EXAMPLE 2.2.2

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

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

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

Solution. The output of the system is defined as

$$y(n) = \sum_{k=-\infty}^{n} x(k) = \sum_{k=-\infty}^{-1} x(k) + \sum_{k=0}^{n} x(k)$$

$$= y(-1) + \sum_{k=0}^{n} x(k)$$

$$= y(-1) + \frac{n(n+1)}{2}$$

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

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

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

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

2.2.2 Block Diagram Representation of Discrete-Time Systems

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

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

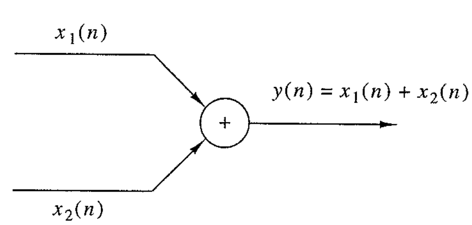

Figure 2.2.2
Graphical representation of an adder.

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

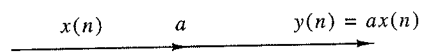

Figure 2.2.3
Graphical representation of a constant multiplier.

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

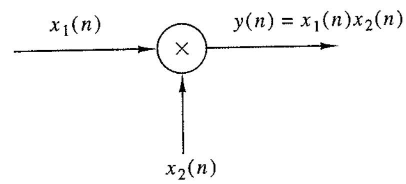

Figure 2.2.4
Graphical representation of a signal multiplier.

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

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

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

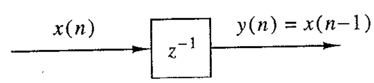

Figure 2.2.5
Graphical representation of the unit delay element.

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

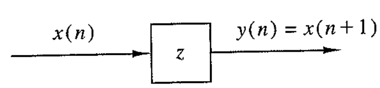

Figure 2.2.6
Graphical representation of the unit advance element.

EXAMPLE 2.2.3

Using basic building blocks introduced above, sketch the block diagram representation of the discrete-time system described by the input–output relation

$$y(n) = \frac{1}{4}y(n-1) + \frac{1}{2}x(n) + \frac{1}{2}x(n-1) \tag{2.2.5}$$

where $x(n)$ is the input and $y(n)$ is the output of the system.

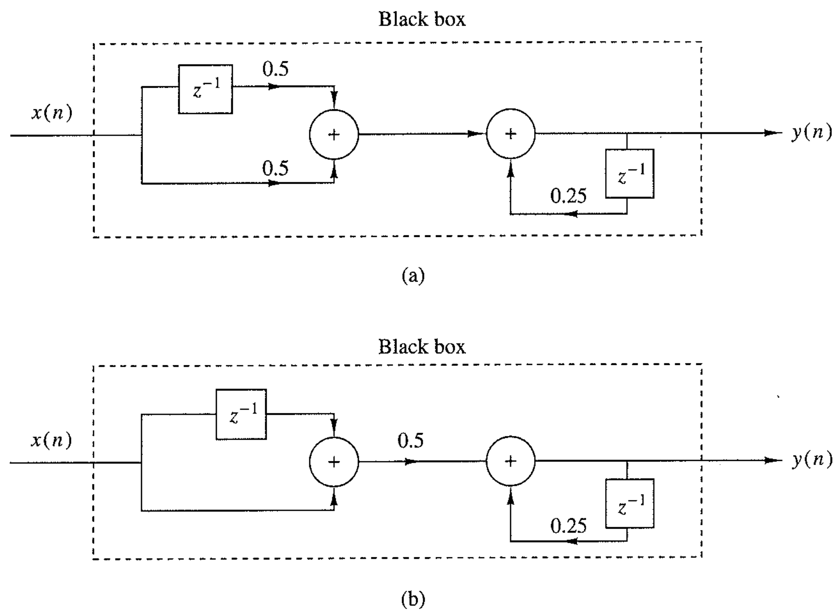

Figure 2.2.7 Block diagram realizations of the system $y(n) = 0.25y(n-1) + 0.5x(n) + 0.5x(n-1)$.

Solution. According to (2.2.5), the output $y(n)$ is obtained by multiplying the input $x(n)$ by 0.5, multiplying the previous input $x(n-1)$ by 0.5, adding the two products, and then adding the previous output $y(n-1)$ multiplied by $\frac{1}{4}$. Figure 2.2.7(a) illustrates this block diagram realization of the system. A simple rearrangement of (2.2.5), namely,

$$y(n) = \frac{1}{4}y(n-1) + \frac{1}{2}[x(n) + x(n-1)] \tag{2.2.6}$$

leads to the block diagram realization shown in Fig. 2.2.7(b). Note that if we treat "the system" from the "viewpoint" of an input–output or an external description, we are not concerned about how the system is realized. On the other hand, if we adopt an internal description of the system, we know exactly how the system building blocks are configured. In terms of such a realization, we can see that a system is *relaxed* at time $n = n_0$ if the outputs of all the *delays* existing in the system are zero at $n = n_0$ (i.e., all memory is *filled* with zeros).

2.2.3 Classification of Discrete-Time Systems

In the analysis as well as in the design of systems, it is desirable to classify the systems according to the general properties that they satisfy. In fact, the mathematical techniques that we develop in this and in subsequent chapters for analyzing and designing discrete-time systems depend heavily on the general characteristics of the systems that are being considered. For this reason it is necessary for us to develop a number of properties or categories that can be used to describe the general characteristics of systems.

We stress the point that for a system to possess a given property, the property must hold for every possible input signal to the system. If a property holds for some input signals but not for others, the system does not possess that property. Thus a counterexample is sufficient to prove that a system does not possess a property. However, to prove that the system has some property, we must prove that this property holds for every possible input signal.

Static versus dynamic systems. A discrete-time system is called *static* or memoryless if its output at any instant n depends at most on the input sample at the same time, but not on past or future samples of the input. In any other case, the system is said to be *dynamic* or to have memory. If the output of a system at time n is completely determined by the input samples in the interval from $n-N$ to $n (N \geq 0)$, the system is said to have *memory* of duration N. If $N = 0$, the system is static. If $0 < N < \infty$, the system is said to have *finite memory*, whereas if $N = \infty$, the system is said to have *infinite memory*.

The systems described by the following input–output equations

$$y(n) = ax(n) \tag{2.2.7}$$

$$y(n) = nx(n) + bx^3(n) \tag{2.2.8}$$

are both static or memoryless. Note that there is no need to store any of the past inputs or outputs in order to compute the present output. On the other hand, the systems described by the following input–output relations

$$y(n) = x(n) + 3x(n-1) \tag{2.2.9}$$

$$y(n) = \sum_{k=0}^{n} x(n-k) \tag{2.2.10}$$

$$y(n) = \sum_{k=0}^{\infty} x(n-k) \tag{2.2.11}$$

are dynamic systems or systems with memory. The systems described by (2.2.9) and (2.2.10) have finite memory, whereas the system described by (2.2.11) has infinite memory.

We observe that static or memoryless systems are described in general by input–output equations of the form

$$y(n) = \mathcal{T}[x(n), n] \tag{2.2.12}$$

and they do not include delay elements (memory).

Time-invariant versus time-variant systems. We can subdivide the general class of systems into the two broad categories, time-invariant systems and time-variant systems. A system is called time-invariant if its input–output characteristics do not change with time. To elaborate, suppose that we have a system $\mathcal{T}$ in a relaxed state which, when excited by an input signal $x(n)$, produces an output signal $y(n)$. Thus we write

$$y(n) = \mathcal{T}[x(n)] \tag{2.2.13}$$

Now suppose that the same input signal is delayed by k units of time to yield $x(n-k)$, and again applied to the same system. If the characteristics of the system do not change with time, the output of the relaxed system will be $y(n-k)$. That is, the output will be the same as the response to $x(n)$, except that it will be delayed by the same k units in time that the input was delayed. This leads us to define a time-invariant or shift-invariant system as follows.

Definition. A relaxed system $\mathcal{T}$ is *time invariant* or *shift invariant* if and only if

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

implies that

$$x(n-k) \xrightarrow{\mathcal{T}} y(n-k) \tag{2.2.14}$$

for every input signal $x(n)$ and every time shift k.

To determine if any given system is time invariant, we need to perform the test specified by the preceding definition. Basically, we excite the system with an arbitrary input sequence $x(n)$, which produces an output denoted as $y(n)$. Next we delay the input sequence by some amount k and recompute the output. In general, we can write the output as

$$y(n,k) = \mathcal{T}[x(n-k)]$$

Now if this output $y(n,k) = y(n-k)$, for all possible values of k, the system is time invariant. On the other hand, if the output $y(n,k) \neq y(n-k)$, even for one value of k, the system is time variant.

EXAMPLE 2.2.4

Determine if the systems shown in Fig. 2.2.8 are time invariant or time variant.

Solution.

(a) This system is described by the input–output equations

$$y(n) = \mathcal{T}[x(n)] = x(n) - x(n-1) \tag{2.2.15}$$

Now if the input is delayed by k units in time and applied to the system, it is clear from the block diagram that the output will be

$$y(n,k) = x(n-k) - x(n-k-1) \tag{2.2.16}$$

On the other hand, from (2.2.14) we note that if we delay $y(n)$ by k units in time, we obtain

$$y(n-k) = x(n-k) - x(n-k-1) \tag{2.2.17}$$

Since the right-hand sides of (2.2.16) and (2.2.17) are identical, it follows that $y(n,k) = y(n-k)$. Therefore, the system is time invariant.

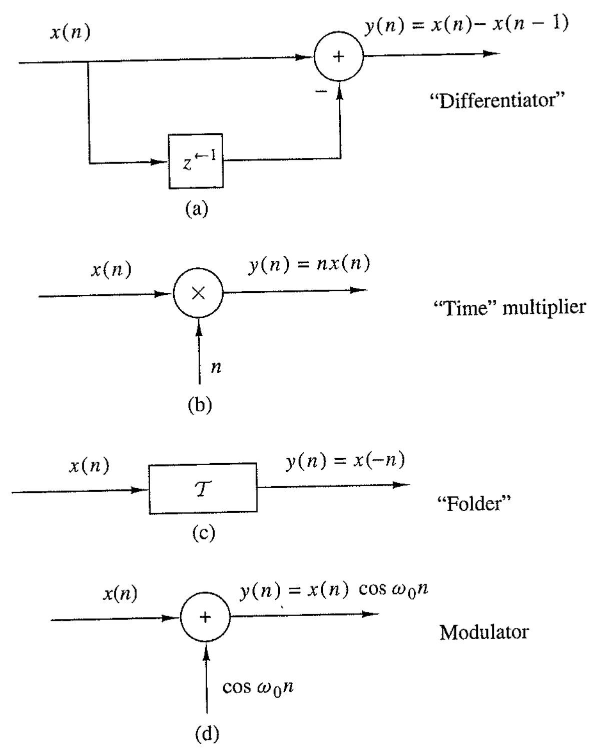

Figure 2.2.8 Examples of a time-invariant (a) and some time-variant systems (b)–(d).

(b) The input–output equation for this system is

$$y(n) = \mathcal{T}[x(n)] = nx(n) \tag{2.2.18}$$

The response of this system to $x(n-k)$ is

$$y(n, k) = nx(n-k) \tag{2.2.19}$$

Now if we delay $y(n)$ in (2.2.18) by k units in time, we obtain

$$\begin{aligned} y(n-k) &= (n-k)x(n-k) \\ &= nx(n-k) - kx(n-k) \end{aligned} \tag{2.2.20}$$

This system is time variant, since $y(n, k) \neq y(n-k)$.

(c) This system is described by the input–output relation

$$y(n) = \mathcal{T}[x(n)] = x(-n) \tag{2.2.21}$$

The response of this system to $x(n-k)$ is

$$y(n, k) = \mathcal{T}[x(n-k)] = x(-n-k) \tag{2.2.22}$$

Now, if we delay the output $y(n)$, as given by (2.2.21), by k units in time, the result will be

$$y(n-k) = x(-n+k) \tag{2.2.23}$$

Since $y(n, k) \neq y(n-k)$, the system is time variant.

(d) The input–output equation for this system is

$$y(n) = x(n)\cos\omega_0 n \tag{2.2.24}$$

The response of this system to $x(n-k)$ is

$$y(n,k) = x(n-k)\cos\omega_0 n \tag{2.2.25}$$

If the expression in (2.2.24) is delayed by k units and the result is compared to (2.2.25), it is evident that the system is time variant.

Linear versus nonlinear systems. The general class of systems can also be subdivided into linear systems and nonlinear systems. A linear system is one that satisfies the *superposition principle.* Simply stated, the principle of superposition requires that the response of the system to a weighted sum of signals be equal to the corresponding weighted sum of the responses (outputs) of the system to each of the individual input signals. Hence we have the following definition of linearity.

Definition. A system is linear if and only if

$$\mathcal{T}[a_1x_1(n) + a_2x_2(n)] = a_1\mathcal{T}[x_1(n)] + a_2\mathcal{T}[x_2(n)] \tag{2.2.26}$$

for any arbitrary input sequences $x_1(n)$ and $x_2(n)$, and any arbitrary constants a_1 and a_2. Figure 2.2.9 gives a pictorial illustration of the superposition principle.

The superposition principle embodied in the relation (2.2.26) can be separated into two parts. First, suppose that $a_2 = 0$. Then (2.2.26) reduces to

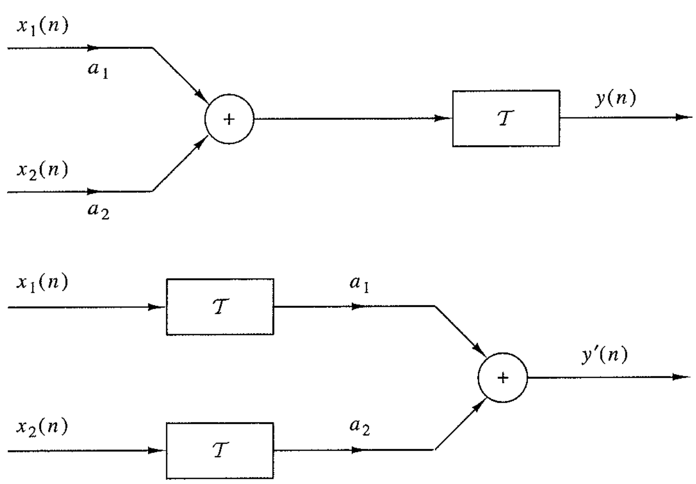

Figure 2.2.9 Graphical representation of the superposition principle. $\mathcal{T}$ is linear if and only if $y(n) = y'(n)$.

$$\mathcal{T}[a_1x_1(n)] = a_1\mathcal{T}[x_1(n)] = a_1y_1(n) \tag{2.2.27}$$

where

$$y_1(n) = \mathcal{T}[x_1(n)]$$

The relation (2.2.27) demonstrates the *multiplicative* or *scaling property* of a linear system. That is, if the response of the system to the input $x_1(n)$ is $y_1(n)$, the response to $a_1x_1(n)$ is simply $a_1y_1(n)$. Thus any scaling of the input results in an identical scaling of the corresponding output.

Second, suppose that $a_1 = a_2 = 1$ in (2.2.26). Then

$$\begin{aligned}\mathcal{T}[x_1(n) + x_2(n)] &= \mathcal{T}[x_1(n)] + \mathcal{T}[x_1(n)] \\ &= y_1(n) + y_2(n)\end{aligned} \tag{2.2.28}$$

This relation demonstrates the *additivity property* of a linear system. The additivity and multiplicative properties constitute the superposition principle as it applies to linear systems.

The linearity condition embodied in (2.2.26) can be extended arbitrarily to any weighted linear combination of signals by induction. In general, we have

$$x(n) = \sum_{k=1}^{M-1} a_kx_k(n) \xrightarrow{\mathcal{T}} y(n) = \sum_{k=1}^{M-1} a_ky_k(n) \tag{2.2.29}$$

where

$$y_k(n) = \mathcal{T}[x_k(n)], \qquad k = 1, 2, \ldots, M-1 \tag{2.2.30}$$

We observe from (2.2.27) that if $a_1 = 0$, then $y(n) = 0$. In other words, a relaxed, linear system with zero input produces a zero output. If a system produces a nonzero output with a zero input, the system may be either nonrelaxed or nonlinear. If a relaxed system does not satisfy the superposition principle as given by the definition above, it is called *nonlinear*.

EXAMPLE 2.2.5

Determine if the systems described by the following input–output equations are linear or nonlinear.

(a) $y(n) = nx(n)$ **(b)** $y(n) = x(n^2)$ **(c)** $y(n) = x^2(n)$
(d) $y(n) = Ax(n) + B$ **(e)** $y(n) = e^{x(n)}$

Solution.

(a) For two input sequences $x_1(n)$ and $x_2(n)$, the corresponding outputs are

$$\begin{aligned}y_1(n) &= nx_1(n) \\ y_2(n) &= nx_2(n)\end{aligned} \tag{2.2.31}$$

A linear combination of the two input sequences results in the output

$$\begin{aligned}y_3(n) &= \mathcal{T}[a_1x_1(n) + a_2x_2(n)] = n[a_1x_1(n) + a_2x_2(n)] \\ &= a_1nx_1(n) + a_2nx_2(n)\end{aligned} \tag{2.2.32}$$

On the other hand, a linear combination of the two outputs in (2.2.31) results in the output

$$a_1 y_1(n) + a_2 y_2(n) = a_1 n x_1(n) + a_2 n x_2(n) \tag{2.2.33}$$

Since the right-hand sides of (2.2.32) and (2.2.33) are identical, the system is linear.

(b) As in part (a), we find the response of the system to two separate input signals $x_1(n)$ and $x_2(n)$. The result is

$$\begin{aligned} y_1(n) &= x_1(n^2) \\ y_2(n) &= x_2(n^2) \end{aligned} \tag{2.2.34}$$

The output of the system to a linear combination of $x_1(n)$ and $x_2(n)$ is

$$y_3(n) = \mathcal{T}[a_1 x_1(n) + a_2 x_2(n)] = a_1 x_1(n^2) + a_2 x_2(n^2) \tag{2.2.35}$$

Finally, a linear combination of the two outputs in (2.2.34) yields

$$a_1 y_1(n) + a_2 y_2(n) = a_1 x_1(n^2) + a_2 x_2(n^2) \tag{2.2.36}$$

By comparing (2.2.35) with (2.2.36), we conclude that the system is linear.

(c) The output of the system is the square of the input. (Electronic devices that have such an input–output characteristic are called square-law devices.) From our previous discussion it is clear that such a system is memoryless. We now illustrate that this system is nonlinear.

The responses of the system to two separate input signals are

$$\begin{aligned} y_1(n) &= x_1^2(n) \\ y_2(n) &= x_2^2(n) \end{aligned} \tag{2.2.37}$$

The response of the system to a linear combination of these two input signals is

$$\begin{aligned} y_3(n) &= \mathcal{T}[a_1 x_1(n) + a_2 x_2(n)] \\ &= [a_1 x_1(n) + a_2 x_2(n)]^2 \\ &= a_1^2 x_1^2(n) + 2a_1 a_2 x_1(n) x_2(n) + a_2^2 x_2^2(n) \end{aligned} \tag{2.2.38}$$

On the other hand, if the system is linear, it will produce a linear combination of the two outputs in (2.2.37), namely,

$$a_1 y_1(n) + a_2 y_2(n) = a_1 x_1^2(n) + a_2 x_2^2(n) \tag{2.2.39}$$

Since the actual output of the system, as given by (2.2.38), is not equal to (2.2.39), the system is nonlinear.

(d) Assuming that the system is excited by $x_1(n)$ and $x_2(n)$ separately, we obtain the corresponding outputs

$$\begin{aligned} y_1(n) &= Ax_1(n) + B \\ y_2(n) &= Ax_2(n) + B \end{aligned} \tag{2.2.40}$$

A linear combination of $x_1(n)$ and $x_2(n)$ produces the output

$$\begin{aligned} y_3(n) &= \mathcal{T}[a_1x_1(n) + a_2x_2(n)] \\ &= A[a_1x_1(n) + a_2x_2(n)] + B \\ &= Aa_1x_1(n) + a_2Ax_2(n) + B \end{aligned} \tag{2.2.41}$$

On the other hand, if the system were linear, its output to the linear combination of $x_1(n)$ and $x_2(n)$ would be a linear combination of $y_1(n)$ and $y_2(n)$, that is,

$$a_1y_1(n) + a_2y_2(n) = a_1Ax_1(n) + a_1B + a_2Ax_2(n) + a_2B \tag{2.2.42}$$

Clearly, (2.2.41) and (2.2.42) are different and hence the system fails to satisfy the linearity test.

The reason that this system fails to satisfy the linearity test is not that the system is nonlinear (in fact, the system is described by a linear equation) but the presence of the constant B. Consequently, the output depends on both the input excitation and on the parameter $B \neq 0$. Hence, for $B \neq 0$, the system is not relaxed. If we set $B = 0$, the system is now relaxed and the linearity test is satisfied.

(e) Note that the system described by the input–output equation

$$y(n) = e^{x(n)} \tag{2.2.43}$$

is relaxed. If $x(n) = 0$, we find that $y(n) = 1$. This is an indication that the system is nonlinear. This, in fact, is the conclusion reached when the linearity test is applied.

Causal versus noncausal systems. We begin with the definition of causal discrete-time systems.

Definition. A system is said to be *causal* if the output of the system at any time n [i.e., $y(n)$] depends only on present and past inputs [i.e., $x(n), x(n-1), x(n-2), \ldots$], but does not depend on future inputs [i.e., $x(n+1), x(n+2), \ldots$]. In mathematical terms, the output of a causal system satisfies an equation of the form

$$y(n) = F[x(n), x(n-1), x(n-2), \ldots] \tag{2.2.44}$$

where $F[\cdot]$ is some arbitrary function.

If a system does not satisfy this definition, it is called *noncausal.* Such a system has an output that depends not only on present and past inputs but also on future inputs.

It is apparent that in real-time signal processing applications we cannot observe future values of the signal, and hence a noncausal system is physically unrealizable (i.e., it cannot be implemented). On the other hand, if the signal is recorded so that the processing is done off-line (nonreal time), it is possible to implement a noncausal system, since all values of the signal are available at the time of processing. This is often the case in the processing of geophysical signals and images.

EXAMPLE 2.2.6

Determine if the systems described by the following input–output equations are causal or noncausal.
(a) $y(n) = x(n) - x(n-1)$ **(b)** $y(n) = \sum_{k=-\infty}^{n} x(k)$ **(c)** $y(n) = ax(n)$ **(d)** $y(n) = x(n) + 3x(n+4)$
(e) $y(n) = x(n^2)$ **(f)** $y(n) = x(2n)$ **(g)** $y(n) = x(-n)$

Solution. The systems described in parts (a), (b), and (c) are clearly causal, since the output depends only on the present and past inputs. On the other hand, the systems in parts (d), (e), and (f) are clearly noncausal, since the output depends on future values of the input. The system in (g) is also noncausal, as we note by selecting, for example, $n = -1$, which yields $y(-1) = x(1)$. Thus the output at $n = -1$ depends on the input at $n = 1$, which is two units of time into the future.

Stable versus unstable systems. Stability is an important property that must be considered in any practical application of a system. Unstable systems usually exhibit erratic and extreme behavior and cause overflow in any practical implementation. Here, we define mathematically what we mean by a stable system, and later, in Section 2.3.6, we explore the implications of this definition for linear, time-invariant systems.

Definition. An arbitrary relaxed system is said to be bounded input–bounded output (BIBO) stable if and only if every bounded input produces a bounded output.

The condition that the input sequence $x(n)$ and the output sequence $y(n)$ are bounded is translated mathematically to mean that there exist some finite numbers, say M_x and M_y, such that

$$|x(n)| \le M_x < \infty, \qquad |y(n)| \le M_y < \infty \tag{2.2.45}$$

for all n. If, for some bounded input sequence $x(n)$, the output is unbounded (infinite), the system is classified as unstable.

EXAMPLE 2.2.7

Consider the nonlinear system described by the input–output equation

$$y(n) = y^2(n-1) + x(n)$$

As an input sequence we select the bounded signal

$$x(n) = C\delta(n)$$

where C is a constant. We also assume that $y(-1) = 0$. Then the output sequence is

$$y(0) = C, \quad y(1) = C^2, \quad y(2) = C^4, \quad \ldots, \quad y(n) = C^{2n}$$

Clearly, the output is unbounded when $1 < |C| < \infty$. Therefore, the system is BIBO unstable, since a bounded input sequence has resulted in an unbounded output.

2.2.4 Interconnection of Discrete-Time Systems

Discrete-time systems can be interconnected to form larger systems. There are two basic ways in which systems can be interconnected: in cascade (series) or in parallel. These interconnections are illustrated in Fig. 2.2.10. Note that the two interconnected systems are different.

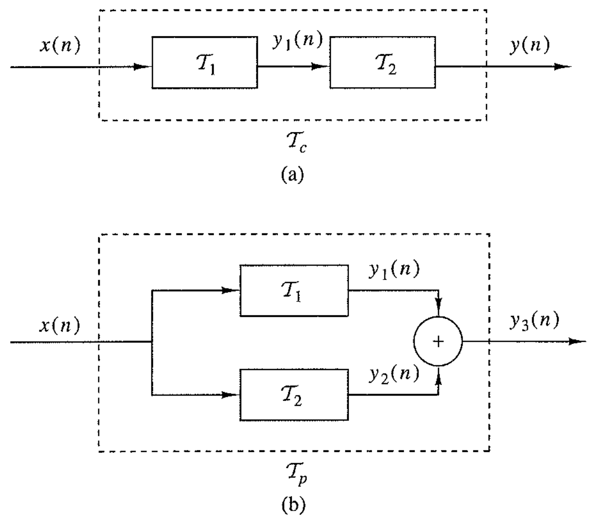

Figure 2.2.10
Cascade (a) and parallel (b) interconnections of systems.

In the cascade interconnection the output of the first system is

$$y_1(n) = \mathcal{T}_1[x(n)] \tag{2.2.46}$$

and the output of the second system is

$$\begin{aligned} y(n) &= \mathcal{T}_2[y_1(n)] \\ &= \mathcal{T}_2\{\mathcal{T}_1[x(n)]\} \end{aligned} \tag{2.2.47}$$

We observe that systems $\mathcal{T}_1$ and $\mathcal{T}_2$ can be combined or consolidated into a single overall system

$$\mathcal{T}_c \equiv \mathcal{T}_2\mathcal{T}_1 \tag{2.2.48}$$

Consequently, we can express the output of the combined system as

$$y(n) = \mathcal{T}_c[x(n)]$$

In general, the order in which the operations $\mathcal{T}_1$ and $\mathcal{T}_2$ are performed is important. That is,

$$\mathcal{T}_2\mathcal{T}_1 \neq \mathcal{T}_1\mathcal{T}_2$$

for arbitrary systems. However, if the systems $\mathcal{T}_1$ and $\mathcal{T}_2$ are linear and time invariant, then (a) $\mathcal{T}_c$ is time invariant and (b) $\mathcal{T}_2\mathcal{T}_1 = \mathcal{T}_1\mathcal{T}_2$, that is, the order in which the systems process the signal is not important. $\mathcal{T}_2\mathcal{T}_1$ and $\mathcal{T}_1\mathcal{T}_2$ yield identical output sequences.

The proof of (a) follows. The proof of (b) is given in Section 2.3.4. To prove time invariance, suppose that $\mathcal{T}_1$ and $\mathcal{T}_2$ are time invariant; then

$$x(n-k) \xrightarrow{\mathcal{T}_1} y_1(n-k)$$

and

$$y_1(n-k) \xrightarrow{\mathcal{T}_2} y(n-k)$$

Thus

$$x(n-k) \xrightarrow{\mathcal{T}_c=\mathcal{T}_2\mathcal{T}_1} y(n-k)$$

and therefore, $\mathcal{T}_c$ is time invariant.

In the parallel interconnection, the output of the system $\mathcal{T}_1$ is $y_1(n)$ and the output of the system $\mathcal{T}_2$ is $y_2(n)$. Hence the output of the parallel interconnection is

$$\begin{aligned} y_3(n) &= y_1(n) + y_2(n) \\ &= \mathcal{T}_1[x(n)] + \mathcal{T}_2[x(n)] \\ &= (\mathcal{T}_1 + \mathcal{T}_2)[x(n)] \\ &= \mathcal{T}_p[x(n)] \end{aligned}$$

where $\mathcal{T}_p = \mathcal{T}_1 + \mathcal{T}_2$.

In general, we can use parallel and cascade interconnection of systems to construct larger, more complex systems. Conversely, we can take a larger system and break it down into smaller subsystems for purposes of analysis and implementation. We shall use these notions later, in the design and implementation of digital filters.

2.3 Analysis of Discrete-Time Linear Time-Invariant Systems

In Section 2.2 we classified systems in accordance with a number of characteristic properties or categories, namely: linearity, causality, stability, and time invariance. Having done so, we now turn our attention to the analysis of the important class of linear, time-invariant (LTI) systems. In particular, we shall demonstrate that such systems are characterized in the time domain simply by their response to a unit sample sequence. We shall also demonstrate that any arbitrary input signal can be decomposed and represented as a weighted sum of unit sample sequences. As a consequence of the linearity and time-invariance properties of the system, the response of the system to any arbitrary input signal can be expressed in terms of the unit sample response of the system. The general form of the expression that relates the unit sample response of the system and the arbitrary input signal to the output signal, called the convolution sum or the convolution formula, is also derived. Thus we are able to determine the output of any linear, time-invariant system to any arbitrary input signal.

2.3.1 Techniques for the Analysis of Linear Systems

There are two basic methods for analyzing the behavior or response of a linear system to a given input signal. One method is based on the direct solution of the input–output equation for the system, which, in general, has the form

$$y(n) = F[y(n-1), y(n-2), \ldots, y(n-N), x(n), x(n-1), \ldots, x(n-M)]$$

where $F[\cdot]$ denotes some function of the quantities in brackets. Specifically, for an LTI system, we shall see later that the general form of the input–output relationship is

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{2.3.1}$$

where $\{a_k\}$ and $\{b_k\}$ are constant parameters that specify the system and are independent of $x(n)$ and $y(n)$. The input–output relationship in (2.3.1) is called a difference equation and represents one way to characterize the behavior of a discrete-time LTI system. The solution of (2.3.1) is the subject of Section 2.4.

The second method for analyzing the behavior of a linear system to a given input signal is first to decompose or resolve the input signal into a sum of elementary signals. The elementary signals are selected so that the response of the system to each signal component is easily determined. Then, using the linearity property of the system, the responses of the system to the elementary signals are added to obtain the total response of the system to the given input signal. This second method is the one described in this section.

To elaborate, suppose that the input signal $x(n)$ is resolved into a weighted sum of elementary signal components $\{x_k(n)\}$ so that

$$x(n) = \sum_k c_k x_k(n) \tag{2.3.2}$$

where the $\{c_k\}$ are the set of amplitudes (weighting coefficients) in the decomposition of the signal $x(n)$. Now suppose that the response of the system to the elementary signal component $x_k(n)$ is $y_k(n)$. Thus,

$$y_k(n) \equiv \mathcal{T}[x_k(n)] \tag{2.3.3}$$

assuming that the system is relaxed and that the response to $c_k x_k(n)$ is $c_k y_k(n)$, as a consequence of the scaling property of the linear system.

Finally, the total response to the input $x(n)$ is

$$\begin{aligned} y(n) = \mathcal{T}[x(n)] &= \mathcal{T}\left[\sum_k c_k x_k(n)\right] \\ &= \sum_k c_k \mathcal{T}[x_k(n)] \\ &= \sum_k c_k y_k(n) \end{aligned} \tag{2.3.4}$$

In (2.3.4) we used the additivity property of the linear system.

Although to a large extent, the choice of the elementary signals appears to be arbitrary, our selection is heavily dependent on the class of input signals that we wish to consider. If we place no restriction on the characteristics of the input signals, their resolution into a weighted sum of unit sample (impulse) sequences proves to be mathematically convenient and completely general. On the other hand, if we restrict our attention to a subclass of input signals, there may be another set of elementary signals that is more convenient mathematically in the determination of the output. For example, if the input signal $x(n)$ is periodic with period N, we have already observed in Section 1.3.3 that a mathematically convenient set of elementary signals is the set of exponentials

$$x_k(n) = e^{j\omega_k n}, \qquad k = 0, 1, \ldots, N-1 \tag{2.3.5}$$

where the frequencies $\{\omega_k\}$ are harmonically related, that is,

$$\omega_k = \left(\frac{2\pi}{N}\right)k, \qquad k = 0, 1, \ldots, N-1 \tag{2.3.6}$$

The frequency $2\pi/N$ is called the fundamental frequency, and all higher-frequency components are multiples of the fundamental frequency component. This subclass of input signals is considered in more detail later.

For the resolution of the input signal into a weighted sum of unit sample sequences, we must first determine the response of the system to a unit sample sequence and then use the scaling and multiplicative properties of the linear system to determine the formula for the output given any arbitrary input. This development is described in detail as follows.

2.3.2 Resolution of a Discrete-Time Signal into Impulses

Suppose we have an arbitrary signal $x(n)$ that we wish to resolve into a sum of unit sample sequences. To utilize the notation established in the preceding section, we select the elementary signals $x_k(n)$ to be

$$x_k(n) = \delta(n-k) \tag{2.3.7}$$

where k represents the delay of the unit sample sequence. To handle an arbitrary signal $x(n)$ that may have nonzero values over an infinite duration, the set of unit impulses must also be infinite, to encompass the infinite number of delays.

Now suppose that we multiply the two sequences $x(n)$ and $\delta(n-k)$. Since $\delta(n-k)$ is zero everywhere except at $n = k$, where its value is unity, the result of this multiplication is another sequence that is zero everywhere except at $n = k$, where its value is $x(k)$, as illustrated in Fig. 2.3.1. Thus

$$x(n)\delta(n-k) = x(k)\delta(n-k) \tag{2.3.8}$$

is a sequence that is zero everywhere except at $n = k$, where its value is $x(k)$. If we repeat the multiplication of $x(n)$ with $\delta(n-m)$, where m is another delay ($m \neq k$), the result will be a sequence that is zero everywhere except at $n = m$, where its value is $x(m)$. Hence

$$x(n)\delta(n-m) = x(m)\delta(n-m) \tag{2.3.9}$$

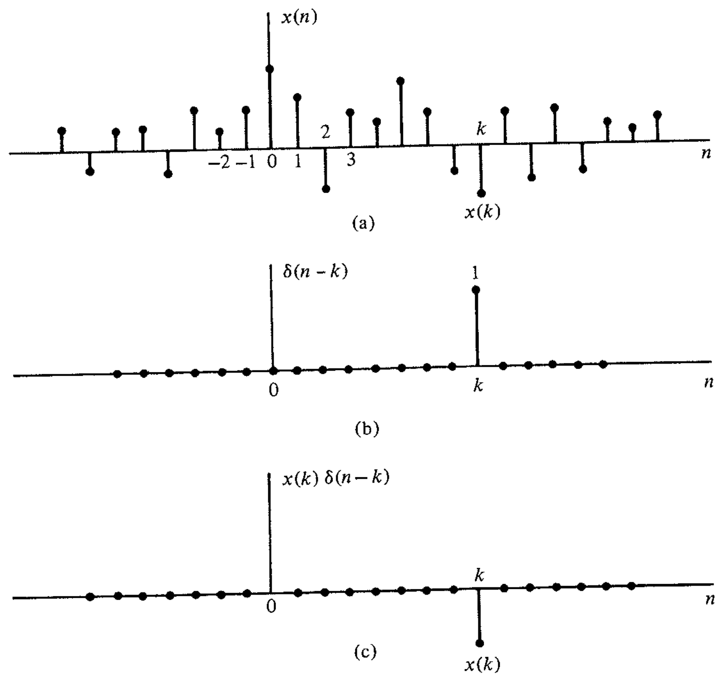

Figure 2.3.1 Multiplication of a signal $x(n)$ with a shifted unit sample sequence.

In other words, each multiplication of the signal $x(n)$ by a unit impulse at some delay k, [i.e., $\delta(n-k)$], in essence picks out the single value $x(k)$ of the signal $x(n)$ at the delay where the unit impulse is nonzero. Consequently, if we repeat this multiplication over all possible delays, $-\infty < k < \infty$, and sum all the product sequences, the result will be a sequence equal to the sequence $x(n)$, that is,

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \tag{2.3.10}$$

We emphasize that the right-hand side of (2.3.10) is the summation of an infinite number of scaled unit sample sequences where the unit sample sequence $\delta(n-k)$ has an amplitude value of $x(k)$. Thus the right-hand side of (2.3.10) gives the resolution or decomposition of any arbitrary signal $x(n)$ into a weighted (scaled) sum of shifted unit sample sequences.

EXAMPLE 2.3.1

Consider the special case of a finite-duration sequence given as

$$x(n) = \{2, \underset{\uparrow}{4}, 0, 3\}$$

Resolve the sequence $x(n)$ into a sum of weighted impulse sequences.

Solution. Since the sequence $x(n)$ is nonzero for the time instants $n = -1, 0, 2$, we need three impulses at delays $k = -1, 0,$ Following (2.3.10) we find that

$$x(n) = 2\delta(n+1) + 4\delta(n) + 3\delta(n-2)$$

2.3.3 Response of LTI Systems to Arbitrary Inputs: The Convolution Sum

Having resolved an arbitrary input signal $x(n)$ into a weighted sum of impulses, we are now ready to determine the response of any relaxed linear system to any input signal. First, we denote the response $y(n, k)$ of the system to the input unit sample sequence at $n = k$ by the special symbol $h(n, k)$, $-\infty < k < \infty$. That is,

$$y(n, k) \equiv h(n, k) = \mathcal{T}[\delta(n-k)] \tag{2.3.11}$$

In (2.3.11) we note that n is the time index and k is a parameter showing the location of the input impulse. If the impulse at the input is scaled by an amount $c_k \equiv x(k)$, the response of the system is the correspondingly scaled output, that is,

$$c_k h(n, k) = x(k)h(n, k) \tag{2.3.12}$$

Finally, if the input is the arbitrary signal $x(n)$ that is expressed as a sum of weighted impulses, that is,

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \tag{2.3.13}$$

then the response of the system to $x(n)$ is the corresponding sum of weighted outputs, that is,

$$\begin{aligned} y(n) = \mathcal{T}[x(n)] &= \mathcal{T}\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right] \\ &= \sum_{k=-\infty}^{\infty} x(k)\mathcal{T}[\delta(n-k)] \qquad (2.3.14) \\ &= \sum_{k=-\infty}^{\infty} x(k)h(n, k) \end{aligned}$$

Clearly, (2.3.14) follows from the superposition property of linear systems, and is known as the *superposition summation.*

We note that (2.3.14) is an expression for the response of a linear system to any arbitrary input sequence $x(n)$. This expression is a function of both $x(n)$ and the responses $h(n, k)$ of the system to the unit impulses $\delta(n-k)$ for $-\infty < k < \infty$. In deriving (2.3.14) we used the linearity property of the system but not its time-invariance property. Thus the expression in (2.3.14) applies to any relaxed linear (time-variant) system.

If, in addition, the system is time invariant, the formula in (2.3.14) simplifies considerably. In fact, if the response of the LTI system to the unit sample sequence $\delta(n)$ is denoted as $h(n)$, that is,

$$h(n) \equiv \mathcal{T}[\delta(n)] \tag{2.3.15}$$

then by the time-invariance property, the response of the system to the delayed unit sample sequence $\delta(n-k)$ is

$$h(n-k) = \mathcal{T}[\delta(n-k)] \tag{2.3.16}$$

Consequently, the formula in (2.3.14) reduces to

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{2.3.17}$$

Now we observe that the relaxed LTI system is completely characterized by a single function $h(n)$, namely, its response to the unit sample sequence $\delta(n)$. In contrast, the general characterization of the output of a time-variant, linear system requires an infinite number of unit sample response functions, $h(n, k)$, one for each possible delay.

The formula in (2.3.17) that gives the response $y(n)$ of the LTI system as a function of the input signal $x(n)$ and the unit sample (impulse) response $h(n)$ is called a *convolution sum*. We say that the input $x(n)$ is convolved with the impulse response $h(n)$ to yield the output $y(n)$. We shall now explain the procedure for computing the response $y(n)$, both mathematically and graphically, given the input $x(n)$ and the impulse response $h(n)$ of the system.

Suppose that we wish to compute the output of the system at some time instant, say $n = n_0$. According to (2.3.17), the response at $n = n_0$ is given as

$$y(n_0) = \sum_{k=-\infty}^{\infty} x(k)h(n_0-k) \tag{2.3.18}$$

Our first observation is that the index in the summation is k, and hence both the input signal $x(k)$ and the impulse response $h(n_0 - k)$ are functions of k. Second, we observe that the sequences $x(k)$ and $h(n_0 - k)$ are multiplied together to form a product sequence. The output $y(n_0)$ is simply the sum over all values of the product sequence. The sequence $h(n_0 - k)$ is obtained from $h(k)$ by, first, folding $h(k)$ about $k = 0$ (the time origin), which results in the sequence $h(-k)$. The folded sequence is then shifted by n_0 to yield $h(n_0 - k)$. To summarize, the process of computing the convolution between $x(k)$ and $h(k)$ involves the following four steps.

1. *Folding*. Fold $h(k)$ about $k = 0$ to obtain $h(-k)$.
2. *Shifting*. Shift $h(-k)$ by n_0 to the right (left) if n_0 is positive (negative), to obtain $h(n_0 - k)$.
3. *Multiplication*. Multiply $x(k)$ by $h(n_0 - k)$ to obtain the product sequence $v_{n_0}(k) \equiv x(k)h(n_0 - k)$.
4. *Summation*. Sum all the values of the product sequence $v_{n_0}(k)$ to obtain the value of the output at time $n = n_0$.

We note that this procedure results in the response of the system at a single time instant, say $n = n_0$. In general, we are interested in evaluating the response of the system over all time instants $-\infty < n < \infty$. Consequently, steps 2 through 4 in the summary must be repeated, for all possible time shifts $-\infty < n < \infty$.

In order to gain a better understanding of the procedure for evaluating the convolution sum, we shall demonstrate the process graphically. The graphs will aid us in explaining the four steps involved in the computation of the convolution sum.

EXAMPLE 2.3.2

The impulse response of a linear time-invariant system is

$$h(n) = \{1, \underset{\uparrow}{2}, 1, -1\} \tag{2.3.19}$$

Determine the response of the system to the input signal

$$x(n) = \{\underset{\uparrow}{1}, 2, 3, 1\} \tag{2.3.20}$$

Solution. We shall compute the convolution according to the formula (2.3.17), but we shall use graphs of the sequences to aid us in the computation. In Fig. 2.3.2(a) we illustrate the input signal sequence $x(k)$ and the impulse response $h(k)$ of the system, using k as the time index in order to be consistent with (2.3.17).

The first step in the computation of the convolution sum is to fold $h(k)$. The folded sequence $h(-k)$ is illustrated in Fig. 2.3.2(b). Now we can compute the output at $n = 0$, according to (2.3.17), which is

$$y(0) = \sum_{k=-\infty}^{\infty} x(k)h(-k) \tag{2.3.21}$$

Since the shift $n = 0$, we use $h(-k)$ directly without shifting it. The product sequence

$$v_0(k) \equiv x(k)h(-k) \tag{2.3.22}$$

is also shown in Fig. 2.3.2(b). Finally, the sum of all the terms in the product sequence yields

$$y(0) = \sum^{\infty} h = -\infty v_0(k) = 4$$

We continue the computation by evaluating the response of the system at $n = 1$. According to (2.3.17),

$$y(1) = \sum_{h=-\infty}^{\infty} x(k)h(1-k) \tag{2.3.23}$$

The sequence $h(1-k)$ is simply the folded sequence $h(-k)$ shifted to the right by one unit in time. This sequence is illustrated in Fig. 2.3.2(c). The product sequence

$$v_1(k) = x(k)h(1-k) \tag{2.3.24}$$

is also illustrated in Fig. 2.3.2(c). Finally, the sum of all the values in the product sequence yields

$$y(1) = \sum_{k=-\infty}^{\infty} v_1(k) = 8$$

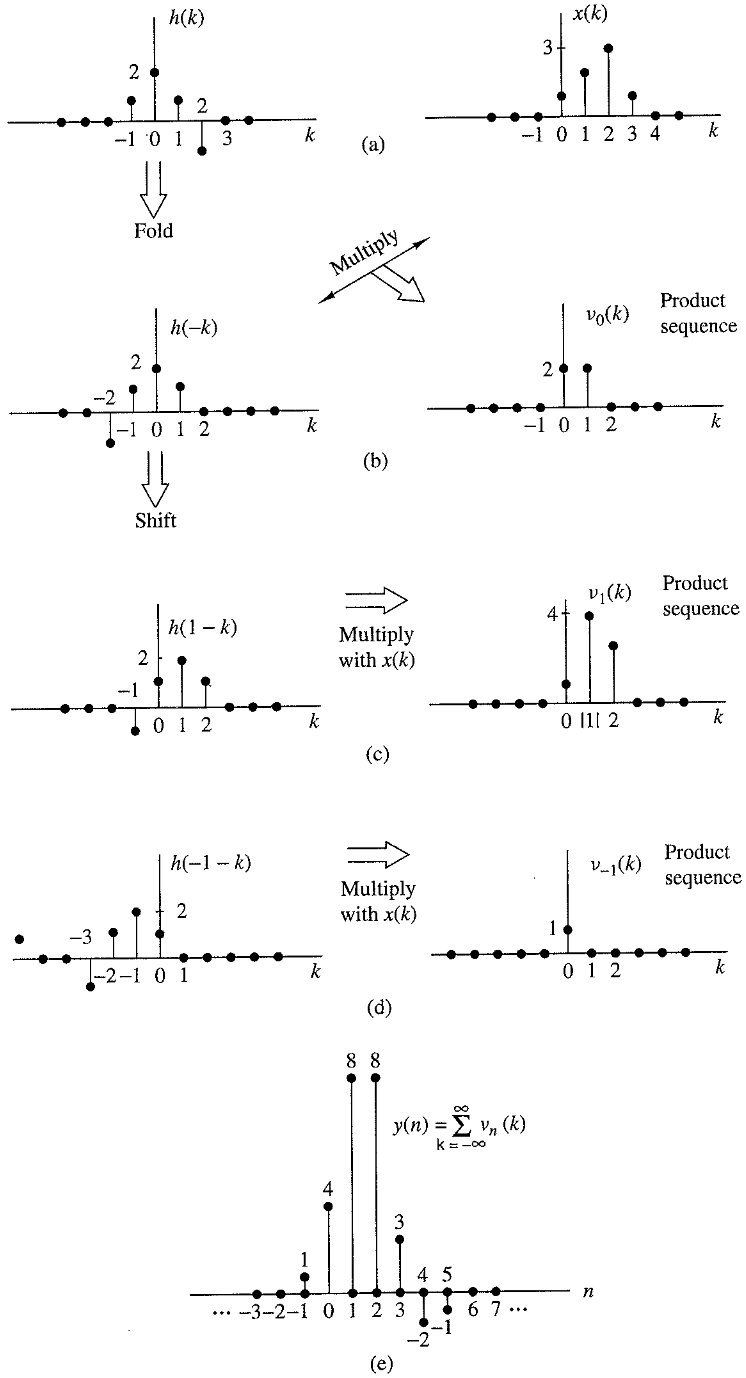

Figure 2.3.2 Graphical computation of convolution.

In a similar manner, we obtain $y(2)$ by shifting $h(-k)$ two units to the right, forming the product sequence $v_2(k) = x(k)h(2-k)$ and then summing all the terms in the product sequence obtaining $y(2) = 8$. By shifting $h(-k)$ farther to the right, multiplying the corresponding sequence, and summing over all the values of the resulting product sequences, we obtain $y(3) = 3$, $y(4) = -2$, $y(5) = -1$. For $n > 5$, we find that $y(n) = 0$ because the product sequences contain all zeros. Thus we have obtained the response $y(n)$ for $n > 0$.

Next we wish to evaluate $y(n)$ for $n < 0$. We begin with $n = -1$. Then

$$y(-1) = \sum_{k=-\infty}^{\infty} x(k)h(-1-k) \tag{2.3.25}$$

Now the sequence $h(-1-k)$ is simply the folded sequence $h(-k)$ shifted one time unit to the left. The resulting sequence is illustrated in Fig. 2.3.2(d). The corresponding product sequence is also shown in Fig. 2.3.2(d). Finally, summing over the values of the product sequence, we obtain

$$y(-1) = 1$$

From observation of the graphs of Fig. 2.3.2, it is clear that any further shifts of $h(-1-k)$ to the left always result in an all-zero product sequence, and hence

$$y(n) = 0 \qquad \text{for } n \leq -2$$

Now we have the entire response of the system for $-\infty < n < \infty$, which we summarize below as

$$y(n) = \{\ldots, 0, 0, 1, \underset{\uparrow}{4}, 8, 8, 3, -2, -1, 0, 0, \ldots\} \tag{2.3.26}$$

In Example 2.3.2 we illustrated the computation of the convolution sum, using graphs of the sequences to aid us in visualizing the steps involved in the computation procedure.

Before working out another example, we wish to show that the convolution operation is commutative in the sense that it is irrelevant which of the two sequences is folded and shifted. Indeed, if we begin with (2.3.17) and make a change in the variable of the summation, from k to m, by defining a new index $m = n - k$, then $k = n - m$ and (2.3.17) becomes

$$y(n) = \sum_{m=-\infty}^{\infty} x(n-m)h(m) \tag{2.3.27}$$

Since m is a dummy index, we may simply replace m by k so that

$$y(n) = \sum_{k=-\infty}^{\infty} x(n-k)h(k) \tag{2.3.28}$$

The expression in (2.3.28) involves leaving the impulse response $h(k)$ unaltered, while the input sequence is folded and shifted. Although the output $y(n)$ in (2.3.28) is identical to (2.3.17), the product sequences in the two forms of the convolution formula are not identical. In fact, if we define the two product sequences as

$$v_n(k) = x(k)h(n-k)$$

$$w_n(k) = x(n-k)h(k)$$

it can be easily shown that

$$v_n(k) = w_n(n-k)$$

and therefore,

$$y(n) = \sum_{k=-\infty}^{\infty} v_n(k) = \sum_{k=-\infty}^{\infty} w_n(n-k)$$

since both sequences contain the same sample values in a different arrangement. The reader is encouraged to rework Example 2.3.2 using the convolution sum in (2.3.28).

EXAMPLE 2.3.3

Determine the output $y(n)$ of a relaxed linear time-invariant system with impulse response

$$h(n) = a^{nu}(n),\ |a|1$$

when the input is a unit step sequence, that is,

$$x(n) = u(n)$$

Solution. In this case both $h(n)$ and $x(n)$ are infinite-duration sequences. We use the form of the convolution formula given by (2.3.28) in which $x(k)$ is folded. The sequences $h(k)$, $x(k)$, and $x(-k)$ are shown in Fig. 2.3.3. The product sequences $v_0(k)$, $v_1(k)$, and $v_2(k)$ corresponding to $x(-k)h(k)$, $x(1-k)h(k)$, and $x(2-k)h(k)$ are illustrated in Fig. 2.3.3(c), (d), and (e), respectively. Thus we obtain the outputs

$$y(0) = 1$$

$$y(1) = 1 + a$$

$$y(2) = 1 + a + a^2$$

Clearly, for $n > 0$, the output is

$$\begin{aligned} y(n) &= 1 + a + a^2 + \cdots + a^n \\ &= \frac{1 - a^{n+1}}{1 - a} \end{aligned} \tag{2.3.29}$$

On the other hand, for $n < 0$, the product sequences consist of all zeros. Hence

$$y(n) = 0, \qquad n < 0$$

A graph of the output $y(n)$ is illustrated in Fig. 2.3.3(f) for the case $0 < a < 1$. Note the exponential rise in the output as a function of n. Since $|a| < 1$, the final value of the output as n approaches infinity is

$$y(\infty) = \lim_{n \to \infty} y(n) = \frac{1}{1-a} \tag{2.3.30}$$

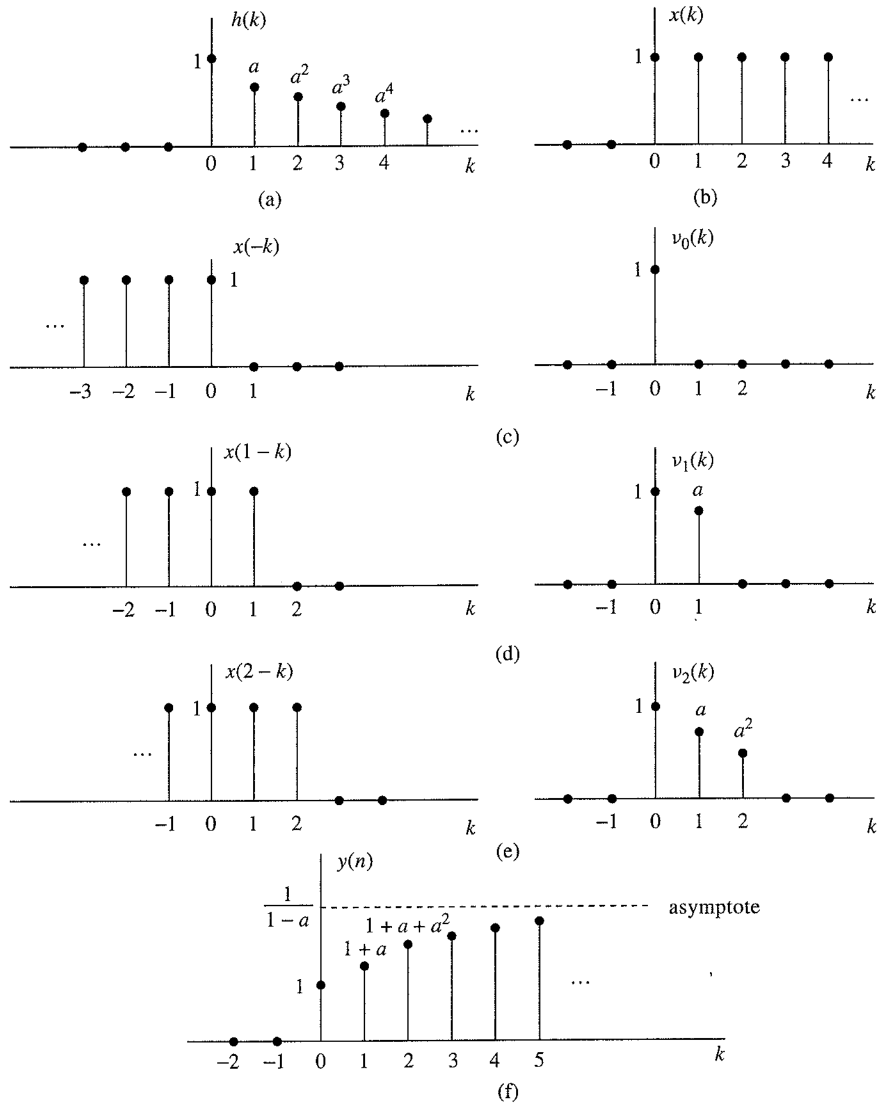

Figure 2.3.3 Graphical computation of convolution in Example 2.3.3.

To summarize, the convolution formula provides us with a means for computing the response of a relaxed, linear time-invariant system to any arbitrary input signal $x(n)$. It takes one of two equivalent forms, either (2.3.17) or (2.3.28), where $x(n)$ is the input signal to the system, $h(n)$ is the impulse response of the system, and $y(n)$ is the *output* of the system in response to the input signal $x(n)$. The evaluation of the convolution formula involves four operations, namely: *folding* either the impulse response as specified by (2.3.17) or the input sequence as specified by (2.3.28) to yield either $h(-k)$ or $x(-k)$, respectively, *shifting* the folded sequence by n units in time to yield either $h(n-k)$ or $x(n-k)$, *multiplying* the two sequences to yield the product sequence, either $x(k)h(n-k)$ or $x(n-k)h(k)$, and finally *summing* all the values in the product sequence to yield the output $y(n)$ of the system at time n. The folding operation is done only once. However, the other three operations are repeated for all possible shifts $-\infty < n < \infty$ in order to obtain $y(n)$ for $-\infty < n < \infty$.

2.3.4 Properties of Convolution and the Interconnection of LTI Systems

In this section we investigate some important properties of convolution and interpret these properties in terms of interconnecting linear time-invariant systems. We should stress that these properties hold for every input signal.

It is convenient to simplify the notation by using an asterisk to denote the convolution operation. Thus

$$y(n) = x(n) * h(n) \equiv \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{2.3.31}$$

In this notation the sequence following the asterisk [i.e., the impulse response $h(n)$] is folded and shifted. The input to the system is $x(n)$. On the other hand, we also showed that

$$y(n) = h(n) * x(n) \equiv \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{2.3.32}$$

In this form of the convolution formula, it is the input signal that is folded. Alternatively, we may interpret this form of the convolution formula as resulting from an interchange of the roles of $x(n)$ and $h(n)$. In other words, we may regard $x(n)$ as the impulse response of the system and $h(n)$ as the excitation or input signal. Figure 2.3.4 illustrates this interpretation.

Identity and Shifting Properties. We also note that the unit sample sequence $\delta(n)$ is the identity element for convolution, that is

$$y(n) = x(n) * \delta(n) = x(n)$$

If we shift $\delta(n)$ by k, the convolution sequence is shifted also by k, that is

$$x(n) * \delta(n-k) = y(n-k) = x(n-k)$$

We can view convolution more abstractly as a mathematical operation between two signal sequences, say $x(n)$ and $h(n)$, that satisfies a number of properties. The property embodied in (2.3.31) and (2.3.32) is called the commutative law.

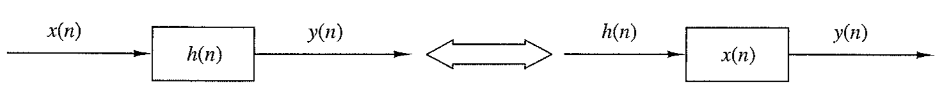

Figure 2.3.4 Interpretation of the commutative property of convolution.

Commutative law

$$x(n) * h(n) = h(n) * x(n) \tag{2.3.33}$$

Viewed mathematically, the convolution operation also satisfies the associative law, which can be stated as follows.

Associative law

$$[x(n) * h_1(n)] * h_2(n) = x(n) * [h_1(n) * h_2(n)] \tag{2.3.34}$$

From a physical point of view, we can interpret $x(n)$ as the input signal to a linear time-invariant system with impulse response $h_1(n)$. The output of this system, denoted as $y_1(n)$, becomes the input to a second linear time-invariant system with impulse response $h_2(n)$. Then the output is

$$\begin{aligned} y(n) &= y_1(n) * h_2(n) \\ &= [x(n) * h_1(n)] * h_2(n) \end{aligned}$$

which is precisely the left-hand side of (2.3.34). Thus the left-hand side of (2.3.34) corresponds to having two linear time-invariant systems in cascade. Now the right-hand side of (2.3.34) indicates that the input $x(n)$ is applied to an equivalent system having an impulse response, say $h(n)$, which is equal to the convolution of the two impulse responses. That is,

$$h(n) = h_1(n) * h_2(n)$$

and

$$y(n) = x(n) * h(n)$$

Furthermore, since the convolution operation satisfies the commutative property, one can interchange the order of the two systems with responses $h_1(n)$ and $h_2(n)$ without altering the overall input–output relationship. Figure 2.3.5 graphically illustrates the associative property.

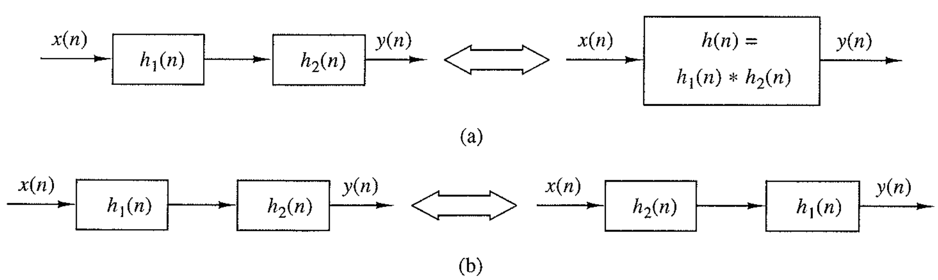

Figure 2.3.5 Implications of the associative (a) and the associative and commutative (b) properties of convolution.

EXAMPLE 2.3.4

Determine the impulse response for the cascade of two linear time-invariant systems having impulse responses

$$h_1(n) = (\frac{1}{2})^n u(n)$$

and

$$h_2(n) = (\frac{1}{4})^n u(n)$$

Solution. To determine the overall impulse response of the two systems in cascade, we simply convolve $h_1(n)$ with $h_2(n)$. Hence

$$h(n) = \sum_{k=-\infty}^{\infty} h_1(k)h_2(n-k)$$

where $h_2(n)$ is folded and shifted. We define the product sequence

$$\begin{aligned} v_n(k) &= h_1(k)h_2(n-k) \\ &= (\frac{1}{2})^k(\frac{1}{4})^{n-k} \end{aligned}$$

which is nonzero for $k \geq 0$ and $n - k \geq 0$ or $n \geq k \geq 0$. On the other hand, for $n < 0$, we have $v_n(k) = 0$ for all k, and hence

$$h(n) = 0, n < 0$$

For $n \geq k \geq 0$, the sum of the values of the product sequence $v_n(k)$ over all k yields

$$\begin{aligned} h(n) &= \sum_{k=0}^{n}(\frac{1}{2})^k(\frac{1}{4})^{n-k} \\ &= (\frac{1}{4})^n \sum_{k=0}^{n} 2^k \\ &= (\frac{1}{4})^n(2^{n+1} - 1) \\ &= (\frac{1}{2})^n[2 - (\frac{1}{2})^n], \quad n \geq 0 \end{aligned}$$

The generalization of the associative law to more than two systems in cascade follows easily from the discussion given above. Thus if we have L linear time-invariant systems in cascade with impulse responses $h_1(n), h_2(n), \ldots, h_L(n)$, there is an equivalent linear time-invariant system having an impulse response that is equal to the $(L-1)$-fold convolution of the impulse responses. That is,

$$h(n) = h_1(n) * h_2(n) * \cdots * h_L(n) \tag{2.3.35}$$

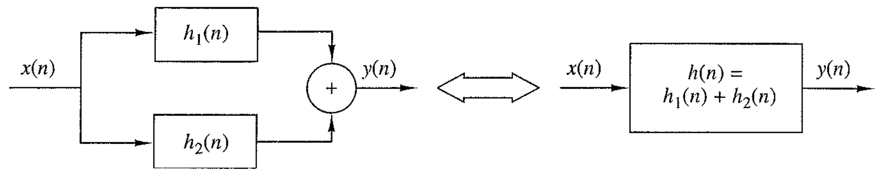

Figure 2.3.6 Interpretation of the distributive property of convolution: two LTI systems connected in parallel can be replaced by a single system with $h(n) = h_1(n) + h_2(n)$.

The commutative law implies that the order in which the convolutions are performed is immaterial. Conversely, any linear time-invariant system can be decomposed into a cascade interconnection of subsystems. A method for accomplishing the decomposition will be described later.

Another property that is satisfied by the convolution operation is the distributive law, which may be stated as follows.

Distributive law

$$x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n) \tag{2.3.36}$$

Interpreted physically, this law implies that if we have two linear time-invariant systems with impulse responses $h_1(n)$ and $h_2(n)$ excited by the same input signal $x(n)$, the sum of the two responses is identical to the response of an overall system with impulse response

$$h(n) = h_1(n) + h_2(n)$$

Thus the overall system is viewed as a parallel combination of the two linear time-invariant systems as illustrated in Fig. 2.3.6.

The generalization of (2.3.36) to more than two linear time-invariant systems in parallel follows easily by mathematical induction. Thus the interconnection of L linear time-invariant systems in parallel with impulse responses $h_1(n), h_2(n), \ldots, h_L(n)$ and excited by the same input $x(n)$ is equivalent to one overall system with impulse response

$$h(n) = \sum_{j=1}^{L} h_j(n) \tag{2.3.37}$$

Conversely, any linear time-invariant system can be decomposed into a parallel interconnection of subsystems.

2.3.5 Causal Linear Time-Invariant Systems

In Section 2.2.3 we defined a causal system as one whose output at time n depends only on present and past inputs but does not depend on future inputs. In other words, the output of the system at some time instant n, say $n = n_0$, depends only on values of $x(n)$ for $n \leq n_0$.

In the case of a linear time-invariant system, causality can be translated to a condition on the impulse response. To determine this relationship, let us consider a linear time-invariant system having an output at time $n = n_0$ given by the convolution formula

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

Suppose that we subdivide the sum into two sets of terms, one set involving present and past values of the input [i.e., $x(n)$ for $n \leq n_0$] and one set involving future values of the input [i.e., $x(n)$, $n > n_0$]. Thus we obtain

$$\begin{aligned} y(n_0) &= \sum_{k=0}^{\infty} h(k)x(n_0 - k) + \sum_{k=-\infty}^{-1} h(k)x(n_0 - k) \\ &= [h(0)x(n_0) + h(1)x(n_0 - 1) + h(2)x(n_0 - 2) + \cdots] \\ &\quad + [h(-1)x(n_0 + 1) + h(-2)x(n_0 + 2) + \cdots] \end{aligned}$$

We observe that the terms in the first sum involve $x(n_0)$, $x(n_0 - 1), \ldots,$ which are the present and past values of the input signal. On the other hand, the terms in the second sum involve the input signal components $x(n_0+1)$, $x(n_0+2), \ldots$. Now, if the output at time $n = n_0$ is to depend only on the present and past inputs, then, clearly, the impulse response of the system must satisfy the condition

$$h(n) = 0, \qquad n < 0 \tag{2.3.38}$$

Since $h(n)$ is the response of the relaxed linear time-invariant system to a unit impulse applied at $n = 0$, it follows that $h(n) = 0$ for $n < 0$ is both a necessary and a sufficient condition for causality. Hence an *LTI system is causal if and only if its impulse response is zero for negative values of* n.

Since for a causal system, $h(n) = 0$ for $n < 0$, the limits on the summation of the convolution formula may be modified to reflect this restriction. Thus we have the two equivalent forms

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n - k) \tag{2.3.39}$$

$$= \sum_{k=-\infty}^{n} x(k)h(n - k) \tag{2.3.40}$$

As indicated previously, causality is required in any real-time signal processing application, since at any given time n we have no access to future values of the input signal. Only the present and past values of the input signal are available in computing the present output.

It is sometimes convenient to call a sequence that is zero for $n < 0$, a *causal sequence*, and one that is nonzero for $n < 0$ and $n > 0$, a *noncausal sequence*. This terminology means that such a sequence could be the unit sample response of a causal or a noncausal system, respectively.

If the input to a causal linear time-invariant system is a causal sequence [i.e., if $x(n) = 0$ for $n < 0$], the limits on the convolution formula are further restricted. In this case the two equivalent forms of the convolution formula become

$$y(n) = \sum_{k=0}^{n} h(k)x(n-k) \tag{2.3.41}$$

$$= \sum_{k=0}^{n} x(k)h(n-k) \tag{2.3.42}$$

We observe that in this case, the limits on the summations for the two alternative forms are identical, and the upper limit is growing with time. Clearly, the response of a causal system to a causal input sequence is causal, since $y(n) = 0$ for $n < 0$.

EXAMPLE 2.3.5

Determine the unit step response of the linear time-invariant system with impulse response

$$h(n) = a^n u(n), \qquad |a| < 1$$

Solution. Since the input signal is a unit step, which is a causal signal, and the system is also causal, we can use one of the special forms of the convolution formula, either (2.3.41) or (2.3.42). Since $x(n) = 1$ for $n \geq 0$, (2.3.41) is simpler to use. Because of the simplicity of this problem, one can skip the steps involved with sketching the folded and shifted sequences. Instead, we use direct substitution of the signals sequences in (2.3.41) and obtain

$$y(n) = \sum_{k=0}^{n} a^k$$

$$= \frac{1 - a^{n+1}}{1 - a}$$

and $y(n) = 0$ for $n < 0$. We note that this result is identical to that obtained in Example 2.3.3. In this simple case, however, we computed the convolution algebraically without resorting to the detailed procedure outlined previously.

2.3.6 Stability of Linear Time-Invariant Systems

As indicated previously, stability is an important property that must be considered in any practical implementation of a system. We defined an arbitrary relaxed system as BIBO stable if and only if its output sequence $y(n)$ is bounded for every bounded input $x(n)$.

If $x(n)$ is bounded, there exists a constant M_x such that

$$|x(n)| \leq M_x < \infty$$

Similarly, if the output is bounded, there exists a constant M_y such that

$$|y(n)| < M_y < \infty$$

for all n.

Now, given such a bounded input sequence $x(n)$ to a linear time-invariant system, let us investigate the implications of the definition of stability on the characteristics of the system. Toward this end, we work again with the convolution formula

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

If we take the absolute value of both sides of this equation, we obtain

$$|y(n)| = \left|\sum_{k=-\infty}^{\infty} h(k)x(n-k)\right|$$

Now, the absolute value of the sum of terms is always less than or equal to the sum of the absolute values of the terms. Hence

$$|y(n)| \le \sum_{k=-\infty}^{\infty} |h(k)||x(n-k)|$$

If the input is bounded, there exists a finite number M_x such that $|x(n)| \le M_x$. By substituting this upper bound for $x(n)$ in the equation above, we obtain

$$|y(n)| \le M_x \sum_{k=-\infty}^{\infty} |h(k)|$$

From this expression we observe that the output is bounded if the impulse response of the system satisfies the condition

$$S_h \equiv \sum_{k=-\infty}^{\infty} |h(k)| < \infty \qquad (2.3.43)$$

That is, *a linear time-invariant system is stable if its impulse response is absolutely summable.* This condition is not only sufficient but it is also necessary to ensure the stability of the system. Indeed, we shall show that if $S_h = \infty$, there is a bounded input for which the output is not bounded. We choose the bounded input

$$x(n) = \begin{cases} \dfrac{h^*(-n)}{|h(-n)|}, & h(n) \neq 0 \\ 0, & h(n) = 0 \end{cases}$$

where $h^*(n)$ is the complex conjugate of $h(n)$. It is sufficient to show that there is one value of n for which $y(n)$ is unbounded. For $n = 0$ we have

$$y(0) = \sum_{k=-\infty}^{\infty} x(-k)h(k) = \sum_{k=-\infty}^{\infty} \frac{|h(k)|^2}{|h(k)|} = S_h$$

Thus, if $S_h = \infty$, a bounded input produces an unbounded output since $y(0) = \infty$.

The condition in (2.3.43) implies that the impulse response $h(n)$ goes to zero as n approaches infinity. As a consequence, the output of the system goes to zero as n approaches infinity if the input is set to zero beyond $n > n_0$. To prove this, suppose that $|x(n)| < M_x$ for $n < n_0$ and $x(n) = 0$ for $n \geq n_0$. Then, at $n = n_0 + N$, the system output is

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

But the first sum is zero since $x(n) = 0$ for $n \geq n_0$. For the remaining part, we take the absolute value of the output, which is

$$|y(n_0 + N)| = \left| \sum_{k=N}^{\infty} h(k)x(n_0 + N - k) \right| \leq \sum_{k=N}^{\infty} |h(k)||x(n_0 + N - k)|$$

$$\leq M_x \sum_{k=N}^{\infty} |h(k)|$$

Now, as N approaches infinity,

$$\lim_{N \to \infty} \sum_{k=N}^{\infty} |h(n)| = 0$$

and hence

$$\lim_{N \to \infty} |y(n_0 + N)| = 0$$

This result implies that any excitation at the input to the system, which is of a finite duration, produces an output that is "transient" in nature; that is, its amplitude decays with time and dies out eventually, when the system is stable.

EXAMPLE 2.3.6

Determine the range of values of the parameter a for which the linear time-invariant system with impulse response

$$h(n) = a^n u(n)$$

is stable.

Solution. First, we note that the system is causal. Consequently, the lower index on the summation in (2.3.43) begins with $k = 0$. Hence

$$\sum_{k=0}^{\infty} |a^k| = \sum_{k=0}^{\infty} |a|^k = 1 + |a| + |a|^2 + \cdots$$

Clearly, this geometric series converges to

$$\sum_{k=0}^{\infty} |a|^k = \frac{1}{1 - |a|}$$

provided that $|a| < 1$. Otherwise, it diverges. Therefore, the system is stable if $|a| < 1$. Otherwise, it is unstable. In effect, $h(n)$ must decay exponentially toward zero as n approaches infinity for the system to be stable.

EXAMPLE 2.3.7

Determine the range of values of a and b for which the linear time-invariant system with impulse response

$$h(n) = \begin{cases} a^n, & n \geq 0 \\ b^n, & n < 0 \end{cases}$$

is stable.

Solution. This system is noncasual. The condition on stability given by (2.3.43) yields

$$\sum_{n=-\infty}^{\infty} |h(n)| = \sum_{n=0}^{\infty} |a|^n + \sum_{n=-\infty}^{-1} |b|^n$$

From Example 2.3.6 we have already determined that the first sum converges for $|a| < 1$. The second sum can be manipulated as follows:

$$\sum_{n=-\infty}^{-1} |b|^n = \sum_{n=1}^{\infty} \frac{1}{|b|^n} = \frac{1}{|b|}\left(1 + \frac{1}{|b|} + \frac{1}{|b|^2} + \cdots\right)$$

$$= \beta(1 + \beta + \beta^2 + \cdots) = \frac{\beta}{1-\beta}$$

where $\beta = 1/|b|$ must be less than unity for the geometric series to converge. Consequently, the system is stable if both $|a| < 1$ and $|b| > 1$ are satisfied.

2.3.7 Systems with Finite-Duration and Infinite-Duration Impulse Response

Up to this point we have characterized a linear time-invariant system in terms of its impulse response $h(n)$. It is also convenient, however, to subdivide the class of linear time-invariant systems into two types, those that have a finite-duration impulse response (FIR) and those that have an infinite-duration impulse response (IIR). Thus an FIR system has an impulse response that is zero outside of some finite time interval. Without loss of generality, we focus our attention on causal FIR systems, so that

$$h(n) = 0, \qquad n < 0 \text{ and } n \geq M$$

The convolution formula for such a system reduces to

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

A useful interpretation of this expression is obtained by observing that the output at any time n is simply a weighted linear combination of the input signal samples $x(n)$, $x(n-1), \ldots, x(n-M+1)$. In other words, the system simply weights, by the values of the impulse response $h(k)$, $k = 0, 1, \ldots, M-1$, the most recent M signal samples