Chapter 3

CT and DT Systems

3.1 Systems as a mapping

In Chapter 1, we explored the mathematical representation of a variety of waveforms of interest, including sensor outputs, recorded data measurements or other information-bearing signals as either continuous-time or discrete-time signals. In this chapter, we develop the notion of a "system" as a means for transforming one signal into another signal. Specifically, a discrete-time system can be viewed as a transformation, or mapping, from one set of sequence values, called the input to the system, to another set of sequence values, called the output of the system.

For the discrete-time system shown in Figure 3.1, we can view the operation of the system as one that takes as input, the signal $x[n]$ and produces as output the signal $y[n]$. From this notation, the system enclosed within the box is may appear to be simply mapping each value of the input signal $x[n]$ into a corresponding output value $y[n]$, however this point of view would not capture systems that contain memory. For example, suppose that the system simply delayed the input signal by two samples by using simple delay elements, such that $y[n] = x[n-2]$. Rather than this sample-at-a-time thought process, we therefore must consider the system in the box as one that is capable of stepping outside the limitations of the time-index, and is capable of viewing the entire signal $x[n]$, from $n = -\infty$ to $n = \infty$, and then producing a new signal $y[n]$ which is also defined over the time index $n = -\infty$to $n = \infty$. This notion of a system as a general mapping from one signal spanning the entire time horizon onto another entire signal spanning the same time horizon will enable us to consider a wide variety of practical systems in general, and properties of a number of specific systems of interest in practice.

This general notion or definition of a system as a mapping from one signal into another can be used to describe transformations of both continuous-time as well as discrete-time signals, as shown in figure 3.2.

Now that we have a general viewpoint of a system as a mapping from one into another signal, we can more formally define a system by wrapping some mathematical language around this framework. Specifically, we define a system as follows

A **discrete-time system** is a mapping, or a transformation, T, that maps one discrete-time signal, called the input signal, $x[n]$, into another discrete-time signal, called the output signal, $y[n]$, such that $y[n] = T\{x[n]\}$. We will sometimes use this formal system notation, i.e. $y[n] = T\{x[n]\}$, and, when convenient, we may adopt the shorthand $x[n] \rightarrow y[n]$, which carries the same meaning. That is, both should be read, "When the input to the system is $x[n]$, the output of the system is $y[n]$."

A **continuous-time system** is a mapping, or a transformation, T, that maps one continuous-

$$x[n] \longrightarrow \boxed{\text{DT System}} \longrightarrow y[n]$$

Figure 3.1: A discrete-time system as a mapping from the input signal $x[n]$, to the output signal $y[n]$.

$$x(t) \longrightarrow \boxed{\text{CT System}} \longrightarrow y(t)$$

Figure 3.2: A continuous-time system as a mapping from the input signal $x(t)$, to the output signal $y(t)$.

$$x(t) \longrightarrow \underset{\uparrow T}{\boxed{\text{A/D}}} \longrightarrow x[n]$$

Figure 3.3: An A/D converter taking samples $x[n] = x(nT)$.

time signal, called the input signal, $x(t)$, into another continuous-time signal, called the output signal, $y(t)$, such that $y(t) = T\{x(t)\}$.

With these definitions of discrete-time and continuous-time systems, we can begin to explore some of the many properties of systems that process signals in discrete and continuous-time. However before we do, we will consider one more example of a system, but this system does not satisfy either of the definitions given above. Since one way in which discrete-time signals are created is through the process of sampling continuous-time signals with periodic sampling, we will briefly consider the ideal-sampling system as one that takes a continuous-time signal as its input and produces a discrete-time signal as its output. Later, when we consider analog-to-digital converters and digital-to-analog converters, we will explore these concepts more carefully, and will also consider a system that takes a discrete-time signal as its input and produces a continuous-time signal as its output.

3.2 Introduction to Sampling

One example of a *mixed-signal* system is an ideal analog-to-digital converter, or A/D converter in shorthand. While it is mathematically simple to describe the operation of an ideal A/D converter, as we will see when we discuss such systems in more detail, the design and operation of a practical A/D converter can be a complicated process. However, for our purposes here, we will stick with the simple, ideal case. In this case, we consider a system that takes an analog, continuous-time signal $x(t)$ as input and produces a discrete-time signal $x[n]$ as output, as shown in figure 3.3.

Mathematically, the discrete-time output $x[n]$ is related to the continupus-time input signal $x(t)$ through the relation $x[n] = x(nT)$,where the parameter T is called the *sampling period* of the A/D converter and has the sample units as the time index t of the input. While the relationship between the input and outputs of the ideal A/D converter can be succinctly stated, as we will see in later chapters of this text, under certain conditions the origional continuous-time signal $x(t)$ may be completely recovered from its samples $x[n]$. As you might imagine, this will place some restrictions on the class of signals $x(t)$, however we will see that this can be accomplished in a way that permits a wide range of possible input signals of interest to be sampled, stored, and completely recovered from it samples. One simple example of a set of signals that can be recovered easily from periodic samples is the class of polynomial signals, i.e. $x(t) = a_0 + a_1 t + a_2 t^2 + ... + a_N t^N$, which can be completely recovered from only $N+1$ samples. When the number of samples taken becomes large and even infinite, the class of signals that can be perfectly reconstructed from its periodic samples becomes rather large. However, as the class of signals becomes large and complex, more sophisticated means must be used to reconstruct the origional continuous-time signal.

For the system in Figure 3.3, we recall the when the input signal $x(t)$ is bandlimited, such that its Fourier transform magnitude satisfies $|X(\omega)| = 0$,for $|\omega| > 2\pi B$, then so long as the sampling rate is sufficiently high enough, i.e. $\frac{1}{T} > 2B$, then the signal $x(t)$ can be completely reconstructed from its samples, $x[n] = x(nT)$, as follows

$$x(t) = \sum_{-\infty}^{\infty} x[n] \text{sinc}(\frac{\pi}{T}(t - nT)). \tag{3.1}$$

This relationship enables us to reconstruct $x(t)$ from the samples $x[n] = x(nT)$, for bandlimited signals.

While this relationship is tremendously powerful, it will be equally important for us to see how the Fourier representations of the signals $x(t)$ and $x[n]$ are also related. We will show that the relationship between $X(\Omega)$, the CTFT of $x(t)$, and $X_d(\omega)$, the DTFT of $x[n]$ is given by

$$X_d(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X\left(\frac{\omega + 2\pi k}{T}\right), \tag{3.2}$$

which provides the following relationships between a continuous-time signal $x(t)$ and the sequence of samples $x[n] = x(nT)$,

$x(t)$	$\longrightarrow \boxed{\text{A/D}} \longrightarrow$ $\uparrow T$	$x[n]$
$x(t) = \sum_{n=-\infty}^{\infty} x[n]\text{sinc}\left(\frac{\pi}{T}(t - nT)\right)$	time domain	$x[n] = x(nT)$
$X(\Omega) = \begin{cases} X_d(\Omega T), & \lvert\Omega\rvert < \frac{\pi}{T} \\ 0 & \text{else} \end{cases}$	frequency domain	$X_d(\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X\left(\frac{\omega+2\pi k}{T}\right)$

We will show how these relationships can be derived, beginning in the lower right and working our way counterclockwise through the chart.

We begin with a continuous-time signal $x(t)$ that has a well-defined CTFT given by $X(\Omega)$. By taking periodic samples of the signal $x[n] = x(nT)$, we also assume that the signal $x(t)$ is well-defined at the sampling instants $t = nT$. Taking a closer look at the CTFT synthesis equation, and using the sampling instants ans the link, we have

$$\begin{aligned}
x[n] &= x(nT) \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\Omega) e^{j\Omega(nT)} d\Omega \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} X\left(\frac{\omega}{T}\right) e^{j\frac{\omega}{T}(nT)} d\frac{\omega}{T} \\
&= \frac{1}{2\pi T} \int_{-\infty}^{\infty} X\left(\frac{\omega}{T}\right) e^{j\omega n} d\omega \\
&= \frac{1}{2\pi T} \sum_{k=-\infty}^{\infty} \int_{0}^{2\pi} X\left(\frac{\omega + 2\pi k}{T}\right) e^{j\omega n} d\omega \\
&= \frac{1}{2\pi} \int_{0}^{2\pi} \underbrace{\left[\frac{1}{T} \sum_{k=-\infty}^{\infty} X\left(\frac{\omega + 2\pi k}{T}\right)\right]}_{X_d(\omega)} e^{j\omega n} d\omega \\
&= \frac{1}{2\pi} \int_{0}^{2\pi} \left[X_d(\omega)\right] e^{j\omega n} d\omega,
\end{aligned}$$

which yields the relationship of (3.2) by the uniqueness property of the DTFT and the last line above. The first line above arises from setting $t = nT$ in the CTFT synthesis equation; the second line comes from setting $\omega = \Omega T$, the fourth line comes from breaking the infinite integral into a sum of integrals of length 2πeach. It is helpful to consider what (3.2) represents graphically. We see three components of interest. First, the DTFT is represented by a scaled-frequency axis version of the original CTFT, that is $X(\omega/T)$; next we see that this scaled version is also scaled in amplitude by the factor $\frac{1}{T}$; and finally that copies if

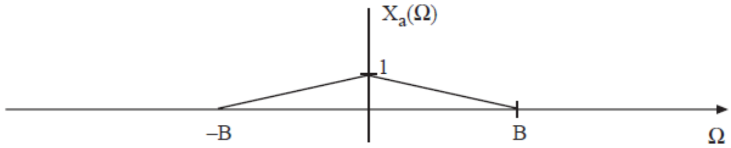


Figure 3.4: An example of a bandlimited CTFT of a signal $x_a(t)$.

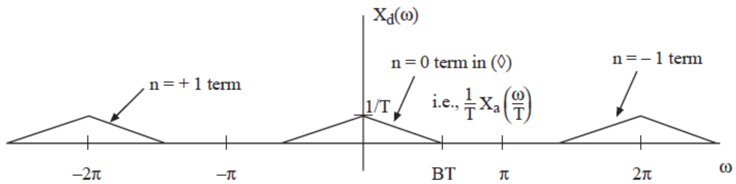


Figure 3.5: Graphical depiction of three of the terms in (3.2) for the bandlimited CTFT signal $x_a(t)$ shown in (3.4).

this amplitude and frequency scaled version of $X(\Omega)$ are placed at equally-spaced intervales on the ω axis. Note that if the original spectrum, $X(\Omega)$is zero for $|\Omega| > \Omega$, then this periodic replication simply provides the necessary 2π periodicity of the DTFT and within the interval $|\omega| < \pi$, we have the simpler relation

$$X_d(\omega) = \frac{1}{T} X\left(\frac{\omega}{T}\right), \quad |\omega| < \pi,$$

and is 2π periodic elsewhere. However, if the original spectrum $X(\Omega)$ is not bandlimited in this manner, then the summation of shifted replicas of the scaled spectrum $X(\frac{\omega+2\pi k}{T})$ each may have overlap into the interval $|\omega| < \pi$, and each of the terms in the summation must be taken carefully into account. We will consider several examples to illustrate this point.

As an example, let us consider a signal $x_a(t)$ that has a bandlimited spectrum as depicted in Figure 3.4.

When the sampling interval is such that $\frac{\pi}{T} > B$, the terms in the summation (3.2) do not overlap, and we have no detrimental "aliasing" of frequency components, which is what we the effect of copies of the CTFT that are centered outside the region from $-\pi$ to π in discrete frequency, i.e. the region occupied primarily by the $k = 0$ term from (3.2). When frequencies overlap, and masquerade in a band to which they do not belong, we are no longer able to determine those frequency components from the original continuous-time signal from those hiding behind an alias, owing to the periodic replication of the CTFT.

Note that since we assumed that $\frac{\pi}{T} > B$, and since the CTFT $X_a(\Omega)$ is zero for $|\Omega| > B$, then when we substitute $\frac{\omega}{T}$ for Ωin $X_a(\Omega)$, we have that $X_a(\frac{\omega}{T}) = 0$ for $|\omega| > BT$. As a result, the $k = 0$ term is confined to the region $-\pi < \omega < \pi$, and the $k = 1$ term is confined to the region $\pi < \omega < 3\pi$ and more generally, the k^{th} term is confined to the region $-\pi < \omega + k2\pi < \pi$, or $-\pi - k2\pi < \omega < \pi - k2\pi$. Note that when $BT > \pi$, the contribution from the $k = 0$ term and the $k = -1$ term will overlap. This is called "aliasing" and eliminates our ability to completely recover the continuous-time signal $x_a(t)$ from its samples. The condition $BT < \pi$ is equivalent to the Nyquist condition stated previously, since we have $BT < \pi \implies T < \frac{\pi}{B} \implies \frac{1}{T} > \frac{B}{\pi} \implies \frac{1}{T} > 2\left(\frac{B}{2\pi}\right)$, where $\frac{1}{T}$ is the sampling frequency in samples per second, and $\frac{B}{2\pi}$ is the bandwidth of $x_a(t)$ in Hz.

In a second example, we consider the case when $BT = \frac{5\pi}{3}$, or $T = \frac{5\pi}{3B}$, which indeed will give rise to aliasing. In this case we no longer have the simple relationship $X_d(\omega) = \frac{1}{T} X\left(\frac{\omega}{T}\right), |\omega| < \pi$, but rather need to use the more complicated expression (3.2), which contains the infinite sum of copies of $X(\Omega)$, scaled in amplitude, scaled in frequency, and shifted in frequency, $\frac{1}{T}X(\frac{\omega-2\pi k}{T})$. For this situation, with $BT = \frac{5\pi}{3}$, we can readily see (either graphically, or through simple calculation) that we need to account for the $k = -1, 0, 1$ terms in the summation. This can be graphically depicted as in Fig. 3.6.

The three terms that comprise the scaled, shifted spectrum, $\frac{1}{T}X_a(\frac{\omega+2\pi}{T}) + \frac{1}{T}X_a(\frac{\omega}{T}) + \frac{1}{T}X_a(\frac{\omega-2\pi}{T})$, when added together create the complete DTFT $X_d(\omega)$ shown in Figure3.7.

The resulting spectrum $X_d(\omega)$, no longer appears as a periodic replication of a scaled in amplitude and

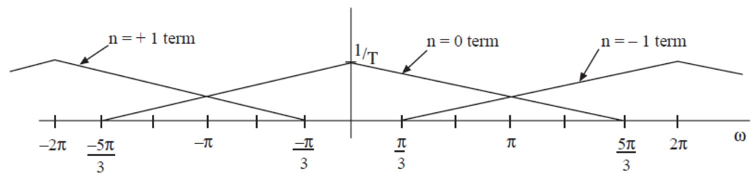


Figure 3.6: A graphical depiction of the various terms that contribute to the DTFT $X_d(\omega)$ is shown when the signal $x_a(t)$ with CTFT in 3.4 is sampled producing $x[n] = x_a(nT)$, for $BT = \frac{5\pi}{3}$. As seen graphically, the induced aliasing requires that three terms in the summation of (3.2).

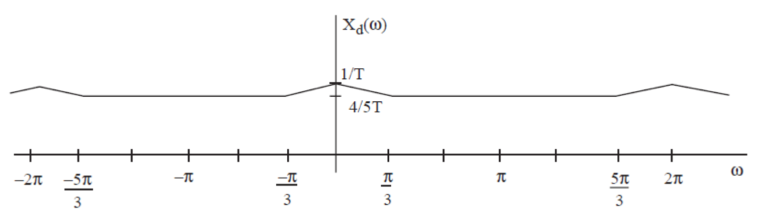


Figure 3.7: The DTFT $X_d(\omega)$ is shown when the signal $x_a(t)$ with CTFT in 3.4 is sampled producing $x[n] = x_a(nT)$, for $BT = \frac{5\pi}{3}$.

in frequency version of $X_a(\Omega)$, but rather a distorted version of this. This is because some of the higher frequencies from the copy of $X_a(\frac{\omega}{T})$ extend outside the range $|\omega| < \pi$ and similarly, some of the higher frequencies from the copy $X_a(\frac{\omega-2\pi}{T})$ extend into this frequency range from the right, and masquerade as lower frequencies. The same is true from the copy $X_a(\frac{\omega+2\pi}{T})$ from the left. This is the reason we use the term "aliasing" to describe this phenomenon.

If we were to select an even slower rate of sampling, i.e. a larger value of the sampling period T, then additional terms beyond these three would also need to be incorporated. For simplicity, we can exploit the periodicity of DTFTs and only concentrate on the terms $X_a(\frac{\omega-2\pi k}{T})$ that fall into the range $-\pi < \omega < \pi$, and then periodically replicate this resulding spectrum with periodicity 2πin ω. The net result will be the same as if we considered all of the terms in the infinite summation making up $X_d(\omega)$.

When there is no aliasing in the resulting spectrum and we have that $X_d(\omega) = \frac{1}{T}X_a(\frac{\omega}{T})$ in the range $-\pi < \omega < \pi$, then we can, at least conceptually, derive a reconstruction formula for recovering $x_a(t)$ perfectly from the samples $x[n] = x_a(nT)$. The "algorithm" for doing so would follow along these lines:

$$x_a(t) = \mathrm{CTFT}^{-1}\left[\begin{cases} T \times \mathrm{DTFT}x[n]|_{\omega=\Omega T} & |\Omega| < \frac{\pi}{T} \\ 0 & \text{otherwise} \end{cases}\right].$$

We can follow through this conceptual algorithm, mathematically, to produce the ideal reconstruction formula

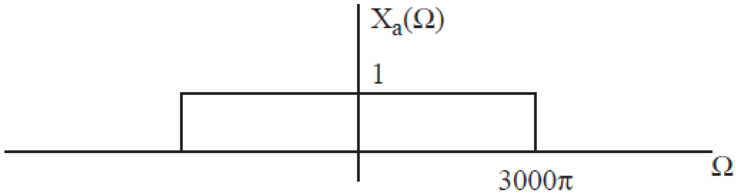


Figure 3.8: A bandlimited signal $x_a(t)$ with spectrum $X_a(\Omega)$ bandlimited to 3000π.

as follows.

$$
\begin{aligned}
x_a(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\begin{cases} T\times \mathrm{DTFT} x[n]|_{\omega=\Omega T} & |\Omega|<\frac{\pi}{T} \\ 0 & \text{otherwise}\end{cases}\right] e^{j\Omega t}d\Omega \\
&= \frac{T}{2\pi}\int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \sum_{n=-\infty}^{\infty} x[n]e^{-j(\Omega T)n}e^{j\Omega t}d\Omega \\
&= \frac{T}{2\pi}\sum_{n=-\infty}^{\infty} x[n]\int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} e^{j\Omega(t-nT)}d\Omega \\
&= \sum_{n=-\infty}^{\infty} x[n]\frac{\left[e^{j\pi(t-nT)/T}-e^{-j\pi(t-nT)/T}\right]}{2\pi j(t-nT)/T} \\
&= \sum_{n=-\infty}^{\infty} x[n]\frac{2j\sin(\pi(t-nT)/T)}{2\pi j(t-nT)/T} \\
&= \sum_{n=-\infty}^{\infty} x[n]\mathrm{sinc}(\frac{\pi}{T}(t-nT)), \qquad (3.3)
\end{aligned}
$$

where in the last three lines of the derivation, we exclude the case $t = nT$, and consider that case separately. When $t = nT$,we have in the integral in teh second line of the derivation, an integral of 1 over the interval of length $2\pi/T$. This, together with the scale factor $T/2\pi$ outside the integral tells us that the integral evaluates to one for $t = nT$, which is why we use the sinc function in the last line of the derivation, which then is valid for all values of t.

We consider another example in which we sample a continuous time signal to produce a discrete-time signal and observe how the frequency content of the signal as depicted in the CTFT is mapped to discrete time. Specifically, for the continuous time signal $x_a(t)$ with CTFT given by $X_a(\Omega) = 1$, for $|\Omega| < 3000\pi$, and zero elsewhere, i.e. as shown in Figure 3.8.

We consider sampling the signal $x_a(t)$ at sampling rates of $4kHz$, $6kHz$, and $12kHz$, i.e. $x[n] = x_a(nT)$ for $T = 1/4,000$, $T = 1/6,000$ and $T = 1/12,000$. As shown in Figure 3.9, we see that as the sampling rate increases (i.e. when T decreases) the spectrum $X_a(\Omega)$ is mapped to a smaller and smaller set of frequencies in discrete-time.

The ideal reconstruction formula given in 3.3 can be viewed as a mathematical description of the operation of an ideal digital-to-analog (D/A) converter, which we have referred to as an ideal discrete-to-continuous (D/C) converter. The process of ideal continuous-to-discrete conversion, depicted as

$$x[n] \longrightarrow \underset{\uparrow T}{\boxed{\text{D/C}}} \longrightarrow x(t)$$

can be thought of as a process of clocking out a specific pulse shape (a sinc function) at integer multiples of

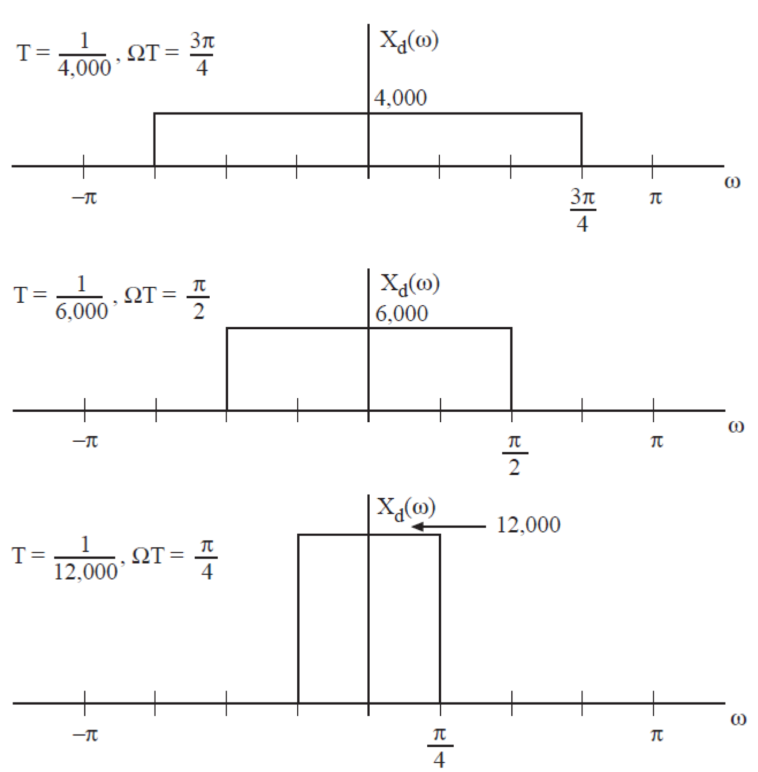


Figure 3.9: Three different DTFTs are generated when $x[n] = x_a(nT)$ for $T = 1/4,000$, $T = 1/6,000$ and $T = 1/12,000$.

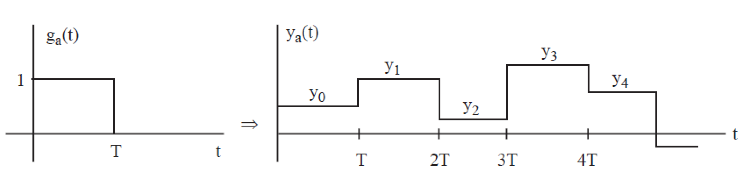


Figure 3.10: The reconstruction of $y_a(t) = \sum_{n=-\infty}^{\infty} x[n]g(t-nT)$, for $g(t)$ given by a rectangular pulse. This is called zero-order-hold (ZOH) reconstruction.

the sampling period T scaled by the samples $x[n]$. Mathematically, we can view the process as follows,

$$y(t) = \sum_{n=-\infty}^{\infty} x[n]g(t-nT),$$

where we consider an arbitrary reconstruction pulse $g(t)$ to be used in the C/D converter, and in this case we call the output $y(t)$ to indicate the possibility that $y(t)$ may differ from $x_a(t)$ from which the samples were taken to produce $x[n]$. When $g(t) = \text{sinc}(\pi(t - nT)/T)$, we have perfect reconstruction and $y(t) = x_a(t)$. However, more generally, we may want to consider how $y(t)$ relates to the original signal $x_a(t)$ and the discrete-time signal $x[n]$. For example, later in the text, we will consider the case when $g(t)$ is a rectangular pulse as shown in Figure 3.10.

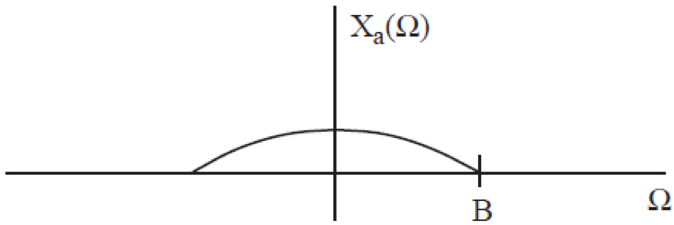


Figure 3.11: Bandlimited spectrum $X_a(\Omega)$.

Mathematically, we can explore the spectrum $Y_a(\Omega)$ from this reconstruction in general. We have

$$\begin{aligned}
y_a(t) &= \sum_{n=-\infty}^{\infty} x[n]g(t-nT) \\
Y_a(\Omega) &= \int_{-\infty}^{\infty} y_a(t)e^{-j\Omega t}dt \\
&= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n]g(t-nT)e^{-j\Omega t}dt \\
&= \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} g(t-nT)e^{-j\Omega t}dt \\
&= \sum_{n=-\infty}^{\infty} x[n]G(\Omega)e^{-j\Omega(nT)} \\
&= G(\Omega) \sum_{n=-\infty}^{\infty} x[n]e^{-j(\Omega T)n} \\
&= G(\Omega)X_d(\Omega T),
\end{aligned}$$

where in the fifth line, we used the delay property of CTFTs, for the CTFT of $g(t-nT)$, and in the sixth line we observe the expression for the DTFT of $x[n]$ with the frequency variable ΩT. We note that the resulting expression

$$Y_a(\Omega) = G(\Omega)X_d(\Omega T)$$

is valid for all $-\infty < \Omega < \infty$. When $G(\Omega)$ is an ideal lowpass filter with cutoff frequency $\frac{\pi}{T}$, then only a single period of the periodic spectrum $X_d(\Omega T)$ remains. However for general $G(\Omega)$, we may observe not only the period that is centered at $\Omega = 0$, but also those centered at integer multiples of $\frac{2\pi}{T}$, filtered by $G(\Omega)$. When the signal $x[n]$ arises from sampling $x_a(t)$, we can substitute our expression for $X_d(\omega)$ in terms of $X_a(\Omega)$ and obtain

$$Y_a(\Omega) = G(\Omega)\frac{1}{T}\sum_{k=-\infty}^{\infty} X_a(\frac{\omega+2\pi k}{T}).$$

If the signal $x_a(t)$ were bandlimited to $B < \pi/T$, then we may have a spectrum as depicted in Figure 3.11.

Since $B < \pi/T$, there is no aliasing so the resulting $X_a(\Omega T)$ appears as shown in Fig. 3.12.

Since there is no aliasing, we ahve that $\sum_{k=-\infty}^{\infty} X_a(\frac{\Omega T+2\pi k}{T}) = X_a(\Omega)$ in the range $-\frac{\pi}{T} < \Omega < \frac{\pi}{T}$. For an ideal D/C converter, i.e., for $g(t) = \text{sinc}(\pi(t-nT)/T)$, we have that $G(\Omega) = T$ in the range $-\frac{\pi}{T} < \Omega < \frac{\pi}{T}$, and is zero elsewhere, i.e. $G(\Omega)$ is as shown in Figure 3.13.

We can now graphically observe the effect of using this reconstruction filter in the D/C converter. As shown in Figure 3.14, the ideal reconstruction filter passes through only the term centered at $\Omega = 0$, and rejects all other periodic replications of $X_a(\Omega)$.

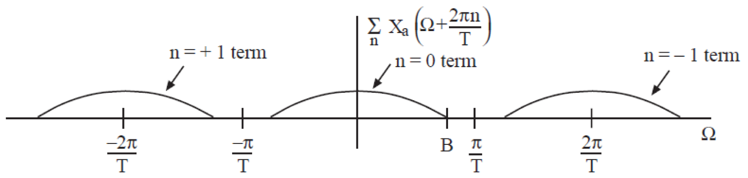


Figure 3.12: The discrete-time spectrum that results from sampling $x_a(t)$ at rate $1/T$.

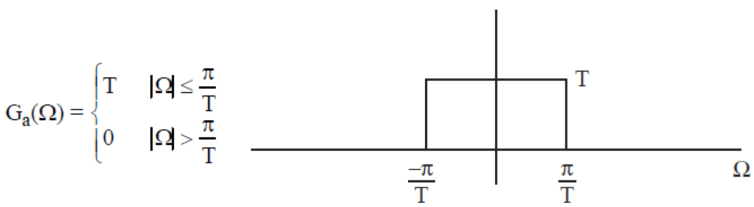


Figure 3.13: The CTFT of the ideal reconstruction filter $G(\Omega)$ in an ideal D/C converter.

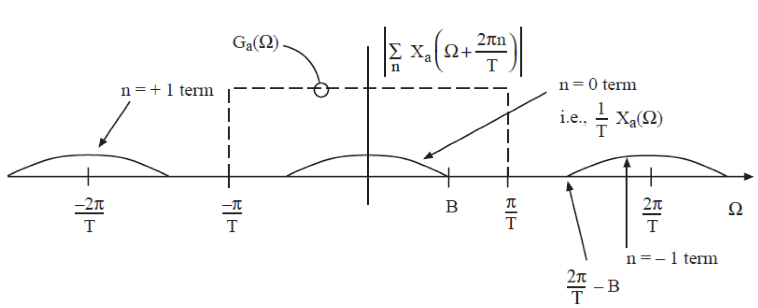


Figure 3.14: Graphical depiction of the operation of an ideal reconstruction filter in an ideal D/C converter.

As a result, we have for the CTFT of the output,

$$\begin{aligned} Y_a(\Omega) &= G(\Omega)\frac{1}{T}\sum_{k=-\infty}^{\infty} X_a(\Omega + \frac{2\pi k}{T}) \\ &= \begin{cases} X_a(\Omega), & |\Omega| < \frac{\pi}{T} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Therefore, if $X_a(\Omega)$ is bandlimited to $\frac{\pi}{T}$, then we have that $Y_a(\Omega) = X_a(\Omega)$, and $y_a(t) = x_a(t)$, i.e. we have perfect reconstruction of the original continuous-time signal from its samples.

3.3 Some Examples of Discrete-time Systems

Before we investigate some of the properties of discrete-time systems, we begin by exploring some simple systems through their input-output relationships. Along the way, we will introduce a few important concepts. For example, a system with input $x[n]$ and output $y[n]$ is referred to as memoryless if the output at time n is only a function of the input $x[n]$ at the same value of n. For example, consider the following memoryless systems.

System 3.1 One example of a memoryless system is a simple amplification system, i.e. one for which the input-output relationship is given by

$$y[n] = ax[n],$$

where a is a real-valued constant.

System 3.2 Another example of a memoryless system might be as follows

$$y[n] = |x[n]|,$$

where once again, the output at time n is completely determined by the input at time n.

Systems for which the output at time n depend on more information than the current value of the input are said to have memory. We call the minimal set of information in addition to the values of the input $x[n],\ n \geq m$ required to uniquely determine the output $y[n]$ for all time $n \geq m$ the **state** of the system. For example, in the following system, the state of the system consists of $y[n-1]$.

System 3.3 A system that has memory requires storage of the state of the system in order to compute its output. In this system, the state is one single previous value of the output.

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

System 3.4 Another example of a system with state is a simple delay.

$$y[n] = x[n-n_0].$$

here, the state of the system is the previous values of the input, $x[n-1], x[n-2], \ldots, x[n-n_0]$. Storing only the single value $x[n-n_0]$ would enable us to compute only one value of $y[n]$, but if $n_0 > 1$, we would be unable to proceed further. As a result, we need to store all intervening values of $x[n]$ in order to uniquely determine $y[n]$, given only future values of the input.

3.4 Linear Systems

While there are a myriad of properties of systems that are of importance to study, perhaps the property of primary importance is ability to distinguish between linear and nonlinear systems. This is largely because some of the mathematical tools we will develop apply only to linear systems. The series of definitions below culminates in the definition of linearity.

A system satisfies the **decomposition property** if its output $y[n]$ can be written as follows

$$y[n] = y_x[n] + y_s[n],$$

where $y_x[n]$ is the response of the system due only to the input $x[n]$, while the initial conditions of the system are set to zero, i.e., all state in the system (if there is any) is set to have zero values in each position, and $y_s[n]$ is the response of the system due only to the state of the system with the input set to zero.

When the output of the system can be decomposed into these two responses, we call $y_x[n]$ the zero state response (ZSR) of the system and $y_s[n]$ the zero input response (ZIR) of the system. We first consider a property called **zero-state linearity**, which considers the input-output functionality of the system when the initial conditions, or state, of the system has been initially set to zero.

A system is **zero-state linear**, if , when the initial conditions of the system are set to zero before the input is applied, it is satisfies both homogeneity and additivity, defined as follows.

A system with input $x[n]$ and output $y[n]$ satisfies **homogeneity** if for every input $x[n]$ and for every positive constant a, the following holds,

$$\text{if } x[n] \rightarrow y[n], \text{ then } ax[n] \rightarrow ay[n].$$

This is also sometimes referred to as the scaling property of systems.
A system with input $x[n]$ and output $y[n]$ satisfies **additivity**, if for every pair of inputs $x_1[n]$ and $x_2[n]$, and their corresponding outputs, $y_1[n] = T\big\{x_1[n]\big\}$ and $y_2[n] = T\big\{x_2[n]\big\}$, the following holds

$$\text{if } x_1[n] \rightarrow y_1[n] \text{ and } x_2[n] \rightarrow y_2[n], \text{ then } x_1[n] + x_2[n] \rightarrow y_1[n] + y_2[n].$$

When the two properties of homogeneity and additivity are combined into a single form, we obtain a more compact relation known as the **superposition** property of linear systems. This is summarized as follows,

A system with input $x[n]$ and output $y[n]$ satisfies **superposition** if for every pair of inputs $x_1[n]$ and $x_2[n]$, and their corresponding outputs, $y_1[n] = T\big\{x_1[n]\big\}$ and $y_2[n] = T\big\{x_2[n]\big\}$, the following holds

$$\text{if } x_1[n] \rightarrow y_1[n] \text{ and } x_2[n] \rightarrow y_2[n], \text{ then } ax_1[n] + b[n] \rightarrow ay_1[n] + by_2[n],$$

for all real-valued constants a and b.

For systems that satisfy the decomposition property, we are able to set aside the response of the system to any intial conditions (or initial state of the system) and study the input-output behavior of the system due solely to the input. When the system is also **zero-input linear**, then we can extend the notions of linearity to the initial conditions of the system and the responses to these initial conditions.

A system satisfying the decomposition property with a set of N initial conditions $\big\{y_k[n_k] = c_k\big\}_{\{k=0\}}^{N}$ and the corresponding zero input response to these intial conditions $y_{s,1}[n]$ and a second set of N initial conditions $\big\{y_m[n_m] = d_m\big\}_{\{m=0\}}^{N}$ and the corresponding zero input response to these intial conditions $y_{s,2}[n]$ is **zero-input linear** if the following holds,

$$\text{if } \big\{y_k[n_k] = c_k\big\}_{\{k=0\}}^{N} \rightarrow y_{s,1}[n] \text{ and } \big\{y_m[n_m] = d_m\big\}_{\{m=0\}}^{N} \rightarrow y_{s,2}[n] \text{ then}$$

$$\big\{y_k[n_k] = ac_k + bd_k\big\}_{\{k=0\}}^{N} \rightarrow ay_{s,1}[n] + by_{s,2}[n],$$

for all real-valued constants a and b.

We can now more formally define a linear system as follows.

> A discrete-time system is **linear** if it satisfies the decomposition property and it is both zero-state linear and zero-input linear.

In this section we focus primarily on zero-state linearity, since the decomposition property enables us to park issues of initial conditions outside our immediate focus and return to treat them later as necessary. As we will later see, when systems are not only linear, but also stable, the steady state behavior (long term behavior) will depend little on the effects of the initial conditions, and it is precisely the input-ouput zero-state behavior that is often of primary interest. We continue our exploration of linear systems by considering a few examples.

Example 1 Determine whether the system described by $y[n] = |x[n]|$ is linear or nonlinear. This system satisfies neither homogeneity nor additivity and is therefore nonlinear. To prove the failure of homogeneity note that $x_1[n] = 1, \forall n$ and $x_2[n] = -1 \forall n$ produce the same output and yet $x_2[n] = -x_1[n]$. Similarly additivity fails because $x_1[n] + x_2[n]$ does not produce the sum of the outputs due to $x_1[n]$ and $x_2[n]$ acting individually.

Example 2 Consider a system described by $y[n] = [x[n-4]]^2/x[n]$. Is this system linear or nonlinear? Check homogeneity: $ax[n] \rightarrow [ax[n-4]]^2/[ax[n]] = ax[n-4]^2/x[n] = ay[n]\checkmark$. So the system satisfies homogeneity. But, it looks like additivity will fail. Therefore, we must conclude that the system is not linear. Let us find an $x_1[n]$and $x_2[n]$ to demonstrate this. Let $x_1[n] = 1, \forall n$. Then we have that

$$y_1[n] = \frac{1^2}{1} = 1 \forall n.$$

Now let $x_2[n] = (\frac{1}{2})^n, \forall n$, which gives rise to

$$y_2[n] = \frac{[(\frac{1}{2})^{n-4}]^2}{(\frac{1}{2})^n} = \frac{(\frac{1}{2})^{2n} 2^8}{(\frac{1}{2})^n} = 2^8 (\frac{1}{2})^n, \forall n.$$

For the input $x_1[n] + x_2[n] = 1 + (\frac{1}{2})^n, \forall n$, we have

$$y[n] = \frac{[1 + (\frac{1}{2})^{n-4}]^2}{[1 + (\frac{1}{2})^n]} \neq y_1[n] + y_2[n] = 1 + 2^8 (\frac{1}{2})^n, \forall n,$$

since, for $n = 4$,we have

$$y[4] = \frac{[1+1]^2}{[1 + (\frac{1}{2})^4]} = \frac{4}{1 + \frac{1}{16}} \neq y_1[n] + y_2[n] = 1 + 2^8 (\frac{1}{2})^4 = 1 + 2^4.$$

Example 3 For the following averaging filter, we have $y[n] = \frac{1}{3}[x[n-1] + x[n] + x[n+1]]$. This is a simple example of a low-pass digital filter that could be used to smooth signals and reduce noise by replacing each sample in the output by the average of three adjacent values of the input signal. Is this system linear or nonlinear? We can seek the answer to this by checking superposition, i.e. by checking homogenity and additivity all at once. We have that

$$x_1[n] \rightarrow y_1[n] = \frac{1}{3}[x_1[n-1] + x_1[n] + x_1[n+1]],$$

and

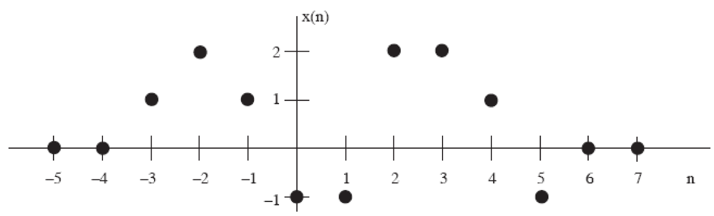
$$x_2[n] \rightarrow y_2[n] = \frac{1}{3}[x_2[n-1] + x_2[n] + x_2[n+1]].$$

Now for an input that is a linear combination, we have

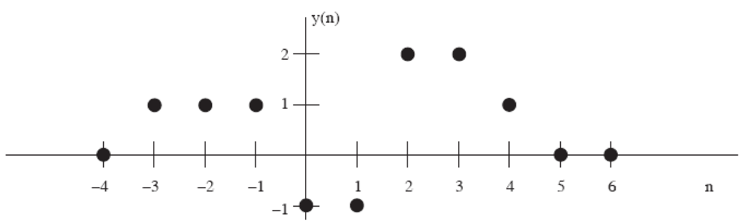
$$\begin{aligned} ax_1[n] + bx_2[n] &\rightarrow y[n] = \frac{1}{3}[(ax_1[n-1] + bx_2[n-1]) + (ax_1[n] + bx_2[n]) + (ax_1[n+1] + bx_2[n+1])] \\ &\rightarrow y[n] = a\frac{1}{3}[x_1[n-1] + x_1[n] + x_1[n+1]] + b\frac{1}{3}[x_2[n-1] + x_2[n] + x_2[n+1]] \\ &\rightarrow y[n] = ay_1[n] + by_2[n] \checkmark. \end{aligned}$$

So we have shown that the sytem is indeed linear, since the inputs $x_1[n]$ and $x_2[n]$ and coefficients a and b are arbitrary.

Example 4 A median filter is often used in data analysis when there may be outliers in the data, i.e. spurious samples that might be erroneous or artificially large or small, such that a local average of the data would be dominated by their magnitude. For example, we might employ the median filter $y[n] = \text{med}\,\{x[n-1], x[n], x[n+1]\}$. This operation would amount to looking at each value of the input sequence, and replacing each value by the middle value, in numerical order, of the current, most recent, and next, value of the imput. Thus, for example, the input sequence



would produce the median filter outut



Notice that for any value of n, the median filter output is always equal to one of the elements of the input sequence $\{x[n]\}_{\{n=-\infty\}}^{\infty}$. It is easy to visualize the output $\{y[n]\}_{\{n=-\infty\}}^{\infty}$ by mentally sliding a length-three window over the input sequence and then simply taking the output to be the middle element (in algebraic value) among those three input elements falling within the window. So, for example shown, $y[-2] = \text{med}\{x[-3], x[-2], x[-1]\} = \text{med}\{1, 2, 1\} = 1$, and $y[1] = \text{med}\{x[0], x[1], x[2]\} = \text{med}\{-1, -1, 2\} = -1$. Is the median filter in this example linear? Owing to properties of the median, homogeneity would be satisfied, since scaling the samples in an ordered set maintains the ordering with the possibility of reversing the elements if the sign of the scale factor is negative, however this would leave the median of the scaled values equal to the scaled median of the original values. What about additivity? Perhaps we can find a simple set of two input sequences for which we can demonstrate a violation of additivity. Consider for example

$$x_1[n] = \delta[n] \text{ and } x_2[n] = \delta[n-1],$$

for which the output of our three point local median filter would be

$$y_1[n] = 0, \forall n, \text{ and } y_2[n] = 0 \forall n,$$

however, for the sum of these two inputs, we have that

$$y[n] = T\{\delta[n] + \delta[n-1]\} = \delta[n] + \delta[n-1] \neq y_1[n] + y_2[n] = 0 \forall n,$$

and hence the system is not linear. Median filters are often useful in image processing, because unlike linear averaging filters, median filters can remove noise while preserving edge structure. Linear filters tend to blur edges, which is often objectionable in image processing. In the 1-D case, it is easy to see that median filters preserve edges. Consider the edge signal $x[n] = u[n]$. For this signal, the output of the median filter would be $y[n] = u[n]$, thus preserving the edge. However, consider the output of the three-point local averaging filter from example 3 above, for which the output would be $y[n] = \frac{1}{3}\delta[n+1] + \frac{2}{3}\delta[n] + u[n-1]$, which is a blurred version of the original edge.

Example 5 For a modulator of the form $y[n] = \cos(\omega_0 n)x[n]$, we may again check linearity directly using the superposition property. For the pair of inputs

$$x_1[n] \to y_1[n] = \cos(\omega_0 n)x_1[n] \text{ and } x_2[n] \to y_2[n] = \cos(\omega_0 n)x_2[n],$$

we have that

$$\begin{aligned} ax_1[n] + bx_2[n] \quad &\to y[n] \quad = \cos(\omega_0 n)(ax_1[n] + bx_2[n]) \\ &\to y[n] \quad = a\cos(\omega_0 n)x_1[n] + b\cos(\omega_0 n)x_2[n] \\ &\to y[n] \quad = ay_1[n] + by_2[n] \checkmark. \end{aligned}$$

So the modulator system is indeed linear.

Example 6 A linear constant-coefficient difference equation (LCCDE) is given by the following input-output relation

$$y[n] + a_1 y[n-1] + \ldots a_N y[n-N] = b_0 x[n] + \ldots b_M x[n-M],$$

which can be more compactly written

$$y[n] + \sum_{k=1}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]. \tag{3.4}$$

This can be shown to be linear more easily when we have considered z-transforms later in this text, however we can show linearity with some facility with such relations. Consider the input-output pair $x_1[n] \to y_1[n]$ and the input-output pair $x_2[n] \to y_2[n]$. We have that each satisfy the LCCDE individually, i.e.

$$y_1[n] + \sum_{k=1}^{N} a_k y_1[n-k] = \sum_{k=0}^{M} b_k x_1[n-k], \text{ and } y_2[n] + \sum_{k=1}^{N} a_k y_2[n-k] = \sum_{k=0}^{M} b_k x_2[n-k].$$

Now by considering the input $x_3[n] = cx_1[n] + dx_2[n]$,we also know that the system must satisfy, by the definition of the system,

$$y_3[n] + \sum_{k=1}^{N} a_k y_3[n-k] = \sum_{k=0}^{M} b_k (cx_1[n-k] + dx_2[n-k]).$$

From the definition of the system, if we take a linear combination of the left hand side of the relations for $y_1[n]$ and $y_2[n]$,we obtain

$$\begin{aligned} c\left(y_1[n] + \sum_{k=1}^{N} a_k y_1[n-k]\right) + d\left(y_2[n] + \sum_{k=1}^{N} a_k y_2[n-k]\right) \quad &= \quad c\sum_{k=0}^{M} b_k x_1[n-k] + d\sum_{k=0}^{M} b_k x_2[n-k] \\ (cy_1[n] + dy_2[n]) + \sum_{k=1}^{N} a_k (cy_1[n-k] + dy_2[n-k]) \quad &== \quad \sum_{k=0}^{M} b_k (cx_1[n-k] + dx_2[n-k]) \\ y_3[n] + \sum_{k=1}^{N} a_k y_3[n-k] \quad &= \quad \sum_{k=0}^{M} b_k (cx_1[n-k] + dx_2[n-k]), \end{aligned}$$

where the last line follows from the definition of $y_3[n]$ as the response of the system to the input $x_3[n]$ and the LCCDE for the system definition. Putting these together, we have from the second and third lines above that the corresponding output $y_3[n]$ satisfies $y_3[n] = cy_1[n] + dy_2[n]$, which indeed demonstrates linearity of the LCCDE.

3.5 Shift-invariant systems

Another property that will prove immensely powerful in the analysis of discrete-time systems is that of shift-invariance. The ability to characterize the input-output behavior of discrete-time systems would be rather difficult if it were necessary to completely recharacterize the properties of the system depending on the precise time at which a given input was presented. For example, imagine how complicated it would be if every time a driver were to consider driving an automobile, that the driver would need to consult the time of day and make changes appropriately. Suppose that the steering wheel was on the left side of the car during odd-valued hours of the day, but on the right hand side of the car on even-valued hours of the day. Additionally, perhaps the gas pedal would be to the right during the first quarter hour and then in the middle during the second quarter hour, and then on the left for the remainder of the hour. The brake and clutch pedals similarly moving about might cause no end to the confusion and difficulty in vehicle operation. Needless to say, the invariance of the input-output properties of the car, i.e. the driving behavior of the car, is one of the properties of a car that enable a driver to not only operate their vehicle routinely without too much distraction, but also to enable any driver of a car to drive any other car.

A system is **shift invariant** if a shift in the input always leads to a corresponding shift in the output, i.e., the system satisfies

$$\text{if } x[n] \to y[n], \text{ then } x[n-n_0] \to y[n-n_0], \forall n_0 \text{ and } \forall x[n].$$

The systems in Examples 1, 2, 3 and 5 are shift-invariant. The system in Example 4 is shiftvarying. Proving that a discrete-time system is indeed shift-invariant or that it is shift varying can be a challenge the first time around, however if you follow the steps taken in the following example, you will come to fewer stumbling blocks. Let us first revisit Example 3.

Example 3 (revisited) For the system defined by the output relation $y[n] = \frac{1}{3}[x[n-1] + x[n] + x[n+1]]$, we consider a shifted version of the input, say, $x_0[n] = x[n-n_0]$. For this input, we can find the corresponding output as follows

$$\begin{aligned} x_0[n] \quad &\to \quad y_0[n] = \frac{1}{3}[x_0[n-1] + x_0[n] + x_0[n+1]] \\ &\to \quad y_0[n] = \frac{1}{3}[x[(n-n_0)-1] + x[(n-n_0)] + x[(n-n_0)+1]] \\ &\to \quad y_0[n] = y[n-n_0] \checkmark \end{aligned}$$

where the last equality follows from the system definition. Therefore the system is shift-invartiant.

Example 5 (revisited) We revisit the modulation example given by

$$y[n] = \cos(\omega_0 n)x[n]$$

and consider whether it is shift invariant or not. Let us once again let $x_0[n] = x[n-n_0]$ and consider the system response to this input, we have

$$\begin{aligned} y_0[n] &= \cos(\omega_0 n)x_0[n] \\ &= \cos(\omega_0 n)x[n-n_0] \\ &\neq y[n-n_0] = \cos(\omega_0(n-n_0))x[n-n_0], \end{aligned}$$

so the system is not shift invariant. Note that while the result may hold for certain values of n_0, for example, if $\omega_0 = 2\pi/3$, then for $n_0 = 3$, the output would be equivalent to a shifted-by-three version of the input. However, for shift invariance to hold, the output must be a shifted version of the input for all possible (integer) values of n_0. In this case, shifting the input does not shift the cosine modulation, and as a result the system is shift-varying.

3.6 Causal Systems

A system is **causal** if for every n, $y[n]$ depends only on $x[m]$, $m \leq n$. Thus, for causal systems, current outputs do not depend on future inputs. Systems that are not causal are called noncausal. For systems for which the independent variable n is indeed a time variable, noncausal systems are not physically realizable if the output $y[n]$ must be computed immediately upon acquiring $x[n]$. However, in many DSP systems, data $x[n]$ is acquired and stored before processing (e.g., stored as an image $\{x[n,m]\}$ prior to processing, as in a digital camera). These systems can be noncausal without violating any physical laws of nature. The systems in Examples 1 and 2 are causal. The systems in Examples 3 and 4 are noncausal, since they require $x[n+1]$ in order to compute the three-sample average or median, respectively. The system in Example 6 can be either causal or noncausal, depending on the "direction" in which the equation is iterated. For example, rewriting (3.4) so that $y[n-kN]$ is computed from $x[n]$ and $y[n], y[n-1], \ldots, y[n-N+1]$,suggests a noncausal realization, i.e.

$$y[n-N] = \frac{1}{a_N}\left[-y[n] - \sum_{k=1}^{N-1} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]\right].$$

So an LCCDE describing the relationship between the sequences $y[n]$ and $x[n]$ can be either causal or noncausal, depending on the specific implementation of the LCCDE. Unless specified otherwise, we will typically assume that such an LCCDE corresponds to the causal implementation, i.e. the implementation for which the output $y[n]$ is computed in terms of the present and past values of the input and past values of the output, i.e. using an algorithm in the form of (3.4).

Example 7 For the following system, described by

$$y[n] = \frac{x[n]}{x[5]},$$

the system is nonlinear, shift-varying, and noncausal.

Example 8 The system described by the relation,

$$y[n] = x[-n],$$

corresponds to a time-reversal of the imput. This system is linear, shift-varying, and noncausal.

Example 9 The system whose input-output relation is

$$y[n] = x[|n|],$$

is linear, shift-varying, and noncausal.

3.7 LSI systems and convolution

We have seen that systems that are linear satisfy superposition, that is, they satisfy homogeneity (scaling) and additivity. We have also see that systems that are shift-invariant will generate a shifted version of their output when their input is shifted accordingly. Together, these two properties make up an important class of systems that we will explore, namely, linear shiftinvariant systems, or LSI systems. In many contexts, the independent variable of a sequence is referred to as time, and the term linear time invariant (LTI) is also used to describe such systems. For discrete-time sequences, we may refer to the time index of a sequence, but realize that the properties of discrete-time signals and systems hold more generally for sequences indexed on a wide variety of non-time based indexes, such as computer memory storage location, trading day in the stock market, individuals in a line of people, units of a product on an assembly line, pixels of a digital image stored in an array, antenna elements in a phased-array, and many, many more examples.

Let us return to the problem of identifying the response of a discrete-time LSI system to an arbitrary input by recalling the notionof shift-invariance in discrete-time systems. Linear, constant coefficient difference equations describe shift-invariant systems, and as such, we are particularly interested in the properties of systems that can be expressed in terms of such difference equations. Recall that when zero initial conditions are applied, i.e. when a system is initially at rest, that a system is *shift-invariant*, if the response to an input sequence $\{x[n]\}_{n=-\infty}^{\infty}$produces the output sequence $\{y[n]\}_{n=-\infty}^{\infty}$and for any$n_0$, the response to the same input, delayed by n_0, $\{x[n-n_0]\}_{n=-\infty}^{\infty}$ produces the same output sequence, delayed by n_0, $\{y[n-n_0]\}_{n=-\infty}^{\infty}$. Graphically this is depicted as

$$x[n] \longrightarrow \boxed{\text{system}} \longrightarrow y[n] \implies x[n-n_0] \longrightarrow \boxed{\text{system}} \longrightarrow y[n-n_0].$$

These two properties, when taken together, yield the powerful relationship between the input and the output of an LSI system known as the convolution sum. In the general case, the relationship between the input and output in an LSI system is given by the convolution sum,

$$\boxed{y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k],} \tag{3.5}$$

where the sequence $h[n]$ is the response of the system to the unit sample function, or discrete time impulse, $\delta[n]$. As a result, $h[n]$ often referred to as the impulse response, or unit sample response, of the discrete-time LSI system. By making the change of variable, $m = n - k$,we obtain an equivalent form of the convolution sum,

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

which we will write in short-hand notation as $y[n] = x[n] * h[n]$,where $h[n]$ is system unit pulse response (or impulse response) of the LSI system. This notation is somewhat an "abuse of notation", in that the output at time n appears to depend only on the input and impulse response at time n. As shown in Equation (3.5), the output at time n actually depends on the entire input sequence and the entire impulse response sequence. Alternative notation that would, perhaps be more explicit in showing this relationship could be $\{y[n]\}_{n=-\infty}^{\infty} = \{x[n]\}_{n=-\infty}^{\infty} * \{h[n]\}_{n=-\infty}^{\infty}$, however we will often drop this explicit sequence notation and assume that in general when we write $y[n]$, we are referring to the entire sequence, and not just the value at a given time n, unless we make this explicit from the context. A perhaps even more correct notation would note that the convolution operation in Equation (3.5)actually produces one sequence, and if we would like to refer to a given time instance of that sequence, we might write $y[n] = (x * h)[n]$, for simplicity, we will stick with the more standard, $y[n] = x[n] * h[n]$.

The convolution sum, Equation (3.5), can be shown as a consequence of the properties of linearity and shift invariance, and as a result, we could even define an LSI system as one whose input and output satisfy the convolution sum. That is,

$$\boxed{y[n] = \sum_{m=-\infty}^{\infty} h[m]x[n-m] \Leftrightarrow \text{The system is LSI.}}$$

The way to show this involves writing the input $x[n]$ in terms if impulses and applying both linearity and shift invariance to the resulting output. Specifically, we write the input as

$$\begin{aligned} x[n] &= \cdots + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + x[2]\delta[n-2] + \cdots \\ &= \sum_{k=-\infty}^{\infty} x[k]\delta[n-k], \end{aligned}$$

which can be viewed as a superposition of delayed and scaled discrete-time impulses, where the amplitide of each discrete-time impulse is scaled by the value of $x[n]$ at that time. In this manner, we can view the entire sequence $x[n]$ as a linear superposition of simpler sequences, where each of these simpler sequences has only one non-zero sample. The weighting in the linear superposition is simply the corresponding value of the sequence $x[n]$. We can now use linearity to write the output of an LSI system in response to the input $x[n]$ as a sum of the responses to each of the delayed and scaled impulses. Specifically, for a linear system, if we know the response to an impulse at time $n = k$, is, say $h_k[n]$, then we also know the response of the system to the input $x[k]\delta[n-k]$. By applying the homogeneity property of linear systems, we know that

$$\delta[n-k] \rightarrow h_k[n] \implies x[k]\delta[n-k] \rightarrow x[k]h_k[n].$$

Now by the additivity property of linear systems, we also can construct the response to the entire input $x[n]$, by adding up the responses to each of the simpler signals that make up the input, that is,

$$\delta[n-k] \rightarrow h_k[n] \implies \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \rightarrow \sum_{k=-\infty}^{\infty} x[k]h_h[n],$$

which corresponds to

$$x[n] \rightarrow \sum_{k=-\infty}^{\infty} x[k]h_h[n],$$

where $h_k[n]$ is the response of the linear system to a discrete-time impulse at time $n = k$. Now in order to construct the output due to an arbitrary input, we would need to know$h_k[n]$ for all possible values of k, for which the input is non-zero. This would indeed be too much information to keep track of. However, if a system is also shift-invariant, in addition to being linear, then we know that the response of the system due to an impulse at time k is just a delayed version of the response of the system due to an impulse at time 0. That is,

$$\delta[n-k] \rightarrow h_0[n-k] \triangleq h[n-k],$$

where we drop the subscript, since through shift-invariance, we have that $h_k[n] = h[n-k]$. By adding homogeneity, we have that

$$x[k]\delta[n-k] \rightarrow x[k]h[n-k],$$

for each value of k, and more generally, using superposition and the representation of $x[n]$ as a superposition of delayed and scaled discrete time impulses, that

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

which proves the convolution sum representation of the input-output relationship for LSI systems.

3.8 Properties of DT LSI Systems

Given that we can completely express the nature of LSI systems through the convolution sum, we can now go back and see how properties of LSI systems affect the resulting convolution sum representation. As a result, we will see that we will be able to deduce properties of LSI systems directly by observing the impulse response as it appears in the convolution sum. We begin by considering the property of causality in LSI systems and consider how the convolution sum representation of a causal system may differ from that of a non-causal system.

Recall that a system is causal if for any n_0,$y[n_0]$depends only on $x[n], n \leq n_0$. We can now relate the notion of causality to the impulse response of a LSI system by using the convolution sum representation. Starting with the convolution sum representation of $y[n_0]$,

$$y[n_0] = \sum_{k=-\infty}^{\infty} x[k]h[n_0-k],$$

we see that $y[n_0]$ depends, in general on all values of $x[n]$ such that $h[n_0-k]$ is non-zero. For the corresponding LSI system to be causal, we would require that $h[n_0-k] = 0$ for $k > n_0$. This is equivalent to requiring that $h[n] = 0$,for $n < 0$. This leads to the following property of LSI systems:

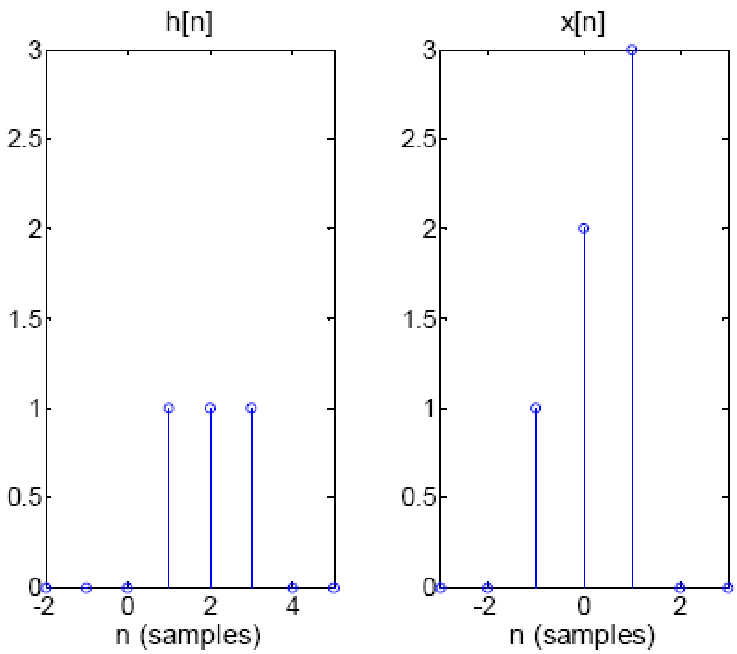
A discrete-time LSI system is causal, if and only if the impulse response satisfies $h[n] = 0$, for $n < 0$.

In this case, the convolution sum reduces to

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

for causal LSI systems.

Example: Graphical View of Convolution Given the two sequences $x[n]$ and $h[n]$ shown below,



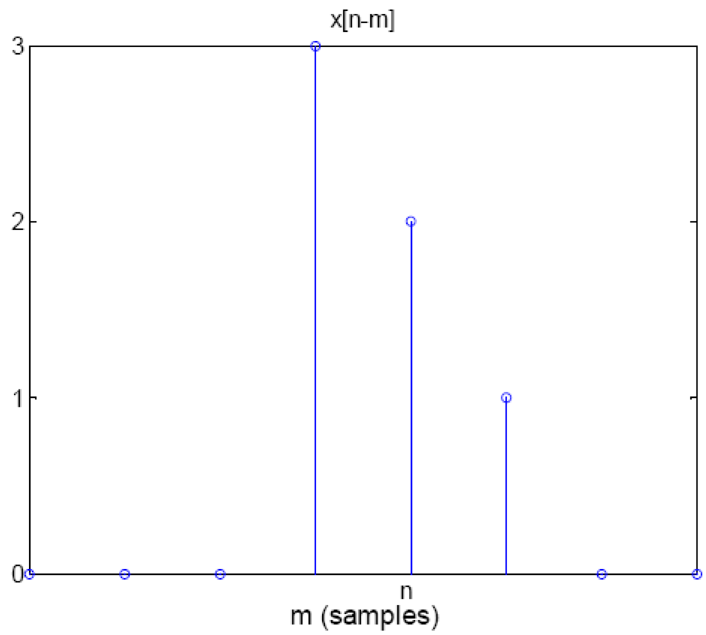
determine $x[n] * h[n]$, i.e. find

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

To do this, we will find the output one term at a time, by plotting $x[n-m]$ versus m, and then summing up the product of $h[m]$ and $x[n-m]$ over m. In order to plot $x[n-m]$ versus m, we want to view this as a sequence over m, where, here, n plays the role of a fixed shift of the sequence. So, to proceed, we first consider the sequence$x[-m]$, which is simply a time-reversed version of $x[m]$. We then desire $x[n-m] = x[-(m-n)]$, which is a shift of the sequence $x[-m]$ to the right by n samples (i.e. a delay of n samples of the time-reversed sequence $x[-m]$). In steps, this corresponds to

$$x[m] \longrightarrow \boxed{\begin{array}{c}\text{time}\\ \text{reversal}\end{array}} \xrightarrow{x[-m]} \boxed{\begin{array}{c}\text{delay by}\\ n \text{ samples}\end{array}} \xrightarrow{x[-(m-n)]} x[n-m]$$

which for this example yields the following result,

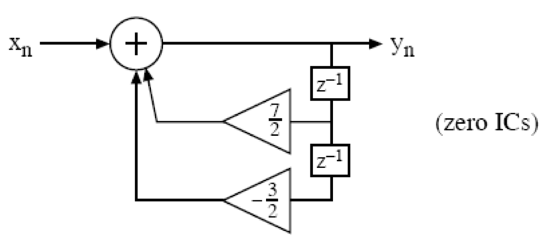


from the figure, we can see that, term by term, the output $y[n]$ can be computed as follows

$$y[n] = \begin{cases} 0, & n < -1, \\ 1 \times 1 = 1, & n = -1, \\ 2 \times 1 + 1 \times 1 = 3, & n = 0, \\ 3 \times 1 + 2 \times 1 + 1 \times 1 = 6, & n = 1, \\ 3 \times 1 + 2 \times 1 = 5, & n = 2, \\ 3 \times 1 = 3, & n = 3, \\ 0, & n > 3. \end{cases}$$

Noting from the figure that $h[n] = 0$ for $n < 0$, we expect that the output $y[n]$ will not depend on the input $x[k]$, for $k > n$, and we see that graphically, only past values of $x[n]$ contribute to the output.

Example Given the discrete-time sequence $x[n] = (1/4)^n u[n]$, determine the output of the causal discrete-time LSI system shown in the figure below.



The system in the figure is described by the following difference equation

$$y[n] = \frac{7}{2}y[n-1] - \frac{3}{2}y[n-2] + x[n]$$

or

$$y[n] - \frac{7}{2}y[n-1] + \frac{3}{2}y[n-2] = \left(\frac{1}{4}\right)^n, n \geq 0, y[-1] = y[-2] = 0.$$

We could solve this using a classical solution method for LCCDEs. However, this time, we will use the convolution formula, since this difference equation describes a LSI system, that can be characterized by its response to a discrete-time impulse, i.e, we can write

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

How do we obtain the impulse response from the given information? One way is to find $h[n]$ directly by solving an appropriate difference equation. As such, from definition of the system and that of the discrete-time impulse response, we have that $h[n]$ is defined to be the solution to the system equations when the input is a discrete-time impulse, i.e.,

$$h[n] - \frac{7}{2}h[n-1] + \frac{3}{2}h[n-2] = \delta[n], n \geq 0, h[-1] = h[-2] = 0.$$

How do we solve this difference equation? The input term has a form that changes with n. We could employ classical methods that require the selection of a particular solution, but we can sidestep this issue by noting that

$$h[n] - \frac{7}{2}h[n-1] + \frac{3}{2}h[n-2] = 0, n > 0.$$

However, now for initial conditions, we would require $h[0]$ and $h[-1]$. To find $h[0]$,we can simply use just use the system definition,

$$\begin{aligned} h[0] - \frac{7}{2}h[-1] + \frac{3}{2}h[-2] &= 1 \\ h[0] &= 1. \end{aligned}$$

Now to find $h[n]$for $n \geq 1$, we solve

$$h[n] - \frac{7}{2}h[n-1] + \frac{3}{2}h[n-2] = 0, n \geq 1, h[0] = 1, h[-1] = 0.$$

In order to solve this equation, we use the knowledge that a homogensous LCCDE, i.e. one for which the right hand side is zero, have solutions of the form $y[n] = cz^n$, for complex numbers z. Plugging in a general solution of this form, provides

$$\begin{aligned} z^n - \frac{7}{2}z^{n-1} + \frac{3}{2}z^{n-2} &= 0 \\ z^2 - \frac{7}{2}z + \frac{3}{2} &= 0 \end{aligned}$$

where the last line is known as the characteristic equation for the homogeneous LCCDE. The roots of the characteristic equation are given by

$$z = \frac{1}{2}, z = 3.$$

This implies that the impulse response takes the form

$$h[n] = c_1 \left(\frac{1}{2}\right)^n + c_2 3^n.$$

By applying the initial conditions, we can solve for the unknown constants $c_1$and c_2. This yields,

$$\begin{aligned} h[0] &= 1 = c_1 + c_2 \\ h[-1] &= 0 = 2c_1 + \frac{1}{3}c_2 \end{aligned}$$

which we can solve to obtain

$$c_1 = -\frac{1}{5}, \text{ and } c_2 = \frac{6}{5}.$$

This implies that $h[n]$ is given by

$$h[n] = -\frac{1}{5}\left(\frac{1}{2}\right)^n + \frac{6}{5}(3)^n, n \geq 0.$$

We know that $h[n] = 0$ for $n < 0$ from the initial conditions, so we have

$$h[n] = \left[-\frac{1}{5}\left(\frac{1}{2}\right)^n + \frac{6}{5}(3)^n\right] u[n].$$

Now, for the convolution sum, we have

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} h[k]x[n-k] \\ &= \sum_{k=-\infty}^{\infty} \left[-\frac{1}{5}\left(\frac{1}{2}\right)^k + \frac{6}{5}(3)^k\right] u[k] \left(\frac{1}{4}\right)^{n-k} u[n-k]. \end{aligned}$$

Since the term $u[k]$is inside the summation, we can replace the lower limit in the sum to obtain,

$$y[n] = \sum_{k=0}^{\infty} \left[-\frac{1}{5}\left(\frac{1}{2}\right)^k + \frac{6}{5}(3)^k\right] \left(\frac{1}{4}\right)^{n-k} u[n-k].$$

Note that the term $u[n-k]$ is zero for $k > n$, which leads to

$$y[n] = \sum_{k=0}^{n} \left[-\frac{1}{5}\left(\frac{1}{2}\right)^k + \frac{6}{5}(3)^k\right] u[k] \left(\frac{1}{4}\right)^{n-k}.$$

Now, we can also note that the summation index is k, and that the term $(1/4)^n$ inside the sum does not need to be there, so we can invite it out front, to leave only terms depending on k in side the summation, leading to

$$y[n] = \left(\frac{1}{4}\right)^n \sum_{k=0}^{n} \left[-\frac{1}{5}\left(\frac{1}{2}\right)^k + \frac{6}{5}(3)^k\right] \left(\frac{1}{4}\right)^{-k}.$$

We can now take each of the remaining finite-length geometric sums one at a time,

$$\begin{aligned} y[n] &= -\frac{1}{5}\left(\frac{1}{4}\right)^n \sum_{k=0}^{n} (2)^k + \frac{6}{5}\left(\frac{1}{4}\right)^n \sum_{k=0}^{n} (12)^k \\ &= -\frac{1}{5}\left(\frac{1}{4}\right)^n \frac{1-2^{n+1}}{1-2} + \frac{6}{5}\left(\frac{1}{4}\right)^n \frac{1-12^{n+1}}{1-12} \\ &= \frac{1}{5}\left(\frac{1}{4}\right)^n (1-2^{n+1}) + -\frac{6}{55}\left(\frac{1}{4}\right)^n (1-12^{n+1}) \\ &= \frac{1}{5}\left(\frac{1}{4}\right)^n - \frac{2}{5}\left(\frac{1}{2}\right)^n + -\frac{6}{55}\left(\frac{1}{4}\right)^n - \frac{72}{55}(3)^n \\ &= \frac{1}{11}\left(\frac{1}{4}\right)^n - \frac{2}{5}\left(\frac{1}{2}\right)^n + \frac{72}{55}(3)^n, n \geq 0. \end{aligned}$$

While it took quite a bit of work to find $h[n]$, once $h[n]$ was known, the output due to any input could be found via the convolution formula. This is one of the key properties of linear, shift invariant systems that make them attractive for analysis and implementation. We can always implement an LSI system by directly implementing the convolution sum. This may not be the most efficient implementation, for example, if the system is described by a low-order difference equation, but it will always work. We will later develop an easier way to find $h[n]$ using the z-transform.

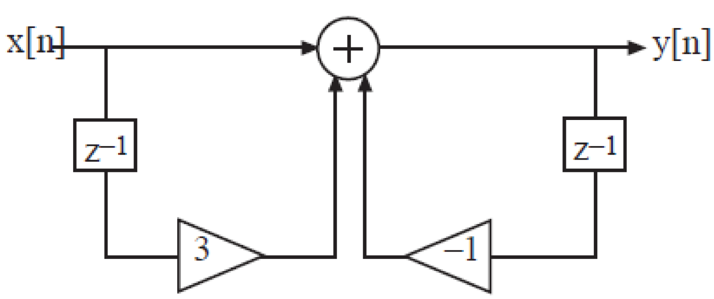


Figure 3.15: Example discrete-time linear, shift invariant (LSI) system. The elements labeld with z^{-1}correspond to delay elements.

3.8.1 Convolution and the unit pulse response (impulse response)

We continue by considering another example system, as shown in Figure 3.15.

The output of this system is given by the relation

$$y[n] = -y[n-1] + x[n] + 3x[n-1].$$

When the input to the system is a discrete-time impulse, $x[n] = \delta[n]$, when the system has zero initial conditions, we have

$$h[n] = -h[n-1] + \delta[n] + 3\delta[n-1], \qquad n \geq 0, h[-1] = 0.$$

We can use the initial rest condition, i.e. $h[-1] = 0$, to shift the equation two samples forward to obtain,

$$h[n] + h[n-1] = 0, \qquad n \geq 2.$$

From the initial condition, we have

$$h[0] = -h[-1] + \delta[0] + 3\delta[-1] = 1,$$

and we also have

$$h[1] = -h[0] + \delta[1] + 3\delta[0] = 2.$$

Then for $n \geq 2$, we have to solve a simpler, homogeneous difference equation of the form

$$h[n] + h[n-1] = 0, \qquad n \geq 2,\ h[1] = 2.$$

We saw previously, that the natural form of a solution to a linear constant-coefficient homogenous difference equation takes the form $cz^n$for (possibly complex) constants c and z. Substituting this form into the difference equation and factoring out the common terms, yields the characteristic equation,

$$z + 1 = 0,$$

which has the solution,

$$z = -1.$$

Therefore, the solution to the homogenous difference equation is

$$h[n] = c(-1)^n, \qquad n \geq 2,$$

where we can select the term c based on the condition that $h[1] = 2$, this yields

$$h[1] = c(-1) = 2 \implies c = -2.$$

Finally we have that

$$h[n] = -2(-1)^n, \qquad n \geq 1, \quad h[0] = 1.$$

We have that $h[n] = 0, n < 0$, since we assumed the system started with zero initial state prior to the input. The net result for the discrete-time impulse response is

$$h[n] = \begin{cases} 0, & n < 0, \\ 1, & n = 0, \\ -2(-1)^n & n \geq 1. \end{cases}$$

The approach to discrete-time convolution is similar to that of continuous-time convolution, with the exception that summations are used, rather than integration. We will illustrate the basic mechanics of discrete-time convolution through several examples.

Example Consider a system with impulse response $h[n] = (1/4)^n u[n]$ and input $x[n] = u[n-7]$. Let us determine the output of this system that would be obtained through the convolution sum

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

which we recall involves flipping the time axis of the input to produce $x[-k]$, then sliding this flipped version along to line up with a particular time sample in the impulse response $h[k]$, such that we have $x[n-k]$, and then taking the pointwise product $h[k]x[n-k]$, and summing the result, to provide the single output $y[n]$. If you are wondering why this process involves flipping the time axis of the input, let us consider the difference between how signals are plotted (for example, in a text like this one) and how the appear in time. When we plot a signal in a text, we typically place the first samples of the signal to the left and let the time axis increase to the right. However, with $x[n]$ plotted in this manner it appears as if we would have the largest values of the time axis (and hence the last values of the input) appearing first at the input to the system. Since for a signal that has its first non-zero value at $n = 0$, this is precisely why we need to flip the time axis of $x[n]$ as a first step in peforming convolution graphically.

Returning to the example, we have for the output

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

that we can interpret graphically, by plotting, for each value of n, the sequence $h[k]$ and the sequence $x[n-k]$. It is important to note that we seek a particular value of the output, $y[n]$, that is, we seek the output at time n. As a result, and to match the notation of the summation, we will graphically depict the sequences $h[k]$ and $x[n-k]$ as a function of the independent variable k. In this context, $x[n-k]$ corresponds to $x[-(k-n)]$, that is, the sequence $x[k]$ is flipped in time so that the sequence values are reversed with respect to the k axis, and then it is shifted so that the value of $x[0]$ sits on top of the time index $k = n$. Graphically, for the example above, we have the sequences as shown in Figure 3.16.

First we proceed with the convolution using the information provided by inspecting the signals graphically. We can observe from the plots of $h[k]$ and $x[n-k]$ that there will be no overlap of non-zero terms whenever the entire sequence $x[n-k]$ lies entirely to the left of $h[k]$. This will occur whenever we have $n - 7 < 0$ or equivalently, whenever $n < 7$. Hence, we have

$$y[n] = 0, \quad n < 7.$$

For $n \geq 7$, we have at least one non-zero term of overlap and we can readily compute the output as

$$\begin{aligned} y[n] &= \sum_{k=0}^{n-7} \left(\frac{1}{4}\right)^k = \frac{1 - (\frac{1}{4})^{n-7+1}}{1 - \frac{1}{4}} \\ &= \frac{4}{3} - \frac{4}{3}(\frac{1}{4})^{n-6}, \qquad , n \geq 7. \end{aligned}$$

This process can also be observed from a purely mathematical (algebraic) perspective. Beginning with the

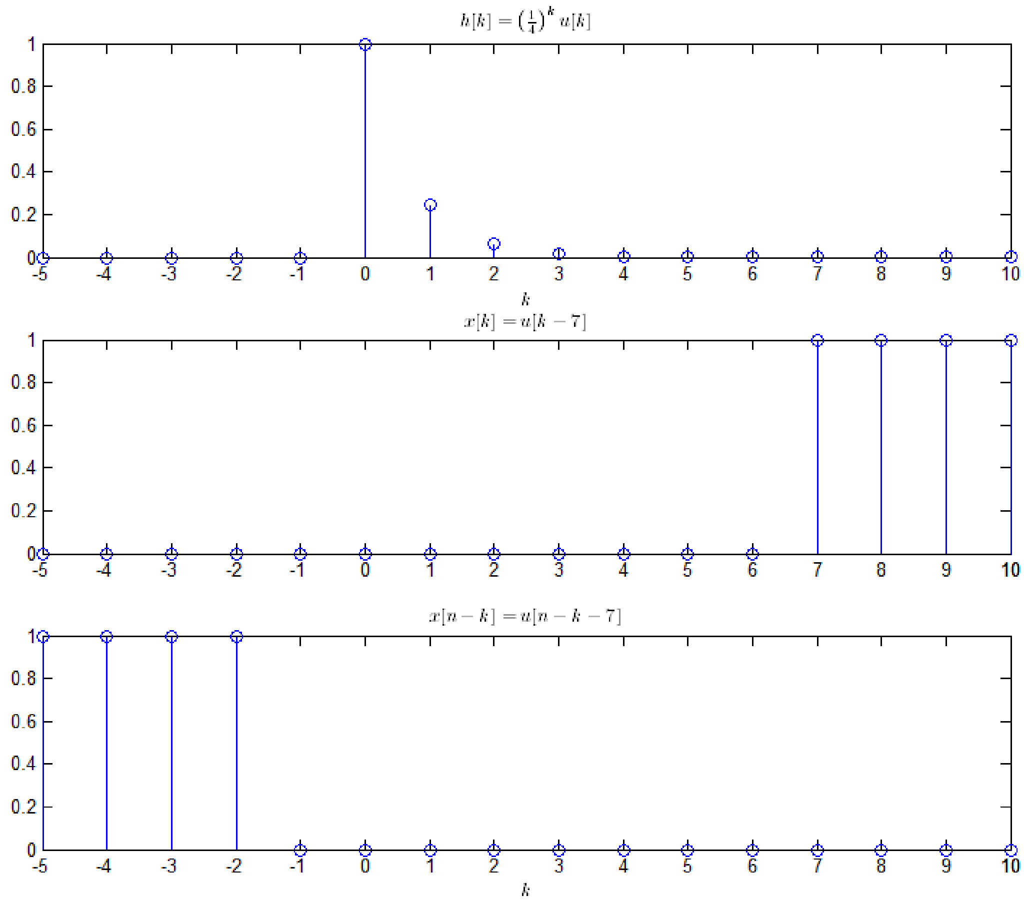


Figure 3.16: The sequences $h[k] = (1/4)^k u[k]$, $x[k] = u[k-7]$, and $x[n-k]$, for $n = -2$, as they participate in the convolution $y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$.

expression for the output from the convolution sum, we have

$$\begin{aligned}
y[n] &= \sum_{k=-\infty}^{\infty} h[k]x[n-k] \\
&= \sum_{k=-\infty}^{\infty} \left(\frac{1}{4}\right)^k u[k]u[(n-k)-7] \\
&= \sum_{k=0}^{\infty} \left(\frac{1}{4}\right)^k u[(n-k)-7] \\
&= \sum_{k=0}^{n-7} \left(\frac{1}{4}\right)^k \\
&= \begin{cases} 0, & n < 7 \\ \frac{4}{3} - \frac{4}{3}(\frac{1}{4})^{n-6}, & n \geq 7. \end{cases}
\end{aligned}$$

Taking a purely mathematical viewpoint, we began above with the definition of the output in terms of the convolution sum. The second line follows by expressing $h[k]$ and $x[n-k]$ in functional form. The third line follows from noting that $u[k] = 0$, for $k < 0$, and therefore the infinite sum will have no contribution for $k < 0$. The fourth line follows from a similar argument by noting that $u[n-k-7] = 0$ when $n-k-7 < 0$ which holds when $n-7 < k$; therefore, there will be no contribution to the infinite sum for values of $k > n-7$. Having established the limits to the summation first, we now have precisely the same form for the summation that we had in the graphical approach. Now we have one more issue to contend with. From the way that we arranged the limits of the summation, it is clear that whenever the term $u[k]u[n-k-7]$ is zero, there will be no contribution to the sum. Therefore, when the lower limit of the sum is greater than the upper limit of the sum, i.e. when $0 > n-7$ or $n < 7$, the output will be zero. This is the first term in the last line above, and the last term arises from summation of the finite-length geometric series. These simple steps will go a long way in either graphically, or algebraically computing convolutions with geometric terms containing unit step sequences.

Example We consider another, slightly more complicated example next. We consider the output $y[n]$ when the input is given by $x[n] = (\frac{1}{2})^n u[n]$ and the impulse response is give by $h[n] = (\frac{1}{4})^n u[n]$. Setting up the convolution sum, we have

$$y[n] = \sum_{k=-\infty}^{\infty} \left(\frac{1}{4}\right)^k u[k] \left(\frac{1}{2}\right)^{n-k} u[n-k].$$

This time we will proceed algebraically first, then graphically. Following the steps outlined in the previous example, we first use the unit step inside the infinite summation to determine the limits of the sum.

$$\begin{aligned}
y[n] &= \sum_{k=-\infty}^{\infty} \left(\frac{1}{4}\right)^k u[k] \left(\frac{1}{2}\right)^{n-k} u[n-k] \\
&= \sum_{k=0}^{n} \left(\frac{1}{4}\right)^k \left(\frac{1}{2}\right)^{n-k},
\end{aligned}$$

where we again used that $u[k] = 0$ for negative k and that $u[n-k]$ is zero for $k > n$. Combining terms, and

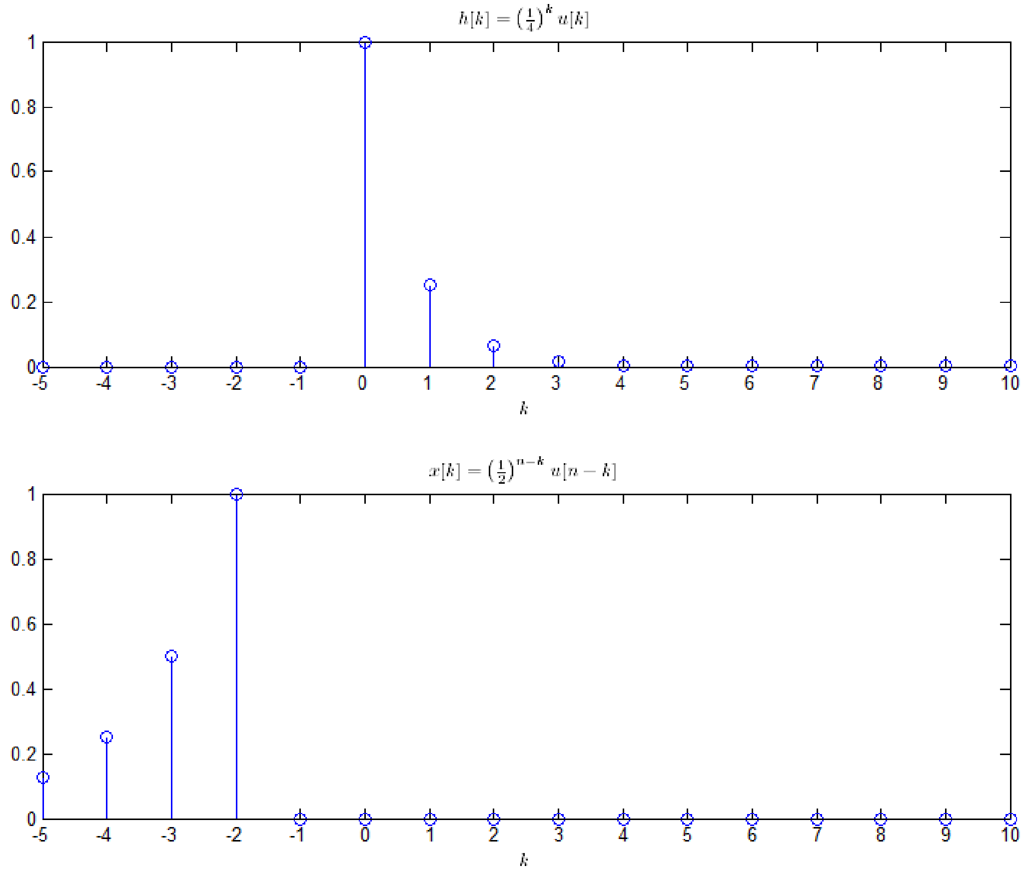


Figure 3.17: Sequences $h[k]$ and $x[n-k]$ shown for $n = -2$.

factoring out terms that do not depend on k, we obtain

$$\begin{aligned} y[n] &= \left(\frac{1}{2}\right)^n \sum_{k=0}^{n} \left(\frac{1}{4}\right)^k \left(\frac{1}{2}\right)^{-k} \\ &= \left(\frac{1}{2}\right)^n \sum_{k=0}^{n} \left(\frac{1}{2}\right)^k \\ &= \begin{cases} 0, & n < 0 \\ \left(\frac{1}{2}\right)^n \frac{1-\left(\frac{1}{2}\right)^{n+1}}{1-\frac{1}{2}}, & n > 0 \end{cases} \\ &= \begin{cases} 0, & n < 0 \\ \left(\frac{1}{2}\right)^{n-1} - \left(\frac{1}{4}\right)^n, & n > 0. \end{cases} \end{aligned}$$

This algebraic appraoch proceeded simply using the rules outlined above for determining limits of the convolution sum using the non-zero overlap regions of the unit step functions. These can also be determined graphically, by observing the sequence $h[k]$ and $x[n-k]$ graphically on similar axes as shown in Figure 3.17. It is clear from the figure, that for $n < 0$, there will be no overlap in the two sequences, while for $n \geq 0$, thee result will be a finite-length sum (growing with n) that arises from the product of two geometric terms. The result is the algebraic sum we have just solved.

Example We now consider a finite-length input to an LSI system with an infinite-length impulse response.

We want to determine the output $y[n]$ when the input is $x[n] = u[n] - u[n-N]$, for some $N > 0$, and the impulse response is given by $h[n] = a^n u[n]$. We will solve for the result in terms of N and a in general form. We begin again with the convolution sum,

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} h[k]x[n-k] \\ &= \sum_{k=-\infty}^{\infty} a^k u[k](u[n-k] - u[n-k-N]) \\ &= \sum_{k=0}^{\infty} a^k (u[n-k] - u[n-k-N]). \end{aligned}$$

At this point, there are a few ways to proceed. We could either break this sum into two sums, which would provide

$$\begin{aligned} y[n] &= \sum_{k=0}^{\infty} a^k u[n-k] - \sum_{k=0}^{\infty} a^k u[n-k-N] \\ &= \sum_{k=0}^{n} a^k u[n-k] - \sum_{k=0}^{n-N} a^k u[n-k-N] \\ &= \frac{1-a^{n+1}}{1-a} u[n] - \frac{1-a^{n-N+1}}{1-a} u[n-N], \end{aligned}$$

or we could have recognized that there will be three regions of interest:$n < 0, \;\; 0 \le n < N$, and $n \ge N$. This would have led us to evaluate on the the first term in the sum above for values of $0 \le n < N$, and then introduced the second sum for $n \ge N$ which would enable use to combine the two algebraic expressions in each of these two regions. Sometimes this is a simpler approach, sometimes the purely algebraic approach is simpler. The results will always be the same in the end, if the calculations are carried out carefully.

Example We now consider a more complicated example with a two-sided convolution. Let the input signal $x[n]$ be given by $x[n] = (\frac{1}{3})^n u[n] + 4^n u[-n-1]$, which is a two-sided sequence, and let the impulse response be $h[n] = u[n-1]$. We will tackle this example using the mathematical appraoch described above. First, we write the output $y[n]$ in terms if of the two signals, i.e.

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} \left[\left(\frac{1}{3}\right)^k u[k] + 4^k u[-k-1] \right] u[n-k-1] \\ &= \sum_{k=-\infty}^{\infty} \left[\left(\frac{1}{3}\right)^k u[k] + 4^k u[-k-1] \right] u[n-k-1] \\ &= \sum_{k=-\infty}^{n-1} \left(\frac{1}{3}\right)^k u[k] + \sum_{k=-\infty}^{\infty} 4^k u[-k-1]u[n-k-1] \\ &= \sum_{k=0}^{n-1} \left(\frac{1}{3}\right)^k + \sum_{k=-\infty}^{\infty} 4^k u[-k-1]u[n-k-1] \\ &= \frac{1-(\frac{1}{3})^n}{1-\frac{1}{3}} u[n-1] + \sum_{m=-\infty}^{\infty} 4^{-m} u[m-1]u[m+n-1] \end{aligned}$$

where we have taken care of the first term through methods just like the previous example. We used the term $u[k]$ in the summation to eliminate values of k from the summation for $k < 0$ and we used the term $u[n-k-1]$ to eliminate terms from the summation for values of $k > n-1$. In the second summation above, in the last line, we have made the substitution $m = -k$ to help us handle the product $u[m-1]u[m+n-1]$, which will take some care to work through. Since each of these terms is of the form $u[n-N]$ for some N, then each is a right-sided unit step sequence, i.e., each is zero for $m < N$, for some N and then one for all $m \ge N$.

Therefore, when you multiply them you obtain $u[m-N_1]u[m-N_2] = u[m-\max(N_1,N_2)]$. We next note that $u[m-1]$ is equal to zero for values of $m<1$, i.e. $N_1 = 1$. Next, we note that $u[m+n-1] = u[m-(1-n)]$ is equal to zero for values of $m < 1-n$, i.e. $N_2 = 1-n$. So, since the summation will start at either $m = 0$ or at $m = 1-n$, we need to determine which of these terms wins out, i.e. we need to find $N = \max(1, 1-n)$. We see that when $n \geq 1$, we have $u[m-1]u[m+n-1] = u[m-1]$ (the sum starts at $m = 1$) and when $n < 1$, we have that $u[m-1]u[m+n-1] = u[m+n-1]$, i.e. the sum starts at $m = 1-n$. We can now proceed, setting

$$
\begin{aligned}
&= y[n] = \frac{1-(\frac{1}{3})^n}{\frac{2}{3}}u[n-1] + \sum_{m=-\infty}^{\max(1,1-n)} 4^{-m}u[m-1]u[m+n-1] \\
&= \frac{3-(\frac{1}{3})^{n-1}}{2}u[n-1] + \begin{cases} \sum_{m=1}^{\infty} 4^{-m}, & n \geq 1 \\ \sum_{m=1-n}^{\infty} 4^{-m} & n < 1 \end{cases} \\
&= \frac{3-(\frac{1}{3})^{n-1}}{2}u[n-1] + \begin{cases} \frac{\frac{1}{4}}{1-\frac{1}{4}}, & n \geq 1 \\ \frac{4^{-(1-n)}}{1-\frac{1}{4}} & n < 1 \end{cases} \\
&= \frac{3-(\frac{1}{3})^{n-1}}{2}u[n-1] + \begin{cases} \frac{1}{3}, & n \geq 1 \\ \frac{4^n}{3} & n < 1 \end{cases} \\
&= \left(\frac{3}{2} + \frac{1}{3} - \frac{1}{2}\left(\frac{1}{3}\right)^{n-1}\right)u[n-1] + \frac{1}{3}4^n u[1-n] \\
&= \left(\frac{11}{6} - \frac{1}{2}\left(\frac{1}{3}\right)^{n-1}\right)u[n-1] + \frac{1}{3}4^n u[1-n].
\end{aligned}
$$

3.9 Difference equations

Recall from (3.4) that a linear constant-coefficient difference equation (LCCDE) is given by the following input-output relation

$$y[n] + a_1 y[n-1] + \ldots a_N y[n-N] = b_0 x[n] + \ldots b_M x[n-M],$$

which can be more compactly written

$$y[n] + \sum_{k=1}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k].$$

This is an input-output relationship between the input $x[n]$ and the output $y[n]$ that is both linear and shift-invariant. We note that the difference equation alone does not uniquely characterize the output $y[n]$ for a given $x[n]$, since the LCCDE depends on values of the output that occur either prior to the input, or after the input has terminated. If we consider a causal system described by an LCCDE, then we know that the output $y[n]$ for $n \geq m$ cannot depend on values of the input $x[n]$ for $n > m$. Therefore, if the input $x[n]$ is zero for $n < 0$, in order to determine the output $y[n]$, we need some auxilliary conditions, i.e. initial conditions $y[-k]$, $k = 1, ..., N$. Since the system is LSI, we know that we can always decompose the output into the sum of two components, $y[n] = y_x[n] + y_s[n]$, i.e. that contribution to the outout from the input when the initial conditions are set to zero, $y_x[n]$, and the contribution to the output due only to the initial conditions (or the initial state of the system), when the input is set to zero, $y_s[n]$. We will see later in this text how to readily handle such situations using the one-sided (unilateral) z-transform. We note that it is also possible to directly find the solution for the output of an LCCDE given the input and initial conditions by solving the homogenous equation, as was done in a previous example, where we obtained the characteristic equation by setting the input to zero, and then adding in a particular solution that is matched to the form of the input. Then, since the homogenous equation leaves the right hand side of the LCCDE equal to zero, we can then use undetermined constants from the homogeneous solution to match the initial conditions.

When the initial conditions are set to zero, another method for finding the output of an LCCDE for a given input would be to solve for the homogeneous solution using the characteristic equation as before, but then use the implied initial conditions that result from application of a discrete-time impulse as input. We can then use the undetermined coefficients from the homogenous solution to find the response of the system to a discrete-time impulse, or, the impulse response of the system. Then, for any given input, we can find the output using convolution methods, as described earlier in this chapter.