Invertibility

A system property that is important in applications such as channel equalization and deconvolution is *invertibility*. A system is said to be *invertible* if the input to the system may be uniquely determined from the output. In order for a system to be invertible, it is necessary for distinct inputs to produce distinct outputs. In other words, given any two inputs $x_1(n)$ and $x_2(n)$ with $x_1(n) \neq x_2(n)$, it must be true that $y_1(n) \neq y_2(n)$.

EXAMPLE 1.3.6 The system defined by

$$y(n) = x(n)g(n)$$

is invertible if and only if $g(n) \neq 0$ for all n. In particular, given $y(n)$ with $g(n)$ nonzero for all n, $x(n)$ may be recovered from $y(n)$ as follows:

$$x(n) = \frac{y(n)}{g(n)}$$

1.4 CONVOLUTION

The relationship between the input to a linear shift-invariant system, $x(n)$, and the output, $y(n)$, is given by the convolution sum

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

Because convolution is fundamental to the analysis and description of LSI systems, in this section we look at the mechanics of performing convolutions. We begin by listing some properties of convolution that may be used to simplify the evaluation of the convolution sum.

1.4.1 Convolution Properties

Convolution is a linear operator and, therefore, has a number of important properties including the commutative, associative, and distributive properties. The definitions and interpretations of these properties are summarized below.

Commutative Property

The commutative property states that the order in which two sequences are convolved is not important. Mathematically, the commutative property is

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

From a systems point of view, this property states that a system with a unit sample response $h(n)$ and input $x(n)$ behaves in exactly the same way as a system with unit sample response $x(n)$ and an input $h(n)$. This is illustrated in Fig. 1-5(a).

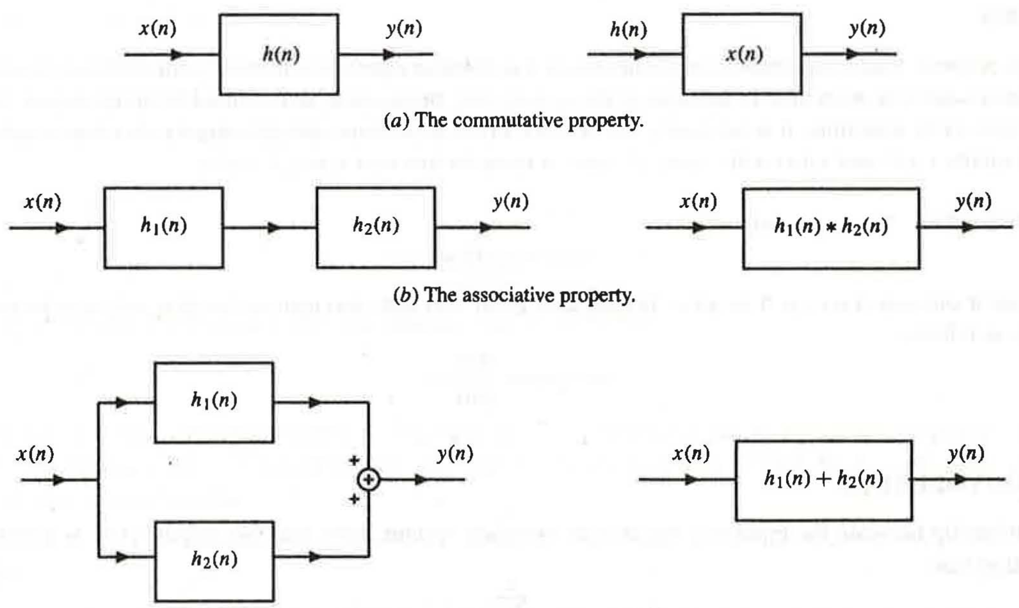
Associative Property

The convolution operator satisfies the associative property, which is

$$\{x(n) * h_1(n)\} * h_2(n) = x(n) * \{h_1(n) * h_2(n)\}$$

From a systems point of view, the associative property states that if two systems with unit sample responses $h_1(n)$ and $h_2(n)$ are connected in cascade as shown in Fig. 1-5(b), an equivalent system is one that has a unit sample response equal to the convolution of $h_1(n)$ and $h_2(n)$:

$$h_{eq}(n) = h_1(n) * h_2(n)$$



(a) The commutative property.

(b) The associative property.

(c) The distributive property.

Fig. 1-5. The interpretation of convolution properties from a systems point of view.

Distributive Property

The distributive property of the convolution operator states that

$$x(n) * \{h_1(n) + h_2(n)\} = x(n) * h_1(n) + x(n) * h_2(n)$$

From a systems point of view, this property asserts that if two systems with unit sample responses $h_1(n)$ and $h_2(n)$ are connected in parallel, as illustrated in Fig. 1-5(*c*), an equivalent system is one that has a unit sample response equal to the sum of $h_1(n)$ and $h_2(n)$:

$$h_{eq}(n) = h_1(n) + h_2(n)$$

1.4.2 Performing Convolutions

Having considered some of the properties of the convolution operator, we now look at the mechanics of performing convolutions. There are several different approaches that may be used, and the one that is the easiest will depend upon the form and type of sequences that are to be convolved.

Direct Evaluation

When the sequences that are being convolved may be described by simple closed-form mathematical expressions, the convolution is often most easily performed by directly evaluating the sum given in Eq. (*1.7*). In performing convolutions directly, it is usually necessary to evaluate finite or infinite sums involving terms of the form α^n or $n\alpha^n$. Listed in Table 1-1 are closed-form expressions for some of the more commonly encountered series.

EXAMPLE 1.4.1 Let us perform the convolution of the two signals

$$x(n) = a^n u(n) = \begin{cases} a^n & n \ge 0 \\ 0 & n < 0 \end{cases}$$

and

$$h(n) = u(n)$$

Table 1-1 Closed-form Expressions for Some Commonly Encountered Series

$\sum_{n=0}^{N-1} a^n = \frac{1-a^N}{1-a}$	$\sum_{n=0}^{\infty} a^n = \frac{1}{1-a} \qquad \|a\| < 1$
$\sum_{n=0}^{N-1} na^n = \frac{(N-1)a^{N+1} - Na^N + a}{(1-a)^2}$	$\sum_{n=0}^{\infty} na^n = \frac{a}{(1-a)^2} \qquad \|a\| < 1$
$\sum_{n=0}^{N-1} n = \frac{1}{2}N(N-1)$	$\sum_{n=0}^{N-1} n^2 = \frac{1}{6}N(N-1)(2N-1)$

With the direct evaluation of the convolution sum we find

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

Because $u(k)$ is equal to zero for $k < 0$ and $u(n-k)$ is equal to zero for $k > n$, when $n < 0$, there are no nonzero terms in the sum and $y(n) = 0$. On the other hand, if $n \geq 0$,

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

Therefore,

$$y(n) = \frac{1-a^{n+1}}{1-a} u(n)$$

Graphical Approach

In addition to the direct method, convolutions may also be performed graphically. The steps involved in using the graphical approach are as follows:

1. Plot both sequences, $x(k)$ and $h(k)$, as functions of k.
2. Choose one of the sequences, say $h(k)$, and time-reverse it to form the sequence $h(-k)$.
3. Shift the time-reversed sequence by n. [*Note:* If $n > 0$, this corresponds to a shift to the right (delay), whereas if $n < 0$, this corresponds to a shift to the left (advance).]
4. Multiply the two sequences $x(k)$ and $h(n-k)$ and sum the product for all values of k. The resulting value will be equal to $y(n)$. This process is repeated for all possible shifts, n.

EXAMPLE 1.4.2 To illustrate the graphical approach to convolution, let us evaluate $y(n) = x(n)*h(n)$ where $x(n)$ and $h(n)$ are the sequences shown in Fig. 1-6 (*a*) and (*b*), respectively.To perform this convolution, we follow the steps listed above:

1. Because $x(k)$ and $h(k)$ are both plotted as a function of k in Fig. 1-6 (*a*) and (*b*), we next choose one of the sequences to reverse in time. In this example, we time-reverse $h(k)$, which is shown in Fig. 1-6 (*c*).
2. Forming the product, $x(k)h(-k)$, and summing over k, we find that $y(0) = 1$.
3. Shifting $h(k)$ to the right by one results in the sequence $h(1-k)$ shown in Fig. 1-6 (*d*). Forming the product, $x(k)h(1-k)$, and summing over k, we find that $y(1) = 3$.
4. Shifting $h(1-k)$ to the right again gives the sequence $h(2-k)$ shown in Fig. 1-6 (*e*). Forming the product, $x(k)h(2-k)$, and summing over k, we find that $y(2) = 6$.
5. Continuing in this manner, we find that $y(3) = 5$, $y(4) = 3$, and $y(n) = 0$ for $n > 4$.
6. We next take $h(-k)$ and shift it to the left by one as shown in Fig. 1-6 (*f*). Because the product, $x(k)h(-1-k)$, is equal to zero for all k, we find that $y(-1) = 0$. In fact, $y(n) = 0$ for all $n < 0$.

Figure 1-6 (*g*) shows the convolution for all n.

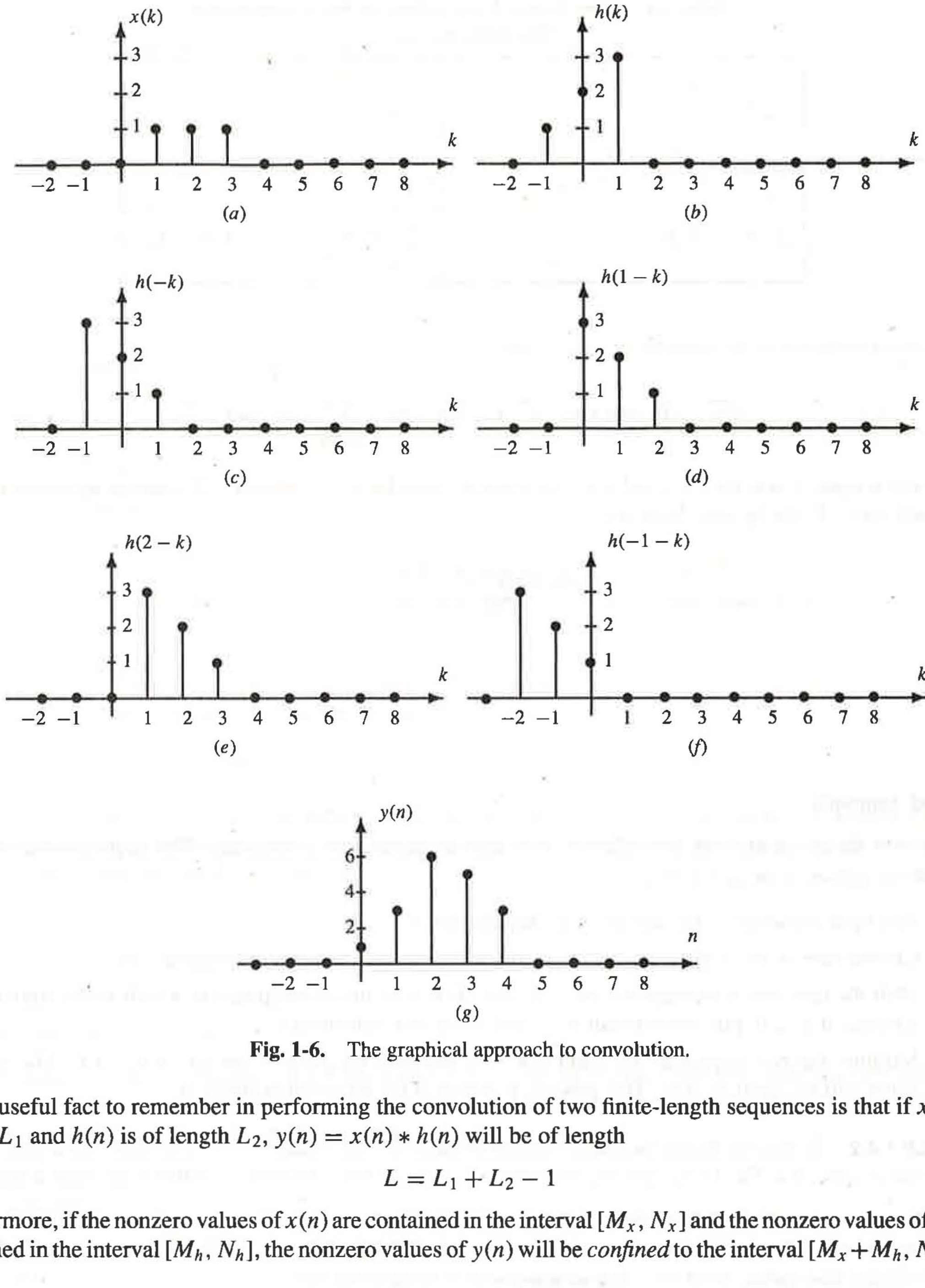


Fig. 1-6. The graphical approach to convolution.

A useful fact to remember in performing the convolution of two finite-length sequences is that if $x(n)$ is of length L_1 and $h(n)$ is of length L_2, $y(n) = x(n) * h(n)$ will be of length

$$L = L_1 + L_2 - 1$$

Furthermore, if the nonzero values of $x(n)$ are contained in the interval $[M_x, N_x]$ and the nonzero values of $h(n)$ are contained in the interval $[M_h, N_h]$, the nonzero values of $y(n)$ will be *confined* to the interval $[M_x + M_h, N_x + N_h]$.

EXAMPLE 1.4.3 Consider the convolution of the sequence

$$x(n) = \begin{cases} 1 & 10 \le n \le 20 \\ 0 & \text{otherwise} \end{cases}$$

with

$$h(n) = \begin{cases} n & -5 \le n \le 5 \\ 0 & \text{otherwise} \end{cases}$$

Because $x(n)$ is zero outside the interval [10, 20], and $h(n)$ is zero outside the interval [−5, 5], the nonzero values of the convolution, $y(n) = x(n) * h(n)$, will be contained in the interval [5, 25].

Slide Rule Method

Another method for performing convolutions, which we call the *slide rule method*, is particularly convenient when both $x(n)$ and $h(n)$ are finite in length and short in duration. The steps involved in the slide rule method are as follows:

1. Write the values of $x(k)$ along the top of a piece of paper, and the values of $h(-k)$ along the top of another piece of paper as illustrated in Fig. 1-7.
2. Line up the two sequence values $x(0)$ and $h(0)$, multiply each pair of numbers, and add the products to form the value of $y(0)$.
3. Slide the paper with the time-reversed sequence $h(k)$ to the right by one, multiply each pair of numbers, sum the products to find the value $y(1)$, and repeat for all shifts to the right by $n > 0$. Do the same, shifting the time-reversed sequence to the left, to find the values of $y(n)$ for $n < 0$.

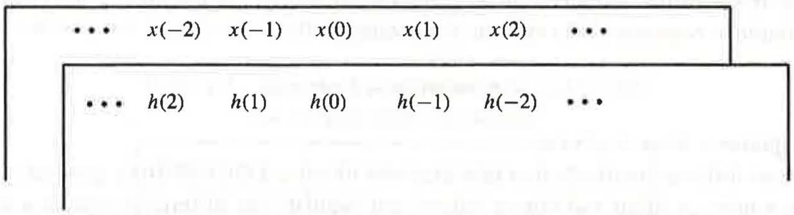


Fig. 1-7. The slide rule approach to convolution.

In Chap. 2 we will see that another way to perform convolutions is to use the Fourier transform.

1.5 DIFFERENCE EQUATIONS

The convolution sum expresses the output of a linear shift-invariant system in terms of a linear combination of the input values $x(n)$. For example, a system that has a unit sample response $h(n) = \alpha^n u(n)$ is described by the equation

$$y(n) = \sum_{k=0}^{\infty} \alpha^k x(n-k) \tag{1.9}$$

Although this equation allows one to compute the output $y(n)$ for an arbitrary input $x(n)$, from a computational point of view this representation is not very efficient. In some cases it may be possible to more efficiently express the output in terms of past values of the output in addition to the current and past values of the input. The previous system, for example, may be described more concisely as follows:

$$y(n) = \alpha y(n-1) + x(n) \tag{1.10}$$

Equation *(1.10)* is a special case of what is known as a *linear constant coefficient difference equation*, or LCCDE. The general form of a LCCDE is

$$y(n) = \sum_{k=0}^{q} b(k)x(n-k) - \sum_{k=1}^{p} a(k)y(n-k) \tag{1.11}$$

where the coefficients $a(k)$ and $b(k)$ are constants that define the system. If the difference equation has one or more terms $a(k)$ that are nonzero, the difference equation is said to be *recursive*. On the other hand, if all of the coefficients $a(k)$ are equal to zero, the difference equation is said to be *nonrecursive*. Thus, Eq. *(1.10)* is an example of a first-order recursive difference equation, whereas Eq. *(1.9)* is an infinite-order nonrecursive difference equation.

Difference equations provide a method for computing the response of a system, $y(n)$, to an arbitrary input $x(n)$. Before these equations may be solved, however, it is necessary to specify a set of *initial conditions*. For example, with an input $x(n)$ that begins at time $n = 0$, the solution to Eq. *(1.11)* at time $n = 0$ depends on the