Chapter 3: Time-Domain Analysis of Discrete-Time Systems

In this chapter we introduce the basic concepts of discrete-time signals and systems. We shall study convolution method of analysis of linear, time-invariant, discrete-time (LTID) systems. Classical methods of analysis of these systems will also be examined.

3.1 INTRODUCTION

Discrete-time signal is basically a sequence of numbers. Such signals arise naturally in inherently discrete-time situations such as population studies, amortization problems, national income models, and radar tracking. They may also arise as a result of sampling continuous-time signals in sampled data systems and digital filtering. Such signals can be denoted by $x[n]$, $y[n]$, and so on, where the variable n takes integer values, and $x[n]$ denotes the nth number in the sequence labeled x. In this notation, the discrete-time variable n is enclosed in square brackets instead of parentheses, which we have reserved for enclosing continuous-time variable, such as t.

Systems whose inputs and outputs are discrete-time signals are called *discrete-time systems.* A digital computer is a familiar example of this type of system. A discrete-time signal is a sequence of numbers, and a discrete-time system processes a sequence of numbers $x[n]$ to yield another sequence $y[n]$ as the output.[†]

A discrete-time signal, when obtained by uniform sampling of a continuous-time signal $x(t)$, can also be expressed as $x(nT)$, where T is the sampling interval and n, the discrete variable taking on integer values. Thus, $x(nT)$ denotes the value of the signal $x(t)$ at $t = nT$. The signal $x(nT)$ is a sequence of numbers (sample values), and hence, by definition, is a discrete-time signal. Such a signal can also be denoted by the customary discrete-time notation $x[n]$, where $x[n] = x(nT)$. A typical discrete-time signal is depicted in Fig. 3.1, which shows both forms of notation. By way of an example, a continuous-time exponential $x(t) = e^{-t}$, when sampled every $T = 0.1$ second, results in a discrete-time signal $x(nT)$ given by

$$x(nT) = e^{-nT} = e^{-0.1n}$$

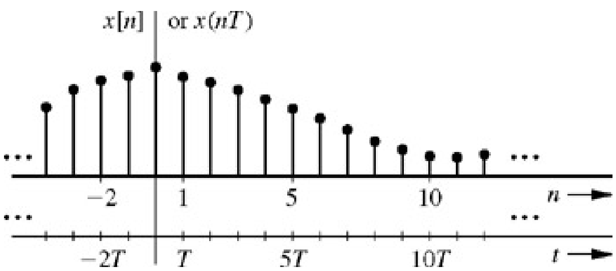

Figure 3.1: A discrete-time signal.

Clearly, this signal is a function of n and may be expressed as $x[n]$. Such representation is more convenient and will be followed throughout this book even for signals resulting from sampling continuous-time signals.

Digital filters, can process continuous-time signals by discrete-time systems, using appropriate interfaces at the input and the output, as illustrated in Fig. 3.2. A continuous-time signal $x(t)$ is first sampled to convert it into a discrete-time signal $x[n]$, which is then processed by a discrete-time system to yield the output $y[n]$. A continuous-time signal $y(t)$ is finally constructed from $y[n]$. We shall use the notations C/D and D/C for conversion from continuous to discrete-time and from discrete to continuous time. By using the interfaces in this manner, we can use an appropriate discrete-time system to process a continuous-time signal. As we shall see later in our discussion, discrete-time systems have several advantages over continuous-time systems. For this reason, there is an accelerating trend toward processing continuous-time signals with discrete-time systems.

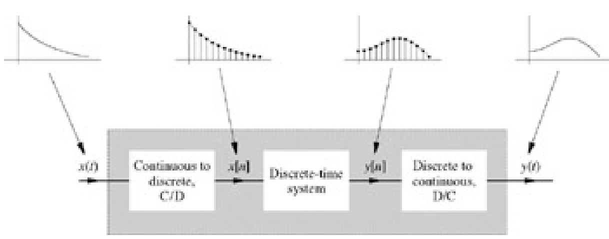

Figure 3.2: Processing a continuous-time signal by means of a discrete-time system.

3.1-1 Size of a Discrete-Time Signal

Arguing along the lines similar to those used for continuous-time signals, the size of a discrete-time signal $x[n]$ will be measured by its energy E_x, defined by

$$E_x = \sum_{n=-\infty}^{\infty} |x[n]|^2 \qquad (3.1)$$

This definition is valid for real or complex $x[n]$. For this measure to be meaningful, the energy of a signal must be finite. A necessary condition for the energy to be finite is that the signal amplitude must $\to 0$ as $|n| \to \infty$. Otherwise the sum in Eq. (3.1) will not converge. If E_x is finite, the signal is called an *energy signal*.

In some cases, for instance, when the amplitude of $x[n]$ does not $\to 0$ as $|n| \to \infty$, then the signal energy is infinite, and a more meaningful measure of the signal in such a case would be the time average of the energy (if it exists), which is the signal power P_x, defined by

$$P_x = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{-N}^{N} |x[n]|^2 \qquad (3.2)$$

In this equation, the sum is divided by $2N + 1$ because there are $2N + 1$ samples in the interval from $-N$ to N. For periodic signals, the time averaging need be performed over only one period in view of the periodic repetition of the signal. If P_x is finite and nonzero, the signal is called a *power signal*. As in the continuous-time case, a discrete-time signal can either be an energy signal or a power signal, but cannot be both at the same time. Some signals are neither energy nor power signals.

EXAMPLE 3.1

Find the energy of the signal $x[n] = n$, shown in Fig. 3.3a and the power for the periodic signal $\gamma[n]$ in Fig. 3.3b.

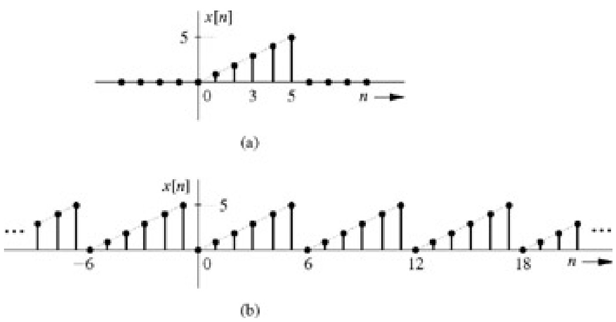

Figure 3.3: (a) Energy and **(b)** power computation for a signal.

By definition

$$E_x = \sum_{n=0}^{5} n^2 = 55$$

A periodic signal $x[n]$ with period N_0 is characterized by the fact that

$$x[n] = x[n + N_0]$$

The smallest value of N_0 for which the preceding equation holds is the *fundamental period*. Such a signal is called N_0 *periodic*. Figure 3.3b shows an example of a periodic signal $\gamma[n]$ of period $N_0 = 6$ because each period contains 6 samples. Note that if the first sample is taken at $n = 0$, the last sample is at $n = N_0 - 1 = 5$, not at $n = N_0 = 6$. Because the signal $\gamma[n]$ is periodic, its power P_y can be found by averaging its energy over one period. Averaging the energy over one period, we obtain

$$P_y = \frac{1}{6} \sum_{n=0}^{5} n^2 = \frac{55}{6}$$

EXERCISE E3.1

Show that the signal $x[n] = a^n u[n]$ is an energy signal of energy $E_x = 1/(1 - |a|^2)$ if $|a| < 1$. It is a power signal of power $P_x = 0.5$ if $|a| = 1$. It is neither an energy signal nor a power signal if $|a| > 1$.

[†]There may be more than one input and more than one output.

3.2 USEFUL SIGNAL OPERATIONS

Signal operations for *shifting*, and *scaling*, as discussed for continuous-time signals also apply to discrete-time signals with some modifications.

SHIFTING

Consider a signal $x[n]$ (Fig. 3.4a) and the same signal delayed (right-shifted) by 5 units (Fig. 3.4b), which we shall denote by $x_s[n]$.[†] Using the argument employed for a similar operation in continuous-time signals (Section 1.2), we obtain

$$x_s[n] = x[n-5]$$

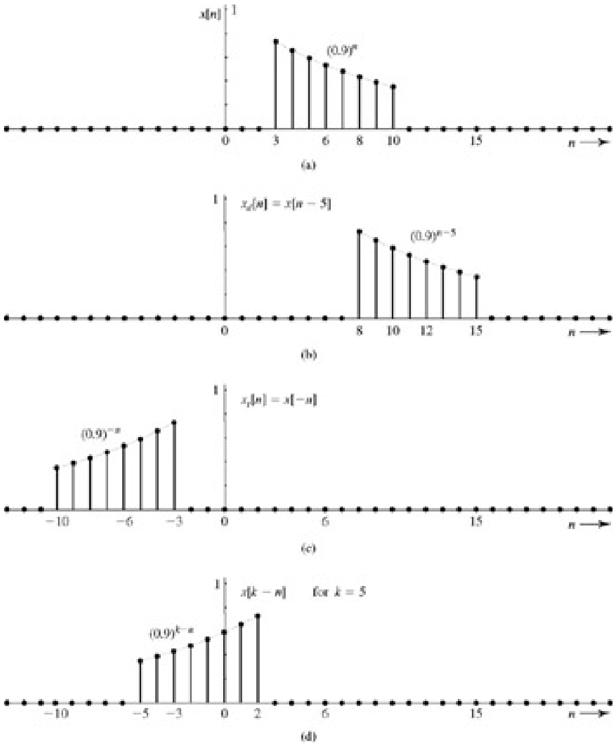

Figure 3.4: Shifting and time reversal of a signal.

Therefore, to shift a sequence by M units (M integer), we replace n with $n - M$. Thus $x[n - M]$ represents $x[n]$ shifted by M units. If M is positive, the shift is to the right (delay). If M is negative, the shift is to the left (advance). Accordingly, $x[n - 5]$ is $x[n]$ delayed (right-shifted) by 5 units, and $x[n + 5]$ is $x[n]$ advanced (left-shifted) by 5 units.

TIME REVERSAL

To time-reverse $x[n]$ in Fig. 3.4a, we rotate $x[n]$ about the vertical axis to obtain the time reversed signal $x_r[n]$ shown in Fig. 3.4c. Using the argument employed for a similar operation in continuous-time signals (Section 1.2), we obtain

$$x_r[n] = x[-n]$$

Therefore, to time-reverse a signal, we replace n with $-n$ so that $x[-n]$ is the time reversed $x[n]$. For example, if $x[n] = (0.9)^{-n}$ for $3 \le n \le 10$, then $x_r[n] = (0.9)^{-n}$ for $3 \le -n \le 10$; that is, $-3 \le n \le -10$, as shown in Fig. 3.4c.

The origin $n = 0$ is the anchor point, which remains unchanged under time-reversal operation because at $n = 0$, $x[n] = x[-n] = x[0]$. Note that while the reversal of $x[n]$ about the vertical axis is $x[-n]$, the reversal of $x[n]$ about the horizontal axis is $-x[n]$.

EXAMPLE 3.2

In the convolution operation, discussed later, we need to find the function $x[k - n]$ from $x[n]$.

This can be done in two steps: (i) time-reverse the signal $x[n]$ to obtain $x[-n]$; (ii) now, right-shift $x[-n]$ by k. Recall that right-shifting is accomplished by replacing n with $n - k$. Hence, right-shifting $x[-n]$by k units is $x[-(n-k] = x[k-n]$. Figure 3.4d shows $x[5 - n]$, obtained this way. We first time-reverse $x[n]$ to obtain $x[-n]$ in Fig. 3.4c. Next, we shift $x[-n]$ by $k = 5$ to obtain $x[k - n] = x[5 - n]$, as shown in Fig. 3.4d.

In this particular example, the order of the two operations employed is interchangeable. We can first left-shift $x[k]$ to obtain $x[n + 5]$. Next, we time-reverse $x[n + 5]$ to obtain $x[-n + 5] = x[5 - n]$. The reader is encouraged to verify that this procedure yields the same result, as in Fig. 3.4d.

SAMPLING RATE ALTERATION: DECIMATION AND INTERPOLATION

Alteration of the sampling rate is somewhat similar to time-scaling in continuous-time signals. Consider a signal $x[n]$ compressed by factor M. Compressing a signal $x[n]$ by factor M yields $x_d[n]$ given by

$$x_d[n] = x[Mn] \qquad (3.3)$$

Because of the restriction that discrete-time signals are defined only for integer values of the argument, we must restrict M to integer values. The values of $x[Mn]$ at $n = 0, 1, 2, 3,...$ are $x[0]$, $x[M]$, $x[2M]$, $x[3M]$,.... This means $x[Mn]$ selects every Mth sample of $x[n]$ and deletes all the samples in between. For this reason, this operation is called *decimation.* It reduces the number of samples by factor M. If $x[n]$ is obtained by sampling a continuous-time signal, this operation implies reducing the sampling rate by factor M. For this reason, decimation is also called *downsampling.* Figure 3.5a shows a signals $x[n]$ and Fig. 3.5b shows the signal $x[2n]$, which is obtained by deleting odd numbered samples of $x[n]$.[†]

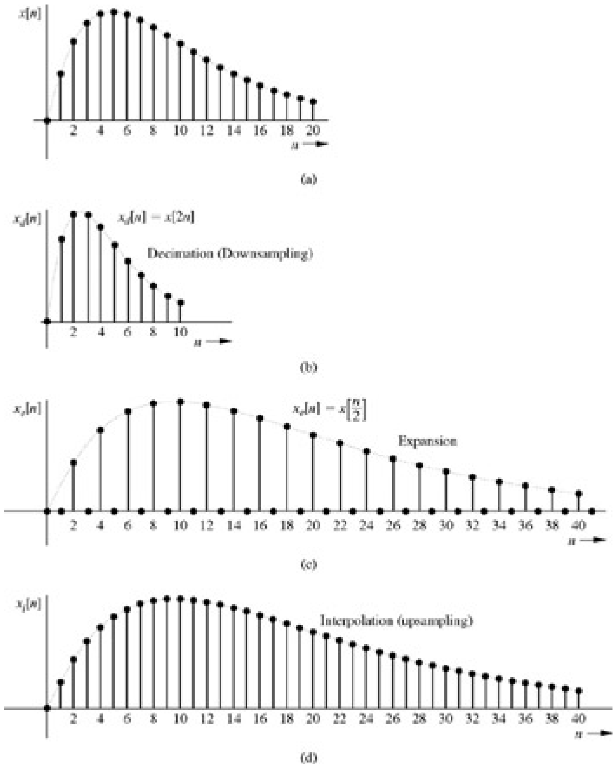

Figure 3.5: Compression (decimation) and expansion (interpolation) of a signal.

In the continuous-time case, time compression merely speeds up the signal without loss of any data. In contrast, decimation of $x[n]$ generally causes loss of data. Under certain conditions-for example, if $x[n]$ is the result of oversampling some continuous-time signal-then $x_d[n]$ may still retain the complete information about $x[n]$.

An *interpolated* signal is generated in two steps; first, we expand $x[n]$ by an integer factor L to obtain the expanded signal $x_e[n]$, as

$$x_e[n] = \begin{cases} x[n/L] & n = 0, \pm L \pm 2L, \dots, \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

To understand this expression, consider a simple case of expanding $x[n]$ by a factor 2 (L = 2). The expanded signal $x_e[n] = x[n/2]$. When n is odd, $n/2$ is noninteger. But $x[n]$ is defined only for integer values of n and is zero otherwise. Therefore, $x_e[n] = 0$, for odd n, that is, $x_e[1] = x_e[3] = x_e[5]$,... are all zero, as depicted in Fig. 3.5c. Moreover, $n/2$ is integer for even n, and the values of $x_e[n] = x[n/2]$ for n = 0, 2, 4, 6,..., are $x[0]$, $x[1]$, $x[2]$, $x[3]$,..., as shown in Fig. 3.5c. In general, for n = 0, 1, 2,..., $x_e[n]$ is given by the sequence

$$x[0], \underbrace{0, 0, \dots, 0, 0}_{L-1 \text{ zeros}}, x[1], \underbrace{0, 0, \dots, 0, 0}_{L-1 \text{ zeros}}, x[2], \underbrace{0, 0, \dots, 0, 0}_{L-1 \text{ zeros}}, \dots$$

Thus, the sampling rate of $x_e[n]$ is L times that of $x[n]$. Hence, this operation is called *expanding.* The expanded signal $x_e[n]$ contains all the data of $x[n]$, although in an expanded form.

In the expanded signal in Fig. 3.5c, the missing (zero-valued) odd-numbered samples can be reconstructed from the non-zero-valued samples by using some suitable interpolation formula. Figure 3.5d shows such an interpolated signal $x_i[n]$, where the missing samples are constructed by using an interpolating filter. The optimum interpolating filter is usually an ideal lowpass filter, which is realizable only approximately. In practice, we may use an interpolation that is nonoptimum but realizable. Further discussion of interpolation is beyond our scope. This process of filtering to interpolate the zero-valued samples is called *interpolation.* Since the interpolated data are computed from the existing data, interpolation does not result in gain of information.

EXERCISE E3.2

Show that $x[n]$ in Fig. 3.4a left-shifted by 3 units can be expressed as $0.729(0.9)^n$ for $0 \le n \le 7$, and zero otherwise. Sketch the shifted signal.

EXERCISE E3.3

Sketch the signal $x[n] = e^{-0.5n}$ for $-3 \le n \le 2$, and zero otherwise. Sketch the corresponding time-reversed signal and show that it can be expressed as $x_r[n] = e^{0.5n}$ for $-2 \le n \le 3$.

EXERCISE E3.4

Show that $x[-k - n]$ can be obtained from $x[n]$ by first right-shifting $x[n]$ by k units and then time-reversing this shifted signal.

EXERCISE E3.5

A signal $x[n]$ is expanded by factor 2 to obtain signal $x[n/2]$. The odd-numbered samples (n odd) in this signal have zero value. Show that the linearly interpolated odd-numbered samples are given by $x_i[n] = (1/2)\{x[n - 1] + x[n + 1]\}$.

[†]The terms "delay" and "advance" are meaningful only when the independent variable is time. For other independent variables, such as frequency or distance, it is more appropriate to refer to the "right shift" and "left shift" of a sequence.

[†]Odd-numbered samples of $x[n]$ can be retained (and even numbered samples deleted) by using the transformation $x_d[n] = x[2n + 1]$.

3.3 SOME USEFUL DISCRETE-TIME SIGNAL MODELS

We now discuss some important discrete-time signal models that are encountered frequently in the study of discrete-time signals and systems.

3.3-1 Discrete-Time Impulse Function $\delta[n]$

The discrete-time counterpart of the continuous-time impulse function $\delta(t)$ is $\delta[n]$, a Kronecker delta function, defined by

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{3.5}$$

This function, also called the unit impulse sequence, is shown in Fig. 3.6a. The shifted impulse sequence $\delta[n - m]$ is depicted in Fig. 3.6b. Unlike its continuous-time counterpart $\delta(t)$ (the Dirac delta), the Kronecker delta is a very simple function, requiring no special esoteric knowledge of distribution theory.

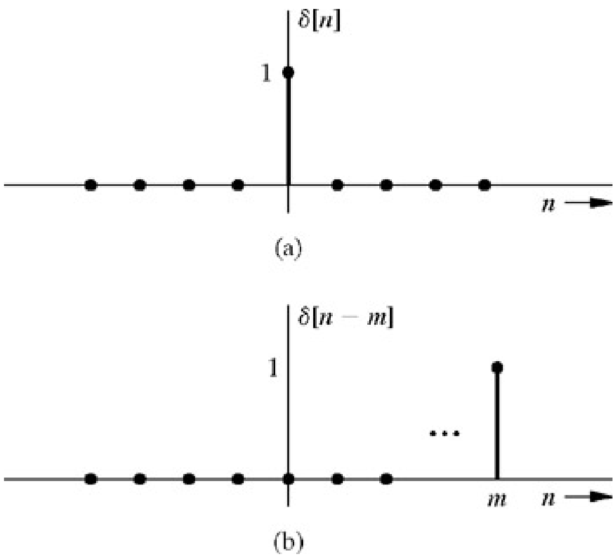

Figure 3.6: Discrete-time impulse function: **(a)** unit impulse sequence and **(b)** shifted impulse sequence.

3.3-2 Discrete-Time Unit Step Function $u[n]$

The discrete-time counterpart of the unit step function $u(t)$ is $u[n]$ (Fig. 3.7a), defined by

$$u[n] = \begin{cases} 1 & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases} \tag{3.6}$$

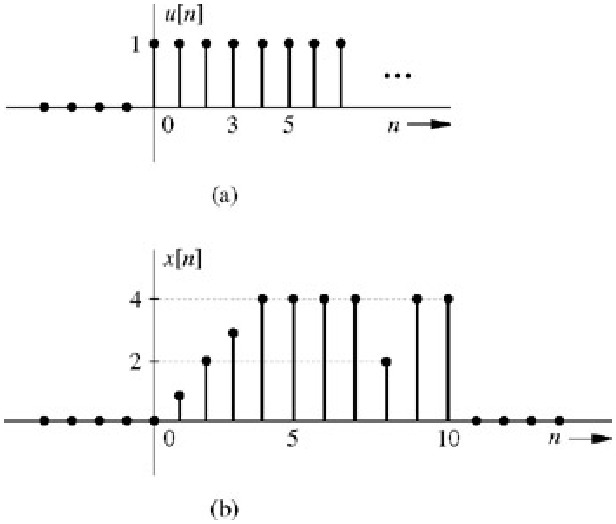

Figure 3.7: **(a)** A discrete-time unit step function $u[n]$ and **(b)** its application.

If we want a signal to start at $n = 0$ (so that it has a zero value for all $n < 0$), we need only multiply the signal by $u[n]$.

EXAMPLE 3.3

Describe the signal $x[n]$ shown in Fig. 3.7b by a single expression valid for all n.

There are many different ways of viewing $x[n]$. Although each way of viewing yields a different expression, they are all equivalent. We shall consider here just one possible expression.

The signal $x[n]$ can be broken into three components: (1) a ramp component $x_1[n]$ from $n = 0$ to 4, (2) a step component $x_2[n]$ from $n = 5$ to 10, and (3) an impulse component $x_3[n]$ represented by the negative spike at $n = 8$. Let us consider each one separately.

We express $x_1[n] = n(u[n] -$

3.4 EXAMPLES OF DISCRETE-TIME SYSTEMS

We shall give here four examples of discrete-time systems. In the first two examples, the signals are inherently of the discrete-time variety. In the third and fourth examples, a continuous-time signal is processed by a discrete-time system, as illustrated in Fig. 3.2, by discretizing the signal through sampling.

EXAMPLE 3.4: (Savings Account)

A person makes a deposit (the input) in a bank regularly at an interval of T (say, 1 month). The bank pays a certain interest on the account balance during the period T and mails out a periodic statement of the account balance (the output) to the depositor. Find the equation relating the output $y[n]$ (the balance) to the input $x[n]$ (the deposit).

In this case, the signals are inherently discrete time. Let

$$x[n] = \text{deposit made at the } n\text{th discrete instant}$$

$$\begin{aligned} y[n] = &\text{ account balance at the } n\text{th instant computed} \\ &\text{ immediately after receipt of the } n\text{th deposit } x[n] \end{aligned}$$

$$r = \text{interest per dollar per period } T$$

The balance $y[n]$ is the sum of (i) the previous balance $y[n-1]$, (ii) the interest on $y[n-1]$ during the period T, and (iii) the deposit $x[n]$

$$\begin{aligned} y[n] &= y[n-1] + ry[n-1] + x[n] \\ &= (1+r)y[n-1] + x[n] \end{aligned}$$

or

$$y[n] - ay[n-1] = x[n] \qquad a = 1 + r \tag{3.9a}$$

In this example the deposit $x[n]$ is the input (cause) and the balance $y[n]$ is the output (effect).

A withdrawal from the account is a negative deposit. Therefore, this formulation can handle deposits as well as withdrawals. It also applies to a loan payment problem with the initial value $y[0] = -M$, where M is the amount of the loan. A loan is an initial deposit with a negative value. Alternately, we may treat a loan of M dollars taken at $n = 0$ as an input of $-M$ at $n = 0$ (see Prob. 3.8-16).

We can express Eq. (3.9a) in an alternate form. The choice of index n in Eq. (3.9a) is completely arbitrary, so we can substitute $n + 1$ for n to obtain

$$y[n+1] - ay[n] = x[n+1] \tag{3.9b}$$

We also could have obtained Eq. (3.9b) directly by realizing that $y[n+1]$, the balance at the $(n+1)$st instant, is the sum of $y[n]$ plus $ry[n]$ (the interest on $y[n]$) plus the deposit (input) $x[n+1]$ at the $(n+1)$st instant.

The difference equation in (3.9a) uses delay operation, whereas the form in Eq. (3.9b) uses advance operation. We shall call the form (3.9a) the *delay operator* form and the form (3.9b) the *advance operator* form. The delay operator form is more natural because operation of delay is a causal, and hence realizable. In contrast, advance operation, being noncausal, is unrealizable. We use the advance operator form primarily for its mathematical convenience over the delay form.[†]

We shall now represent this system in a block diagram form, which is basically a road map to hardware (or software) realization of the system. For this purpose, the causal (realizable) delay operator form in Eq. (3.9a) will be used. There are three basic operations in this equation: *addition, scalar multiplication*, and *delay*. Figure 3.11 shows their schematic representation. In addition, we also have a *pickoff* node (Fig. 3.11d), which is used to provide multiple copies of a signal at its input.

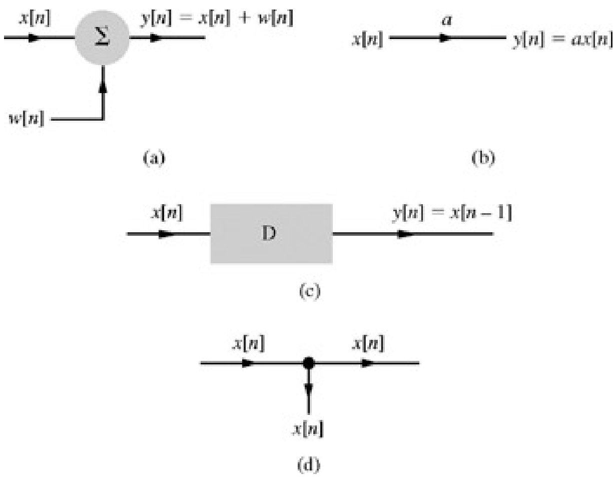

Figure 3.11: Schematic representations of basic operations on sequences.

Equation (3.9a) can be rewritten as

$$y[n] = ay[n-1] + x[n] \qquad a = 1 + r \tag{3.9c}$$

Figure 3.12 shows in block diagram form a system represented by Eq. (3.9c). To understand this realization, assume that the output $y[n]$ is available at the pickoff node N. Unit delay of $y[n]$ results in $y[n-1]$, which is multiplied by a scalar of value a to yield $ay[n-1]$. Next, we generate $y[n]$ by adding the input $x[n]$ and $ay[n-1]$ according to Eq. (3.9c).[‡] Observe that the node N is a pickoff node, from which two copies of the output signal flow out; one as the feedback signal, and the other as the output signal.

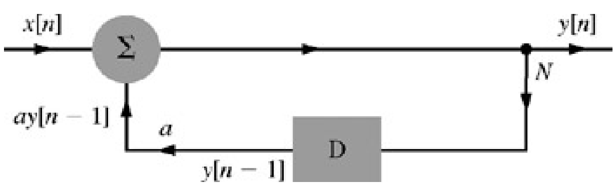

Figure 3.12: Realization of the savings account system.

EXAMPLE 3.5: (Sales Estimate)

In an nth semester, $x[n]$ students enroll in a course requiring a certain textbook. The publisher sells $y[n]$ new copies of the book in the nth semester. On the average, one-quarter of students with books in salable condition resell the texts at the end of the semester, and the book life is three semesters. Write the equation relating $y[n]$, the new books sold by the publisher, to $x[n]$, the number of students enrolled in the nth semester, assuming that every student buys a book.

In the nth semester, the total books $x[n]$ sold to students must be equal to $y[n]$ (new books from the publisher) plus used books from students enrolled in the preceding two semesters (because the book life is only three semesters). There are $y[n-1]$ new books sold in the $(n-1)$st semester, and one-quarter of these books, that is, $(1/4)y[n-1]$, will be resold in the nth semester. Also, $y[n-2]$ new books are sold in the $(n-2)$nd semester, and one-quarter of these, that is, $(1/4)y[n-2]$, will be resold in the $(n-1)$st semester. Again a quarter of these, that is, $(1/16)y[n-2]$, will be resold in the nth semester. Therefore, $x[n]$ must be equal to the sum of $y[n]$, $(1/4)y[n-1]$, and $(1/16)y[n-2]$.

$$y[n] + \tfrac{1}{4}y[n-1] + \tfrac{1}{16}y[n-2] = x[n] \tag{3.10a}$$

Equation (3.10a) can also be expressed in an alternative form by realizing that this equation is valid for any value of n. Therefore, replacing n by $n + 2$, we obtain

$$y[n+2] + \tfrac{1}{4}y[n+1] + \tfrac{1}{16}y[n] = x[n+2] \tag{3.10b}$$

This is the alternative form of Eq. (3.10a).

For a realization of a system with this input-output equation, we rewrite the delay form Eq. (3.10a) as

$$y[n] = -\tfrac{1}{4}y[n-1] - \tfrac{1}{16}y[n-2] + x[n] \tag{3.10c}$$

Figure 3.13 shows a hardware realization of Eq. (3.10c) using two unit delays in cascade.[†]

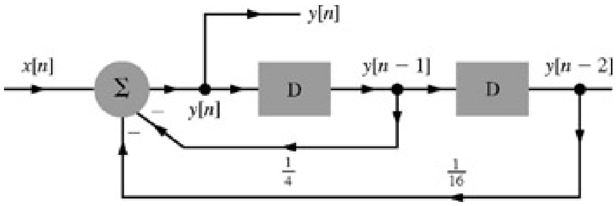

Figure 3.13: Realization of the system representing sales estimate in Example 3.5.

EXAMPLE 3.6: (Digital Differentiator)

Design a discrete-time system, like the one in Fig. 3.2, to differentiate continuous-time signals. This differentiator is used in an audio system having an input signal bandwidth below 20 kHz.

In this case, the output $y(t)$ is required to be the derivative of the input $x(t)$. The discrete-time processor (system) G processes the samples of $x(t)$ to produce the discrete-time output $y[n]$. Let $x[n]$ and $y[n]$ represent the samples T seconds apart of the signals $x(t)$ and $y(t)$, respectively, that is,

$$x[n] = x(nT) \qquad \text{and} \qquad y[n] = y(nT) \tag{3.11}$$

The signals $x[n]$ and $y[n]$ are the input and the output for the discrete-time system G. Now, we require that

$$y(t) = \frac{dx}{dt}$$

Therefore, at $t = nT$ (see Fig. 3.14a)

$$\begin{aligned} y(nT) &= \left.\frac{dx}{dt}\right|_{t=nT} \\ &= \lim_{T\to 0} \frac{1}{T}[x(nT) - x[(n-1)T]] \end{aligned}$$

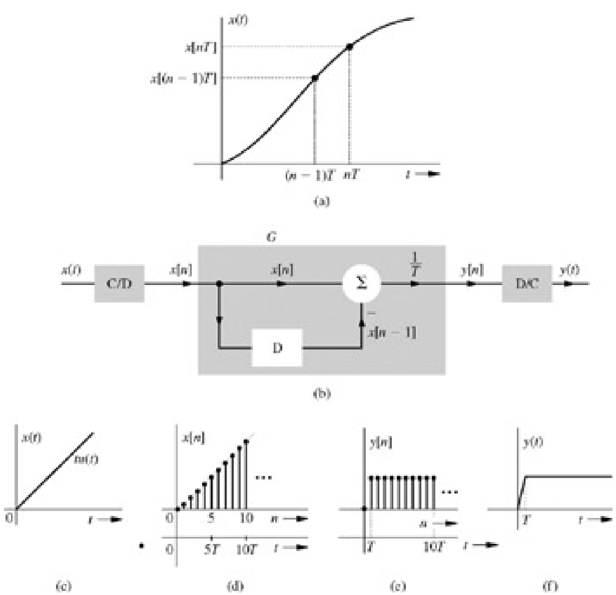

Figure 3.14: Digital differentiator and its realization.

By using the notation in Eq. (3.11), the foregoing equation can be expressed as

$$y[n] = \lim_{T\to 0} \frac{1}{T}\{x[n] - x[n-1]\}$$

This is the input-output relationship for G required to achieve our objective. In practice, the sampling interval T cannot be zero. Assuming T to be sufficiently small, the equation just given can be expressed as

$$y[n] = \frac{1}{T}\{x[n] - x[n-1]\} \tag{3.12}$$

The approximation improves as T approaches 0. A discrete-time processor G to realize Eq. (3.12) is shown inside the shaded box in Fig. 3.14b. The system in Fig. 3.14b acts as a differentiator. This example shows how a continuous-time signal can be processed by a discrete-time system. The considerations for determining the sampling interval T are discussed in Chapters 5 and 8, where it is shown that to process frequencies below 20 kHz, the proper choice is

$$T \leq \frac{1}{2 \times \text{highest frequency}} = \frac{1}{40{,}000} = 25\ \mu\text{s}$$

To see how well this method of signal processing works, let us consider the differentiator in Fig. 3.14b with a ramp input $x(t) = t$, depicted in Fig. 3.14c. If the system were to act as a differentiator, then the output $y(t)$ of the system should be the unit step function $u(t)$. Let us investigate how the system performs this particular operation and how well the system achieves the objective.

The samples of the input $x(t) = t$ at the interval of T seconds act as the input to the discrete-time system G. These samples, denoted by a compact notation $x[n]$, are, therefore,

$$x[n] = x(t)|_{t=nT} = t|_{t=nT} \qquad t \geq 0$$

$$= nT \qquad n \geq 0$$

Figure 3.14d shows the sampled signal $x[n]$. This signal acts as an input to the discrete-time system G. Figure 3.14b shows that the operation of G consists of subtracting a sample from the preceding (delayed) sample and then multiplying the difference with $1/T$. From Fig. 3.14d, it is clear that the difference between the successive samples is a constant $nT - (n - 1)T = T$ for all samples, except for the sample at $n = 0$ (because there is no preceding sample at $n = 0$). The output of G is $1/T$ times the difference T, which is unity for all values of n, except $n = 0$, where it is zero. Therefore, the output $y[n]$ of G consists of samples of unit values for $n \geq 1$, as illustrated in Fig. 3.14e. The D/C (discrete-time to continuous-time) converter converts these samples into a continuous-time signal $y(t)$, as shown in Fig. 3.14f. Ideally, the output should have been $y(t) = u(t)$. This deviation from the ideal is due to our use of a nonzero sampling interval T. As T approaches zero, the output $y(t)$ approaches the desired output $u(t)$.

The digital differentiator in Eq. (3.12) is an example of what is known as the *backward difference* system. The reason for calling it so is obvious from Fig. 3.14a. To compute the derivative of $y(t)$, we are using the difference between the present sample value and the preceding (backward) sample value. If we use the difference between the next (forward) sample at $t = (n + 1)T$ and the present sample at $t = nT$, we obtain the forward difference form of differentiator as

$$y[n] = \frac{1}{T}\{x[n+1] - x[n]\} \qquad (3.13)$$

EXAMPLE 3.7: (Digital Integrator)

Design a digital integrator along the same lines as the digital differentiator in Example 3.6.

For an integrator, the input $x(t)$ and the output $y(t)$ are related by

$$y(t) = \int_{-\infty}^{t} x(\tau)\, d\tau$$

Therefore, at $t = nT$ (see Fig. 3.14a)

$$y(nT) = \lim_{T \to 0} \sum_{k=-\infty}^{n} x(kT)T$$

Using the usual notation $x(kT) = x[k]$, $y(nT) = y[n]$, and so on, this equation can be expressed

$$y[n] = \lim_{T \to 0} T \sum_{k=-\infty}^{n} x[k]$$

Assuming that T is small enough to justify the assumption $T \to 0$, we have

$$y[n] = T \sum_{k=-\infty}^{n} x[k] \qquad (3.14a)$$

This equation represents an example of *accumulator* system. This digital integrator equation can be expressed in an alternate form. From Eq. (3.14a), it follows that

$$y[n] - y[n-1] = T\,x[n] \qquad (3.14b)$$

This is an alternate description for the digital integrator. Equations (3.14a) and (3.14b) are equivalent; the one can be derived from the other. Observe that the form of Eq. (3.14b) is similar to that of Eq. (3.9a). Hence, the block diagram representation of the digital differentiator in the form (3.14b) is identical to that in Fig. 3.12 with $a = 1$ and the input multiplied by T.

RECURSIVE AND NONRECURSIVE FORMS OF DIFFERENCE EQUATION

If Eq. (3.14b) expresses Eq. (3.14a) in another form, what is the difference between these two forms? Which form is preferable? To answer these questions, let us examine how the output is computed by each of these forms. In Eq. (3.14a), the output $y[n]$ at any instant n is computed by adding all the past input values till n. This can mean a large number of additions. In contrast, Eq. (3.14b) can be expressed as $y[n] = y[n - 1] + Tx[n]$. Hence, computation of $y[n]$ involves addition of only two values; the preceding output value $y[n - 1]$ and the present input value $x[n]$. The computations are done recursively by using the preceding output values. For example, if the input starts at $n = 0$, we first compute $y[0]$. Then we use the computed value $y[0]$ to compute $y[1]$. Knowing $y[1]$, we compute $y[2]$, and so on. The computations are recursive. This is why the form (3.14b) is called *recursive* form and the form (3.14a) is called *nonrecursive* form. Clearly, "recursive" and "nonrecursive" describe two different ways of presenting the same information. Equations (3.9), (3.10), and (3.14b) are examples of recursive form, and Eqs. (3.12) and (3.14a) are examples of nonrecursive form.

KINSHIP OF DIFFERENCE EQUATIONS TO DIFFERENTIAL EQUATIONS

We now show that a digitized version of a differential equation results in a difference equation. Let us consider a simple first-order differential equation

$$\frac{dy}{dt} + cy(t) = x(t) \tag{3.15a}$$

Consider uniform samples of $x(t)$ at intervals of T seconds. As usual, we use the notation $x[n]$ to denote $x(nT)$, the nth sample of $x(t)$. Similarly, $y[n]$ denotes $y[nT]$, the nth sample of $y(t)$. From the basic definition of a derivative, we can express Eq. (3.15a) at $t = nT$ as

$$\lim_{T\to 0} \frac{y[n] - y[n-1]}{T} + cy[n] = x[n]$$

Clearing the fractions and rearranging the terms yields (assuming nonzero, but very small T)

$$y[n] + \alpha y[n-1] = \beta x[n] \tag{3.15b}$$

where

$$\alpha = \frac{-1}{1+cT} \qquad \text{and} \qquad \beta = \frac{T}{1+cT}$$

We can also express Eq. (3.15b) in advance operator form as

$$y[n+1] + \alpha y[n] = \beta x[n+1] \tag{3.15c}$$

It is clear that a differential equation can be approximated by a difference equation of the same order. In this way, we can approximate an nth-order differential equation by a difference equation of nth order. Indeed, a digital computer solves differential equations by using an equivalent difference equation, which can be solved by means of simple operations of addition, multiplication, and shifting. Recall that a computer can perform only these simple operations. It must necessarily approximate complex operation like differentiation and integration in terms of such simple operations. The approximation can be made as close to the exact answer as possible by choosing sufficiently small value for T.

At this stage, we have not developed tools required to choose a suitable value of the sampling interval T. This subject is discussed in Chapter 5 and also in Chapter 8. In Section 5.7, we shall discuss a systematic procedure (impulse invariance method) for finding a discrete-time system with which to realize an Nth-order LTIC system.

ORDER OF A DIFFERENCE EQUATION

Equations (3.9), (3.10), (3.13), (3.14b), and (3.15) are examples of difference equations. The highest-order difference of the output signal or the input signal, whichever is higher represents the *order* of the difference equation. Hence, Eqs. (3.9), (3.13), (3.14b), and (3.15) are first-order difference equations, whereas Eq. (3.10) is of the second order.

EXERCISE E3.8

Design a digital integrator in Example 3.7 using the fact that for an integrator, the output $y(t)$ and the input $x(t)$ are related by $dy/dt = x(t)$. Approximation (similar to that in Example 3.6) of this equation at $t = nT$ yields the recursive form in Eq. (3.14b).

ANALOG, DIGITAL, CONTINUOUS-TIME, AND DISCRETE-TIME SYSTEMS

The basic difference between continuous-time systems and analog systems, as also between discrete-time and digital systems, is fully explained in Sections 1.7-5 and 1.7-6.[†] Historically, discrete-time systems have been realized with digital computers, where continuous-time signals are processed through digitized samples rather than unquantized samples. Therefore, the terms *digital filters* and *discrete-time systems* are used synonymously in the literature. This distinction is irrelevant in the analysis of discrete-time systems. For this reason, we follow this loose convention in this book, where the term *digital filter* implies a *discrete-time system*, and *analog filter* means *continuous-time system.* Moreover, the terms C/D (continuous-to-discrete-time) and D/C will occasionally be used interchangeably with terms A/D (analog-to-digital) and D/A, respectively.

ADVANTAGES OF DIGITAL SIGNAL PROCESSING

1. Digital systems operation can tolerate considerable variation in signal values, and, hence, are less sensitive to changes in the component parameter values caused by temperature variation, aging and other factors. This results in greater degree of precision and stability. Since they are binary circuits, their accuracy can be increased by using more complex circuitry to increase word length, subject to cost limitations.
2. Digital systems do not require any factory adjustment and can be easily duplicated in volume without having to worry about precise component values. They can be fully integrated, and even highly complex systems can be placed on a single chip by using *VLSI* (very-large-scale integrated) circuits.
3. Digital filters are more flexible. Their characteristics can be easily altered simply by changing the program. Digital hardware implementation permits the use of microprocessors, miniprocessors, digital switching, and large-scale integrated circuits.
4. A greater variety of filters can be realized by digital systems.
5. Digital signals can be stored easily and inexpensively on magnetic tapes or disks without deterioration of signal quality. It is also possible to search and select information from distant electronic storehouses.
6. Digital signals can be coded to yield extremely low error rates and high fidelity, as well as privacy. Also, more sophisticated signal processing algorithms can be used to process digital signals.

7. Digital filters can be easily time-shared and therefore can serve a number of inputs simultaneously. Moreover, it is easier and more efficient to multiplex several digital signals on the same channel.
8. Reproduction with digital messages is extremely reliable without deterioration. Analog messages such as photocopies and films, for example, lose quality at each successive stage of reproduction and have to be transported physically from one distant place to another, often at relatively high cost.

   One must weigh these advantages against such disadvantages, as increased system complexity due to use of A/D and D/A interfaces, limited range of frequencies available in practice (about tens of megahertz), and use of more power than is needed for the passive analog circuits. Digital systems use power-consuming active devices.

3.4-1 Classification of Discrete-Time Systems

Before examining the nature of discrete-time system equations, let us consider the concepts of linearity, time invariance (or shift invariance), and causality, which apply to discrete-time systems also.

LINEARITY AND TIME INVARIANCE

For discrete-time systems, the definition of *linearity* is identical to that for continuous-time systems, as given in Eq. (1.40). We can show that the systems in Examples 3.4, 3.5, 3.6, and 3.7 are all linear.

Time invariance (or *shift invariance*) for discrete-time systems is also defined in a way similar to that for continuous-time systems. Systems whose parameters do not change with time (with n) are *time-invariant* or shift-invariant (also *constant-parameter*) systems. For such a system, if the input is delayed by k units or samples, the output is the same as before but delayed by k samples (assuming the initial conditions also are delayed by k). Systems in Examples 3.4, 3.5, 3.6, and 3.7 are time invariant because the coefficients in the system equations are constants (independent of n). If these coefficients were functions of n (time), then the systems would be linear *time-varying* systems. Consider, for example, a system described by

$$y[n] = e^{-n}x[n]$$

For this system, let a signal $x_1[n]$ yield the output $y_1[n]$, and another input $x_2[n]$ yield the output $y_2[n]$. Then

$$y_1[n] = e^{-n}x_1[n] \qquad \text{and} \qquad y_2[n] = e^{-n}x_2[n]$$

If we let $x_2[n] = x_1[n - N_0]$, then

$$y_2[n] = e^{-n}x_2[n] = e^{-n}x_1[n - N_0] \neq y_1[n - N_0]$$

Clearly, this is a time-varying parameter system.

CAUSAL AND NONCAUSAL SYSTEMS

A *causal* (also known as a *physical* or *nonanticipative*) system is one for which the output at any instant $n = k$ depends only on the value of the input $x[n]$ for $n \leq k$. In other words, the value of the output at the present instant depends only on the past and present values of the input $x[n]$, not on its future values. As we shall see, the systems in Examples 3.4, 3.5, 3.6, and 3.7 are all causal.

INVERTIBLE AND NONINVERTIBLE SYSTEMS

A discrete-time system S is invertible if an inverse system S_i exists such that the cascade of S and S_i results in an *identity* system. An identity system is defined as one whose output is identical to the input. In other words, for an invertible system, the input can be uniquely determined from the corresponding output. For every input there is a unique output. When a signal is processed through such a system, its input can be reconstructed from the corresponding output. There is no loss of information when a signal is processed through an invertible system.

A cascade of a unit delay with a unit advance results in an identity system because the output of such a cascaded system is identical to the input. Clearly, the inverse of an ideal unit delay is ideal unit advance, which is a noncausal (and unrealizable) system. In contrast, a compressor $y[n] = x[Mn]$ is not invertible because this operation loses all but every Mth sample of the input, and, generally, the input cannot be reconstructed. Similarly, an operations, such as, $y[n] = \cos x[n]$ or $y[n] = |x[n]|$ are not invertible.

EXERCISE E3.9

Show that a system specified by equation $y[n] = ax[n] + b$ is invertible, but the system $y[n] = |x[n]|^2$ is noninvertible.

STABLE AND UNSTABLE SYSTEMS

The concept of stability is similar to that in continuous-time systems. Stability can be *internal* or *external.* If every *bounded input* applied at the input terminal results in a *bounded output*, the system is said to be stable *externally.* The external stability can be ascertained by measurements at the external terminals of the system. This type of stability is also known as the stability in the BIBO (bounded-input/bounded-output) sense. Both, internal and external stability are discussed in greater detail in Section 3.10.

MEMORYLESS SYSTEMS AND SYSTEMS WITH MEMORY

The concepts of memoryless (or instantaneous) systems and those without memory (or dynamic) are identical to the corresponding concepts of the continuous-time case. A system is memoryless if its response at any instant n depends at most on the input at the same instant n. The output at any instant of a system with memory generally depends on the past, present, and future values of the input. For example, $y[n] = \sin x[n]$ is an example of instantaneous system, and $y[n] - y[n - 1] = x[n]$ is an example of a dynamic system or a system with memory.

[†]Use of the advance operator form results in discrete-time system equations that are identical in form to those for continuous-time systems. This will become apparent later. In the transform analysis, use of the advance operator permits the use of more convenient variable z instead of the clumsy z^{-1} needed in the delay operator form.

[‡]A unit delay represents one unit of time delay. In this example one unit of delay in the output corresponds to period T for the actual output.

[†]The comments in the preceding footnote apply here also. Although one unit of delay in this example is one semester, we need not use this value in the hardware realization. Any value other than one semester results in a time-scaled output.

[†]The terms *discrete-time* and *continuous-time* qualify the nature of a signal along the time axis (horizontal axis). The terms *analog* and *digital*, in contrast, qualify the nature of the signal amplitude (vertical axis).

3.5 DISCRETE-TIME SYSTEM EQUATIONS

In this section we discuss time-domain analysis of LTID (linear time-invariant, discrete-time systems). With minor differences, the procedure is parallel to that for continuous-time systems.

DIFFERENCE EQUATIONS

Equations (3.9), (3.10), (3.12), and (3.15) are examples of difference equations. Equations (3.9), (3.12), and (3.15) are first-order difference equations, and Eq. (3.10) is a second-order difference equation. All these equations are linear, with constant (not time-varying) coefficients.[†] Before giving a general form of an Nth-order linear difference equation, we recall that a difference equation can be written in two forms; the first form uses delay terms such as $y[n-1]$, $y[n-2]$, $x[n-1]$, $x[n-2]$, and so on, and the alternate form uses advance terms such as $y[n+1]$, $y[n+2]$, and so on. Although the delay form is more natural, we shall often prefer the advance form, not just for the general notational convenience, but also for resulting notational uniformity with the operational form for differential equations. This facilitates the commonality of the solutions and concepts for continuous-time and discrete-time systems.

We start here with a general difference equation, using the advance operator form

$$\begin{aligned} y[n+N] + a_1 y[n+N-1] + \cdots + a_{N-1} y[n+1] + a_N y[n] &= b_{N-M} x[n+M] \\ + b_{N-M+1} x[n+M-1] + \cdots + b_{N-1} x[n+1] + b_N x[n] & \qquad (3.16) \end{aligned}$$

This is a linear difference equation whose order is Max(N, M). We have assumed the coefficient of $y[n+N]$ to be unity ($a_0 = 1$) without loss of generality. If $a_0 \neq 1$, we can divide the equation throughout by a_0 to normalize the equation to have $a_0 = 1$.

CAUSALITY CONDITION

For a causal system the output cannot depend on future input values. This means that when the system equation is in the advance operator form (3.16), the causality requires $M \le N$. If M were to be greater than N, then $y[n+N]$, the output at $n+N$ would depend on $x[n+M]$, which is the input at the later instant $n+M$. For a general causal case, $M = N$, and Eq. (3.16) can be expressed as

$$\begin{aligned} y[n+N] + a_1 y[n+N-1] + \cdots + a_{N-1} y[n+1] + a_N y[n] &= b_0 x[n+N] \\ + b_1 x[n+N-1] + \cdots + b_{N-1} x[n+1] + b_N x[n] & \qquad (3.17a) \end{aligned}$$

where some of the coefficients on either side can be zero. In this Nth-order equation, a_0, the coefficient of $y[n+N]$, is normalized to unity. Equation (3.17a) is valid for all values of n. Therefore, it is still valid if we replace n by $n-N$ throughout the equation [see Eqs. (3.9a) and (3.9b)]. Such replacement yields the alternative form (the delay operator form) of Eq. (3.17a):

$$\begin{aligned} y[n] + a_1 y[n-1] + \cdots + a_{N-1} y[n-N+1] + a_N y[n-N] &= b_0 x[n] \\ + b_1 x[n-1] + \cdots + b_{N-1} x[n-N+1] + b_N x[n-N] & \qquad (3.17b) \end{aligned}$$

3.5-1 Recursive (Iterative) Solution of Difference Equation

Equation (3.17b) can be expressed as

$$\begin{aligned} y[n] = -a_1 y[n-1] - a_2 y[n-2] - \cdots - a_N y[n-N] \\ + b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N] \qquad (3.17c) \end{aligned}$$

In Eq. (3.17c), $y[n]$ is computed from $2N+1$ pieces of information; the preceding N values of the output: $y[n-1]$, $y[n-2]$, ..., $y[n-N]$, and the preceding N values of the input: $x[n-1]$, $x[n-2]$, ..., $x[n-N]$, and the present value of the input $x[n]$. Initially, to compute $y[0]$, the N initial conditions $y[-1]$, $y[-2]$,..., $y[-N]$ serve as the preceding N output values. Hence, knowing the N initial conditions and the input, we can determine the entire output $y[0]$, $y[1]$, $y[2]$, $y[3]$,...recursively, one value at a time. For instance, to find $y[0]$ we set $n = 0$ in Eq. (3.17c). The left-hand side is $y[0]$, and the right-hand side is expressed in terms of N initial conditions $y[-1]$, $y[-2]$,..., $y[-N]$ and the input $x[0]$ if $x[n]$ is causal (because of causality, other input terms $x[-n] = 0$). Similarly, knowing $y[0]$ and the input, we can compute $y[1]$ by setting $n = 1$ in Eq. (3.17c). Knowing $y[0]$ and $y[1]$, we find $y[2]$, and so on. Thus, we can use this recursive procedure to find the complete response $y[0]$, $y[1]$, $y[2]$,.... For this reason, this equation is classed as a recursive form. This method basically reflects the manner in which a computer would solve a recursive difference equation, given the input and initial conditions. Equation (3.17) is nonrecursive if all the $N-1$ coefficients $a_i = 0$ ($i = 1,2,..., N-1$). In this case, it can be seen that $y[n]$ is computed only from the input values and without using any previous outputs. Generally speaking, the recursive procedure applies only to equations in the recursive form. The recursive (iterative) procedure is demonstrated by the following examples.

EXAMPLE 3.8

Solve iteratively

$$y[n] - 0.5y[n-1] = x[n] \qquad (3.18a)$$

with initial condition $y[-1] = 16$ and causal input $x[n] = n^2$ (starting at $n = 0$). This equation can be expressed as

$$y[n] = 0.5y[n-1] + x[n] \qquad (3.18b)$$

If we set $n = 0$ in this equation, we obtain

$$y[0] = 0.5y[-1] + x[0]$$
$$= 0.5(16) + 0 = 8$$

Now, setting $n = 1$ in Eq. (3.18b) and using the value $y[0] = 8$ (computed in the first step) and $x[1] = (1)^2 = 1$, we obtain

$$y[1] = 0.5(8) + (1)^2 = 5$$

Next, setting $n = 2$ in Eq. (3.18b) and using the value $y[1] = 5$ (computed in the previous step) and $x[2] = (2)^2$, we obtain

$$y[2] = 0.5(5) + (2)^2 = 6.5$$

Continuing in this way iteratively, we obtain

$$y[3] = 0.5(6.5) + (3)^2 = 12.25$$
$$y[4] = 0.5(12.25) + (4)^2 = 22.125$$
$$\vdots$$

The output $y[n]$ is depicted in Fig. 3.15.

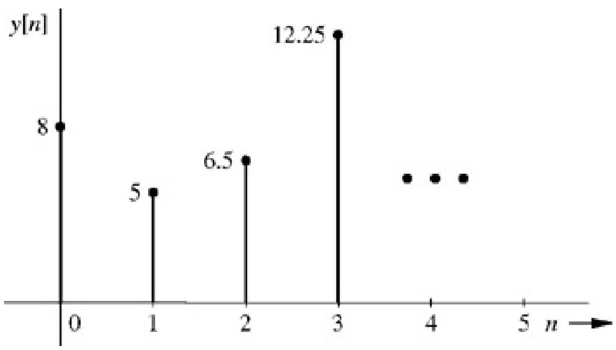

Figure 3.15: Iterative solution of a difference equation.

We now present one more example of iterative solution-this time for a second-order equation. The iterative method can be applied to a difference equation in delay or advance operator form. In Example 3.8 we considered the former. Let us now apply the iterative method to the advance operator form.

EXAMPLE 3.9

Solve iteratively

$$y[n+2] - y[n+1] + 0.24y[n] = x[n+2] - 2x[n+1] \qquad (3.19)$$

with initial conditions $y[-1] = 2$, $y[-2] = 1$ and a causal input $x[n] = n$ (starting at $n = 0$). The system equation can be expressed as

$$y[n+2] = y[n+1] - 0.24y[n] + x[n+2] - 2x[n+1] \qquad (3.20)$$

Setting $n = -2$ and then substituting $y[-1] = 2$, $y[-2] = 1$, $x[0] = x[-1] = 0$ (recall that $x[n] = n$ starting at $n = 0$), we obtain

$$y[0] = 2 - 0.24(1) + 0 - 0 = 1.76$$

Setting $n = -1$ in Eq. (3.20) and then substituting $y[0] = 1.76$, $y[-1] = 2$, $x[1] = 1$, $x[0] = 0$, we obtain

$$y[1] = 1.76 - 0.24(2) + 1 - 0 = 2.28$$

Setting $n = 0$ in Eq. (3.20) and then substituting $y[0] = 1.76$, $y[1] = 2.28$, $x[2] = 2$ and $x[1] = 1$ yields

$$y[2] = 2.28 - 0.24(1.76) + 2 - 2(1) = 1.8576$$

and so on.

Note carefully the recursive nature of the computations. From the N initial conditions (and the input) we obtained $y[0]$ first. Then, using this value of $y[0]$ and the preceding $N - 1$ initial conditions (along with the input), we find $y[1]$. Next, using $y[0]$, $y[1]$ along with the past $N - 2$ initial conditions and input, we obtained $y[2]$, and so on. This method is general and can be applied to a recursive difference equation of any order. It is interesting that the hardware realization of Eq. (3.18a) depicted in Fig. 3.12 (with $a = 0.5$) generates the solution precisely in this (iterative) fashion.

EXERCISE E3.10

Using the iterative method, find the first three terms of $y[n]$ for

$$y[n+1] - 2y[n] = x[n]$$

?

The initial condition is $y[-1] = 10$ and the input $x[n] = 2$ starting at $n = 0$.

Answers

$y[0] = 20 \qquad y[1] = 42 \qquad y[2] = 86$

COMPUTER EXAMPLE C3.3

Use MATLAB to solve Example 3.9.

```
>> n = (-2:10)'; y=[1;2;zeros(length(n)-2,1)]; x=[0;0;n(3:end)];
>> for k = 1:length(n)-2,
>> y(k + 2) = y(k+1) - 0.24*y(k) + x(k+2) - 2*x(k+1);
>> end;
>> clf; stem(n,y,'k'); xlabel('n'); ylabel('y[n]');
>> disp(' n            y'); disp([num2str([n,y])]);
n            y
-2             1
-1             2
0          1.76
1          2.28
2        1.8576
3        0.3104
4      -2.13542
5      -5.20992
6      -8.69742
7       -12.447
8      -16.3597
9      -20.3724
10     -24.4461
```

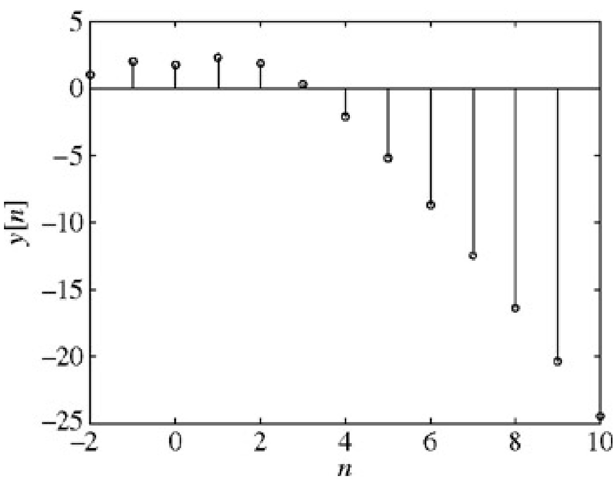

Figure C3.3

We shall see in the future that the solution of a difference equation obtained in this direct (iterative) way is useful in many situations. Despite the many uses of this method, a closed-form solution of a difference equation is far more useful in study of the system behavior and its dependence on the input and the various system parameters. For this reason we shall develop a systematic procedure to analyze discrete-time systems along lines similar to those used for continuous-time systems.

OPERATIONAL NOTATION

In difference equations it is convenient to use operational notation similar to that used in differential equations for the sake of compactness. In continuous-time systems we used the operator *D* to denote the operation of differentiation. For discrete-time systems we shall use the operator *E* to denote the operation for advancing a sequence by one time unit. Thus

$$\begin{aligned} Ex[n] &\equiv x[n+1] \\ E^2x[n] &\equiv x[n+2] \\ &\vdots \\ E^Nx[n] &\equiv x[n+N] \end{aligned} \tag{3.21}$$

The first-order difference equation of the savings account problem was found to be [see Eq. (3.9b)]

$$y[n+1] - ay[n] = x[n+1] \tag{3.22}$$

Using the operational notation, we can express this equation as

$$Ey[n] - ay[n] = Ex[n]$$

or

$$(E-a)y[n] = Ex[n] \tag{3.23}$$

The second-order difference equation (3.10b)

$$y[n+2]+\tfrac{1}{4}y[n+1]+\tfrac{1}{16}y[n]=x[n+2]$$

can be expressed in operational notation as

$$\left(E^2+\tfrac{1}{4}E+\tfrac{1}{16}\right)y[n]=E^2x[n]$$

A general Nth-order difference Eq. (3.17a) can be expressed as

$$\begin{aligned}(E^N+a_1E^{N-1}+\cdots+a_{N-1}E+a_N)y[n]\\ =(b_0E^N+b_1E^{N-1}+\cdots+b_{N-1}E+b_N)x[n]\end{aligned} \tag{3.24a}$$

or

$$Q[E]y[n]=P[E]x[n] \tag{3.24b}$$

where $Q[E]$ and $P[E]$ are Nth−order polynomial operators

$$Q[E]=E^N+a_1E^{N-1}+\cdots+a_{N-1}E+a_N \tag{3.25}$$

$$P[E]=b_0E^N+b_1E^{N-1}+\cdots+b_{N-1}E+b_N \tag{3.26}$$

RESPONSE OF LINEAR DISCRETE-TIME SYSTEMS

Following the procedure used for continuous-time systems, we can show that Eq. (3.24) is a linear equation (with constant coefficients). A system described by such an equation is a linear time-invariant, discrete-time (LTID) system. We can verify, as in the case of LTIC systems (see the footnote on page 152), that the general solution of Eq. (3.24) consists of zero-input and zero-state components.

[†]Equations such as (3.9), (3.10), (3.12), and (3.15) are considered to be linear according to the classical definition of linearity. Some authors label such equations as *incrementally linear.* We prefer the classical definition. It is just a matter of individual choice and makes no difference in the final results.

3.6 SYSTEM RESPONSE TO INTERNAL CONDITIONS: THE ZERO-INPUT RESPONSE

The zero-input response $y_0[n]$ is the solution of Eq. (3.24) with $x[n] = 0$; that is

$$Q[E]y_0[n]=0 \tag{3.27a}$$

or

$$(E^N+a_1E^{N-1}+\cdots+a_{N-1}E+a_N)y_0[n]=0 \tag{3.27b}$$

or

$$y_0[n+N]+a_1y_0[n+N-1]+\cdots+a_{N-1}y_0[n+1]+a_Ny_0[n]=0 \tag{3.27c}$$

We can solve this equation systematically. But even a cursory examination of this equation points to its solution. This equation states that a linear combination of $y_0[n]$ and advanced $y_0[n]$ is zero, *not for some values of n, but for all n.* Such situation is possible *if and only if* $y_0[n]$ and advanced $y_0[n]$ have the same form. Only an exponential function γ^n has this property, as the following equation indicates.

$$E^k\{\gamma^n\}=\gamma^{n+k}=\gamma^k\gamma^n \tag{3.28}$$

Equation (3.28) shows that γ^n advanced by k units is a constant (γ^k) times γ^n. Therefore, the solution of Eq. (3.27) must be of the form[†]

$$y_0[n]=c\gamma^n \tag{3.29}$$

To determine c and γ, we substitute this solution in Eq. (3.27b). Equation (3.29) yields

$$E^ky_0[n]=y_0[n+k]=c\gamma^{n+k} \tag{3.30}$$

Substitution of this result in Eq. (3.27b) yields

$$c(\gamma^N+a_1\gamma^{N-1}+\cdots+a_{N-1}\gamma+a_N)\gamma^n=0 \tag{3.31}$$

For a nontrivial solution of this equation

$$\gamma^N+a_1\gamma^{n-1}+\cdots+a_{N-1}\gamma+a_N=0 \tag{3.32a}$$

or

$$Q[\gamma]=0 \tag{3.32b}$$

Our solution $c\gamma^n$ [Eq. (3.29)] is correct, provided γ satisfies Eq. (3.32). Now, $Q[\gamma]$ is an Nth−order polynomial and can be expressed in the factored form (assuming all distinct roots):

$$(\gamma-\gamma_1)(\gamma-\gamma_2)\cdots(\gamma-\gamma_N)=0 \tag{3.32c}$$

Clearly, γ has N solutions γ_1, $?_2$,..., γ_N and, therefore, Eq. (3.27) also has N solutions $c_1\gamma^n{}_1$, $c_2?^n{}_2$,..., $c_n\gamma^n{}_N$. In such a case, we have shown that the general solution is a linear combination of the N solutions (see the footnote on page 153). Thus

$$y_0[n] = c_1\gamma_1^n + c_2\gamma_2^n + \cdots + c_n\gamma_N^n \qquad (3.33)$$

where $?_1$, $?_2$,..., γ_n are the roots of Eq. (3.32) and c_1, c_2,..., c_n are arbitrary constants determined from N auxiliary conditions, generally given in the form of initial conditions. The polynomial $Q[\gamma]$ is called the *characteristic polynomial* of the system, and

$$Q[\gamma] = 0 \qquad (3.34)$$

is the *characteristic equation* of the system. Moreover, $?_1$, $?_2$,..., γ_n, the roots of the characteristic equation, are called *characteristic roots or characteristic values* (also *eigenvalues*) of the system. The exponentials $\gamma^n{}_i$(i = 1, 2,..., N) are the *characteristic modes or natural modes* of the system. A characteristic mode corresponds to each characteristic root of the system, and the *zero-input response is a linear combination of the characteristic modes of the system.*

REPEATED ROOTS

So far we have assumed the system to have N distinct characteristic roots γ_1, γ_2,..., γ_N with corresponding characteristic modes $\gamma^n{}_1$, $?^n{}_2$,..., $\gamma^n{}_N$. If two or more roots coincide (repeated roots), the form of characteristic modes is modified. Direct substitution shows that if a root γ repeats r times (root of multiplicity r), the corresponding characteristic modes for this root are γ^n, $n\gamma^n$, $n^2?^n$..., $n^{r-1}\gamma^n$. Thus, if the characteristic equation of a system is

$$Q[\gamma] = (\gamma - \gamma_1)^r(\gamma - \gamma_{r+1})(\gamma - \gamma_{r+2})\cdots(\gamma - \gamma_N) \qquad (3.35)$$

the zero-input response of the system is

$$y_0[n] = (c_1 + c_2 n + c_3 n^2 + \cdots + c_r n^{r-1})\gamma_1^n + c_{r+1}\gamma_{r+1}^n + c_{r+2}\gamma_{r+2}^n + \cdots + c_n\gamma_N^n \qquad (3.36)$$

COMPLEX ROOTS

As in the case of continuous-time systems, the complex roots of a discrete-time system will occur in pairs of conjugates if the system equation coefficients are real. Complex roots can be treated exactly as we would treat real roots. However, just as in the case of continuous-time systems, we can also use the real form of solution as an alternative.

First we express the complex conjugate roots γ and γ * in polar form. If $|\gamma|$ is the magnitude and β is the angle of γ, then

$$\gamma = |\gamma|e^{j\beta} \qquad \text{and} \qquad \gamma^* = |\gamma|e^{-j\beta}$$

The zero-input response is given by

$$\begin{aligned} y_0[n] &= c_1\gamma^n + c_2(\gamma^*)^n \\ &= c_1|\gamma|^n e^{j\beta n} + c_2|\gamma|^n e^{-j\beta n} \end{aligned}$$

For a real system, c_1 and c_2 must be conjugates so that $y_0[n]$ is a real function of n. Let

$$c_1 = \frac{c}{2}e^{j\theta} \qquad \text{and} \qquad c_2 = \frac{c}{2}e^{-j\theta} \qquad (3.37a)$$

Then

$$\begin{aligned} y_0[n] &= \frac{c}{2}|\gamma|^n \left[e^{j(\beta n+\theta)} + e^{-j(\beta n+\theta)}\right] \\ &= c|\gamma|^n \cos(\beta n + \theta) \end{aligned} \qquad (3.37b)$$

where c and θ are arbitrary constants determined from the auxiliary conditions. This is the solution in real form, which avoids dealing with complex numbers.

EXAMPLE 3.10

---

a. For an LTID system described by the difference equation

$$y[n+2] - 0.6y[n+1] - 0.16y[n] = 5x[n+2] \qquad (3.38a)$$

find the total response if the initial conditions are $y[-1] = 0$ and $y[-2] = 25/4$, and if the input $x[n] = 4^{-n}u[n]$. In this example we shall determine the zero-input component $y_0[n]$ only. The zero-state component is determined later, in Example 3.14.

The system equation in operational notation is

$$(E^2 - 0.6E - 0.16)y[n] = 5E^2x[n] \qquad (3.38b)$$

The characteristic polynomial is

$$\gamma^2 - 0.6\gamma - 0.16 = (\gamma + 0.2)(\gamma - 0.8)$$

The characteristic equation is

$$(\gamma + 0.2)(\gamma - 0.8) = 0 \qquad (3.39)$$

The characteristic roots are $\gamma_1 = -0.2$ and $\gamma_2 = 0.8$. The zero-input response is

$$y_0[n] = c_1(-0.2)^n + c_2(0.8)^n \qquad (3.40)$$

To determine arbitrary constants c_1 and c_2, we set $n = -1$ and -2 in Eq. (3.40), then substitute $y_0[-1] = 0$ and $y_0[-2] = 25/4$ to obtain[†]

$$\left.\begin{aligned} 0 &= -5c_1 + \tfrac{5}{4}c_2 \\ \tfrac{25}{4} &= 25c_1 + \tfrac{25}{16}c_2 \end{aligned}\right\} \implies \begin{aligned} c_1 &= \tfrac{1}{5} \\ c_2 &= \tfrac{4}{5} \end{aligned}$$

Therefore

$$y_0[n] = \tfrac{1}{5}(-0.2)^n + \tfrac{4}{5}(0.8)^n \qquad n \ge 0 \qquad (3.41)$$

The reader can verify this solution by computing the first few terms using the iterative method (see Examples 3.8 and 3.9).

b. A similar procedure may be followed for repeated roots. For instance, for a system specified by the equation

$$(E^2 + 6E + 9)y[n] = (2E^2 + 6E)x[n]$$

Let us determine $y_0[n]$, the zero-input component of the response if the initial conditions are $y_0[-1] = -1/3$ and $y_0[-2] = -2/9$.

The characteristic polynomial is $\gamma_2 + 6\gamma + 9 = (\gamma + 3)^2$, and we have a repeated characteristic root at $\gamma = -3$. The characteristic modes are $(-3)^n$ and $n(-3)^n$. Hence, the zero-input response is

$$y_0[n] = (c_1 + c_2 n)(-3)^n$$

We can determine the arbitrary constants c_1 and c_2 from the initial conditions following the procedure in part (a). It is left as an exercise for the reader to show that $c_1 = 4$ and $c_2 = 3$ so that

$$y_0[n] = (4 + 3n)(-3)^n$$

c. For the case of complex roots, let us find the zero-input response of an LTID system described by the equation

$$(E^2 - 1.56E + 0.81)y[n] = (E + 3)x[n]$$

when the initial conditions are $y_0[-1] = 2$ and $y_0[-2] = 1$.

The characteristic polynomial is $(\gamma^2 - 1.56\gamma + 0.81) = (\gamma - 0.78 - j0.45)(\gamma - 0.78 + j0.45)$. The characteristic roots are $0.78 \pm j0.45$; that is, $0.9e^{\pm j(\pi/6)}$. We could immediately write the solution as

$$y_0[n] = c(0.9)ne^{j\pi n/6} + c^*(0.9)ne^{-j\pi n/6}$$

Setting $n = -1$ and -2 and using the initial conditions $y_0[-1] = 2$ and $y_0[-2] = 1$, we find $c = 2.34e^{-j0.17}$ and $c^* = 2.34e^{j0.17}$.

Alternately, we could also find the solution by using the real form of solution, as given in Eq. (3.37b). In the present case, the roots are $0.9e^{\pm j(\pi/6)}$. Hence, $|\gamma| = 0.9$ and $\beta = \pi/6$, and the zero-input response, according to Eq. (3.37b), is given by

$$y_0[n] = c(0.9)^n \cos\left(\frac{\pi}{6}n + \theta\right)$$

To determine the arbitrary constants c and θ, we set $n = -1$ and -2 in this equation and substitute the initial conditions $y_0[-1] = 2$, $y_0[-2] = 1$ to obtain

$$2 = \frac{c}{0.9}\cos\left(-\frac{\pi}{6} + \theta\right) = \frac{c}{0.9}\left[\frac{\sqrt{3}}{2}\cos\theta + \frac{1}{2}\sin\theta\right]$$

$$1 = \frac{c}{(0.9)^2}\cos\left(-\frac{\pi}{3} + \theta\right) = \frac{c}{0.81}\left[\frac{1}{2}\cos\theta + \frac{\sqrt{3}}{2}\sin\theta\right]$$

or

$$\frac{\sqrt{3}}{1.8}c\cos\theta + \frac{1}{1.8}c\sin\theta = 2$$

$$\frac{1}{1.62}c\cos\theta + \frac{\sqrt{3}}{1.62}c\sin\theta = 1$$

These are two simultaneous equations in two unknowns $c\cos\theta$ and $c\sin\theta$. Solution of these equations yields

$$c\cos\theta = 2.308$$

$$c\sin\theta = -0.397$$

Dividing $c\sin\theta$ by $c\cos\theta$ yields

$$\tan\theta = \frac{-0.397}{2.308} = \frac{-0.172}{1}$$
$$\theta = \tan^{-1}(-0.172) = -0.17 \text{ rad}$$

Substituting θ = −0.17 radian in $c \cos\theta = 2.308$ yields $c = 2.34$ and

$$y_0[n] = 2.34(0.9)^n \cos\left(\frac{\pi}{6}n - 0.17\right) n \geq 0$$

Observe that here we have used radian unit for both β and θ. We also could have used the degree unit, although this practice is not recommended. The important consideration is to be consistent and to use the same units for both β and θ.

EXERCISE E3.11

Find and sketch the zero-input response for the systems described by the following equations:

a. $y[n + 1] - 0.8y[n] = 3x[n + 1]$

b. $y[n + 1] + 0.8y[n] = 3x[n + 1]$

In each case the initial condition is $y[-1] = 10$. Verify the solutions by computing the first three terms using the iterative method.

Answers

a. $8(0.8)^n$

b. $-8(-0.8)^n$

EXERCISE E3.12

Find the zero-input response of a system described by the equation

$$y[n] + 0.3y[n-1] - 0.1y[n-2] = x[n] + 2x[n-1]$$

The initial conditions are $y_0[-1] = 1$ and $y_0[-2] = 33$. Verify the solution by computing the first three terms iteratively.

Answers

$y_0[n] = (0.2)^n + 2(-0.5)^n$

EXERCISE E3.13

Find the zero-input response of a system described by the equation

$$y[n] + 4y[n-2] = 2x[n]$$

The initial conditions are $y_0[-1] = -1/(2\sqrt{2})$ and $y_0[-2] = 1/(4\sqrt{2})$. Verify the solution by computing the first three terms iteratively.

Answers

$$y_0[n] = (2)^n \cos\left(\frac{\pi}{2}n - \frac{3\pi}{4}\right)$$

COMPUTER EXAMPLE C3.4

Using the initial conditions $y[-1] = 2$ and $y[-2] = 1$, find and sketch the zero-input response for the system described by $(E^2 - 1.56E + 0.81)y[n] = (E + 3)x[n]$.

```
>> n = (-2:20)'; y=[1;2;zeros(length(n)-2,1)];
>> for k = 1:length(n)-2
>>     y(k+2) = 1.56*y(k+1) -0.81*y(k);
>> end
>> clf; stem(n,y,'k'); xlabel('n'); ylabel(y[n]');
```

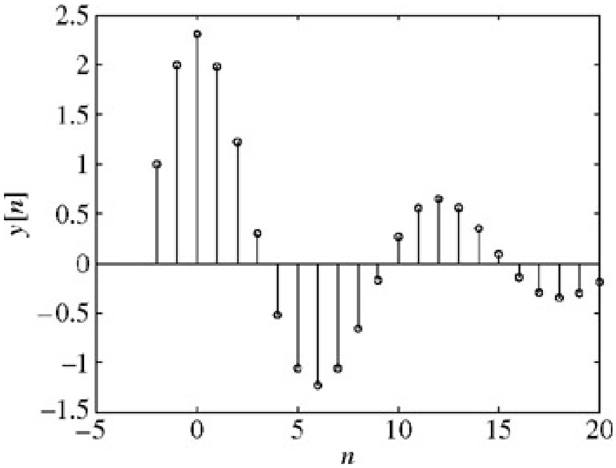

Figure C3.4

[†]A signal of the form $n^m \gamma^n$ also satisfies this requirement under certain conditions (repeated roots), discussed later.

[†]The initial conditions $y[-1]$ and $y[-2]$ are the conditions given on the total response. But because the input does not start until $n = 0$, the zero-state response is zero for $n < 0$. Hence, at $n = -1$ and -2 the total response consists of only the zero-input component, so that $y[-1] = y_0[-1]$ and $y[-2] = y_0[-2]$.

3.7 THE UNIT IMPULSE RESPONSE $h[n]$

Consider an nth-order system specified by the equation

$$(E^N + a_1 E^{N-1} + \cdots + a_{N-1}E + a_N)y[n] = (b_0 E^N + b_1 E^{N-1} + \cdots + b_{N-1}E + b_N)x[n] \tag{3.42a}$$

or

$$Q[E]y[n] = P[E]x[n] \tag{3.42b}$$

The unit impulse response $h[n]$ is the solution of this equation for the input $\delta[n]$ with all the initial conditions zero; that is,

$$Q[E]h[n] = P[E]\delta[n] \tag{3.43}$$

subject to initial conditions

$$h[-1] = h[-2] = \cdots = h[-N] = 0 \tag{3.44}$$

Equation (3.43) can be solved to determine $h[n]$ iteratively or in a closed form. The following example demonstrates the iterative solution.

EXAMPLE 3.11: (Iterative Determination of $h[n]$)

Find $h[n]$, the unit impulse response of a system described by the equation

$$y[n] - 0.6y[n-1] - 0.16y[n-2] = 5x[n] \tag{3.45}$$

To determine the unit impulse response, we let the input $x[n] = \delta[n]$ and the output $y[n] = h[n]$ in Eq. (3.45) to obtain

$$h[n] - 0.6h[n-1] - 0.16h[n-2] = 5\delta[n] \tag{3.46}$$

subject to zero initial state; that is, $h[-1] = h[-2] = 0$.

Setting $n = 0$ in this equation yields

$$h[0] - 0.6(0) - 0.16(0) = 5(1) \quad \Longrightarrow \quad h[0] = 5$$

Next, setting $n = 1$ in Eq. (3.46) and using $h[0] = 5$, we obtain

$$h[1] - 0.6(5) - 0.16(0) = 5(0) \quad \Longrightarrow \quad h[1] = 3$$

Continuing this way, we can determine any number of terms of $h[n]$. Unfortunately, such a solution does not yield a closed-form expression for $h[n]$. Nevertheless, determining a few values of $h[n]$ can be useful in determining the closed-form solution, as the following development shows.

THE CLOSED-FORM SOLUTION OF $h[n]$

Recall that $h[n]$ is the system response to input $\delta[n]$, which is zero for $n > 0$. We know that when the input is zero, only the characteristic modes can be sustained by the system. Therefore, $h[n]$ must be made up of characteristic modes for $n > 0$. At $n = 0$, it may have some nonzero value A_0, so that a general form of $h[n]$ can be expressed as[†]

$$h[n] = A_0\delta[n] + y_c[n]u[n] \tag{3.47}$$

where $y_c[n]$ is a linear combination of the characteristic modes. We now substitute Eq. (3.47) in Eq. (3.43) to obtain $Q[E]$ $(A_0\delta[n] + y_c[n]u[n] = P[E]\delta[n])$. Because $y_c[n]$ is made up of characteristic modes. $Q[E]y_c[n]u[n] = 0$, and we obtain $A_0 Q[E]\delta[n] = P[E]\delta[n]$, that is,

$$A_0(\delta[n+N] + a_1\delta[n+N-1] + \cdots + a_N\delta[n]) = b_0\delta[n+N] + \cdots + b_N\delta[n]$$

Setting $n = 0$ in this equation and using the fact that $\delta[m] = 0$ for all $m \neq 0$, and $\delta[0] = 1$, we obtain

$$A_0 a_N = b_N \quad \Longrightarrow \quad A_0 = \frac{b_N}{a_N} \tag{3.48}$$

Hence[‡]

$$h[n] = \frac{b_N}{a_N}\delta[n] + y_c[n]u[n] \tag{3.49}$$

The N unknown coefficients in $y_c[n]$ (on the right-hand side) can be determined from a knowledge of N values of $h[n]$. Fortunately, it is a straightforward task to determine values of $h[n]$ iteratively, as demonstrated in Example 3.11. We compute N values $h[0], h[1], h[2], \ldots, h[N-1]$ iteratively. Now, setting $n = 0, 1, 2, \ldots, N-1$ in Eq. (3.49), we can determine the N unknowns in $y_c[n]$. This point will become clear in the following example.

EXAMPLE 3.12

Determine the unit impulse response $h[n]$ for a system in Example 3.11 specified by the equation

$$y[n] - 0.6y[n-1] - 0.16y[n-2] = 5x[n]$$

This equation can be expressed in the advance operator form as

$$y[n+2] - 0.6y[n+1] - 0.16y[n] = 5x[n+2] \tag{3.50}$$

or

$$(E^2 - 0.6E - 0.16)y[n] = 5E^2x[n] \tag{3.51}$$

The characteristic polynomial is

$$\gamma^2 - 0.6\gamma - 0.16 = (\gamma + 0.2)(\gamma - 0.8)$$

The characteristic modes are $(-0.2)^n$ and $(0.8)^n$. Therefore

$$y_c[n] = c_1(-0.2)^n + c_2(0.8)^n \tag{3.52}$$

Also, from Eq. (3.51), we have $a_N = -0.16$ and $b_N = 0$. Therefore, according to Eq. (3.49)

$$h[n] = [c_1(-0.2)^n + c_2(0.8)^n]u[n] \tag{3.53}$$

To determine c_1 and c_2, we need to find two values of $h[n]$ iteratively. This step is already taken in Example 3.11, where we determined that $h[0] = 5$ and $h[1] = 3$. Now, setting $n = 0$ and 1 in Eq. (3.53) and using the fact that $h[0] = 5$ and $h[1] = 3$, we obtain

$$\left.\begin{aligned} 5 &= c_1 + c_2 \\ 3 &= -0.2c_1 + 0.8c_2 \end{aligned}\right\} \quad \Longrightarrow \quad \begin{aligned} c_1 &= 1 \\ c_2 &= 4 \end{aligned}$$

Therefore

$$h[n] = [(-0.2)^n + 4(0.8)^n]u[n] \tag{3.54}$$

COMPUTER EXAMPLE C3.5

Use MATLAB to solve Example 3.12.

There are several ways to find the impulse response using MATLAB. In this method, we first specify the input as a unit impulse function. Vectors `a` and `b` are created to specify the system. The `filter` command is then used to determine the impulse response. In fact, this method can be used to determine the zero-state response for any input.

```
>> n = (0: 19); x = inline ('n==0');
>> a = [1 -0.6 -0.16]; b = [5 0 0];
>> h = filter (b,a,x(n));
>> clf; stem(n,h,'k'); xlabel ('n'); ylabel('h[n]');
```

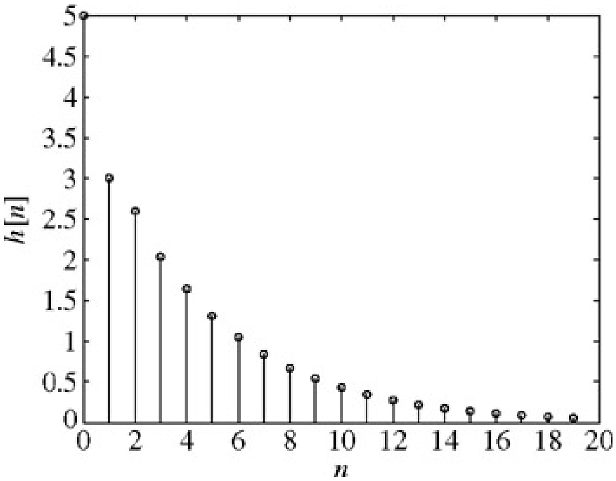

Figure C3.5

Comment. Although it is relatively simple to determine the impulse response $h[n]$ by using the procedure in this section, in Chapter 5 we shall discuss the much simpler method of the z-transform.

EXERCISE E3.14

Find $h[n]$, the unit impulse response of the LTID systems specified by the following equations:

a. $y[n+1] - y[n] = x[n]$

b. $y[n] - 5y[n-1] + 6y[n-2] = 8x[n-1] - 19x[n-2]$

c. $y[n+2] - 4y[n+1] + 4y[n] = 2x[n+2] - 2x[n+1]$

d. $y[n] = 2x[n] - 2x[n-1]$

Answers

a. $h[n] = u[n-1]$

b. $h[n] = -\frac{19}{6}\delta[n] + \left[\frac{3}{2}(2)^n + \frac{5}{3}(3)^n\right]u[n]$

c. $h[n] = (2+n)2^n u[n]$

d. $h[n] = 2\delta[n] - 2\delta[n-1]$

[†]We assumed that the term $y_c[n]$ consists of characteristic modes for $n > 0$ only. To reflect this behavior, the characteristic terms should be expressed in the form $\gamma^n_j u[n-1]$. But because $u[n-1] = u[n] - \delta[n]$, $c_j \gamma^n_j u[n-1] = c_j \gamma^n_j u[n] - c_j \delta[n]$, and $y_c[n]$ can be expressed in terms of exponentials $\delta^n_j u[n]$ (which start at $n = 0$), plus an impulse at $n = 0$.

[‡]If $a_N = 0$, then A_0 cannot be determined by Eq. (3.48). In such a case, we show in Section 3.12 that $h[n]$ is of the form $A_0\delta[n] + A_1\delta[n-1] + y_c[n]u[n]$. We have here $N + 2$ unknowns, which can be determined from $N + 2$ values $h[0]$, $h[1]$,..., $h[N+1]$ found iteratively.

3.8 SYSTEM RESPONSE TO EXTERNAL INPUT: THE ZERO-STATE RESPONSE

The zero-state response $y[n]$ is the system response to an input $x[n]$ when the system is in the zero state. In this section we shall assume that systems are in the zero state unless mentioned otherwise, so that the zero-state response will be the total response of the system. Here we follow the procedure parallel to that used in the continuous-time case by expressing an arbitrary input $x[n]$ as a sum of impulse components. A signal $x[n]$ in Fig. 3.16a can be expressed as a sum of impulse components such as those depicted in Fig. 3.16b-3.16f. The component of $x[n]$ at $n = m$ is $x[m]\delta[n-m]$, and $x[n]$ is the sum of all these components summed from $m = -\infty$ to ∞. Therefore

$$\begin{aligned} x[n] &= x[0]\delta[n] + x[1]\delta[n-1] + x[2]\delta[n-2] + \cdots \\ &\quad + x[-1]\delta[n+1] + x[-2]\delta[n+2] + \cdots \\ &= \sum_{m=-\infty}^{\infty} x[m]\delta[n-m] \end{aligned} \tag{3.55}$$

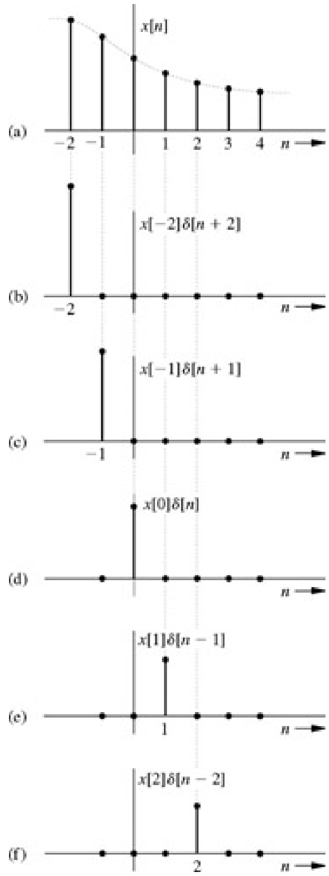

Figure 3.16: Representation of an arbitrary signal *x*[*n*] in terms of impulse components.

For a linear system, knowing the system response to impulse δ[*n*], the system response to any arbitrary input could be obtained by summing the system response to various impulse components. Let *h*[*n*] be the system response to impulse input δ[*n*]. We shall use the notation

$$x[n] \Longrightarrow y[n]$$

to indicate the input and the corresponding response of the system. Thus, if

$$\delta[n] \Longrightarrow h[n]$$

then because of time invariance

$$\delta[n-m] \Longrightarrow h[n-m]$$

and because of linearity

$$x[m]\delta[n-m] \Longrightarrow x[m]h[n-m]$$

and again because of linearity

$$\underbrace{\sum_{m=-\infty}^{\infty} x[m]\delta[n-m]}_{x[n]} \quad \Longrightarrow \quad \underbrace{\sum_{m=-\infty}^{\infty} x[m]h[n-m]}_{y[n]}$$

The left-hand side is *x*[*n*] [see Eq. (3.55)], and the right-hand side is the system response *y*[*n*] to input *x*[*n*]. Therefore[†]

$$y[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m] \qquad (3.56)$$

The summation on the right-hand side is known as the *convolution sum* of *x*[*n*] and *h*[*n*], and is represented symbolically by *x*[*n*] * *h*[*n*]

$$x[n] * h[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m] \qquad (3.57)$$

PROPERTIES OF THE CONVOLUTION SUM

The structure of the convolution sum is similar to that of the convolution integral. Moreover, the properties of the convolution sum are similar to those of the convolution integral. We shall enumerate these properties here without proof. The proofs are similar to those for the convolution integral and may be derived by the reader.

The Commutative Property.

$$x_1[n] * x_2[n] = x_2[n] * x_1[n] \qquad (3.58)$$

The Distributive Property.

$$x_1[n] * (x_2[n] + x_3[n]) = x_1[n] * x_2[n] + x_1[n] * x_3[n] \qquad (3.59)$$

The Associative Property.

$$x_1[n] * (x_2[n] * x_3[n]) = (x_1[n] * x_2[n]) * x_3[n] \qquad (3.60)$$

The Shifting Property. If

$$x_1[n] * x_2[n] = c[n]$$

then

$$x_1[n-m] * x_2[n-p] = c[n-m-p] \qquad (3.61)$$

The Convolution with an Impulse.

$$x[n] * \delta[n] = x[n] \qquad (3.62)$$

The Width Property. If $x_1[n]$ and $x_2[n]$ have finite widths of W_1 and W_2, respectively, then the width of $x_1[n] * x_2[n]$ is $W_1 + W_2$. The width of a signal is one less than the number of its elements (length). For instance, the signal in Fig. 3.17h has six elements (length of 6) but a width of only 5. Alternately, the property may be stated in terms of lengths as follows: if $x_1[n]$ and $x_2[n]$ have finite lengths of L_1 and L_2 elements, respectively, then the length of $x_1[n] * x_2[n]$ is $L_1 + L_2 - 1$ elements.

CAUSALITY AND ZERO-STATE RESPONSE

In deriving Eq. (3.56), we assumed the system to be linear and time invariant. There were no other restrictions on either the input signal or the system. In our applications, almost all the input signals are causal, and a majority of the systems are also causal. These restrictions further simplify the limits of the sum in Eq. (3.56). If the input $x[n]$ is causal, $x[m] = 0$ for $m < 0$. Similarly, if the system is causal (i.e., if $h[n]$ is causal), then $h[x] = 0$ for negative x, so that $h[n-m] = 0$ when $m > n$. Therefore, if $x[n]$ and $h[n]$ are both causal, the product $x[m]h[n-m] = 0$ for $m < 0$ and for $m > n$, and it is nonzero only for the range $0 \le m \le n$. Therefore, Eq. (3.56) in this case reduces to

$$y[n] = \sum_{m=0}^{n} x[m]h[n-m] \qquad (3.63)$$

We shall evaluate the convolution sum first by an analytical method and later with graphical aid.

EXAMPLE 3.13

Determine $c[n] = x[n] * g[n]$ for

$$x[n] = (0.8)^n u[n] \qquad \text{and} \qquad g[n] = (0.3)^n u[n]$$

We have

$$c[n] = \sum_{m=-\infty}^{\infty} x[m]g[n-m]$$

Note that

$$x[m] = (0.8)^m u[m] \qquad \text{and} \qquad g[n-m] = (0.3)^{n-m} u[n-m]$$

Both $x[n]$ and $g[n]$ are causal. Therefore, [see Eq. (3.63)]

$$\begin{aligned} c[n] &= \sum_{m=0}^{n} x[m]g[n-m] \\ &= \sum_{m=0}^{n} (0.8)^m u[m]\,(0.3)^{n-m} u[n-m] \end{aligned} \qquad (3.64)$$

In this summation, m lies between 0 and n ($0 \le m \le n$). Therefore, if $n \ge 0$, then both m and $n - m \le 0$, so that $u[m] = u[n-m] = 1$. If $n < 0$, m is negative because m lies between 0 and n, and $u[m] = 0$. Therefore, Eq. (3.64) becomes

$$\begin{aligned} c[n] &= \sum_{m=0}^{n} (0.8)^m\,(0.3)^{n-m} && n \ge 0 \\ &= 0 && n < 0 \end{aligned}$$

and

$$c[n] = (0.3)^n \sum_{m=0}^{n} \left(\frac{0.8}{0.3}\right)^m u[n]$$

This is a geometric progression with common ratio (0.8/0.3). From Section B.7-4 we have

$$
\begin{aligned}
c[n] &= (0.3)^n \frac{(0.8)^{n+1} - (0.3)^{n+1}}{(0.3)^n (0.8 - 0.3)} u[n] \\
&= 2[(0.8)^{n+1} - (0.3)^{n+1}]u[n] \qquad (3.65)
\end{aligned}
$$

EXERCISE E3.15

Show that $(0.8)^n u[n] * u[n] = 5[1 - (0.8)^{n+1}]u[n]$.

CONVOLUTION SUM FROM A TABLE

Just as in the continuous-time case, we have prepared a table (Table 3.1) from which convolution sums may be determined directly for a variety of signal pairs. For example, the convolution in Example 3.13 can be read directly from this table (pair 4) as

$$(0.8)^n u[n] * (0.3)^n u[n] = \frac{(0.8)^{n+1} - (0.3)^{n+1}}{0.8 - 0.3} u[n] = 2[(0.8)^{n+1} - (0.3)^{n+1}]u[n]$$

Table 3.1: Convolution Sums

No.	$x_1[n]$	$x_2[n]$	$x_1[n] * x_2[n] = x_2[n] * x_1[n]$
1	$\delta[n-k]$	$x[n]$	$x[n-k]$
2	$\gamma^n u[n]$	$u[n]$	$\left[\frac{1-\gamma^{n+1}}{1-\gamma}\right]u[n]$
3	$u[n]$	$u[n]$	$(n+1)u[n]$
4	$\gamma_1^n u[n]$	$\gamma_2^n u[n]$	$\left[\frac{\gamma_1^{n+1}-\gamma_2^{n+1}}{\gamma_1-\gamma_2}\right]u[n] \quad \gamma_1 \neq \gamma_2$
5	$u[n]$	$nu[n]$	$\frac{n(n+1)}{2}u[n]$
6	$\gamma^n u[n]$	$nu[n]$	$\left[\frac{\gamma(\gamma^n-1)+n(1-\gamma)}{(1-\gamma)^2}\right]u[n]$
7	$nu[n]$	$nu[n]$	$\frac{1}{6}n(n-1)(n+1)u[n]$
8	$\gamma^n u[n]$	$\gamma^n u[n]$	$(n+1)\gamma^n u[n]$
9	$n\gamma_1^n u[n]$	$\gamma_2^n u[n]$	$\frac{\gamma_1\gamma_2}{(\gamma_1-\gamma_2)^2}\left[\gamma_2^n - \gamma_1^n + \frac{\gamma_1-\gamma_2}{\gamma_2}n\gamma_1^n\right]u[n] \quad \gamma_1 \neq \gamma_2$
10	$\lvert\gamma_1\rvert^n \cos(\beta n + \theta)u[n]$	$\gamma_2^n u[n]$	$\frac{1}{R}[\lvert\gamma_1\rvert^{n+1}\cos[\beta(n+1)+\theta-\phi] - \gamma_2^{n+1}\cos(\theta-\phi)]u[n] \quad \gamma_2$ real $R = \left[\lvert\gamma_1\rvert^2 + \gamma_2^2 - 2\lvert\gamma_1\rvert\gamma_2\cos\beta\right]^{1/2}$ $\phi = \tan^{-1}\left[\frac{(\lvert\gamma_1\rvert\sin\beta)}{(\lvert\gamma_1\rvert\cos\beta - \gamma_2)}\right]$
11	$\gamma_1^n u[n]$	$\gamma_2^n u[-(n+1)]$	$\frac{\gamma_1}{\gamma_2-\gamma_1}\gamma_1^n u[n] + \frac{\gamma_2}{\gamma_2-\gamma_1}\gamma_2^n u[-(n+1)] \quad \lvert\gamma_2\rvert > \lvert\gamma_1\rvert$

We shall demonstrate the use of the convolution table in the following example.

EXAMPLE 3.14

Find the (zero-state) response $y[n]$ of an LTID system described by the equation

$$y[n+2] - 0.6y[n+1] - 0.16y[n] = 5x[n+2]$$

If the input $x[n] = 4^{-n}u[n]$.

The input can be expressed as $x[n] = 4^{-n}u[n] = (1/4)^n u[n] = (0.25)^n u[n]$. The unit impulse response of this system was obtained in Example 3.12.

$$h[n] = [(-0.2)^n + 4(0.8)^n]u[n]$$

Therefore

$$
\begin{aligned}
y[n] &= x[n] * h[n] \\
&= (0.25)^n u[n] * [(-0.2)^n u[n] + 4(0.8)^n u[n]] \\
&= (0.25)^n u[n] * (-0.2)^n u[n] + (0.25)^n u[n] * 4(0.8)^n u[n]
\end{aligned}
$$

We use pair 4 (Table 3.1) to find the foregoing convolution sums.

$$
\begin{aligned}
y[n] &= \left[\frac{(0.25)^{n+1} - (-0.2)^{n+1}}{0.25 - (-0.2)} + 4\frac{(0.25)^{n+1} - (0.8)^{n+1}}{0.25 - 0.8}\right]u[n] \\
&= (2.22[(0.25)^{n+1} - (-0.2)^{n+1}] - 7.27[(0.25)^{n+1} - (0.8)^{n+1}])u[n] \\
&= [-5.05(0.25)^{n+1} - 2.22(-0.2)^{n+1} + 7.27(0.8)^{n+1}]u[n]
\end{aligned}
$$

Recognizing that

$$\gamma^{n+1} = \gamma(\gamma)^n$$

We can express $y[n]$ as

$$
\begin{aligned}
y[n] &= [-1.26(0.25)^n + 0.444(-0.2)^n + 5.81(0.8)^n]u[n] \\
&= [-1.26(4)^{-n} + 0.444(-0.2)^n + 5.81(0.8)^n]u[n]
\end{aligned}
$$

EXERCISE E3.16

Show that

$$(0.8)^{n+1}u[n] * u[n] = 4[1 - 0.8(0.8)^n]u[n]$$

EXERCISE E3.17

Show that

$$n\,3^{-n}u[n] * (0.2)^n u[n] = \tfrac{15}{4}\left[(0.2)^n - \left(1 - \tfrac{2}{3}n\right)3^{-n}\right]u[n]$$

EXERCISE E3.18

Show that

$$e^{-n}u[n] * 2^{-n}u[n] = \frac{2}{2-e}\left[e^{-n} - \frac{e}{2}2^{-n}\right]u[n]$$

COMPUTER EXAMPLE C3.6

Find and sketch the zero-state response for the system described by $(E^2 + 6E + 9)y[n] = (2E^2 + 6E)x[n]$ for the input $x[n] = 4^{-n}u[n]$.

Although the input is bounded and quickly decays to zero, the system itself is unstable and an unbounded output results.

```
>> n = (0:11); x = inline('(4.^(-n)).*(n>=0)');
>> a = [1 6 9]; b = [2 6 0];
>> y = filter (b,a,x(n));
>> clf; stem(n,y,'k'); xlabel ('n'); ylabel('y[n]');
```

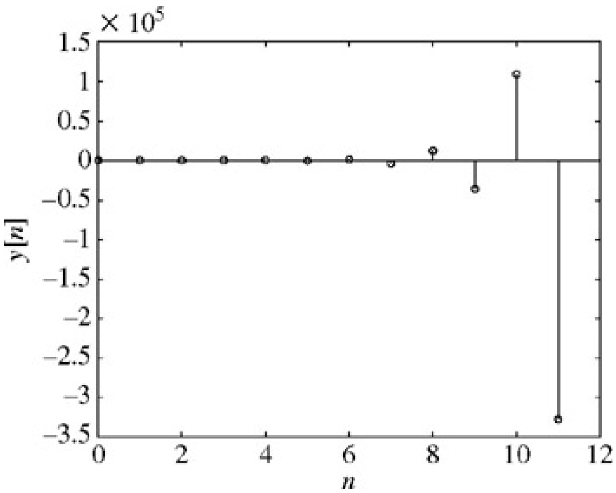

Figure C3.6

RESPONSE TO COMPLEX INPUTS

As in the case of real continuous-time systems, we can show that for an LTID system with real $h[n]$, if the input and the output are expressed in terms of their real and imaginary parts, then the real part of the input generates the real part of the response and the imaginary part of the input generates the imaginary part. Thus, if

$$x[n] = x_r[n] + jx_i[n] \quad \text{and} \quad y[n] = y_r[n] + jy_i[n] \qquad (3.66a)$$

using the right-directed arrow to indicate the input-output pair, we can show that

$$x_r[n] \Longrightarrow y_r[n] \quad \text{and} \quad x_i[n] \Longrightarrow y_i[n] \qquad (3.66b)$$

The proof is similar to that used to derive Eq. (2.40) for LTIC systems.

MULTIPLE INPUTS

Multiple inputs to LTI systems can be treated by applying the superposition principle. Each input is considered separately, with all other inputs assumed to be zero. The sum of all these individual system responses constitutes the total system output when all the inputs are applied simultaneously.

3.8-1 Graphical Procedure for the Convolution Sum

The steps in evaluating the convolution sum are parallel to those followed in evaluating the convolution integral. The convolution sum of causal signals $x[n]$ and $g[n]$ is given by

$$c[n] = \sum_{m=0}^{n} x[m]g[n-m]$$

We first plot $x[m]$ and $g[n-m]$ as functions of m (not n), because the summation is over m. Functions $x[m]$ and $g[m]$ are the same as $x[n]$ and $g[n]$, plotted respectively as functions of m (see Fig. 3.17). The convolution operation can be performed as follows:

1. Invert $g[m]$ about the vertical axis ($m = 0$) to obtain $g[-m]$ (Fig. 3.17d). Figure 3.17e shows both $x[m]$ and $g[-m]$.
2. Shift $g[-m]$ by n units to obtain $g[n-m]$. For $n > 0$, the shift is to the right (delay); for $n < 0$, the shift is to the left (advance). Figure 3.17f shows $g[n-m]$ for $n > 0$; for $n < 0$, see Fig. 3.17g.

3. Next we multiply $x[m]$ and $g[n-m]$ and add all the products to obtain $c[n]$. The procedure is repeated for each value of n over the range $-\infty$ to ∞.

We shall demonstrate by an example the graphical procedure for finding the convolution sum. Although both the functions in this example are causal, this procedure is applicable to general case.

EXAMPLE 3.15

Find

$$c[n] = x[n] * g[n]$$

where $x[n]$ and $g[n]$ are depicted in Fig. 3.17a and 3.17b, respectively.

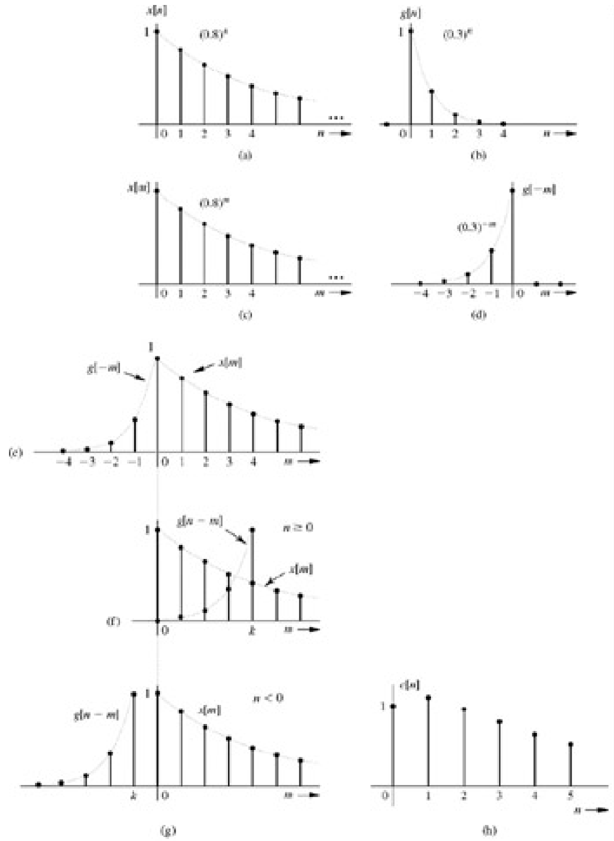

Figure 3.17: Graphical understanding of convolution of $x[n]$ and $g[n]$.

We are given

$$x[n] = (0.8)^n \qquad \text{and} \qquad g[n] = (0.3)^n$$

Therefore

$$x[m] = (0.8)^m \qquad \text{and} \qquad g[n-m] = (0.3)^{n-m}$$

Figure 3.17f shows the general situation for $n \geq 0$. The two functions $x[m]$ and $g[n-m]$ overlap over the interval $0 \leq m \leq n$. Therefore

$$\begin{aligned} c[n] &= \sum_{m=0}^{n} x[m]g[n-m] \\ &= \sum_{m=0}^{n} (0.8)^m (0.3)^{n-m} \\ &= (0.3)^n \sum_{m=0}^{n} \left(\frac{0.8}{0.3}\right)^m \\ &= 2[(0.8)^{n+1} - (0.3)^{n+1}] \qquad n \geq 0 \qquad \text{(see Section B.7-4)} \end{aligned}$$

For $n < 0$, there is no overlap between $x[m]$ and $g[n-m]$, as shown in Fig. 3.17g so that

$$c[n] = 0 \qquad n < 0$$

and

$$c[n] = 2[(0.8)^{n+1} - (0.3)^{n+1}]u[n]$$

which agrees with the earlier result in Eq. (3.65).

EXERCISE E3.19

Find $(0.8)^n u[n] * u[n]$ graphically and sketch the result.

Answers

$$5(1-(0.8)^{n+1})u[n]$$

AN ALTERNATIVE FORM OF GRAPHICAL PROCEDURE: THE SLIDING-TAPE METHOD

This algorithm is convenient when the sequences $x[n]$ and $g[n]$ are short or when they are available only in graphical form. The algorithm is basically the same as the graphical procedure in Fig. 3.17. The only difference is that instead of presenting the data as graphical plots, we display it as a sequence of numbers on tapes. Otherwise the procedure is the same, as will become clear in the following example.

EXAMPLE 3.16

Use the sliding-tape method to convolve the two sequences $x[n]$ and $g[n]$ depicted in Fig. 3.18a and 3.18b, respectively.

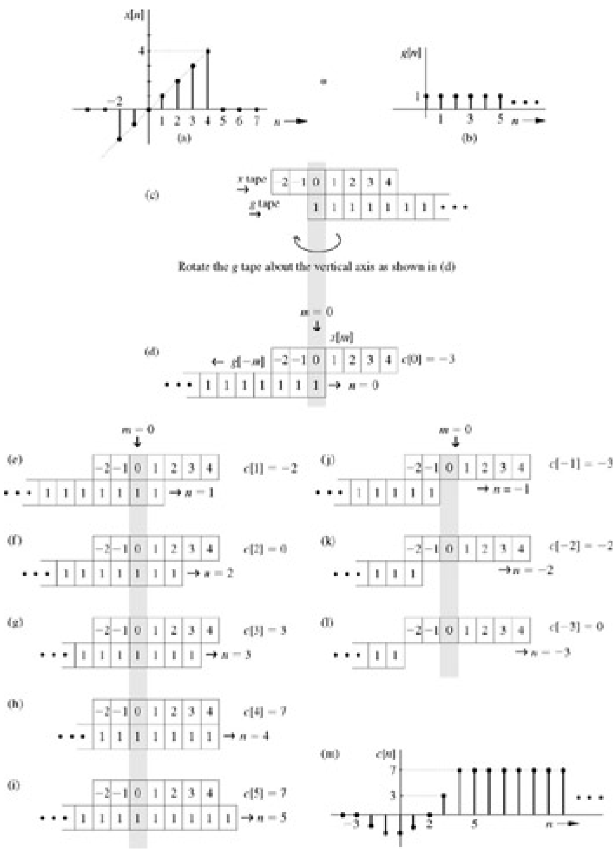

Figure 3.18: Sliding-tape algorithm for discrete-time convolution.

In this procedure we write the sequences $x[n]$ and $g[n]$ in the slots of two tapes: x tape and g tape (Fig. 3.18c). Now leave the x tape stationary (to correspond to $x[m]$). The $g[-m]$ tape is obtained by inverting the $g[m]$ tape about the origin ($m = 0$) so that the slots corresponding to $x[0]$ and $g[0]$ remain aligned (Fig. 3.18d). We now shift the inverted tape by n slots, multiply values on two tapes in adjacent slots, and add all the products to find $c[n]$. Figure 3.18d-3.18i shows the cases for n = 0-5. Figure 3.18j-3.18k, and 3.181 shows the cases for $n = -1, -2$, and -3, respectively.

For the case of $n = 0$, for example (Fig. 3.18d)

$$c[0] = (-2\times 1) + (-1\times 1) + (0\times 1) = -3$$

For $n = 1$ (Fig. 3.18e)

$$c[1] = (-2\times 1) + (-1\times 1) + (0\times 1) + (1\times 1) = -2$$

Similarly,

$$c[2] = (-2\times 1) + (-1\times 1) + (0\times 1) + (1\times 1) + (2\times 1) = 0$$

$$c[3] = (-2\times 1) + (-1\times 1) + (0\times 1) + (1\times 1) + (2\times 1) + (3\times 1) = 3$$

$$c[4] = (-2\times 1) + (-1\times 1) + (0\times 1) + (1\times 1) + (2\times 1) + (3\times 1) + (4\times 1) = 7$$

$$c[5] = (-2\times 1) + (-1\times 1) + (0\times 1) + (1\times 1) + (2\times 1) + (3\times 1) + (4\times 1) = 7$$

Figure 3.18i shows that $c[n] = 7$ for $n \geq 4$

Similarly, we compute $c[n]$ for negative n by sliding the tape backward, one slot at a time, as shown in the plots corresponding to $n = -1, -2$, and -3, respectively (Fig. 3.18j, 3.18k, and 3.181).

$$c[-1] = (-2 \times 1) + (-1 \times 1) = -3$$

$$c[-2] = (-2 \times 1) = -2$$

$$c[-3] = 0$$

Figure 3.181 shows that $c[n] = 0$ for $n \leq 3$. Figure 3.18m shows the plot of $c[n]$.

EXERCISE E3.20

Use the graphical procedure of Example 3.16 (sliding-tape technique) to show that $x[n]*g[n] = c[n]$ in Fig. 3.19. Verify the width property of convolution.

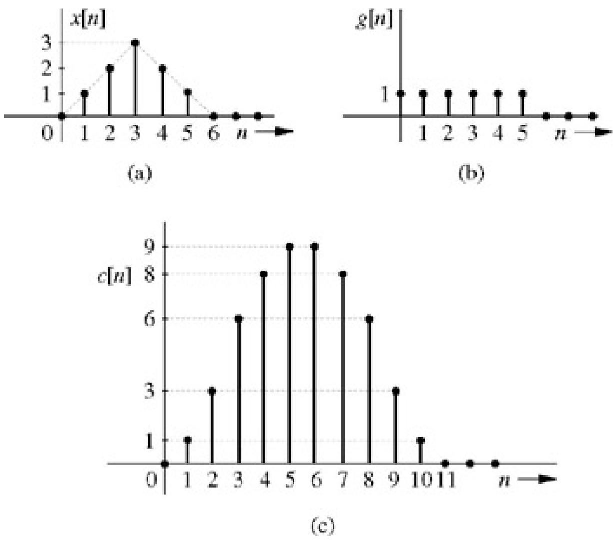

Figure 3.19

COMPUTER EXAMPLE C3.7

For the signals $x[n]$ and $g[n]$ depicted in Fig. 3.19, use MATLAB to compute and plot $c[n] = x[n] * g[n]$.

```
>> x = [0 1 2 3 2 1]; [1 1 1 1 1 1];
>> n= (0:1:length(x)+length(g)-2);
>> c=conv(x,g);
>> clf; stem(n, c, 'k'); xlabel('n'); ylabel('c[n]');
```

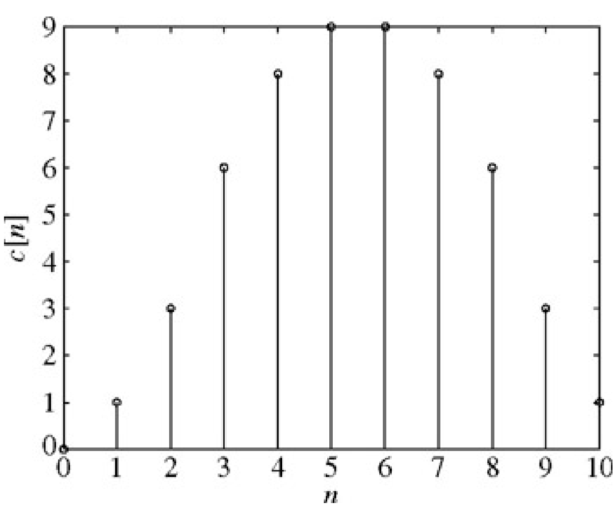

Figure C3.7

3.8-2 Interconnected Systems

As with continuous-time case, we can determine the impulse response of systems connected in parallel (Fig. 3.20a) and cascade (Fig. 3.20b, 3.20c). We can use arguments identical to those used for the continuous-time systems in Section 2.4-3 to show that if two LTID systems S_1 and S_2 with impulse response $h_1[n]$ and $h_2[n]$, respectively, are connected in parallel, the composite parallel system impulse response is $h_1[n] + h_2[n]$. Similarly, if these systems are connected in cascade (in any order), the impulse response of the composite system is $h_1[n] * h_2[n]$. Moreover, because $h_1[n] * h_2[n] = h_2[n] * h_1[n]$, linear systems commute. Their orders can be interchanged without affecting the composite system behavior.

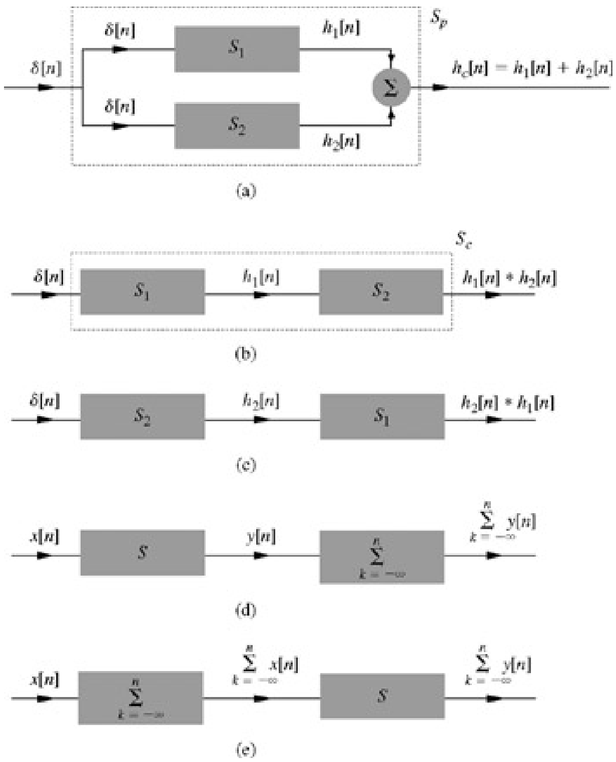

Figure 3.20: Interconnected systems.

INVERSE SYSTEMS

If the two systems in cascade are inverse of each other, with impulse responses $h[n]$ and $h_i[n]$, respectively, then the impulse response of the cascade of these systems is $h[n] * h_i[n]$. But, the cascade of a system with its inverse is an identity system, whose output is the same as the input.

Hence, the unit impulse response of an identity system is $\delta[n]$. Consequently

$$h[n] * h_i[n] = \delta[n] \tag{3.67}$$

As an example, we show that an accumulator system and a backward difference system are inverse of each other. An accumulator system is specified by[†]

$$y[n] = \sum_{k=-\infty}^{n} x[k] \tag{3.68a}$$

The backward difference system is specified by

$$y[n] = x[n] - x[n-1] \tag{3.68b}$$

From Eq. (3.68a), we find $h_{acc}[n]$, the impulse response of the accumulator, as

$$h_{acc}[n] = \sum_{k=-\infty}^{n} \delta[k] = u[n] \tag{3.69a}$$

Similarly, from Eq. (3.68b), $h_{bdf}[n]$, the impulse response of the backward difference system is given by

$$h_{bdf}[n] = \delta[n] - \delta[n-1] \tag{3.69b}$$

We can verify that

$$h_{acc} * h_{bdf} = u[n] * \{\delta[n] - \delta[n-1]\} = u[n] - u[n-1] = \delta[n]$$

Roughly speaking, in discrete-time systems, an accumulator is analogous to an integrator in continuous-time systems, and a backward difference system is analogous to a differentiator. We have already encountered examples of these systems in Examples 3.6 and 3.7 (digital differentiator and integrator).

SYSTEM RESPONSE TO $\sum_{k=-\infty}^{n} x[k]$

Figure 3.20d shows a cascade of two LTID systems: a system S with impulse response $h[n]$, followed by an accumulator. Figure 3.20e shows a cascade of the same two systems in reverse order: an accumulator followed by S. In Fig. 3.20d, if the input $x[n]$ to S results in the output $y[n]$, then the output of the system 3.20d is the $\Sigma\, y[k]$. In Fig. 3.20e, the output of the accumulator is the sum $\Sigma\, x[k]$. Because the output of the system in Fig. 3.20e is identical to that of system Fig. 3.20d, it follows that

$$\text{if} \quad x[n] \Longrightarrow y[n]$$

$$\text{then} \quad \sum_{k=-\infty}^{n} x[k] \Longrightarrow \sum_{k=-\infty}^{n} y[k] \qquad (3.70a)$$

If we let $x[n] = \delta[n]$ and $y[n] = h[n]$ in Eq. (3.70a), we find that $g[n]$, the unit step response of an LTID system with impulse response $h[n]$, is given by

$$g[n] = \sum_{k=-\infty}^{n} h[k] \qquad (3.70b)$$

The reader can readily prove the inverse relationship

$$h[n] = g[n] - g[n-1] \qquad (3.70c)$$

3.8-3 A Very Special Function for LTID Systems: The Everlasting Exponential z^n

In Section 2.4-4, we showed that there exists one signal for which the response of an LTIC system is the same as the input within a multiplicative constant. The response of an LTIC system to an everlasting exponential input e^{st} is $H(s)e^{st}$, where $H(s)$ is the system transfer function. We now show that for an LTID system, the same role is played by an everlasting exponential z^n. The system response $y[n]$ in this case is given by

$$\begin{aligned} y[n] &= h[n] * z^n \\ &= \sum_{m=-\infty}^{\infty} h[m] z^{n-m} \\ &= z^n \sum_{m=-\infty}^{\infty} h[m] z^{-m} \end{aligned}$$

For causal $h[n]$, the limits on the sum on the right-hand side would range from 0 to ∞. In any case, this sum is a function of z. Assuming that this sum converges, let us denote it by $H[z]$. Thus,

$$y[n] = H[z]z^n \qquad (3.71a)$$

where

$$H[z] = \sum_{m=-\infty}^{\infty} h[m] z^{-m} \qquad (3.71b)$$

Equation (3.71a) is valid only for values of z for which the sum on the right-hand side of Eq. (3.71b) exists (converges). Note that $H[z]$ is a constant for a given z. Thus, the input and the output are the same (within a multiplicative constant) for the everlasting exponential input z^n.

$H[z]$, which is called the *transfer function* of the system, is a function of the complex variable z. An alternate definition of the transfer function $H[z]$ of an LTID system from Eq. (3.71a) as

$$H[z] = \left.\frac{\text{output signal}}{\text{input signal}}\right|_{\text{input=everlasting exponential } z^n} \qquad (3.72)$$

The transfer function is defined for, and is meaningful to, LTID systems only. It does not exist for nonlinear or time-varying systems in general.

We repeat again that in this discussion we are talking of the everlasting exponential, which starts at $n = -\infty$, not the causal exponential $z^n u[n]$, which starts at $n = 0$.

For a system specified by Eq. (3.24), the transfer function is given by

$$H[z] = \frac{P[z]}{Q[z]} \qquad (3.73)$$

This follows readily by considering an everlasting input $x[n] = z^n$. According to Eq. (3.72), the output is $y[n] = H[z]z^n$. Substitution of this $x[n]$ and $y[n]$ in Eq. (3.24b) yields

$$H[z]\{Q[E]z^n\} = P[E]z^n$$

Moreover

$$E^k z^n = z^{n+k} = z^k z^n$$

Hence

$$P[E]z^n = P[z]z^n \qquad \text{and} \qquad Q[E]z^n = Q[z]z^n$$

Consequently,

$$H[z] = \frac{P[z]}{Q[z]}$$

EXERCISE E3.21

Show that the transfer function of the digital differentiator in Example 3.6 (big shaded block in Fig. 3.14b) is given by $H[z] = (z - 1)/Tz$, and the transfer function of an unit delay, specified by $y[n] = x[n - 1]$, is given by $1/z$.

3.8-4 Total Response

The total response of an LTID system can be expressed as a sum of the zero-input and zero-state components:

$$\text{total response} = \underbrace{\sum_{j=1}^{N} c_j \gamma_j^n}_{\text{zero-input component}} + \underbrace{x[n] * h[n]}_{\text{zero-state component}}$$

In this expression, the zero-input component should be appropriately modified for the case of repeated roots. We have developed procedures to determine these two components. From the system equation, we find the characteristic roots and characteristic modes. The zero-input response is a linear combination of the characteristic modes. From the system equation, we also determine $h[n]$, the impulse response, as discussed in Section 3.7. Knowing $h[n]$ and the input $x[n]$, we find the zero-state response as the convolution of $x[n]$ and $h[n]$. The arbitrary constants $c_1, c_2, \ldots, c_n$ in the zero-input response are determined from the n initial conditions. For the system described by the equation

$$y[n+2] - 0.6y[n+1] - 0.16y[n] = 5x[n+2]$$

with initial conditions $y[-1] = 0$, $y[-2] = 25/4$ and input $x[n] = (4)^{-n}u[n]$, we have determined the two components of the response in Examples 3.10a and 3.14, respectively. From the results in these examples, the total response for $n \geq 0$ is

$$\text{total response} = \underbrace{0.2(-0.2)^n + 0.8(0.8)^n}_{\text{zero-input component}} \underbrace{-1.26(4)^{-n} + 0.444(-0.2)^n + 5.81(0.8)^n}_{\text{zero-state component}} \qquad (3.74)$$

NATURAL AND FORCED RESPONSE

The characteristic modes of this system are $(-0.2)^n$ and $(0.8)^n$. The zero-input component is made up of characteristic modes exclusively, as expected, but the characteristic modes also appear in the zero-state response. When all the characteristic mode terms in the total response are lumped together, the resulting component is the *natural response.* The remaining part of the total response that is made up of noncharacteristic modes is the *forced response.* For the present case, Eq. (3.74) yields

$$\text{total response} = \underbrace{0.644(-0.2)^n + 6.61(0.8)^n}_{\text{natural response}} - \underbrace{1.26(4)^{-n}}_{\text{forced response}} \qquad n \geq 0 \qquad (3.75)$$

[†]In deriving this result, we have assumed a time-invariant system. The system response to input $\delta[n - m]$ for a time-varying system cannot be expressed as $h[n - m]$, but instead has the form $h[n, m]$. Using this form, Eq. (3.56) is modified as follows:

$$y[n] = \sum_{m=-\infty}^{\infty} x[m]h[n, m]$$

[†]Equations (3.68a) and (3.68b) are identical to Eqs. (3.14a) and (3.12), respectively, with $T = 1$.

3.9 CLASSICAL SOLUTION OF LINEAR DIFFERENCE EQUATIONS

As in the case of LTIC systems, we can use the classical method, in which the response is obtained as a sum of natural and forced components of the response, to analyze LTID systems.

FINDING NATURAL AND FORCED RESPONSE

As explained earlier, the *natural response* of a system consists of all the characteristic mode terms in the response. The remaining noncharacteristic mode terms form the *forced response.* If $y_c[n]$ and $y_\phi[n]$ denote the natural and the forced response respectively, then the total response is given by

$$\text{total response} = \underbrace{y_c[n]}_{\text{modes}} + \underbrace{y_\phi[n]}_{\text{nonmodes}} \qquad (3.76)$$

Because the total response $y_c[n] + y_\phi[n]$ is a solution of the system equation (3.24b), we have

$$Q[E](y_c[n] + y_\phi[n]) = P[E]x[n] \qquad (3.77)$$

But since $y_c[n]$ is made up of characteristic modes,

$$Q[E]y_c[n] = 0 \qquad (3.78)$$

Substitution of this equation in Eq. (3.77) yields

$$Q[E]y_\phi[n] = P[E]x[n] \qquad (3.79)$$

The natural response is a linear combination of characteristic modes. The arbitrary constants (multipliers) are determined from suitable auxiliary conditions usually given as $y[0]$, $y[1]$,..., $y[n-1]$. The reasons for using auxiliary instead of initial conditions are explained later. If we are given initial conditions $y[-1]$, $y[-2]$,..., $y[-N]$, we can readily use iterative procedure to derive the auxiliary conditions $y[0]$, $y[1]$,..., $y[N-1]$. We now turn our attention to the forced response.

Table 3.2: Forced Response

➡ Open table as spreadsheet

No.	Input $x[n]$	Forced response $y_\varphi[n]$
1	r^n r ? $?_i$ (i = 1, 2,..., N)	cr^n
2	r^n $r = \gamma_i$	cnr^n
3	$\cos(\beta n + \theta)$	$c \cos(\beta n + \phi)$
4	$\left(\sum_{i=0}^{m} \alpha_i n^i\right) r^n$	$\left(\sum_{i=0}^{m} c_i n^i\right) r^n$
Note: By definition, $y\phi[n]$ cannot have any characteristic mode terms. Should any term shown in the right-hand column for the forced response be a characteristic mode of the system, the correct form of the forced response must be modified to $n^i y\phi[n]$, where i is the smallest integer that will prevent $n^i y\phi[n]$ from having a characteristic mode term. For example, when the input is r^n, the forced response in the right-hand column is of the form cr^n. But if r^n happens to be a natural mode of the system, the correct form of the forced response is cnr^n(see pair 2).		

The forced response $y\phi[n]$ satisfies Eq. (3.79) and, by definition, contains only nonmode terms. To determine the forced response, we shall use the method of undetermined coefficients, the same method used for the continuous-time system. However, rather than retracing all the steps of the continuous-time system, we shall present a table (Table 3.2) listing the inputs and the corresponding forms of forced function with undetermined coefficients. These coefficients can be determined by substituting $y\phi[n]$ in Eq. (3.79) and equating the coefficients of similar terms.

EXAMPLE 3.17

Solve

$$(E^2 - 5E + 6)y[n] = (E - 5)x[n] \qquad (3.80)$$

if the input $x[n] = (3n + 5)u[n]$ and the auxiliary conditions are $y[0] = 4$, $y[1] = 13$.

The characteristic equation is

$$\gamma^2 - 5\gamma + 6 = (\gamma - 2)(\gamma - 3) = 0$$

Therefore, the natural response is

$$y_c[n] = B_1(2)^n + B_2(3)^n$$

To find the form of forced response $y\phi[n]$, we use Table 3.2, pair 4, with $r = 1$, $m = 1$. This yields

$$y_\phi[n] = c_1 n + c_0$$

Therefore

$$y_\phi[n+1] = c_1(n+1) + c_0 = c_1 n + c_1 + c_0$$

$$y_\phi[n+2] = c_1(n+2) + c_0 = c_1 n + 2c_1 + c_0$$

Also

$$x[n] = 3n + 5$$

and

$$x[n+1] = 3(n+1) + 5 = 3n + 8$$

Substitution of these results in Eq. (3.79) yields

$$c_1 n + 2c_1 + c_0 - 5(c_1 n + c_1 + c_0) + 6(c_1 n + c_0) = 3n + 8 - 5(3n + 5)$$

or

$$2c_1 n - 3c_1 + 2c_0 = -12n - 17$$

Comparison of similar terms on the two sides yields

$$\left.\begin{aligned} 2c_1 &= -12 \\ -3c_1 + 2c_0 &= -17 \end{aligned}\right\} \Longrightarrow \begin{aligned} c_1 &= -6 \\ c_2 &= -\tfrac{35}{2} \end{aligned}$$

Therefore

$$y_\phi[n] = -6n - \tfrac{35}{2}$$

The total response is

$$y[n] = y_c[n] + y_\phi[n]$$
$$= B_1(2)^n + B_2(3)^n - 6n - \tfrac{35}{2} \qquad n \geq 0$$

To determine arbitrary constants B_1 and B_2 we set n = 0 and 1 and substitute the auxiliary conditions $y[0]$ = 4, $y[1]$ = 13 to obtain

$$\left.\begin{aligned} 4 &= B_1 + B_2 - \tfrac{35}{2} \\ 13 &= 2B_1 + 3B_2 - \tfrac{47}{2} \end{aligned}\right\} \Longrightarrow \begin{aligned} B_1 &= 28 \\ B_2 &= \tfrac{-13}{2} \end{aligned}$$

Therefore

$$y_c[n] = 28(2)^n - \tfrac{13}{2}(3)^n \qquad (3.81)$$

and

$$y[n] = \underbrace{28(2)^n - \tfrac{13}{2}(3)^n}_{y_c[n]} \underbrace{-6n - \tfrac{35}{2}}_{y_\phi[n]} \qquad (3.82)$$

COMPUTER EXAMPLE C3.8

Use MATLAB to solve Example 3.17.

```
>> n = (-0:10)'; y=[4;13;zeros(length(n)-2,1); x = [3*n+5).*(n>=0);
>> for k = 1:length(n)-2
>> y(k+2) = 5*y(k+1) - 6*y(k) + x(k+1) - 5*x(k);
>> end;
>> clf; stem(n,y,'k'); xlabel('n'); ylabel('y[n]');
>> disp(' n y'); disp([num2str([n,y])]);
n       y
0       4
1      13
2      24
3      13
4    -120
5    -731
6   -3000
7  -10691
8  -35544
9 -113675
10 -355224
```

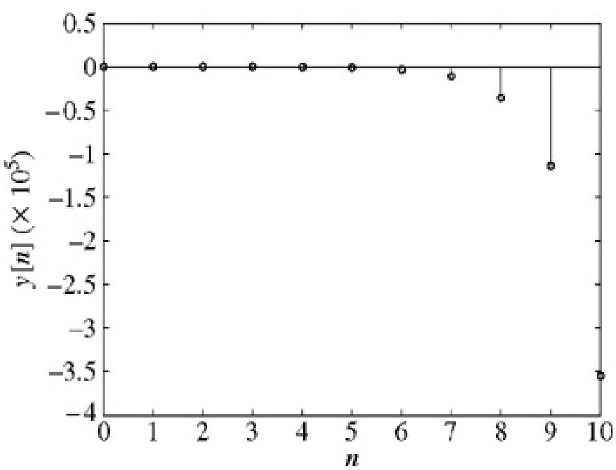

Figure C3.8

EXAMPLE 3.18

Find the sum $y[n]$ if

$$y[n] = \sum_{k=0}^{n} k^2 \qquad (3.83)$$

Such problems can be solved by finding an appropriate difference equation that has $y[n]$ as the response. From Eq. (3.83), we observe that $y[n + 1] = y[n] + (n + 1)^2$. Hence

$$y[n+1] - y[n] = (n+1)^2 \qquad (3.84)$$

This is the equation we seek. For this first-order difference equation, we need one auxiliary condition, the value of $y[n]$ at n = 0. From Eq. (3.83), it follows that $y[0]$ = 0. Thus, we seek the solution of Eq. (3.83) subject to an auxiliary condition $y[0]$ = 0.

The characteristic equation of Eq. (3.83) is $\gamma - 1 = 0$, the characteristic root is $\gamma = 1$, and the characteristic mode is $c(1)^n = cu[n]$, where c is an arbitrary constant. Clearly, the natural response is $cu[n]$.

The input, $x[n] = (n + 1)^2 = n^2 + 2n + 1$, is of the form in pair 4 (Table 3.2) with $r = 1$ and $m = 2$. Hence, the desired forced response is

$$y_\phi[n] = \beta_2 n^2 + \beta_1 n + \beta_0$$

Note, however, the term β_0 in $y\phi[n]$ is of the form of characteristic mode. Hence, the correct form is $y\phi[n] = \beta_3 n^3 + \beta_1 n^2 + \beta_0 n$. Therefore

$$Ey_\phi[n] = y_\phi[n+1] = \beta_2(n+1)^3 + \beta_1(n+1)^2 + \beta_0(n+1)$$

From Eq. (3.79), we obtain

$$(E-1)y_\phi[n] = n^2 + 2n + 1$$

or

$$[\beta_2(n+1)^3 + \beta_1(n+1)^2 + \beta_0(n+1)] - [\beta_2 n^3 + \beta_1 n^2 + \beta_0 n] = n^2 + 2n + 1$$

Equating the coefficients of similar powers, we obtain

$$\beta_0 = \frac{1}{6} \qquad \beta_1 = \frac{1}{2} \qquad \beta_2 = \frac{1}{3}$$

Hence

$$y[n] = c + \frac{2n^3 + 3n^2 + n}{6} = c + \frac{n(n+1)(2n+1)}{6}$$

Setting $n = 0$ in this equation and using the auxiliary condition $y[0] = 0$, we find $c = 0$ and

$$y[n] = \frac{2n^3 + 3n^2 + n}{6} = \frac{n(n+1)(2n+1)}{6}$$

COMMENTS ON AUXILIARY CONDITIONS

This (classical) method requires auxiliary conditions $y[0]$, $y[1]$,..., $y[N - 1]$. This is because at n = -1, -2,..., $-N$, only the zero-input component exists, and these initial conditions can be applied to the zero-input component only. In the classical method, the zero-input and zero-state components cannot be separated. Consequently, the initial conditions must be applied to the total response, which begins at $n = 0$. Hence, we need the auxiliary conditions $y[0]$, $y[1]$,..., $y[N - 1]$. If we are given the initial conditions $y[-1]$, $y[-2]$,..., $y[-N]$, we can use iterative procedure to derive the auxiliary conditions $y[0]$, $y[1]$,..., $y[n - 1]$.

AN EXPONENTIAL INPUT

As in the case of continuous-time systems, we can show that for a system specified by the equation

$$Q[E]y[n] = P[E]x[n] \tag{3.85}$$

the forced response for the exponential input $x[n] = r^n$ is given by

$$y_\phi[n] = H[r]r^n \qquad r \neq \gamma_i \text{ (a characteristic mode)} \tag{3.86}$$

where

$$H[r] = \frac{P[r]}{Q[r]} \tag{3.87}$$

The proof follows from the fact that if the input $x[n] = r^n$, then from Table 3.2 (pair 4), $y\phi[n] = cr^n$. Therefore

$$E^i x[n] = x[n+i] = r^{n+i} = r^i r^n \qquad \text{and} \qquad P[E]x[n] = P[r]r^n$$

$$E^k y_\phi[n] = y_\phi[n+k] = cr^{n+k} = cr^k r^n \qquad \text{and} \qquad Q[E]y[n] = cQ[r]r^n$$

so that Eq. (3.85) reduces to

$$cQ[r]r^n = P[r]r^n$$

which yields $c = P[r]/Q[r] = H[r]$.

This result is valid only if r is not a characteristic root of the system. If r is a characteristic root, then the forced response is cnr^n where c is determined by substituting $y\phi[n]$ in Eq. (3.79) and equating coefficients of similar terms on the two sides. Observe that the exponential r^n includes a wide variety of signals such as a constant C, a sinusoid $\cos(\beta n + \theta)$, and an exponentially growing or decaying sinusoid $|\gamma|^n \cos(\beta n + \theta)$.

A Constant Input *x*[*n*] = *C*. This is a special case of exponential Cr^n with $r = 1$. Therefore, from Eq. (3.86), we have

$$y_\phi[n] = C\frac{P[1]}{Q[1]} = CH[1] \tag{3.88}$$

A Sinusoidal Input. The input $e^{j\Omega n}$ is an exponential r^n with $r = e^{j\Omega}$. Hence

$$y_\phi[n] = H[e^{j\Omega}]e^{j\Omega n} = \frac{P[e^{j\Omega}]}{Q[e^{j\Omega}]}e^{j\Omega n}$$

$-j\Omega n$

Similarly, for the input *e*

$$y_\phi[n] = H[e^{-j\Omega}]e^{-j\Omega n}$$

Consequently, if the input $x[n] = \cos \Omega n = 1/2(e^{j\Omega n} + e^{-j\Omega n})$, then

$$y_\phi[n] = \tfrac{1}{2}\{H[e^{j\Omega}]e^{j\Omega n} + H[e^{-j\Omega}]e^{-j\Omega n}\}$$

Since the two terms on the right-hand side are conjugates,

$$y_\phi[n] = \mathrm{Re}\{H[e^{j\Omega}]e^{j\Omega n}\}$$

If

$$H[e^{j\Omega}] = |H[e^{j\Omega}]|e^{j\angle H[e^{j\Omega}]}$$

then

$$\begin{aligned} y_\phi[n] &= \mathrm{Re}\{|H[e^{j\Omega}]|e^{j(\Omega n + \angle H[e^{j\Omega}])}\} \\ &= |H[e^{j\Omega}]|\cos(\Omega n + \angle H[e^{j\Omega}]) \qquad (3.89a) \end{aligned}$$

Using a similar argument, we can show that for the input

$$x[n] = \cos(\Omega n + \theta)$$
$$y_\phi[n] = |H[e^{j\Omega}]|\cos(\Omega n + \theta + \angle H[e^{j\Omega}]) \qquad (3.89b)$$

EXAMPLE 3.19

For a system specified by the equation

$$(E^2 - 3E + 2)y[n] = (E + 2)x[n]$$

find the forced response for the input $x[n] = (3)^n u[n]$.

In this case

$$H[r] = \frac{P[r]}{Q[r]} = \frac{r+2}{r^2 - 3r + 2}$$

and the forced response to input $(3)^n u[n]$ is $H[3](3)^n u[n]$; that is,

$$y_\phi[n] = \frac{3+2}{(3)^2 - 3(3) + 2}(3)^n u[n] = \frac{5}{2}(3)^n u[n] \qquad n \geq 0$$

EXAMPLE 3.20

For an LTID system described by the equation

$$(E^2 - E + 0.16)y[n] = (E + 0.32)x[n]$$

determine the forced response $y_\phi[n]$ if the input is

$$x[n] = \cos\left(2n + \frac{\pi}{3}\right)u[n]$$

Here

$$H[r] = \frac{P[r]}{Q[r]} = \frac{r + 0.32}{r^2 - r + 0.16}$$

For the input $\cos(2n + (\pi/3))u[n]$, the forced response is

$$y_\phi[n] = |H[e^{j2}]|\cos\left(2n + \frac{\pi}{3} + \angle H[e^{j2}]\right)u[n]$$

where

$$\begin{aligned} H[e^{j2}] &= \frac{e^{j2} + 0.32}{(e^{j2})^2 - e^{j2} + 0.16} = \frac{(-0.416 + j0.909) + 0.32}{(-0.654 - j0.757) - (-0.416 + j0.909) + 0.16} \\ &= 0.548e^{j3.294} \end{aligned}$$

Therefore

$$|H[e^{j2}]| = 0.548 \qquad \text{and} \qquad \angle H[e^{j2}] = 3.294$$

so that

$$\begin{aligned} y_\phi[n] &= 0.548\cos\left(2n + \frac{\pi}{3} + 3.294\right)u[n] \\ &= 0.548\cos(2n + 4.34)u[n] \end{aligned}$$

ASSESSMENT OF THE CLASSICAL METHOD

The remarks in Chapter 2 concerning the classical method for solving differential equations also apply to difference equations.

3.10 SYSTEM STABILITY: THE EXTERNAL (BIBO) STABILITY CRITERION

Concepts and criteria for the BIBO (external) stability and the internal (asymptotic) stability for discrete-time systems are identical to those corresponding to continuous-time systems. The comments in Section 2.6 for LTIC systems concerning the distinction between the external and the internal stability are also valid for LTID systems.

Recall that

$$y[n] = h[n] * x[n]$$
$$= \sum_{m=-\infty}^{\infty} h[m]x[n-m]$$

and

$$|y[n]| = \left| \sum_{m=-\infty}^{\infty} h[m]x[n-m] \right|$$
$$\leq \sum_{m=-\infty}^{\infty} |h[m]|\,|x[n-m]|$$

If $x[n]$ is bounded, then $|x[n-m]| < K_1 < \infty$, and

$$|y[n]| \leq K_1 \sum_{m=-\infty}^{\infty} |h[m]|$$

Clearly the output is bounded if the summation on the right-hand side is bounded; that is, if

$$\sum_{n=-\infty}^{\infty} |h[n]| < K_2 < \infty \tag{3.90}$$

This is a sufficient condition for BIBO stability. We can show that this is also a necessary condition (see Prob. 3.10-1). Therefore, for an LTID system, if its impulse response $h[n]$ is absolutely summable, the system is (BIBO) stable. Otherwise it is unstable.

All the comments about the nature of the external and the internal stability in Chapter 2 apply to discrete-time case. We shall not elaborate them further.

3.10-1 Internal (Asymptotic) Stability

For LTID systems, as in the case of LTIC systems, the internal stability, called the asymptotical stability or the stability in the sense of Lyapunov (also the zero-input stability) is defined in terms of the zero-input response of a system.

For an LTID system specified by a difference equation of the form (3.17) [or (3.24)], the zero-input response consists of the characteristic modes of the system. The mode corresponding to a characteristic root γ is γ^n. To be more general, let γ be complex so that

$$\gamma = |\gamma|e^{j\beta} \qquad \text{and} \qquad \gamma^n = |\gamma|^n e^{j\beta n}$$

Since the magnitude of $e^{n\beta n}$ is always unity regardless of the value of n, the magnitude of γ^n is $|\gamma|^n$. Therefore

$$\begin{array}{llll} \text{if} & |\gamma| < 1, & \gamma^n \to 0 & \text{as } n \to \infty \\ \text{if} & |\gamma| > 1, & \gamma^n \to \infty & \text{as } n \to \infty \\ \text{and if} & |\gamma| = 1, & |\gamma|^n = 1 & \text{for all } n \end{array}$$

Figure 3.21 shows the characteristic modes corresponding to characteristic roots at various locations in the complex plane

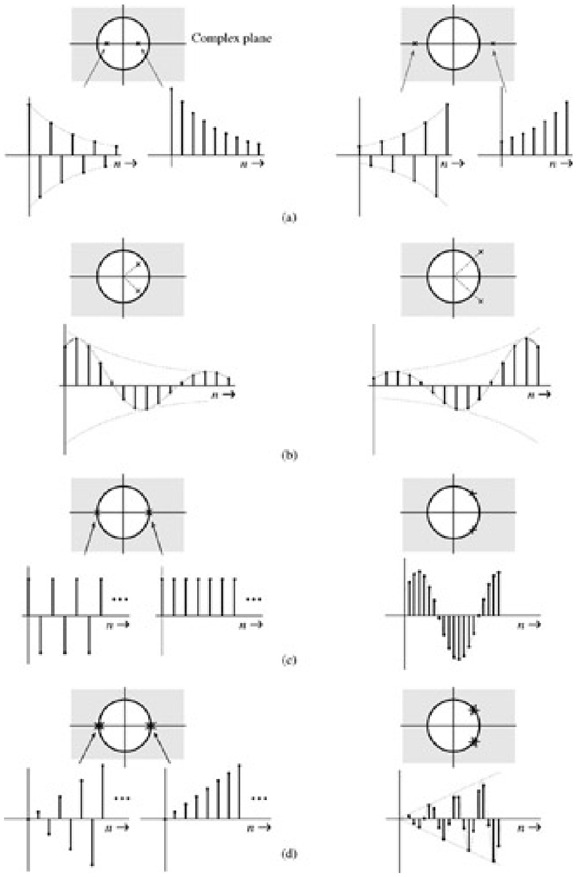

Figure 3.21: Characteristic roots location and the corresponding characteristic modes.

These results can be grasped more effectively in terms of the location of characteristic roots in the complex plane. Figure 3.22 shows a circle of unit radius, centered at the origin in a complex plane. Our discussion shows that if all characteristic roots of the system lie inside the *unit circle*, $|\gamma_i| < 1$ for all i and the system is asymptotically stable. On the other hand, even if one characteristic root lies outside the unit circle, the system is unstable. If none of the characteristic roots lie outside the unit circle, but some simple (unrepeated) roots lie on the circle itself, the system is marginally stable. If two or more characteristic roots coincide on the unit circle (repeated roots), the system is unstable. The reason is that for repeated roots, the zero-input response is of the form $n^{r-1}\gamma^n$, and if $|\gamma| = 1$, then $|n^{r-1}\gamma^n| = n^{r-1} \to \infty$ as $n \to \infty$.[†]

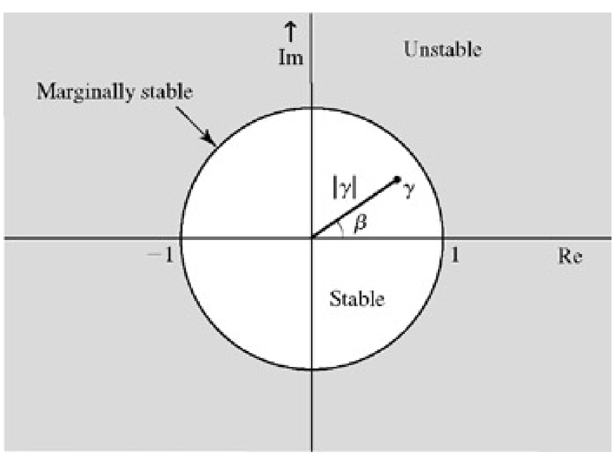

Figure 3.22: Characteristic root locations and system stability.

Note, however, that repeated roots inside the unit circle do not cause instability. To summarize:

1. An LTID system is asymptotically stable if and only if all the characteristic roots are inside the unit circle. The roots may be simple or repeated.
2. An LTID system is unstable if and only if either one or both of the following conditions exist: (i) at least one root is outside the unit circle; (ii) there are repeated roots on the unit circle.
3. An LTID system is marginally stable if and only if there are no roots outside the unit circle and there are some unrepeated roots on the unit circle.

3.10-2 Relationship Between BIBO and Asymptotic Stability

For LTID systems, the relation between the two types of stability is similar to those in LTIC systems. For a system specified by Eq. (3.17), we can readily show that if a characteristic root γ_k is inside the unit circle, the corresponding mode $\gamma^n{}_k$ is absolutely summable. In contrast, if γ_k lies outside the unit circle, or on the unit circle, $\gamma^n{}_k$ is not absolutely summable.[†]

This means that an asymptotically stable system is BIBO stable. Moreover, a marginally stable or asymptotically unstable system is BIBO unstable. The converse is not necessarily true. The stability picture portrayed by the external description is of questionable value. BIBO (external) stability cannot ensure internal (asymptotic) stability, as the following example shows.

EXAMPLE 3.21

An LTID systems consists of two subsystems S_1 and S_2 in cascade (Fig. 3.23). The impulse response of these systems are $h_1[n]$ and $h_2[n]$, respectively, given by

$$h_1[n] = 4\delta[n] - 3(0.5)^n u[n] \qquad \text{and} \qquad h_2[n] = 2^n u[n]$$

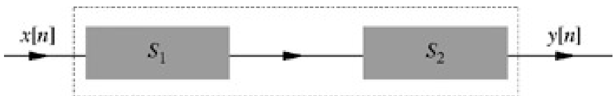

Figure 3.23: BIBO and asymptotic stability.

The composite system impulse response $h[n]$ is given by

$$\begin{aligned} h[n] = h_1[n] * h_2[n] = h_2[n] * h_1[n] &= 2^n u[n] * (4\delta[n] - 3(0.5)^n u[n]) \\ &= 4(2)^n u[n] - 3\left[\frac{2^{n+1} - (0.5)^{n+1}}{2 - 0.5}\right] u[n] \\ &= (0.5)^n u[n] \end{aligned}$$

If the composite cascade system were to be enclosed in a black box with only the input and the output terminals accessible, any measurement from these external terminals would show that the impulse response of the system is $(0.5)^n u[n]$, without any hint of the unstable system sheltered inside the composite system.

The composite system is BIBO stable because its impulse response $(0.5)^n u[n]$ is absolutely summable. However, the system S_2 is asymptotically unstable because its characteristic root, 2, lies outside the unit circle. This system will eventually burn out (or saturate) because of the unbounded characteristic response generated by intended or unintended initial conditions, no matter how small.

The system is asymptotically unstable, though BIBO stable. This example shows that BIBO stability does not necessarily ensure asymptotic stability when a system is either uncontrollable, or unobservable, or both. The internal and the external descriptions of a system are equivalent only when the system is controllable and observable. In such a case, BIBO stability means the system is asymptotically stable, and vice versa.

Fortunately, uncontrollable or unobservable systems are not common in practice. Henceforth, in determining system stability, we shall assume that unless otherwise mentioned, the internal and the external descriptions of the system are equivalent, implying that the system is controllable and observable.

EXAMPLE 3.22

Determine the internal and external stability of systems specified by the following equations. In each case plot the characteristic roots in the complex plane.

a. $y[n+2] + 2.5y[n+1] + y[n] = x[n+1] - 2x[n]$
b. $y[n] - y[n-1] + 0.21y[n-2] = 2x[n-1] + 3x[n-2]$
c. $y[n+3] + 2y[n+2] + \frac{3}{2}y[n+1] + \frac{1}{2}y[n] = x[n+1]$
d. $(E^2 - E + 1)^2 y[n] = (3E + 1)x[n]$

a. The characteristic polynomial is

$$\gamma^2 + 2.5\gamma + 1 = (\gamma + 0.5)(\gamma + 2)$$

The characteristic roots are -0.5 and -2. Because $|-2| > 1$ (-2 lies outside the unit circle), the system is BIBO unstable and also asymptotically unstable (Fig. 3.24a).

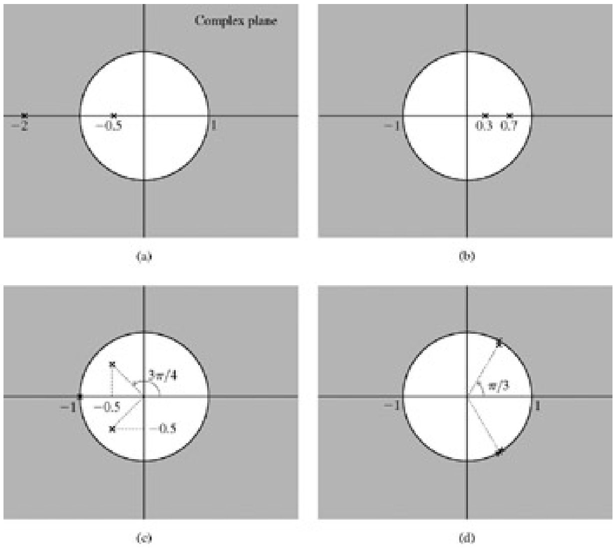

Figure 3.24: Location of characteristic roots for the systems.

b. The characteristic polynomial is

$$\gamma^2 - \gamma + 0.21 = (\gamma - 0.3)(\gamma - 0.7)$$

The characteristic roots are 0.3 and 0.7, both of which lie inside the unit circle. The system is BIBO stable and asymptotically stable (Fig. 3.24b).

c. The characteristic polynomial is

$$\gamma^3 + 2\gamma^2 + \tfrac{3}{2}\gamma + \tfrac{1}{2} = (\gamma + 1)(\gamma^2 + \gamma + \tfrac{1}{2}) = (\gamma + 1)(\gamma + 0.5 - j0.5)(\gamma + 0.5 + j0.5)$$

The characteristic roots are -1, $-0.5 \pm j0.5$ (Fig. 3.24c). One of the characteristic roots is on the unit circle and the remaining two roots are inside the unit circle. The system is BIBO unstable but marginally stable.

d. The characteristic polynomial is

$$(\gamma^2 - \gamma + 1)^2 = \left(\gamma - \tfrac{1}{2} - j\tfrac{\sqrt{3}}{2}\right)^2 \left(\gamma - \tfrac{1}{2} + j\tfrac{\sqrt{3}}{2}\right)^2$$

The characteristic roots are $(1/2) \pm j(\sqrt{3}/2) = 1e^{\pm j(\pi/3)}$ repeated twice, and they lie on the unit circle (Fig. 3.24d). The system is BIBO unstable and asymptotically unstable.

EXERCISE E3.22

Find and sketch the location in the complex plane of the characteristic roots of the system specified by the following equation: ?

$$(E + 1)(E^2 + 6E + 25)y[n] = 3Ex[n]$$

Determine the external and the internal stability of the system.

Answers

BIBO and asymptotically unstable

EXERCISE E3.23

Repeat Exercise E3.22 for

$$(E - 1)^2(E + 0.5)y[n] = (E^2 + 2E + 3)x[n]$$?

Answers

BIBO and asymptotically unstable

[†]If the development of discrete−time systems is parallel to that of continuous-time systems, we wonder why the parallel breaks down here. Why, for instance, are LHP and RHP not the regions demarcating stability and instability? The reason lies in the form of the characteristic modes. In continuous-time systems, we chose the form of characteristic mode as $e^{\lambda_i t}$. In discrete-time systems, for computational convenience, we choose the form to be γ^n_i. Had we chosen this form to be $e^{\lambda_i n}$ where $\gamma_i = e^{\lambda_i}$, then the LHP and RHP (for the location of λ_i) again would demarcate stability and instability. The reason is that if $\gamma = e^{\lambda}$ $|\gamma| = 1$ implies $|e^{\lambda}| = 1$, and therefore $\lambda = j\omega$. This shows that the unit circle in γ plane maps into the imaginary axis in the λ plane.

[†]This conclusion follows from the fact that (see Section B.7-4)

$$\sum_{n=-\infty}^{\infty} |\gamma_k^n| u[n] = \sum_{n=0}^{\infty} |\gamma_k|^n = \frac{1}{1-|\gamma_k|} \qquad |\gamma_k| < 1$$

Moreover, if $|\gamma| \geq 1$, the sum diverges and goes to ∞. These conclusions are valid also for the modes of the form $n^r \gamma^n{}_k$

3.11 INTUITIVE INSIGHTS INTO SYSTEM BEHAVIOR

The intuitive insights into the behavior of continuous-time systems and their qualitative proofs, discussed in Section 2.7, also apply to discrete-time systems. For this reason, we shall merely mention here without discussion some of the insights presented in Section 2.7.

The system's entire (zero-input and zero-state) behavior is strongly influenced by the characteristic roots (or modes) of the system. The system responds strongly to input signals similar to its characteristic modes and poorly to inputs very different from its characteristic modes. In fact, when the input is a characteristic mode of the system, the response goes to infinity, provided the mode is a nondecaying signal. This is the resonance phenomenon. The width of an impulse response $h[n]$ indicates the response time (time required to respond fully to an input) of the system. It is the time constant of the system.[†] Discrete-time pulses are generally dispersed when passed through a discrete-time system. The amount of dispersion (or spreading out) is equal to the system time constant (or width of $h[n]$). The system time constant also determines the rate at which the system can transmit information. Smaller time constant corresponds to higher rate of information transmission, and vice versa.

[†]This part of the discussion applies to systems with impulse response $h[n]$ that is a mostly positive (or mostly negative) pulse.

3.12 APPENDIX 3.1: IMPULSE RESPONSE FOR A SPECIAL CASE

When $a_N = 0$, $A_0 = b_N/a_N$ becomes indeterminate, and the procedure needs to be modified slightly. When $a_N = 0$, $Q[E]$ can be expressed as $E\hat{Q}[E]$ and Eq. (3.43) can be expressed as

$$E\hat{Q}[E]h[n] = P[E]\delta[n] = P[E]\{E\delta[n-1]\} = EP[E]\delta[n-1]$$

Hence

$$\hat{Q}[E]h[n] = P[E]\delta[n-1]$$

In this case the input vanishes not for $n \geq 1$, but for $n \geq 2$. Therefore, the response consists not only of the zero-input term and an impulse $A_0\delta[n]$ (at $n = 0$), but also of an impulse $A_1\,\delta[n-1]$ (at $n = 1$). Therefore

$$h[n] = A_0\delta[n] + A_1\delta[n-1] + y_c[n]u[n]$$

We can determine the unknowns A_0, A_1, and the $N-1$ coefficients in $y_c[n]$ from the $N+1$ number of initial values $h[0], h[1],..., h[N]$, determined as usual from the iterative solution of the equation $Q[E]h[n] = P[E]\delta[n]$.[‡] Similarly, if $a_N = a_{N-1} = 0$, we need to use the form $h[n] = A_0\delta[n] + A_1\delta[n-1] + A_2\delta[n-2] + y_c[n]u[n]$. The $N+1$ unknown constants are determined from the $N+1$ values $h[0], h[1],..., h[N]$, determined iteratively, and so on.

[‡] $\hat{Q}[\gamma]$ is now an $(N-1)$-order polynomial. Hence there are only $N-1$ unknowns in $y_c[n]$.

3.13 SUMMARY

This chapter discusses time-domain analysis of LTID (linear, time-invariant, discrete-time) systems. The analysis is parallel to that of LTIC systems, with some minor differences. Discrete-time systems are described by difference equations. For an Nth-order system, N auxiliary conditions must be specified for a unique solution. Characteristic modes are discrete-time exponentials of the form γ^n corresponding to an unrepeated root γ, and the modes are of the form $n^i\,\gamma^n$ corresponding to a repeated root γ.

The unit impulse function $\delta[n]$ is a sequence of a single number of unit value at $n = 0$. The unit impulse response $h[n]$ of a discrete-time system is a linear combination of its characteristic modes.[†]

The zero-state response (response due to external input) of a linear system is obtained by breaking the input into impulse components and then adding the system responses to all the impulse components. The sum of the system responses to the impulse components is in the form of a sum, known as the convolution sum, whose structure and properties are similar to the convolution integral. The system response is obtained as the convolution sum of the input $x[n]$ with the system's impulse response $h[n]$. Therefore, the knowledge of the system's impulse response allows us to determine the system response to any arbitrary input.

LTID systems have a very special relationship to the everlasting exponential signal z^n because the response of an LTID system to such an input signal is the same signal within a multiplicative constant. The response of an LTID system to the everlasting exponential input z^n is $H[z]z^n$, where $H[z]$ is the transfer function of the system.

Difference equations of LTID systems can also be solved by the classical method, in which the response is obtained as a sum of the natural and the forced components. These are not the same as the zero-input and zero-state components, although they satisfy the same equations, respectively. Although simple, this method is applicable to a restricted class of input signals, and the system response cannot be expressed as an explicit function of the input. These limitations diminish its value considerably in the theoretical study of systems.

The external stability criterion, the bounded-input/bounded-output (BIBO) stability criterion, states that a system is stable if and only if every bounded input produces a bounded output. Otherwise the system is unstable.

The internal stability criterion can be stated in terms of the location of characteristic roots of the system as follows:

1. An LTID system is asymptotically stable if and only if all the characteristic roots are inside the unit circle. The roots may be repeated or unrepeated.
2. An LTID system is unstable if and only if either one or both of the following conditions exist: (i) at least one root is outside the unit circle; (ii) there are repeated roots on the unit circle.
3. An LTID system is marginally stable if and only if there are no roots outside the unit circle and some unrepeated roots on the unit circle.

An asymptotically stable system is always BIBO stable. The converse is not necessarily true.

[†]There is a possibility of an impulse $\delta[n]$ in addition to characteristic modes.

MATLAB SESSION 3: DISCRETE-TIME SIGNALS AND SYSTEMS

MATLAB is naturally and ideally suited to discrete-time signals and systems. Many special functions are available for discrete-time data operations, including the `stem`, `filter`, and `conv` commands. In this session, we investigate and apply these and other commands.

M3.1 Discrete-Time Functions and Stem Plots

Consider the discrete-time function $f[n] = e^{-n/5} \cos(\pi n/5)u[n]$. In MATLAB, there are many ways to represent $f[n]$ including M-files or, for particular n, explicit command line evaluation. In this example, however, we use an inline object

```
>> f = inline('exp(-n/5).*cos(pi*n/5).*(n>=0)','n');
```

A true discrete-time function is undefined (or zero) for noninteger n. Although inline object `f` is intended as a discrete-time function, its present construction does not restrict n to be integer, and it can therefore be misused. For example, MATLAB dutifully returns 0.8606 to `f (0.5)` when a NaN (not-a-number) or zero is more appropriate. The user is responsible for appropriate function use.

Next, consider plotting the discrete-time function $f[n]$ over $(-10 \le n \le 10)$. The `stem` command simplifies this task.

```
>> n = (-10:10)';
>> stem(n, f(n),'k');
>> xlabel ('n'); ylabel('f[n]');
```

Here, `stem` operates much like the `plot` command: dependent variable `f (n)` is plotted against independent variable `n` with black lines. The `stem` command emphasizes the discrete-time nature of the data, as Fig. M3.1 illustrates.

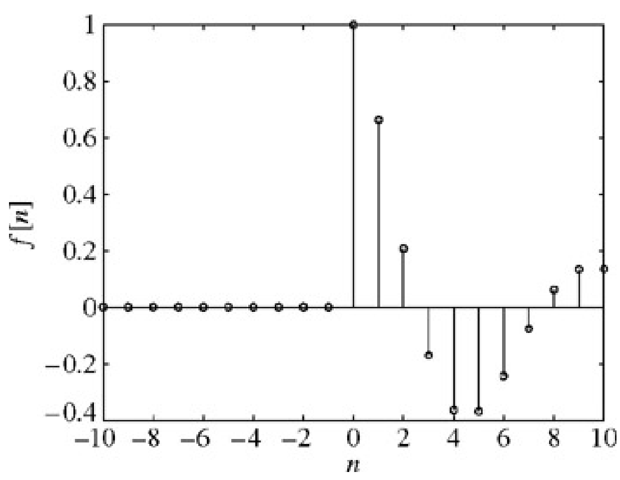

Figure M3.1: $f[n]$ over $(-10 \le n \le 10)$.

For discrete-time functions, the operations of shifting, inversion, and scaling can have surprising results. Compare $f[-2n]$ with $f[-2n + 1]$. Contrary to the continuous case, the second is not a shifted version of first. We can use separate subplots, each over $(-10 \le n \le 10)$, to help illustrate this fact. Notice that unlike the `plot` command, the `stem` command cannot simultaneously plot multiple functions on a single axis; overlapping stem lines would make such plots difficult to read anyway.

```
>> subplot (2,1,1); stem(n, f(-2*n), 'k'); ylabel('f[-2n]');
>> subplot(2,1,2); stem(n, f(-2*n+1),'k'); ylabel('f[-2n+1]'); xlabel('n');
```

The results are shown in Fig. M3.2. Interestingly, the original function $f[n]$ can be recovered by interleaving samples of $f[-2n]$ and $f[-2n + 1]$ and then time-reflecting the result.

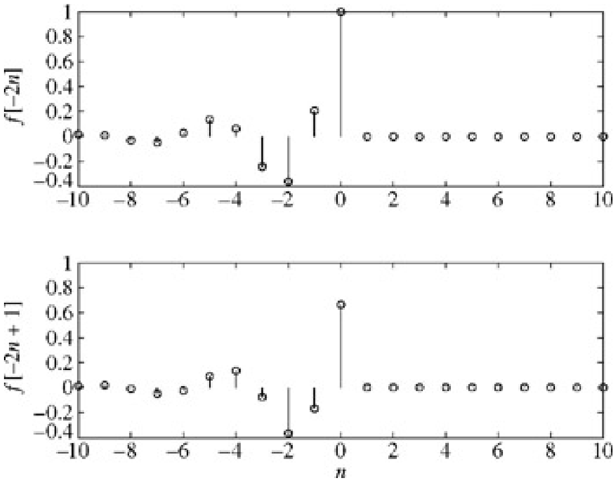

Figure M3.2: $f[-2n]$ and $f[-2n + 1]$ over $(-10 \le n \le 10)$.

Care must always be taken to ensure that MATLAB performs the desired computations. Our inline function `f` is a case in point: although it correctly downsamples, it does not properly upsample (see Prob. 3.M-1). MATLAB does what it is told, but it is not always told how to do everything correctly!

M3.2 System Responses Through Filtering

MATLAB's `filter` command provides an efficient way to evaluate the system response of a constant coefficient linear difference equation represented in delay form as

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{N} b_k x[n-k] \qquad \text{(M3.1)}$$

In the simplest form, `filter` requires three input arguments: a length$-(N + 1)$ vector of feedforward coefficients $[b_0, b_1, \ldots, b_N]$, a length-$(N + 1)$ vector of feedback coefficients $[a_0, a_1, \ldots, a_N]$, and an input vector.[†] Since no initial conditions are specified, the output corresponds to the system's zero-state response.

To serve as an example, consider a system described by $y[n] - y[n-1] + y[n-2] = x[n]$. When $x[n] = \delta[n]$, the zero-state response is equal to the impulse response $h[n]$, which we compute over $(0 \le n \le 30)$.

```
>> b = [1 0 0]; a = [1 -1 1];
>> n = (0:30)'; delta = inline('n==0','n');
>> h = filter(b,a,delta(n));
>> stem(n,h,'k'); axis([-.5 30.5 -1.1 1.1]);
>> xlabel ('n'); ylabel('h[n]');
```

As shown in Fig. M3.3, $h[n]$ appears to be $(N_0 = 6)-$periodic for $n \ge 0$. Since periodic signals are not absolutely summable, $\sum_{n=-\infty}^{\infty} |h[n]|$ is not finite and the system is not BIBO stable. Furthermore, the sinusoidal input $x[n] = \cos(2\pi n/6)u[n]$, which is $(N_0 = 6)-$periodic for $n \ge 0$, should generate a resonant zero-state response.

```
>> x = inline('cos(2*pi*n/6).*(n>=0)','n');
>> y = filter(b,a,x(n));
>> stem(n,y,'k'); xlabel('n'); ylabel('y[n]');
```

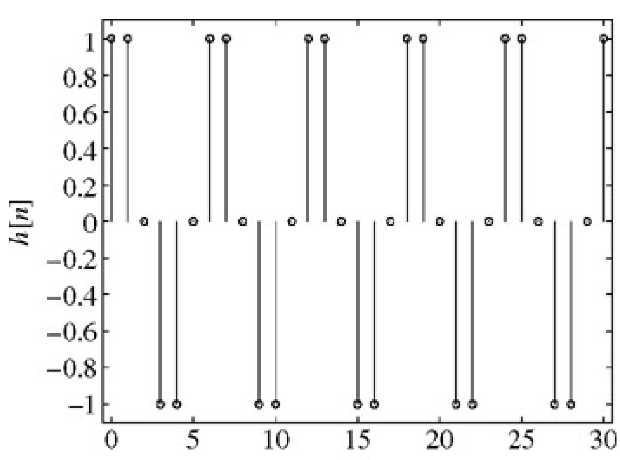

Figure M3.3: $h[n]$ for $y[n] - y[n-1] + y[n-2] = x[n]$.

The response's linear envelope, shown in Fig. M3.4, confirms a resonant response. The characteristic equation of the system is $\gamma^2 - \gamma + 1$, which has roots $\gamma = e^{\pm j\pi/3}$. Since the input $x[n] = \cos(2\pi n/6)u[n] = (1/2)(e^{j\pi n/3} + e^{-j\pi n/3})u[n]$ coincides with the characteristic roots, a resonant response is guaranteed.

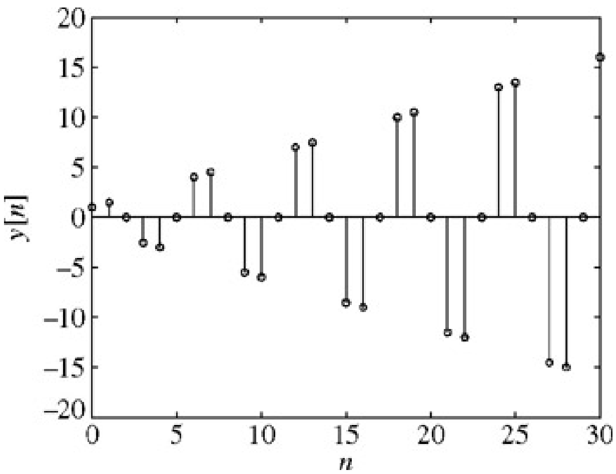

Figure M3.4: Resonant zero-state response $y[n]$ for $x[n] = \cos(2\pi n/6)u[n]$.

By adding initial conditions, the `filter` command can also compute a system's zero-input response and total response. Continuing the preceding example, consider finding the zero-input response for $y[-1] = 1$ and $y[-2] = 2$ over $(0 \le n \le 30)$.

```
>> z_i = filtic(b,a,[1 2]);
>> y_0 = filter(b,a,zeros(size(n)),z_i);
>> stem(n,y_0,'k'); xlabel('n'); ylabel('y_{0} [n]');
>> axis([-0.5 30.5 -2.1 2.1]);
```

There are many physical ways to implement a particular equation. MATLAB implements Eq. (M3.1) by using the popular direct form II transposed structure.[†] Consequently, initial conditions must be compatible with this implementation structure. The signal processing toolbox function `filtic` converts the traditional $y[-1], y[-2], \ldots, y[-N]$ initial conditions for use with the `filter` command. An input of zero is created with the `zeros` command. The dimensions of this zero input are made to match the vector `n` by using the `size` command. Finally, `_{ }` forces subscript text in the graphics window, and `^{ }` forces superscript text. The results are shown in Fig. M3.5.

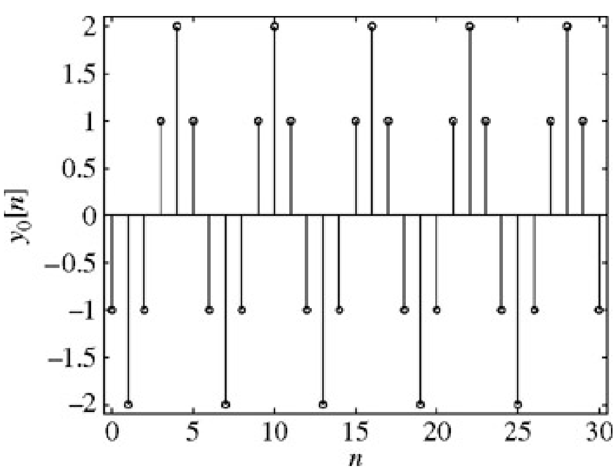

Figure M3.5: Zero-input response $y_0[n]$ for $y[-1] = 1$ and $y[-2] = 2$.

Given $y[-1] = 1$ and $y[-2] = 2$ and an input $x[n] = \cos(2\pi n/6)u[n]$, the total response is easy to obtain with the `filter` command.

```
>> y_total = filter(b,a,x(n),z_i);
```

Summing the zero-state and zero-input response gives the same result. Computing the total absolute error provides a check.

```
>> sum(abs(y_total-(y + y_0)))
ans = 1.8430e-014
```

Within computer round-off, both methods return the same sequence.

M3.3 A Custom Filter Function

The `filtic` command is only available if the signal processing toolbox is installed. To accommodate installations without the signal processing toolbox and to help develop your MATLAB skills, consider writing a function similar in syntax to `filter` that directly uses the ICS $y[-1], y[-2], \ldots, y[-N]$. Normalizing $a_0 = 1$ and solving Eq. (M3.1) for $y[n]$ yields

$$y[n] = \sum_{k=0}^{N} b_k x[n-k] - \sum_{k=1}^{N} a_k y[n-k]$$

This recursive form provides a good basis for our custom filter function.

```
function [y] = MS3P1(b, a, x, yi);
% MS3P1.m : MATLAB Session 3, Program 1
% Function M-file filters data x to create y
% INPUTS:   b = vector of feedforward coefficients
%           a = vector of veedback coefficients
%           x = input data vector
%           yi = vector of initial conditions [y[-1], y[-2], ...]
% OUTPUTS:  y = vector of filtered output data

yi = flipud(yi(:)); % Properly format IC's.
y = [yi;zeros(length(x),1)]; % Preinitialize y, beginning with IC's
x = [zeros(length(yi),1);x(:)]; % Append x with zeros to match size of y.
b = b/a(1);a = a/a(1); % Normalize coefficients.
for n = length(yi)+1:length(y),
for nb = 0:length(b)-1,
y(n) = y(n) + b(nb+1)*x(n-na); % Feedforward terms.
end
for na = 1:length(a)-1
y(n) = y(n) - a(na+1)*y(n-na); % Feedback terms.
end
end
y = y(length)(yi)+1:end); % Strip of IC's for final output.
```

Most instructions in `MS3P1` have been discussed; now we turn to the `flipud` instruction. The flip up-down command `flipud` reverses the order of elements in a column vector. Although not used here, the flip left-right command `fliplr` reverses the order of elements in a row vector. Note that typing `help` ***`filename`*** displays the first contiguous set of comment lines in an M-file. Thus, it is good programming practice to document M-files, as in `MS3P1`, with an initial block of clear comment lines.

As an exercise, the reader should verify that `MS3P1` correctly computes the impulse response $h[n]$, the zero-state response $y[n]$, the zero-input response $y_0[n]$, and the total response $y[n] + y_0[n]$.

M3.4 Discrete-Time Convolution

Convolution of two finite-duration discrete-time signals is accomplished by using the `conv` command. For example, the discrete-time convolution of two length-4 rectangular pulses, $g[n] = (u[n] - u[n-4]) * (u[n] - u[n-4])$, is a length-$(4 + 4 - 1 = 7)$ triangle. Representing $u[n] - u[n-4]$ by the vector [1, 1, 1, 1], the convolution is computed by:

```
>> conv ( [1 1 1 1], [1 1 1 1])
ans = 1      2      3      4      3      2      1
```

Notice that $(u[n+4] - u[n])*(u[n] - u[n-4])$ is also computed by `conv ([1 1 1 1], [1 1 1 1])` and obviously yields the same result. The difference between these two cases is the regions of support: $(0 \le n \le 6)$ for the first and $(-4 \le n \le 2)$ for the second. Although the `conv` command does not compute the region of support, it is relatively easy to obtain. If vector w begins at $n = n_w$ and vector v begins at $n = n_v$, then `conv (w, v)` begins at $n = n_w + n_v$.

In general, the `conv` command cannot properly convolve infinite-duration signals. This is not too surprising, since computers themselves cannot store an infinite-duration signal. For special cases, however, `conv` can correctly compute a portion of such convolution problems. Consider the common case of convolving two causal signals. By passing the first N samples of each, `conv` returns a length-$(2N - 1)$ sequence. The first N samples of this sequence are valid; the remaining $N - 1$ samples are not.

To illustrate this point, reconsider the zero-state response $y[n]$ over $(0 \le n \le 30)$ for system $y[n] - y[n-1] + y[n-2] = x[n]$ given input $x[n] = \cos(2\pi n/6)u[n]$. The results obtained by using a filtering approach are shown in Fig. M3.4.

The response can also be computed using convolution according to $y[n] = h[n]*x[n]$. The impulse response of this system is[†]

$$h[n] = \left\{\cos(\pi n/3) + \frac{1}{\sqrt{3}}\sin(\pi n/3)\right\} u[n]$$

Both $h[n]$ and $x[n]$ are causal and have infinite duration, so `conv` can be used to obtain a portion of the convolution.

```
>> h = inline('(cos(pi*n/3)+sin(pi*n/3)/sqrt(3)).*(n>=0)','n');
>> y = conv(h(n), x(n));
>> stem([0:60],y,'k'); xlabel('n'); ylabel('y[n]');
```

The `conv` output is fully displayed in Fig. M3.6. As expected, the results are correct over $(0 \le n \le 30)$. The remaining values are clearly incorrect; the output envelope should continue to grow, not decay. Normally, these incorrect values are not displayed.

```
>> stem(n,y(1:31),'k'); xlabel('n'); ylabel('y[n]');
```

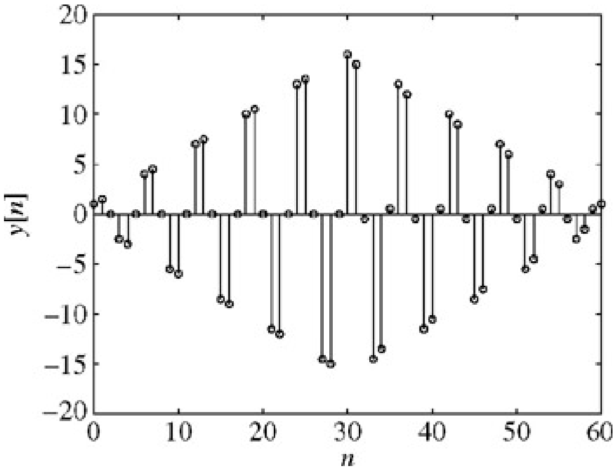

Figure M3.6: $y[n]$ for $x[n] = \cos(2\pi n/6)u[n]$ computed with conv.

The resulting plot is identical to Fig. M3.4.

[†]It is important to pay close attention to the inevitable notational differences found throughout engineering documents. In MATLAB help documents, coefficient subscripts begin at 1 rather than 0 to better conform with MATLAB indexing conventions. That is, MATLAB labels a_0 as `a (1)`, b_0 as `b (1)`, and so forth.

[†]Implementation structures, such as direct form II transposed, are discussed in Chapter 4.

[†]Techniques to analytically determine $h[n]$ are presented in Chapter 5.

PROBLEMS

3.1.1 Find the energy of the signals depicted in Figs. P3.1-1.

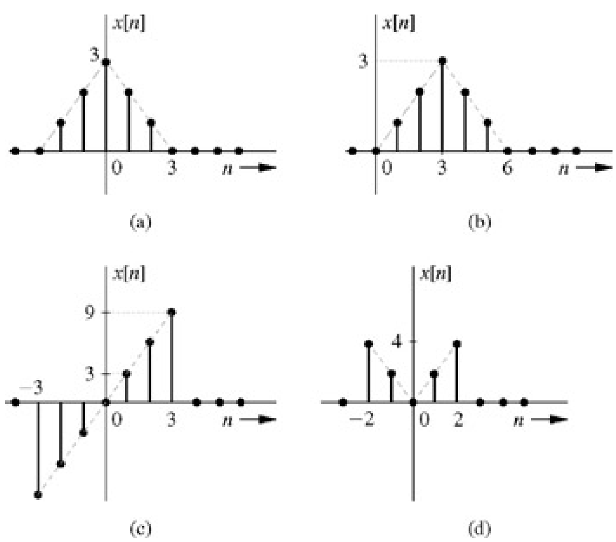

Figure P3.1-1

3.1.2 Find the power of the signals illustrated in Figs. P3.1-2.

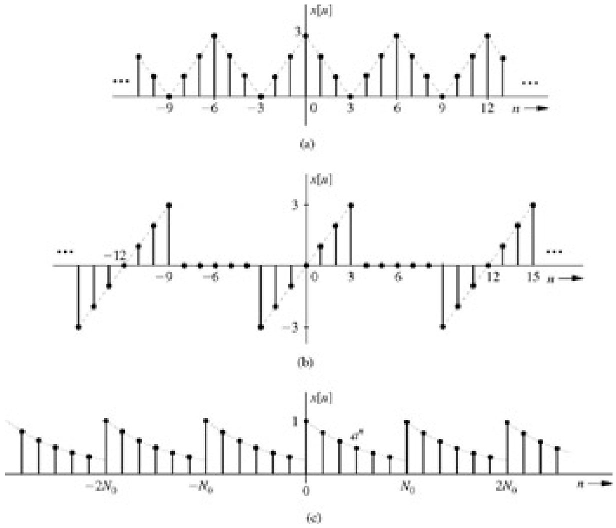

Figure P3.1-2

3.1.3 Show that the power of a signal $De^{j(2\pi/N_0)n}$ is $|D|^2$. Hence, show that the power of a signal

$$x[n] = \sum_{r=0}^{N_0-1} \mathcal{D}_r e^{jr(2\pi/N_0)n} \text{ is } P_x = \sum_{r=0}^{N_0-1} |\mathcal{D}_r|^2$$

Use the fact that

$$\sum_{k=0}^{N_0-1} e^{j(r-m)2\pi k/N_0} = \begin{cases} N_0 & r = m \\ 0 & \text{otherwise} \end{cases}$$

3.1.4

a. Determine even and odd components of the signal $x[n] = (0.8)^n u[n]$.

b. Show that the energy of $x[n]$ is the sum of energies of its odd and even components found in part (a).

c. Generalize the result in part (b) for any finite energy signal.

3.1.5

a. If $x_e[n]$ and $x_0[n]$ are the even and the odd components of a real signal $x[n]$, then show that $E_{x_e} = E_{x_0} = 0.5E_x$.

b. Show that the cross-energy of x_e and x_0 is zero, that is,

$$\sum_{n=-\infty}^{\infty} x_e[n]x_o[n] = 0$$

3.2.1 If the energy of a signal $x[n]$ is E_x, then find the energy of the following:

a. $x[-n]$

b. $x[n - m]$

c. $x[m - n]$

d. $Kx[n]$ (m integer and K constant)

3.2.2 If the power of a periodic signal $x[n]$ is P_x, find and comment on the powers and the rms values of the following:

a. $-x[n]$

b. $x[-n]$

c. $x[n - m]$ (m integer)

d. $cx[n]$

e. $x[m - n]$ (m integer)

3.2.3 For the signal shown in Fig. P3.1-1b, sketch the following signals:

a. $x[-n]$

b. $x[n + 6]$

c. $x[n - 6]$

d. $x[3n]$

e. $x\left[\frac{n}{3}\right]$

f. $x[3 - n]$

3.2.4 Repeat Prob. 3.2-3 for the signal depicted in Fig.P3.1-1c.

3.3.1 Sketch, and find the power of, the following signals:

a. $(1)^n$

b. $(-1)^n$

c. $u[n]$

d. $(-1)^n u[n]$

e. $\cos\left[\frac{\pi}{3}n + \frac{\pi}{6}\right]$

3.3.2 Show that

a. $\delta[n] + \delta[n - 1] = u[n] - u[n - 2]$

b. $2^{n-1} \sin\left(\frac{\pi n}{3}\right) u[n] = \frac{1}{2}\, 2^n \sin\left(\frac{\pi n}{3}\right) u[n-1]$

c. $n(n - 1)\gamma^n u[n] = n(n - 1)\gamma^n u[n - 2]$

d. $(u[n] + (-1)^n u[n]) \sin\left(\frac{\pi n}{2}\right) = 0$

e. $(u[n]+(-1)^{n+1}u[n]) \cos\left(\frac{\pi n}{2}\right) = 0$ for all n

3.3.3 Sketch the following signals:

a. $u[n - 2] - u[n - 6]$

b. $n\{u[n] - u[n - 7]\}$

c. $(n - 2)\{u[n - 2] - u[n - 6]\}$

d. $(-n + 8)\{u[n - 6] - u[n - 9]\}$

e. $(n - 2)\{u[n - 2] - u[n - 6]\} + (-n + 8)\{u[n - 6] - u[n - 9]\}$

3.3.4 Describe each of the signals in Fig. P3.1-1 by a single expression valid for all n.

3.3.5 The following signals are in the form $e^{\lambda n}$. Express them in the form γ^n:

a. $e^{-0.5n}$

b. $e^{0.5n}$

c. $e^{-j\pi n}$

d. $e^{j\pi n}$

In each case show the locations of λ and γ in the complex plane. Verify that an exponential is growing if γ lies outside the unit circle (or if λ lies in the RHP), is decaying if γ lies within the unit circle (or if λ lies in the LHP), and has a constant amplitude if γ lies on the unit circle (or if λ lies on the imaginary axis).

3.3.6 Express the following signals, which are in the form $e^{\lambda n}$, in the form ^{n}:

a. $e^{-(1+j\pi)n}$

b. $e^{-(1-j\pi)n}$

c. $e^{(1+j\pi)n}$

d. $e^{(1-j\pi)n}$

e. $e^{-[1+j(\pi/3)]n}$

f. $e^{[1-j(\pi/3)]n}$

3.3.7 The concepts of even and odd functions for discrete-time signals are identical to those of the continuous-time signals discussed in Section 1.5. Using these concepts, find and sketch the odd and the even components of the following:

a. $u[n]$

b. $nu[n]$

c. $\sin\left(\frac{\pi n}{4}\right)$

d. $\cos\left(\frac{\pi n}{4}\right)$

3.4.1 A cash register output $y[n]$ represents the total cost of n items rung up by a cashier. The input $x[n]$ is the cost of the nth item.

a. Write the difference equation relating $y[n]$ to $x[n]$.

b. Realize this system using a time-delay element.

3.4.2 Let $p[n]$ be the population of a certain country at the beginning of the nth year. The birth and death rates of the population during any year are 3.3 and 1.3%, respectively. If $i[n]$ is the total number of immigrants entering the country during the nth year, write the difference equation relating $p[n + 1]$, $p[n]$, and $i[n]$. Assume that the immigrants enter the country throughout the year at a uniform rate.

3.4.3 A moving average is used to detect a trend of a rapidly fluctuating variable such as the stock market average. A variable may fluctuate (up and down) daily, masking its long-term (secular) trend. We can discern the long-term trend by smoothing or averaging the past N values of the variable. For the stock market average, we may consider a 5-day moving average $y[n]$ to be the mean of the past 5 days' market closing values $x[n]$, $x[n - 1]$,..., $x[n - 4]$.

a. Write the difference equation relating $y[n]$ to the input $x[n]$.

b. Use time-delay elements to realize the 5-day moving-average filter.

3.4.4 The digital integrator in Example 3.7 is specified by

$$y[n] - y[n-1] = Tx[n]$$

If an input $u[n]$ is applied to such an integrator, show that the output is $(n + 1)Tu[n]$, which approaches the desired ramp $nTu[n]$ as $T \rightarrow 0$.

3.4.5 Approximate the following second-order differential equation with a difference equation.

$$\frac{d^2y}{dt^2} + a_1\frac{dy}{dt} + a_0y(t) = x(t)$$

3.4.6 The voltage at the nth node of a resistive ladder in Fig. P3.4-6 is $v[n]$

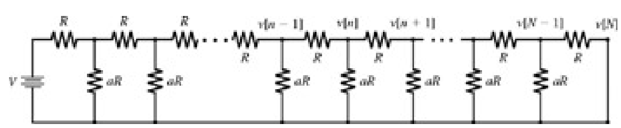

Figure P3.4-6

(n = 0, 1, 2,..., N). Show that $v[n]$ satisfies the second-order difference equation

$$v[n+2] - Av[n+1] + v[n] = 0 \quad A = 2 + \frac{1}{a}$$

[Hint: Consider the node equation at the nth node with voltage $v[n]$.]

3.4.7 Determine whether each of the following statements is true or false. If the statement is false, demonstrate by proof or example why the statement is false. If the statement is true, explain why.

a. A discrete-time signal with finite power cannot be an energy signal.

b. A discrete-time signal with infinite energy must be a power signal.

c. The system described by $y[n] = (n+1)x[n]$ is causal.

d. The system described by $y[n - 1] = x[n]$ is causal.

e. If an energy signal $x[n]$ has energy E, then the energy of $x[an]$ is $E/|a|$.

3.4.8 A linear, time-invariant system produces output $y_1[n]$ in response to input $x_1[n]$, as shown in Fig. P3.4-8. Determine and sketch the output $y_2[n]$ that results when input $x_2[n]$ is applied to the same system.

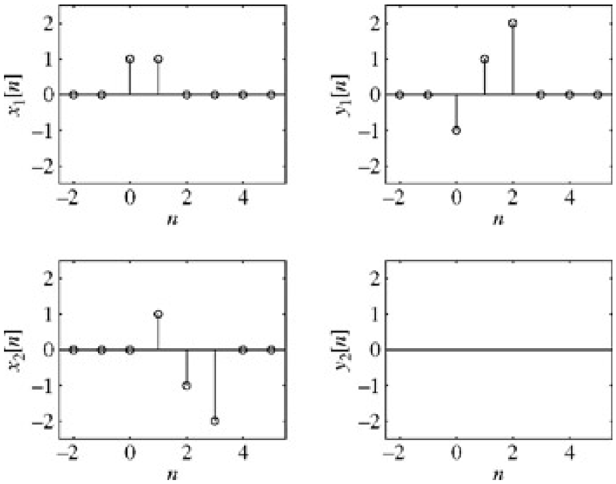

Figure P3.4-8: Input-output plots.

3.4.9 A system is described by

$$y[n] = \frac{1}{2} \sum_{k=-\infty}^{\infty} x[k](\delta[n-k] + \delta[n+k])$$

a. Explain what this system does.

b. Is the system BIBO stable? Justify your answer.

c. Is the system linear? Justify your answer.

d. Is the system memoryless? Justify your answer.

e. Is the system causal? Justify your answer.

f. Is the system time invariant? Justify your answer.

3.4.10 A discrete-time system is given by

$$y[n+1] = \frac{x[n]}{x[n+1]}$$

a. Is the system BIBO stable? Justify your answer.

b. Is the system memoryless? Justify your answer.

c. Is the system causal? Justify your answer.

3.4.11 Explain why the continuous-time system $y(t) = x(2t)$ is always invertible and yet the corresponding discrete-time system $y[n] = x[2n]$ is not invertible.

3.4.12 Consider the input-output relationships of two similar discrete-time systems:

$$y_1[n] = \sin\left(\frac{\pi}{2}n + 1\right) x[n]$$

and

$$y_2[n] = \sin\left(\frac{\pi}{2}(n+1)\right) x[n]$$

Explain why $x[n]$ can be recovered from $y_1[n]$ yet $x[n]$ cannot be recovered from $y_2[n]$.

3.4.13 Consider a system that multiplies a given input by a ramp function, $r[n]$. That is, $y[n] = x[n]r[n]$.

a. Is the system BIBO stable? Justify your answer.

b. Is the system linear? Justify your answer.

c. Is the system memoryless? Justify your answer.

d. Is the system causal? Justify your answer.

e. Is the system time invariant? Justify your answer.

3.4.14 A jet-powered car is filmed using a camera operating at 60 frames per second. Let variable n designate the film frame, where $n = 0$ corresponds to engine ignition (film before ignition is discarded). By analyzing each frame of the film, it is possible to determine the car position $x[n]$, measured in meters, from the original starting position $x[0] = 0$.

From physics, we know that velocity is the time derivative of position,

$$v(t) = \frac{d}{dt}x(t)$$

Furthermore, we know that acceleration is the time derivative of velocity,

$$a(t) = \frac{d}{dt}v(t)$$

We can estimate the car velocity from the film data by using a simple difference equation $v[n] = k(x[n] - x[n - 1])$

a. Determine the appropriate constant k to ensure $v[n]$ has units of meters per second.

b. Determine a standard-form constant coefficient difference equation that outputs an estimate of acceleration, $a[n]$, using an input of position, $x[n]$. Identify the advantages and shortcomings of estimating acceleration $a(t)$ with $a[n]$. What is the impulse response $h[n]$ for this system?

3.4.15 Do part (a) of Prob. 3.M-2.

3.5.1 Solve recursively (first three terms only):

a. $y[n + 1] - 0.5y[n] = 0$, with $y[-1] = 10$

b. $y[n + 1] + 2y[n] = x[n + 1]$, with $x[n] = e^{-n}u[n]$ and $y[-1] = 0$

3.5.2 Solve the following equation recursively (first three terms only):

$$y[n] - 0.6y[n-1] - 0.16y[n-2] = 0$$

with

$$y[-1] = -25,\ y[-2] = 0.$$

3.5.3 Solve recursively the second-order difference Eq. (3.10b) for sales estimate (first three terms only), assuming $y[-1] = y[-2] = 0$ and $x[n] = 100u[n]$.

3.5.4 Solve the following equation recursively (first three terms only):

$$y[n+2] + 3y[n+1] + 2y[n]$$
$$= x[n+2] + 3x[n+1] + 3x[n]$$

with $x[n] = (3)^n u[n]$, $y[-1] = 3$, and $y[-2] = 2$

3.5.5 Repeat Prob. 3.5−4 for

$$y[n] + 2y[n-1] + y[n-2] = 2x[n] - x[n-1]$$

with $x[n] = (3)^{-n}u[n]$, $y[-1] = 2$, and $y[-2] = 3$.

3.6.1 Solve

$$y[n+2] + 3y[n+1] + 2y[n] = 0$$

if $y[-1] = 0,\ y[-2] = 1.$

3.6.2 Solve

$$y[n+2] + 2y[n+1] + y[n] = 0$$

if $y[-1] = 1,\ y[-2] = 1.$

3.6.3 Solve

$$y[n+2] - 2y[n+1] + 2y[n] = 0$$

if $y[-1] = 1,\ y[-2] = 0.$

3.6.4 For the general Nth-order difference Eq. (3.17b), letting

$$a_1 = a_2 = \cdots = a_{N-1} = 0$$

results in a general causal Nth-order LTI *nonrecursive* difference equation

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots$$
$$+ b_{N-} x[n - N + 1] + b_N x[n - N]$$

Show that the characteristic roots for this system are zero, and hence, the zero-input response is zero. Consequently, the total response consists of the zero-state component only.

3.6.5 Leonardo Pisano Fibonacci, a famous thirteenth-century mathematician, generated the sequence of integers {0, 1, 1, 2, 3, 5, 8, 13, 21, 34,....} while addressing, oddly enough, a problem involving rabbit reproduction. An element of the Fibonacci sequence is the sum of the previous two.

a. Find the constant-coefficient difference equation whose zero-input response $f[n]$ with auxiliary conditions $f[1] = 0$ and $f[2] = 1$ is a Fibonacci sequence. Given $f[n]$ is the system output, what is the system input?

b. What are the characteristic roots of this system? Is the system stable?

c. Designating 0 and 1 as the first and second Fibonacci numbers, determine the fiftieth Fibonacci number. Determine the thousandth Fibonacci number.

3.6.6 Find $v[n]$, the voltage at the nth node of the resistive ladder depicted in Fig. P3.4-6, if V = 100 volts and a = 2. [Hint 1: Consider the node equation at the nth node with voltage $v[n]$. Hint 2: See Prob. 3.4-6 for the equation for $v[n]$. The auxiliary conditions are $v[0]$ = 100 and $v[N]$ = 0.]

3.6.7 Consider the discrete-time system $y[n] + y[n-1] + 0.25y[n-2] = \sqrt{x[n-8]}$. Find the zero input response, $y_0[n]$, if $y_0[-1] = 1$ and $y_0[1] = 1$

3.7.1 Find the unit impulse response $h[n]$ of systems specified by the following equations:

a. $y[n+1] + 2y[n] = x[n]$

b. $y[n] + 2y[n-1] = x[n]$

3.7.2 Repeat Prob. 3.7−1 for

$$(E^2 - 6E + 9)y[n] = Ex[n]$$

3.7.3 Repeat Prob. 3.7−1 for

$$y[n] - 6y[n-1] + 25y[n-2] =$$

$$2x[n] - 4x[n-1]$$

3.7.4

a. For the general Nth-order difference Eq. (3.17), letting

$$a_0 = a_1 = a_2 = \cdots = a_{N-1} = 0$$

results in a general causal Nth-order LTI *nonrecursive* difference equation

$$y[n] = b_0x[n] + b_1x[n-1] + \cdots$$

$$+ b_{N-1}x[n-N+1] + b_Nx[n-N]$$

Find the impulse response $h[n]$ for this system. [Hint: The characteristic equation for this case is $\gamma^n = 0$. Hence, all the characteristic roots are zero. In this case, $y_c[n] = 0$, and the approach in Section 3.7 does not work. Use a direct method to find $h[n]$ by realizing that $h[n]$ is the response to unit impulse input.]

b. Find the impulse response of a nonrecursive LTID system described by the equation

$$y[n] = 3x[n] - 5x[n-1] - 2x[n-3]$$

Observe that the impulse response has only a finite (N) number of nonzero elements. For this reason, such systems are called *finite-impulse response* (FIR) systems. For a general recursive case [Eq. (3.24)], the impulse response has an infinite number of nonzero elements, and such systems are called *infinite-impulse response* (IIR) systems.

3.8.1 Find the (zero-state) response $y[n]$ of an LTID system whose unit impulse response is

$$h[n] = (-2)^n u[n-1]$$

and the input is $x[n] = e^{-n}u[n+1]$. Find your answer by computing the convolution sum and also by using the convolution table (Table 3.1).

3.8.2 Find the (zero-state) response $y[n]$ of an LTID system if the input is $x[n] = 3^{n-1}u[n+2]$ and

$$h[n] = \tfrac{1}{2}[\delta[n-2] - (-2)^{n+1}]u[n-3]$$

3.8.3 Find the (zero-state) response $y[n]$ of an LTID system if the input $x[n] = (3)^{n+2}u[n+1]$, and

$$h[n] = [(2)^{n-2} + 3(-5)^{n+2}]u[n-1]$$

3.8.4 Find the (zero-state) response $y[n]$ of an LTID system if the input $x[n] = (3)^{-n+2}u[n+3]$, and

$$h[n] = 3(n-2)(2)^{n-3}u[n-4]$$

3.8.5 Find the (zero-state) response $y[n]$ of an LTID system if its input $x[n] = (2)^n u[n-1]$, and

$$h[n] = (3)^n \cos\left(\frac{\pi}{3}n - 0.5\right)u[n]$$

Find your answer using only Table 3.1, the convolution table.

3.8.6 Derive the results in entries 1, 2 and 3 in Table 3.1. [Hint: You may need to use the information in Section B.7-4.]

3.8.7 Derive the results in entries 4, 5, and 6 in Table 3.1.

3.8.8 Derive the results in entries 7 and 8 in Table 3.1. [Hint: You may need to use the information in Section B.7-4.]

3.8.9 Derive the results in entries 9 and 11 in Table 3.1. [Hint: You may need to use the information in Section B.7-4.]

3.8.10 Find the total response of a system specified by the equation

$$y[n+1] + 2y[n] = x[n+1]$$

if $y[-1] = 10$, and the input $x[n] = e^{-n}u[n]$.

3.8.11 Find an LTID system (zero-state) response if its impulse response $h[n] = (0.5)^n u[n]$, and the input $x[n]$ is

a. $2^n u[n]$

b. $2^{n-3} u[n]$

c. $2^n u[n-2]$

[Hint: You may need to use the shift property (3.61) of the convolution.]

3.8.12 For a system specified by equation

$$y[n] = x[n] - 2x[n-1]$$

Find the system response to input $x[n] = u[n]$. What is the order of the system? What type of system (recursive or nonrecursive) is this? Is the knowledge of initial condition(s) necessary to find the system response? Explain.

3.8.13

a. A discrete-time LTI system is shown in Fig. P3.8-13. Express the overall impulse response of the system, $h[n]$, in terms of $h_1[n]$, $h_2[n]$, $h_3[n]$, $h_4[n]$, and $h_5[n]$.

b. Two LTID systems in cascade have impulse response $h_1[n]$ and $h_2[n]$, respectively. Show that if $h_1[n] = (0.9)^n u[n] - 0.5(0.9)^{n-1} u[n-1]$ and $h_2[n] = (0.5)^n u[n] - 0.9(0.5)^{n-1} u[n-1]$, the cascade system is an identity system.

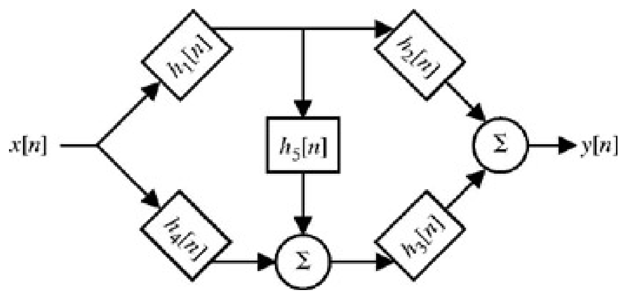

Figure P3.8-13

3.8.14

a. Show that for a causal system, Eq. (3.70b) can also be expressed as

$$g[n] = \sum_{k=0}^{n} h[n-k]$$

b. How would the expressions in part (a) change if the system is not causal?

3.8.15 In the savings account problem described in Example 3.4, a person deposits \$500 at the beginning of every month, starting at $n = 0$ with the exception at $n = 4$, when instead of depositing \$500, she withdraws \$1000. Find $y[n]$ if the interest rate is 1% per month ($r = 0.01$).

3.8.16 To pay off a loan of M dollars in N number of payments using a fixed monthly payment of P dollars, show that

$$P = \frac{rM}{1-(1+r)^{-N}}$$

where r is the interest rate per dollar per month. [Hint: This problem can be modeled by Eq. (3.9a) with the payments of P dollars starting at $n = 1$. The problem can be approached in two ways (1) Consider the loan as the initial condition $y_0[0] = -M$, and the input $x[n] = Pu[n-1]$. The loan balance is the sum of the zero-input component (due to the initial condition) and the zero-state component $h[n] * x[n]$. (2) Consider the loan as an input $-M$ at $n = 0$ along with the input due to payments. The loan balance is now exclusively a zero-state component $h[n] * x[n]$. Because the loan is paid off in N payments, set $y[N] = 0$.]

3.8.17 A person receives an automobile loan of \$10,000 from a bank at the interest rate of 1.5% per month. His monthly payment is \$500, with the first payment due one month after he receives the loan. Compute the number of payments required to pay off the loan. Note that the last payment may not be exactly \$500. [Hint: Follow the procedure in Prob. 3.8−16 to determine the balance $y[n]$. To determine N, the number of payments, set $y[N] = 0$. In general, N will not be an integer. The number of payments K is the largest integer $\le N$. The residual payment is $|y[K]|$.]

3.8.18 Using the sliding-tape algorithm, show that

a. $u[n] * u[n] = (n+1)u[n]$

b. $(u[n] - u[n-m]) * u[n] = (n+1)u[n] - (n-m+1)u[n-m]$

3.8.19 Using the sliding-tape algorithm, find $x[n] * g[n]$ for the signals shown in Fig. P3.8-19.

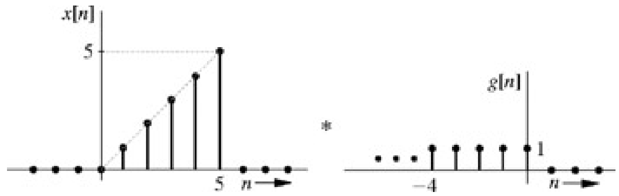

Figure P3.8-19

3.8.20 Repeat Prob. 3.8-19 for the signals shown in Fig. P3.8-20.

Figure P3.8-20

3.8.21 Repeat Prob. 3.8-19 for the signals depicted in Fig. P3.8-21.

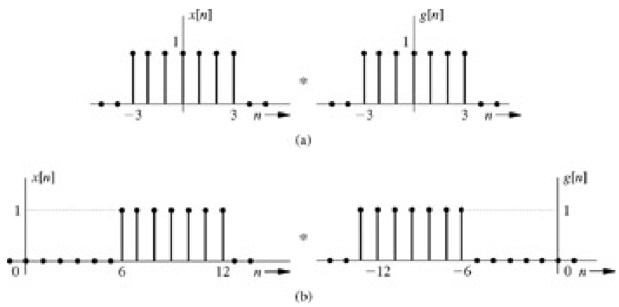

Figure P3.8-21

3.8.22 The convolution sum in Eq. (3.63) can be expressed in a matrix form as

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[n] \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} h[0] & 0 & 0 & \dots & 0 \\ h[1] & h[0] & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ h[n] & h[n-1] & \dots & \dots & h[0] \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[n] \end{bmatrix}}_{\mathbf{f}}$$

or

$$\mathbf{y} = \mathbf{Hf}$$

and

$$\mathbf{f} = \mathbf{H}^{-1}\mathbf{y}$$

Knowing $h[n]$ and the output $y[n]$, we can determine the input $x[n]$. This operation is the reverse of the convolution and is known as the *deconvolution*. Moreover, knowing $x[n]$ and $y[n]$, we can determine $h[n]$. This can be done by expressing the foregoing matrix equation as $n + 1$ simultaneous equations in terms of $n + 1$ unknowns $h[0]$, $h[1]$,..., $h[n]$. These equations can readily be solved iteratively. Thus, we can synthesize a system that yields a certain output $y[n]$ for a given input $x[n]$.

a. Design a system (i.e., determine $h[n]$) that will yield the output sequence (8, 12, 14, 15, 15.5, 15.75,...) for the input sequence (1, 1, 1, 1, 1, 1,...).

b. For a system with the impulse response sequence (1, 2, 4,...), the output sequence was (1, 7/3, 43/9,...). Determine the input sequence.

3.8.23 A second-order LTID system has zero-input response

$$y_0[n] = \left[3, 2\tfrac{1}{3}, 2\tfrac{1}{9}, 2\tfrac{1}{27}, \dots\right]$$

$$= \sum_{k=0}^{\infty} \left\{2 + \left(\tfrac{1}{3}\right)^k\right\} \delta[n-k]$$

a. Determine the characteristic equation of this system, $a_0\gamma^2 + a_1\gamma + a_2 = 0$.

b. Find a bounded, causal input with infinite duration that would cause a strong response from this system. Justify your choice.

c. Find a bounded, causal input with infinite duration that would cause a weak response from this system. Justify your choice.

3.8.24 An LTID filter has an impulse response function given by $h_1[n] = \delta[n+2] - \delta[n-2]$. A second LTID system has an impulse response function given by $h_2[n] = n(u[n+4] - u[n-4])$.

a. Carefully sketch the functions $h_1[n]$ and $h_2[n]$ over $(-10 \le n \le 10)$.

b. Assume that the two systems are connected in parallel as shown in Fig. P3.8-24. Determine the impulse response $h_p[n]$ for the parallel system in terms of $h_1[n]$ and $h_2[n]$. Sketch $h_p[n]$ over $(-10 \le n \le 10)$.

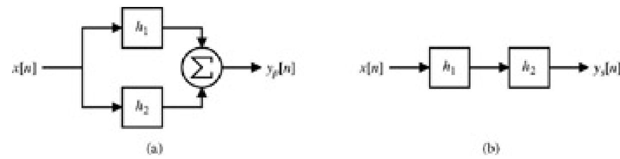

Figure P3.8-24: Parallel and series system connections.

c. Assume that the two systems are connected in cascade as shown in Fig. P3.8-24. Determine the impulse response $h_s[n]$ for the cascade system in terms of $h_1[n]$ and $h_2[n]$. Sketch $h_s[n]$ over $(-10 \le n \le 10)$.

3.8.25 This problem investigates an interesting application of discrete-time convolution: the expansion of certain polynomial expressions.

a. By hand, expand $(z^3 + z^2 + z + 1)^2$. Compare the coefficients to [1, 1, 1, 1] * [1, 1, 1, 1].

b. Formulate a relationship between discrete-time convolution and the expansion of constant-coefficient polynomial expressions.

c. Use convolution to expand $(z^{-4} - 2z^{-3} + 3z^{-2})^4$.

d. Use convolution to expand $(z^5 + 2z^4 + 3z^2 + 5)^2(z^{-4} - 5z^{-2} + 13)$.

3.8.26 Joe likes coffee, and he drinks his coffee according to a very particular routine. He begins by adding two teaspoons of sugar to his mug, which he then fills to the brim with hot coffee. He drinks 2/3 of the mug's contents, adds another two teaspoons of sugar, and tops the mug off with steaming hot coffee. This refill procedure continues, sometimes for many, many cups of coffee. Joe has noted that his coffee tends to taste sweeter with the number of refills.

Let independent variable n designate the coffee refill number. In this way, $n = 0$ indicates the first cup of coffee, $n = 1$ is the first refill, and so forth. Let $x[n]$ represent the sugar (measured in teaspoons) added into the system (a coffee mug) on refill n. Let $y[n]$ designate the amount of sugar (again, teaspoons) contained in the mug on refill n.

a. The sugar (teaspoons) in Joe's coffee can be represented using a standard second-order constant coefficient difference equation $y[n] + a_1 y[n-1] + a_2 y[n-2] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$. Determine the constants a_1, a_2, b_0, b_1, and b_2.

b. Determine $x[n]$, the driving function to this system.

c. Solve the difference equation for $y[n]$. This requires finding the total solution. Joe always starts with a clean mug from the dishwasher, so $y[-1]$ (the sugar content before the first cup) is zero.

d. Determine the steady-state value of $y[n]$. That is, what is $y[n]$ as $n \to \infty$? If possible, suggest a way of modifying $x[n]$ so that the sugar content of Joe's coffee remains a constant for all nonnegative n.

3.8.27 A system is called complex if a real-valued input can produce a complex-valued output. Consider a causal complex system described by a first-order constant coefficient linear difference equation:

$$(jE + 0.5)y[n] = (-5E)x[n]$$

a. Determine the impulse response function $h[n]$ for this system.

b. Given input $x[n] = u[n-5]$ and initial condition $y_0[-1] = j$, determine the system's total output $y[n]$ for $n \ge 0$.

3.8.28 A discrete-time LTI system has impulse response function $h[n] = n(u[n-2] - u[n+2])$.

a. Carefully sketch the function $h[n]$ over $(-5 \le n \le 5)$.

b. Determine the difference equation representation of this system, using $y[n]$ to designate the output and $x[n]$ to designate the input.

3.8.29 Do part (a) of Prob. 3.M-3.

3.8.30 Consider three discrete-time signals: $x[n]$, $y[n]$, and $z[n]$. Denoting convolution as *, identify the expression(s) that is(are) equivalent to $x[n](y[n] * z[n])$:

a. $(x[n] * y[n])z[n]$

b. $(x[n]y[n]) * (x[n]z[n])$

c. $(x[n]y[n]) * z[n]$

d. none of the above

Justify your answer!

3.9.1 Use the classical method to solve
$$y[n+1]+2y[n]=x[n+1]$$
with the input $x[n] = e^{-n}u[n]$, and the auxiliary condition $y[0] = 1$.

3.9.2 Use the classical method to solve
$$y[n]+2y[n-1]=x[n-1]$$
with the input $x[n] = e^{-n}u[n]$ and the auxiliary condition $y[-1] = 0$. [Hint: You will have to determine the auxiliary condition $y[0]$ by using the iterative method.]

3.9.3

a. Use the classical method to solve
$$y[n+2]+3y[n+1]+2y[n]=$$
$$x[n+2]+3x[n+1]+3x[n]$$
with the input $x[n] = (3)^n$ and the auxiliary conditions $y[0] = 1$, $y[1] = 3$.

b. Repeat part (a) for auxiliary conditions $y[-1] = y[-2] = 1$. [Hint: Use the iterative method to determine to $y[0]$ and $y[1]$.]

3.9.4 Use the classical method to solve
$$y[n]+2y[n-1]+y[n-2]=$$
$$2x[n]-x[n-1]$$
with the input $x[n] = 3^{-n}u[n]$ and the auxiliary conditions $y[0] = 2$ and $y[1] = -13/3$.

3.9.5 Use the classical method to find the following sums:

a. $\displaystyle\sum_{k=0}^{n} k$

b. $\displaystyle\sum_{k=0}^{n} k^3$

3.9.6 Repeat Prob. 3.9-5 to find $\Sigma^n_{k=0} kr^k$.

3.9.7 Use the classical method to solve
$$(E^2-E+0.16)y[n]=Ex[n]$$
with the input $x[n] = (0.2)^n u[n]$ and the auxiliary conditions $y[0] = 1$, $y[1] = 2$. [Hint: The input is a natural mode of the system.]

3.9.8 Use the classical method to solve
$$(E^2-E+0.16)y[n]=Ex[n]$$
with the input
$$x[n]=\cos\left(\frac{\pi}{2}n+\frac{\pi}{3}\right)u[n]$$
and the initial conditions $y[-1] = y[-2] = 0$. [Hint: Find $y[0]$ and $y[1]$ iteratively.]

3.10.1 In Section 3.10 we showed that for BIBO stability in an LTID system, it is sufficient for its impulse response $h[n]$ to satisfy Eq. (3.90). Show that this is also a necessary condition for the system to be BIBO stable. In other words, show that if Eq. (3.90) is not satisfied, there exists a bounded input that produces unbounded output. [Hint: Assume that a system exists for which $h[n]$ violates Eq. (3.90), yet its output is bounded for every bounded input. Establish the contradiction in this statement by considering an input $x[n]$ defined by $x[n_1 - m] = 1$ when $h[m] > 0$ and $x[n_1 - m] = -1$ when $h[m] < 0$, where n_1 is some fixed integer.]

3.10.2 Each of the following equations specifies an LTID system. Determine whether each of these systems is BIBO stable or unstable. Determine also whether each is asymptotically stable, unstable, or marginally stable.

a. $y[n+2]+0.6y[n+1]-0.16y[n] = x[n+1]-2x[n]$

b. $y[n]+3y[n-1]+2y[n-2] = x[n-1]+2x[n-2]$

c. $(E-1)^2 (E+1/2)\, y[n] = x[n]$

d. $y[n]+2y[n-1]+0.96y[n-2] = x[n]$

e. $y[n]+y[n-1]-2y[n-2] = x[n] + 2x[n-1]$

f. $(E^2 - 1)(E^2 + 1)y[n] = x[n]$

3.10.3 Consider two LTIC systems in cascade, as illustrated in Fig. 3.23. The impulse response of the system S_1 is $h_1[n] = 2^n u[n]$ and the impulse response of the system S_2 is $h_2[n] = \delta[n] - 2\delta[n - 1]$. Is the cascaded system asymptotically stable or unstable? Determine the BIBO stability of the composite system.

3.10.4 Figure P3.10-4 locates the characteristic roots of nine causal, LTID systems, labeled A through I. Each system has only two roots and is described using operator notation as $Q(E)y[n] = P(E)x[n]$. All plots are drawn to scale, with the unit circle shown for reference. For each of the following parts, identify all the answers that are correct.

a. Identify all systems that are unstable.

b. Assuming all systems have $P(E) = E^2$, identify all systems that are real. Recall that a real system always generates a real-valued response to a real-valued input.

c. Identify all systems that support oscillatory natural modes.

d. Identify all systems that have at least one mode whose envelop decays at a rate of 2^{-n}.

e. Identify all systems that have only one mode.

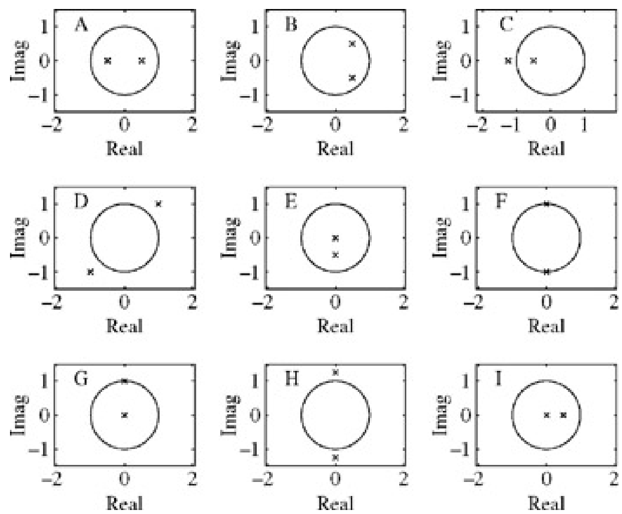

Figure P3.10-4: Characteristic roots for systems A through I.

3.10.5 A discrete-time LTI system has impulse response given by

$$h[n] = \delta[n] + \left(\frac{1}{3}\right)^n u[n-1]$$

a. Is the system stable? Is the system causal? Justify your answers.

b. Plot the signal $x[n] = u[n - 3] - u[n + 3]$.

c. Determine the system's zero-state response $y[n]$ to the input $x[n] = u[n - 3] - u[n + 3]$. Plot $y[n]$ over $(-10 \le n \le 10)$.

3.10.6 An LTID system has impulse response given by

$$h[n] = \left(\frac{1}{2}\right)^{|n|}$$

a. Is the system causal? Justify your answer.

b. Compute $\Sigma^{\infty}_{n=-\infty} |h[n]|$. Is this system BIBO stable?

c. Compute the energy and power of input signal $x[n] = 3u[n - 5]$.

d. Using input $x[n] = 3u[n - 5]$, determine the zero-state response of this system at time $n = 10$. That is, determine $y_{zs}[10]$.

3.m.1 Consider the discrete-time function $f[n] = e^{-n/5}\cos(\pi n/5)u[n]$. MATLAB Session 3 uses an inline object in describing this function.

```
>> f = inline('exp(-n/5).* cos(pi*n/5).* (n>=0)','n');
```

While this inline object operates correctly for a downsampling operation such as `f[2n]`, it does not operate correctly for an upsampling operation such as `f[n/2]`. Modify the inline object `f` so that it also correctly accommodates upsampling operations. Test your code by computing and plotting `f[n/2]` over $(-10 \le n \le 10)$.

3.m.2 An indecisive student contemplates whether he should stay home or take his final exam, which is being held 2 miles away. Starting at home, the student travels half the distance to the exam location before changing his mind. The student turns around and travels half the distance between his current location and his home before changing his mind again. This process of changing direction and traveling half the remaining distance continues until the student either reaches a destination or dies from exhaustion.

a. Determine a suitable difference equation description of this system.

b. Use MATLAB to simulate the difference equation in part (a). Where does the student end up as $n \to \infty$? How do your answer change if the student goes two-thirds the way each time, rather than halfway?

c. Determine a closed-form solution to the equation in part (a). Use this solution to verify the results in part (b).

3.m.3 The cross-correlation function between $x[n]$ and $y[n]$ is given as

$$r_{xy}[k] = \sum_{n=-\infty}^{\infty} x[n]y[n-k]$$

Notice that $r_{xy}[k]$ is quite similar to the convolution sum. The independent variable k corresponds to the relative *shift* between the two inputs.

a. Express $r_{xy}[k]$ in terms of convolution. Is $r_{xy}[k] = r_{yx}[k]$?

b. Cross-correlation is said to indicate similarity between two signals. Do you agree? Why or why not?

c. If $x[n]$ and $y[n]$ are both finite duration, MATLAB's `conv` command is well suited to compute $r_{xy}[k]$.

i. Write a MATLAB function that computes the cross-correlation function using the `conv` command. Four vectors are passed to the function (`x, y, nx,` and `ny`) corresponding to the inputs $x[n]$, $y[n]$, and their respective time vectors. Notice that, `x` and `y` are not necessarily the same length. Two outputs should be created (`rxy` and `k`) corresponding to $r_{xy}[k]$ and its shift vector.

ii. Test your code from part c(i), using $x[n] = u[n-5] - u[n-10]$ over $(0 \le n = n_x \le 20)$ and $y[n] = u[-n-15] - u[-n-10] + \delta[n-2]$ over $(-20 \le n = n_y \le 10)$. Plot the result `rxy` as a function of the shift vector `k`. What shift k gives the largest magnitude of $r_{xy}[k]$? Does this make sense?

3.m.4 A causal N-point max filter assigns $y[n]$ to the maximum of $\{x[n],..., x[n-(N-1)]\}$.

a. Write a MATLAB function that performs N-point max filtering on a length-M input vector `x`. The two function inputs are vector `x` and scalar `N`. To create the length-M output vector `Y`, initially pad the input vector with $N-1$ zeros. The MATLAB command `max` may be helpful.

b. Test your filter and MATLAB code by filtering a length-45 input defined as $x[n] = \cos(\pi n/5) + \delta[n-30] - \delta[n-35]$. Separately plot the results for $N = 4$, $N = 8$, and $N = 12$. Comment on the filter behavior.

3.m.5 A causal N-point min filter assigns $y[n]$ to the minimum of $\{x[n],..., x[n-(N-1)]\}$.

a. Write a MATLAB function that performs N-point min filtering on a length-M input vector `x`. The two function inputs are vector `x` and scalar `N`. To create the length-M output vector `Y`, initially pad the input vector with $N-1$ zeros. The MATLAB command `min` may be helpful.

b. Test your filter and MATLAB code by filtering a length-45 input defined as $x[n] = \cos(\pi n/5) + \delta[n-30] - \delta[n-35]$. Separately plot the results for $N = 4$, $N = 8$, and $N = 12$. Comment on the filter behavior.

3.m.6 A causal N-point median filter assigns $y[n]$ to the median of $\{x[n],..., x[n-(N-1)]\}$. The median is found by sorting sequence $\{x[n],..., x[n-(N-1)]\}$ and choosing the middle value (odd N) or the average of the two middle values (even N).

a. Write a MATLAB function that performs N-point median filtering on a length-M input vector `x`. The two function inputs are vector `x` and scalar `N`. To create the length-M output vector `Y`, initially pad the input vector with $N-1$ zeros. The MATLAB command `sort` or `median` may be helpful.

b. Test your filter and MATLAB code by filtering a length-45 input defined as $x[n] = \cos(\pi n/5) + \delta[n-30] - \delta[n-35]$. Separately plot the results for $N = 4$, $N = 8$, and $N = 12$. Comment on the filter behavior.

3.m.7 Recall that $y[n] = x[n/N]$ represents an up-sample by N operation. An interpolation filter replaces the inserted zeros with more realistic values. A linear interpolation filter has impulse response

$$h[n] = \sum_{k=-(N-1)}^{N-1} \left(1 - \left|\frac{k}{N}\right|\right) \delta(n-k)$$

a. Determine a constant coefficient difference equation that has impulse response $h[n]$.

b. The impulse response $h[n]$ is noncausal. What is the smallest time shift necessary to make the filter causal? What is the effect of this shift on the behavior of the filter?

c. Write a MATLAB function that will compute the parameters necessary to implement an interpolation `filter` using MATLAB's `filter` command. That is, your function should output filter vectors `b` and `a` given an input scalar N.

d. Test your filter and MATLAB code. To do this, create $x[n] = \cos(n)$ for $(0 \le n \le 9)$. Upsample $x[n]$ by $N = 10$ to create a new signal $x_{up}[n]$. Design the corresponding $N = 10$ linear interpolation filter, filter $x_{up}[n]$ to produce $y[n]$, and plot the results.

3.m.8 A causal N-point moving-average filter has impulse response $h[n] = (u[n] - u[n-N])/\mathrm{N}$.

a. Determine a constant coefficient difference equation that has impulse response $h[n]$.

b. Write a MATLAB function that will compute the parameters necessary to implement an N-point moving-average filter using MATLAB's `filter` command. That is, your function should output filter vectors `b` and `a` given a scalar input N.

c. Test your filter and MATLAB code by filtering a length-45 input defined as $x[n] = \cos(\pi n/5) + \delta[n - 30] - \delta[n - 35]$. Separately plot the results for $N = 4$, $N = 8$, and $N = 12$. Comment on the filter behavior.

d. Problem 3.M-7 introduces linear interpolation filters, for use following an up-sample by N operation. Within a scale factor, show that a cascade of two $N-$point moving-average filters is equivalent to the linear interpolation filter. What is the scale factor difference? Test this idea with MATLAB. Create $x[n] = \cos(n)$ for $(0 \le n \le 9)$. Upsample $x[n]$ by $N = 10$ to create a new signal $x_{up}[n]$. Design an $N = 10$ moving-average filter. Filter $x_{up}[n]$ twice and scale to produce $y[n]$. Plot the results. Does the output from the cascaded pair of moving-average filters linearly interpolate the upsampled data?