2.5 Implementation of Discrete-Time Systems

Our treatment of discrete-time systems has been focused on the time-domain characterization and analysis of linear time-invariant systems described by constant-coefficient linear difference equations. Additional analytical methods are developed in the next two chapters, where we characterize and analyze LTI systems in the frequency domain. Two other important topics that will be treated later are the design and implementation of these systems.

In practice, system design and implementation are usually treated jointly rather than separately. Often, the system design is driven by the method of implementation and by implementation constraints, such as cost, hardware limitations, size limitations, and power requirements. At this point, we have not as yet developed the necessary analysis and design tools to treat such complex issues. However, we have developed sufficient background to consider some basic implementation methods for realizations of LTI systems described by linear constant-coefficient difference equations.

2.5.1 Structures for the Realization of Linear Time-Invariant Systems

In this subsection we describe structures for the realization of systems described by linear constant-coefficient difference equations. Additional structures for these systems are introduced in Chapter 9.

As a beginning, let us consider the first-order system

$$y(n) = -a_1 y(n-1) + b_0 x(n) + b_1 x(n-1) \tag{2.5.1}$$

which is realized as in Fig. 2.5.1(a). This realization uses separate delays (memory) for both the input and output signal samples and it is called a *direct form I structure*. Note that this system can be viewed as two linear time-invariant systems in cascade. The first is a nonrecursive system described by the equation

$$v(n) = b_0 x(n) + b_1 x(n-1) \tag{2.5.2}$$

whereas the second is a recursive system described by the equation

$$y(n) = -a_1 y(n-1) + v(n) \tag{2.5.3}$$

However, as we have seen in Section 2.3.4, if we interchange the order of the cascaded linear time-invariant systems, the overall system response remains the same. Thus if we interchange the order of the recursive and nonrecursive systems, we obtain an alternative structure for the realization of the system described by (2.5.1). The resulting system is shown in Fig. 2.5.1(b). From this figure we obtain the two difference equations

$$w(n) = -a_1 w(n-1) + x(n) \tag{2.5.4}$$

$$y(n) = b_0 w(n) + b_1 w(n-1) \tag{2.5.5}$$

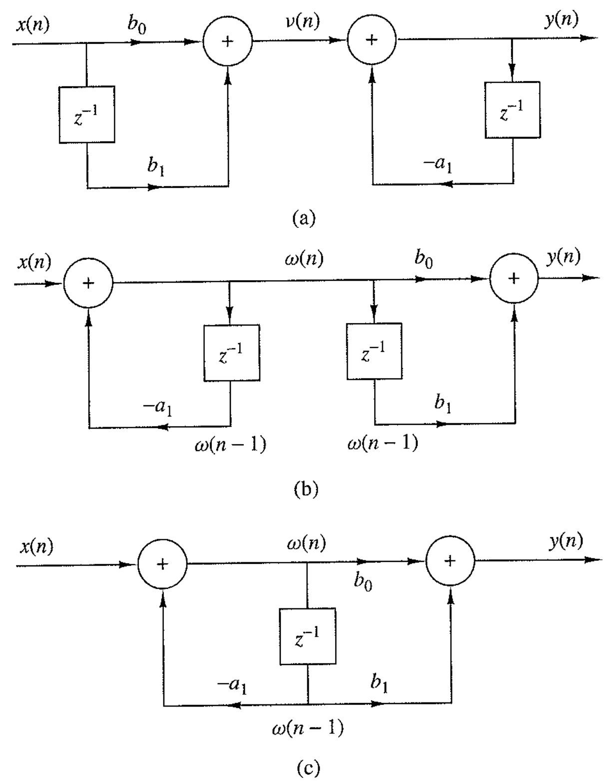

Figure 2.5.1
Steps in converting from the direct form I realization in (a) to the direct form II realization in (c).

which provide an alternative algorithm for computing the output of the system described by the single difference equation given in (2.5.1). In other words, the two difference equations (2.5.4) and (2.5.5) are equivalent to the single difference equation (2.5.1).

A close observation of Fig. 2.5.1 reveals that the two delay elements contain the same input $w(n)$ and hence the same output $w(n-1)$. Consequently, these two elements can be merged into one delay, as shown in Fig. 2.5.1(c). In contrast to the direct form I structure, this new realization requires only one delay for the auxiliary quantity $w(n)$, and hence it is more efficient in terms of memory requirements. It is called the *direct form II structure* and it is used extensively in practical applications.

These structures can readily be generalized for the general linear time-invariant recursive system described by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{2.5.6}$$

Figure 2.5.2 illustrates the direct form I structure for this system. This structure requires $M + N$ delays and $N + M + 1$ multiplications. It can be viewed as the cascade of a nonrecursive system

$$v(n) = \sum_{k=0}^{M} b_k x(n-k) \qquad (2.5.7)$$

and a recursive system

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + v(n) \qquad (2.5.8)$$

By reversing the order of these two systems, as was previously done for the first-order system, we obtain the direct form II structure shown in Fig. 2.5.3 for $N > M$. This structure is the cascade of a recursive system

$$w(n) = -\sum_{k=1}^{N} a_k w(n-k) + x(n) \qquad (2.5.9)$$

followed by a nonrecursive system

$$y(n) = \sum_{k=0}^{M} b_k w(n-k) \qquad (2.5.10)$$

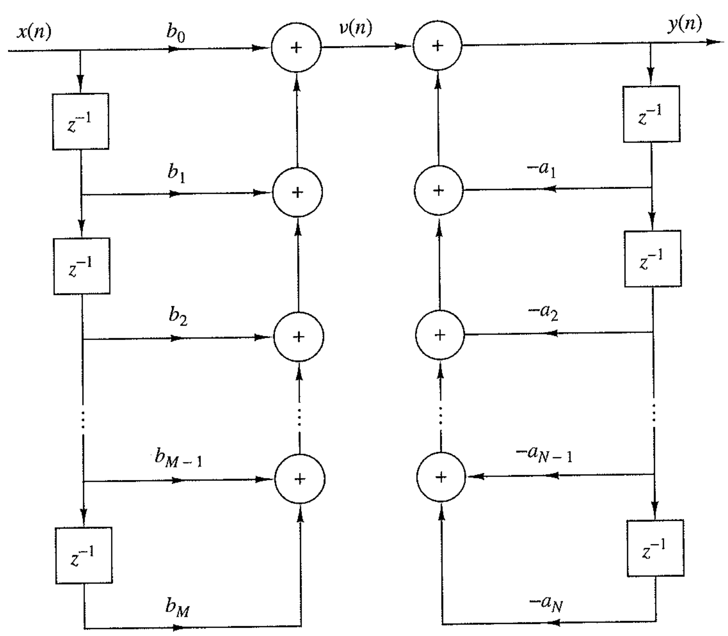

Figure 2.5.2 Direct form I structure of the system described by (2.5.6).

We observe that if $N \geq M$, this structure requires a number of delays equal to the order N of the system. However, if $M > N$, the required memory is specified by M. Figure 2.5.3 can easily be modified to handle this case. Thus the direct form II structure requires $M + N + 1$ multiplications and $\max\{M, N\}$ delays. Because it requires the minimum number of delays for the realization of the system described by (2.5.6), it is sometimes called a *canonic form.*

A special case of (2.5.6) occurs if we set the system parameters $a_k = 0$, $k = 1, \ldots, N$. Then the input–output relationship for the system reduces to

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) \tag{2.5.11}$$

which is a nonrecursive linear time-invariant system. This system views only the most recent $M + 1$ input signal samples and, prior to addition, weights each sample by the appropriate coefficient b_k from the set $\{b_k\}$. In other words, the system output is basically a *weighted moving average* of the input signal. For this reason it is sometimes called a *moving average (MA) system.* Such a system is an FIR system

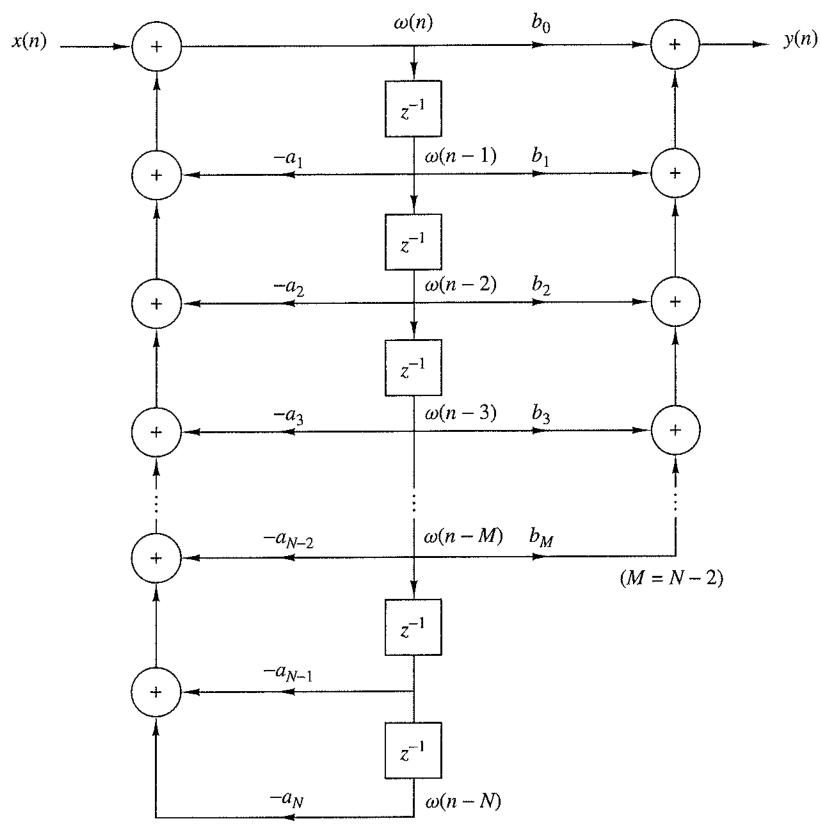

Figure 2.5.3 Direct form II structure for the system described by (2.5.6).

with an impulse response $h(k)$ equal to the coefficients b_k, that is,

$$h(k) = \begin{cases} b_k, & 0 \le k \le M \\ 0, & \text{otherwise} \end{cases} \tag{2.5.12}$$

If we return to (2.5.6) and set $M = 0$, the general linear time-invariant system reduces to a "purely recursive" system described by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + b_0 x(n) \tag{2.5.13}$$

In this case the system output is a weighted linear combination of N past outputs and the present input.

Linear time-invariant systems described by a second-order difference equation are an important subclass of the more general systems described by (2.5.6) or (2.5.10) or (2.5.13). The reason for their importance will be explained later when we discuss quantization effects. Suffice to say at this point that second-order systems are usually used as basic building blocks for realizing higher-order systems.

The most general second-order system is described by the difference equation

$$\begin{aligned} y(n) = &- a_1 y(n-1) - a_2 y(n-2) + b_0 x(n) \\ &+ b_1 x(n-1) + b_2 x(n-2) \end{aligned} \tag{2.5.14}$$

which is obtained from (2.5.6) by setting $N = 2$ and $M = 2$. The direct form II structure for realizing this system is shown in Fig. 2.5.4(a). If we set $a_1 = a_2 = 0$, then (2.5.14) reduces to

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) \tag{2.5.15}$$

which is a special case of the FIR system described by (2.5.11). The structure for realizing this system is shown in Fig. 2.5.4(b). Finally, if we set $b_1 = b_2 = 0$ in (2.5.14), we obtain the purely recursive second-order system described by the difference equation

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) + b_0 x(n) \tag{2.5.16}$$

which is a special case of (2.5.13). The structure for realizing this system is shown in Fig. 2.5.4(c).

2.5.2 Recursive and Nonrecursive Realizations of FIR Systems

We have already made the distinction between FIR and IIR systems, based on whether the impulse response $h(n)$ of the system has a finite duration, or an infinite duration. We have also made the distinction between recursive and nonrecursive systems. Basically, a causal recursive system is described by an input–output equation of the form

$$y(n) = F[y(n-1), \ldots, y(n-N), x(n), \ldots, x(n-M)] \tag{2.5.17}$$

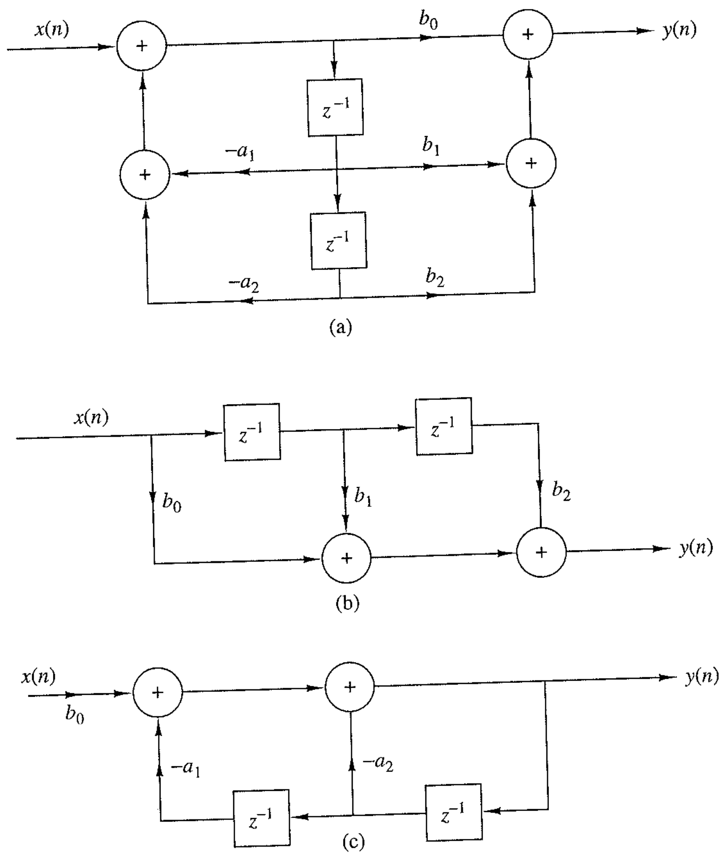

Figure 2.5.4 Structures for the realization of second-order systems: (a) general second-order system; (b) FIR system; (c) "purely recursive system."

and for a linear time-invariant system specifically, by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{2.5.18}$$

On the other hand, causal nonrecursive systems do not depend on past values of the output and hence are described by an input–output equation of the form

$$y(n) = F[x(n), x(n-1), \ldots, x(n-M)] \tag{2.5.19}$$

and for linear time-invariant systems specifically, by the difference equation in (2.5.18) with $a_k = 0$ for $k = 1, 2, \ldots, N$.

In the case of FIR systems, we have already observed that it is always possible to realize such systems nonrecursively. In fact, with $a_k = 0$, $k = 1, 2, \ldots, N$, in (2.5.18),

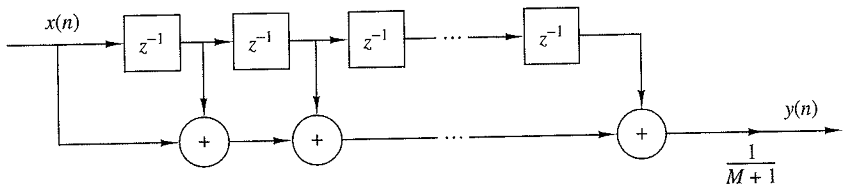

Figure 2.5.5 Nonrecursive realization of an FIR moving average system.

we have a system with an input–output equation

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) \tag{2.5.20}$$

This is a nonrecursive and FIR system. As indicated in (2.5.12), the impulse response of the system is simply equal to the coefficients $\{b_k\}$. Hence every FIR system can be realized nonrecursively. On the other hand, any FIR system can also be realized recursively. Although the general proof of this statement is given later, we shall give a simple example to illustrate the point.

Suppose that we have an FIR system of the form

$$y(n) = \frac{1}{M+1} \sum_{k=0}^{M} x(n-k) \tag{2.5.21}$$

for computing the *moving average* of a signal $x(n)$. Clearly, this system is FIR with impulse response

$$h(n) = \frac{1}{M+1}, \qquad 0 \le n \le M$$

Figure 2.5.5 illustrates the structure of the nonrecursive realization of the system. Now, suppose that we express (2.5.21) as

$$\begin{aligned} y(n) &= \frac{1}{M+1} \sum_{k=0}^{M} x(n-1-k) \\ &\quad + \frac{1}{M+1}[x(n) - x(n-1-M)] \\ &= y(n-1) + \frac{1}{M+1}[x(n) - x(n-1-M)] \end{aligned} \tag{2.5.22}$$

Now, (2.5.22) represents a recursive realization of the FIR system. The structure of this recursive realization of the moving average system is illustrated in Fig. 2.5.6.

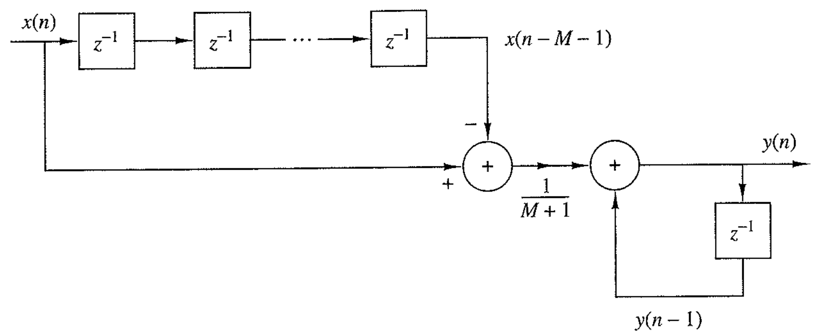

Figure 2.5.6 Recursive realization of an FIR moving average system.

In summary, we can think of the terms FIR and IIR as general characteristics that distinguish a type of linear time-invariant system, and of the terms *recursive* and *nonrecursive* as descriptions of the structures for realizing or implementing the system.

2.6 Correlation of Discrete-Time Signals

A mathematical operation that closely resembles convolution is correlation. Just as in the case of convolution, two signal sequences are involved in correlation. In contrast to convolution, however, our objective in computing the correlation between the two signals is to measure the degree to which the two signals are similar and thus to extract some information that depends to a large extent on the application. Correlation of signals is often encountered in radar, sonar, digital communications, geology, and other areas in science and engineering.

To be specific, let us suppose that we have two signal sequences $x(n)$ and $y(n)$ that we wish to compare. In radar and active sonar applications, $x(n)$ can represent the sampled version of the transmitted signal and $y(n)$ can represent the sampled version of the received signal at the output of the analog-to-digital (A/D) converter. If a target is present in the space being searched by the radar or sonar, the received signal $y(n)$ consists of a delayed version of the transmitted signal, reflected from the target, and corrupted by additive noise. Figure 2.6.1 depicts the radar signal reception problem.

We can represent the received signal sequence as

$$y(n) = \alpha x(n - D) + w(n) \tag{2.6.1}$$

where α is some attenuation factor representing the signal loss involved in the round-trip transmission of the signal $x(n)$, D is the round-trip delay, which is assumed to be an integer multiple of the sampling interval, and $w(n)$ represents the additive noise that is picked up by the antenna and any noise generated by the electronic components and amplifiers contained in the front end of the receiver. On the other hand, if there is no target in the space searched by the radar and sonar, the received signal $y(n)$ consists of noise alone.

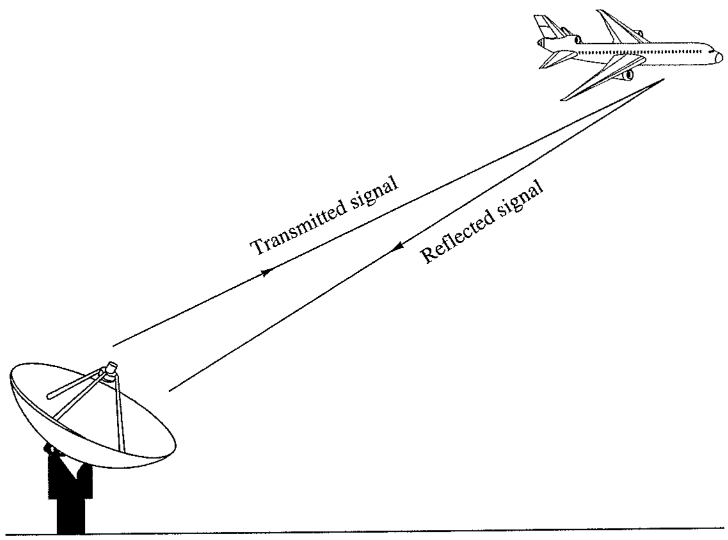

Figure 2.6.1 Radar target detection.

Having the two signal sequences, $x(n)$, which is called the reference signal or transmitted signal, and $y(n)$, the received signal, the problem in radar and sonar detection is to compare $y(n)$ and $x(n)$ to determine if a target is present and, if so, to determine the time delay D and compute the distance to the target. In practice, the signal $x(n-D)$ is heavily corrupted by the additive noise to the point where a visual inspection of $y(n)$ does not reveal the presence or absence of the desired signal reflected from the target. Correlation provides us with a means for extracting this important information from $y(n)$.

Digital communications is another area where correlation is often used. In digital communications the information to be transmitted from one point to another is usually converted to binary form, that is, a sequence of zeros and ones, which are then transmitted to the intended receiver. To transmit a 0 we can transmit the signal sequence $x_0(n)$ for $0 \leq n \leq L-1$, and to transmit a 1 we can transmit the signal sequence $x_1(n)$ for $0 \leq n \leq L-1$, where L is some integer that denotes the number of samples in each of the two sequences. Very often, $x_1(n)$ is selected to be the negative of $x_0(n)$. The signal received by the intended receiver may be represented as

$$y(n) = x_i(n) + w(n), \qquad i = 0, 1, \qquad 0 \leq n \leq L-1 \tag{2.6.2}$$

where now the uncertainty is whether $x_0(n)$ or $x_1(n)$ is the signal component in $y(n)$, and $w(n)$ represents the additive noise and other interference inherent in any communication system. Again, such noise has its origin in the electronic components contained in the front end of the receiver. In any case, the receiver knows the possible transmitted sequences $x_0(n)$ and $x_1(n)$ and is faced with the task of comparing the received signal $y(n)$ with both $x_0(n)$ and $x_1(n)$ to determine which of the two signals better matches $y(n)$. This comparison process is performed by means of the correlation operation described in the following subsection.

2.6.1 Crosscorrelation and Autocorrelation Sequences

Suppose that we have two real signal sequences $x(n)$ and $y(n)$ each of which has finite energy. The *crosscorrelation* of $x(n)$ and $y(n)$ is a sequence $r_{xy}(l)$, which is defined as

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l), \qquad l = 0, \pm 1, \pm 2, \ldots \tag{2.6.3}$$

or, equivalently, as

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n+l)y(n), \qquad l = 0, \pm 1, \pm 2, \ldots \tag{2.6.4}$$

The index l is the (time) shift (or *lag*) parameter and the subscripts xy on the crosscorrelation sequence $r_{xy}(l)$ indicate the sequences being correlated. The order of the subscripts, with x preceding y, indicates the direction in which one sequence is shifted, relative to the other. To elaborate, in (2.6.3), the sequence $x(n)$ is left unshifted and $y(n)$ is shifted by l units in time, to the right for l positive and to the left for l negative. Equivalently, in (2.6.4), the sequence $y(n)$ is left unshifted and $x(n)$ is shifted by l units in time, to the left for l positive and to the right for l negative. But shifting $x(n)$ to the left by l units relative to $y(n)$ is equivalent to shifting $y(n)$ to the right by l units relative to $x(n)$. Hence the computations (2.6.3) and (2.6.4) yield identical crosscorrelation sequences.

If we reverse the roles of $x(n)$ and $y(n)$ in (2.6.3) and (2.6.4) and therefore reverse the order of the indices xy, we obtain the crosscorrelation sequence

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y(n)x(n-l) \tag{2.6.5}$$

or, equivalently,

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y(n+l)x(n) \tag{2.6.6}$$

By comparing (2.6.3) with (2.6.6) or (2.6.4) with (2.6.5), we conclude that

$$r_{xy}(l) = r_{yx}(-l) \tag{2.6.7}$$

Therefore, $r_{yx}(l)$ is simply the folded version of $r_{xy}(l)$, where the folding is done with respect to $l = 0$. Hence, $r_{yx}(l)$ provides exactly the same information as $r_{xy}(l)$, with respect to the similarity of $x(n)$ to $y(n)$.

EXAMPLE 2.6.1

Determine the crosscorrelation sequence $r_{xy}(l)$ of the sequences

$$x(n) = \{\ldots, 0, 0, 2, -1, 3, 7, \underset{\uparrow}{1}, 2, -3, 0, 0, \ldots\}$$

$$y(n) = \{\ldots, 0, 0, 1, -1, 2, -2, \underset{\uparrow}{4}, 1, -2, 5, 0, 0, \ldots\}$$

Solution. Let us use the definition in (2.6.3) to compute $r_{xy}(l)$. For $l = 0$ we have

$$r_{xy}(0) = \sum_{n=-\infty}^{\infty} x(n)y(n)$$

The product sequence $v_0(n) = x(n)y(n)$ is

$$v_0(n) = \{\ldots, 0, 0, 2, 1, 6, -14, \underset{\uparrow}{4}, 2, 6, 0, 0, \ldots\}$$

and hence the sum over all values of n is

$$r_{xy}(0) = 7$$

For $l > 0$, we simply shift $y(n)$ to the right relative to $x(n)$ by l units, compute the product sequence $v_l(n) = x(n)y(n-l)$, and finally, sum over all values of the product sequence. Thus we obtain

$$\begin{array}{llll} r_{xy}(1) = 13, & r_{xy}(2) = -18, & r_{xy}(3) = 16, & r_{xy}(4) = -7 \\ r_{xy}(5) = 5, & r_{xy}(6) = -3, & r_{xy}(l) = 0, & l \geq 7 \end{array}$$

For $l < 0$, we shift $y(n)$ to the left relative to $x(n)$ by l units, compute the product sequence $v_l(n) = x(n)y(n-l)$, and sum over all values of the product sequence. Thus we obtain the values of the crosscorrelation sequence

$$\begin{array}{llll} r_{xy}(-1) = 0, & r_{xy}(-2) = 33, & r_{xy}(-3) = -14, & r_{xy}(-4) = 36 \\ r_{xy}(-5) = 19, & r_{xy}(-6) = -9, & r_{xy}(-7) = 10, & r_{xy}(l) = 0, \; l \leq -8 \end{array}$$

Therefore, the crosscorrelation sequence of $x(n)$ and $y(n)$ is

$$r_{xy}(l) = \{10, -9, 19, 36, -14, 33, 0, \underset{\uparrow}{7}, 13, -18, 16, -7, 5, -3\}$$

The similarities between the computation of the crosscorrelation of two sequences and the convolution of two sequences is apparent. In the computation of convolution, one of the sequences is folded, then shifted, then multiplied by the other sequence to form the product sequence for that shift, and finally, the values of the product sequence are summed. Except for the folding operation, the computation of the crosscorrelation sequence involves the same operations: shifting one of the sequences, multiplying the two sequences, and summing over all values of the product sequence. Consequently, if we have a computer program that performs convolution, we can use it to perform crosscorrelation by providing as inputs to the program the sequence $x(n)$ and the folded sequence $y(-n)$. Then the convolution of $x(n)$ with $y(-n)$ yields the crosscorrelation $r_{xy}(l)$, that is,

$$r_{xy}(l) = x(l) * y(-l) \tag{2.6.8}$$

We note that the absence of folding makes crosscorrelation a noncommutative operation. In the special case where $y(n) = x(n)$, we have the *autocorrelation* of $x(n)$, which is defined as the sequence

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n-l) \tag{2.6.9}$$

or, equivalently, as

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n+l)x(n) \tag{2.6.10}$$

In dealing with finite-duration sequences, it is customary to express the autocorrelation and crosscorrelation in terms of the finite limits on the summation. In particular, if $x(n)$ and $y(n)$ are causal sequences of length N [i.e., $x(n) = y(n) = 0$ for $n < 0$ and $n \geq N$], the crosscorrelation and autocorrelation sequences may be expressed as

$$r_{xy}(l) = \sum_{n=l}^{N-|k|-1} x(n)y(n-l) \tag{2.6.11}$$

and

$$r_{xx}(l) = \sum_{n=i}^{N-|k|-1} x(n)x(n-l) \tag{2.6.12}$$

where $i = l$, $k = 0$ for $l \geq 0$, and $i = 0$, $k = l$ for $l < 0$.

2.6.2 Properties of the Autocorrelation and Crosscorrelation Sequences

The autocorrelation and crosscorrelation sequences have a number of important properties that we now present. To develop these properties, let us assume that we have two sequences $x(n)$ and $y(n)$ with finite energy from which we form the linear combination,

$$ax(n) + by(n-l)$$

where a and b are arbitrary constants and l is some time shift. The energy in this signal is

$$\begin{aligned}\sum_{n=-\infty}^{\infty} [ax(n) + by(n-l)]^2 &= a^2 \sum_{n=-\infty}^{\infty} x^2(n) + b^2 \sum_{n=-\infty}^{\infty} y^2(n-l) \\ &\quad + 2ab \sum_{n=-\infty}^{\infty} x(n)y(n-l) \\ &= a^2 r_{xx}(0) + b^2 r_{yy}(0) + 2ab r_{xy}(l)\end{aligned} \tag{2.6.13}$$

First, we note that $r_{xx}(0) = E_x$ and $r_{yy}(0) = E_y$, which are the energies of $x(n)$ and $y(n)$, respectively. It is obvious that

$$a^2 r_{xx}(0) + b^2 r_{yy}(0) + 2ab r_{xy}(l) \geq 0 \tag{2.6.14}$$

Now, assuming that $b \neq 0$, we can divide (2.6.14) by b^2 to obtain

$$r_{xx}(0)\left(\frac{a}{b}\right)2 + 2r_{xy}(l)\left(\frac{a}{b}\right) + r_{yy}(0) \geq 0$$

We view this equation as a quadratic with coefficients $r_{xx}(0)$, $2r_{xy}(l)$, and $r_{yy}(0)$. Since the quadratic is nonnegative, it follows that the discriminant of this quadratic must be nonpositive, that is,

$$4[r_{xy}^2(l) - r_{xx}(0)r_{yy}(0)] \leq 0$$

Therefore, the crosscorrelation sequence satisfies the condition that

$$|r_{xy}(l)| \leq \sqrt{r_{xx}(0)r_{yy}(0)} = \sqrt{E_x E_y} \tag{2.6.15}$$

In the special case where $y(n) = x(n)$, (2.6.15) reduces to

$$|r_{xx}(l)| \leq r_{xx}(0) = E_x \tag{2.6.16}$$

This means that the autocorrelation sequence of a signal attains its maximum value at zero lag. This result is consistent with the notion that a signal matches perfectly with itself at zero shift. In the case of the crosscorrelation sequence, the upper bound on its values is given in (2.6.15).

Note that if any one or both of the signals involved in the crosscorrelation are scaled, the shape of the crosscorrelation sequence does not change; only the amplitudes of the crosscorrelation sequence are scaled accordingly. Since scaling is unimportant, it is often desirable, in practice, to normalize the autocorrelation and crosscorrelation sequences to the range from -1 to 1. In the case of the autocorrelation sequence, we can simply divide by $r_{xx}(0)$. Thus the normalized autocorrelation sequence is defined as

$$\rho_{xx}(l) = \frac{r_{xx}(l)}{r_{xx}(0)} \tag{2.6.17}$$

Similarly, we define the normalized crosscorrelation sequence

$$\rho_{xy}(l) = \frac{r_{xy}(l)}{\sqrt{r_{xx}(0)r_{yy}(0)}} \tag{2.6.18}$$

Now $|\rho_{xx}(l)| \leq 1$ and $|\rho_{xy}(l)| \leq 1$, and hence these sequences are independent of signal scaling.

Finally, as we have already demonstrated, the crosscorrelation sequence satisfies the property

$$r_{xy}(l) = r_{yx}(-l)$$

With $y(n) = x(n)$, this relation results in the following important property for the autocorrelation sequence

$$r_{xx}(l) = r_{xx}(-l) \tag{2.6.19}$$

Hence the autocorrelation function is an even function. Consequently, it suffices to compute $r_{xx}(l)$ for $l \geq 0$.

EXAMPLE 2.6.2

Compute the autocorrelation of the signal

$$x(n) = a^n u(n), 0 < a < 1$$

Solution. Since $x(n)$ is an infinite-duration signal, its autocorrelation also has infinite duration. We distinguish two cases.

If $l \geq 0$, from Fig. 2.6.2 we observe that

$$r_{xx}(l) = \sum_{n=l}^{\infty} x(n)x(n-l) = \sum_{n=l}^{\infty} a^n a^{n-l} = a^{-l} \sum_{n=l}^{\infty} (a^2)^n$$

Since $a < 1$, the infinite series *converges* and we obtain

$$r_{xx}(l) = \frac{1}{1-a^2} a^{|l|}, \qquad l \geq 0$$

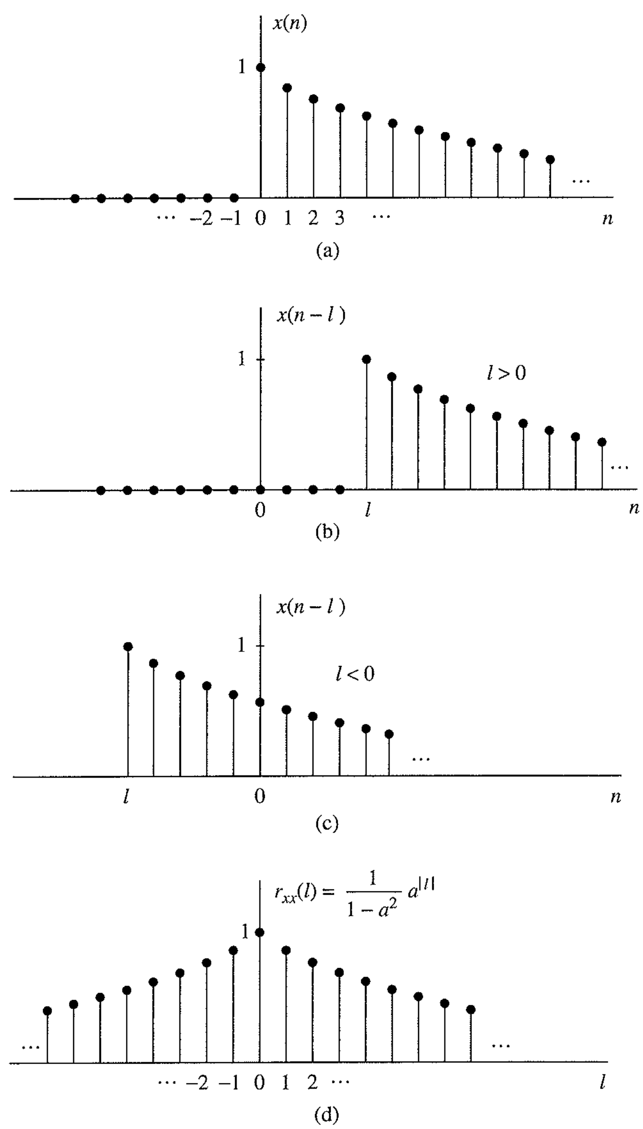

Figure 2.6.2 Computation of the autocorrelation of the signal $x(n) = a^n$, $0 < a < 1$.

For $l < 0$ we have

$$r_{xx}(l) = \sum_{n=0}^{\infty} x(n)x(n-l) = a^{-l}\sum_{n=0}^{\infty}(a^2)^n = \frac{1}{1-a^2}a^{-l}, \qquad l < 0$$

But when l is negative, $a^{-l} = a^{|l|}$. Thus the two relations for $r_{xx}(l)$ can be combined into the following expression:

$$r_{xx}(l) = \frac{1}{1-a^2}a^{|l|}, \qquad -\infty < l < \infty \tag{2.6.20}$$

The sequence $r_{xx}(l)$ is shown in Fig. 2.6.2(d). We observe that

$$r_{xx}(-l) = r_{xx}(l)$$

and

$$r_{xx}(0) = \frac{1}{1-a^2}$$

Therefore, the normalized autocorrelation sequence is

$$\rho_{xx}(l) = \frac{r_{xx}(l)}{r_{xx}(0)} = a^{|l|}, \qquad -\infty < l < \infty \tag{2.6.21}$$

2.6.3 Correlation of Periodic Sequences

In Section 2.6.1 we defined the crosscorrelation and autocorrelation sequences of energy signals. In this section we consider the correlation sequences of power signals and, in particular, periodic signals.

Let $x(n)$ and $y(n)$ be two power signals. Their crosscorrelation sequence is defined as

$$r_{xy}(l) = \lim_{M\to\infty}\frac{1}{2M+1}\sum_{n=-M}^{M} x(n)y(n-l) \tag{2.6.22}$$

If $x(n) = y(n)$, we have the definition of the autocorrelation sequence of a power signal as

$$r_{xx}(l) = \lim_{M\to\infty}\frac{1}{2M+1}\sum_{n=-M}^{M} x(n)x(n-l) \tag{2.6.23}$$

In particular, if $x(n)$ and $y(n)$ are two periodic sequences, each with period N, the averages indicated in (2.6.22) and (2.6.23) over the infinite interval are identical to the averages over a single period, so that (2.6.22) and (2.6.23) reduce to

$$r_{xy}(l) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)y(n-l) \tag{2.6.24}$$

and

$$r_{xx}(l) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n-l) \tag{2.6.25}$$

It is clear that $r_{xy}(l)$ and $r_{xx}(l)$ are periodic correlation sequences with period N. The factor $1/N$ can be viewed as a normalization scale factor.

In some practical applications, correlation is used to identify periodicities in an observed physical signal which may be corrupted by random interference. For example, consider a signal sequence $y(n)$ of the form

$$y(n) = x(n) + w(n) \tag{2.6.26}$$

where $x(n)$ is a periodic sequence of some unknown period N and $w(n)$ represents an additive random interference. Suppose that we observe M samples of $y(n)$, say $0 \le n \le M-1$, where $M >> N$. For all practical purposes, we can assume that $y(n) = 0$ for $n < 0$ and $n \ge M$. Now the autocorrelation sequence of $y(n)$, using the normalization factor of $1/M$, is

$$r_{yy}(l) = \frac{1}{M}\sum_{n=0}^{M-1} y(n)y(n-l) \tag{2.6.27}$$

If we substitute for $y(n)$ from (2.6.26) into (2.6.27) we obtain

$$\begin{aligned} r_{yy}(l) &= \frac{1}{M}\sum_{n=0}^{M-1}[x(n)+w(n)][x(n-l)+w(n-l)] \\ &= \frac{1}{M}\sum_{n=0}^{M-1} x(n)x(n-l) \\ &\quad + \frac{1}{M}\sum_{n=0}^{M-1}[x(n)w(n-l)+w(n)x(n-l)] \\ &\quad + \frac{1}{M}\sum_{n=0}^{M-1} w(n)w(n-l) \\ &= r_{xx}(l) + r_{xw}(l) + r_{wx}(l) + r_{ww}(l) \end{aligned} \tag{2.6.28}$$

The first factor on the right-hand side of (2.6.28) is the autocorrelation sequence of $x(n)$. Since $x(n)$ is periodic, its autocorrelation sequence exhibits the same periodicity, thus containing relatively large peaks at $l = 0, N, 2N$, and so on. However, as the shift l approaches M, the peaks are reduced in amplitude due to the fact that we have a finite data record of M samples so that many of the products $x(n)x(n-l)$ are zero. Consequently, we should avoid computing $r_{yy}(l)$ for large lags, say, $l > M/2$.

The crosscorrelations $r_{xw}(l)$ and $r_{wx}(l)$ between the signal $x(n)$ and the additive random interference are expected to be relatively small as a result of the expectation that $x(n)$ and $w(n)$ will be totally unrelated. Finally, the last term on the right-hand side of (2.6.28) is the autocorrelation sequence of the random sequence $w(n)$. This correlation sequence will certainly contain a peak at $l = 0$, but because of its random characteristics, $r_{ww}(l)$ is expected to decay rapidly toward zero. Consequently, only $r_{xx}(l)$ is expected to have large peaks for $l > 0$. This behavior allows us to detect the presence of the periodic signal $x(n)$ buried in the interference $w(n)$ and to identify its period.

An example that illustrates the use of autocorrelation to identify a hidden periodicity in an observed physical signal is shown in Fig. 2.6.3. This figure illustrates the autocorrelation (normalized) sequence for the Wölfer sunspot numbers in the 100-year period 1770–1869 for $0 \le l \le 20$, where any value of l corresponds to one year. There is clear evidence in this figure that a periodic trend exists, with a period of 10 to 11 years.

EXAMPLE 2.6.3

Suppose that a signal sequence $x(n) = \sin(\pi/5)n$, for $0 \le n \le 99$ is corrupted by an additive noise sequence $w(n)$, where the values of the additive noise are selected independently from sample to sample, from a uniform distribution over the range $(-\Delta/2, \Delta/2)$, where Δ is a parameter of the distribution. The observed sequence is $y(n) = x(n) + w(n)$. Determine the autocorrelation sequence $r_{yy}(l)$ and thus determine the period of the signal $x(n)$.

Solution. The assumption is that the signal sequence $x(n)$ has some unknown period that we are attempting to determine from the noise-corrupted observations $\{y(n)\}$. Although $x(n)$ is periodic with period 10, we have only a finite-duration sequence of length $M = 100$ [i.e., 10 periods of $x(n)$]. The noise power level P_w in the sequence $w(n)$ is determined by the parameter Δ. We simply state that $P_w = \Delta^2/12$. The signal power level is $P_x = \frac{1}{2}$. Therefore, the signal-to-noise ratio (SNR) is defined as

$$\frac{P_x}{P_w} = \frac{\frac{1}{2}}{\Delta^2/12} = \frac{6}{\Delta^2}$$

Usually, the SNR is expressed on a logarithmic scale in decibels (dB) as $10\log_{10}(P_x/P_w)$.

Figure 2.6.4 illustrates a sample of a noise sequence $w(n)$, and the observed sequence $y(n) = x(n) + w(n)$ when the SNR $= 1$ dB. The autocorrelation sequence $r_{yy}(l)$ is illustrated in Fig. 2.6.4(c). We observe that the periodic signal $x(n)$, embedded in $y(n)$, results in a periodic autocorrelation function $r_{xx}(l)$ with period $N = 10$. The effect of the additive noise is to add to the peak value at $l = 0$, but for $l \neq 0$, the correlation sequence $r_{ww}(l) \approx 0$ as a result of the fact that values of $w(n)$ were generated independently. Such noise is usually called *white noise*. The presence of this noise explains the reason for the large peak at $l = 0$. The smaller, nearly equal peaks at $l = \pm 10, \pm 20, \ldots$ are due to the periodic characteristics of $x(n)$.

2.6.4 Input–Output Correlation Sequences

In this section we derive two input–output relationships for LTI systems in the "correlation domain." Let us assume that a signal $x(n)$ with known autocorrelation $r_{xx}(l)$

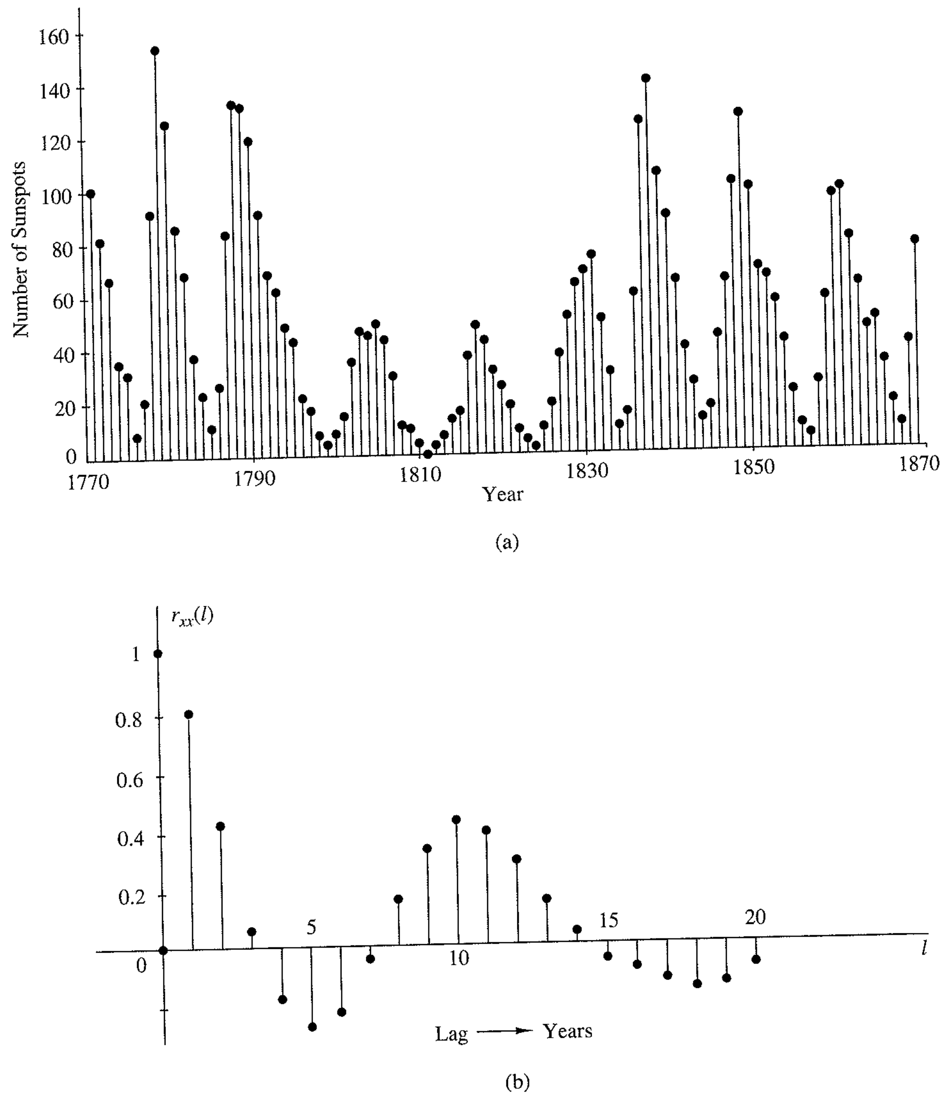

Figure 2.6.3 Identification of periodicity in the Wölfer sunspot numbers: (a) annual Wölfer sunspot numbers; (b) normalized autocorrelation sequence.

is applied to an LTI system with impulse response $h(n)$, producing the output signal

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

The crosscorrelation between the output and the input signal is

$$r_{yx}(l) = y(l) * x(-l) = h(l) * [x(l) * x(-l)]$$

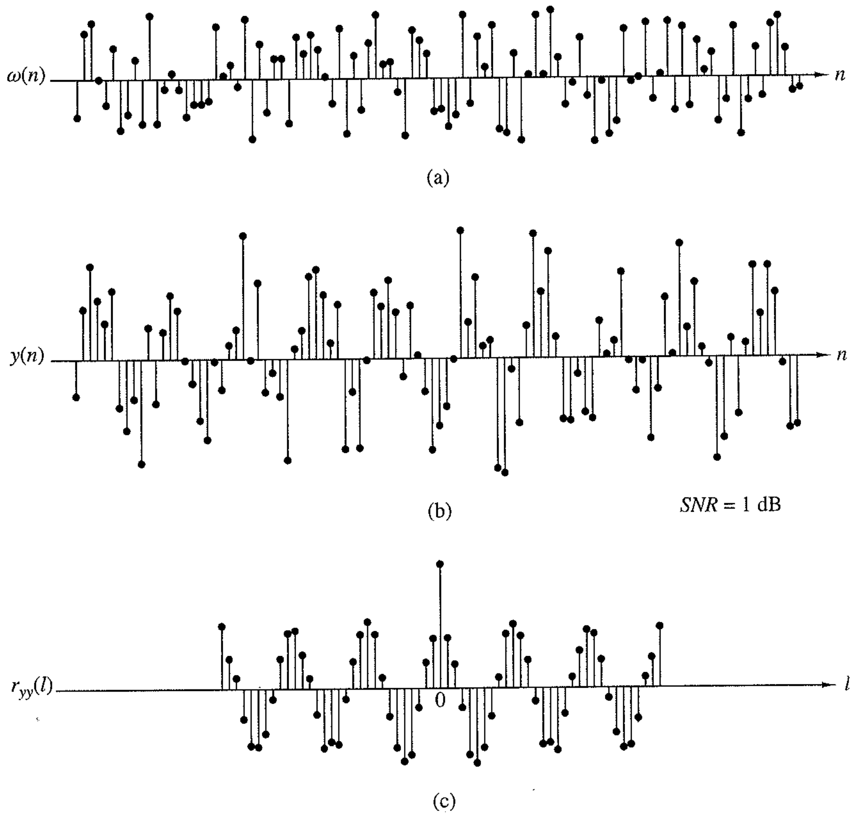

Figure 2.6.4 Use of autocorrelation to detect the presence of a periodic signal corrupted by noise.

or

$$r_{yx}(l) = h(l) * r_{xx}(l) \tag{2.6.29}$$

where we have used (2.6.8) and the properties of convolution. Hence the cross-correlation between the input and the output of the system is the convolution of the impulse response with the autocorrelation of the input sequence. Alternatively, $r_{yx}(l)$ may be viewed as the output of the LTI system when the input sequence is $r_{xx}(l)$. This is illustrated in Fig. 2.6.5. If we replace l by $-l$ in (2.6.29), we obtain

$$r_{xy}(l) = h(-l) * r_{xx}(l)$$

The autocorrelation of the output signal can be obtained by using (2.6.8) with $x(n) = y(n)$ and the properties of convolution. Thus we have

$$\begin{aligned} r_{yy}(l) &= y(l) * y(-l) \\ &= [h(l) * x(l)] * [h(-l) * x(-l)] \\ &= [h(l) * h(-l)] * [x(l) * x(-l)] \\ &= r_{hh}(l) * r_{xx}(l) \end{aligned} \tag{2.6.30}$$

The autocorrelation $r_{hh}(l)$ of the impulse response $h(n)$ exists if the system is stable. Furthermore, the stability insures that the system does not change the type (energy or power) of the input signal. By evaluating (2.6.30) for $l = 0$ we obtain

$$r_{yy}(0) = \sum_{k=-\infty}^{\infty} r_{hh}(k) r_{xx}(k) \tag{2.6.31}$$

which provides the energy (or power) of the output signal in terms of autocorrelations. These relationships hold for both energy and power signals. The direct derivation of these relationships for energy and power signals, and their extensions to complex signals, are left as exercises for the student.

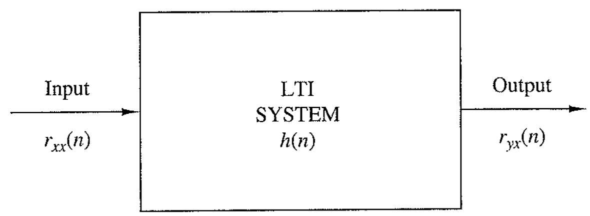

Figure 2.6.5 Input–output relation for crosscorrelation $r_{yx}(n)$.

2.7 Summary and References

The major theme of this chapter is the characterization of discrete-time signals and systems in the time domain. Of particular importance is the class of linear time-invariant (LTI) systems which are widely used in the design and implementation of digital signal processing systems. We characterized LTI systems by their unit sample response $h(n)$ and derived the convolution summation, which is a formula for determining the response $y(n)$ of the system characterized by $h(n)$ to any given input sequence $x(n)$.

The class of LTI systems characterized by linear difference equations with constant coefficients is by far the most important of the LTI systems in the theory and application of digital signal processing. The general solution of a linear difference equation with constant coefficients was derived in this chapter and shown to consist of two components: the solution of the homogeneous equation, which represents the natural response of the system when the input is zero, and the particular solution, which represents the response of the system to the input signal. From the difference equation, we also demonstrated how to derive the unit sample response of the LTI system.

Linear time-invariant systems were generally subdivided into FIR (finite-duration impulse response) and IIR (infinite-duration impulse response) depending on whether $h(n)$ has finite duration or infinite duration, respectively. The realizations of such systems were briefly described. Furthermore, in the realization of FIR systems, we made the distinction between recursive and nonrecursive realizations. On the other hand, we observed that IIR systems can be implemented recursively, only.

2.62 Determine the autocorrelation sequences of the following signals.

(a) $x(n) = \{\underset{\uparrow}{1}, 2, 1, 1\}$

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

What is your conclusion?

2.63 What is the normalized autocorrelation sequence of the signal $x(n)$ given by

$$x(n) = \begin{cases} 1, & -N \le n \le N \\ 0, & \text{otherwise} \end{cases}$$

2.64 An audio signal $s(t)$ generated by a loudspeaker is reflected at two different walls with reflection coefficients r_1 and r_2. The signal $x(t)$ recorded by a microphone close to the loudspeaker, after sampling, is

$$x(n) = s(n) + r_1 s(n - k_1) + r_2 s(n - k_2)$$

where k_1 and k_2 are the delays of the two echoes.

(a) Determine the autocorrelation $r_{xx}(l)$ of the signal $x(n)$.

(b) Can we obtain r_1, r_2, k_1, and k_2 by observing $r_{xx}(l)$?

(c) What happens if $r_2 = 0$?

2.65 *Time-delay estimation in radar* Let $x_a(t)$ be the transmitted signal and $y_a(t)$ be the received signal in a radar system, where

$$y_a(t) = ax_a(t - t_d) + v_a(t)$$

and $v_a(t)$ is additive random noise. The signals $x_a(t)$ and $y_a(t)$ are sampled in the receiver, according to the sampling theorem, and are processed digitally to determine the time delay and hence the distance of the object. The resulting discrete-time signals are

$$\begin{aligned} x(n) &= x_a(nT) \\ y(n) &= y_a(nT) = ax_a(nT - DT) + v_a(nT) \\ &\triangleq ax(n - D) + v(n) \end{aligned}$$

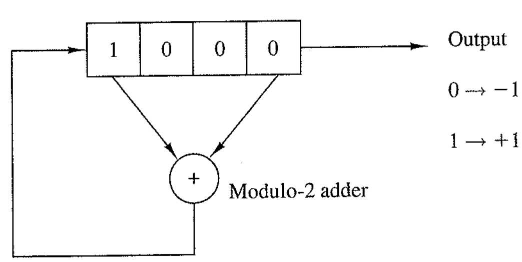

Figure P2.65
Linear feedback shift register.

(a) Explain how we can measure the delay D by computing the crosscorrelation $r_{xy}(l)$.

(b) Let $x(n)$ be the 13-point *Barker sequence*

$$x(n) = \{+1, +1, +1, +1, +1, -1, -1, +1, +1, -1, +1, -1, +1\}$$

and $v(n)$ be a Gaussian random sequence with zero mean and variance $\sigma^2 = 0.01$. Write a program that generates the sequence $y(n)$, $0 \le n \le 199$ for $a = 0.9$ and $D = 20$. Plot the signals $x(n)$, $y(n)$, $0 \le n \le 199$.

(c) Compute and plot the crosscorrelation $r_{xy}(l)$, $0 \le l \le 59$. Use the plot to estimate the value of the delay D.

(d) Repeat parts (b) and (c) for $\sigma^2 = 0.1$ and $\sigma^2 = 1$.

(e) Repeat parts (b) and (c) for the signal sequence

$$x(n) = \{-1, -1, -1, +1, +1, +1, +1, -1, +1, -1, +1, +1, -1, -1, +1\}$$

which is obtained from the four-stage feedback shift register shown in Fig. P2.65. Note that $x(n)$ is just one period of the periodic sequence obtained from the feedback shift register.

(f) Repeat parts (b) and (c) for a sequence of period $N = 2^7 - 1$, which is obtained from a seven-stage feedback shift register. Table 2.2 gives the stages connected to the modulo-2 adder for (maximal-length) shift-register sequences of length $N = 2^m - 1$.

TABLE 2.2 Shift-Register Connections for Generating Maximal-Length Sequences

m	Stages Connected to Modulo-2 Adder
1	1
2	1, 2
3	1, 3
4	1, 4
5	1, 4
6	1, 6
7	1, 7
8	1, 5, 6, 7
9	1, 6
10	1, 8
11	1, 10
12	1, 7, 9, 12
13	1, 10, 11, 13
14	1, 5, 9, 14
15	1, 15
16	1, 5, 14, 16
17	1, 15