of numbers), we use the index "m" to keep track of individual elements in $X_1(m)$. We can list frequency-domain sequences just as we did with the time sequence in Eq. (1-2). For example $X_{sum}(m)$ is listed as

$$
\begin{array}{ll}
X_{sum}(0) = 0 & \text{(1st } X_{sum}(m) \text{ value, index } m = 0) \\
X_{sum}(1) = 1.0 & \text{(2nd } X_{sum}(m) \text{ value, index } m = 1) \\
X_{sum}(2) = 0.4 & \text{(3rd } X_{sum}(m) \text{ value, index } m = 2) \\
X_{sum}(3) = 0 & \text{(4th } X_{sum}(m) \text{ value, index } m = 3) \\
\quad \ldots & \quad \ldots \\
 & \text{and so on,}
\end{array}
$$

where the frequency index m is the integer sequence 0, 1, 2, 3, etc. Third, because the $x_1(n) + x_2(n)$ sinewaves have a phase shift of zero degrees relative to each other, we didn't really need to bother depicting this phase relationship in $X_{sum}(m)$ in Figure 1-3(c). In general, however, phase relationships in frequency-domain sequences are important, and we'll cover that subject in Chapters 3 and 5.

A key point to keep in mind here is that we now know three equivalent ways to describe a discrete-time waveform. Mathematically, we can use a time-domain equation like Eq. (1-6) for example. We can also represent a time-domain waveform graphically as we did on the left side of Figure 1-3, and we can depict its corresponding, discrete, frequency-domain equivalent as that on the right side of Figure 1-3.

As it turns out, the discrete-time domain signals we're concerned with are not only quantized in time; their amplitude values are also quantized. Because we represent all digital quantities with binary numbers, there's a limit to the resolution, or granularity, that we have in representing the values of discrete numbers. Although signal amplitude quantization can be an important consideration—we cover that particular topic in Chapter 9—we won't worry about it just now.

1.2 Signal Amplitude, Magnitude, Power

Let's define two important terms that we'll be using throughout this book: amplitude and magnitude. It's not surprising that, to the layman, these terms are typically used interchangeably. When we check our thesaurus, we find that they are synonymous.† In engineering, however, they mean

† Of course, layman are "other people." To the engineer, the brain surgeon is the layman. To the brain surgeon, the engineer is the layman.

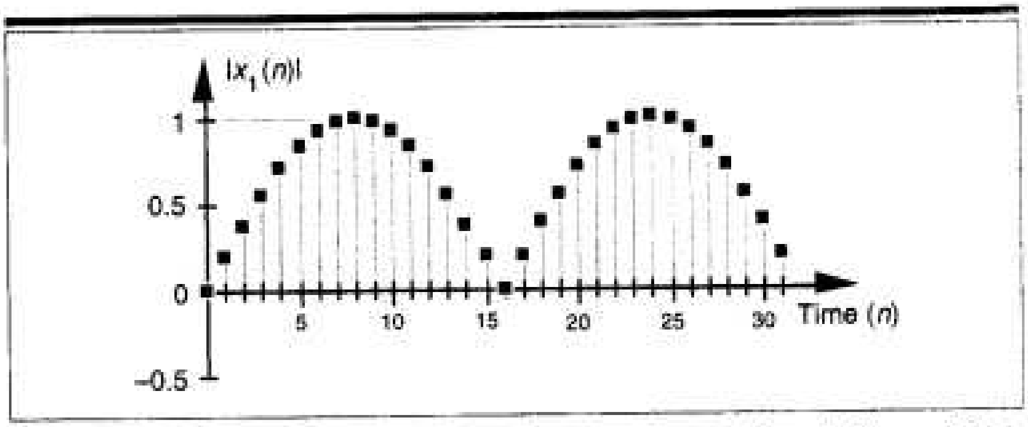

Figure 1-4 Magnitude samples, $|x_1(n)|$, of the time waveform in Figure 1-3(a).

two different things, and we must keep that difference clear in our discussions. The amplitude of a variable is the measure of how far, and in what direction, that variable differs from zero. Thus, signal amplitudes can be either positive or negative. The time-domain sequences in Figure 1-3 presented the sample value amplitudes of three different waveforms. Notice how some of the individual discrete amplitude values were positive and others were negative.

The magnitude of a variable, on the other hand, is the measure of how far, regardless of direction, its quantity differs from zero. So magnitudes are always positive values. Figure 1-4 illustrates how the magnitude of the $x_1(n)$ time sequence in Figure 1-3(a) is equal to the amplitude, but with the sign always being positive for the magnitude. We use the modulus symbol (| |) to represent the magnitude of $x_1(n)$. Occasionally, in the literature of digital signal processing, we'll find the term *magnitude* referred to as the *absolute value*.

When we examine signals in the frequency domain, we'll often be interested in the power level of those signals. The power of a signal is proportional to its amplitude (or magnitude) squared. If we assume that the proportionality constant is one, we can express the power of a sequence in the time or frequency domains as

$$x_{\mathrm{pwr}}(n) = x(n)^2 = |x(n)|^2 , \tag{1-8}$$

or

$$X_{\mathrm{pwr}}(m) = X(m)^2 = |X(m)|^2 . \tag{1-8'}$$

Very often we'll want to know the difference in power levels of two signals in the frequency domain. Because of the squared nature of power,

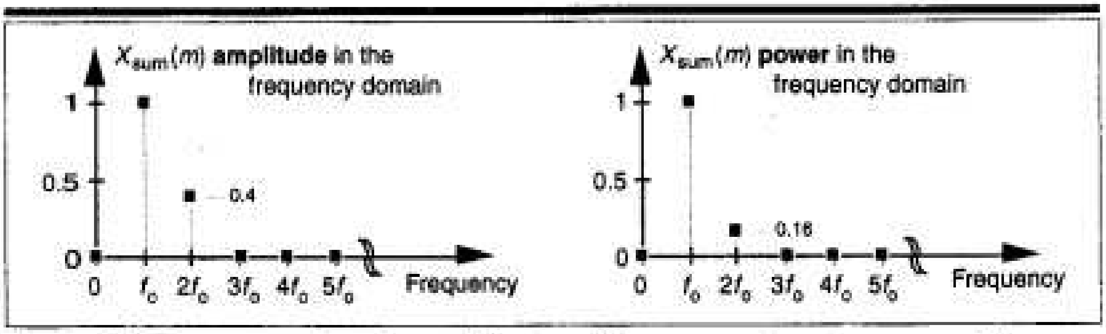

Figure 1-5 Frequency-domain amplitude and frequency-domain power of the $x_{sum}(n)$ time waveform in Figure 1-3(c).

two signals with moderately different amplitudes will have a much larger difference in their relative powers. In Figure 1-3, for example, signal $x_1(n)$'s amplitude is 2.5 times the amplitude of signal $x_2(n)$, but its power level is 6.25 that of $x_2(n)$'s power level. This is illustrated in Figure 1-5 where both the amplitude and power of $X_{sum}(m)$ are shown.

Because of their squared nature, plots of power values often involve showing both very large and very small values on the same graph. To make these plots easier to generate and evaluate, practitioners usually employ the decibel scale as described in Appendix E.

1.3 Signal Processing Operational Symbols

We'll be using block diagrams to graphically depict the way digital signal-processing operations are implemented. Those block diagrams will comprise an assortment of fundamental processing symbols, the most common of which are illustrated and mathematically defined in Figure 1-6.

Figure 1-6(a) shows the addition, element for element, of two discrete sequences to provide a new sequence. If our sequence index n begins at 0, we say that the first output sequence value is equal to the sum of the first element of the b sequence and the first element of the c sequence, or $a(0) = b(0) + c(0)$. Likewise, the second output sequence value is equal to the sum of the second element of the b sequence and the second element of the c sequence, or $a(1) = b(1) + c(1)$. Equation (1-7) is an example of adding two sequences. The subtraction process in Figure 1-6(b) generates an output sequence that's the element-for-element difference of the two input sequences. There are times when we must calculate a sequence whose elements are the sum of more than two values. This operation, illustrated in Figure 1-6(c), is called summation and is very common in

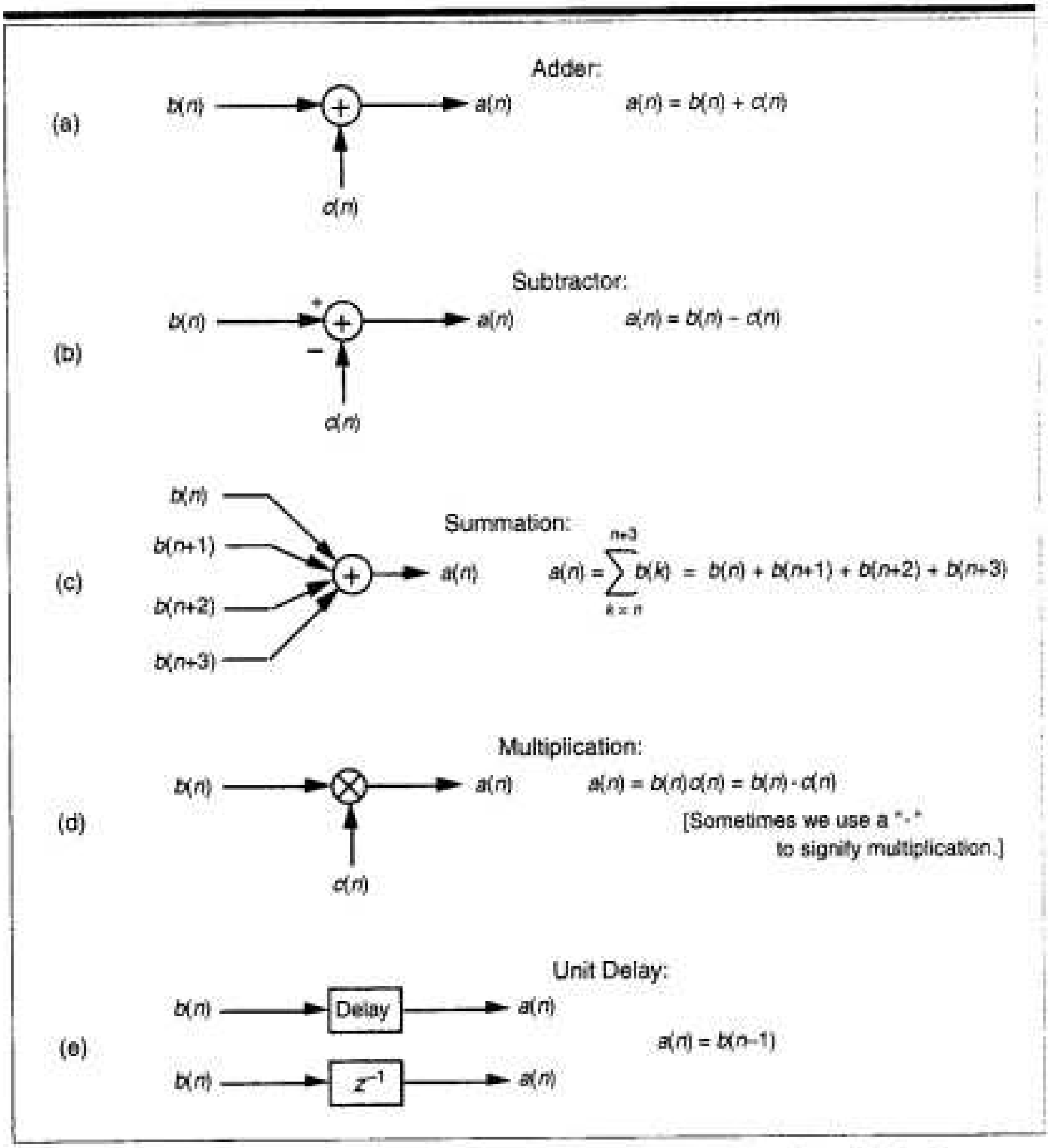

Figure 1-6 Terminology and symbols used in digital signal processing block diagrams.

digital signal processing. Notice how the lower and upper limits of the summation index k in the expression in Figure 1-6(c) tell us exactly which elements of the b sequence to sum to obtain a given $a(n)$ value. Because we'll encounter summation operations so often, let's make sure we understand their notation. If we repeat the summation equation from Figure 1-6(c) here we have

$$a(n) = \sum_{k=n}^{n+3} b(k) \,. \tag{1-9}$$

This means that

when $n = 0$, index k goes from 0 to 3, so $a(0) = b(0) + b(1) + b(2) + b(3)$
when $n = 1$, index k goes from 1 to 4, so $a(1) = b(1) + b(2) + b(3) + b(4)$
when $n = 2$, index k goes from 2 to 5, so $a(2) = b(2) + b(3) + b(4) + b(5)$
when $n = 3$, index k goes from 3 to 6, so $a(3) = b(3) + b(4) + b(5) + b(6)$

. . . . . .

and so on. (1-10)

We'll begin using summation operations in earnest when we discuss digital filters in Chapter 5.

The multiplication of two sequences is symbolized in Figure 1-6(d). Multiplication generates an output sequence that's the element-for-element product of two input sequences: $a(0) = b(0)c(0)$, $a(1) = b(1)c(1)$, and so on. The last fundamental operation that we'll be using is called the *unit delay* in Figure 1-6(e). While we don't need to appreciate its importance at this point, we'll merely state that the unit delay symbol signifies an operation where the output sequence $a(n)$ is equal to a delayed version of the $b(n)$ sequence. For example, $a(5) = b(4)$, $a(6) = b(5)$, $a(7) = b(6)$, etc. As we'll see in Chapter 6, due to the mathematical techniques used to analyze digital filters, the unit delay is very often depicted using the term z^{-1}.

The symbols in Figure 1-6 remind us of two important aspects of digital signal processing. First, our processing operations are always performed on sequences of individual discrete values, and second, the elementary operations themselves are very simple. It's interesting that, regardless of how complicated they appear to be, the vast majority of digital signal processing algorithms can be performed using combinations of these simple operations. If we think of a digital signal processing algorithm as a recipe, then the symbols in Figure 1-6 are the ingredients.

1.4 Introduction to Discrete Linear Time-Invariant Systems

In keeping with tradition, we'll introduce the subject of linear time-invariant (LTI) systems at this early point in our text. Although an appreciation for LTI systems is not essential in studying the next three chapters of this book, when we begin exploring digital filters, we'll build on the strict definitions of linearity and time invariance. We need to recognize and understand the notions of linearity and time invariance not just because the vast majority of discrete systems used in practice are LTI systems, but also because LTI systems are very accommodating when it comes to their analysis. That's good news for us because we can use straightforward methods to predict the performance of any digital signal processing scheme as long as it's linear and time invariant. Because linearity and time invariance are two important system characteristics having very special properties, we'll discuss them now.

1.5 Discrete Linear Systems

The term *linear* defines a special class of systems where the output is the superposition, or sum, of the individual outputs had the individual inputs been applied separately to the system. For example, we can say that the application of an input $x_1(n)$ to a system results in an output $y_1(n)$. We symbolize this situation with the following expression:

$$x_1(n) \xrightarrow{\text{results in}} y_1(n) \tag{1-11}$$

Given a different input $x_2(n)$, the system has a $y_2(n)$ output as

$$x_2(n) \xrightarrow{\text{results in}} y_2(n) \ . \tag{1-12}$$

For the system to be linear, when its input is the sum $x_1(n) + x_2(n)$, its output must be the sum of the individual outputs so that

$$x_1(n)+x_2(n) \xrightarrow{\text{results in}} y_1(n)+y_2(n) \ . \tag{1-13}$$

One way to paraphrase expression (1-13) is to state that a linear system's output is the sum of the outputs of its parts. Also, part of this description of linearity is a proportionality characteristic. This means that if the inputs are scaled by constant factors c_1 and c_2 then the output sequence parts are also scaled by those factors as

$$c_1x_1(n)+c_2x_2(n) \xrightarrow{\text{results in}} c_1y_1(n)+c_2y_2(n) \ . \tag{1-14}$$

In the literature, this proportionality attribute of linear systems in expression (1-14) is sometimes called the *homogeneity property*. With these thoughts in mind, then, let's demonstrate the concept of system linearity.

1.5.1 Example of a Linear System

To illustrate system linearity, let's say we have the discrete system shown in Figure 1-7(a) whose output is defined as

$$y(n) = \frac{-x(n)}{2}, \tag{1-15}$$

that is, the output sequence is equal to the negative of the input sequence with the amplitude reduced by a factor of two. If we apply an $x_1(n)$ input sequence representing a 1-Hz sinewave sampled at a rate of 32 samples

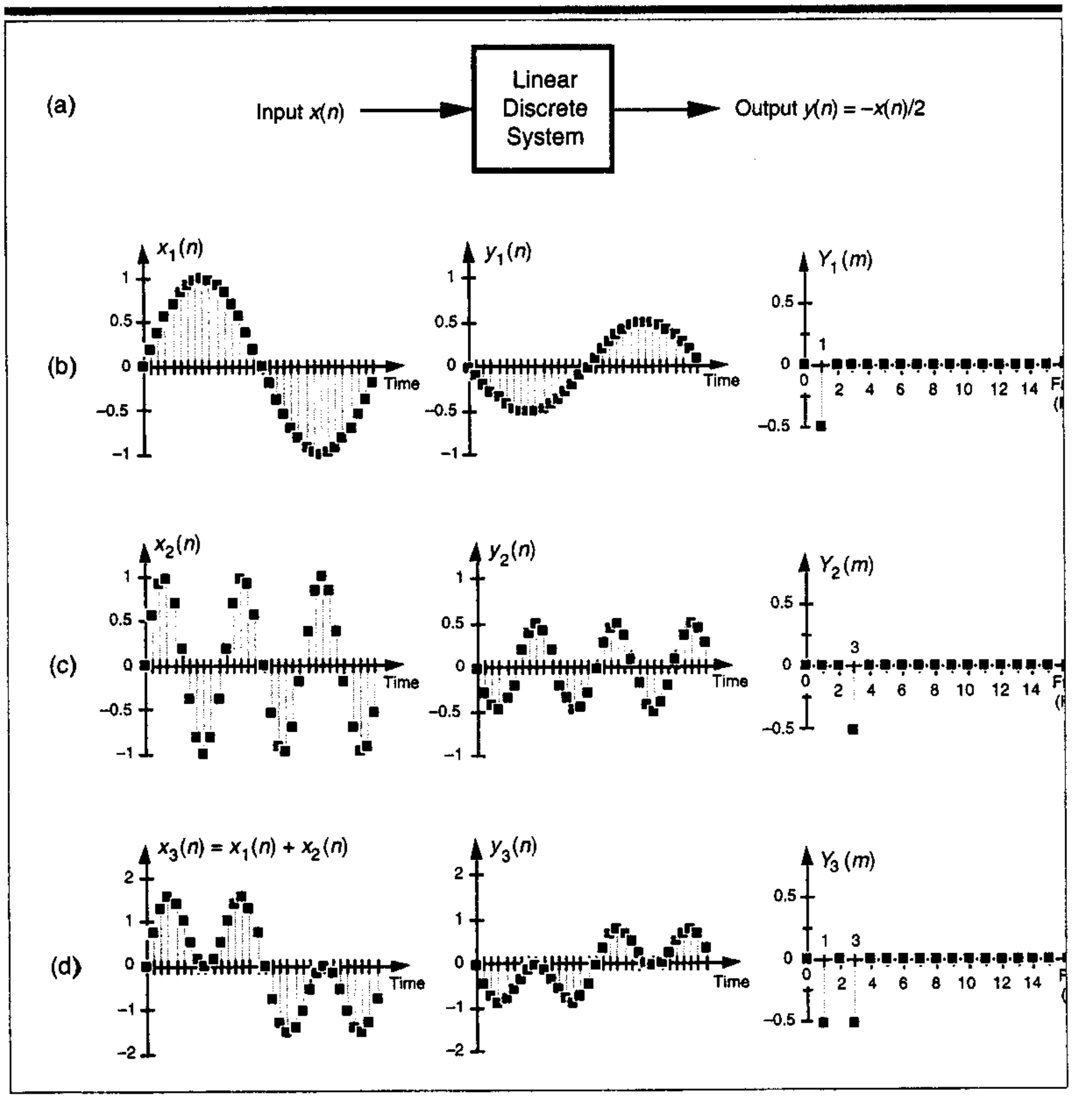

Figure 1-7 Linear system input-to-output relationships: (a) system block diagram where $y(n) = -x(n)/2$; (b) system input and output with a 1-Hz sinewave applied; (c) with a 3-Hz sinewave applied; (d) with the sum of 1-Hz and 3-Hz sinewaves applied.

per cycle, we'll have a $y_1(n)$ output as shown in the center of Figure 1-7(b). The frequency-domain spectral amplitude of the $y_1(n)$ output is the plot on the right side of Figure 1-7(b) indicating that the output comprises a single tone of peak amplitude equal to –0.5 whose frequency is 1 Hz. Next, applying an $x_2(n)$ input sequence representing a 3-Hz sinewave, the system provides a $y_2(n)$ output sequence, as shown in the center of Figure 1-7(c). The spectrum of the $y_2(n)$ output, $Y_2(m)$, confirming a single 3-Hz sinewave output is shown on the right side of Figure 1-7(c). Finally—here's where the linearity comes in—if we apply an $x_3(n)$ input sequence that's the sum of a 1-Hz sinewave and a 3-Hz sinewave, the $y_3(n)$ output is as shown in the center of Figure 1-7(d). Notice how $y_3(n)$ is the sample-for-sample sum of $y_1(n)$ and $y_2(n)$. Figure 1-7(d) also shows that the output spectrum $Y_3(m)$ is the sum of $Y_1(m)$ and $Y_2(m)$. That's linearity.

1.5.2 Example of a Nonlinear System

It's easy to demonstrate how a nonlinear system yields an output that is not equal to the sum of $y_1(n)$ and $y_2(n)$ when its input is $x_1(n) + x_2(n)$. A simple example of a nonlinear discrete system is that in Figure 1-8(a) where the output is the square of the input described by

$$y(n) = [x(n)]^2 . \tag{1-16}$$

We'll use a well-known trigonometric identity and a little algebra to predict the output of this nonlinear system when the input comprises simple sinewaves. Following the form of Eq. (1-3), let's describe a sinusoidal sequence, whose frequency $f_o = 1$ Hz, by

$$x_1(n) = \sin(2\pi f_o n t_s) = \sin(2\pi \cdot 1 \cdot n t_s) . \tag{1-17}$$

Equation (1-17) describes the $x_1(n)$ sequence on the left side of Figure 1-8(b). Given this $x_1(n)$ input sequence, the $y_1(n)$ output of the nonlinear system is the square of a 1-Hz sinewave, or

$$y_1(n) = [x_1(n)]^2 = \sin(2\pi \cdot 1 \cdot n t_s) \cdot \sin(2\pi \cdot 1 \cdot n t_s) . \tag{1-18}$$

We can simplify our expression for $y_1(n)$ in Eq. (1-18) by using the following trigonometric identity:

$$\sin(\alpha) \cdot \sin(\beta) = \frac{\cos(\alpha - \beta)}{2} - \frac{\cos(\alpha + \beta)}{2} . \tag{1-19}$$

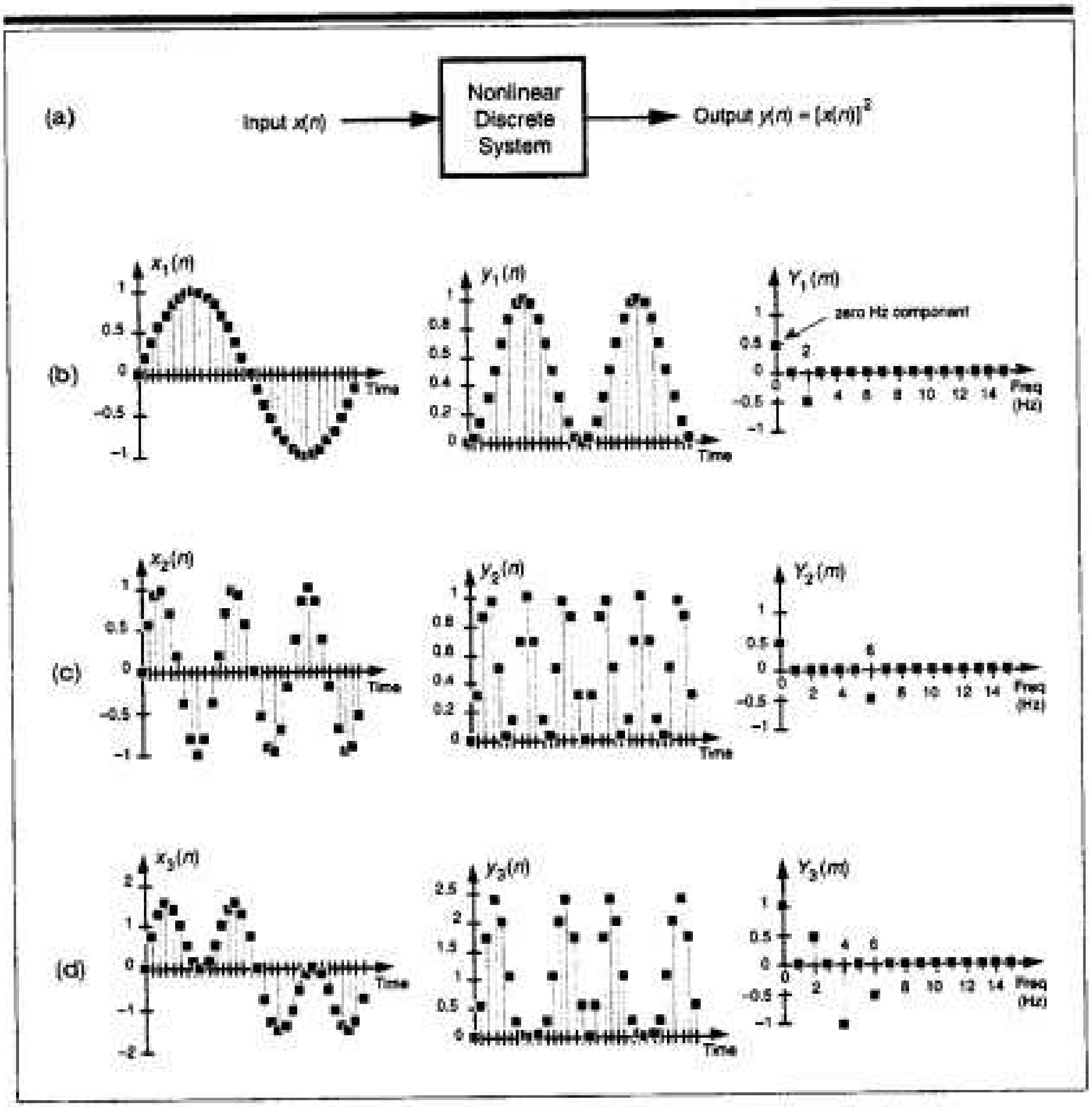

Figure 1-8 Nonlinear system input-to-output relationships: (a) system block diagram where $y(n) = (x(n))^2$; (b) system input and output with a 1-Hz sinewave applied; (c) with a 3-Hz sinewave applied; (d) with the sum of 1-Hz and 3-Hz sinewaves applied.

Using Eq. (1-19), we can express $y_1(n)$ as

$$y_1(n) = \frac{\cos(2\pi \cdot 1 \cdot nt_s - 2\pi \cdot 1 \cdot nt_s)}{2} - \frac{\cos(2\pi \cdot 1 \cdot nt_s + 2\pi \cdot 1 \cdot nt_s)}{2}$$

$$= \frac{\cos(0)}{2} - \frac{\cos(4\pi \cdot 1 \cdot nt_s)}{2} = \frac{1}{2} - \frac{\cos(2\pi \cdot 2 \cdot nt_s)}{2}, \qquad (1\text{-}20)$$

which is shown as the all positive sequence in the center of Figure 1-8(b). Because Eq. (1-19) results in a frequency sum ($\alpha + \beta$) and frequency difference ($\alpha - \beta$) effect when multiplying two sinusoids, the $y_1(n)$ output sequence will be a cosine wave of 2 Hz and a peak amplitude of −0.5, added to a constant value of 1/2. The constant value of 1/2 in Eq. (1-20) is interpreted as a zero Hz frequency component, as shown in the $Y_1(m)$ spectrum in Figure 1-8(b). We could go through the same algebraic exercise to determine that, when a 3-Hz sinewave $x_2(n)$ sequence is applied to this nonlinear system, the output $y_2(n)$ would contain a zero Hz component and a 6 Hz component, as shown in Figure 1-8(c).

System nonlinearity is evident if we apply an $x_3(n)$ sequence comprising the sum of a 1-Hz and a 3-Hz sinewave as shown in Figure 1-8(d). We can predict the frequency content of the $y_3(n)$ output sequence by using the algebraic relationship

$$(a+b)^2 = a^2+2ab+b^2 , \tag{1-21}$$

where a and b represent the 1-Hz and 3-Hz sinewaves, respectively. From Eq. (1-19), the a^2 term in Eq. (1-21) generates the zero-Hz and 2-Hz output sinusoids in Figure 1-8(b). Likewise, the b^2 term produces in $y_3(n)$ another zero-Hz and the 6-Hz sinusoid in Figure 1-8(c). However, the $2ab$ term yields additional 2-Hz and 4-Hz sinusoids in $y_3(n)$. We can show this algebraically by using Eq. (1-19) and expressing the $2ab$ term in Eq. (1-21) as

$$\begin{aligned} 2ab &= 2\cdot\sin(2\pi\cdot 1\cdot nt_s)\cdot\sin(2\pi\cdot 3\cdot nt_s) \\ &= \frac{2\cos(2\pi\cdot 1\cdot nt_s - 2\pi\cdot 3\cdot nt_s)}{2} - \frac{2\cos(2\pi\cdot 1\cdot nt_s + 2\pi\cdot 3\cdot nt_s)}{2} \\ &= \cos(2\pi\cdot 2\cdot nt_s) - \cos(2\pi\cdot 4\cdot nt_s)\ .^\dagger \end{aligned} \tag{1-22}$$

Equation (1-22) tells us that two additional sinusoidal components will be present in $y_3(n)$ because of the system's nonlinearity, a 2-Hz cosine wave whose amplitude is +1 and a 4-Hz cosine wave having an amplitude of −1. These spectral components are illustrated in $Y_3(m)$ on the right side of Figure 1-8(d).

Notice that, when the sum of the two sinewaves is applied to the nonlinear system, the output contained sinusoids, Eq. (1-22), that were not present in either of the outputs when the individual sinewaves alone were applied. Those extra sinusoids were generated by an interaction of the

† The first term in Eq. (1-22) is $\cos(2\pi\cdot nt_s - 6\pi\cdot nt_s) = \cos(-4\pi\cdot nt_s) = \cos(-2\pi\cdot 2\cdot nt_s)$. However, because the cosine function is even, $\cos(-\alpha) = \cos(\alpha)$, we can express that first term as $\cos(2\pi\cdot 2\cdot nt_s)$.

two input sinusoids due to the squaring operation. That's nonlinearity; expression (1-13) was not satisfied. (Electrical engineers recognize this effect of internally generated sinusoids as *intermodulation distortion.*) Although nonlinear systems are usually difficult to analyze, they are occasionally used in practice. References [2], [3], and [4], for example, describe their application in nonlinear digital filters. Again, expressions (1-13) and (1-14) state that a linear system's output resulting from a sum of individual inputs, is the superposition (sum) of the individual outputs. They also stipulate that the output sequence $y_1(n)$ depends only on $x_1(n)$ combined with the system characteristics, and not on the other input $x_2(n)$, i.e., there's no interaction between inputs $x_1(n)$ and $x_2(n)$ at the output of a linear system.

1.6 Time-Invariant Systems

A time-invariant system is one where a time delay (or shift) in the input sequence causes a equivalent time delay in the system's output sequence. Keeping in mind that n is just an indexing variable we use to keep track of our input and output samples, let's say a system provides an output $y(n)$ given an input of $x(n)$, or

$$x(n) \xrightarrow{\text{results in}} y(n) \ . \tag{1-23}$$

For a system to be time invariant, with a shifted version of the original $x(n)$ input applied, $x'(n)$, the following applies:

$$x'(n) = x(n+k) \xrightarrow{\text{results in}} y'(n) = y(n+k) \ , \tag{1-24}$$

where k is some integer representing k sample period time delays. For a system to be time invariant, expression (1-24) must hold true for any integer value of k and any input sequence.

1.6.1 Example of a Time-Invariant System

Let's look at a simple example of time invariance illustrated in Figure 1-9. Assume that our initial $x(n)$ input is a unity-amplitude 1-Hz sinewave sequence with a $y(n)$ output, as shown in Figure 1-9(b). Consider a different input sequence $x'(n)$, where

$$x'(n) = x(n+4) \ . \tag{1-25}$$

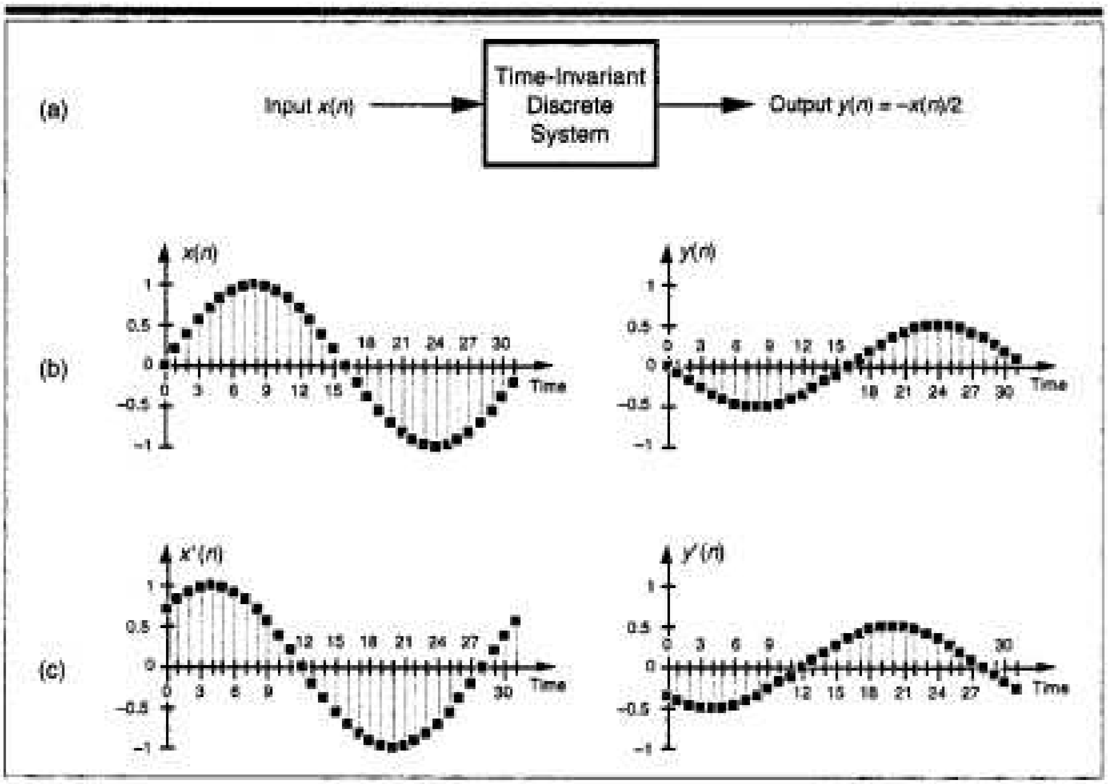

Figure 1-9 Time-invariant system input-to-output relationships: (a) system block diagram where $y(n) = -x(n)/2$; (b) system input and output with a 1-Hz sinewave applied; (c) system input and output when a 1-Hz sinewave, delayed by four samples, is applied. When $x'(n) = x(n+4)$, then, $y'(n) = y(n+4)$.

Equation (1-25) tells us that the input sequence $x'(n)$ is equal to sequence $x(n)$ shifted four samples to the left, that is, $x'(0) = x(4)$, $x'(1) = x(5)$, $x'(2) = x(6)$, and so on, as shown on the left of Figure 1-9(c). The discrete system is time invariant because the $y'(n)$ output sequence is equal to the $y(n)$ sequence shifted to the left by four samples, or $y'(n) = y(n+4)$. We can see that $y'(0) = y(4)$, $y'(1) = y(5)$, $y'(2) = y(6)$, and so on, as shown in Figure 1-9(c). For time-invariant systems, the y time shift is equal to the x time shift.

Some authors succumb to the urge to define a time-invariant system as one whose parameters do not change with time. That definition is incomplete and can get us in trouble if we're not careful. We'll just stick with the formal definition that a time-invariant system is one where a time shift in an input sequence results in an equal time shift in the output sequence. By the way, time-invariant systems in the literature are often called *shift-invariant* systems.†

† An example of a discrete process that's not time-invariant is the downsampling, or decimation, process described in Section 7.3.

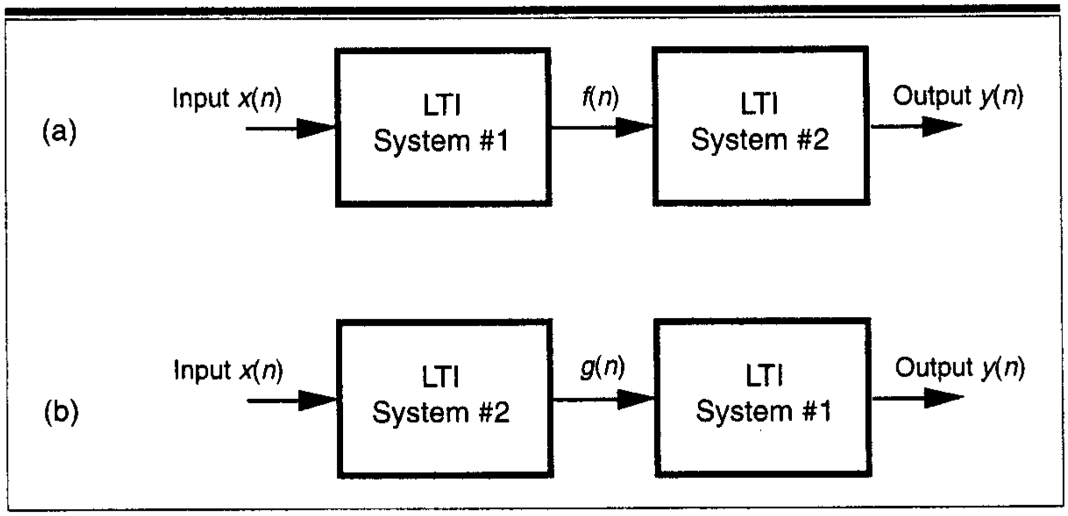

Figure 1-10 Linear time-invariant (LTI) systems in series: (a) block diagram of two LTI systems; (b) swapping the order of the two systems does not change the resultant output $y(n)$.

1.7 The Commutative Property of Linear Time-Invariant Systems

Although we don't substantiate this fact until we reach Section 6.8, it's not too early to realize that LTI systems have a useful commutative property by which their sequential order can be rearranged with no change in their final output. This situation is shown in Figure 1-10 where two different LTI systems are configured in series. Swapping the order of two cascaded systems does not alter the final output. Although the intermediate data sequences $f(n)$ and $g(n)$ will usually not be equal, the two pairs of LTI systems will have identical $y(n)$ output sequences. This commutative characteristic comes in handy for designers of digital filters, as we'll see in Chapters 5 and 6.

1.8 Analyzing Linear Time-Invariant Systems

As previously stated, LTI systems can be analyzed to predict their performance. Specifically, if we know the *unit impulse response* of an LTI system, we can calculate everything there is to know about the system; that is, the system's unit impulse response completely characterizes the system. By unit impulse response, we mean the system's time-domain output sequence when the input is a single unity-valued sample (unit impulse) preceded and followed by zero-valued samples as shown in Figure 1-11(b).

Knowing the (unit) impulse response of an LTI system, we can determine the system's output sequence for any input sequence because the output is equal to the *convolution* of the input sequence and the system's impulse response. Moreover, given an LTI system's time-domain impulse

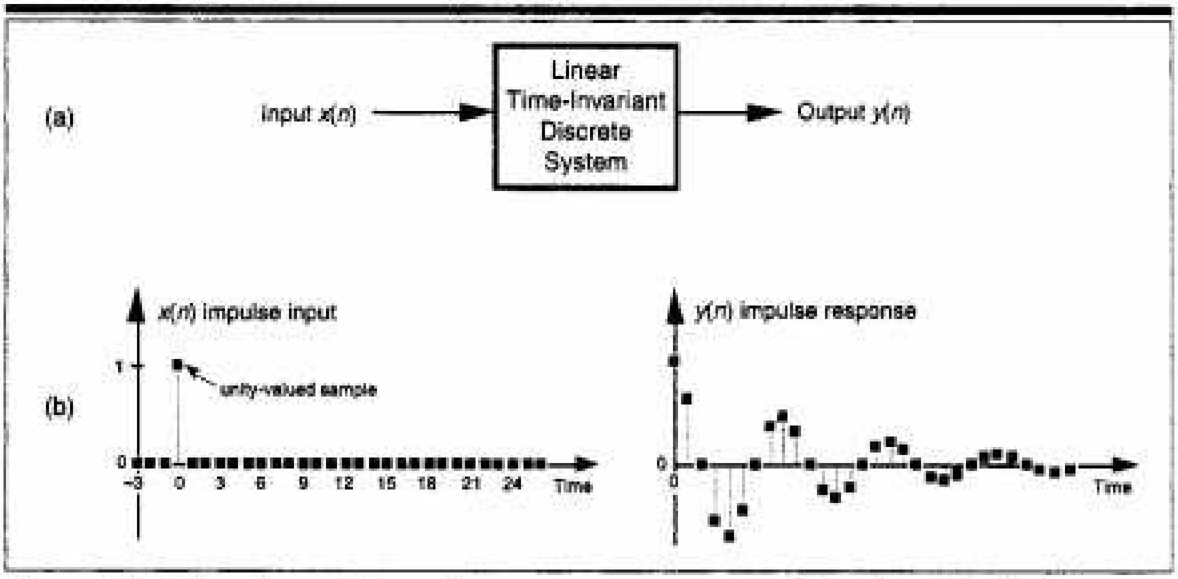

Figure 1-11 LTI system unit impulse response sequences: (a) system block diagram; (b) impulse input sequence $x(n)$ and impulse response output sequence $y(n)$.

response, we can find the system's *frequency response* by taking the Fourier transform in the form of a *discrete Fourier transform* of that impulse response[5].

Don't be alarmed if you're not exactly sure what is meant by convolution, frequency response, or the discrete Fourier transform. We'll introduce these subjects and define them slowly and carefully as we need them in later chapters. The point to keep in mind here is that LTI systems can be designed and analyzed using a number of straightforward and powerful analysis techniques. These techniques will become tools that we'll add to our signal processing toolboxes as we journey through the subject of digital signal processing.

References

[1] Karplus, W. J., and Soroka, W. W. *Analog Methods*, Second Edition, McGraw-Hill, New York, 1959, p. 117.

[2] Mikami, N., Kobayashi, M., and Yokoyama, Y. "A New DSP-Oriented Algorithm for Calculation of the Square Root Using a Nonlinear Digital Filter," *IEEE Trans. on Signal Processing*, Vol. 40, No. 7, July 1992.

[3] Heinen, P., and Neuvo, Y. "FIR-Median Hybrid Filters," *IEEE Trans. on Acoust. Speech, and Signal Processing*, Vol. ASSP-35, No. 6, June 1987.

[4] Oppenheim, A., Schafer, R., and Stockham, T. "Nonlinear Filtering of Multiplied and Convolved Signals," *Proc. IEEE*, Vol. 56, August 1968.

[5] Pickerd, John. "Impulse-Response Testing Lets a Single Test Do the Work of Thousands," *EDN*, April 27, 1995.