-j) \tag{1.10}$$

where the a_j and the b_j are constant coefficients. Applying the z- transform to both sides of Eq. (1.10) and using the time-shifting property of the z-transform, we may readily express the transfer function of the filter as

$$\begin{aligned} H(z) &= \frac{U(z)}{V(z)} \\ &= \frac{\sum_{j=0}^{N} a_j z^{-j}}{\sum_{j=0}^{N} b_j z^{-j}} \end{aligned} \tag{1.11}$$

Equivalently, we may express the rational transfer function of Eq. (1.11) in the factored form

$$H(z) = \frac{a_0}{b_0} \frac{\prod_{k=1}^{N} (1 - c_k z^{-1})}{\prod_{k=1}^{N} (1 - d_k z^{-1})} \tag{1.12}$$

Each factor $(1 - c_k z^{-1})$ in the numerator on the right-hand side of Eq. (1.12) contributes a zero at $z = c_k$ and a pole at $z = 0$, whereas each factor $(1 - d_k z^{-1})$ in the denominator contributes a pole at $z = d_k$ and a zero at $z = 0$. Thus, except for the scaling factor a_0/b_0, the transfer function $H(z)$ of the filter is uniquely defined in terms of its poles and zeros. Note that with the time-domain behavior of the filter defined by a constant-coefficient difference equation of the form given in Eq. (1.10), the poles and zeros of the transfer function $H(z)$ are real or else appear in complex-conjugate pairs.

Based on the representation given in Eq. (1.12), we may distinguish between two distinct types of linear time-invariant filters:

1. *Finite-duration impulse response (FIR) filters.* For this type of filter, d_k is zero for all k, which means that the filter is an *all-zero filter* in that the poles of its transfer function $H(z)$ are all confined to $z = 0$. Correspondingly, the impulse response $h(n)$ of the filter has a finite duration; hence the descriptor "finite-duration impulse response."
2. *Infinite-duration impulse response (IIR) filters.* In this second type of filter, the transfer function $H(z)$ has at least one nonzero pole that is not canceled by a zero. Correspondingly, the impulse response $h(n)$ of the filter has an infinite duration; hence the descriptor "infinite-duration impulse response." When c_k is zero for all k, the IIR filter is said to be an *all pole filter*, in that the zeros of its transfer function $H(z)$ are all confined to $z = 0$.

Figures 1.1(a) and 1.1(b) show examples of FIR and IIR filters, respectively. The boxes labeled z^{-1} represent unit-delay elements, and the circles labeled $a_1, a_2, \ldots, a_N$ represent filter coefficients. Note that the FIR filter of Fig. 1.1(a) involves feedforward paths only, whereas the IIR filter of Fig. 1.1(b) involves both feedforward and feedback paths. In both cases, the basic functional blocks needed to build the filters consist of unit-delay elements, multipliers, and adders.

Causality and Stability

A linear time-invariant filter is said to be *causal* if its impulse response $h(n)$ is zero for negative time, as shown by

$$h(n) = 0 \qquad \text{for } n < 0 \tag{1.13}$$

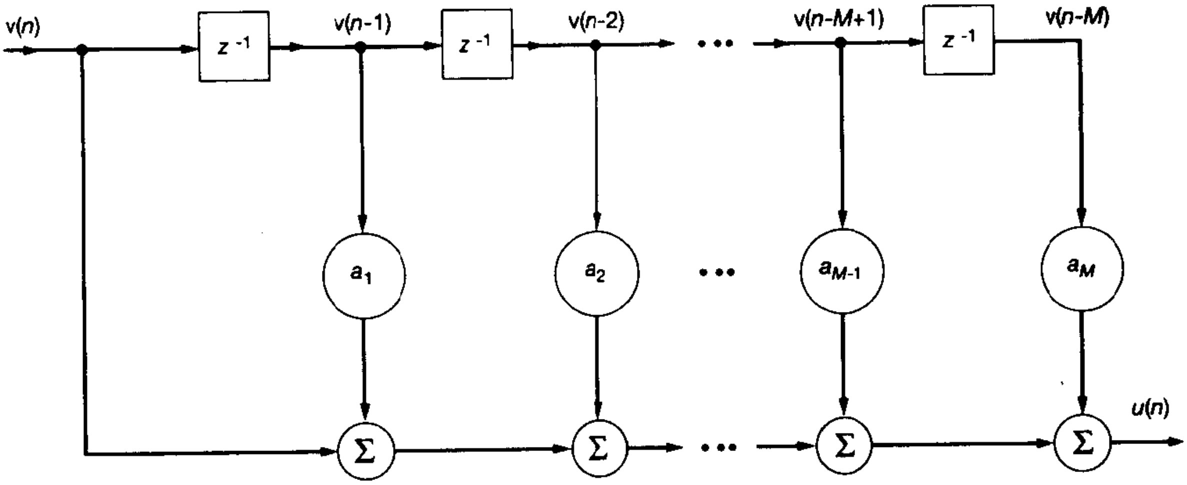
Clearly, for a filter to operate in real time, it would have to be causal. However, causality is not a necessary requirement for physical realizability. There are many applications in which the signal to be processed is available in stored form; in these situations, the filter can be noncausal and yet physically realizable.

The filter is said to be *stable* if the output sequence (response) of the filter is bounded for all bounded input sequences (excitations). This requirement is called the *bounded input–bounded output (BIBO) stability criterion*, the application of which is well suited for linear time-invariant filters. From Eq. (1.7) we readily see that the necessary and sufficient condition for BIBO stability is

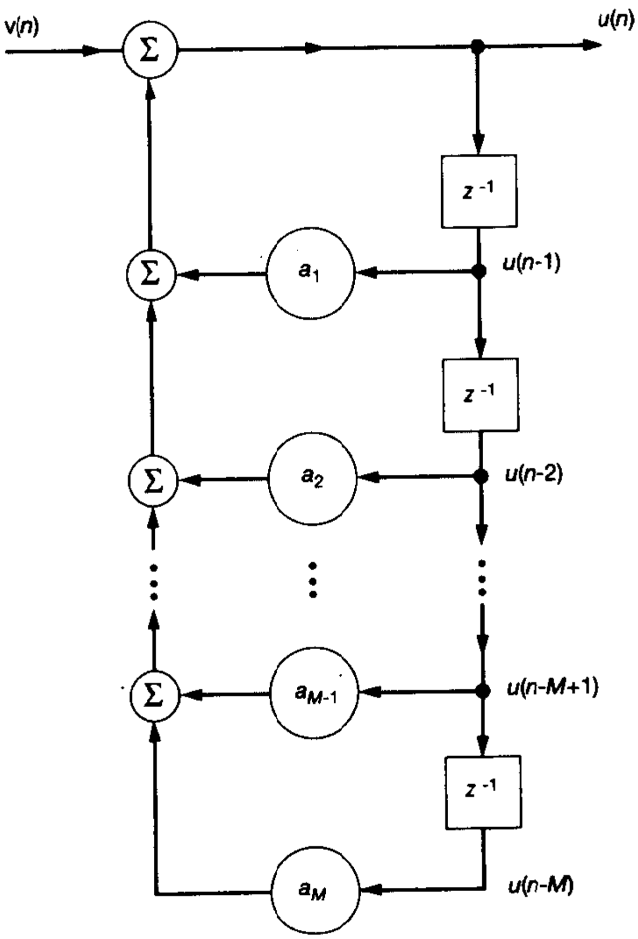
$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty \tag{1.14}$$

That is, the impulse response of the filter must be absolutely summable.

Causality and stability are not necessarily compatible requirements. For a linear time-invariant filter defined by the difference equation (1.10) to be both causal and stable,



(a) FIR filter



(b) IIR filter

Figure 1.1 Two basic types of filters.

the region of convergence of the filter's transfer function $H(z)$ must satisfy two requirements (Oppenheim and Schafer, 1989):

1. It must lie outside the outermost poles of $H(z)$.
2. It must include the unit circle in the z-plane.

Clearly, these requirements can only be satisfied if all the poles of $H(z)$ lie inside the unit circle, as indicated in Fig. 1.2. We may thus make the following important statement on the issue of stability: *A causal, linear time-invariant filter is stable if and only if all of the*

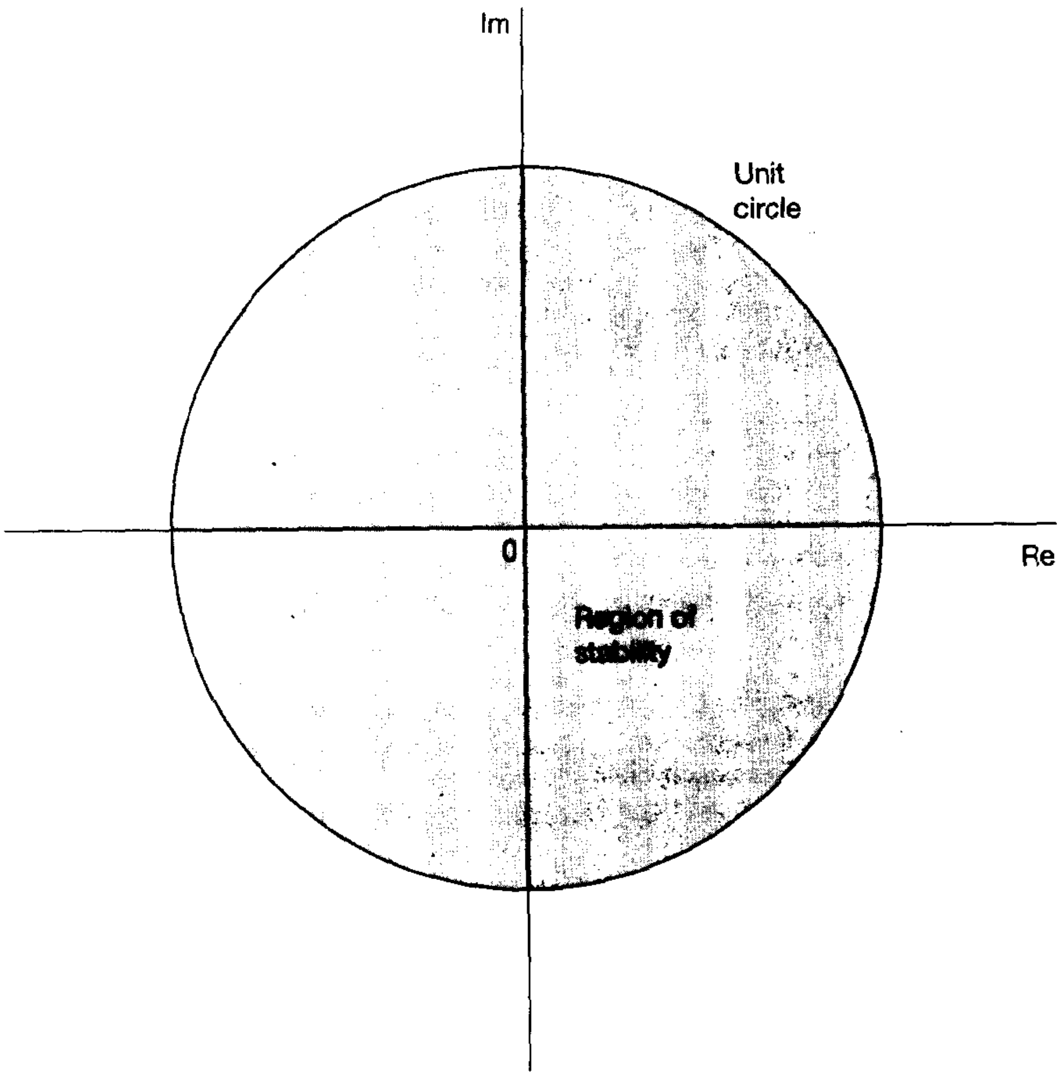


Figure 1.2 z-plane.

poles of the filter's transfer function lie inside the unit circle in the z-plane. Note that this statement says nothing about the zeros of the filter's transfer function $H(z)$. Insofar as causality and stability are concerned, the zeros of $H(z)$ can indeed lie anywhere in the z-plane.

1.3 MINIMUM-PHASE FILTERS

The unit circle plays a critical role not only in the stability criterion of a causal filter, but also in the evaluation of its frequency response. Specifically, setting

$$z = e^{j2\pi f}$$

in the expression for the transfer function $H(z)$, we get the filter's *frequency response* denoted by $H(e^{j2\pi f})$, where f denotes the frequency in Hertz. Expressing $H(e^{j2\pi f})$ in its polar form, we may define the frequency response of the filter in terms of two components:

- The *magnitude (amplitude) response*, denoted by $|H(e^{j2\pi f})|$
- The *phase response*, denoted by ang $(H(e^{j2\pi f}))$

In the case of a special class of filters known as *minimum-phase filters*, the magnitude response and phase response of the filter are uniquely related to each other, in that if we are given one of them, we can compute the other component uniquely (Oppenheim and Schafer, 1989). A minimum-phase filter derives its name from the fact that, for a specified magnitude response, it has the minimum phase response possible for all values of z on the unit circle.

The minimum-phase property of a linear time-invariant filter places restrictions of its own on possible locations of the zeros of the filter's transfer function $H(z)$. Specifically, the zeros of $H(z)$ must satisfy the following requirements:

- The zeros of $H(z)$ may lie anywhere inside the unit circle in the z-plane.
- Zeros are permitted to lie on the unit circle, provided that they are *simple* (i.e., they are of order one).

A minimum-phase filter has the following interesting property: given a minimum-phase filter of transfer function $H(z)$, we may define an *inverse filter* with transfer function $1/H(z)$ that is both causal and stable, provided that $H(z)$ does not have zeros on the unit circle. The cascade connection of such a pair of filters has a transfer function equal to unity.

Finally, we note that a *nonminimum-phase filter*, whose transfer function $H(z)$ has zeros outside the unit circle, can always be treated as the cascade connection of a minimum-phase filter and an all-pass filter. An *all-pass filter* is defined as a filter whose transfer function has poles and zeros that are the reciprocals of each other with respect to the unit circle; naturally, the poles are confined to the interior of the unit circle, in which case all the zeros are confined to the exterior of the unit circle. Consequently, the magnitude response of an all-pass filter is equal to unity, which means that it passes all the frequency components of the input signal with no change in amplitude. When the nonminimum-phase filter has all of its zeros located outside the unit circle, it is said to be a *maximum-phase filter*.

1.4 DISCRETE FOURIER TRANSFORM

The *Fourier transform* of a sequence is readily obtained from its z-transform simply by setting the complex variable z equal to $\exp(j2\pi f)$, where f is the real frequency variable. When the sequence of interest has a finite duration, we may go one step further and develop a Fourier representation for it by defining the *discrete Fourier transform (DFT)*. The DFT is itself made up of a sequence of samples, uniformly spaced in frequency. The DFT has established itself as a powerful tool in digital signal processing by virtue of the fact that there exist efficient algorithms for its numerical computation; these algorithms are known collectively as *fast Fourier transform (FFT) algorithms* (Oppenheim and Schafer, 1989).

Consider a finite-duration sequence $u(n)$, assumed to be of length N. The DFT of $u(n)$ is defined by

$$U(k) = \sum_{n=0}^{N-1} u(n) \exp\left(-\frac{j2\pi kn}{N}\right), \qquad k = 0, \ldots, N-1 \tag{1.15}$$

The *inverse discrete Fourier transform* (IDFT) of $U(k)$ is defined by

$$u(n) = \frac{1}{N} \sum_{k=0}^{N-1} U(k) \exp\left(\frac{j2\pi kn}{N}\right), \qquad n = 0, 1, \ldots, N-1 \tag{1.16}$$

Note that both the original sequence $u(n)$ and its DFT $U(k)$ are of the same length, N. We thus speak of the discrete Fourier transform as an "N-point DFT."

The discrete Fourier transform has an interesting interpretation in terms of the z-transform, as described here: the DFT of a finite-duration sequence may be obtained by evaluating the z-transform of that same sequence at N points uniformly spaced on the unit circle in the z-plane. This "sampling" process is illustrated in Fig. 1.3 for $N = 8$.

Though the sequence $u(n)$ and its DFT $U(k)$ are defined as "finite-length" sequences, in reality they both represent a single period of their respective periodic sequences. This double periodicity is the direct consequence of sampling a continuous-time signal as well as its continuous Fourier transform.

1.5 IMPLEMENTING CONVOLUTIONS USING THE DFT

The underlying "double-periodic" nature of the discrete Fourier transform just mentioned imparts to it certain properties that distinguish it from the continuous Fourier transform. In particular, the *linear convolution* of two sequences, $h(n)$ and $v(n)$, say, involves multiplying one sequence by a time-reversed and linearly shifted version of the other sequence and

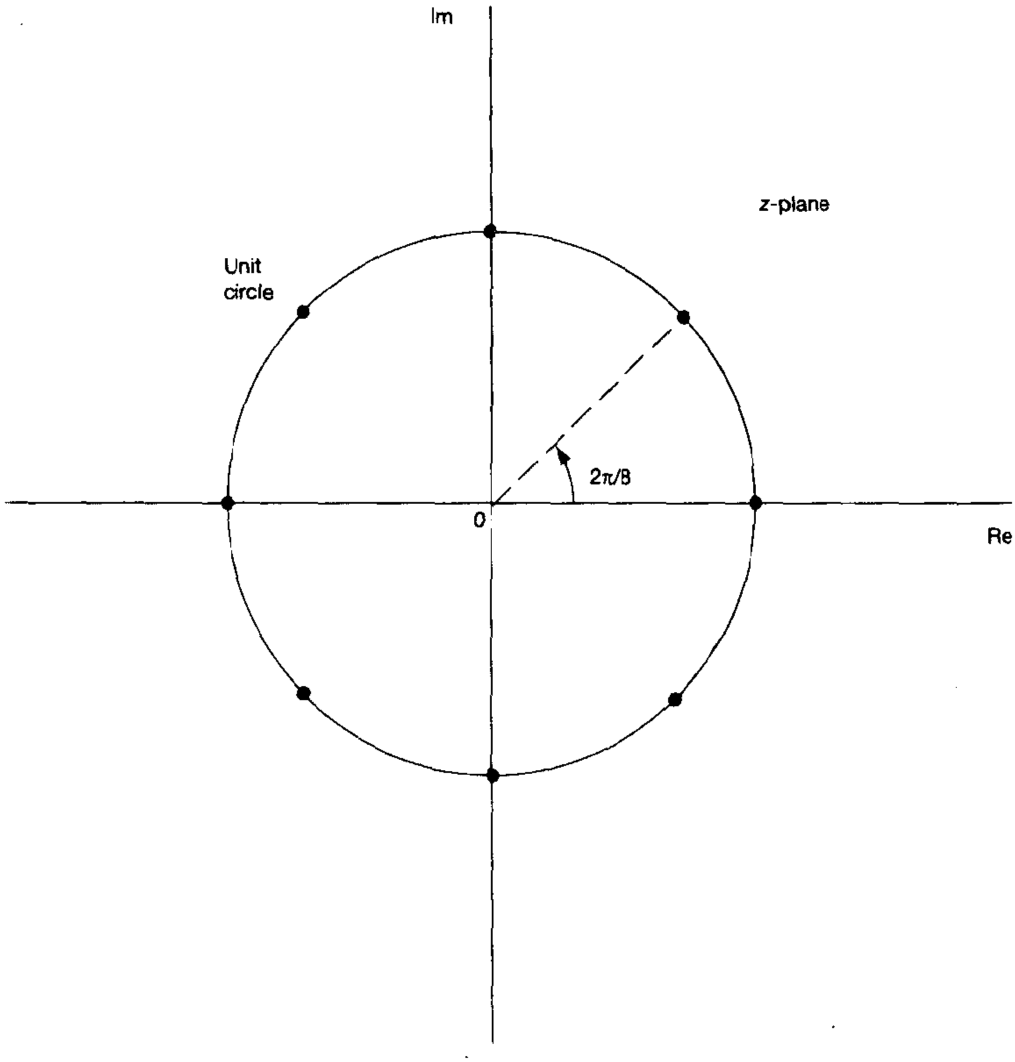


Figure 1.3 A set of N (=8) uniformly spaced points on the unit circle in the z-plane.

then summing the product $h(i)v(n - i)$ over all i, as described in Eq. (1.7). In contrast, in the case of DFT we have a *circular convolution* in which the second sequence is circularly time-reversed and circularly shifted with respect to the first sequence. In other words, in circular convolution both sequences have length N (or less) and the sequences are shifted modulo N. It is only when convolution is defined in this way that the convolution of two sequences in the time domain is transformed into the product of their DFTs in the frequency domain (Oppenheim and Schafer, 1989). Stating this property in another way, if we multiply the DFTs of two finite-duration sequences and then evaluate the IDFT of the product, the result so obtained is equivalent to a circular convolution of the original sequences.

With circular convolution being markedly different from linear convolution, the key issue is how to use the DFT to perform linear convolution. To illustrate how we may do this, consider two sequences $v(n)$ and $h(n)$, assuming that they are of lengths L and P, respectively. The linear convolution of these two sequences is a finite-duration sequence of length $L + P - 1$. Recognizing that the convolution of two periodic sequences is another periodic sequence of the same period, we may proceed as follows:

- Append an appropriate number of zero-valued samples to $v(n)$ and $h(n)$ to make them both N-point sequences, where $N = L + P - 1$; this process is referred to as *zero padding*.
- Compute the N-point DFTs of the appended versions of the sequences $v(n)$ and $h(n)$, multiply the DFTs, and then compute the IDFT of the product.
- Use one period of the circular convolution so computed as the linear convolution of the original sequences $v(n)$ and $h(n)$.

The procedure described here works perfectly well for finite-duration sequences. But, what about linear filtering applications where the input signal is, for all practical purposes, of infinite duration? In situations of this kind, we may use two widely used techniques known as the *overlap-add* and *overlap-save* sectioning methods, which are described next.

Overlap-Add Method

The best way to explain the overlap-add method is by way of an example. Consider the sequences $v(n)$ and $h(n)$ shown in Fig. 1.4; it is assumed that the sequence $v(n)$ is effectively of "infinite" length, and the sequence $h(n)$ is of some finite length P. The sequence $v(n)$ is first sectioned into nonoverlapping blocks, each of length $Q = N - P$ for some predetermined N, as illustrated in Fig. 1.5(a). It may therefore be represented as the sum of shifted finite-duration sequences, as shown by

$$v(n) = \sum_{r=0}^{\infty} v_r(n) \tag{1.17}$$

where

$$v_r(n) = \begin{cases} v(n + rQ), & n = 0,1,\ldots,Q-1 \\ 0, & \text{otherwise} \end{cases} \tag{1.18}$$

Next, each section is padded with $P - 1$ zero-valued samples to form one period of a periodic sequence, as illustrated in Fig. 1.5(a). We may thus describe the first section by writing

$$v_0(n) = \begin{cases} v(n), & n = 0,1,\ldots,N-P \\ 0, & n = N-P+1,\ldots,N-1 \end{cases} \tag{1.19}$$

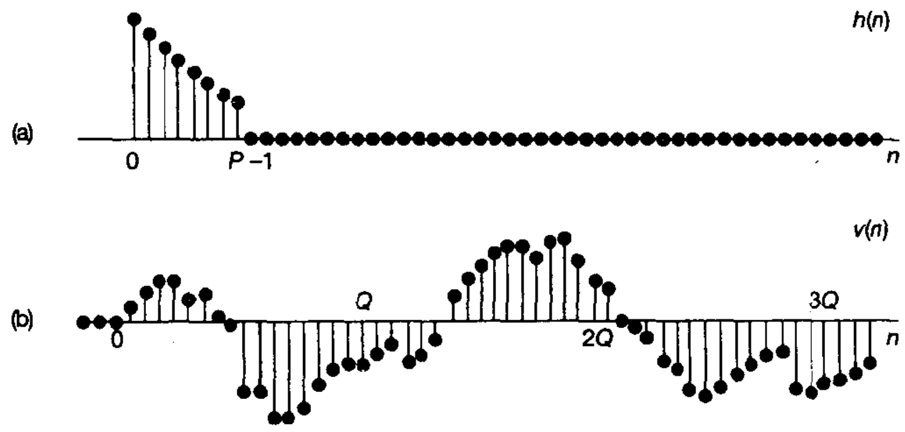


Figure 1.4 Finite-length impulse response $h(n)$ and indefinite-length signal $v(n)$ to be filtered by $h(n)$ (reproduced, with permission, from Oppenheim and Schafer, 1989).

The circular convolution of $v_0(n)$ with $h(n)$ yields the output sequence $u_0(n)$ shown in the first trace of Fig. 1.5(b).

The second section $v_1(n)$ and all other sections of the "infinitely" long sequence $v(n)$ are treated in a similar manner. The resulting output sequences $u_1(n)$, and $u_2(n)$ are also illustrated in Fig. 1.5(b) for the input sections $v_1(n)$ and $v_2(n)$, respectively. Finally, the output sequences $u_0(n)$, $u_1(n)$, $u_2(n)$, . . . are combined to yield the overall output sequence $u(n)$. Note that $u_1(n)$, $u_2(n)$, . . . are shifted by the appropriate values, namely, N, $2N$, . . . , before they are added to $u_0(n)$. The sectioned convolution technique described here is called the *overlap-add method* for two reasons: the output sequences tend to overlap each other, and they are added together to produce the correct result.

Overlap-Save Method

The overlap-save method differs from the overlap-add method in that it involves overlapping input sections rather than output sections. Specifically, the "infinitely" long sequence is sectioned into N-point blocks that overlap by $P - 1$ samples, where P is the length of the "short" sequence $h(n)$, as illustrated in Fig. 1.6(a). The N-point circular convolution of $h(n)$ and $v_r(n)$ is computed for $r = 0,1,2,\ldots$. The resulting output sequences $u_0(n)$, $u_1(n)$, and $u_2(n)$ for the sections $v_0(n)$, $v_1(n)$, and $v_2(n)$ are illustrated in Fig. 1.6(b). The first $P - 1$ samples of each output sequence $u_r(n)$, $r = 0,1,2,\ldots$ are ignored, because they are due to the wraparound (end) effect of the circular convolution. Finally, the remaining samples of the output sequences $u_0(n)$, $u_1(n)$, $u_2(n)$, . . . are added after they have been shifted by appropriate values, yielding the correct output sequence $u(n)$. For obvious reasons, this second sectioning technique is referred to as the *overlap-save method*.

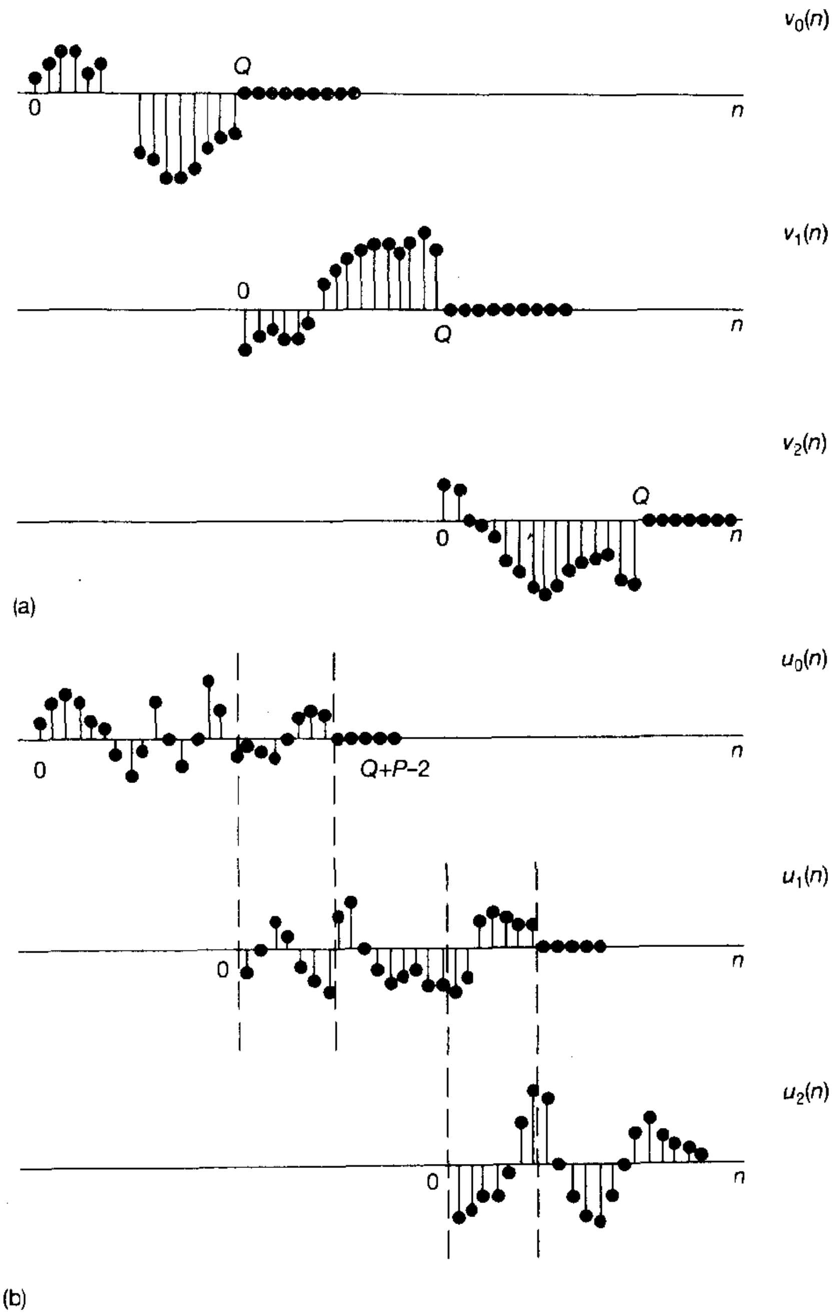


Figure 1.5 (a) Decomposition of the input signal $v(n)$ in Fig. 1.4 into nonoverlapping sections, each of length Q. (b) Result of convolving each such section with $h(n)$ (reproduced, with permission, from Oppenheim and Schafer, 1989).

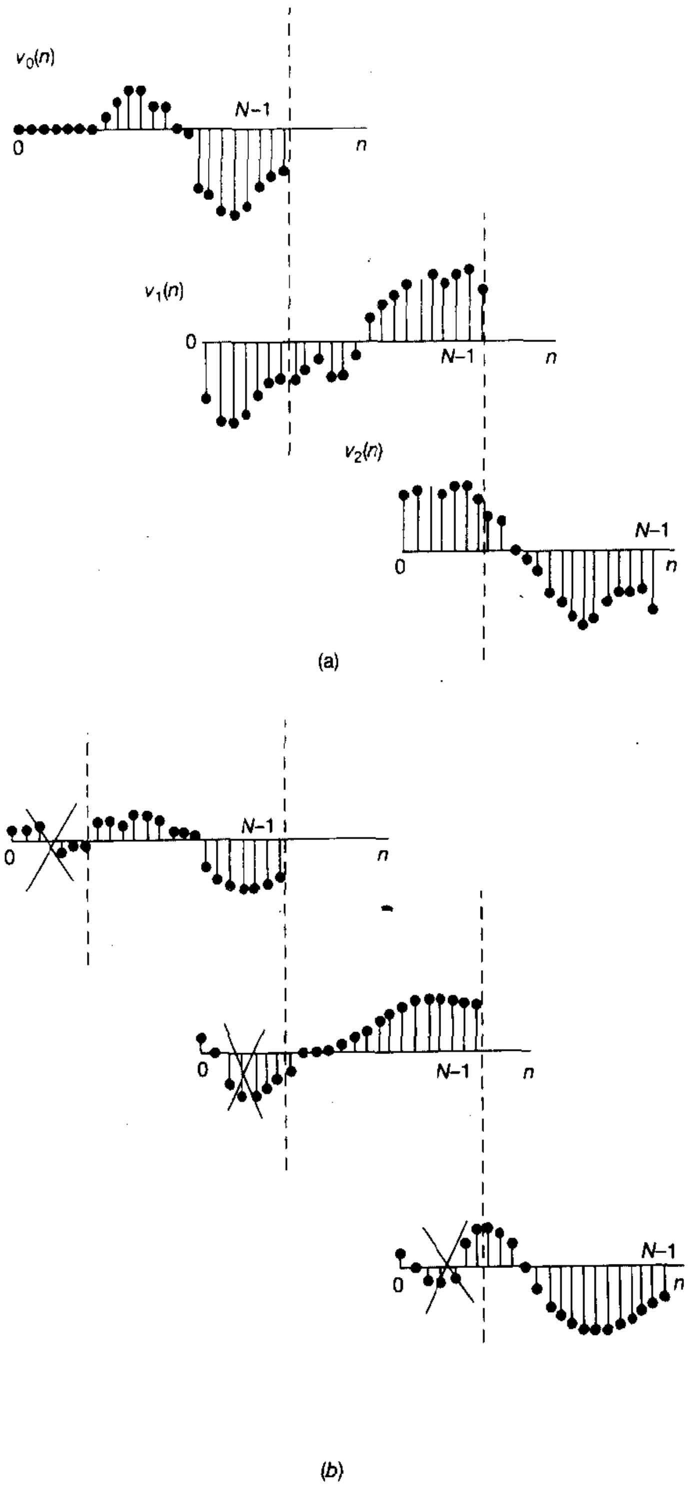


Figure 1.6 (a) Decomposition of the input signal $v(n)$ in Fig. 1.4 into overlapping sections, each of length N. (b) Result of convolving each section with $h(n)$; the portions of each filtered section to be discarded in forming the linear convolution are indicated (reproduced, with permission, from Oppenheim and Schafer, 1989).

Thus, we may use the overlap-add method or the overlap-save method to compute the linear convolution of a short sequence $h(n)$ with a much longer sequence $v(n)$ by first sectioning the latter sequence into small blocks, then indirectly computing the circular convolution of each such block with the short sequence $h(n)$ via the DFT, and finally piecing the individual results together in an appropriate fashion. The utility of the overlap-add and overlap-save methods is made a practical reality by virtue of the availability of highly efficient algorithms (i.e., FFT algorithms) for computing the DFT. The indirect computation of convolution using the overlap-add method or overlap-save method via the FFT is referred to as *fast convolution*, as it is faster than its direct computation.

1.6 DISCRETE COSINE TRANSFORM

Another transform that features in certain applications of digital signal processing is the *discrete cosine transform* (DCT). Unlike the DFT, the DCT may be defined in several different ways (Rao and Yip, 1990). For the purpose of our present discussion, the DCT of an N-point sequence $u(n)$ is defined by

$$U(m) = k_m \sum_{n=0}^{N-1} u(n)\cos\left(\frac{(2n+1)m\pi}{2N}\right), \qquad m = 0, 1, \ldots, N-1 \tag{1.20}$$

and the *inverse discrete cosine transform* (IDCT) of $U(m)$ is defined by

$$u(n) = \frac{2}{N} \sum_{m=0}^{N-1} k_m U(m)\cos\left(\frac{(2n+1)m\pi}{2N}\right), \qquad n = 0, 1, \ldots, N-1 \tag{1.21}$$

The constant k_m in Eqs. (1.20) and (1.21) is itself defined by

$$k_m = \begin{cases} 1/\sqrt{2}, & m = 0 \\ 1, & m = 1, \ldots, N-1 \end{cases} \tag{1.22}$$

The DCT is related to the DFT, as one would expect. Specifically, we first construct a $2N$-point sequence $\tilde{u}(n)$ related to the original sequence $u(n)$ as follows:

$$\tilde{u}(n) = \begin{cases} u(n), & n = 0, 1, \ldots, N-1 \\ u(2N-n-1), & n = N, N+1, \ldots, 2N-1 \end{cases} \tag{1.23}$$

Thus, the $\tilde{u}(n)$ is an even extension of $u(n)$. The $2N$-point DFT of the sequence $\tilde{u}(n)$ is given by

$$\begin{aligned} \tilde{U}(m) &= \sum_{n=0}^{2N-1} \tilde{u}(n)\exp\left(-\frac{j2\pi mn}{2N}\right) \\ &= \sum_{n=0}^{N-1} \tilde{u}(n)\exp\left(-\frac{j2\pi mn}{2N}\right) + \sum_{n=N}^{2N-1} \tilde{u}(n)\exp\left(-\frac{j2\pi mn}{2N}\right) \end{aligned} \tag{1.24}$$

Substituting Eq. (1.23) into (1.24), we get

$$\begin{aligned}\tilde{U}(m) &= \sum_{n=0}^{N-1} u(n)\exp\left(-\frac{j2\pi mn}{2N}\right) + \sum_{n=N}^{2N-1} u(2N-n-1)\exp\left(-\frac{j2\pi mn}{2N}\right) \\ &= \sum_{n=0}^{N-1} u(n)\left[\exp\left(-\frac{j2\pi mn}{2N}\right) + \exp\left(\frac{2j\pi m(n+1)}{2N}\right)\right]\end{aligned} \tag{1.25}$$

Introducing the phase shift $m\pi/2N$ and the weighting factor $k_m/2$ into Eq. (1.25), we may write

$$\begin{aligned}\frac{1}{2}k_m\exp\left(-\frac{jm\pi}{2N}\right)\tilde{U}(m) &= \frac{1}{2}k_m\sum_{n=0}^{N-1} u(n)\left[\exp\left(-\frac{j(2n+1)m\pi}{2N}\right) + \exp\left(\frac{j(2n+1)m\pi}{2N}\right)\right] \\ &= k_m\sum_{n=0}^{N-1} u(n)\cos\left(\frac{(2n+1)m\pi}{2N}\right)\end{aligned} \tag{1.26}$$

The right-hand side of Eq. (1.26) is recognized as the definition for the DCT of the original sequence $u(n)$. It follows, therefore, that the discrete cosine transform $U(m)$ of the sequence $u(n)$ and the discrete Fourier transform $\tilde{U}(m)$ of its extended version $\tilde{u}(n)$ are related as follows:

$$U(m) = \frac{1}{2}k_m\exp\left(-\frac{jm\pi}{2N}\right)\tilde{U}(m), \qquad m = 0, 1, \ldots, N-1 \tag{1.27}$$

This relation shows that, whereas the DFT is periodic with period N, the DCT is periodic with period $2N$.

1.7 SUMMARY AND DISCUSSION

In this chapter we reviewed the z-transform, the discrete Fourier transform, and the discrete cosine transform; these transforms are all related to each other. The discrete Fourier transform represents an important example of a general class of finite-length orthogonal transforms, which may be defined by the following pair of relations:

$$U(k) = \sum_{n=0}^{N-1} u(n)\varphi_k^*(n), \qquad k = 0, 1, \ldots, N-1$$

$$u(n) = \frac{1}{N}\sum_{k=0}^{N-1} U(k)\varphi_k(n), \qquad n = 0, 1, \ldots, N-1$$

where $u(n)$ is the given sequence and $U(k)$ is its discrete transform. The sequences $\varphi_k(n)$ for different k constitute an *orthogonal set*, as shown by

$$\sum_{n=0}^{N-1} \varphi_k(n)\varphi_l^*(n) = \begin{cases} N, & l = k \\ 0, & \text{otherwise} \end{cases}$$

Our interest in the z-transform is motivated by the fact it provides a basic tool for the characterization of linear time-invariant systems, which constitute an important class of systems of particular interest in the study of linear adaptive filtering. As for the discrete Fourier transform and the discrete cosine transform, they provide the necessary tools for the implementation of adaptive filtering in the frequency domain, which is sometimes found to be preferable to adaptive filtering in the time domain, an issue that is discussed later in this book.

PROBLEMS

1. An all-pole filter is characterized by the second-order difference equation:

$$u(n) - 0.1\,u(n-1) - 0.8\,u(n-2) = v(n)$$

   (a) Determine the transfer function $H(z)$ of the filter.
   (b) Plot the pole-zero map of $H(z)$.
   (c) Find the impulse response of the filter.
2. The inverse of the filter described in Problem 1 consists of an all-zero filter.
   (a) Plot the pole-zero map of the transfer function for this inverse filter.
   (b) Find the difference equation that describes the time-domain behavior of the inverse filter.
   (c) Find the impulse response of the inverse filter.
3. A second-order nonminimum phase system has the transfer function

$$H(z) = \frac{2(1 + z^{-1} - 2z^{-2})}{1 - 0.2828\,z^{-1} + z^{-2}}.$$

   (a) Plot a pole-zero map for $H(z)$.
   (b) The system described here may be considered to be equivalent to the cascade connection of a minimum phase system and an all-pass system. Determine the transfer functions of these two systems, and plot their individual pole-zero maps.
4. When considering the inverse of a minimum-phase system characterized by the transfer function $H(z)$, it is not permissible for $H(z)$ to have any zero on the unit circle in the z-plane. Why?
5. Convolution, be it linear or circular, is a commutative operation. Demonstrate this property.
6. If $U(e^{j2\pi f})$ is the Fourier transform of a finite-duration sequence $u(n)$, the Fourier transform of the time-shifted sequence $u(n-m)$ is $e^{-j2\pi mf}\,U(e^{j2\pi f})$. How is the corresponding time-shifting property of the discrete Fourier transform for the sequence $u(n)$ described?