CHAPTER

DISCRETE-TIME SIGNALS AND SYSTEMS

9

CONTENTS

It's like deja-vu, all over again.

Lawrence "Yogi" Berra, Yankees baseball player (1925)

9.1 INTRODUCTION

As you will see in this chapter, the basic theory of discrete-time signals and systems is very much like that of continuous-time signals and systems. However, there are significant differences that need to be understood.

Signals and Systems Using MATLAB®. https://doi.org/10.1016/B978-0-12-814204-2.00020-X

Discrete-time signals resulting from sampling of continuous-time signals are only available at uniform times determined by the sampling period—they are not defined in between samples. It is important to emphasize the significance of sampling according to the Nyquist sampling rate condition since the characteristics of discrete-time signals will depend on it. Given the knowledge of the sampling period, discrete-time signals are functions of an integer variable n which unifies the treatment of discrete-time signals obtained from analog signals by sampling and those that are naturally discrete. It will be seen also that the frequency in the discrete domain differs from the analog frequency. This radian discrete frequency cannot be measured and depends on the sampling period used whenever the discrete-time signals result from sampling.

Although the concept of periodicity of discrete-time signals coincides with that, for continuous-time signals, there are significant differences. As functions of an integer variable, discrete-time periodic signals must have integer periods. This imposes some restrictions that do not exist in continuous-time periodic signals. For instance, analog sinusoids are always periodic as their period can be a positive real number, however, that will not be the case for discrete sinusoids. It is possible to have discrete sinusoids that are not periodic, even if they resulted from the uniform sampling of periodic continuous-time sinusoids. Characteristics such as energy, power, and symmetry of continuous-time signals are conceptually the same for discrete-time signals. Likewise, one can define a set of basic signals just like those for continuous-time signals. However, some of these basic signals do not display the mathematical complications of their analog counterparts. For instance, the discrete impulse signal is defined at every integer value in contrast with the continuous impulse which is not defined at zero.

The discrete approximation of derivatives and integrals provides an approximation of ordinary differential equations, representing dynamic continuous-time systems, by difference equations. Extending the concept of linear time invariance to discrete-time systems, we obtain a convolution sum to represent them. Thus dynamic discrete-time systems can be represented by difference equations and convolution sums. A computationally significant difference with continuous-time systems is that the solution of difference equations can be recursively obtained, and that the convolution sum provides a class of systems that do not have a counterpart in the analog domain.

In this chapter, we also introduce the theory of two-dimensional signals and systems. Particular two-dimensional signals of interest are images that are processed using two-dimensional systems. We will see how the theory of one-dimensional signals and systems can be extended to two dimensions, and how this theory is much richer. In fact, although many of the one-dimensional properties are valid for two dimensions not all are; it is actually the one-dimensional signals and systems that are subsets of the two-dimensional signals and systems.

9.2 DISCRETE-TIME SIGNALS

A **discrete-time signal** $x[n]$ can be thought of as a real- or complex-valued function of the integer sample index n:

$$\begin{aligned} x[.] : \mathcal{I} &\rightarrow \mathcal{R} \;\; (\mathcal{C}) \\ n \;\; & \;\; x[n]. \end{aligned} \tag{9.1}$$

The above means that for discrete-time signals the independent variable is an integer n, the sample index, and that the value of the signal at n, $x[n]$, is either a real- or a complex value. Thus, the signal is only defined at integer values n, no definition exists for values between the integers.

Remarks

1. It should be understood that a sampled signal $x(nT_s) = x(t)|_{t=nT_s}$ is a discrete-time signal $x[n]$ which is a function of n only. Once the value of T_s is known, the sampled signal only depends on n, the sample index. However, this should not prevent us in some situations from considering a discrete-time signal obtained through sampling as a function of time t where the signal values only exist at discrete times $\{nT_s\}$.
2. Although in many situations, discrete-time signals are obtained from continuous-time signals by sampling, that is not always the case. There are many signals which are **inherently discrete**. Think, for instance, of a signal consisting of the final values attained daily by the shares of a company in the stock market. Such a signal would consist of the values reached by the share in the days when the stock market opens and has no connection with a continuous-time signal. This signal is naturally discrete. A signal generated by a random number generator in a computer would be a sequence of real values and can be considered a discrete-time signal. Telemetry signals, consisting of measurements—e.g., voltages, temperatures, pressures—from a certain process, taken at certain times are also naturally discrete.

Example 9.1. Consider a sinusoidal signal $x(t) = 3\cos(2\pi t + \pi/4)$, $-\infty < t < \infty$. Determine an appropriate sampling period T_s according to the Nyquist sampling rate condition, and obtain the discrete-time signal $x[n]$ corresponding to the largest allowed sampling period.

Solution: To sample $x(t)$ so that no information is lost, the Nyquist sampling rate condition indicates that the sampling period should be

$$T_s \le \frac{\pi}{\Omega_{max}} = \frac{\pi}{2\pi} = 0.5 \text{ s/sample}.$$

For the largest allowed sampling period $T_s = 0.5$ s/sample we obtain

$$x[n] = 3\cos(2\pi t + \pi/4)|_{t=0.5n} = 3\cos(\pi n + \pi/4) \qquad -\infty < n < \infty,$$

which is a function of the integer n. □

Example 9.2. To generate the celebrated Fibonacci sequence of numbers, $\{x[n],\ n \ge 0\}$, we use the following recursive equation:

$$x[n] = x[n-1] + x[n-2], \quad n \ge 2, \quad \text{initial conditions: } x[0] = 0, \quad x[1] = 1,$$

which is a difference equation with zero-input and two initial conditions. The Fibonacci sequence has been used to model different biological systems.[1] Find the Fibonacci sequence.

[1]Leonardo of Pisa (also known as Fibonacci) in his book Liber Abaci described how his sequence could be used to model the reproduction of rabbits over a number of months assuming bunnies begin breeding when they are a few months old.

Solution: The given equation allows us to compute the Fibonacci sequence recursively. For $n \geq 2$ we find

$$x[2] = 1 + 0 = 1, \quad x[3] = 1 + 1 = 2, \quad x[4] = 2 + 1 = 3, \quad x[5] = 3 + 2 = 5, \quad \cdots,$$

where we are simply adding the previous two numbers in the sequence. The sequence is purely discrete as it is not related to a continuous-time signal. □

9.2.1 PERIODIC AND APERIODIC DISCRETE-TIME SIGNALS

A discrete-time signal $x[n]$ is periodic if

- it is defined for all possible values of n, $-\infty < n < \infty$, and
- there is a positive integer N such that

$$x[n + kN] = x[n] \tag{9.2}$$

for any integer k. The smallest value of N is called the **fundamental period** of $x[n]$.

An aperiodic signal does not satisfy one or both of the above conditions.
Periodic discrete-time sinusoids, of fundamental period N, are of the form

$$x[n] = A\cos\left(\frac{2\pi m}{N} n + \theta\right) \qquad -\infty < n < \infty \tag{9.3}$$

where the discrete frequency is $\omega_0 = 2\pi m/N$ (rad), for positive integers m and N which are not divisible by each other, and θ is the phase angle.

The definition of a discrete-time periodic signal is similar to that of continuous-time periodic signals, except for the fundamental period being an integer. That periodic discrete-time sinusoids are of the given form can easily be shown: shifting the sinusoid in (9.3) by a multiple k of the fundamental period N, we have

$$x[n + kN] = A\cos\left(\frac{2\pi m}{N}(n + kN) + \theta\right) = A\cos\left(\frac{2\pi m}{N} n + 2\pi mk + \theta\right) = x[n],$$

since we add to the original angle a multiple mk (an integer) of 2π, which does not change the angle. If the discrete frequency is not in the form $2\pi m/N$, the sinusoid is not periodic.

Remark
The unit of the discrete frequency ω is radians. Moreover, discrete frequencies repeat every 2π, i.e., $\omega = \omega + 2\pi k$ for any integer k, and as such we only need to consider the range $-\pi \leq \omega < \pi$. This is in contrast with the analog frequency Ω which has rad/s as unit, and its range is from $-\infty$ to ∞.

Example 9.3. Consider the sinusoids

$$x_1[n] = 2\cos(\pi n - \pi/3), \quad x_2[n] = 3\sin(3\pi n + \pi/2) \qquad -\infty < n < \infty.$$

From their frequencies determine if these signals are periodic, and if periodic their corresponding fundamental periods.

Solution: The frequency of $x_1[n]$ can be written

$$\omega_1 = \pi = \frac{2\pi}{2}$$

for integers $m = 1$ and $N = 2$, so that $x_1[n]$ is periodic of fundamental period $N_1 = 2$. Likewise the frequency of $x_2[n]$ can be written

$$\omega_2 = 3\pi = \frac{2\pi}{2}3$$

for integers $m = 3$ and $N = 2$, so that $x_2[n]$ is also periodic of fundamental period $N_2 = 2$, which can be verified as follows:

$$x_2[n+2] = 3\sin(3\pi(n+2) + \pi/2) = 3\sin(3\pi n + 6\pi + \pi/2) = x_2[n]. \quad \square$$

Example 9.4. What is true for analog sinusoids—that they are always periodic—is not true for discrete sinusoids. Discrete sinusoids can be non-periodic even if they result from uniformly sampling a periodic continuous-time sinusoid. Consider the discrete-time signal $x[n] = \cos(n + \pi/4)$, $-\infty < n < \infty$, which is obtained by sampling the analog sinusoid $x(t) = \cos(t + \pi/4)$, $-\infty < t < \infty$, with a sampling period $T_s = 1$ s/sample. Is $x[n]$ periodic? If so, indicate its fundamental period. Otherwise, determine values of the sampling period satisfying the Nyquist sampling rate condition that when used in sampling $x(t)$ result in periodic signals.

Solution: The sampled signal $x[n] = x(t)|_{t=nT_s} = \cos(n + \pi/4)$ has a discrete frequency $\omega = 1$ (rad) which cannot be expressed as $2\pi m/N$ for any integers m and N because π is an irrational number. So $x[n]$ is not periodic.

Since the frequency of the continuous-time signal $x(t) = \cos(t + \pi/4)$ is $\Omega_0 = 1$, the sampling period, according to the Nyquist sampling rate condition, should be

$$T_s \leq \frac{\pi}{\Omega_0} = \pi$$

and for the sampled signal $x(t)|_{t=nT_s} = \cos(nT_s + \pi/4)$ to be periodic of fundamental period N or

$$\cos((n+N)T_s + \pi/4) = \cos(nT_s + \pi/4) \quad \text{it is necessary that} \quad NT_s = 2k\pi$$

for an integer k, i.e., a multiple of 2π. Thus $T_s = 2k\pi/N \leq \pi$ satisfies the Nyquist sampling condition at the same time that ensures the periodicity of the sampled signal. For instance, if we wish to have a discrete sinusoid with fundamental period $N = 10$, then $T_s = k\pi/5$, for k chosen so that the Nyquist sampling rate condition is satisfied, i.e.,

$$0 < T_s = k\pi/5 \leq \pi \quad \text{so that} \quad 0 < k \leq 5.$$

From these possible values for k we choose $k = 1$ and 3 so that N and k are not divisible by each other and we get the desired fundamental period $N = 10$ (the values $k = 2$ and 4 would give 5 as the period, and $k = 5$ would give a period of 2 instead of 10). Indeed, if we let $k = 1$ then $T_s = 0.2\pi$ satisfies the Nyquist sampling rate condition, and we obtain the sampled signal

$$x[n] = x(t)|_{t=0.2\pi n} = \cos(0.2\pi n + \pi/4) = \cos\left(\frac{2\pi}{10}n + \frac{\pi}{4}\right),$$

which according to its frequency is periodic of fundamental period 10. For $k = 3$, we have $T_s = 0.6\pi < \pi$ and

$$x[n] = x(t)|_{t=0.6\pi n} = \cos(0.6\pi n + \pi/4) = \cos\left(\frac{2\pi \times 3}{10}n + \frac{\pi}{4}\right)$$

also of fundamental period $N = 10$. □

When sampling an analog sinusoid

$$x(t) = A\cos(\Omega_0 t + \theta) \qquad -\infty < t < \infty \tag{9.4}$$

of fundamental period $T_0 = 2\pi/\Omega_0$, $\Omega_0 > 0$, we obtain a **periodic discrete sinusoid**

$$x[n] = A\cos(\Omega_0 T_s n + \theta) = A\cos\left(\frac{2\pi T_s}{T_0}n + \theta\right) \tag{9.5}$$

provided that

$$\frac{T_s}{T_0} = \frac{m}{N} \tag{9.6}$$

for positive integers N and m which are not divisible by each other. To avoid frequency aliasing the sampling period should also satisfy the Nyquist sampling condition,

$$T_s \leq \frac{\pi}{\Omega_0} = \frac{T_0}{2}. \tag{9.7}$$

Indeed, sampling a continuous-time signal $x(t)$ using as sampling period T_s we obtain

$$x[n] = A\cos(\Omega_0 T_s n + \theta) = A\cos\left(\frac{2\pi T_s}{T_0}n + \theta\right)$$

where the discrete frequency is $\omega_0 = 2\pi T_s/T_0$. For this signal to be periodic we should be able to express this frequency as $2\pi m/N$ for non-divisible positive integers m and N. This requires that

$$\frac{T_s}{T_0} = \frac{m}{N}$$

be rational with integers m and N not divisible by each other, or that

$$mT_0 = NT_s, \tag{9.8}$$

which says that a period ($m = 1$) or several periods ($m > 1$) should be divided into $N > 0$ segments of duration T_s s. If this condition is not satisfied, then the discretized sinusoid is not periodic. To avoid frequency aliasing, the sampling period should be chosen so that

$$T_s \leq \frac{\pi}{\Omega_0} = \frac{T_0}{2}$$

as determined by the Nyquist sampling condition.

> The sum $z[n] = x[n] + y[n]$ of periodic signals $x[n]$ with fundamental period N_1, and $y[n]$ with fundamental period N_2 is periodic if the ratio of periods of the summands is rational, i.e.,
>
> $$\frac{N_2}{N_1} = \frac{p}{q}$$
>
> and p and q are integers not divisible by each other. If so, the fundamental period of $z[n]$ is $qN_2 = pN_1$.

If $qN_2 = pN_1$, since pN_1 and qN_2 are multiples of the periods of $x[n]$ and $y[n]$, we have

$$z[n + pN_1] = x[n + pN_1] + y[n + pN_1] = x[n] + y[n + qN_2] = x[n] + y[n] = z[n]$$

or $z[n]$ is periodic of fundamental period $qN_2 = pN_1$.

Example 9.5. The signal $z[n] = v[n] + w[n] + y[n]$ is the sum of three periodic signals $v[n]$, $w[n]$ and $y[n]$ of fundamental periods $N_1 = 2$, $N_2 = 3$ and $N_3 = 4$, respectively. Determine if $z[n]$ is periodic, and if so give its fundamental period.

Solution: Let $x[n] = v[n] + w[n]$, so that $z[n] = x[n] + y[n]$. The signal $x[n]$ is periodic since $N_2/N_1 = 3/2$ is a rational number with non-divisible factors, and its fundamental period is $N_4 = 3N_1 = 2N_2 = 6$. The signal $z[n]$ is also periodic since

$$\frac{N_4}{N_3} = \frac{6}{4} = \frac{3}{2}$$

is rational with non-divisible factors. The fundamental period of $z[n]$ is $N = 2N_4 = 3N_3 = 12$. Thus $z[n]$ is periodic of fundamental period 12, indeed

$$z[n + 12] = v[n + 6N_1] + w[n + 4N_2] + y[n + 3N_3] = v[n] + w[n] + y[n] = z[n]. \quad \square$$

Example 9.6. Determine if the signal

$$x[n] = \sum_{m=0}^{\infty} X_m \cos(m\omega_0 n), \qquad \omega_0 = \frac{2\pi}{N_0}$$

is periodic, and if so determine its fundamental period.

Solution: The signal $x[n]$ consists of the sum of a constant X_0 and cosines of frequency

$$m\omega_0 = \frac{2\pi m}{N_0} \qquad m = 1, 2, \cdots .$$

The periodicity of $x[n]$ depends on the periodicity of the cosines. According to the frequency of the cosines, they are periodic of fundamental period N_0. Thus $x[n]$ is periodic of fundamental period N_0, indeed

$$x[n+N_0] = \sum_{m=0}^{\infty} X_m \cos(m\omega_0(n+N_0)) = \sum_{m=0}^{\infty} X_m \cos(m\omega_0 n + 2\pi m) = x[n]. \quad \square$$

9.2.2 FINITE-ENERGY AND FINITE-POWER DISCRETE-TIME SIGNALS

For discrete-time signals, we obtain definitions for energy and power similar to those for continuous-time signals by replacing integrals by summations.

For a discrete-time signal $x[n]$ we have the following definitions:

$$\textbf{Energy:} \qquad \varepsilon_x = \sum_{n=-\infty}^{\infty} |x[n]|^2 \tag{9.9}$$

$$\textbf{Power:} \qquad P_x = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2. \tag{9.10}$$

- $x[n]$ is said to have **finite energy** or to be **square summable** if $\varepsilon_x < \infty$.
- $x[n]$ is called **absolutely summable** if

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty. \tag{9.11}$$

- $x[n]$ is said to have **finite power** if $P_x < \infty$.

Example 9.7. A "causal" sinusoid, obtained from a signal generator after it is switched on, is

$$x(t) = \begin{cases} 2\cos(\Omega_0 t - \pi/4) & t \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

The signal $x(t)$ is sampled using a sampling period of $T_s = 0.1$ s to obtain a discrete-time signal

$$x[n] = x(t)|_{t=0.1n} = 2\cos(0.1\Omega_0 n - \pi/4) \qquad n \geq 0$$

and zero otherwise. Determine if this discrete-time signal has finite energy, finite power and compare these characteristics with those of the continuous-time signal $x(t)$ when $\Omega_0 = \pi$ and when $\Omega_0 = 3.2$ rad/s (an upper approximation of π).

Solution: The continuous-time signal $x(t)$ has infinite energy, and so does the discrete-time signal $x[n]$, for both values of Ω_0. Indeed, the energy of $x[n]$ is

$$\varepsilon_x = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \sum_{n=0}^{\infty} 4\cos^2(0.1\Omega_0 n - \pi/4) \to \infty$$

for both values of Ω_0. Although the continuous-time and the discrete-time signals have infinite energy, they have finite power. That the continuous-time signal $x(t)$ has finite power can be shown as indicated in Chapter 1. For the discrete-time signal, $x[n]$, we have for the two frequencies:
(i) For $\Omega_0 = \pi$, $x[n] = 2\cos(\pi n/10 - \pi/4) = 2\cos(2\pi n/20 - \pi/4)$ for $n \geq 0$ and zero otherwise. Thus $x[n]$ repeats every $N_0 = 20$ samples for $n \geq 0$, and its power is

$$\begin{aligned}
P_x &= \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2 = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=0}^{N} |x[n]|^2 \\
&= \lim_{N\to\infty} \frac{1}{2N+1} N \underbrace{\left[\frac{1}{N_0} \sum_{n=0}^{N_0-1} |x[n]|^2\right]}_{\text{power of a period}} \\
&= \lim_{N\to\infty} \frac{1}{2+1/N} \left[\frac{1}{N_0} \sum_{n=0}^{N_0-1} |x[n]|^2\right] = \frac{1}{2N_0} \sum_{n=0}^{N_0-1} |x[n]|^2 < \infty
\end{aligned}$$

where we used the causality of the signal ($x[n] = 0$ for $n < 0$), and considered N periods of $x[n]$ for $n \geq 0$ and for each calculated its power to get the final result. Thus for $\Omega_0 = \pi$ the discrete-time signal $x[n]$ has finite power and can be computed using a period for $n \geq 0$. To find the power we use the trigonometric identity (or Euler's equation) $\cos^2(\theta) = 0.5(1 + \cos(2\theta))$ and so replacing $x[n]$ we have recalling that $N_0 = 20$:

$$P_x = \frac{4}{40} 0.5 \left[\sum_{n=0}^{19} 1 + \sum_{n=0}^{19} \cos\left(\frac{2\pi n}{10} - \pi/2\right)\right] = \frac{2}{40}[20 + 0] = 1$$

where the sum of the cosine is zero, as we are adding the values of the periodic cosine over two periods.
(ii) For $\Omega_0 = 3.2$, $x[n] = 2\cos(3.2n/10 - \pi/4)$ for $n \geq 0$ and zero otherwise. The signal now does not repeat periodically after $n = 0$, as the frequency $3.2/10$ (which equals the rational $32/100$) cannot be expressed as $2\pi m/N$ (since π is an irrational value) for integers m and N. Thus in this case we do not have a closed form for the power, we can simply say that the power is

$$P_x = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2 = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=0}^{N} |x[n]|^2$$

and conjecture that because the analog signal has finite power, so would $x[n]$. Thus we use MATLAB to compute the power for both cases.

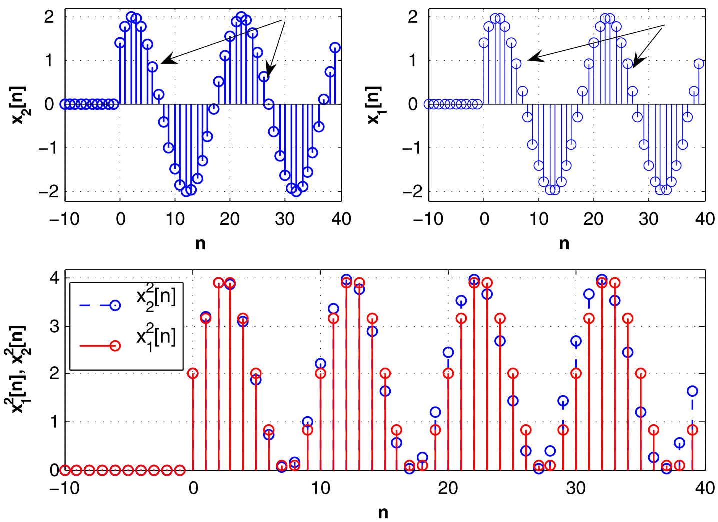

FIGURE 9.1

Top figures: signals $x_2[n]$ (non-periodic for $n \geq 0$) and $x_1[n]$ (periodic for $n \geq 0$). The arrows show that the values are not equal for $x_2[n]$ and equal for $x_1[n]$. Bottom figure: the squares of the signals differ slightly suggesting that if $x_1[n]$ has finite power so does $x_2[n]$.

```
%%
% Example 9.7---Power
%%
clear all;clf
n=0:100000;
x2=2*cos(0.1*n*3.2-pi/4);          % non-periodic for positive n
x1=2*cos(0.1*n*pi-pi/4);           % periodic for positive n
N=length(x1)
Px1=sum(x1.^2)/(2*N+1)             % power of x1
Px2=sum(x2.^2)/(2*N+1)             % power of x2
P1=sum(x1(1:20).^2)/(20);          % power in one period of x1
```

The signal $x_1[n]$ in the script has unit power and so does the signal $x_2[n]$ when we consider 100,001 samples. (See Fig. 9.1.) □

Example 9.8. Determine if a discrete-time exponential $x[n] = 2\,(0.5)^n$, $n \geq 0$, and zero otherwise, has finite energy, finite power or both.

Solution: The energy is given by

$$\varepsilon_x = \sum_{n=0}^{\infty} 4(0.5)^{2n} = 4\sum_{n=0}^{\infty}(0.25)^n = \frac{4}{1-0.25} = \frac{16}{3}$$

thus $x[n]$ is a finite-energy signal. Just as with continuous-time signals, a finite-energy signal is a finite power (actually zero power) signal. Indeed,

$$P_x = \lim_{N\to\infty} \frac{1}{2N+1}\varepsilon_x = 0. \quad \square$$

9.2.3 EVEN AND ODD DISCRETE-TIME SIGNALS

Time shifting and scaling of discrete-time signals are very similar to the continuous-time cases, the only difference being that the operations are now done using integers.

> A discrete-time signal $x[n]$ is said to be
>
> - **delayed** by L (an integer) samples if $x[n-L]$ is $x[n]$ shifted to the right L samples,
> - **advanced** by M (an integer) samples if $x[n+M]$ is $x[n]$ shifted to the left M samples,
> - **reflected** if the variable n in $x[n]$ is negated, i.e., $x[-n]$.

The shifting to the right or the left can be readily seen by considering when $x[0]$ is attained. For $x[n-L]$, this is when $n=L$, i.e., L samples to the right of the origin or equivalently $x[n]$ is delayed by L samples. Likewise for $x[n+M]$ the $x[0]$ appears when $n=-M$, or advanced by M samples. Negating the variable n flips the signal over with respect to the origin.

Example 9.9. A triangular discrete pulse is defined as

$$x[n] = \begin{cases} n & 0 \le n \le 10, \\ 0 & \text{otherwise.} \end{cases}$$

Find an expression for $y[n] = x[n+3] + x[n-3]$ and $z[n] = x[-n] + x[n]$ in terms of n and carefully plot them.

Solution: Replacing n by $n+3$ and by $n-3$ in the definition of $x[n]$ we get the advanced and delayed signals

$$x[n+3] = \begin{cases} n+3 & -3 \le n \le 7, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$x[n-3] = \begin{cases} n-3 & 3 \le n \le 13, \\ 0 & \text{otherwise,} \end{cases}$$

so that when added we get

$$y[n] = x[n+3] + x[n-3] = \begin{cases} n+3 & -3 \le n \le 2, \\ 2n & 3 \le n \le 7, \\ n-3 & 8 \le n \le 13, \\ 0 & \text{otherwise.} \end{cases}$$

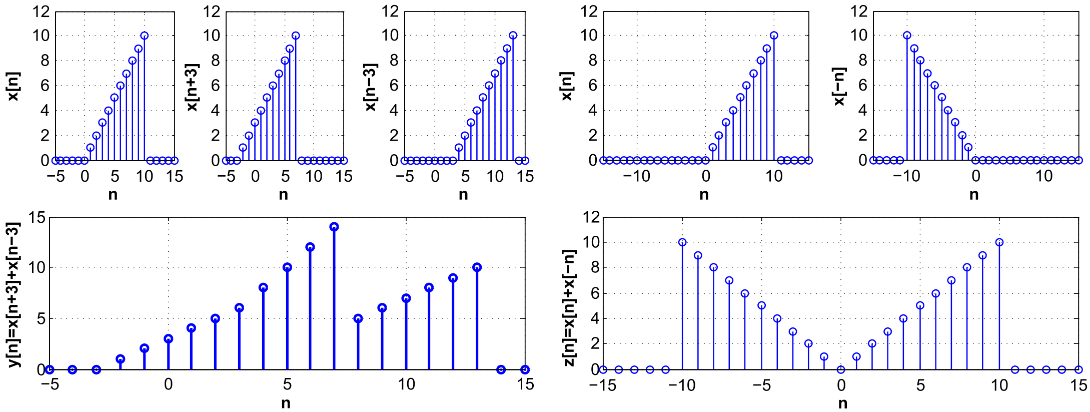

FIGURE 9.2

Generation of $y[n] = x[n+3] + x[n-3]$ (left) and $z[n] = x[n] + x[-n]$ (right).

Likewise, we have

$$z[n] = x[n] + x[-n] = \begin{cases} n & 1 \le n \le 10, \\ 0 & n = 0, \\ -n & -10 \le n \le -1, \\ 0 & \text{otherwise.} \end{cases}$$

The results are shown in Fig. 9.2. □

Example 9.10. We will see that in the convolution sum we need to figure out how a signal $x[n-k]$ behaves as a function of k for different values of n. Consider the signal

$$x[k] = \begin{cases} k & 0 \le k \le 3, \\ 0 & \text{otherwise.} \end{cases}$$

Obtain an expression for $x[n-k]$ for $-2 \le n \le 2$ and determine in which direction it shifts as n increases from -2 to 2.

Solution: Although $x[n-k]$, as a function of k, is reflected it is not clear if it is advanced or delayed as n increases from -2 to 2. If $n = 0$,

$$x[-k] = \begin{cases} -k & -3 \le k \le 0, \\ 0 & \text{otherwise.} \end{cases}$$

For $n \neq 0$, we have

$$x[n-k] = \begin{cases} n-k & n-3 \le k \le n, \\ 0 & \text{otherwise.} \end{cases}$$

As n increases from -2 to 2, $x[n-k]$ moves to the right. For $n=-2$ the support of $x[-2-k]$ is $-5 \leq k \leq -2$, while for $n=0$ the support of $x[-k]$ is $-3 \leq k \leq 0$, and for $n=2$ the support of $x[2-k]$ is $-1 \leq k \leq 2$, each shifted to the right. □

We can thus use the above to define even and odd signals and obtain a general decomposition of any signal in terms of even and odd signals.

Even and odd discrete-time signals are defined as

$$x[n] \text{ is } \textbf{even} \text{ if } x[n] = x[-n], \tag{9.12}$$

$$x[n] \text{ is } \textbf{odd} \text{ if } x[n] = -x[-n]. \tag{9.13}$$

Any discrete-time signal $x[n]$ can be represented as the sum of an even component and an odd component

$$\begin{aligned} x[n] &= \underbrace{\frac{1}{2}(x[n]+x[-n])}_{x_e[n]} + \underbrace{\frac{1}{2}(x[n]-x[-n])}_{x_o[n]} \\ &= x_e[n]+x_o[n]. \end{aligned} \tag{9.14}$$

The even and odd decomposition can easily be seen. The even component $x_e[n]=0.5(x[n]+x[-n])$ is even since $x_e[-n]=0.5(x[-n]+x[n])$ equals $x_e[n]$, and the odd component $x_o[n]=0.5(x[n]-x[-n])$ is odd, since $x_o[-n]=0.5(x[-n]-x[n])=-x_o[n]$.

Example 9.11. Find the even and odd components of the following discrete-time signal:

$$x[n] = \begin{cases} 4-n & 0 \leq n \leq 4, \\ 0 & \text{otherwise.} \end{cases}$$

Solution: The even component of $x[n]$ is given by

$$x_e[n] = 0.5(x[n]+x[-n]).$$

When $n=0$ then $x_e[0]=0.5 \times 2x[0]=4$, when $n>0$ then $x_e[n]=0.5x[n]$, and when $n<0$, then $x_e[n]=0.5x[-n]$ giving

$$x_e[n] = \begin{cases} 2+0.5n & -4 \leq n \leq -1, \\ 4 & n=0, \\ 2-0.5n & 1 \leq n \leq 4, \\ 0 & \text{otherwise,} \end{cases}$$

while the odd component

$$x_o[n] = 0.5(x[n]-x[-n]) = \begin{cases} -2-0.5n & -4 \leq n \leq -1, \\ 0 & n=0, \\ 2-0.5n & 1 \leq n \leq 4, \\ 0 & \text{otherwise.} \end{cases}$$

The sum of these two components gives $x[n]$. □

Remark

Expansion and compression of discrete-time signals are more complicated that in the continuous-time. In the discrete domain, expansion and compression can be related to the change of the sampling period in the sampling. Thus, if a continuous-time signal $x(t)$ is sampled using a sampling period T_s, by changing the sampling period to MT_s, for an integer $M > 1$, we obtain fewer samples, and by changing the sampling period to T_s/L, for an integer $L > 1$, we increase the number of samples.

For the corresponding discrete-time signal $x[n]$, increasing the sampling period would give $x[Mn]$ which is called the **down-sampling** of $x[n]$ by M. Unfortunately, because the arguments of the discrete-time signal must be integers, it is not clear what $x[n/L]$ is unless the values for n are multiples of L, i.e., $n = 0, \pm L, \pm 2L, \cdots$, with no clear definition when n takes other values. This leads to the definition of the **up-sampled** signal as

$$x_u[n] = \begin{cases} x[n/L] & n = 0, \pm L, \pm 2L, \cdots, \\ 0 & \text{otherwise.} \end{cases} \tag{9.15}$$

To replace the zero entries with the values obtained by decreasing the sampling period we need to low-pass filter the up-sampled signal—this is equivalent to interpolating the nonzero values to replace the zero values of the up-sampled signal. MATLAB provides the functions *decimate* and *interp* to implement decimation (related to down-sampling) and interpolation (related to up-sampling). In Chapter 10, we will continue the discussion of these operations including their frequency characterization.

9.2.4 BASIC DISCRETE-TIME SIGNALS

The representation of discrete-time signals via basic signals is simpler than in the continuous-time domain. This is due to the lack of ambiguity in the definition of the impulse and the unit-step discrete-time signals. The definitions of impulses and unit-step signals in the continuous-time domain are more abstract.

Discrete-Time Complex Exponential

Given complex numbers $A = |A|e^{j\theta}$ and $\alpha = |\alpha|e^{j\omega_0}$, a **discrete-time complex exponential** is a signal of the form

$$\begin{aligned} x[n] &= A\alpha^n = |A||\alpha|^n e^{j(\omega_0 n+\theta)} \\ &= |A||\alpha|^n \left[\cos(\omega_0 n + \theta) + j\sin(\omega_0 n + \theta)\right] \end{aligned} \tag{9.16}$$

where ω_0 is a discrete frequency in radians.

Remarks

1. The discrete-time complex exponential looks different from the continuous-time complex exponential. This can be explained by sampling the continuous-time complex exponential

$$x(t) = e^{(-a+j\Omega_0)t}$$

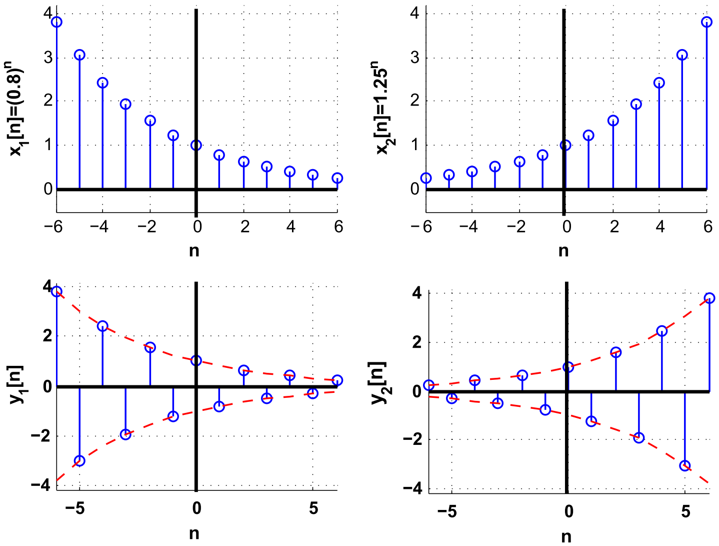

FIGURE 9.3

Real exponential $x_1[n] = 0.8^n$, $x_2[n] = 1.25^n$ (top) and modulated $y_1[n] = x_1[n]\cos(\pi n)$ and $y_2[n] = x_2[n]\cos(\pi n)$.

using as sampling period T_s. The sampled signal is

$$x[n] = x(nT_s) = e^{(-anT_s + j\Omega_0 nT_s)} = (e^{-aT_s})^n e^{j(\Omega_0 T_s)n} = A\alpha^n e^{j\omega_0 n}$$

where we let $\alpha = e^{-aT_s}$ and $\omega_0 = \Omega_0 T_s$.

2. Just as with the continuous-time complex exponential, we obtain different signals depending on the chosen parameters A and α (see Fig. 9.3 for examples). For instance, the real part of $x[n]$ in (9.16) is a real signal

$$g[n] = \mathcal{R}e(x[n]) = |A||\alpha|^n \cos(\omega_0 n + \theta)$$

when $|\alpha| < 1$ it is a damped sinusoid, and when $|\alpha| > 1$ it is a growing sinusoid (see Fig. 9.3). If $\alpha = 1$ then the above signal is a sinusoid.

3. It is important to realize that for a real $\alpha > 0$ the exponential

$$x[n] = (-\alpha)^n = (-1)^n \alpha^n = \alpha^n \cos(\pi n),$$

i.e., a modulated exponential.

Example 9.12. Given the analog signal $x(t) = e^{-at}\cos(\Omega_0 t)u(t)$, determine the values of $a > 0$, Ω_0, and T_s that permit us to obtain a discrete-time signal

$$y[n] = \alpha^n \cos(\omega_0 n) \qquad n \geq 0$$

and zero otherwise. Consider the case when $\alpha = 0.9$ and $\omega_0 = \pi/2$, find a, Ω_0, and T_s that will permit us to obtain $y[n]$ from $x(t)$ by sampling. Plot $x(t)$ and $y[n]$ using MATLAB.

Solution: Comparing the sampled continuous-time signal $x(nT_s) = (e^{-aT_s})^n \cos((\Omega_0 T_s)n)u[n]$ with $y[n]$ we obtain the following two equations:

$$(i)\ \alpha = e^{-aT_s},\ \ (ii)\ \omega_0 = \Omega_0 T_s$$

with three unknowns (a, Ω_0, and T_s) when α and ω_0 are given, so there is no unique solution. However, we know that according to the Nyquist condition

$$T_s \leq \frac{\pi}{\Omega_{max}}.$$

Assuming the maximum frequency is $\Omega_{max} = N\Omega_0$ for $N \geq 2$ (since the signal $x(t)$ is not band-limited the maximum frequency is not known, to estimate it we could use Parseval's result as indicated in the previous chapter; instead we are assuming that is a multiple of Ω_0) if we let $T_s = \pi/N\Omega_0$ after replacing it in the above equations we get

$$\alpha = e^{-a\pi/N\Omega_0}, \quad \omega_0 = \Omega_0\pi/N\Omega_0 = \pi/N.$$

If we want $\alpha = 0.9$ and $\omega_0 = \pi/2$, we have $N = 2$ and

$$a = -\frac{2\Omega_0}{\pi} \log 0.9$$

for any frequency $\Omega_0 > 0$. For instance, if $\Omega_0 = 2\pi$ then $a = -4 \log 0.9$, and $T_s = 0.25$. Fig. 9.4 displays the continuous and the discrete-time signals generated using the above parameters. The following script is used. The continuous-time and the discrete-time signals coincide at the sample times. □

```
%%
% Example 9.12
%%
a=-4*log(0.9);Ts=0.25;                          % parameters
alpha=exp(-a*Ts);
n=0:30; y=alpha.^n.*cos(pi*n/2);                % discrete-time signal
t=0:0.001:max(n)*Ts; x=exp(-a*t).*cos(2*pi*t);  % analog signal
stem(n,y,'r'); hold on
plot(t/Ts,x); grid; legend('y[n]','x(t)'); hold off
```

Example 9.13. Show how to obtain the casual discrete-time exponential $x[n] = (-1)^n$ for $n \geq 0$ and zero otherwise, by sampling a continuous-time signal $x(t)$.

Solution: Because the values of $x[n]$ are 1 and -1, $x[n]$ cannot be generated by sampling an exponential $e^{-at}u(t)$, indeed $e^{-at} > 0$ for any values of a and t. The discrete signal can be written

$$x[n] = (-1)^n = \cos(\pi n)$$

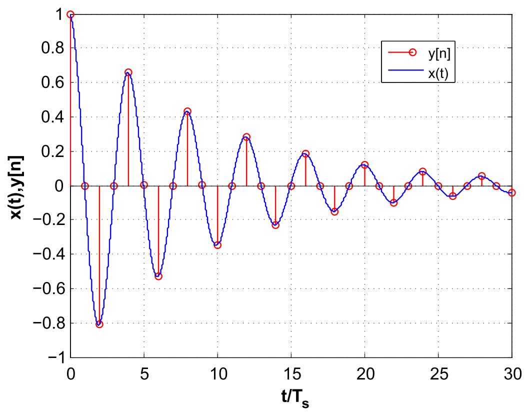

FIGURE 9.4

Determination of parameters for a continuous signal $x(t) = e^{-at}\cos(\Omega_0 t)u(t)$ that when sampled gives a desired discrete-time signal $y[n]$.

for $n \geq 0$ and zero otherwise. If we sample an analog signal $x(t) = \cos(\Omega_0 t)u(t)$ with a sampling period T_s and equate it to the desired discrete signal we get

$$x[n] = x(nT_s) = \cos(\Omega_0 n T_s) = \cos(\pi n)$$

for $n \geq 0$ and zero otherwise. Thus, $\Omega_0 T_s = \pi$ giving $T_s = \pi/\Omega_0$. For instance, for $\Omega_0 = 2\pi$, then $T_s = 0.5$. □

Discrete-Time Sinusoids

A discrete-time sinusoid is a special case of the complex exponential, letting $\alpha = e^{j\omega_0}$ and $A = |A|e^{j\theta}$, we have according to Equation (9.16)

$$x[n] = A\alpha^n = |A|e^{j(\omega_0 n+\theta)} = |A|\cos(\omega_0 n + \theta) + j|A|\sin(\omega_0 n + \theta) \tag{9.17}$$

so the real part of $x[n]$ is a cosine, while the imaginary part is a sine. As indicated before, discrete sinusoids of amplitude A and phase shift θ are periodic if they can be expressed as

$$A\cos(\omega_0 n + \theta) = A\sin(\omega_0 n + \theta + \pi/2) \qquad -\infty < n < \infty \tag{9.18}$$

where $w_0 = 2\pi m/N$ (rad) is the discrete frequency, for integers m and $N > 0$ which are not divisible. Otherwise, discrete-time sinusoids are not periodic.

Because ω is given in radians, it repeats periodically with 2π as fundamental period, i.e., adding a multiple $2\pi k$ (k positive or negative) to a discrete frequency ω_0 gives back the frequency ω_0

$$\omega_0 + 2\pi k = \omega_0 \qquad k \text{ positive or negative integer.} \tag{9.19}$$

To avoid this ambiguity, we will let $-\pi < \omega \leq \pi$ as the possible range of discrete frequencies and convert any frequency ω_1 outside this range using the following modular representation. For a positive

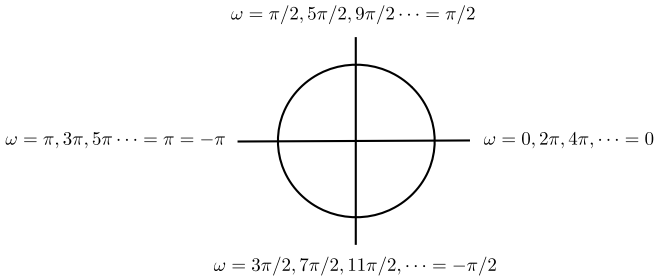

FIGURE 9.5

Discrete frequencies ω and their equivalent frequencies in mod 2π.

integer k two frequencies $\omega_1 = \omega + 2\pi k$, and ω where $-\pi \leq \omega \leq \pi$ are equal modulo 2π, which can be written

$$\omega_1 \equiv \omega \pmod{2\pi}.$$

Thus as shown in Fig. 9.5, frequencies not in the range $[-\pi, \pi)$ can be converted into equivalent discrete frequencies in that range. For instance, $\omega_0 = 2\pi$ can be written as $\omega_0 = 0 + 2\pi$ so that an equivalent frequency is 0; $\omega_0 = 7\pi/2 = (8-1)\pi/2 = 2 \times 2\pi - \pi/2$ is equivalent to a frequency $-\pi/2$. According to this $\pmod{2\pi}$ representation of the frequencies, a signal $\sin(3\pi n)$ is identical to $\sin(\pi n)$, and a signal $\sin(1.5\pi n)$ is identical to $\sin(-0.5\pi n) = -\sin(0.5\pi n)$.

Example 9.14. Consider the following four sinusoids:

$$x_1[n] = \sin(0.1\pi n), \quad x_2[n] = \sin(0.2\pi n), \quad x_3[n] = \sin(0.6\pi n), \quad x_4[n] = \sin(0.7\pi n) \quad -\infty < n < \infty.$$

Find if they are periodic and if so find their fundamental periods. Are these signals harmonically related? Use MATLAB to plot these signals from $n = 0, \cdots, 40$. Comment on which of these signals resemble sampled analog sinusoids and indicate why some do not.

Solution: To find if they are periodic, we rewrite the given signals as

$$\begin{aligned} x_1[n] &= \sin(0.1\pi n) = \sin\left(\frac{2\pi}{20} n\right), \quad x_2[n] = \sin(0.2\pi n) = \sin\left(\frac{2\pi}{20} 2n\right), \\ x_3[n] &= \sin(0.6\pi n) = \sin\left(\frac{2\pi}{20} 6n\right), \quad x_4[n] = \sin(0.7\pi n) = \sin\left(\frac{2\pi}{20} 7n\right) \end{aligned}$$

indicating the signals are periodic of fundamental periods 20, with frequencies harmonically related. When plotting these signals using MATLAB the first two resemble analog sinusoids but not the other two. See Fig. 9.6.

One might think that $x_3[n]$ and $x_4[n]$ look like that because of aliasing, but that is not the case. To obtain $\cos(\omega_0 n)$ we could sample an analog sinusoid $\cos(\Omega_0 t)$ using a sampling period $T_s = 1$ so that according to the Nyquist condition

$$T_s = 1 \leq \frac{\pi}{\Omega_0}$$

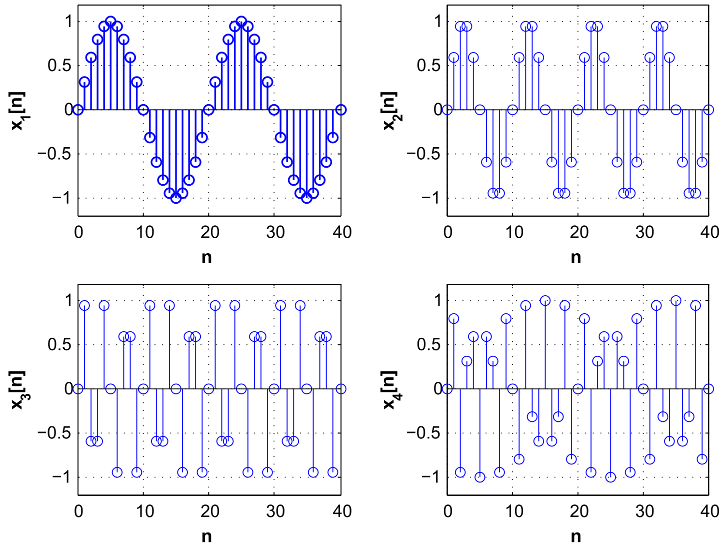

FIGURE 9.6

Periodic signals $x_i[n]$, $i = 1, 2, 3, 4$, given in Example 9.14.

where π/Ω_0 is the maximum value permitted for the sampling period for no aliasing. Thus $x_3[n] = \sin(0.6\pi n) = \sin(0.6\pi t)|_{t=nT_s=n}$ when $T_s = 1$ and in this case

$$T_s = 1 \leq \frac{\pi}{0.6\pi} \approx 1.66.$$

Comparing this with $x_2[n] = \sin(0.2\pi n) = \sin(0.2\pi t)|_{t=nT_s=n}$ we get

$$T_s = 1 \leq \frac{\pi}{0.2\pi} = 5.$$

Thus when obtaining $x_2[n]$ from $\sin(0.2\pi t)$ we are oversampling more than when we obtain $x_3[n]$ from $\sin(0.6\pi t)$ using the same sampling period, as such $x_2[n]$ resembles more an analog sinusoid than $x_3[n]$, but no aliasing is present. Similarly for $x_4[n]$. □

Remarks

1. The discrete-time sine and cosine signals, as in the continuous-time case, are out of phase $\pi/2$ radians.
2. The discrete frequency ω is given in radians since n, the sample index, does not have units. This can also be seen when we sample a sinusoid using a sampling period T_s so that

$$\cos(\Omega_0 t)|_{t=nTs} = \cos(\Omega_0 T_s n) = \cos(\omega_0 n)$$

where we defined $\omega_0 = \Omega_0 T_s$, and since Ω_0 has as unit rad/s and T_s has s as unit, ω_0 has rad as unit.

3. The frequency Ω of analog sinusoids can vary from 0 (dc frequency) to ∞. Discrete frequencies ω as radian frequencies can only vary from 0 to π. Negative frequencies are needed in the analysis of real-valued signals, thus $-\infty < \Omega < \infty$ and $-\pi < \omega \leq \pi$. A discrete-time cosine of frequency 0 is constant for all n, and a discrete-time cosine of frequency π varies from -1 to 1 from sample to sample, giving the largest variation possible for the discrete-time signal.

Discrete-Time Unit-Step and Unit-Sample Signals

The unit-step $u[n]$ and the unit-sample $\delta[n]$ discrete-time signals are defined as

$$u[n] = \begin{cases} 1 & n \geq 0, \\ 0 & n < 0, \end{cases} \tag{9.20}$$

$$\delta[n] = \begin{cases} 1 & n = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{9.21}$$

These two signals are related as follows:

$$\delta[n] = u[n] - u[n-1], \tag{9.22}$$

$$u[n] = \sum_{k=0}^{\infty} \delta[n-k] = \sum_{m=-\infty}^{n} \delta[m]. \tag{9.23}$$

It is easy to see the relation between the two signals $u[n]$ and $\delta[n]$:

$$\begin{aligned} \delta[n] &= u[n] - u[n-1], \\ u[n] &= \delta[n] + \delta[n-1] + \cdots = \sum_{k=0}^{\infty} \delta[n-k] = \sum_{m=-\infty}^{n} \delta[m] \end{aligned}$$

where the last expression[2] is obtained by a change of variable, $m = n - k$. These two equations should be contrasted with the ones for $u(t)$ and $\delta(t)$. Instead of the derivative relation $\delta(t) = du(t)/dt$ we have a difference relation and instead of the integral connection

$$u(t) = \int_0^{\infty} \delta(t - \zeta)d\zeta = \int_{-\infty}^{t} \delta(\tau)d\tau$$

we now have a sum relation between $u[n]$ and $\delta[n]$.

[2]It may not be clear how the second sum gives $u[n]$. For $n < 0$, the sum is of unit-sample signals $\{\delta[m]\}$ for which $m \leq n < 0$ (i.e., the arguments m are all negative and so all are zero) gives $u[n] = 0$, $n < 0$. For $n = n_0 \geq 0$, the sum gives

$$u[n_0] = \cdots + \delta[n_0 - 3] + \delta[n_0 - 2] + \delta[n_0 - 1] + \delta[n_0]$$

so that only one of the unit-sample signals would have an argument of 0 and that one would be equal to 1, the other unit-sample signals would give zero. For instance, $u[2] = \cdots + \delta[-1] + \delta[0] + \delta[1] + \delta[2] = 1$. As such $u[n] = 1$ for $n \geq 0$.

Remarks

1. Notice that there is no ambiguity in the definition of $u[n]$ or of $\delta[n]$ as there is for their continuous-time counterparts $u(t)$ and $\delta(t)$. Moreover, the definitions of these signals do not depend on $u(t)$ or $\delta(t)$; $u[n]$ and $\delta[n]$ are not sampled versions of $u(t)$ and $\delta(t)$.
2. The discrete ramp function $r[n]$ is defined as

$$r[n] = nu[n]. \tag{9.24}$$

As such, it can be expressed in terms of delta and unit-step signals as

$$r[n] = \sum_{k=0}^{\infty} k\delta[n-k] = \sum_{k=0}^{\infty} u[n-k]. \tag{9.25}$$

Moreover, $u[n] = r[n] - r[n-1]$ which can easily be shown.[3] Recall that $dr(t)/dt = u(t)$, so that the derivative is replaced by the difference in the discrete domain.

Generic Representation of Discrete-Time Signals

> Any discrete-time signal $x[n]$ is represented using unit-sample signals as
>
> $$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]. \tag{9.26}$$

The representation of any signal $x[n]$ in terms of $\delta[n]$ results from the **sifting property** of the unit-sample signal:

$$x[n]\delta[n-n_0] = x[n_0]\delta[n-n_0],$$

which is due to

$$\delta[n-n_0] = \begin{cases} 1 & n = n_0, \\ 0 & \text{otherwise.} \end{cases}$$

Thus considering $x[n]$ a sequence of samples

$$\cdots\ x[-1]\ x[0]\ x[1]\ \cdots,$$

---

[3] We have

$$\begin{aligned} r[n] - r[n-1] &= \sum_{k=0}^{\infty} u[n-k] - \sum_{k=0}^{\infty} u[n-(k-1)] \\ &= u[n] + \sum_{k=1}^{\infty} u[n-k] - \sum_{m=1}^{\infty} u[n-m] = u[n] \end{aligned}$$

according to the new variable $m = k + 1$.

at sample times $\cdots\ -1,\ 0,\ 1, \cdots$, we can write $x[n]$ as

$$x[n] = \cdots + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + \cdots = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k].$$

The generic representation (9.26) of any signal $x[n]$ will be useful in finding the output of a discrete-time linear time-invariant system.

Example 9.15. Consider a discrete pulse

$$x[n] = \begin{cases} 1 & 0 \leq n \leq N-1, \\ 0 & \text{otherwise;} \end{cases}$$

obtain representations of $x[n]$ using unit-sample and unit-step signals.

Solution: The signal $x[n]$ can be represented as

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

and using $\delta[n] = u[n] - u[n-1]$ we obtain a representation of the discrete pulse in terms of unit-step signals

$$\begin{aligned} x[n] &= \sum_{k=0}^{N-1}(u[n-k] - u[n-k-1]) \\ &= (u[n] - u[n-1]) + (u[n-1] - u[n-2]) + \cdots (u[n-N+1] - u[n-N]) \\ &= u[n] - u[n-N] \end{aligned}$$

because of the cancellation of consecutive terms. □

Example 9.16. Consider how to generate a periodic train of triangular, discrete-time pulses $t[n]$ of fundamental period $N = 11$. A period of $t[n]$ is

$$\tau[n] = t[n](u[n] - u[n-11]) = \begin{cases} n & 0 \leq n \leq 5, \\ -n+10 & 6 \leq n \leq 10, \\ 0 & \text{otherwise.} \end{cases}$$

Find then an expression for its finite difference $d[n] = t[n] - t[n-1]$.

Solution: The periodic signal can be generated by adding shifted versions of $\tau[n]$, or

$$t[n] = \cdots + \tau[n+11] + \tau[n] + \tau[n-11] + \cdots = \sum_{k=-\infty}^{\infty} \tau[n-11k].$$

The finite difference $d[n]$ is then

$$d[n] = t[n] - t[n-1] = \sum_{k=-\infty}^{\infty} (\tau[n-11k] - \tau[n-11k-1]).$$

The signal $d[n]$ is also periodic of the same fundamental period $N = 11$ as $t[n]$. If we let

$$s[n] = \tau[n] - \tau[n-1] = \begin{cases} 0 & n = 0, \\ 1 & 1 \leq n \leq 5, \\ -1 & 6 \leq n \leq 10, \\ 0 & \text{otherwise,} \end{cases}$$

then

$$d[n] = \sum_{k=-\infty}^{\infty} s[n-11k]. \quad \square$$

Example 9.17. Consider the discrete-time signal

$$y[n] = 3r(t+3) - 6r(t+1) + 3r(t) - 3u(t-3)|_{t=0.15n},$$

obtained by sampling with a sampling period $T_s = 0.15$ a continuous-time signal formed by ramp and unit-step signals. Write MATLAB functions to generate the ramp and the unit-step signals and obtain $y[n]$. Write then a MATLAB function that provides the even and odd decomposition of $y[n]$.

Solution: The signal $y(t)$ is obtained by sequentially adding the different signals as we go from $-\infty$ to ∞:

$$y(t) = \begin{cases} 0 & t < -3, \\ 3r(t+3) = 3t + 9 & -3 \leq t < -1, \\ 3t + 9 - 6r(t+1) = 3t + 9 - 6(t+1) = -3t + 3 & -1 \leq t < 0, \\ -3t + 3 + 3r(t) = -3t + 3 + 3t = 3 & 0 \leq t < 3, \\ 3 - 3u(t-3) = 3 - 3 = 0 & t \geq 3. \end{cases}$$

The three functions *ramp, unitstep*, and *evenodd* for this example are shown below. The following script shows how they can be used to generate the ramp signals, with the appropriate slopes and time shifts, as well as the unit-step signal with the desired delay and then how to compute the even and odd decomposition of $y[n]$:

```
%%
% Example 9.17
%%
Ts=0.15;                                     % sampling period
t=-5:Ts:5;                                   % time support
y1=ramp(t,3,3); y2=ramp(t,-6,1);
y3=ramp(t,3,0);                              % ramp signals
y4=-3*unitstep(t,-3);                        % unit-step signal
y=y1+y2+y3+y4;
[ye,yo]=evenodd(y);
```

We choose as support $-5 \leq t \leq 5$ for the continuous-time signal $y(t)$ which translates into a support $-5 \leq 0.15n \leq 5$ or $-5/0.15 \leq n \leq 5/0.15$ for the discrete-time signal when $T_s = 0.15$. Since the limits are not integers, to make them integers (as required because n is an integer) we use the MATLAB function *floor* to find integers smaller than $-5/0.15$ and $5/0.15$ giving a range $[-34, 32]$. This is used when plotting $y[n]$.

The following function generates a ramp signal for a range of time values, for different slopes and time shifts:

```
function y=ramp(t,m,ad)
% ramp generation
% t: time support
% m: slope of ramp
% ad : advance (positive), delay (negative) factor
N=length(t);
y=zeros(1,N);
for i=1:N,
    if t(i)>=-ad,
    y(i)=m*(t(i)+ad);
    end
end
```

Likewise, the following function generates unit-step signals with different time-shifts (notice the similarities with the *ramp* function).

```
function y=unitstep(t,ad)
% generation of unit step
% t: time support
% ad : advance (positive), delay (negative)
N=length(t);
y=zeros(1,N);
for i=1:N,
    if t(i)>=-ad,
    y(i)=1;
    end
end
```

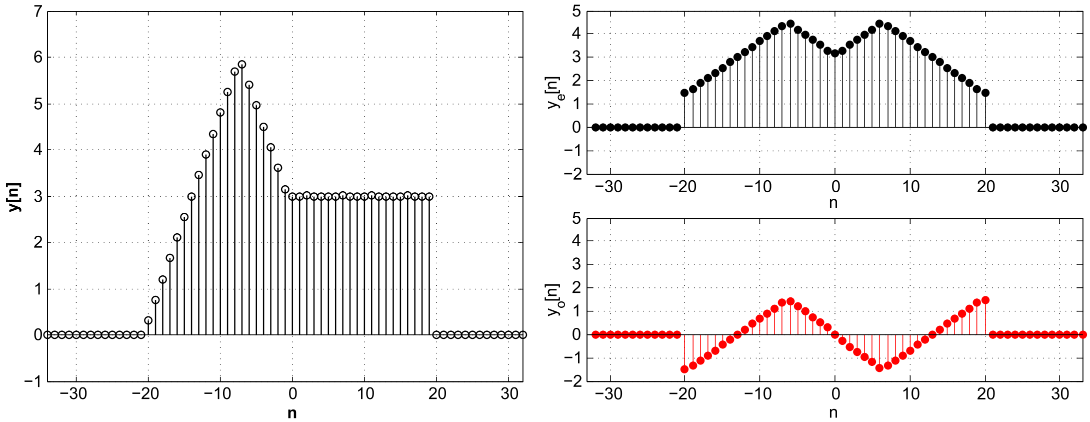

FIGURE 9.7

Discrete-time signal $y[n]$ (left), even $y_e[n]$ and odd $y_o[n]$ components (right).

Finally, the following function can be used to compute the even and the odd decomposition of a discrete-time signal. The MATLAB function *flliplr* reflects the signal as needed in the generation of the even and odd components.

```
function [ye,yo]=evenodd(y)
% even/odd decomposition
% NOTE: the support of the signal should be
%       symmetric about the origin
% y: analog signal
% ye, yo: even and odd components
yr=fliplr(y);      % reflect
ye=0.5*(y+yr);
yo=0.5*(y-yr);
```

The results are shown in Fig. 9.7. □

9.3 DISCRETE-TIME SYSTEMS

Just as with continuous-time systems, a discrete-time system is a transformation of a discrete-time input signal $x[n]$ into a discrete-time output signal $y[n]$, i.e.,

$$y[n] = \mathcal{S}\{x[n]\}. \tag{9.27}$$

We are interested in dynamic systems $\mathcal{S}\{.\}$ having the following properties:

- Linearity
- Time invariance
- Stability
- Causality

just as we were when we studied the continuous-time systems.

A discrete-time system $\mathcal{S}$ is said to be

1. **Linear:** if for inputs $x[n]$ and $v[n]$, and constants a and b, it satisfies the following:
   - **Scaling:** $\mathcal{S}\{ax[n]\} = a\mathcal{S}\{x[n]\}$
   - **Additivity:** $\mathcal{S}\{x[n] + v[n]\} = \mathcal{S}\{x[n]\} + \mathcal{S}\{v[n]\}$

   or equivalently if **superposition** applies, i.e.,

$$\mathcal{S}\{ax[n] + bv[n]\} = a\mathcal{S}\{x[n]\} + b\mathcal{S}\{v[n]\}. \tag{9.28}$$

2. **Time invariant:** if for an input $x[n]$ the corresponding output is $y[n] = \mathcal{S}\{x[n]\}$, the output corresponding to an advanced or a delayed version of $x[n]$, $x[n \pm M]$, for an integer M, is $y[n \pm M] = \mathcal{S}\{x[n \pm M]\}$, or the same as before but shifted as the input. In other words, the system is not changing with time.

Example 9.18 (A Square-Root Computation System). The input–output relation characterizing a discrete-time system is non-linear if there are non-linear terms that include the input, $x[n]$, the output, $y[n]$, or both (e.g., a square root of $x[n]$, products of $x[n]$ and $y[n]$, etc.). Consider the development of an iterative algorithm to compute the square root of a positive real number α. If the result of the algorithm is $y[n]$ as $n \to \infty$, then $y^2[n] = \alpha$ and likewise $y^2[n-1] = \alpha$, thus $y[n] = 0.5(y[n-1] + y[n-1])$ and replacing $y[n-1] = \alpha/y[n-1]$ in this equation, the following difference equation, with some initial condition $y[0]$, can be used to find the square root of α:

$$y[n] = 0.5\left[y[n-1] + \frac{\alpha}{y[n-1]}\right] \qquad n > 0.$$

Find recursively the solution of this difference equation. Use the results of finding the square roots of 4 and 2 to show the system is non-linear. Use MATLAB to solve the difference equation, and then plot the results for $\alpha = 4$, and 2.

Solution: The given difference equation is of first order, non-linear (expanding it you get the product of $y[n]$ with $y[n-1]$ and $y^2[n-1]$, which are non-linear terms) with constant coefficients. This equation can be solved recursively for $n > 0$ by replacing $y[0]$ to get $y[1]$, and we use this to get $y[2]$ and so on, i.e.,

$$y[1] = 0.5\left[y[0] + \frac{\alpha}{y[0]}\right], \quad y[2] = 0.5\left[y[1] + \frac{\alpha}{y[1]}\right], \quad y[3] = 0.5\left[y[2] + \frac{\alpha}{y[2]}\right], \quad \cdots.$$

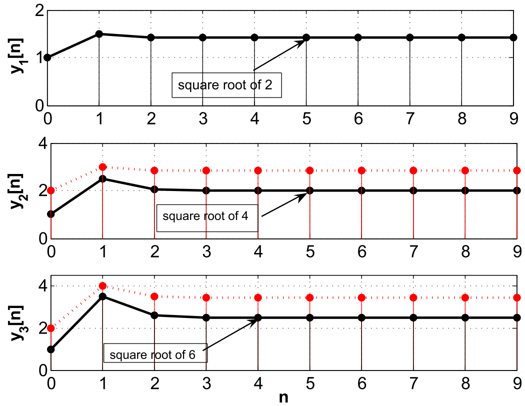

FIGURE 9.8

Square root of 2 (top); square root of 4 compared with twice the square root of 2 (middle); sum of previous responses compared with response of square root of 2 + 4 (bottom). Unity initial condition. Middle figure shows scaling property does not hold, and the bottom figure that additivity property does not hold, either. System is nonlinear.

For instance, let $y[0] = 1$, and $\alpha = 4$ (i.e., we wish to find the square root of 4),

$$y[0] = 1, \quad y[1] = 0.5\left[1 + \frac{4}{1}\right] = 2.5, \quad y[2] = 0.5\left[2.5 + \frac{4}{2.5}\right] = 2.05, \quad \cdots,$$

which is converging to 2, the square root of 4 (see Fig. 9.8). Thus, as indicated before, when $n \to \infty$ then $y[n] = y[n-1] = Y$, for some value Y which according to the difference equation satisfies the relation $Y = 0.5Y + 0.5(4/Y)$ or $Y = \sqrt{4} = 2$.

Suppose then that the input is $\alpha = 2$, half of what it was before. If the system is linear we should get half the previous output according to the scaling property. That is not the case, however. For the same initial condition $y[0] = 1$ we obtain recursively for $\alpha = 2$

$$y[0] = 1, \quad y[1] = 0.5[1 + 2] = 1.5, \quad y[2] = 0.5\left[1.5 + \frac{2}{1.5}\right] = 1.4167, \quad \cdots;$$

this solution is clearly not half of the previous one. Moreover, as $n \to \infty$, we expect $y[n] = y[n-1] = Y$, for Y, that satisfies the relation $Y = 0.5Y + 0.5(2/Y)$ or $Y = \sqrt{2} = 1.4142$, so that the solution is tending to $\sqrt{2}$ and not to 2 as it should if the system were linear. Finally, if we add the signals in the above two cases and compare the resulting signal with the one we obtain when finding the square root of $2 + 4$, they do not coincide. The additive condition is not satisfied either, verifying once more that the system is not linear. See Fig. 9.8. □

9.3.1 RECURSIVE AND NON-RECURSIVE DISCRETE-TIME SYSTEMS

Depending on the relation between the input $x[n]$ and the output $y[n]$, two types of discrete-time systems of interest are:

- **Recursive system**

$$y[n] = -\sum_{k=1}^{N-1} a_k y[n-k] + \sum_{m=0}^{M-1} b_m x[n-m] \qquad n \geq 0,$$
$$\text{initial conditions } \; y[-k],\; k = 1, \cdots, N-1. \tag{9.29}$$

  This system is also called an **infinite impulse response (IIR)** system.
- **Non-recursive system**

$$y[n] = \sum_{m=0}^{M-1} b_m x[n-m]. \tag{9.30}$$

  This system is also called a **finite impulse response (FIR)** system.

The recursive system is analogous to a continuous-time system represented by an ordinary differential equation. For this type of system the discrete-time input $x[n]$ and the discrete-time output $y[n]$ are related by a **difference equation** such as

$$y[n] = -\sum_{k=1}^{N-1} a_k y[n-k] + \sum_{m=0}^{M-1} b_m x[n-m] \qquad n \geq 0,$$
$$\text{initial conditions } \; y[-k],\; k = 1, \cdots, N-1.$$

As in the continuous-time case, if the difference equation is linear, with constant coefficients, zero initial conditions and the input is zero for $n < 0$, then it represents a linear and time-invariant system. For these systems, the output at a present time n, $y[n]$, depends or recurs on past values of the output $\{y[n-k],\; k = 1, \cdots, N-1\}$ and thus they are called recursive. We will see that these systems are also called infinite impulse response or IIR because their impulse responses are typically of infinite length.

On the other hand, if the output $y[n]$ does not depend on previous values of the output, but only on weighted and shifted inputs $\{b_m x[n-m],\; m = 0, \cdots, M-1\}$ the system with an input/output equation such as

$$y[n] = \sum_{m=0}^{M-1} b_m x[n-m]$$

is called non-recursive. We will see that the impulse response of non-recursive systems is of finite length, and as such these systems are also called finite impulse response or FIR.

Example 9.19 (Moving-Average Discrete System). A third-order moving-average system (also called a **smoother** as it smooths out the input signal) is an FIR system for which the input $x[n]$ and the output

$y[n]$ are related by

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2]).$$

Show that this system is linear and time invariant.

Solution: This is a non-recursive system that uses a present sample, $x[n]$, and two past values $x[n-1]$ and $x[n-2]$ to get an average $y[n]$ at every n. Thus its name of moving-average system.

Linearity—If we let the input be $ax_1[n] + bx_2[n]$ and assume that $\{y_i[n],\ i = 1, 2\}$ are the corresponding outputs to $\{x_i[n],\ \ i = 1, 2\}$, the system output is

$$\frac{1}{3}[(ax_1[n] + bx_2[n]) + (ax_1[n-1] + bx_2[n-1]) + (ax_1[n-2] + bx_2[n-2])] = ay_1[n] + by_2[n],$$

thus linear.

Time invariance—If the input is $x_1[n] = x[n-N]$ the corresponding output to it is

$$\frac{1}{3}(x_1[n] + x_1[n-1] + x_1[n-2]) = \frac{1}{3}(x[n-N] + x[n-N-1] + x[n-N-2]) = y[n-N],$$

i.e., the system is time invariant. □

Example 9.20 (Autoregressive Discrete System). The recursive discrete-time system represented by the following first-order difference equation (with initial condition $y[-1]$):

$$y[n] = ay[n-1] + bx[n] \qquad n \geq 0$$

is called an **autoregressive (AR)** system. "Autoregressive" refers to the feedback in the output, i.e., the present value of the output $y[n]$ depends on its previous value $y[n-1]$. Find recursively the solution of the difference equation and determine under what conditions the system represented by this difference equation is linear and time invariant.

Solution: Let us first discuss why the initial condition is $y[-1]$. The initial condition is the value needed to compute $y[0]$. According to the difference equation to compute

$$y[0] = ay[-1] + bx[0]$$

we need the initial condition $y[-1]$ since $x[0]$ is known.

Assume that the initial condition $y[-1] = 0$, and that the input $x[n] = 0$ for $n < 0$, i.e., the system is not energized for $n < 0$. The solution of the difference equation, when the input $x[n]$ is not defined can be found by a repetitive substitution of the input/output relationship. Thus replacing $y[n-1] = ay[n-2] + bx[n-1]$ in the difference equation, and then letting $y[n-2] = ay[n-3] + bx[n-2]$, replacing it and so on, we obtain

$$\begin{aligned} y[n] &= a(ay[n-2] + bx[n-1]) + bx[n] = a(a(ay[n-3] + bx[n-2])) + abx[n-1] + bx[n] \\ &\cdots = bx[n] + abx[n-1] + a^2bx[n-2] + a^3bx[n-3] + \cdots \end{aligned}$$

until we reach a term with $x[0]$. The solution can be written as

$$y[n] = \sum_{k=0}^{n} ba^k x[n-k] \qquad n \geq 0, \tag{9.31}$$

which we will see in the next section is the **convolution sum** of the **impulse response** of the system and the input.

To verify that (9.31) is actually the solution of the above difference equation, we need to show that when replacing the above expression for $y[n]$ in the right-hand term of the difference equation we obtain the left-hand term $y[n]$. Indeed, we have

$$\begin{aligned} ay[n-1] + bx[n] &= a\left[\sum_{k=0}^{n-1} ba^k x[n-1-k]\right] + bx[n] = \sum_{m=1}^{n} ba^m x[n-m] + bx[n] \\ &= \sum_{m=0}^{n} ba^m x[n-m] = y[n] \end{aligned}$$

where the dummy variable k in the first sum was changed to $m = k+1$, so that the limits of the summation became $m = 1$ when $k = 0$, and $m = n$ when $k = n-1$. The final equation is identical to $y[n]$.

To establish if the system represented by the difference equation is linear, we use the solution (9.31) with input $x[n] = \alpha x_1[n] + \beta x_2[n]$, where the outputs $\{y_i[n],\ i = 1, 2\}$ correspond to inputs $\{x_i[n],\ i = 1, 2\}$, and α, β are constants. The output for $x[n]$ is

$$\begin{aligned} \sum_{k=0}^{n} ba^k x[n-k] &= \sum_{k=0}^{n} ba^k \left(\alpha x_1[n-k] + \beta x_2[n-k]\right) \\ &= \alpha \sum_{k=0}^{n} ba^k x_1[n-k] + \beta \sum_{k=0}^{n} ba^k x_2[n-k] = \alpha y_1[n] + \beta y_2[n]. \end{aligned}$$

So the system is linear.

The time invariance is shown by letting the input be $v[n] = x[n-N]$, $n \geq N$, and zero otherwise. The corresponding output according to (9.31) is

$$\begin{aligned} \sum_{k=0}^{n} ba^k v[n-k] &= \sum_{k=0}^{n} ba^k x[n-N-k] \\ &= \sum_{k=0}^{n-N} ba^k x[n-N-k] + \sum_{k=n-N+1}^{n} ba^k x[n-N-k] = y[n-N], \end{aligned}$$

since the summation

$$\sum_{k=n-N+1}^{n} ba^k x[n-N-k] = 0$$

given that $x[-N] = \cdots = x[-1] = 0$, as assumed. Thus the system represented by the above difference equation is linear and time invariant. As in the continuous-time case, however, if the initial condition $y[-1]$ is not zero, or if $x[n] \neq 0$ for $n < 0$ the system characterized by the difference equation is not LTI. □

Example 9.21 (Autoregressive Moving-Average System). A recursive system represented by the first-order difference equation

$$y[n] = 0.5y[n-1] + x[n] + x[n-1] \qquad n \geq 0,\ \ y[-1]$$

is called an **autoregressive moving-average** system given that it is the combination of the two systems discussed before. Consider two cases:

- Let the initial condition be $y[-1] = -2$, and the input $x[n] = u[n]$ first and then $x[n] = 2u[n]$. Find the corresponding outputs.
- Let the initial condition be $y[-1] = 0$, and the input $x[n] = u[n]$ first and then $x[n] = 2u[n]$. Find the corresponding outputs.

Use the above results to determine in each case if the system is linear. Find the steady-state response, i.e., $\lim_{n\to\infty} y[n]$.

Solution: For an initial condition $y[-1] = -2$ and $x[n] = u[n]$ we get recursively

$$\begin{aligned}&y[0] = 0.5y[-1] + x[0] + x[-1] = 0, \quad y[1] = 0.5y[0] + x[1] + x[0] = 2,\\ &y[2] = 0.5y[1] + x[2] + x[1] = 3, \quad \cdots .\end{aligned}$$

Let us then double the input, i.e., $x[n] = 2u[n]$, and call the response $y_1[n]$. As the initial condition remains the same, i.e., $y_1[-1] = -2$ we get

$$\begin{aligned}&y_1[0] = 0.5y_1[-1] + x[0] + x[-1] = 1, \quad y_1[1] = 0.5y_1[0] + x[1] + x[0] = 4.5,\\ &y_1[2] = 0.5y_1[1] + x[2] + x[1] = 6.25, \quad \cdots .\end{aligned}$$

It is clear that the response $y_1[n]$ is not $2y[n]$. Due to the initial condition not being zero, the system is non-linear.

If the initial condition is set to zero, and the input $x[n] = u[n]$, the response is

$$\begin{aligned}&y[0] = 0.5y[-1] + x[0] + x[-1] = 1, \quad y[1] = 0.5y[0] + x[1] + x[0] = 2.5\\ &y[2] = 0.5y[1] + x[2] + x[1] = 3.25, \quad \cdots\end{aligned}$$

and if we double the input, i.e., $x[n] = 2u[n]$, and call the response $y_1[n]$, $y_1[-1] = 0$, we obtain

$$\begin{aligned}&y_1[0] = 0.5y_1[-1] + x[0] + x[-1] = 2, \quad y_1[1] = 0.5y_1[0] + x[1] + x[0] = 5,\\ &y_1[2] = 0.5y_1[1] + x[2] + x[1] = 6.5, \quad \cdots .\end{aligned}$$

For the zero initial condition, it is clear that $y_1[n] = 2y[n]$ when we double the input. One can also show that superposition holds for this system. For instance if we let the input be the sum of the previous inputs, $x[n]=u[n]+2u[n]=3u[n]$ and let $y_{12}[n]$ be the response when the initial condition is zero, $y_{12}[0]=0$, we obtain

$$y_{12}[0]=0.5y_{12}[-1]+x[0]+x[-1]=3, \quad y_{12}[1]=0.5y_{12}[0]+x[1]+x[0]=7.5$$
$$y_{12}[2]=0.5y_{12}[1]+x[2]+x[1]=9.75, \quad \cdots,$$

showing that $y_{12}[n]$ is the sum of the responses for inputs $u[n]$ and $2u[n]$. Thus, the system represented by the given difference equation with a zero initial condition is linear.

Although when the initial condition is zero and $x[n]=u[n]$ we cannot find a closed form for the response, we can see that the response is going toward a final value or a steady-state response. Assuming that as $n\to\infty$ we have $Y=y[n]=y[n-1]$ and since $x[n]=x[n-1]=1$, according to the difference equation the steady-state value Y is found from

$$Y=0.5Y+2 \quad \text{or} \quad Y=4.$$

For this system, the steady-state response is independent of the initial condition. Likewise, when $x[n]=2u[n]$, the steady-state solution Y is obtained from $Y=0.5Y+4$ or $Y=8$, and again independent of the initial condition. □

Remarks

1. Like in the continuous-time, to show that a discrete-time system is linear and time invariant an explicit expression relating the input and the output is needed, i.e., the output should be expressed as a function of the input only.
2. Although the solution of linear difference equations can be obtained in the time domain, just like with ordinary differential equations, we will see that their solution can also be obtained using the Z-transform, just like the Laplace transform is used to solve linear ordinary differential equations.

9.3.2 DYNAMIC DISCRETE-TIME SYSTEMS REPRESENTED BY DIFFERENCE EQUATIONS

A recursive discrete-time system is represented by a difference equation

$$y[n]=-\sum_{k=1}^{N-1}a_k y[n-k]+\sum_{m=0}^{M-1}b_m x[n-m] \qquad n\geq 0,$$
$$\text{initial conditions } \; y[-k],\; k=1,\cdots,N-1, \tag{9.32}$$

characterizing the dynamics of the discrete-time system. This difference equation could be the approximation of an ordinary differential equation representing a continuous-time system being processed discretely. For instance, to approximate a second-order ordinary differential equation by a difference equation, we could approximate the first derivative as

$$\frac{dv_c(t)}{dt}\approx\frac{v_c(t)-v_c(t-T_s)}{T_s}$$

and the second derivative as

$$\frac{d^2v_c(t)}{dt^2} = \frac{d(dv_c(t)/dt)}{dt} \approx \frac{d((v_c(t) - v_c(t-T_s))/T_s)}{dt} = \frac{v_c(t) - 2v_c(t-T_s) + v_c(t-2T_s)}{T_s^2}$$

to obtain a second-order difference equation. Choosing a small value for T_s provides an accurate approximation to the ordinary differential equation. Other transformations can be used; in Chapter 0 we indicated that approximating integrals by the trapezoidal rule gives the **bilinear transformation** which can also be used to change differential into difference equations.

Just as in the continuous-time case, the system being represented by the difference equation is not LTI unless the initial conditions are zero and the input is causal. The complete response of a system represented by the difference equation can be shown to be composed of **zero-input** and **zero-state** responses, i.e., if $y[n]$ is the solution of the difference equation (9.32) with initial conditions not necessarily equal to zero, then

$$y[n] = y_{zi}[n] + y_{zs}[n]. \tag{9.33}$$

The component $y_{zi}[n]$ is the response when the input $x[n]$ is set to zero, thus it is completely due to the initial conditions. The response $y_{zs}[n]$ is due to the input only, as we set the initial conditions to zero. The complete response $y[n]$ is thus seen as the superposition of these two responses. The Z-transform provides an algebraic way to obtain the complete response, whether the initial conditions are zero or not. It is important, as in continuous time, to differentiate the zero-input and the zero-state responses from the **transient** and the **steady-state** responses. Examples illustrating how to obtain these responses using the Z-transform will be given in the next chapter.

9.3.3 THE CONVOLUTION SUM

Let $h[n]$ be the **impulse response** of a linear time-invariant (LTI) discrete-time system, or the output of the system corresponding to an impulse $\delta[n]$ as input, and initial conditions (if needed) equal to zero.

Using the **generic representation** of the input $x[n]$ of the LTI system

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \tag{9.34}$$

the output of the LTI system is given by either of the following equivalent forms of the **convolution sum**:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{m=-\infty}^{\infty} x[n-m]h[m]. \tag{9.35}$$

The impulse response $h[n]$ of a recursive LTI discrete-time system is due exclusively to an input $\delta[n]$, as such the initial conditions are set to zero. Clearly no initial conditions are needed when finding the impulse response of non-recursive systems, as no recursion exists, just the input being $\delta[n]$ is needed.

Now, if $h[n]$ is the response due to $\delta[n]$, by time invariance the response to $\delta[n-k]$ is $h[n-k]$. By superposition, the response due to $x[n]$ with the generic representation

$$x[n] = \sum_k x[k]\delta[n-k]$$

is the sum of responses due to $x[k]\delta[n-k]$ which is $x[k]h[n-k]$ ($x[k]$ is not a function of n) or

$$y[n] = \sum_k x[k]h[n-k]$$

or the convolution sum of the input $x[n]$ with the impulse response $h[n]$ of the system. The second expression of the convolution sum in Equation (9.35) is obtained by a change of variable $m = n-k$.

Remarks

1. The output of non-recursive or FIR systems is the convolution sum of the input and the impulse response of the system. Indeed, if the input/output expression of an FIR system is

$$y[n] = \sum_{k=0}^{N-1} b_k x[n-k] \tag{9.36}$$

its impulse response is found by letting $x[n] = \delta[n]$, which gives

$$h[n] = \sum_{k=0}^{N-1} b_k \delta[n-k] = b_0\delta[n] + b_1\delta[n-1] + \cdots + b_{N-1}\delta[n-(N-1)]$$

so that $h[n] = b_n$ for $n = 0, \cdots, N-1$ and zero otherwise. Now, replacing the b_k coefficients in (9.36) by $h[k]$ we find that the output can be written

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

or the convolution sum of the input and the impulse response. This is a very important result, indicating that FIR systems are obtained by means of the convolution sums rather than difference equations, which gives great significance to the efficient computation of the convolution sum.

2. Considering the convolution sum as an operator, i.e.,

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

it is easily shown to be linear. Indeed, whenever the input is $ax_1[n]+bx_2[n]$, and $\{y_i[n]\}$ are the outputs corresponding to $\{x_i[n]\}$ for $i=1,2$, then we have

$$\begin{aligned}[h*(ax_1+bx_2)][n] &= \sum_k (ax_1[k]+bx_2[k])h[n-k] \\ &= a\sum_k x_1[k]h[n-k]+b\sum_k x_2[k]h[n-k] \\ &= a[h*x_1][n]+b[h*x_2][n]=ay_1[n]+by_2[n]\end{aligned}$$

as expected since the system was assumed to be linear when the expression for the convolution sum was obtained.

We also see that if the output corresponding to $x[n]$ is $y[n]$, given by the convolution sum, then the output corresponding to a shifted version of the input, $x[n-N]$ should be $y[n-N]$. In fact, if we let $x_1[n]=x[n-N]$ the corresponding output is

$$\begin{aligned}[h*x_1][n] &= \sum_k x_1[n-k]h[k]=\sum_k x[n-N-k]h[k] \\ &= [h*x][n-N]=y[n-N];\end{aligned}$$

again this result is expected given that the system was considered time invariant when the convolution sum was obtained.

3. From the equivalent representations for the convolution sum we have

$$[h*x][n]=\sum_k x[k]h[n-k]=\sum_k x[n-k]h[k]=[x*h][n]$$

indicating that the convolution commutes with respect to the input $x[n]$ and the impulse response $h[n]$.

4. Just as with continuous-time systems, when connecting two LTI discrete-time systems (with impulse responses $h_1[n]$ and $h_2[n]$) in cascade or in parallel, their respective impulse responses are given by $[h_1*h_2][n]$ and $h_1[n]+h_2[n]$, respectively. See Fig. 9.9 for the block diagrams. In particular notice that in the cascade connection we can interchange the order of the systems with not change on the output. This is due to the systems being LTI, such an interchange is not valid for non-linear or time-varying systems.

5. There are situations when instead of giving the input and the impulse response to compute the output, the information available is, for instance, the input and the output and we wish to find the impulse response of the system, or we have the output and the impulse response and wish to find the input. This type of problem is called **deconvolution**. We will consider this problem later in this chapter after considering causality, and in the next chapter where we will show that the deconvolution problem can easily be solved using the Z-transform.

6. The computation of the convolution sum is typically difficult. It is made easier when the Z-transform is used as we will see. MATLAB provides the function *conv* to compute the convolution sum.

Example 9.22. Consider a moving-average system

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

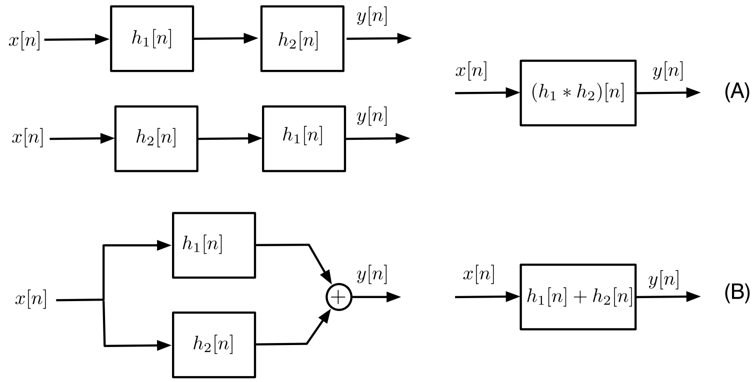

FIGURE 9.9

Cascade (A) and parallel (B) connections of LTI systems with impulse responses $h_1[n]$ and $h_2[n]$. Equivalent systems on the right. Notice the interchange of systems in the cascade connection.

where the input is $x[n]$ and the output is $y[n]$. Find its impulse response $h[n]$. Then:

1. Let $x[n] = u[n]$, find the output $y[n]$ using the input–output relation and the convolution sum.
2. If the input is $x[n] = A\cos(2\pi n/N)u[n]$, determine the values of A, and N, so that the steady-state response of the system is zero. Explain. Use MATLAB to verify your results.

Solution: If the input is $x[n] = \delta[n]$, the output is $y[n] = h[n]$ or the impulse response of the system. No initial conditions are needed. We thus have

$$h[n] = \frac{1}{3}(\delta[n] + \delta[n-1] + \delta[n-2])$$

so that $h[0] = h[1] = h[2] = 1/3$ and $h[n] = 0$ for $n \neq 0,\ 1,\ 2$. Notice that the coefficients of the filter equal the impulse response at $n = 0$, 1, and 2.

For an input $x[n]$ such that $x[n] = 0$ for $n < 0$, let us find a few values of the convolution sum to see what happens as n grows. If $n < 0$, the arguments of $x[n]$, $x[n-1]$ and $x[n-2]$ are negative giving zero values, and so the output is also zero, i.e., $y[n] = 0$, $n < 0$. For $n \geq 0$ we have

$$\begin{aligned}
y[0] &= \frac{1}{3}(x[0] + x[-1] + x[-2]) = \frac{1}{3}x[0],\\
y[1] &= \frac{1}{3}(x[1] + x[0] + x[-1]) = \frac{1}{3}(x[0] + x[1]),\\
y[2] &= \frac{1}{3}(x[2] + x[1] + x[0]) = \frac{1}{3}(x[0] + x[1] + x[2]),\\
y[3] &= \frac{1}{3}(x[3] + x[2] + x[1]) = \frac{1}{3}(x[1] + x[2] + x[3]),\\
&\cdots.
\end{aligned}$$

Thus if $x[n] = u[n]$ then we see that $y[0] = 1/3$, $y[1] = 2/3$, and $y[n] = 1$ for $n \geq 2$.

Noticing that, for $n \geq 2$, the output is the average of the present and past two values of the input, when the input is $x[n] = A\cos(2\pi n/N)$ if we let $N = 3$, and A be any real value the input repeats every 3 samples and the local average of 3 of its values is zero, giving $y[n] = 0$ for $n \geq 2$. Thus, the steady-state response will be zero.

The following MATLAB script uses the function *conv* to compute the convolution sum when the input is either $x[n] = u[n]$ or $x[n] = \cos(2\pi n/3)u[n]$.

```
%%
% Example 9.22---Convolution sum
%%
clear all; clf
x1=[0 0 ones(1,20)]                    % unit-step input
n=-2:19; n1=0:19;
x2=[0 0 cos(2*pi*n1/3)];               % cosine input
h=(1/3)*ones(1,3);                     % impulse response
y=conv(x1,h); y1=y(1:length(n));       % convolution sums
y=conv(x2,h); y2=y(1:length(n));
```

Notice that each of the input sequences has two zeros at the beginning so that the response can be found at $n \geq -2$. Also, when the input is of infinite support, we can only approximate it as a finite sequence in MATLAB and as such the final values of the convolution obtained from *conv* are not correct and should not be considered—in this case, the final two values of the convolution results are not correct and are not considered. The results are shown in Fig. 9.10. □

Example 9.23. Consider an autoregressive system represented by a first-order difference equation

$$y[n] = 0.5y[n-1] + x[n] \qquad n \geq 0.$$

Find the impulse response $h[n]$ of the system and then compute the response of the system to $x[n] = u[n] - u[n-3]$ using the convolution sum. Verify the results with MATLAB.

Solution: The impulse response $h[n]$ can be found recursively. Letting $x[n] = \delta[n]$, $y[n] = h[n]$ and initial condition $y[-1] = h[-1] = 0$ we have

$$h[0] = 0.5h[-1] + \delta[0] = 1, \quad h[1] = 0.5h[0] + \delta[1] = 0.5,$$
$$h[2] = 0.5h[1] + \delta[2] = 0.5^2, \quad h[3] = 0.5h[2] + \delta[3] = 0.5^3, \quad \cdots,$$

from which the general expression for the impulse response is $h[n] = 0.5^n u[n]$.

The response to $x[n] = u[n] - u[n-3]$ using the convolution sum is then given by

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} (u[k] - u[k-3])0.5^{n-k}u[n-k].$$

Since as functions of k, $u[k]u[n-k] = 1$ for $0 \leq k \leq n$, zero otherwise, and $u[k-3]u[n-k] = 1$ for $3 \leq k \leq n$, zero otherwise (in the two cases, draw the two signals as functions of k and verify this

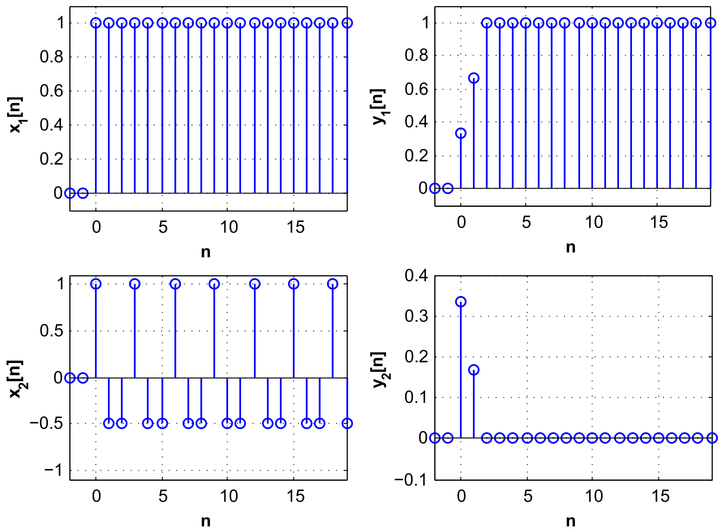

FIGURE 9.10

Convolution sums for a moving-averaging system $y[n] = (x[n] + x[n-1] + x[n-2])/3$ with inputs $x_1[n] = u[n]$ (top) and $x_2[n] = \cos(2\pi n/3)u[n]$ (bottom).

is true), $y[n]$ can be expressed as

$$y[n] = 0.5^n \left[\sum_{k=0}^{n} 0.5^{-k} - \sum_{k=3}^{n} 0.5^{-k}\right] u[n] = \begin{cases} 0 & n < 0, \\ 0.5^n \sum_{k=0}^{n} 0.5^{-k} = 0.5^n(2^{n+1} - 1) & n = 0, 1, 2, \\ 0.5^n \sum_{k=0}^{2} 0.5^{-k} = 7(0.5^n) & n \geq 3. \end{cases}$$

Another way to solve this problem is to notice that the input can be rewritten as

$$x[n] = \delta[n] + \delta[n-1] + \delta[n-2]$$

and since the system is LTI, the output can be written

$$y[n] = h[n] + h[n-1] + h[n-2] = 0.5^n u[n] + 0.5^{n-1} u[n-1] + 0.5^{n-2} u[n-2],$$

which gives zero for $n < 0$ and

$$y[0] = 0.5^0 = 1, \quad y[1] = 0.5^1 + 0.5^0 = \frac{3}{2},$$
$$y[2] = 0.5^2 + 0.5^1 + 0.5^0 = \frac{7}{4}, \quad y[3] = 0.5^3 + 0.5^2 + 0.5 = \frac{7}{8}, \quad \cdots,$$

which coincides with the above more general solution. It should be noticed that even in a simple example like this the computation required by the convolution sum is quite high. We will see that the

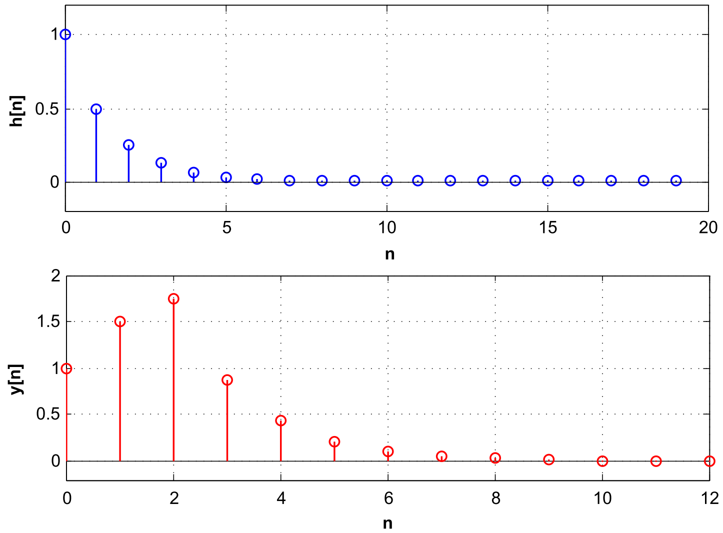

FIGURE 9.11

Impulse response $h[n]$ of first-order autoregressive system $y[n] = 0.5y[n-1] + x[n],\ n \geq 0$, (top), and response $y[n]$ due to $x[n] = u[n] - u[n-3]$ (bottom).

Z-transform reduces the computational load, just like the Laplace transform does in the computation of the convolution integral.

The following MATLAB script is used to verify the above results. The MATLAB function *filter* is used to compute the impulse response and the response of the filter to the pulse. The output obtained with *filter* coincided with the output found using *conv*, as it should. Fig. 9.11 displays the results. □

```
%%
% Example 9.23
%%
a=[1 -0.5];b=1;                   % coefficients
d=[1 zeros(1,99)];                % approximate delta function
h=filter(b,a,d);                  % impulse response
x=[ones(1,3) zeros(1,10)];                % input
y=filter(b,a,x);                          % output from filter function
y1=conv(h,x); y1=y1(1:length(y))          % output from conv
```

9.3.4 LINEAR AND NON-LINEAR FILTERING WITH MATLAB

A recursive or a non-recursive discrete-time system can be used to get rid of undesirable components in a signal. These systems are called linear filters. In this section, we illustrate the use and possible advantages of non-linear filters.

Linear Filtering

To illustrate the way a linear filter works, consider getting rid of a random disturbance $\eta[n]$, which we model as Gaussian noise (this is one of the possible noise signals MATLAB provides) that has been added to a sinusoid $x[n] = \cos(\pi n/16)$. Let $y[n] = x[n] + \eta[n]$. We will use an averaging filter having an input/output equation

$$z[n] = \frac{1}{M} \sum_{k=0}^{M-1} y[n-k].$$

This Mth-order filter averages M past input values $\{y[n-k], k = 0, \cdots, M-1\}$ and assigns this average to the output $z[n]$. The effect is to smooth out the input signal by attenuating the high-frequency components of the signal due to the noise. The larger the value of M the better the results, but at the expense of more complexity and a larger delay in the output signal (this is due to the linear phase frequency response of the filter, as we will see later).

We use 3rd- and 15th-order filters, implemented by our function *averager* given below. The denoising is done by means of the following script:

```
%%
% Linear filtering
%%
N=200;n=0:N-1;
x=cos(pi*n/16);            % input signal
noise=0.2*randn(1,N);      % noise
y=x+noise;                 % noisy signal
z=averager(3,y);           % averaging linear filter with M=3
z1=averager(15,y);         % averaging linear filter with M=15
```

Our function *averager* defines the coefficients of the averaging filter and then uses the MATLAB function *filter* to compute the filter response. The inputs of *filter* are the vector $\mathbf{b} = (1/M)[1 \cdots 1]$, or the coefficients connected of the numerator, the coefficient of the denominator (1), and $\mathbf{x}$ a vector with entries the signal samples we wish to filter. The results of filtering using these two filters are shown in Fig. 9.12. As expected the performance of the filter with $M = 15$ is a lot better, but a delay of 8 samples (or the integer larger than $M/2$) is shown in the filter output.

```
function y=averager(M,x)
% Moving average of signal x
%  M: order of averager
%  x: input signal
%
b=(1/M)*ones(1,M);
y=filter(b,1,x);
```

Non-linear Filtering

Is linear filtering always capable of getting rid of noise? The answer is: It depends on the type of noise. In the previous example we showed that a high-order averaging filter, which is linear, performs well

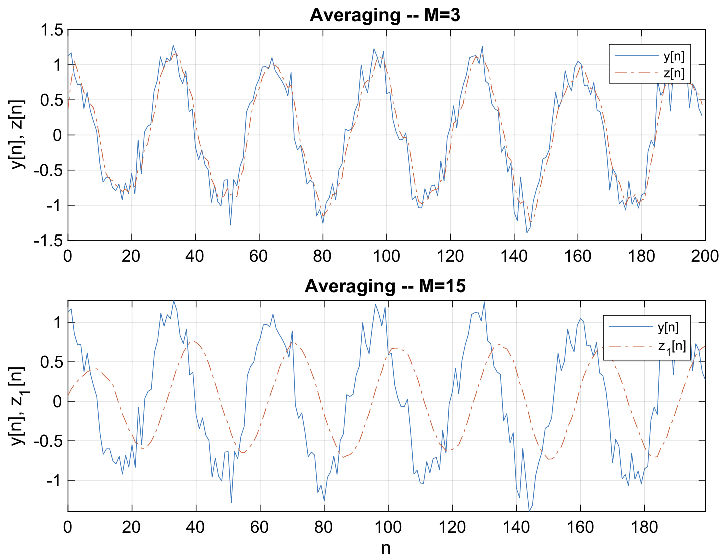

FIGURE 9.12

Averaging filtering with filters of order $M = 3$ (top figure), and of order $M = 15$ result (bottom figure) used to get rid of Gaussian noise added to a sinusoid $x[n] = \cos(\pi n/16)$. Solid line corresponds to the noisy signal, while the dashed line is for the filtered signal. The filtered signal is very much like the noisy signal (see top figure) when $M = 3$ is the order of the filter, while the filtered signal looks like the sinusoid, but shifted, when $M = 15$.

for Gaussian noise. Let us now consider an **impulsive** noise that is either zero or a certain value at random. This is the type of noise occurring in communications whenever cracking sounds are heard in the transmission, or the "salt-and-pepper" noise that appears in images.

It will be shown that even the 15th-order averager—which did well before—is not capable of denoising the signal with impulsive noise. A **median filter** considers a certain number of samples (the example shows the case of a 5th-order median filter), orders them according to their values and chooses the one in the middle (i.e., the median) as the output of the filter. Such a filter is non-linear as it does not satisfy superposition. The following script is used to filter the noisy signal using a linear and a non-linear filter. The results shown in Fig. 9.13. In this case the non-linear filter is able to denoise the signal much better than the linear filter.

```
%%
% Linear and non-linear filtering
%%
clear all; clf
N=200;n=0:N-1;
% impulsive noise
for m=1:N,
    d=rand(1,1);
```

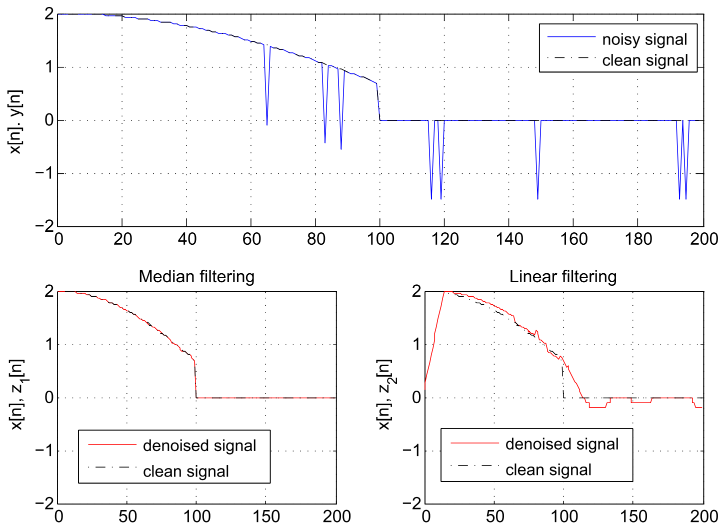

FIGURE 9.13

Non-linear 5th-order median filtering (bottom left) versus linear 15th-order averager (bottom right) corresponding to the noisy signal (dash line) and clean signal (solid line) on top plots. Clean signal (solid line) is superposed on de-noised signal (dashed line) in the bottom figures. Linear filter does not perform as well as the non-linear filter.

```
        if d>=0.95,
            noise(m)=-1.5;
         else
            noise(m)=0 ;
         end
     end
    x=[2*cos(pi*n(1:100)/256) zeros(1,100)];
    y1=x+noise;
    % linear filtering
    z2=averager(15,y1);
    % non-linear filtering -- median filtering
    z1(1)=median([0 0 y1(1) y1(2) y1(3)]);
    z1(2)=median([0 y1(1) y1(2) y1(3) y1(4)]);
    z1(N-1)=median([y1(N-3) y1(N-2) y1(N-1) y1(N) 0]);
    z1(N)=median([y1(N-2) y1(N-1) y1(N) 0 0]);
    for k=3:N-2,
       z1(k)=median([y1(k-2) y1(k-1) y1(k) y1(k+1) y1(k+2)]);
    end
```

Although the theory of non-linear filtering is beyond the scope of this book, it is good to remember that in cases like this when linear filters do not seem to do well, there are other methods to use.

9.3.5 CAUSALITY AND STABILITY OF DISCRETE-TIME SYSTEMS

As with continuous-time systems, two additional independent properties of discrete-time systems are causality and stability. Causality relates to the conditions under which computation can be performed in real-time, while stability relates to the usefulness of the system.

Causality

In many situations signals need to be processed in **real-time**, i.e., the processing must be done as the signal comes into the system. In those situations, the system must be causal. Whenever the data can be stored, not a real-time situation, is not necessary to use a causal system.

> A discrete-time system $\mathcal{S}$ is **causal** if:
>
> - whenever the input $x[n]=0$, and there are no initial conditions, the output is $y[n]=0$,
> - the present output $y[n]$ does not depend on future inputs.

Causality is independent of the linearity and time-invariance properties of a system. For instance, the system represented by the input/output equation

$$y[n]=x^2[n],$$

where $x[n]$ is the input and $y[n]$ the output, is non-linear, time invariant, and according to the above definition causal: the output is zero whenever the input is zero, and the output depends on the present value of the input. Likewise, an LTI system can be noncausal, such is the case of the following LTI system that computes the moving average of the input:

$$y[n]=\frac{1}{3}(x[n+1]+x[n]+x[n-1]).$$

The input/output equation indicates that at the present time n to compute $y[n]$ we need a present value $x[n]$, a past value $x[n-1]$, and a future value $x[n+1]$ of the input. Thus, the system is LTI but noncausal since it requires future values of the input.

> - An LTI discrete-time system is **causal** if the impulse response of the system is such that
>
> $$h[n]=0 \qquad n<0. \tag{9.37}$$
>
> - A signal $x[n]$ is said to be **causal** if
>
> $$x[n]=0 \qquad n<0. \tag{9.38}$$
>
> - For a causal LTI discrete-time system with a causal input $x[n]$ its output $y[n]$ is given by
>
> $$y[n]=\sum_{k=0}^{n} x[k]h[n-k] \qquad n\geq 0 \tag{9.39}$$
>
> where the lower limit of the sum depends on the input causality, $x[k]=0$ for $k<0$, and the upper limit on the causality of the system, $h[n-k]=0$ for $n-k<0$ or $k>n$.

That $h[n] = 0$ for $n < 0$ is the condition for an LTI discrete-time system to be causal is understood by considering that when computing the impulse response, the input $\delta[n]$ only occurs at $n = 0$ and there are no initial conditions so the response for $n < 0$ should be zero. Extending the notion of causality to signals we can then see that the output of a causal LTI discrete-time system can be written in terms of the convolution sum as

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

where we first used the causality of the input ($x[k] = 0$ for $k < 0$) and then that of the system, i.e., $h[n-k] = 0$ whenever $n - k < 0$ or $k > n$. According to this equation the output depends on inputs $\{x[0], \cdots, x[n]\}$ which are past and present values of the input.

Example 9.24. So far we have considered the convolution sum as a way of computing the output $y[n]$ of a LTI system with impulse response $h[n]$ for a given input $x[n]$. But it actually can be used to find either of these three variables given the other two. The problem is then called **deconvolution**. Assume the input $x[n]$ and the output $y[n]$ of a causal LTI system are given, find equations to compute recursively the impulse response $h[n]$ of the system. Consider finding the impulse response $h[n]$ of a causal LTI system with input $x[n] = u[n]$ and output $y[n] = \delta[n]$. Use the MATLAB function *deconv* to find $h[n]$.

Solution: If the system is causal and LTI, the input $x[n]$ and the output $y[n]$ are connected by the convolution sum

$$y[n] = \sum_{m=0}^{n} h[n-m]x[m] = h[n]x[0] + \sum_{m=1}^{n} h[n-m]x[m].$$

To find $h[n]$ from given input and output values, under the condition that $x[0] \neq 0$, the above equation can be rewritten as

$$h[n] = \frac{1}{x[0]}\left[y[n] - \sum_{m=1}^{n} h[n-m]x[m]\right]$$

so that the impulse response of the causal LTI can be found recursively as follows:

$$h[0] = \frac{1}{x[0]}y[0], \quad h[1] = \frac{1}{x[0]}\left(y[1] - h[0]x[1]\right),$$
$$h[2] = \frac{1}{x[0]}\left(y[2] - h[0]x[2] - h[1]x[1]\right) \quad \cdots.$$

For the given case where $y[n] = \delta[n]$ and $x[n] = u[n]$ we get according to the above

$$h[0] = \frac{1}{x[0]}y[0] = 1,$$
$$h[1] = \frac{1}{x[0]}\left(y[1] - h[0]x[1]\right) = 0 - 1 = -1,$$

$$h[2] = \frac{1}{x[0]}(y[2] - h[0]x[2] - h[1]x[1]) = 0 - 1 + 1 = 0,$$
$$h[3] = \frac{1}{x[0]}(y[3] - h[0]x[3] - h[1]x[2] - h[2]x[3]) = 0 - 1 + 1 - 0 = 0, \quad \cdots$$

and in general $h[n] = \delta[n] - \delta[n-1]$.

The length of the convolution $y[n]$ is the sum of the lengths of the input $x[n]$ and of the impulse response $h[n]$ minus one. Thus,

$$\text{length of } h[n] = \text{length of } y[n] - \text{length of } x[n] + 1.$$

When using the MATLAB function *deconv* we need to make sure that the length of $y[n]$ is always larger than that of $x[n]$. If $x[n]$ is of infinite length, like when $x[n] = u[n]$, this would require an even longer $y[n]$, which is not possible. However, MATLAB can only provide a finite support input, so we make the support of $y[n]$ larger. In this example we have found analytically that the impulse response $h[n]$ is of length 2. Thus, if the length of $y[n]$ is chosen larger than the length of $x[n]$ by one we get the correct answer (case (a) in the script below). Otherwise we do not (case (b)). Run the two cases to verify this (get rid of the symbol % to run case (b)).

```
%%
% Example 9.24---Deconvolution
%%
clear all
x=ones(1,100);
y=[1 zeros(1,100)];          % case (a), correct h
% y=[1 zeros(1,99)];         % case (b), wrong h
[h,r]=deconv(y,x)
```

Bounded Input–Bounded Output (BIBO) Stability

Stability characterizes useful systems. A stable system provides well-behaved outputs for well-behaved inputs. Bounded input–bounded output (BIBO) stability establishes that for a bounded (which is what is meant by 'well-behaved') input $x[n]$ the output of a BIBO stable system $y[n]$ is also bounded. This means that if there is a finite bound $M < \infty$ such that $|x[n]| < M$ for all n (you can think of it as an envelope $[-M, M]$ inside which the input $x[n]$ is) the output is also bounded, i.e., $|y[n]| < L$ for $L < \infty$ and all n.

> An LTI discrete-time system is said to be BIBO stable if its impulse response $h[n]$ is absolutely summable
>
> $$\sum_k |h[k]| < \infty \quad \text{(absolutely summable).} \tag{9.40}$$

Assuming that the input $x[n]$ of the system is bounded, or that there is a value $M < \infty$ such that $|x[n]| < M$ for all n, the output $y[n]$ of the system represented by a convolution sum is also bounded or

$$
\begin{aligned}
|y[n]| &\leq \left|\sum_{k=-\infty}^{\infty} x[n-k]h[k]\right| \leq \sum_{k=-\infty}^{\infty} |x[n-k]||h[k]| \\
&\leq M \sum_{k=-\infty}^{\infty} |h[k]| \leq \underbrace{MN}_{L} < \infty
\end{aligned}
$$

provided that $\sum_{k=-\infty}^{\infty} |h[k]| < N < \infty$, or that the impulse response be absolutely summable.

Remarks

1. Non-recursive or FIR systems are BIBO stable. Indeed, the impulse response of such a system is of finite length and therefore absolutely summable.
2. For a recursive or IIR system represented by a difference equation, to established stability according to the above result we need to find the system's impulse response $h[n]$ and determine whether it is absolutely summable or not. A much simpler way to test the stability of an IIR system will be based on the location of the poles of the Z-transform of $h[n]$, as we will see in the next chapter.

Example 9.25. Consider an autoregressive system

$$y[n] = 0.5y[n-1] + x[n].$$

Determine if the system is BIBO stable.

Solution: As shown in Example 9.23, the impulse response of the system is $h[n] = 0.5^n u[n]$, checking the BIBO stability condition we have

$$\sum_{n=-\infty}^{\infty} |h[n]| = \sum_{n=0}^{\infty} 0.5^n = \frac{1}{1-0.5} = 2;$$

thus the system is BIBO stable. □

9.4 TWO-DIMENSIONAL DISCRETE SIGNALS AND SYSTEMS

In this section we consider two-dimensional signals (with special interests in images) and systems. The theory of two-dimensional signals and systems has a great deal of similarities, but also significant differences, with the theory of one-dimensional signals and systems; it implies the characteristics of one-dimensional signals and systems, but not the other way around. For a more detailed presentation refer to [24,49,81].

9.4.1 TWO-DIMENSIONAL DISCRETE SIGNALS

A discrete two-dimensional signal $x[m,n]$ is a mapping of integers $[m,n]$ into real values that is not defined for non-integer values. Two-dimensional signals have a greater variety of supports than one-dimensional signals: they can have finite or infinite support in either of the quadrants of the two-dimensional space, or in a combination of these.

To represent any signal $x[m,n]$ consider a **two-dimensional impulse** $\delta[m,n]$ defined as

$$\delta[m,n] = \begin{cases} 1 & [m,n]=[0,0], \\ 0 & [m,n] \neq [0,0], \end{cases} \tag{9.41}$$

so that a signal $x[m,n]$ defined in a support $[M_1, N_1] \times [M_2, N_2]$, $M_1 < M_2,\ N_1 < N_2$ can be written

$$x[m,n] = \sum_{k=M_1}^{M_2} \sum_{\ell=N_1}^{N_2} x[k,\ell]\delta[m-k, n-\ell]. \tag{9.42}$$

Thus simple signals such as the **two-dimensional unit-step** signal $u_1[m,n]$, with support in the first quadrant,[4] is given by

$$u_1[m,n] = \begin{cases} 1 & m \geq 0,\ n \geq 0 \\ 0 & \text{otherwise} \end{cases} = \sum_{k=0}^{\infty} \sum_{\ell=0}^{\infty} \delta[m-k, n-\ell] \tag{9.43}$$

and a **two-dimensional unit-ramp** signal $r_1[m,n]$, with support in the first quadrant, is given by

$$r_1[m,n] = \begin{cases} mn & m \geq 0,\ n \geq 0 \\ 0 & \text{otherwise} \end{cases} = \sum_{k=0}^{\infty} \sum_{\ell=0}^{\infty} k\ell\delta[m-k, n-\ell]. \tag{9.44}$$

A class of two-dimensional signals of interest are **separable signals** $y[m,n]$ that are the product of two one-dimensional signals, one being a function of m and the other of n:

$$y[m,n] = y_1[m]y_2[n]. \tag{9.45}$$

Given that $\delta[m,n] = \delta[m]\delta[n]$ is separable, one can easily see that $u_1[m,n]$ and $r_1[m,n]$ are also separable. There are, however, simple two-dimensional signals that are not separable. For instance, consider

$$\delta_1[m,n] = \begin{cases} 1 & m=n,\ m \geq 0, n \geq 0, \\ 0 & \text{otherwise}, \end{cases} \tag{9.46}$$

or a sequence of delta functions supported on the diagonal of the first quadrant. This signal cannot be expressed as a product of two one-dimensional sequences. It can, however, be represented by a sum of separable delta functions, or

$$\delta_1[m,n] = \sum_{k=0}^{\infty} \delta[m-k, n-k] = \sum_{k=0}^{\infty} \delta[m-k]\delta[n-k]. \tag{9.47}$$

[4]Whenever it is necessary to indicate the quadrant or quadrants of support the subscript number or numbers will indicate that support. Thus, as $u_1[m,n]$ has support in the first quadrant, $u_3[m,n]$ has support in the third quadrant and $u_{23}[m,n]$ has support in the second and third quadrants.

The best example of two-dimensional discrete signals are discretized images. A discretized image with MN picture elements or pixels, is a positive array $i[m,n]$, $0 \le m \le M-1$, $0 \le n \le N-1$ obtained by sampling an analog image.[5] Such signals are not separable in most cases, but they have a finite support, and they have positive values given that each pixel represents the average illumination of a small area in the analog image.

Example 9.26. In two dimensions (either space and space, or time and space, or any other two variables) the sampling of a continuous two-variable signal $x(\xi,\zeta)$ is much more general than the sampling of one-variable signal. The signal $x(\xi,\zeta)$ can be sampled in general by expressing ξ,ζ for integers m and n and real values $\{M_{i,j},\ 1 \le i,j \le 2\}$ as

$$\begin{bmatrix} \xi \\ \zeta \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} m \\ n \end{bmatrix} \tag{9.48}$$

so that $x(M_{11}m + M_{12}n, M_{21}m + M_{22}n)$ is a discrete signal. Suppose $M_{12} = M_{21} = 0$, how are the values M_{11} and M_{22} chosen? Suppose $M_{11} = M_{12} = 1$ and $M_{21} = -1$ and $M_{22} = 1$; what would be the geometry of the sampling? Is it possible to change the variables ξ and ζ to have the matrix $\mathbf{M}$ with entries $\{M_{i,j}\}$ as the identity matrix?

Solution: When $M_{12} = M_{21} = 0$ we have a rectangular sampling, where the values of the discrete signal are obtained by sampling in a rectangular way with two sampling spacings M_{11} and M_{22}. This is an extension of the sampling in one-dimension, where the samples are at the intersection of $M_{11}m$ and $M_{22}n$, and in that case for band-limited two-dimensional signals $x(\xi,\zeta)$ with maximum frequencies Ω_{1max} and Ω_{2max}, we would let

$$M_{11} \le \frac{\pi}{\Omega_{1max}}, \qquad M_{22} \le \frac{\pi}{\Omega_{2max}},$$

a simple extension of the Nyquist criterion.

When $M_{11} = M_{12} = 1$ and $M_{21} = -1$ and $M_{22} = 1$, we will then have $\xi = m + n$ and $\zeta = -m + n$, and instead of a rectangle in this case we get a hexagonal shape as shown in Fig. 9.14.

The matrix $\mathbf{M}$ formed by the $\{M_{i,j}, 1 \le i,j \le 2\}$ should be invertible. If $\mathbf{M}^{-1}$ is the inverse, then premultiplying Equation (9.48) by it gives the new variables ξ' and ζ' such that

$$\begin{bmatrix} \xi' \\ \zeta' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} m \\ n \end{bmatrix}$$

so that we can have a rectangular sampling when we transform the original variables ξ and ζ using the transformation $\mathbf{M}^{-1}$. □

If the two-dimensional signal is of infinite support it is possible for it to be periodic. A periodic two-dimensional signal $\tilde{x}[m,n]$, $-\infty < m < \infty$, $-\infty < n < \infty$, is such that for positive integers M and N

$$\tilde{x}[m + kM, n + \ell N] = \tilde{x}[m,n], \qquad \text{for any integers } k, \ell. \tag{9.49}$$

[5]When quantized and coded discrete images become digital images.

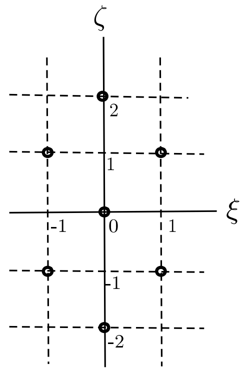

FIGURE 9.14

Sampling pattern when $M_{11} = M_{12} = 1$ and $M_{21} = -1$, $M_{22} = 1$ repeated with a period of 2 in the horizontal direction and 4 in the vertical direction. Notice that there are 7 samples (denoted by dark circles) instead of 11 (given by a rectangular sampling with $M_{11} = M_{22} = 1$ and $M_{12} = M_{21} = 0$) in the area bounded by the hexagon.

Because of the required infinite dimension, periodic signals are not common in practical applications but conceptually they are of great significance. The integer nature of the periods causes that the sampling of periodic analog signals does not always result in periodic discrete signals.

Example 9.27. The periodic analog sinusoid

$$x(t_1, t_2) = \sin(2\pi t_1 + \pi t_2), \qquad -\infty < t_1 < \infty,\ -\infty < t_2 < \infty,$$

has continuous frequencies $f_1 = 1$ Hz and $f_2 = 0.5$ Hz so that $T_1 = 1$ s and $T_2 = 2$ s are the corresponding periods. Suppose $x(t_1, t_2)$ is sampled at times $t_1 = 0.1T_1 m$ and $t_2 = T_2 n/\pi$, for integers m and n. Is the resulting discrete signal still periodic? How would one come up with a two-dimensional periodic signal using approximate sampling times?

Solution: Sampling $x(t_1, t_2)$ by taking samples at $t_1 = 0.1T_1 m$ and $t_2 = T_2 n/\pi$, for integers m, and n, gives the discrete signal

$$x(0.1T_1 m, T_2 n/\pi) = \sin(0.2\pi T_1 m + T_2 n) = \sin\left(\frac{2\pi}{10}m + 2n\right) = x_1[m, n],$$

which is periodic in m with period $M = 10$, but is not periodic in n as there is no positive integer value N such that the discrete frequency $\omega_2 = 2\pi/N$ be equal to 2. The reason for the non-periodicity of the discrete signal is that samples in t_2 are taken at irrational times $T_2 n/\pi$, which in practice cannot be done. To remedy that, suppose the analog sinusoid is sampled at $t_1 = 0.1T_1 m$ as before but at $t_2 = 0.32T_2 n$ (where $0.32 \approx 1/\pi$). We then have

$$x(0.1T_1 m, 0.32T_2 n) = \sin(0.2\pi T_1 m + 0.32\pi T_2 n) = \sin\left(\frac{2\pi}{10}m + \frac{2\pi \times 32}{100}n\right) = x_1[m, n]$$

with discrete frequencies $\omega_1 = 2\pi/10$ and $\omega_2 = 2\pi \times 32/100$ and thus being periodic with periods $M = 10$ and $N = 100$. The expectation that sampling an analog sinusoid, periodic by nature, results in a periodic signal is not always true in two dimensions, just as it was not always true in one dimension. □

A difference between one- and two-dimensional periodicity is that the period of a one-dimensional periodic signal can be chosen so as to give the smallest possible period, i.e., the fundamental period, but that is not always the case in two dimensions. A general way to represent the periodicity in two dimensions is by assigning a periodicity matrix

$$\mathbf{N} = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix}$$

such that

$$\begin{aligned} \tilde{x}[m,n] &= \tilde{x}[m + N_{11}, n + N_{12}] \\ &= \tilde{x}[m + N_{21}, n + N_{22}]. \end{aligned} \tag{9.50}$$

If $N_{11} = M$, $N_{22} = N$, where M and N are the periodicities in Equation (9.49), and $N_{12} = N_{21} = 0$ this is called **rectangular periodicity**, otherwise it is a **block periodicity**. The number of terms in a rectangular period is $N_{11}N_{22} = MN$ which is the absolute value of the determinant of the matrix $\mathbf{N}$ for that case. There are cases when N_{12} and N_{21} are different from zero, and in such cases there is a smaller array that can be taken as the period. Such an array has $|N_{11}N_{22} - N_{12}N_{21}|$ elements, a value that could be smaller than the one corresponding to the rectangular periodicity. The rectangular periodicity does not necessarily have the smallest period but it is a valid periodicity. Because of the complexity in finding the general or block periodicity and a valid, smaller sized period we will only consider the rectangular periodicity.

Example 9.28. To illustrate the connection between the rectangular and the block periodicities, consider the two-dimensional sinusoid

$$\tilde{x}[m,n] = \cos(2\pi\ m/8 + 2\pi\ n/16),$$

which is rectangular periodic in m with a period $M = 8$, and in n with a period $N = 16$. The number of entries of the corresponding period is $M \times N = 8 \times 16 = 128$ which equals the absolute value of the determinant of the periodicity matrix

$$\mathbf{N} = \begin{bmatrix} 8 & 0 \\ 0 & 16 \end{bmatrix}.$$

If the periodicity matrix is chosen to be

$$\mathbf{N}_1 = \begin{bmatrix} 4 & 8 \\ 1 & -2 \end{bmatrix}$$

show the signal is still periodic. What would be an advantage of choosing this matrix over the diagonal matrix?

Solution: Using $\mathbf{N}_1$ we have

$$\tilde{x}[m,n] = \tilde{x}[m+4, n+8] = \tilde{x}[m+1, n-2]$$

as can easily be verified:

$$\tilde{x}[m+4, n+8] = \cos(2\pi m/8 + \pi + 2\pi n/16 + \pi) = \tilde{x}[m,n],$$
$$\tilde{x}[m+1, n-2] = \cos(2\pi m/8 + 2\pi/8 + 2\pi n/16 - 2\pi/8) = \tilde{x}[m,n].$$

This general periodicity provides a period with fewer terms than the previous rectangular periodicity. The absolute value of the determinant of the matrix $\mathbf{N}_1$ is 16. □

9.4.2 TWO-DIMENSIONAL DISCRETE SYSTEMS

A two-dimensional system is an operator $\mathcal{S}$ that maps an input $x[m,n]$ into a unique output

$$y[m,n] = \mathcal{S}(x[m,n]). \tag{9.51}$$

The desirable characteristics of two-dimensional systems are like those of one-dimensional systems. Thus for given inputs $\{x_i[m,n]\}$ having as output $\{y_i[m,n] = \mathcal{S}(x_i[m,n])\}$ and real-valued constants $\{a_i\}$, for $i = 1, 2, \cdots, I$, we see that if

$$\mathcal{S}\left(\sum_{i=1}^{I} a_i x_i[m,n]\right) = \sum_{i=1}^{I} a_i \mathcal{S}(x_i[m,n]) = \sum_{i=1}^{I} a_i y_i[m,n], \tag{9.52}$$

then the system $\mathcal{S}$ is **linear**, and if the shifting of the input does not change the output, i.e., for some integers M and N if

$$\mathcal{S}(x_i[m-M, n-N]) = y_i[m-M, n-N], \tag{9.53}$$

the system $\mathcal{S}$ is **shift-invariant**. A system satisfying these two conditions is called **linear shift-invariant** or **LSI**.

Suppose then the input $x[m,n]$ of an LSI system is represented as in Equation (9.42) and that the response of the system to $\delta[m,n]$ is $h[m,n]$ or the **impulse response** of the system. According to the linearity and shift-invariance characteristics of the system the response to $x[m,n]$ is

$$\begin{aligned} y[m,n] &= \sum_k \sum_\ell x[k,\ell] \mathcal{S}(\delta[m-k, n-\ell]) \\ &= \sum_k \sum_\ell x[k,\ell] h[m-k, n-\ell] = (x * h)[m,n] \end{aligned} \tag{9.54}$$

or the **two-dimensional convolution sum**. The impulse response is the response of the system exclusively to $\delta[m,n]$ and zero-boundary conditions, or the zero-boundary conditions response.

An LSI system is **separable** if its impulse response $h[m,n]$ is a separable sequence. The convolution sum when the system is separable, i.e., its impulse response is $h[m,n] = h_1[m]h_2[n]$, and both the input $x[m,n]$ and the impulse response $h[m,n]$ have finite first quadrant support is

$$\begin{aligned} y[m,n] &= \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} x[k,\ell]h_1[m-k]h_2[n-\ell] \\ &= \sum_{k=-\infty}^{\infty} h_1[m-k]\left[\sum_{\ell=-\infty}^{\infty} x[k,\ell]h_2[n-\ell]\right]. \end{aligned}$$

Noticing that the term in the brackets is the convolution sum of $h_2[n]$ and the input for fixed values of k, if we let

$$y_1[k,n] = \sum_{\ell=-\infty}^{\infty} x[k,\ell]h_2[n-\ell] = \sum_{\ell=0}^{\infty} x[k,\ell]h_2[n-\ell], \tag{9.55}$$

we then have

$$y[m,n] = \sum_{k=-\infty}^{\infty} y_1[k,n]h_1[m-k] = \sum_{k=0}^{\infty} y_1[k,n]h_1[m-k], \tag{9.56}$$

which is a one-dimensional convolution sum of $h_1[m]$ and $y_1[k,n]$ for fixed values of n. Thus, for a system with separable impulse response the two-dimensional convolution results from performing a one-dimensional convolution of columns (or rows) and then rows (or columns).[6]

Example 9.29. To illustrate the above, consider a separable impulse response

$$\begin{aligned} h[m,n] &= \begin{cases} 1 & 0 \le m \le 1, 0 \le n \le 1, \\ 0 & \text{otherwise,} \end{cases} \\ &= (u[m]-u[m-2])(u[n]-u[n-2]) = h_1[m]h_2[n]. \end{aligned}$$

For an input

$$x[m,n] = \begin{cases} 1 & 0 \le m \le 1, 0 \le n \le 1, \\ 0 & \text{otherwise,} \end{cases}$$

find the output of the system $y[m,n]$.

Solution: For $k = 0, 1, 2$ we have

$$y_1[0,n] = \sum_{\ell=0}^{1} x[0,\ell]h_2[n-\ell] = x[0,0]h_2[n] + x[0,1]h_2[n-1] = h_2[n] + h_2[n-1],$$

[6] The final forms of Equations (9.55) and (9.56) are obtained using that both input and impulse response are supported in the first quadrant.

$$y_1[1,n] = \sum_{\ell=0}^{1} x[1,\ell]h_2[n-\ell] = x[1,0]h_2[n] + x[1,1]h_2[n-1] = h_2[n] + h_2[n-1],$$
$$y_1[2,n] = \sum_{\ell=0}^{1} x[2,\ell]h_2[n-\ell] = x[2,0]h_2[n] + x[2,1]h_2[n-1] = 0 + 0,$$

and zero for any value of k larger than 2. For values of $k < 0$ we have $y_1[k,n] = 0$ because the input will be zero. We thus have

$$\begin{array}{lll} y_1[0,0] = 1, & y_1[0,1] = 2, & y_1[0,2] = 1, \\ y_1[1,0] = 1, & y_1[1,1] = 2, & y_1[1,2] = 1, \end{array}$$

and the rest of the values are zero. Notice that the support of $y_1[m,n]$ is 2×3 which is the support of $x[m,n]$ (2×2) plus the support of $h_2[n]$ (1×2) minus 1 for both row and column. The final result is then the one-dimensional convolution

$$\begin{aligned}
y[0,0] &= \sum_{k=0}^{1} y_1[k,0]h_1[-k] = y_1[0,0]h_1[0], \\
y[0,1] &= \sum_{k=0}^{1} y_1[k,1]h_1[-k] = y_1[0,1]h_1[0], \\
y[0,2] &= \sum_{k=0}^{1} y_1[k,2]h_1[-k] = y_1[0,2]h_1[0], \\
y[1,0] &= \sum_{k=0}^{1} y_1[k,0]h_1[1-k] = y_1[0,0]h_1[1] + y_1[1,0]h_1[0], \\
y[1,1] &= \sum_{k=0}^{1} y_1[k,1]h_1[1-k] = y_1[0,1]h_1[1] + y_1[1,1]h_1[0], \\
y[1,2] &= \sum_{k=0}^{1} y_1[k,2]h_1[1-k] = y_1[0,2]h_1[1] + y_1[1,2]h_1[0], \\
y[2,0] &= \sum_{k=0}^{1} y_1[k,0]h_1[2-k] = y_1[1,0]h_1[1], \\
y[2,1] &= \sum_{k=0}^{1} y_1[k,1]h_1[2-k] = y_1[1,1]h_1[1], \\
y[2,2] &= \sum_{k=0}^{1} y_1[k,2]h_1[2-k] = y_1[1,2]h_1[1],
\end{aligned}$$

which gives after replacing the values of $h_1[m]$

$$\begin{array}{lll} y[0,0]=1, & y[0,1]=2, & y[0,2]=1, \\ y[1,0]=2, & y[1,1]=4, & y[1,2]=2, \\ y[2,0]=1, & y[2,1]=2, & y[2,2]=1, \end{array}$$

which is the result of convolving by columns and then by rows. The size of $y[m,n]$ is $(2+2-1) \times (2+2-1)$ or 3×3. To verify this result use the MATLAB function *conv2*. The advantage of using 2D systems that are separable is that only one-dimensional processing is needed. □

A **bounded input–bounded output (BIBO) stable** two-dimensional LSI system is such that if its input $x[m,n]$ is bounded (i.e., there is a positive finite value L such that $|x[m,n]| < L$) the corresponding output $y[m,n]$ is also bounded. Thus, from the convolution sum we have

$$|y[m,n]| \leq \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} |x[k,\ell]|\, |h[m,n]| \leq L \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} |h[m,n]| < \infty \tag{9.57}$$

or that the impulse response $h[m,n]$ be absolutely summable in order for the system to be BIBO stable.

As in the one-dimensional case, two-dimensional systems are **recursive** and **non-recursive**. For instance, a non-recursive two-dimensional system is represented by an input–output equation:

$$y[m,n] = b_{00}x[m,n] + b_{01}x[m,n-1] + b_{10}x[m-1,n] + b_{11}x[m-1,n-1] \tag{9.58}$$

where $x[m,n]$ is the input and $y[m,n]$ is the output, and the $\{b_{i,j}, i=0,1;\ \ j=0,1\}$ are real-valued coefficients. Such a system is LSI, and it is also called a **Finite Impulse Response (FIR)** system given that it has an impulse response $h[m,n]$ of finite support. Indeed, letting $x[m,n] = \delta[m,n]$ the output is $y[m,n] = h[m,n]$ or the impulse response of the system computed according to the above input/output relation:

$$h[m,n] = b_{00}\delta[m,n] + b_{01}\delta[m,n-1] + b_{10}\delta[m-1,n] + b_{11}\delta[m-1,n-1] \tag{9.59}$$

having only four values $h[0,0] = b_{00},\ h[0,1] = b_{01},\ h[1,0] = b_{10},\ h[1,1] = b_{11}$ in its support, the rest are zero. Thus such a system is BIBO if the coefficients are bounded. This result is generalized: any FIR filter is BIBO stable if its coefficients are bounded.

On the other hand, if the two-dimensional system is recursive, or is said to be an **Infinite Impulse Response (IIR)** system, the BIBO stability is more complicated as these systems typically have an impulse response with an infinite size support and as such the absolute summability might be hard to ascertain.

Example 9.30. A recursive system is represented by the difference equation

$$y[m,n] = x[m,n] + y[m-1,n] + y[m,n-1], \qquad m \geq 0, n \geq 0$$

where $x[m,n]$ is the input and $y[m,n]$ is the output. Determine the impulse response $h[m,n]$ of this recursive system and from it determine if the system is BIBO stable.

Solution: The impulse response is the response of the system when $x[m,n]=\delta[m,n]$ and zero-boundary conditions. Since for $m<0$ and/or $n<0$ the input as well as the boundary condition is zero, $h[m,n]=0$ for $m<0$ and/or $n<0$. For other values, $h[m,n]$ is computed recursively (this is the reason for it being called a recursive system) in some order, we thus have

$$\begin{aligned}
&h[0,0]=x[0,0]+h[-1,0]+h[0,-1]=1+0+0=1,\\
&h[0,1]=x[0,1]+h[-1,1]+h[0,0]=0+0+1=1,\\
&h[0,2]=x[0,2]+h[-1,2]+h[0,1]=0+0+1=1,\\
&\cdots\\
&h[1,0]=x[1,0]+h[0,0]+h[1,-1]=0+1+0=1,\\
&h[1,1]=x[1,1]+h[0,1]+h[1,0]=0+1+1=2,\\
&h[1,2]=x[1,2]+h[0,2]+h[1,1]=0+1+2=3,\\
&\cdots\\
&h[2,0]=x[2,0]+h[1,0]+h[2,-1]=0+1+0=1,\\
&h[2,1]=x[2,1]+h[1,1]+h[2,0]=0+2+1=3,\\
&h[2,2]=x[2,2]+h[1,2]+h[2,1]=0+3+3=6,\\
&\cdots.
\end{aligned}$$

Unless we were able to compute the rest of the values of the impulse response, or to obtain a closed form for, it the BIBO stability would be hard to determine from the impulse response. As it will be shown later, a closed form for this impulse response is

$$h[m,n]=\frac{(m+n)!}{n!\,m!}u_1[m,n]$$

where $n!=1\times 2\cdots(n-1)\times n$ is the n factorial.[7]

Using the MATLAB function *nchoosek* to compute the values of the impulse response (see Fig. 9.15) according to this formula, it is seen that as m and n increase the value of $h[m,n]$ also increases, so the system is not BIBO stable. □

Example 9.31. Consider a recursive system is represented by the difference equation

$$y[m,n]=by[m,n-1]+ay[m-1,n]-aby[m-1,n-1]+x[m,n],\quad |a|<1,\ |b|<1. \tag{9.60}$$

Is this system BIBO stable according to its impulse response?

[7]The impulse response can also be expressed as

$$h[m,n]=\binom{m+n}{n}u_1[m,n]$$

using the binomial coefficient notation.

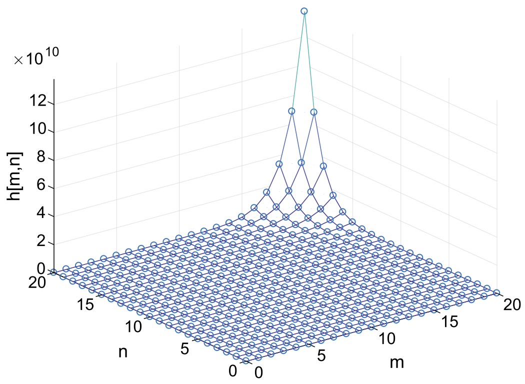

FIGURE 9.15

Impulse response of recursive system $y[m,n]=x[m,n]+y[m-1,n]+y[m,n-1]$, $m \geq 0$, $n \geq 0$.

Solution: The impulse response is obtained by letting $x[m,n]=\delta[m,n]$, and zero-boundary conditions. Then $h[m,n]=y[m,n]$ satisfies the difference equation

$$h[m,n]=bh[m,n-1]+ah[m-1,n]-abh[m-1,n-1]+\delta[m,n].$$

Letting $h_1[m,n]=h[m,n]-ah[m-1,n]$, when replaced in the above difference equation gives

$$\begin{aligned} h_1[m,n] &= b(h[m,n-1]-ah[m-1,n-1])+\delta[m,n] \\ &= bh_1[m,n-1]+\delta[m,n], \end{aligned}$$

which for a fixed m is a first-order difference equation in n with solution $h_1[m,n]=b^n u[n]$. From the definition $h_1[m,n]=h[m,n]-ah[m-1,n]$ replacing $h_1[m,n]=b^n u[n]$ we get a first-order difference equation in m with input a function in n, i.e.,

$$b^n u[n]=h[m,n]-ah[m-1,n]$$

with solution $h[m,n]=(b^n u[n])(a^m u[m])=h_1[n]h_2[m]$, i.e., separable. This system is BIBO stable as $h[m,n]$ is absolutely summable given that $h_1[n]$ as well as $h_2[m]$ are absolutely summable. □

Causality is imposed on one-dimensional systems to allow computations in real-time, i.e., the output of the system at a particular instance is obtainable from present and past values of the input. Lack of causality requires adding delays to the processing units which is not desirable. Causality ensures that the necessary data is available when computing the output. Causality, as such, is the same characteristic for two-dimensional systems, but in practice it is not as necessary as in one-dimension. For instance, processing images can be executed in a causal manner by processing the pixels as they come from the raster but they could also be processed by considering the whole image. In most cases, images are available in frames and their processing does not require causality.

9.5 WHAT HAVE WE ACCOMPLISHED? WHERE DO WE GO FROM HERE?

As you saw in this chapter, the theory of discrete-time signals and systems is very similar to the theory of continuous-time signals and systems. Many of the results in the continuous-time theory are changed by replacing integrals by sums, derivatives by differences and ordinary differential equations by difference equations. However, there are significant differences imposed by the way the discrete-time signals and systems are generated. For instance, the discrete frequency is finite but circular, and it depends on the sampling time. Discrete sinusoids, as another example, are not necessarily periodic. Thus, despite the similarities there are also significant differences between the continuous-time and the discrete-time signals and systems.

Now that we have a basic structure for discrete-time signals and systems, we will start developing the theory of linear time-invariant discrete-time systems. Again you will find a great deal of similarity with the linear time-invariant continuous-time systems but also some very significant differences. Also, notice the relation that exists between the Z-transform and the Fourier representations of discrete-time signals and systems, not only with each other but with the Laplace and Fourier transforms. There is a great deal of connection among all of these transforms, and a clear understanding of this would help you with the analysis and synthesis of discrete-time signals and systems.

A large number of books have been written covering various aspects of digital signal processing [39,55,62,19,30,47,68,61,13,51]. Some of these are classical books in the area that the reader should look into for a different perspective of the material presented in this part of the book.

Extending the theory of one-dimensional signals and systems to two dimensions is very doable, but it is important to recognize that the theory of two-dimensional signals is more general than the one-dimensional. As such, it is clear that the two-dimensional theory is richer conceptually, and that many of the one-dimensional properties are no valid in two dimensions, given that the theory of one-dimensional signals and systems is a subset of the two-dimensional theory. In the next chapters we will see how to obtain a two-dimensional Z-transform and a discrete Fourier transform, and how they can be used to process two-dimensional signals and in particular images.

9.6 PROBLEMS

9.6.1 BASIC PROBLEMS

9.1 For the discrete-time signal

$$x[n] = \begin{cases} 1 & n = -1,\ 0,\ 1, \\ 0.5 & n = 2, \\ 0 & \text{otherwise}, \end{cases}$$

sketch and label carefully the following signals:

(a) $x[n-1]$, $x[-n]$, and $x[2-n]$.
(b) The even component $x_e[n]$ of $x[n]$.
(c) The odd component $x_o[n]$ of $x[n]$.

Answers: $x[2-n] = 0.5\delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3]$; $x_o[n] = 0.25\delta[n-2] - 0.25\delta[n+2]$.

9.2 For the discrete-time periodic signal $x[n] = \cos(0.7\pi n)$, $-\infty < n < \infty$,

(a) Determine its fundamental period N_0.

(b) Suppose we sample the continuous-time signal $x(t) = \cos(\pi t)$ with a sampling period $T_s = 0.7$. Is the Nyquist sampling condition satisfied? How does the sampled signal compare to the given $x[n]$?

(c) Under what conditions would sampling a continuous-time signal $x(t) = \cos(\pi t)$ give a discrete-time sinusoid $x[n]$ that resembles $x(t)$? Explain and give an example.

Answers: $N_0 = 20$; yes, $T_s = 0.7$ satisfies the Nyquist condition; let $T_s = 2/N$ with $N >> 2$.

9.3 Consider the following problems related to the periodicity of discrete-time signals.

(a) Determine whether the following discrete-time signals defined in $-\infty < n < \infty$ are periodic or not. If periodic, determine its fundamental period N_0.

$$(i)\;\; x[n] = 2\cos(\pi n - \pi/2), \qquad (ii)\;\; y[n] = \sin(n - \pi/2),$$
$$(iii)\;\; z[n] = x[n] + y[n], \qquad (iv)\;\; v[n] = \sin(3\pi n/2).$$

(b) Consider two periodic signals $x_1[n]$, of fundamental periods $N_1 = 4$, and $y_1[n]$, of period $N_2 = 6$. Determine what would be the fundamental period of the following:

$$(i)\;\; z_1[n] = x_1[n] + y_1[n], \quad (ii)\;\; v_1[n] = x_1[n]y_1[n], \quad (iii)\;\; w_1[n] = x_1[2n].$$

Answers: (a) $y[n]$ is not periodic; (b) $v_1[n]$ is periodic of fundamental period $N_0 = 12$.

9.4 The following problems relate to periodicity and power of discrete-time signals.

(a) Is the signal $x[n] = e^{j(n-8)/8}$ periodic? If so determine its fundamental period N_0. What if $x_1[n] = e^{j((n-8)\pi/8)}$ (notice the difference with $x[n]$) would this new signal be periodic? If so what would the fundamental period N_1 be?

(b) Given the discrete-time signal $x[n] = \cos(\pi n/5) + \sin(\pi n/10)$, $\quad -\infty < n < \infty$.

i. Is $x[n]$ periodic? If so determine its fundamental frequency ω_0.

ii. Is the power of $x[n]$ the sum of the powers of $x_1[n] = \cos(\pi n/5)$ and $x_2[n] = \sin(\pi n/10)$ defined for $-\infty < n < \infty$? If so, show it.

Answers: (a) $x_1[n]$ periodic, $N_1 = 16$; (b) $\omega_0 = 0.1\pi$; (c) yes.

9.5 The following problems relate to linearity, time invariance, causality, and stability of discrete-time systems.

(a) The output $y[n]$ of a system is related to its input $x[n]$ by $y[n] = x[n]x[n-1]$. Is this system

i. linear? time invariant?

ii. causal? bounded input–bounded output stable?

You may consider $x[n] = u[n]$ as the input to verify your results.

(b) Given the discrete-time system in Fig. 9.16.

i. Is this system time invariant?

ii. Suppose that the input is $x[n] = \cos(\pi n/4)$, $-\infty < n < \infty$, so that the output is $y[n] = \cos(\pi n/4)\cos(n/4)$, $-\infty < n < \infty$. Determine the fundamental period N_0 of $x[n]$. Is $y[n]$ periodic? If so, determine its fundamental period N_1.

Answers: (a) non-linear, time-invariant, BIBO stable system; (b) no; $N_0 = 8$, $y[n]$ not periodic.

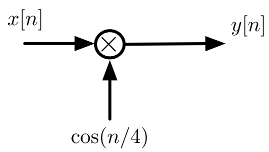

FIGURE 9.16

Problem 9.5.

9.6 Consider a discrete-time system with output $y[n]$ given by $y[n] = x[n]f[n]$ and $x[n]$ is the input and $f[n]$ is a function.

(a) Let the input be $x[n] = 4\cos(\pi n/2)$ and $f[n] = \cos(6\pi n/7)$, $-\infty < n < \infty$. Is $x[n]$ periodic? If so, indicate its fundamental period N_0. Is the output of the system $y[n]$ periodic? If so, indicate its fundamental period N_1.

(b) Suppose now that $f[n] = u[n] - u[n-2]$ and $x[n] = u[n]$. Determine if the system with the above input–output equation is time invariant.

Answers: The system is time varying.

9.7 Consider a system represented by

$$y[n] = \sum_{k=n-2}^{n+4} x[k]$$

where the input is $x[n]$ and the output $y[n]$. Is the system

(a) linear? time invariant?

(b) causal? bounded input–bounded output stable?

Answers: The system is linear, time invariant, and noncausal.

9.8 Determine the impulse response $h[n]$ of a LTI system represented by the difference equation

$$y[n] = -0.5y[n-1] + x[n],$$

where $x[n]$ is the input, $y[n]$ is the output and the initial conditions are zero. Find two different ways to compute the output $y[n]$ when the input is

$$x[n] = \begin{cases} 1 & 0 \le n \le 2, \\ 0 & \text{otherwise.} \end{cases}$$

Answers: $h[n] = (-0.5)^n u[n]$; let $x[n] = \delta[n] + \delta[n-1] + \delta[n-2]$ and use $h[n]$ to get $y[n]$.

9.9 The input of an LTI continuous-time system is $x(t) = u(t) - u(t-3.5)$. The system's impulse response is $h(t) = u(t) - u(t-2.5)$.

(a) Find the system's output $y(t)$ by graphically computing the convolution integral of $x(t)$ and $h(t)$. Sketch $x(t)$, $h(t)$ and the found $y(t)$.

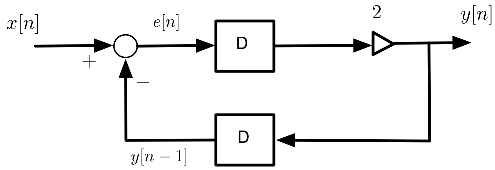

FIGURE 9.17

Problem 9.11.

(b) Suppose we sample $x(t)$, $h(t)$ and $y(t)$ using a sampling period $T_s = 0.5$ s/sample. Sketch these sampled signals as functions of n.

(c) Graphically calculate the convolution sum of $x[n]$ and $h[n]$, and compare it to $y(t)$. Explain your results.

Answers: $y(t) = r(t) - r(t-2.5) - r(t-3.5) + r(t-6)$; $y[n]/6$ approximates $y(t)$.

9.10 You are testing a 1 V dc source and have the following measurements obtained from the source every minute starting at time 0:

n	$x[n]$	n	$x[n]$
0	1.0	3	0.7
1	1.2	4	1.2
2	0.9	5	1.0

To find the average voltages for the first 5 minutes, i.e. to get rid of some of the noise in the data, you use the following averager:

$$y[n] = \frac{x[n] + x[n-1] + x[n-2]}{3}.$$

(a) Use the input/output equation of the filter to compute and then plot the moving averages $y[n]$ for $2 \le n \le 5$.

(b) Find the impulse response $h[n]$ of the averager and compute the output $y[n]$ using the convolution sum. Do you obtain the same results as before?

(c) Suppose you use a median filter of length 3, so that at some sample n we consider the samples values at n, $n-1$ and $n-2$, order these values and the one in the middle is the median output $y_m[n]$. Moving one sample we consider the next 3 values and find the new median. Compute and plot $y_m[n]$ for $2 \le n \le 5$.

Answers: Identical results using difference equation and convolution.

9.11 A causal, LTI discrete-time system is represented by the block diagram shown in Fig. 9.17 where D stands for a one-sample delay.

(a) Find the difference equation relating the input $x[n]$ and the output $y[n]$.

(b) Find the impulse response $h[n]$ of the system and use it to determine if the system is BIBO stable? Explain.

Answer: $h[n] = 2(-2)^{(n-1)/2}$, n odd, 0 otherwise; system is not BIBO stable.

9.12 The input and the output of an LTI discrete-time system are

$$\text{Input: } x[n] = u[n] - u[n-3], \quad \text{Output: } y[n] = u[n-1] - u[n-4].$$

(a) What should be the length of the impulse response $h[n]$ of the system?
(b) Find the impulse response of the system, $h[n]$, and verify the previous answer.
Answers: Impulse response: $h[0] = 0$, $h[1] = 1$ and the rest of the values are zero.

9.13 The following problems relate to the response of LTI discrete-time systems.
(a) The unit-step response of a LTI discrete-time system is found to be

$$s[n] = (3 - 3(0.5)^{n+1})u[n].$$

Use $s[n]$ to find the impulse response $h[n]$ of the system.
(b) The output $y[n]$ of a discrete-time system is the even component of the input $x[n]$, i.e.,

$$y[n] = 0.5(x[n] + x[-n]).$$

i. Consider an input $x[n] = u[n] - u[n-3]$, find the corresponding output $y[n]$. You might want to carefully sketch the input and the output. Is the system causal? Explain.
ii. Use the same input as before, $x[n] = u[n] - u[n-3]$, with the obtained output $y[n]$. If we consider as input $x_1[n] = x[n-1]$, find the corresponding output $y_1[n]$, sketch it and from these results determine if the system is time invariant.
iii. Suppose then that the input is $x[n] = \cos(2\pi n/5)u[n]$, find the corresponding output $y[n]$ and carefully sketch and label the input and output. Is the output periodic?

Answers: (a) $h[n] = (3 - 3(0.5)^{n+1})u[n] - (3 - 3(0.5)^n)u[n-1]$; (b) noncausal.

9.14 An LTI discrete-time system has an impulse response $h[n] = u[n] - u[n-4]$, and as input the signal $x[n] = u[n] - u[n-(N+1)]$ for a positive integer N. The output of the system $y[n]$ is calculated using the convolution sum.
(a) If $N = 4$ what is the length of the output $y[n]$? Explain. For $N = 4$, carefully sketch and label the output $y[n]$ resulting from the convolution sum.
(b) Determine the value of $N \leq 5$ for $x[n]$ so that $y[3] = 3$ and $y[6] = 0$.
Answers: $y[0] = y[7] = 1$, $y[1] = y[6] = 2$, $y[2] = y[5] = 3$, $y[3] = y[4] = 4$, 0 otherwise.

9.15 Consider a discrete-time system represented by the difference equation $y[n] = 0.5y[n-1] + x[n]$ where $x[n]$ is the input and $y[n]$ the output.
(a) An equivalent representation of the system is given by the difference equation

$$y[n] = 0.25y[n-2] + 0.5x[n-1] + x[n].$$

Is it true? Let $x[n] = \delta[n]$ and zero initial conditions and solve the two difference equations to verify this. Determine how to obtain the second difference equation from the first?
(b) Using the first initial difference equation show that the output is

$$y[n] = \sum_{k=0}^{\infty} (0.5)^k x[n-k].$$

What is this expression? determine the impulse response $h[n]$ of the system from this equation? Explain.

(c) If the output is computed using the convolution sum, and the input is

$$x[n] = u[n] - u[n-11],$$

find $y[n]$. Determine the steady-state value of the output, i.e., $y[n]$ as $n \to \infty$.

(d) What is the maximum value achieved by the output $y[n]$? when is it attained?

Answers: Both equations give $y[n] = 0.5^n u[n]$; (d) maximum $y[10] = 2(1 - 0.5^{11})$.

9.16 The following difference equation is used to obtain recursively the ratio α/β:

$$c[n+1] = (1-\beta)c[n] + \alpha \qquad n \geq 0$$

with $c[0]$ as an initial condition. Solve the difference equation, and find under what condition(s) the solution $c[n]$ will converge to the desired answer of α/β as n tends to infinity.

Answers: $0 < \beta < 2$, independent of $c[0]$.

9.17 An LTI causal discrete-time system has the input/output relationship

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

where $x[n]$ is the input of the system, $y[n]$ is the response of the system. There is zero initial energy in the system prior to applying $x[n]$.

(a) Find the impulse response $h[n]$.

(b) Find the unit-step response of the given system.

Answers: $h[n] = (n+2)u[n]$.

9.18 A discrete-time averager is characterized by the following equation relating the input $x(nT_s)$ with the output $y(nT_s)$:

$$y(nT_s) = \frac{1}{2N+1} \sum_{k=-N}^{N} x(nT_s - kT_s).$$

(a) Is this system causal? Explain.

(b) Let $N = 2$ in the above equation. Find and plot the impulse response $h(nT_s)$ of the averager.

(c) For $N = 2$, if the input to the averager is

$$x(nT_s) = \begin{cases} 5 & n = 0, 1, 2 \\ 0 & \text{otherwise} \end{cases}$$

calculate the output $y(nT_s)$.

Answers: (a) This averager is noncausal; (c) $y(nT_s)$ is noncausal of length 7.

9.19 Consider a causal LTI system with impulse response $h[n]$, and input

$$x[n] = x_1[n] - x_1[n-2] + x_1[n-4]$$

where $x_1[n] = u[n] - u[n-2]$. The impulse response of the system is $h[n] = u[n] - u[n-2]$.

(a) Use the convolution sum to find the output of the system, $y[n]$.

(b) If the given system is cascaded with another causal LTI system with impulse response $g[n]$, we know the overall impulse response of the two cascaded systems is

$$h_T[n] = \begin{cases} 1 & n = 0, 3, \\ 0 & \text{otherwise.} \end{cases}$$

Find the impulse response $g[n]$.

Answers: If $y_1[n] = (x_1 * h)[n]$ then $y[n] = y_1[n] - y_1[n-2] + y_1[n-4]$.

9.20 A discrete-time averager is represented by the input/output equation

$$y[n] = (1/3)(x[n+1] + x[n] + x[n-1]),$$

where $x[n]$ is the input and $y[n]$ the output.

(a) Determine whether this system is causal or not. Explain.

(b) Determine whether this system is BIBO stable or not. Explain.

(c) The input of the system is generated by sampling an analog signal $x(t) = 2\cos(10t)$ using sampling periods $T_{s1} = 1$ or $T_{s2} = \pi$ s/sample. If we want the discrete-time signal $x[n] = x(t)|_{t=nT_s}$ to be periodic, which of the two sampling periods $\{T_{si}, i = 1, 2\}$ would you use? For the chosen sampling period what would be the period of $x[n]$?

(d) If $x[n]$ is periodic, would the output of the averager be also periodic? If so, what would be the period of the output? Explain.

Answers: The system is noncausal, but BIBO stable.

9.21 A finite impulse response (FIR) filter has an input/output relation $y[n] = x[n] - x[n-5]$ where $x[n]$ is the input and $y[n]$ the output.

(a) Find the impulse response $h[n]$ of this filter. Plot $h[n]$ as a function of n, and indicate if the filter is causal and BIBO stable or not.

(b) Suppose the input is $x[n] = u[n]$, find the corresponding output $y[n]$ and carefully plot it. Are $x[n]$ and $y[n]$ finite-energy signals?

(c) If $x[n] = \sin(2\pi n/5)u[n]$, find its corresponding output $y[n]$. Determine the energies of the input $x[n]$ and of $y[n]$. Are both finite energy?

(d) Determine the frequencies $\{\omega_0\}$ of the input $x[n] = \sin(\omega_0 n)u[n]$ for which the corresponding output $y[n]$ is finite energy. If you choose a frequency different from these frequencies, is the output finite energy?

Answers: $h[n] = \delta[n] - \delta[n-5]$; for frequencies $\{\omega_0 = 2\pi m/5\}$ for $m = 0, \pm 1, \pm 2, \cdots$, the energy of the output is finite.

9.22 Consider the following problems related to properties of filters.

(a) Filters that operate under real-time conditions need to be causal. When no real-time processing is needed the filter can be noncausal.

i. Consider the case of averaging an input signal $x[n]$ under real-time conditions. Suppose you are given two different filters

$$(A)\ \ y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k], \quad (B)\ \ y[n] = \frac{1}{N}\sum_{k=-N+1}^{N-1} x[n-k],$$

which of these would you use and why.

ii. If you are given a tape with the data which of the two filters would you use? Why? Would you use either? Explain.

(b) A significant difference between IIR and FIR discrete-time systems is stability. Consider an IIR filter represented by the difference equation

$$y_1[n] = x[n] - 0.5y_1[n-1]$$

where $x[n]$ is the input and $y_1[n]$ is the output. Then consider an FIR filter

$$y_2[n] = x[n] + 0.5x[n-1] + 3x[n-2] + x[n-5]$$

where $x[n]$ is the input and $y_2[n]$ is the output.

i. Since to check the stability of these filters we need their impulse responses, find the impulse responses $h_1[n]$ corresponding to the IIR filter by recursion, and $h_2[n]$ corresponding to the FIR filter.

ii. Use the impulse responses $h_1[n]$ and $h_2[n]$ to check the stability of the IIR filter and of the FIR filter, respectively.

iii. Since the impulse response of an FIR filter has a finite number of nonzero terms, would it be correct to say that FIR filters are always stable? Explain.

Answers: (a) Use (A), it is causal; yes, use either; (b) $h_2[n] = \delta[n] + 0.5\delta[n-1] + 3\delta[n-2] + \delta[n-5]$.

9.23 For the two-dimensional signals where $|\alpha| < 1$

$$x[m,n] = \alpha^{m+n} u_{12}[m,n]$$
$$y[m,n] = \alpha^{m+n} u_{14}[m,n].$$

(a) Draw their supports, and express these domains in terms of $u_1[m,n]$.

(b) Let $z[m,n] = x[m,n] - y[m,n]$, and draw its support.

Answer: (b) The support of $z[m,n]$ is in the second and fourth quadrant.

9.24 Consider the line impulses

$$x[m,n] = \sum_{k=-\infty}^{\infty} \delta[0, n-k], \quad y[m,n] = \delta[0] \sum_{\ell=-\infty}^{\infty} \delta[m-\ell]$$

where $-\infty < m < \infty$ and $-\infty < n < \infty$.

(a) Draw the line impulses $x[m,n]$ and $y[m,n]$. Determine if they are separable.

(b) Consider the product $z[m,n] = x[m,n]y[m,n]$. Express $z[m,n]$ in terms of unit-step functions $u_{14}[m,n]$, with support in the first and fourth quadrants, and $u_{23}[m,n]$, with support in the second and third quadrants.

Answer: (b) $z[m,n] = u_{14}[m,n] + u_{23}[m+1,n]$.

9.25 The impulse response

$$h[m,n] = \binom{m+n}{m}$$

satisfies the difference equation

$$h[m,n] = h[m-1,n] + h[m,n-1] + \delta[m,n]$$

with zero-boundary conditions. By definition $\binom{0}{0} = 1$.

(a) Use the difference equation to show that $\binom{-1}{-1} = \binom{-1}{0} = 0$.

(b) Show that for $m \geq 0, n > 0$

$$h[m,n] = \frac{m+n}{n}\, h[m,n-1] + \delta[m,n]$$

determine the values of $h[m,0]$ for $m \geq 0$, and use these equations to compute the impulse response $h[m,n]$. Plot the resulting $h[m,n]$. Is the system with this impulse response BIBO stable? Explain.

9.6.2 PROBLEMS USING MATLAB

9.26 **Finite-energy signals**—Given the discrete signal $x[n] = 0.5^n u[n]$.

(a) Use the function *stem* to plot $x[n]$ for $n = -5$ to 20.

(b) Is this a finite-energy discrete-time signal? i.e., compute the infinite sum

$$\varepsilon_x = \sum_{n=-\infty}^{\infty} |x[n]|^2.$$

(c) Verify your results by using symbolic MATLAB to find an expression for the above sum.

Answers: $\varepsilon_x = 4/3$.

9.27 **Periodicity of sampled signals**—Consider an analog periodic sinusoid $x(t) = \cos(3\pi t + \pi/4)$ being sampled using a sampling period T_s to obtain the discrete-time signal $x[n] = x(t)|_{t=nT_s} = \cos(3\pi T_s n + \pi/4)$.

(a) Determine the discrete frequency of $x[n]$.

(b) For what values of T_s is the discrete-time signal $x[n]$ periodic? Let $T_s = 1/3$, use function *stem* to plot $x[n]$, $0 \leq n \leq 100$ and determine if it is periodic. Let $T_s = 1$, use function *stem* to plot $x[n]$, $0 \leq n \leq 100$ and determine if it is periodic.

Answers: $\omega_0 = 3\pi T_s$ (rad), for $T_s = 1$ the signal $x[n]$ is not periodic.

9.28 Even and odd decomposition and energy—Suppose you sample the analog signal

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

with a sampling period $T_s = 0.25$ to generate $x[n] = x(t)|_{t=nT_s}$.

(a) Use function *stem* to plot $x[n]$ and $x[-n]$ for an appropriate interval.

(b) Find the even, $x_e[n]$, and the odd, $x_o[n]$, components of $x[n]$. Plot them using *stem*. Verify that $x_e[n] + x_o[n] = x[n]$ graphically.

(c) Compute the energy ε_x of $x[n]$ and compare it to the sum of the energies ε_{x_e} and ε_{x_o} of $x_e[n]$ and $x_o[n]$.

Answer: $\varepsilon_x = \varepsilon_{x_e} + \varepsilon_{x_o}$.

9.29 Expansion and compression of discrete-time signals—Consider the discrete-time signal $x[n] = \cos(2\pi n/7)$.

(a) The discrete-time signal can be compressed by getting rid of some of its samples (**down-sampling**). Consider the down-sampling by 2. Write a script to obtain and plot $z[n] = x[2n]$. Plot also $x[n]$ and compare it with $z[n]$, what happened? Explain.

(b) The expansion for discrete-time signals requires interpolation. However, a first step of this process is the so called *up-sampling*. Up-sampling by 2, consists in defining a new signal $y[n]$ such that $y[n] = x[n/2]$ for n even, and $y[n] = 0$ otherwise. Write a script to perform up-sampling on $x[n]$. Plot the resulting signal $y[n]$ and explain its relation with $x[n]$.

(c) If $x[n]$ resulted from sampling a continuous-time signal $x(t) = \cos(2\pi t)$ using a sampling period T_s and with no frequency aliasing, determine T_s. How would you sample the analog signal $x(t)$ to get the down-sampled signals $z[n]$? That is, choose values for the sampling period T_s to get $z[n]$ directly from $x(t)$. Can you choose T_s to get $y[n]$ from $x(t)$ directly? Explain.

Answers: $z[n] = \cos(4\pi n/7)$; $y[n] = \cos(\pi n/7)$.

9.30 Absolutely summable and finite-energy discrete-time signals—Suppose we sample the analog signal $x(t) = e^{-2t}u(t)$, using a sample period $T_s = 1$.

(a) Expressing the sampled signal as $x(nT_s) = x[n] = \alpha^n u[n]$, what is the corresponding value of α? Use *stem* to plot $x[n]$.

(b) Show that $x[n]$ is absolutely summable, i.e., show the following sum is finite:

$$\sum_{n=-\infty}^{\infty} |x[n]|.$$

(c) If you know that $x[n]$ is absolutely summable, could you say that $x[n]$ is a finite-energy signal? Use the function *stem* to plot $|x[n]|$ and $x^2[n]$ in the same plot to help you decide.

(d) In general, for what values of α are signals $y[n] = \alpha^n u[n]$ finite energy? Explain.

Answers: $\alpha = e^{-2} < 1$; $\sum_{n=0}^{\infty} (\alpha^2)^n = e^4/(e^4 - 1) < \infty$.

9.31 Periodicity of sum and product of periodic signals—If $x[n]$ is periodic of period $N_1 > 0$ and $y[n]$ is periodic of period $N_2 > 0$:

(a) What should be the condition for the sum $z[n]$ of $x[n]$ and $y[n]$ to be periodic?

(b) What would be the period of the product $v[n] = x[n]y[n]$?

(c) Would the formula

$$\frac{N_1 N_2}{\text{gcd}[N1, N2]}$$

(gcd$[N_1, N_2]$ stands for the greatest common divisor of N_1 and N_2) give the period of the sum and the product of the two signals $x[n]$ and $y[n]$?

(d) If $x[n] = \cos(2\pi n/3)$, $y[n] = 1 + \sin(6\pi n/7)$, generate and plot using MATLAB their sum $z[n]$ and product $v[n]$, find their periods and verify your analytic results.

Answers: (d) For $N_1 = 3$ and $N_2 = 7$, $z[n]$ and $v[n]$ are periodic of fundamental period $N_0 = 21$.

9.32 **Echoing of music—**An effect similar to multi-path in acoustics is echoing or reverberation. To see the effects of an echo in an acoustic signal consider the simulation of echoes on the *handel.mat* signal $y[n]$. Pretend that this piece is being played in a round theater where the orchestra is in the middle of two concentric circles and the walls on one half side are at a radial distances of 17 m (corresponding to the inner circle), and 34 m (corresponding to the outer circle) on the other side (yes, a usual theater!) from the orchestra. The speed of sound is 345 m/s. Assume that the recorded signal is the sum of the original signal $y[n]$, and attenuated echoes from the two walls so that the recorded signal is given by

$$r[n] = y[n] + 0.8y[n - N_1] + 0.6y[n - N_2]$$

where N_1 is the delay caused by the closest wall and N_2 the delay caused by the farther wall. The recorder is at the center of the auditorium where the orchestra is and we record for 10 s.

(a) Find the values of the two delays N_1 and N_2 (remember these are integers). The sampling frequency F_s of *handel* is given when you load it.

(b) Simulate the echo signal. Plot $r[n]$. Use *sound* to listen to the original and the echoed signals.

Answers: $T_1 = 0.10$, $T_2 = 0.20$ s.

9.33 **A/D converter—**An A/D converter can be thought of composed of three subsystems: a sampler, a quantizer, and a coder.

(a) The sampler, as a system, has as input an analog signal $x(t)$ and as output a discrete-time signal $x(nT_s) = x(t)|_{t=nT_s}$, where T_s is the sampling period. Determine whether the sampler is a linear system or not.

(b) Sample $x(t) = \cos(0.5\pi t)u(t)$ and $x(t - 0.5)$ using $T_s = 1$ to get $y(nT_s)$ and $z(nT_s)$, respectively. Plot $x(t)$, $x(t - 0.5)$ and $y(nT_s)$ and $z(nT_s)$. Is $z(nT_s)$ a shifted version of $y(nT_s)$ so that you can say the sampler is time invariant? Explain.

Answers: Sampler is linear but time varying.

9.34 **Rectangular windowing—**A window is a signal w$[n]$ that is used to highlight part of another signal. The windowing process consists in multiplying an input signal $x[n]$ by the window signal w$[n]$, so that the output is $y[n] = x[n]$w$[n]$. There are different types of windows used in signal processing, one of them is the so called *rectangular window* which is given by

$$\text{w}[n] = u[n] - u[n - N].$$

(a) Determine whether the rectangular windowing system is linear. Explain.

(b) Suppose $x[n] = nu[n]$, use MATLAB to plot the output $y[n]$ of the windowing system (with $N = 6$).

(c) Let the input be $x[n-5]$, use MATLAB to plot the corresponding output of the rectangular windowing system, and indicate whether the rectangular windowing system is time invariant.

Answers: Windowing is linear, but time varying.

9.35 Impulse response of an IIR system—A discrete-time IIR system is represented by the following difference equation:

$$y[n] = 0.15y[n-2] + x[n], \qquad n \geq 0,$$

where $x[n]$ is the input and $y[n]$ is the output.

(a) To find the impulse response $h[n]$ of the system, let $x[n] = \delta[n]$, $y[n] = h[n]$, and the initial conditions be zero. Find recursively the values of $h[n]$ for values of $n \geq 0$.

(b) As a second way to find $h[n]$, replace the relation between the input and the output given by the difference equation to obtain a convolution sum representation which will give the impulse response $h[n]$. What is $h[n]$?

(c) Use the MATLAB function *filter* to get the impulse response $h[n]$ (use *help* to learn about the function *filter*).

Answers: $h[n] = 0.5(1 + (-1)^n)0.15^{n/2}u[n]$.

9.36 FIR filters—An FIR filter has a non-recursive input/output relation

$$y[n] = \sum_{k=0}^{5} kx[n-k].$$

(a) Find the impulse response $h[n]$ of this filter. Is this a causal and stable filter? Explain.

(b) Find the unit-step response $s[n]$ for this filter and plot it.

(c) If the input $x[n]$ for this filter is bounded, $|x[n]| < 3$, what would be a minimum bound M for the output, i.e., $|y[n]| \leq M$.

(d) Use the function *filter* to compute the impulse response $h[n]$ and the unit-step response $s[n]$ for the given filter and plot them.

Answers: $s[n] = u[n-1] + 2u[n-2] + 3u[n-3] + 4u[n-4] + 5u[n-5]$.

9.37 Steady state of IIR systems—Suppose an IIR system is represented by a difference equation

$$y[n] = ay[n-1] + x[n],$$

where $x[n]$ is the input and $y[n]$ is the output.

(a) If the input is $x[n] = u[n]$ and it is known that the steady-state response is $y[n] = 2$, what would be a for that to be possible (hint: in steady state $x[n] = 1$ and $y[n] = y[n-1] = 2$ since $n \to \infty$).

(b) Writing the system input as $x[n] = u[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \cdots$ then according to the linearity and time invariance the output should be

$$y[n] = h[n] + h[n-1] + h[n-2] + \cdots.$$

Use the value for a found above, that the initial condition is zero, i.e., $y[-1]=0$, and that the input is $x[n]=u[n]$, to find the values of the impulse response $h[n]$ for $n \geq 0$ using the above equation. The system is causal.

(c) Use the function *filter* to compute the impulse response $h[n]$ and compare it with the one obtained above.

Answers: $a=0.5$; $h[n]=0.5^n u[n]$.

9.38 **Unit-step vs. impulse response**—The unit-step response of a discrete-time LTI system is

$$s[n]=2[(-0.5)^n-1]u[n].$$

Use this information to find:

(a) The impulse response $h[n]$ of the discrete-time LTI system.

(b) The response of the LTI system to a ramp signal $x[n]=nu[n]$.

Answers: $h[n]=-2(0.5)^n u[n-1]$.

9.39 **Convolution sum**—A discrete-time system has a unit impulse response $h[n]$.

(a) Let the input to the discrete-time system be a pulse $x[n]=u[n]-[n-4]$ compute the output of the system in terms of the impulse response.

(b) Let $h[n]=0.5^n u[n]$; what would be the response of the system $y[n]$ to $x[n]=u[n]-u[n-4]$? Plot the output $y[n]$.

(c) Use the convolution sum to verify your response $y[n]$.

(d) Use the function *conv* to compute the response $y[n]$ to $x[n]=u[n]-u[n-4]$. Plot both the input and the output.

Answers: Use the result that when the input is $x_1[n]=u[n]$ then the output is $y_1[n]=0.5^n(2^{n+1}-1)u[n]$.

9.40 **Discrete envelope detector**—Consider an *envelope detector* that would be used to detect the message sent in an AM system. Consider the envelope detector as a system composed of the cascading of two systems one which computes the absolute value of the input, and a second one that low-pass filters its input. A circuit that is used as an envelope detector consists of a diode circuit that does the absolute value operation, and an RC circuit that does the low-pass filtering. The following is an implementation of these operations in discrete time.

Let the input to the envelope detector be a sampled signal $x(nT_s)=p(nT_s)\cos(2000\pi nT_s)$ where

$$p(nT_s)=u(nT_s)-u(nT_s-20T_s)+u(nT_s-40T_s)-u(nT_s-60T_s)$$

two pulses of duration $20T_s$ and amplitude equal to one.

(a) Choose $T_s=0.01$, and generate 100 samples of the input signal $x(nT_s)$ and plot it.

(b) Consider then the subsystem that computes the absolute value of the input $x(nT_s)$ and compute and plot 100 samples of $y(nT_s)=|x(nT_s)|$.

(c) Let the low-pass filtering be done by a moving averager of order 15, i.e., if $y(nT_s)$ is the input, then the output of the filter is

$$z(nT_s)=\frac{1}{15}\sum_{k=0}^{14} y(nT_s-kT_s).$$

Implement this filter using the function *filter*, and plot the result. Explain your results.

(d) Is this a linear system? Come up with an example using the script developed above to show that the system is linear or not.

9.41 Two-dimensional convolution—Let the input be

$$x[m,n] = \delta[m,n] + 2\delta[m-1,n] + 3\delta[m-2,n] + 4\delta[m,n-1] + 5\delta[m-1,n-1] \\ + 6\delta[m-2,n-1] + 7\delta[m,n-2] + 8\delta[m-1,n-2] + 9\delta[m-2,n-2]$$

and the impulse response of an FIR filter be

$$h[m,n] = \delta[m,n] + \delta[m-1,n] - \delta[m,n-1] - \delta[m-1,n-1].$$

(a) Use the two-dimensional convolution function *conv2* to find the output $y[m,n]$.

(b) Is the impulse response separable, i.e., $h[m,n] = h_1[m]h_2[n]$? If so, determine the one-dimensional impulse responses $h_1[m]$, and $h_2[n]$ and use them to verify the above result.

Answer: Let $\mathbf{x} = [1\ \ 2\ \ 3; 4\ \ 5\ \ 6; 7\ \ 8\ \ 9]$ be an array representing $x[m,n]$.

9.42 Filtering and binarization—Use the function *imread, rgb2gray* and *double* to read the color image *peppers.png*. Convert it into a gray-level image $I[m,n]$ with double precision. Add noise to it using the function *randn* (Gaussian number generator) multiplied by 40. Create a one-dimensional moving-average filter with impulse response

$$h_1[n] = 0.25(\delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3]).$$

Letting a second filter have the same impulse response, i.e., $h_2[n] = h_1[n]$ create a separable two-dimensional filter $h[m,n] = h_1[m]h_2[n]$. Convolve the image with the two-dimensional filter to obtain an image $y[m,n]$.

(a) Use the function *histogram* to look at the distribution of the pixel values of the two images. Explain the effect of the moving-average filtering on the image $I[m,n]$.

(b) Binarizing an image consists in using a gray-level threshold to convert the given image into one that has two levels only. The threshold can be chosen from a histogram of the image to be binarized. We wish to binarize the $y[m,n]$ to obtain two images: $y_1[m,n]$ with gray level of 250 whenever the $y[m,n] > 100$ and 0 otherwise; $y_2[m,n]$ with gray level of 200 whenever $y[m,n] \leq 100$ and 0 otherwise. Use *imshow* to display these images, and *histogram* to display their gray-level distributions.

Answer: The noise is given by the array $40 * randn(M(1), M(2))$ where $M(1)$ and $M(2)$ are the dimensions of the image.

9.43 Edge detection—Detection of edges is a very important application in image processing. Taking the gradient of a two-dimensional function detects the changes the edges of an image. A filter than is commonly used in edge detection is Sobel's filter. Consider the generation of two related impulse responses

$$h_u[m,n] = \delta[m,n] - \delta[m-2,n] + 2\delta[m,n-1] - 2\delta[m-2,n-1] \\ + \delta[m,n-2] - \delta[m-2,n-2]$$

$$h_v[m,n] = \delta[m,n] + 2\delta[m-1,n] + \delta[m-2,n] - \delta[m,n-2] \\ - 2\delta[m-1,n-2] - \delta[m-2,n-2].$$

For a given input $x[m,n]$ the output of the Sobel is

$$y_1[m,n] = (h_u * x)[m,n], \quad y_2[m,n] = (h_v * x)[m,n]$$
$$y[m,n] = \sqrt{(y_1[m,n])^2 + (y_2[m,n])^2}.$$

(a) Use *imread* to read in the image *peppers.png*, and convert it into a gray image with double precision by means of the functions *rgb2gray* and *double*. Letting this image be input $x[m,n]$ implement the Sobel filter to obtain an image $y[m,n]$. Determine a threshold T such that whenever $y[m,n] > T$ we let a final image $z[m,n] = 1$ and zero otherwise. Display the complement of the image $z[m,n]$.

(b) Repeat the above for the image *circuit.tif*, choose a threshold T so that you obtain a binarized image displaying the edges of the image.

Answer: The Sobel filters are given by the arrays $hu = [1, 0, -1; 2, 0, -2; 1, 0, -1]$ and $hv = hu'$, the transpose of hu.