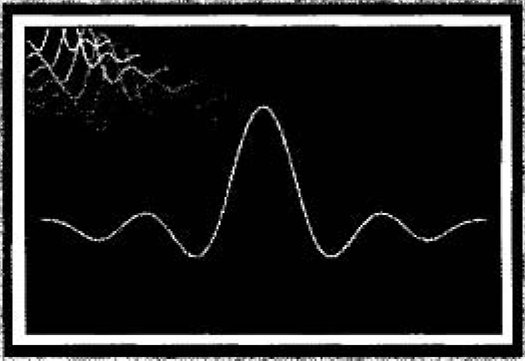
4 Sampling of Continuous-Time Signals



4.0 INTRODUCTION

Discrete-time signals can arise in many ways, but they occur most commonly as representations of sampled continuous-time signals. While sampling will no doubt be familiar to many readers, we shall review many of the basic issues such as the phenomenon of aliasing and the important fact that continuous-time signal processing can be implemented through a process of sampling, discrete-time processing, and reconstruction of a continuous-time signal. After a thorough discussion of these basic issues, we discuss multirate signal processing, A/D conversion, and the use of oversampling in A/D conversion.

4.1 PERIODIC SAMPLING

Discrete representations of signals can take many forms including basis expansions of various types, parametric models for signal modeling (Chapter 11), and nonuniform sampling (see for example Yen (1956), Yao and Thomas (1967) and Eldar and Oppenheim (2000)). Such representations are often based on prior knowledge of properties of the signal that can be exploited to obtain more efficient representations. However, even these alternative representations generally begin with a discrete-time representation of a continuous-time signal obtained through periodic sampling; i.e., a sequence of samples, $x[n]$, is obtained from a continuous-time signal $x_c(t)$ according to the relation

$$x[n] = x_c(nT), \qquad -\infty < n < \infty. \tag{4.1}$$

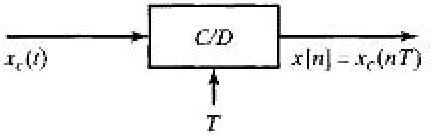


Figure 4.1 Block diagram representation of an ideal continuous-to-discrete-time (C/D) converter.

In Eq. (4.1), T is the *sampling period*, and its reciprocal, $f_s = 1/T$, is the *sampling frequency*, in samples per second. We also express the sampling frequency as $\Omega_s = 2\pi/T$ when we want to use frequencies in radians per second. Since sampling representations rely only on the assumption of a bandlimited Fourier transform, they are applicable to a wide class of signals that arise in many practical applications.

We refer to a system that implements the operation of Eq. (4.1) as an *ideal continuous-to-discrete-time (C/D) converter*, and we depict it in block diagram form as indicated in Figure 4.1. As an example of the relationship between $x_c(t)$ and $x[n]$, in Figure 2.2 we illustrated a continuous-time speech waveform and the corresponding sequence of samples.

In a practical setting, the operation of sampling is implemented by an analog-to-digital (A/D) converter. Such systems can be viewed as approximations to the ideal C/D converter. In addition to sampling rate, which is sufficient to define the ideal C/D converter, important considerations in the implementation or choice of an A/D converter include quantization of the output samples, linearity of quantization steps, the need for sample-and-hold circuits, and limitations on the sampling rate. The effects of quantization are discussed in Sections 4.8.2 and 4.8.3. Other practical issues of A/D conversion are electronic circuit concerns that are outside the scope of this text.

The sampling operation is generally not invertible; i.e., given the output $x[n]$, it is not possible in general to reconstruct $x_c(t)$, the input to the sampler, since many continuous-time signals can produce the same output sequence of samples. The inherent ambiguity in sampling is a fundamental issue in signal processing. However, it is possible to remove the ambiguity by restricting the frequency content of input signals that go into the sampler.

It is convenient to represent the sampling process mathematically in the two stages depicted in Figure 4.2(a). The stages consist of an impulse train modulator, followed by conversion of the impulse train to a sequence. The periodic impulse train is

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT), \tag{4.2}$$

where $\delta(t)$ is the unit impulse function, or Dirac delta function. The product of $s(t)$ and $x_c(t)$ is therefore

$$\begin{aligned} x_s(t) &= x_c(t)s(t) \\ &= x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) = \sum_{n=-\infty}^{\infty} x_c(t)\delta(t - nT). \end{aligned} \tag{4.3}$$

Using the property of the continuous-time impulse function, $x(t)\delta(t) = x(0)\delta(t)$, sometimes called the "sifting property" of the impulse function, (see e.g., Oppenheim and

Willsky, 1997), $x_s(t)$ can be expressed as

$$x_s(t) = \sum_{n=-\infty}^{\infty} x_c(nT)\delta(t - nT), \tag{4.4}$$

i.e., the size (area) of the impulse at sample time nT is equal to the value of the continuous-time signal at that time. In this sense, the impulse train modulation of Eq. (4.3) is a mathematical representation of sampling.

Figure 4.2(b) shows a continuous-time signal $x_c(t)$ and the results of impulse train sampling for two different sampling rates. Note that the impulses $x_c(nT)\delta(t - nT)$ are represented by arrows with length proportional to their area. Figure 4.2(c) depicts the corresponding output sequences. The essential difference between $x_s(t)$ and $x[n]$ is that $x_s(t)$ is, in a sense, a continuous-time signal (specifically, an impulse train) that is zero,

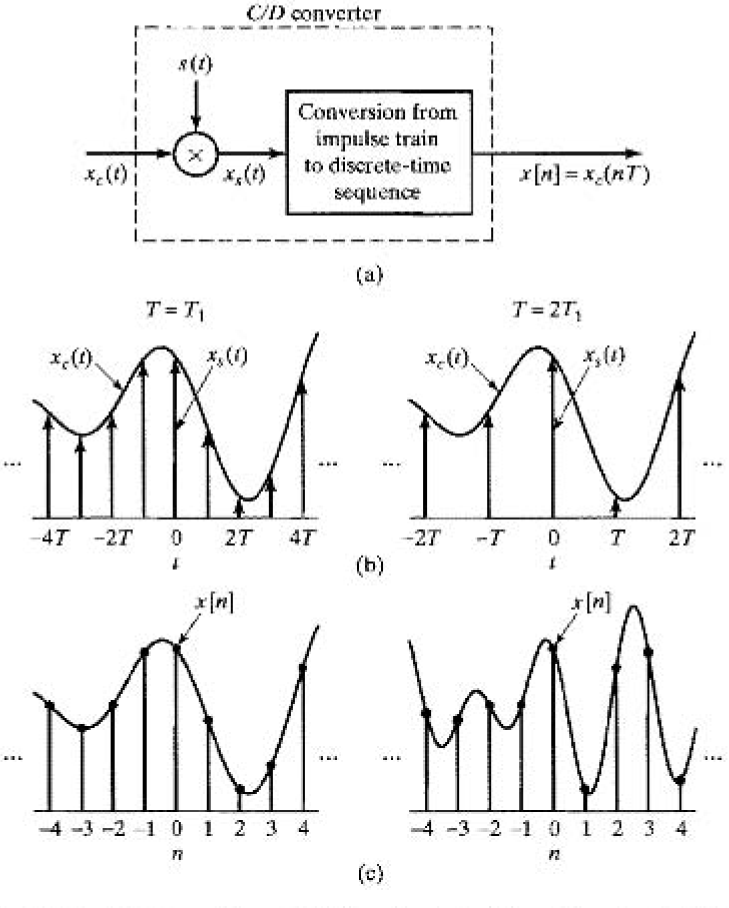


Figure 4.2 Sampling with a periodic impulse train, followed by conversion to a discrete-time sequence. (a) Overall system. (b) $x_s(t)$ for two sampling rates. (c) The output sequence for the two different sampling rates.

except at integer multiples of T. The sequence $x[n]$, on the other hand, is indexed on the integer variable n, which, in effect, introduces a time normalization; i.e., the sequence of numbers $x[n]$ contains no explicit information about the sampling period T. Furthermore, the samples of $x_c(t)$ are represented by finite numbers in $x[n]$ rather than as the areas of impulses, as with $x_s(t)$.

It is important to emphasize that Figure 4.2(a) is strictly a mathematical representation convenient for gaining insight into sampling in both the time domain and frequency domain. It is not a close representation of any physical circuits or systems designed to implement the sampling operation. Whether a piece of hardware can be construed to be an approximation to the block diagram of Figure 4.2(a) is a secondary issue at this point. We have introduced this representation of the sampling operation because it leads to a simple derivation of a key result and because the approach leads to a number of important insights that are difficult to obtain from a more formal derivation based on manipulation of Fourier transform formulas.

4.2 FREQUENCY-DOMAIN REPRESENTATION OF SAMPLING

To derive the frequency-domain relation between the input and output of an ideal C/D converter, consider the Fourier transform of $x_s(t)$. Since, from Eq. (4.3), $x_s(t)$ is the product of $x_c(t)$ and $s(t)$, the Fourier transform of $x_s(t)$ is the convolution of the Fourier transforms $X_c(j\Omega)$ and $S(j\Omega)$ scaled by $\frac{1}{2\pi}$. The Fourier transform of the periodic impulse train $s(t)$ is the periodic impulse train

$$S(j\Omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s), \tag{4.5}$$

where $\Omega_s = 2\pi/T$ is the sampling frequency in radians/s (see Oppenheim and Willsky, 1997 or McClellan, Schafer and Yoder, 2003). Since

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega),$$

where $*$ denotes the operation of continuous-variable convolution, it follows that

$$X_s(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s)). \tag{4.6}$$

Equation (4.6) is the desired relationship between the Fourier transforms of the input and the output of the impulse train modulator in Figure 4.2(a). Equation (4.6) states that the Fourier transform of $x_s(t)$ consists of periodically repeated copies of $X_c(j\Omega)$, the Fourier transform of $x_c(t)$. These copies are shifted by integer multiples of the sampling frequency, and then superimposed to produce the periodic Fourier transform of the impulse train of samples. Figure 4.3 depicts the frequency-domain representation of impulse train sampling. Figure 4.3(a) represents a bandlimited Fourier transform having the property that $X_c(j\Omega) = 0$ for $|\Omega| \geq \Omega_N$. Figure 4.3(b) shows the periodic impulse train $S(j\Omega)$, and Figure 4.3(c) shows $X_s(j\Omega)$, the result of convolving $X_c(j\Omega)$ with $S(j\Omega)$ and scaling by $\frac{1}{2\pi}$. It is evident that when

$$\Omega_s - \Omega_N \geq \Omega_N, \quad \text{or} \quad \Omega_s \geq 2\Omega_N, \tag{4.7}$$

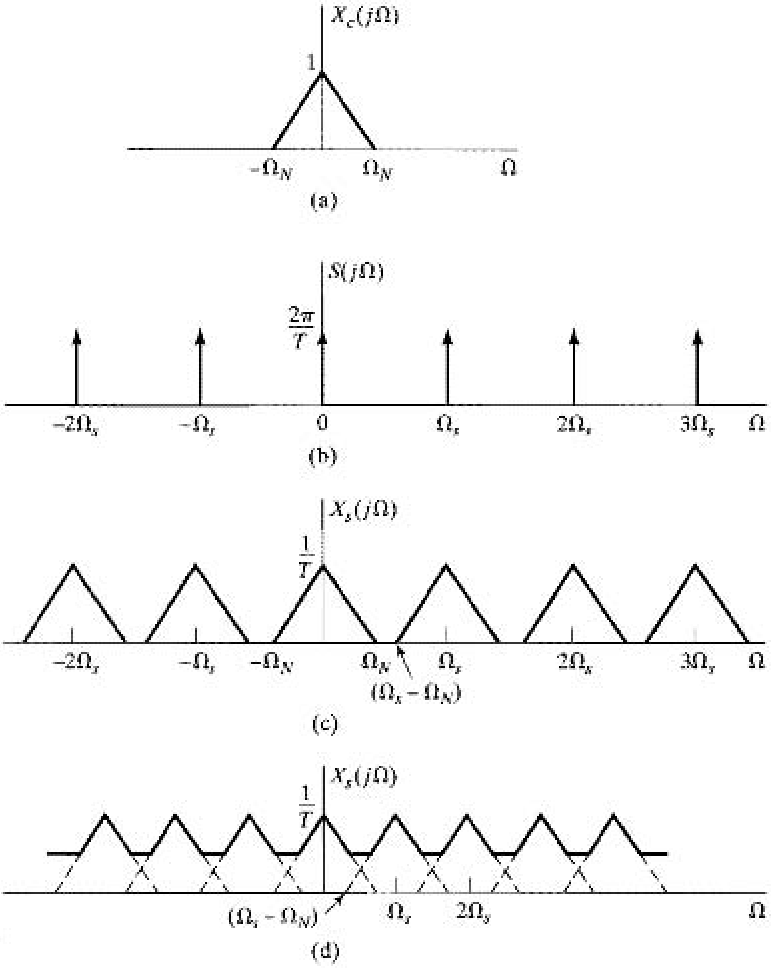


Figure 4.3 Frequency-domain representation of sampling in the time domain. (a) Spectrum of the original signal. (b) Fourier transform of the sampling function. (c) Fourier transform of the sampled signal with $\Omega_s > 2\Omega_N$. (d) Fourier transform of the sampled signal with $\Omega_s < 2\Omega_N$.

as in Figure 4.3(c), the replicas of $X_c(j\Omega)$ do not overlap, and therefore, when they are added together in Eq. (4.6), there remains (to within a scale factor of $1/T$) a replica of $X_c(j\Omega)$ at each integer multiple of Ω_s. Consequently, $x_c(t)$ can be recovered from $x_s(t)$ with an ideal lowpass filter. This is depicted in Figure 4.4(a), which shows the impulse train modulator followed by an LTI system with frequency response $H_r(j\Omega)$. For $X_c(j\Omega)$ as in Figure 4.4(b), $X_s(j\Omega)$ would be as shown in Figure 4.4(c), where it is assumed that $\Omega_s > 2\Omega_N$. Since

$$X_r(j\Omega) = H_r(j\Omega)X_s(j\Omega), \tag{4.8}$$

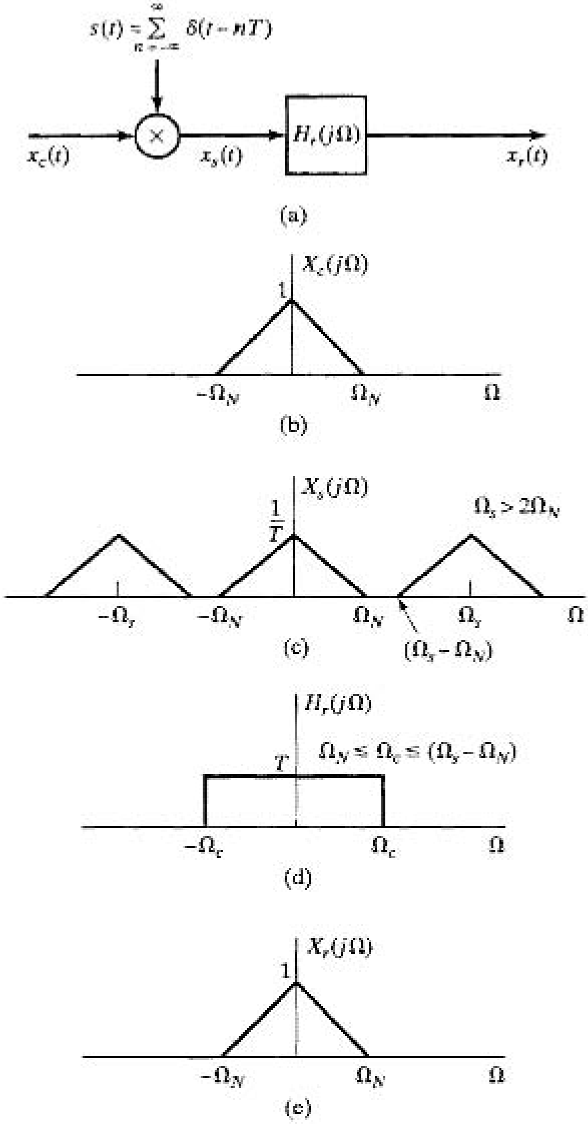


Figure 4.4 Exact recovery of a continuous-time signal from its samples using an ideal lowpass filter.

it follows that if $H_r(j\Omega)$ is an ideal lowpass filter with gain T and cutoff frequency Ω_c such that

$$\Omega_N \le \Omega_c \le (\Omega_s - \Omega_N), \tag{4.9}$$

then

$$X_r(j\Omega) = X_c(j\Omega), \tag{4.10}$$

as depicted in Figure 4.4(e) and therefore $x_r(t) = x_c(t)$.

If the inequality of Eq. (4.7) does not hold, i.e., if $\Omega_s < 2\Omega_N$, the copies of $X_c(j\Omega)$ overlap, so that when they are added together, $X_c(j\Omega)$ is no longer recoverable by lowpass filtering. This is illustrated in Figure 4.3(d). In this case, the reconstructed output $x_r(t)$ in Figure 4.4(a) is related to the original continuous-time input through a distortion referred to as *aliasing distortion*, or, more simply, *aliasing*. Figure 4.5 illustrates aliasing in the frequency domain for the simple case of a cosine signal of the form

$$x_c(t) = \cos \Omega_0 t, \tag{4.11a}$$

whose Fourier transform is

$$X_c(j\Omega) = \pi\delta(\Omega - \Omega_0) + \pi\delta(\Omega + \Omega_0) \tag{4.11b}$$

as depicted in Figure 4.5(a). Note that the impulse at $-\Omega_0$ is dashed. It will be helpful to observe its effect in subsequent plots. Figure 4.5(b) shows the Fourier transform of $x_s(t)$ with $\Omega_0 < \Omega_s/2$, and Figure 4.5(c) shows the Fourier transform of $x_s(t)$ with $\frac{\Omega_s}{2} < \Omega_0 < \Omega_s$. Figures 4.5(d) and (e) correspond to the Fourier transform of the

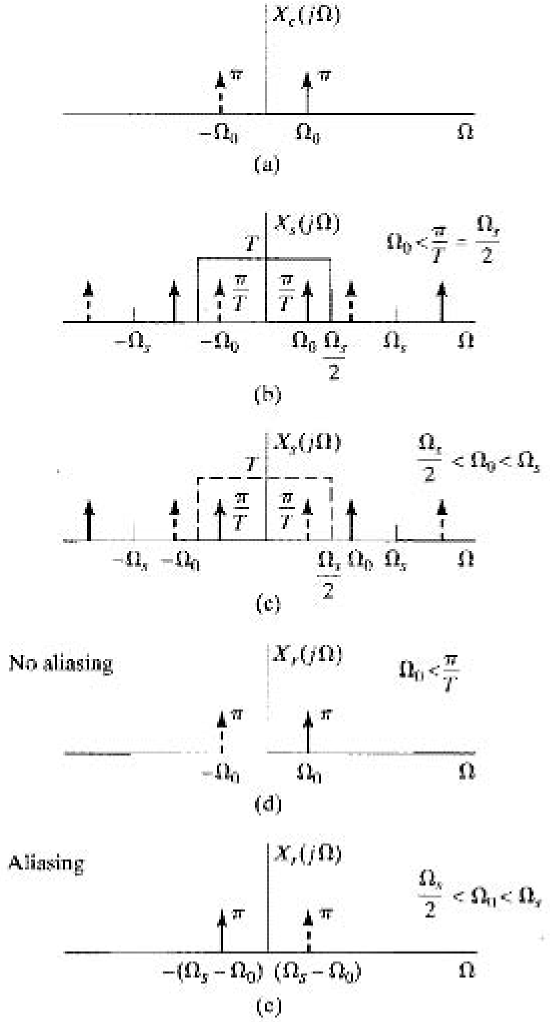


Figure 4.5 The effect of aliasing in the sampling of a cosine signal.

lowpass filter output for $\Omega_0 < \Omega_s/2 = \pi/T$ and $\Omega_s/2 < \Omega_0 < \Omega_s$, respectively, with $\Omega_c = \Omega_s/2$. Figures 4.5(c) and (e) correspond to the case of aliasing. With no aliasing [Figures 4.5(b) and (d)], the reconstructed output is

$$x_r(t) = \cos\Omega_0 t. \tag{4.12}$$

With aliasing, the reconstructed output is

$$x_r(t) = \cos(\Omega_s - \Omega_0)t; \tag{4.13}$$

i.e., the higher frequency signal $\cos\Omega_0 t$ has taken on the identity (alias) of the lower frequency signal $\cos(\Omega_s - \Omega_0)t$ as a consequence of the sampling and reconstruction. This discussion is the basis for the Nyquist sampling theorem (Nyquist 1928; Shannon, 1949), stated as follows.

Nyquist-Shannon Sampling Theorem: Let $x_c(t)$ be a bandlimited signal with

$$X_c(j\Omega) = 0 \qquad \text{for } |\Omega| \geq \Omega_N. \tag{4.14a}$$

Then $x_c(t)$ is uniquely determined by its samples $x[n] = x_c(nT)$, $n = 0, \pm1, \pm2, \ldots$, if

$$\Omega_s = \frac{2\pi}{T} \geq 2\Omega_N. \tag{4.14b}$$

The frequency Ω_N is commonly referred to as the *Nyquist frequency*, and the frequency $2\Omega_N$ as the *Nyquist rate*.

Thus far, we have considered only the impulse train modulator in Figure 4.2(a). Our eventual objective is to express $X(e^{j\omega})$, the discrete-time Fourier transform (DTFT) of the sequence $x[n]$, in terms of $X_s(j\Omega)$ and $X_c(j\Omega)$. Toward this end, let us consider an alternative expression for $X_s(j\Omega)$. Applying the continuous-time Fourier transform to Eq. (4.4), we obtain

$$X_s(j\Omega) = \sum_{n=-\infty}^{\infty} x_c(nT)e^{-j\Omega Tn}. \tag{4.15}$$

Since

$$x[n] = x_c(nT) \tag{4.16}$$

and

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}, \tag{4.17}$$

it follows that

$$X_s(j\Omega) = X(e^{j\omega})|_{\omega=\Omega T} = X(e^{j\Omega T}). \tag{4.18}$$

Consequently, from Eqs. (4.6) and (4.18),

$$X(e^{j\Omega T}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s)), \tag{4.19}$$

or equivalently,

$$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left[j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right]. \tag{4.20}$$

From Eqs. (4.18)–(4.20), we see that $X(e^{j\omega})$ is a frequency-scaled version of $X_s(j\Omega)$ with the frequency scaling specified by $\omega = \Omega T$. This scaling can alternatively be thought of as a normalization of the frequency axis so that the frequency $\Omega = \Omega_s$ in $X_s(j\Omega)$ is normalized to $\omega = 2\pi$ for $X(e^{j\omega})$. The frequency scaling or normalization in the transformation from $X_s(j\Omega)$ to $X(e^{j\omega})$ is directly a result of the time normalization in the transformation from $x_s(t)$ to $x[n]$. Specifically, as we see in Figure 4.2, $x_s(t)$ retains a spacing between samples equal to the sampling period T. In contrast, the "spacing" of sequence values $x[n]$ is always unity; i.e., the time axis is normalized by a factor of T. Correspondingly, in the frequency domain the frequency axis is normalized by $f_s = 1/T$.

For a sinusoid of the form $x_c(t) = \cos(\Omega_0 t)$, the highest (and only) frequency is Ω_0. Since the signal is described by a simple equation, it is easy to compute the samples of the signal. The next two examples use sinusoidal signals to illustrate some important points about sampling.

Example 4.1 Sampling and Reconstruction of a Sinusoidal Signal

If we sample the continuous-time signal $x_c(t) = \cos(4000\pi t)$ with sampling period $T = 1/6000$, we obtain $x[n] = x_c(nT) = \cos(4000\pi Tn) = \cos(\omega_0 n)$, where $\omega_0 = 4000\pi T = 2\pi/3$. In this case, $\Omega_s = 2\pi/T = 12000\pi$, and the highest frequency of the signal is $\Omega_0 = 4000\pi$, so the conditions of the Nyquist sampling theorem are satisfied and there is no aliasing. The Fourier transform of $x_c(t)$ is

$$X_c(j\Omega) = \pi\delta(\Omega - 4000\pi) + \pi\delta(\Omega + 4000\pi).$$

Figure 4.6(a) shows

$$X_s(j\Omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c[j(\Omega - k\Omega_s)] \tag{4.21}$$

for $\Omega_s = 12000\pi$. Note that $X_c(j\Omega)$ is a pair of impulses at $\Omega = \pm 4000\pi$, and we see shifted copies of this Fourier transform centered on $\pm\Omega_s, \pm 2\Omega_s$, etc. Plotting $X(e^{j\omega}) = X_s(j\omega/T)$ as a function of the normalized frequency $\omega = \Omega T$ results in Figure 4.6(b), where we have used the fact that scaling the independent variable of an impulse also scales its area, i.e., $\delta(\omega/T) = T\delta(\omega)$ (Oppenheim and Willsky, 1997). Note that the original frequency $\Omega_0 = 4000\pi$ corresponds to the normalized frequency $\omega_0 = 4000\pi T = 2\pi/3$, which satisfies the inequality $\omega_0 < \pi$, corresponding to the fact that $\Omega_0 = 4000\pi < \pi/T = 6000\pi$. Figure 4.6(a) also shows the frequency response of an ideal reconstruction filter $H_r(j\Omega)$ for the given sampling rate of $\Omega_s = 12000\pi$. This figure shows that the reconstructed signal would have frequency $\Omega_0 = 4000\pi$, which is the frequency of the original signal $x_c(t)$.

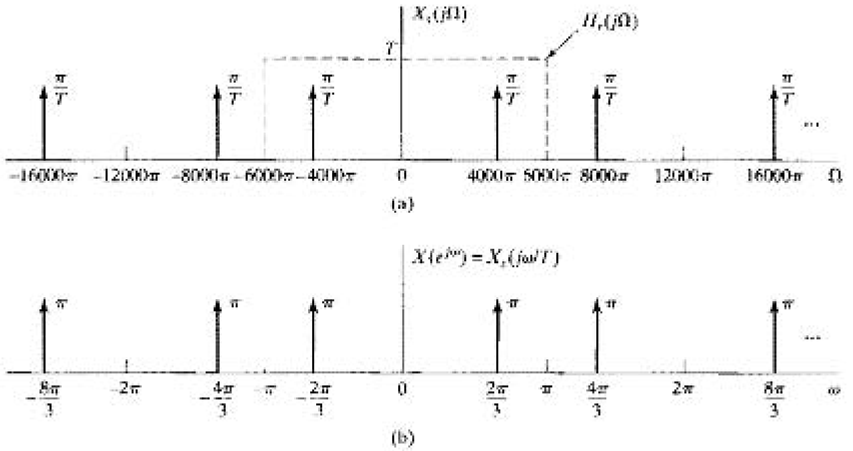


Figure 4.6 (a) Continuous-time and (b) discrete-time Fourier transforms for sampled cosine signal with frequency $\Omega_0 = 4000\pi$ and sampling period $T = 1/6000$.

Example 4.2 Aliasing in Sampling a Sinusoidal Signal

Now suppose that the continuous-time signal is $x_c(t) = \cos(16000\pi t)$, but the sampling period is $T = 1/6000$, as it was in Example 4.1. This sampling period fails to satisfy the Nyquist criterion, since $\Omega_s = 2\pi/T = 12000\pi < 2\Omega_0 = 32000\pi$. Consequently, we expect to see aliasing. The Fourier transform $X_s(j\Omega)$ for this case is identical to that of Figure 4.6(a). However, now the impulse located at $\Omega = -4000\pi$ is from $X_c[j(\Omega - \Omega_s)]$ in Eq. (4.21) rather than from $X_c(j\Omega,)$ and the impulse at $\Omega = 4000\pi$ is from $X_c[j(\Omega + \Omega_s)]$. That is, the frequencies $\pm 4000\pi$ are alias frequencies. Plotting $X(e^{j\omega}) = X_s(j\omega/T)$ as a function of ω yields the same graph as shown in Figure 4.6(b), since we are normalizing by the same sampling period. The fundamental reason for this is that the sequence of samples is the same in both cases; i.e.,

$$\cos(16000\pi n/6000) = \cos(2\pi n + 4000\pi n/6000) = \cos(2\pi n/3).$$

(Recall that we can add any integer multiple of 2π to the argument of the cosine without changing its value.) Thus, we have obtained the same sequence of samples, $x[n] = \cos(2\pi n/3)$, by sampling two different continuous-time signals with the same sampling frequency. In one case, the sampling frequency satisfied the Nyquist criterion, and in the other case it did not. As before, Figure 4.6(a) shows the frequency response of an ideal reconstruction filter $H_r(j\Omega)$ for the given sampling rate of $\Omega_s = 12000\pi$. It is clear from this figure that the signal that would be reconstructed would have the frequency $\Omega_0 = 4000\pi$, which is the alias frequency of the original frequency 16000π with respect to the sampling frequency $\Omega_s = 12000\pi$.

Examples 4.1 and 4.2 use sinusoidal signals to illustrate some of the ambiguities that are inherent in the sampling operation. Example 4.1 verifies that if the conditions of the sampling theorem hold, the original signal can be reconstructed from the samples. Example 4.2 illustrates that if the sampling frequency violates the sampling theorem, we cannot reconstruct the original signal using an ideal lowpass reconstruction filter with cutoff frequency at one-half the sampling frequency. The signal that is reconstructed is one of the alias frequencies of the original signal with respect to the sampling rate used in sampling the original continuous-time signal. In both examples, the sequence of samples was $x[n] = \cos(2\pi n/3)$, but the original continuous-time signal was different. As suggested by these two examples, there are unlimited ways of obtaining this same set of samples by periodic sampling of a continuous-time sinusoid. All ambiguity is removed, however, if we choose $\Omega_s > 2\Omega_0$.

4.3 RECONSTRUCTION OF A BANDLIMITED SIGNAL FROM ITS SAMPLES

According to the sampling theorem, samples of a continuous-time bandlimited signal taken frequently enough are sufficient to represent the signal exactly, in the sense that the signal can be recovered from the samples and with knowledge of the sampling period. Impulse train modulation provides a convenient means for understanding the process of reconstructing the continuous-time bandlimited signal from its samples.

In Section 4.2, we saw that if the conditions of the sampling theorem are met and if the modulated impulse train is filtered by an appropriate lowpass filter, then the Fourier transform of the filter output will be identical to the Fourier transform of the original continuous-time signal $x_c(t)$, and thus, the output of the filter will be $x_c(t)$. If we are given a sequence of samples, $x[n]$, we can form an impulse train $x_s(t)$ in which successive impulses are assigned an area equal to successive sequence values, i.e.,

$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT). \tag{4.22}$$

The n^{th} sample is associated with the impulse at $t = nT$, where T is the sampling period associated with the sequence $x[n]$. If this impulse train is the input to an ideal lowpass continuous-time filter with frequency response $H_r(j\Omega)$ and impulse response $h_r(t)$, then the output of the filter will be

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]h_r(t - nT). \tag{4.23}$$

A block diagram representation of this signal reconstruction process is shown in Figure 4.7(a). Recall that the ideal reconstruction filter has a gain of T [to compensate for the factor of $1/T$ in Eq. (4.19) or (4.20)] and a cutoff frequency Ω_c between Ω_N and $\Omega_s - \Omega_N$. A convenient and commonly used choice of the cutoff frequency is $\Omega_c = \Omega_s/2 = \pi/T$. This choice is appropriate for any relationship between Ω_s and Ω_N that avoids aliasing (i.e., so long as $\Omega_s \geq 2\Omega_N$). Figure 4.7(b) shows the frequency response of the ideal reconstruction filter. The corresponding impulse response, $h_r(t)$, is the inverse Fourier transform of $H_r(j\Omega)$, and for cutoff frequency π/T it is given by

$$h_r(t) = \frac{\sin(\pi t/T)}{\pi t/T}. \tag{4.24}$$

This impulse response is shown in Figure 4.7(c). Substituting Eq. (4.24) into Eq. (4.23) leads to

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]\frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T}. \tag{4.25}$$

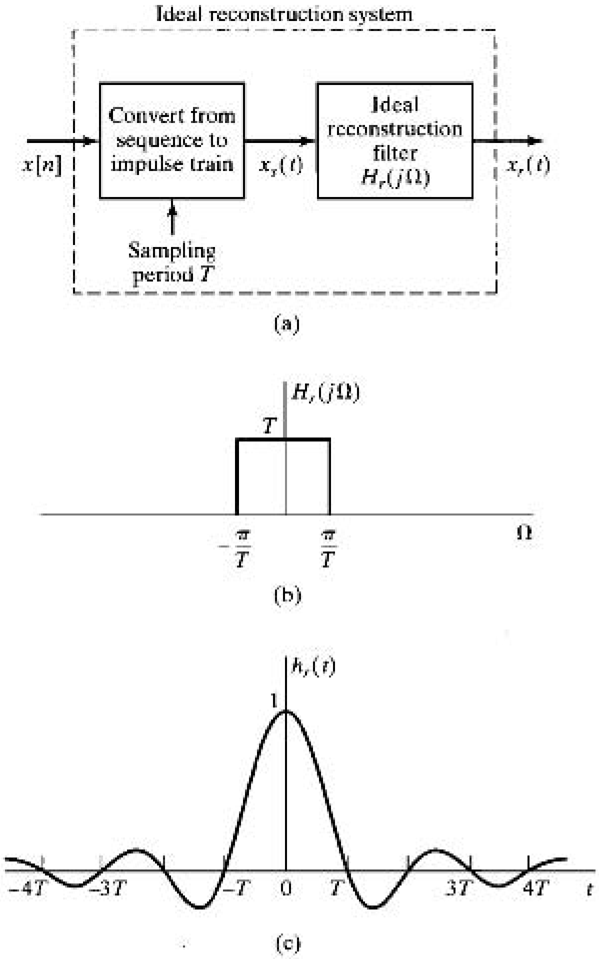


Figure 4.7 (a) Block diagram of an ideal bandlimited signal reconstruction system. (b) Frequency response of an ideal reconstruction filter. (c) Impulse response of an ideal reconstruction filter.

Equations (4.23) and (4.25) express the continuous-time signal in terms of a linear combination of basis functions $h_r(t - nT)$ with the samples $x[n]$ playing the role of coefficients. Other choices of the basis functions and corresponding coefficients could be used to represent other classes of continuous-time functions [see, for example Unser (2000)]. However, the functions in Eq. (4.24) and the samples $x[n]$ are the natural basis functions and coefficients for representing bandlimited continuous-time signals.

From the frequency-domain argument of Section 4.2, we saw that if $x[n] = x_c(nT)$, where $X_c(j\Omega) = 0$ for $|\Omega| \geq \pi/T$, then $x_r(t)$ is equal to $x_c(t)$. It is not immediately obvious that this is true by considering Eq. (4.25) alone. However, useful insight is gained by looking at that equation more closely. First, let us consider the function $h_r(t)$ given by Eq. (4.24). We note that

$$h_r(0) = 1. \qquad (4.26a)$$

This follows from l'Hôpital's rule or the small angle approximation for the sine function. In addition,

$$h_r(nT) = 0 \qquad \text{for } n = \pm 1, \pm 2, \ldots. \tag{4.26b}$$

It follows from Eqs. (4.26a) and (4.26b) and Eq. (4.23) that if $x[n] = x_c(nT)$, then

$$x_r(mT) = x_c(mT) \tag{4.27}$$

for all integer values of m. That is, the signal that is reconstructed by Eq. (4.25) has the same values at the sampling times as the original continuous-time signal, independently of the sampling period T.

In Figure 4.8, we show a continuous-time signal $x_c(t)$ and the corresponding modulated impulse train. Figure 4.8(c) shows several of the terms

$$x[n]\frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T}$$

and the resulting reconstructed signal $x_r(t)$. As suggested by this figure, the ideal lowpass filter *interpolates* between the impulses of $x_s(t)$ to construct a continuous-time signal $x_r(t)$. From Eq. (4.27), the resulting signal is an exact reconstruction of $x_c(t)$ at the sampling times. The fact that, if there is no aliasing, the lowpass filter interpolates the

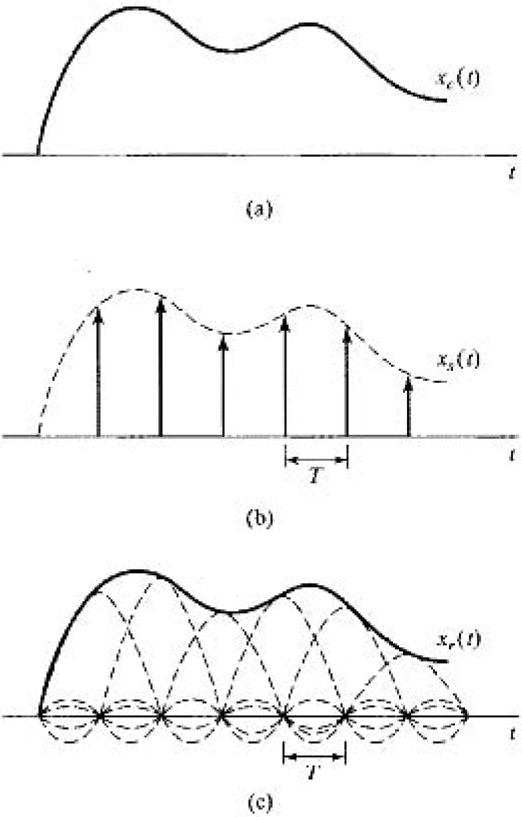


Figure 4.8 Ideal bandlimited interpolation.

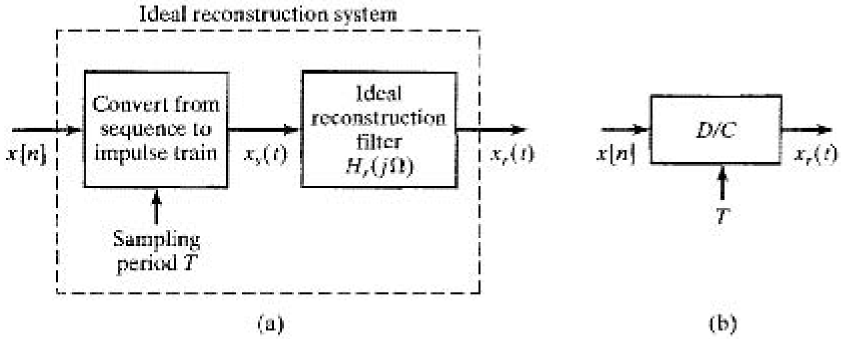

Ideal reconstruction system
Convert from sequence to impulse train
$x[n]$
$x_s(t)$
Ideal reconstruction filter $H_r(j\Omega)$
$x_r(t)$
Sampling period T
D/C
$x[n]$
$x_r(t)$
T
(a)
(b)

Figure 4.9 (a) Ideal bandlimited signal reconstruction. (b) Equivalent representation as an ideal D/C converter.

correct reconstruction between the samples follows from our frequency-domain analysis of the sampling and reconstruction process.

It is useful to formalize the preceding discussion by defining an ideal system for reconstructing a bandlimited signal from a sequence of samples. We will call this system the *ideal discrete-to-continuous-time (D/C) converter*. The desired system is depicted in Figure 4.9. As we have seen, the ideal reconstruction process can be represented as the conversion of the sequence to an impulse train, as in Eq. (4.22), followed by filtering with an ideal lowpass filter, resulting in the output given by Eq. (4.25). The intermediate step of conversion to an impulse train is a mathematical convenience in deriving Eq. (4.25) and in understanding the signal reconstruction process. However, once we are familiar with this process, it is useful to define a more compact representation, as depicted in Figure 4.9(b), where the input is the sequence $x[n]$ and the output is the continuous-time signal $x_r(t)$ given by Eq. (4.25).

The properties of the ideal D/C converter are most easily seen in the frequency domain. To derive an input/output relation in this domain, consider the Fourier transform of Eq. (4.23) or Eq. (4.25), which is

$$X_r(j\Omega) = \sum_{n=-\infty}^{\infty} x[n]H_r(j\Omega)e^{-j\Omega T n}.$$

Since $H_r(j\Omega)$ is common to all the terms in the sum, we can write

$$X_r(j\Omega) = H_r(j\Omega)X(e^{j\Omega T}). \tag{4.28}$$

Equation (4.28) provides a frequency-domain description of the ideal D/C converter. According to Eq. (4.28), $X(e^{j\omega})$ is frequency scaled (in effect, going from the sequence to the impulse train causes ω to be replaced by ΩT). Then the ideal lowpass filter $H_r(j\Omega)$ selects the base period of the resulting periodic Fourier transform $X(e^{j\Omega T})$ and compensates for the $1/T$ scaling inherent in sampling. Thus, if the sequence $x[n]$ has been obtained by sampling a bandlimited signal at the Nyquist rate or higher, the reconstructed signal $x_r(t)$ will be equal to the original bandlimited signal. In any case, it is also clear from Eq. (4.28) that the output of the ideal D/C converter is always bandlimited to at most the cutoff frequency of the lowpass filter, which is typically taken to be one-half the sampling frequency.

4.4 DISCRETE-TIME PROCESSING OF CONTINUOUS-TIME SIGNALS

A major application of discrete-time systems is in the processing of continuous-time signals. This is accomplished by a system of the general form depicted in Figure 4.10. The system is a cascade of a C/D converter, followed by a discrete-time system, followed by a D/C converter. Note that the overall system is equivalent to a continuous-time system, since it transforms the continuous-time input signal $x_c(t)$ into the continuous-time output signal $y_r(t)$. The properties of the overall system are dependent on the choice of the discrete-time system and the sampling rate. We assume in Figure 4.10 that the C/D and D/C converters have the same sampling rate. This is not essential, and later sections of this chapter and some of the problems at the end of the chapter consider systems in which the input and output sampling rates are not the same.

The previous sections of the chapter have been devoted to understanding the C/D and D/C conversion operations in Figure 4.10. For convenience, and as a first step in understanding the overall system of Figure 4.10, we summarize the mathematical representations of these operations.

The C/D converter produces a discrete-time signal

$$x[n] = x_c(nT), \tag{4.29}$$

i.e., a sequence of samples of the continuous-time input signal $x_c(t)$. The DTFT of this sequence is related to the continuous-time Fourier transform of the continuous-time input signal by

$$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left[j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right]. \tag{4.30}$$

The D/C converter creates a continuous-time output signal of the form

$$y_r(t) = \sum_{n=-\infty}^{\infty} y[n]\frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T}, \tag{4.31}$$

where the sequence $y[n]$ is the output of the discrete-time system when the input to the system is $x[n]$. From Eq. (4.28), $Y_r(j\Omega)$, the continuous-time Fourier transform of $y_r(t)$, and $Y(e^{j\omega})$, the DTFT of $y[n]$, are related by

$$Y_r(j\Omega) = H_r(j\Omega)Y(e^{j\Omega T}) = \begin{cases} TY(e^{j\Omega T}), & |\Omega| < \pi/T, \\ 0, & \text{otherwise.} \end{cases} \tag{4.32}$$

Next, let us relate the output sequence $y[n]$ to the input sequence $x[n]$, or equivalently, $Y(e^{j\omega})$ to $X(e^{j\omega})$. A simple example is the identity system, i.e., $y[n] = x[n]$. This

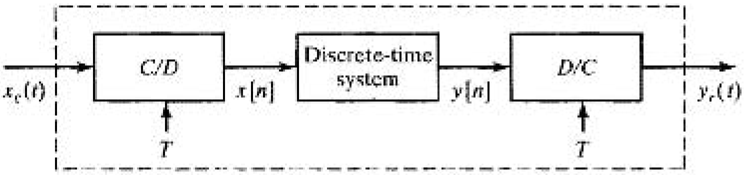


Figure 4.10 Discrete-time processing of continuous-time signals.

is in effect the case that we have studied in detail so far. We know that if $x_c(t)$ has a bandlimited Fourier transform such that $X_c(j\Omega) = 0$ for $|\Omega| \geq \pi/T$ and if the discrete-time system in Figure 4.10 is the identity system such that $y[n] = x[n] = x_c(nT)$, then the output will be $y_r(t) = x_c(t)$. Recall that, in proving this result, we utilized the frequency-domain representations of the continuous-time and discrete-time signals, since the key concept of aliasing is most easily understood in the frequency domain. Likewise, when we deal with systems more complicated than the identity system, we generally carry out the analysis in the frequency domain. If the discrete-time system is nonlinear or time varying, it is usually difficult to obtain a general relationship between the Fourier transforms of the input and the output of the system. (In Problem 4.51, we consider an example of the system of Figure 4.10 in which the discrete-time system is nonlinear.) However, the LTI case leads to a rather simple and generally useful result.

4.4.1 Discrete-Time LTI Processing of Continuous-Time Signals

If the discrete-time system in Figure 4.10 is linear and time invariant, we have

$$Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}), \tag{4.33}$$

where $H(e^{j\omega})$ is the frequency response of the system or, equivalently, the Fourier transform of the unit sample response, and $X(e^{j\omega})$ and $Y(e^{j\omega})$ are the Fourier transforms of the input and output, respectively. Combining Eqs. (4.32) and (4.33), we obtain

$$Y_r(j\Omega) = H_r(j\Omega)H(e^{j\Omega T})X(e^{j\Omega T}). \tag{4.34}$$

Next, using Eq. (4.30) with $\omega = \Omega T$, we have

$$Y_r(j\Omega) = H_r(j\Omega)H(e^{j\Omega T})\frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left[j\left(\Omega - \frac{2\pi k}{T}\right)\right]. \tag{4.35}$$

If $X_c(j\Omega) = 0$ for $|\Omega| \geq \pi/T$, then the ideal lowpass reconstruction filter $H_r(j\Omega)$ cancels the factor $1/T$ and selects only the term in Eq. (4.35) for $k = 0$; i.e.,

$$Y_r(j\Omega) = \begin{cases} H(e^{j\Omega T})X_c(j\Omega), & |\Omega| < \pi/T, \\ 0, & |\Omega| \geq \pi/T. \end{cases} \tag{4.36}$$

Thus, if $X_c(j\Omega)$ is bandlimited and the sampling rate is at or above the Nyquist rate, the output is related to the input through an equation of the form

$$Y_r(j\Omega) = H_{\text{eff}}(j\Omega)X_c(j\Omega), \tag{4.37}$$

where

$$H_{\text{eff}}(j\Omega) = \begin{cases} H(e^{j\Omega T}), & |\Omega| < \pi/T, \\ 0, & |\Omega| \geq \pi/T. \end{cases} \tag{4.38}$$

That is, the overall continuous-time system is equivalent to an LTI system whose *effective* frequency response is given by Eq. (4.38).

It is important to emphasize that the linear and time-invariant behavior of the system of Figure 4.10 depends on two factors. First, the discrete-time system must be linear and time invariant. Second, the input signal must be bandlimited, and the sampling rate must be high enough so that any aliased components are removed by the discrete-time system. As a simple illustration of this second condition being violated, consider the case when $x_c(t)$ is a single finite-duration unit-amplitude pulse whose duration is less than the sampling period. If the pulse is unity at $t = 0$, then $x[n] = \delta[n]$. However, it is clearly possible to shift the pulse so that it is not aligned with any of the sampling times, i.e., $x[n] = 0$ for all n. Such a pulse, being limited in time, is not bandlimited, and the conditions of the sampling theorem cannot hold. Even if the discrete-time system is the identity system, such that $y[n] = x[n]$, the overall system will not be time invariant if aliasing occurs in sampling the input. In general, if the discrete-time system in Figure 4.10 is linear and time invariant, and if the sampling frequency is at or above the Nyquist rate associated with the bandwidth of the input $x_c(t)$, then the overall system will be equivalent to an LTI continuous-time system with an effective frequency response given by Eq. (4.38). Furthermore, Eq. (4.38) is valid even if some aliasing occurs in the C/D converter, as long as $H(e^{j\omega})$ does not pass the aliased components. Example 4.3 is a simple illustration of this.

Example 4.3 Ideal Continuous-Time Lowpass Filtering Using a Discrete-Time Lowpass Filter

Consider Figure 4.10, with the LTI discrete-time system having frequency response

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| < \omega_c, \\ 0, & \omega_c < |\omega| \le \pi. \end{cases} \tag{4.39}$$

This frequency response is periodic with period 2π, as shown in Figure 4.11(a). For bandlimited inputs sampled at or above the Nyquist rate, it follows from Eq. (4.38) that the overall system of Figure 4.10 will behave as an LTI continuous-time system with frequency response

$$H_{\text{eff}}(j\Omega) = \begin{cases} 1, & |\Omega T| < \omega_c \text{ or } |\Omega| < \omega_c/T, \\ 0, & |\Omega T| \ge \omega_c \text{ or } |\Omega| \ge \omega_c/T. \end{cases} \tag{4.40}$$

As shown in Figure 4.11(b), this effective frequency response is that of an ideal lowpass filter with cutoff frequency $\Omega_c = \omega_c/T$

The graphical illustration given in Figure 4.12 provides an interpretation of how this effective response is achieved. Figure 4.12(a) represents the Fourier transform

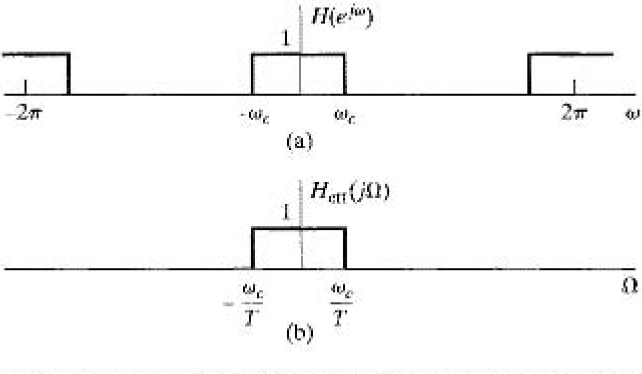


Figure 4.11 (a) Frequency response of discrete-time system in Figure 4.10. (b) Corresponding effective continuous-time frequency response for bandlimited inputs.

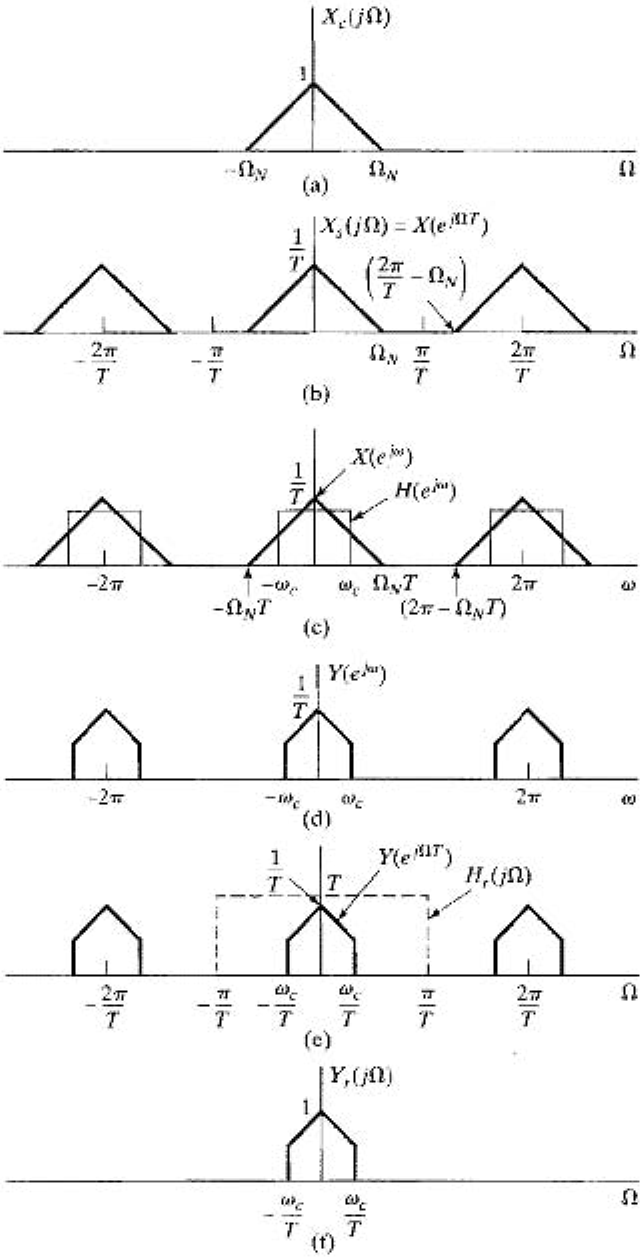


Figure 4.12 (a) Fourier transform of a bandlimited input signal. (b) Fourier transform of sampled input plotted as a function of continuous-time frequency Ω. (c) Fourier transform $X(e^{j\omega})$ of sequence of samples and frequency response $H(e^{j\omega})$ of discrete-time system plotted versus ω. (d) Fourier transform of output of discrete-time system. (e) Fourier transform of output of discrete-time system and frequency response of ideal reconstruction filter plotted versus Ω. (f) Fourier transform of output.

of a bandlimited signal. Figure 4.12(b) shows the Fourier transform of the intermediate modulated impulse train, which is identical to $X(e^{j\Omega T})$, the DTFT of the sequence of samples evaluated for $\omega = \Omega T$. In Figure 4.12(c), the DTFT of the sequence of samples and the frequency response of the discrete-time system are both plotted as a function of the normalized discrete-time frequency variable ω. Figure 4.12(d) shows $Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$, the Fourier transform of the output of the discrete-time system. Figure 4.12(e) illustrates the Fourier transform of the output of the discrete-time system as a function of the continuous-time frequency Ω, together with the frequency response of the ideal reconstruction filter $H_r(j\Omega)$ of the D/C converter. Finally, Figure 4.12(f) shows the resulting Fourier transform of the output of the D/C converter. By comparing Figures 4.12(a) and 4.12(f), we see that the system behaves as an LTI system with frequency response given by Eq. (4.40) and plotted in Figure 4.11(b).

Several important points are illustrated in Example 4.3. First, note that the ideal lowpass discrete-time filter with discrete-time cutoff frequency ω_c has the effect of an ideal lowpass filter with cutoff frequency $\Omega_c = \omega_c/T$ when used in the configuration of Figure 4.10. This cutoff frequency depends on both ω_c and T. In particular, by using a fixed discrete-time lowpass filter, but varying the sampling period T, an equivalent continuous-time lowpass filter with a variable cutoff frequency can be implemented. For example, if T were chosen so that $\Omega_N T < \omega_c$, then the output of the system of Figure 4.10 would be $y_r(t) = x_c(t)$. Also, as illustrated in Problem 4.31, Eq. (4.40) will be valid even if some aliasing is present in Figures 4.12(b) and (c), as long as these distorted (aliased) components are eliminated by the filter $H(e^{j\omega})$. In particular, from Figure 4.12(c), we see that for no aliasing to be present in the output, we require that

$$(2\pi - \Omega_N T) \geq \omega_c, \tag{4.41}$$

compared with the Nyquist requirement that

$$(2\pi - \Omega_N T) \geq \Omega_N T. \tag{4.42}$$

As another example of continuous-time processing using a discrete-time system, let us consider the implementation of an ideal differentiator for bandlimited signals.

Example 4.4 Discrete-Time Implementation of an Ideal Continuous-Time Bandlimited Differentiator

The ideal continuous-time differentiator system is defined by

$$y_c(t) = \frac{d}{dt}[x_c(t)], \tag{4.43}$$

with corresponding frequency response

$$H_c(j\Omega) = j\Omega. \tag{4.44}$$

Since we are considering a realization in the form of Figure 4.10, the inputs are restricted to be bandlimited. For processing bandlimited signals, it is sufficient that

$$H_{\text{eff}}(j\Omega) = \begin{cases} j\Omega, & |\Omega| < \pi/T, \\ 0, & |\Omega| \geq \pi/T, \end{cases} \tag{4.45}$$

as depicted in Figure 4.13(a). The corresponding discrete-time system has frequency response

$$H(e^{j\omega}) = \frac{j\omega}{T}, \qquad |\omega| < \pi, \tag{4.46}$$

and is periodic with period 2π. This frequency response is plotted in Figure 4.13(b). The corresponding impulse response can be shown to be

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\frac{j\omega}{T}\right)e^{j\omega n}d\omega = \frac{\pi n\cos\pi n - \sin\pi n}{\pi n^2 T}, \qquad -\infty < n < \infty,$$

or equivalently,

$$h[n] = \begin{cases} 0, & n = 0, \\ \dfrac{\cos\pi n}{nT}, & n \neq 0. \end{cases} \tag{4.47}$$

Thus, if a discrete-time system with this impulse response was used in the configuration of Figure 4.10, the output for every appropriately bandlimited input would be the derivative of the input. Problem 4.22 concerns the verification of this for a sinusoidal input signal.

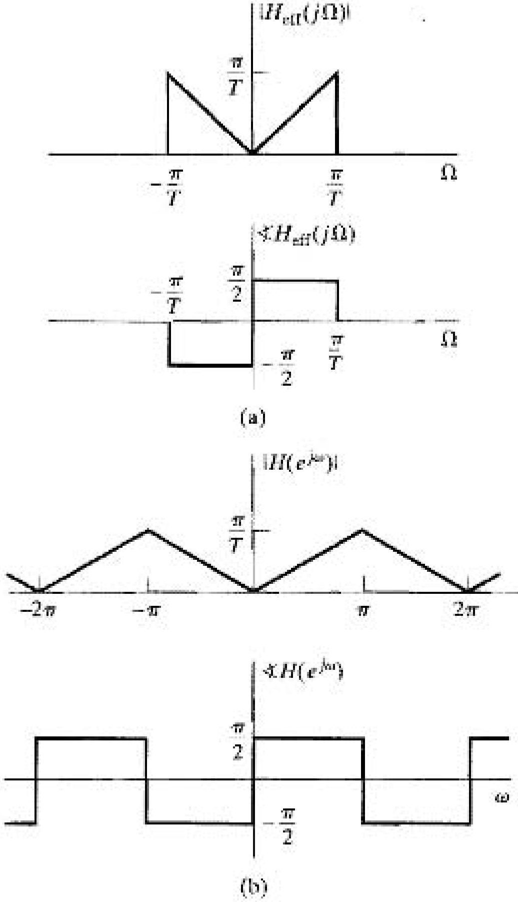


Figure 4.13 (a) Frequency response of a continuous-time ideal bandlimited differentiator $H_c(j\Omega) = j\Omega$, $|\Omega| < \pi/T$. (b) Frequency response of a discrete-time filter to implement a continuous-time bandlimited differentiator.

4.4.2 Impulse Invariance

We have shown that the cascade system of Figure 4.10 can be equivalent to an LTI system for bandlimited input signals. Let us now assume that, as depicted in Figure 4.14, we are given a desired continuous-time system that we wish to implement in the form of Figure 4.10. With $H_c(j\Omega)$ bandlimited, Eq. (4.38) specifies how to choose $H(e^{j\omega})$ so that $H_{\text{eff}}(j\Omega) = H_c(j\Omega)$. Specifically,

$$H(e^{j\omega}) = H_c(j\omega/T), \qquad |\omega| < \pi, \tag{4.48}$$

with the further requirement that T be chosen such that

$$H_c(j\Omega) = 0, \qquad |\Omega| \geq \pi/T. \tag{4.49}$$

Under the constraints of Eqs. (4.48) and (4.49), there is also a straightforward and useful relationship between the continuous-time impulse response $h_c(t)$ and the discrete-time impulse response $h[n]$. In particular, as we shall verify shortly,

$$h[n] = Th_c(nT); \tag{4.50}$$

i.e., the impulse response of the discrete-time system is a scaled, sampled version of $h_c(t)$. When $h[n]$ and $h_c(t)$ are related through Eq. (4.50), the discrete-time system is said to be an *impulse-invariant* version of the continuous-time system.

Equation (4.50) is a direct consequence of the discussion in Section 4.2. Specifically, with $x[n]$ and $x_c(t)$ respectively replaced by $h[n]$ and $h_c(t)$ in Eq. (4.16), i.e.,

$$h[n] = h_c(nT), \tag{4.51}$$

Eq. (4.20) becomes

$$H(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right), \tag{4.52}$$

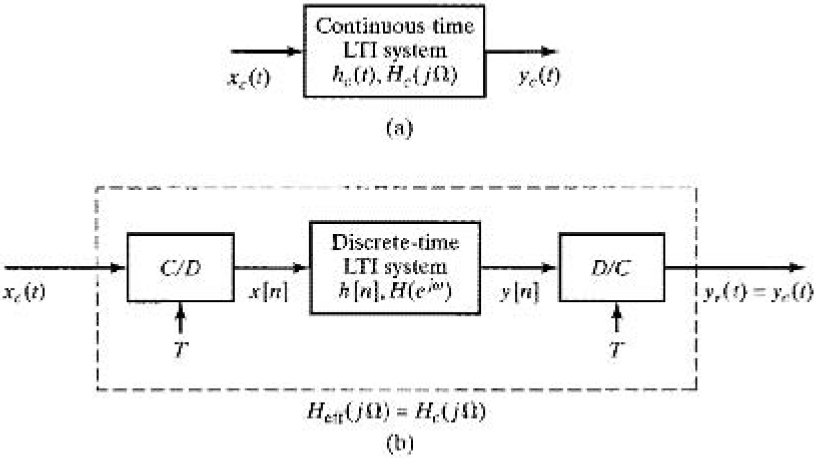


Figure 4.14 (a) Continuous-time LTI system. (b) Equivalent system for bandlimited inputs.

or, if Eq. (4.49) is satisfied,

$$H(e^{j\omega}) = \frac{1}{T} H_c\left(j\frac{\omega}{T}\right), \qquad |\omega| < \pi. \tag{4.53}$$

Modifying Eqs. (4.51) and (4.53) to account for the scale factor of T in Eq. (4.50), we have

$$h[n] = Th_c(nT), \tag{4.54}$$

$$H(e^{j\omega}) = H_c\left(j\frac{\omega}{T}\right), \qquad |\omega| < \pi. \tag{4.55}$$

Example 4.5 A Discrete-Time Lowpass Filter Obtained by Impulse Invariance

Suppose that we wish to obtain an ideal lowpass discrete-time filter with cutoff frequency $\omega_c < \pi$. We can do this by sampling a continuous-time ideal lowpass filter with cutoff frequency $\Omega_c = \omega_c/T < \pi/T$ defined by

$$H_c(j\Omega) = \begin{cases} 1, & |\Omega| < \Omega_c, \\ 0, & |\Omega| \geq \Omega_c. \end{cases}$$

The impulse response of this continuous-time system is

$$h_c(t) = \frac{\sin(\Omega_c t)}{\pi t},$$

so we define the impulse response of the discrete-time system to be

$$h[n] = Th_c(nT) = T\frac{\sin(\Omega_c nT)}{\pi nT} = \frac{\sin(\omega_c n)}{\pi n},$$

where $\omega_c = \Omega_c T$. We have already shown that this sequence corresponds to the DTFT

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| < \omega_c, \\ 0, & \omega_c \leq |\omega| \leq \pi, \end{cases}$$

which is identical to $H_c(j\omega/T)$, as predicted by Eq. (4.55).

Example 4.6 Impulse Invariance Applied to Continuous-Time Systems with Rational System Functions

Many continuous-time systems have impulse responses composed of a sum of exponential sequences of the form

$$h_c(t) = A\,e^{s_0 t}u(t).$$

Such time functions have Laplace transforms

$$H_c(s) = \frac{A}{s - s_0} \qquad \mathcal{R}e(s) > \mathcal{R}e(s_0).$$

If we apply the impulse invariance concept to such a continuous-time system, we obtain the impulse response

$$h[n] = Th_c(nT) = A\,T\,e^{s_0 Tn}u[n],$$

which has z-transform system function

$$H(z) = \frac{AT}{1 - e^{s_0 T} z^{-1}} \qquad |z| > |e^{s_0 T}|$$

and, assuming $\mathcal{R}e(s_0) < 0$, the frequency response

$$H(e^{j\omega}) = \frac{AT}{1 - e^{s_0 T} e^{-j\omega}}.$$

In this case, Eq. (4.55) does not hold exactly, because the original continuous-time system did not have a strictly bandlimited frequency response, and therefore, the resulting discrete-time frequency response is an *aliased* version of $H_c(j\Omega)$. Even though aliasing occurs in such a case as this, the effect may be small. Higher-order systems whose impulse responses are sums of complex exponentials may in fact have frequency responses that fall off rapidly at high frequencies, so that aliasing is minimal if the sampling rate is high enough. Thus, one approach to the discrete-time simulation of continuous-time systems and also to the design of digital filters is through sampling of the impulse response of a corresponding analog filter.

4.5 CONTINUOUS-TIME PROCESSING OF DISCRETE-TIME SIGNALS

In Section 4.4, we discussed and analyzed the use of discrete-time systems for processing continuous-time signals in the configuration of Figure 4.10. In this section, we consider the complementary situation depicted in Figure 4.15, which is appropriately referred to as continuous-time processing of discrete-time signals. Although the system of Figure 4.15 is not typically used to implement discrete-time systems, it provides a useful interpretation of certain discrete-time systems that have no simple interpretation in the discrete domain.

From the definition of the ideal D/C converter, $X_c(j\Omega)$ and therefore also $Y_c(j\Omega)$, will necessarily be zero for $|\Omega| \geq \pi/T$. Thus, the C/D converter samples $y_c(t)$ without aliasing, and we can express $x_c(t)$ and $y_c(t)$ respectively as

$$x_c(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T} \tag{4.56}$$

and

$$y_c(t) = \sum_{n=-\infty}^{\infty} y[n] \frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T}, \tag{4.57}$$

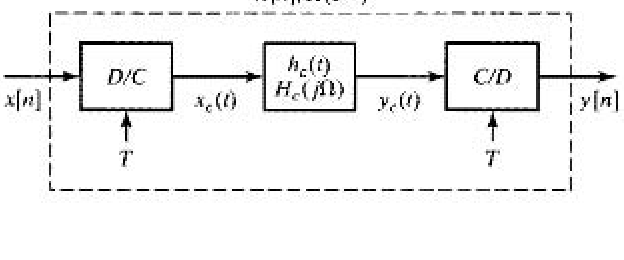


Figure 4.15 Continuous-time processing of discrete-time signals.

where $x[n] = x_c(nT)$ and $y[n] = y_c(nT)$. The frequency-domain relationships for Figure 4.15 are

$$X_c(j\Omega) = TX(e^{j\Omega T}), \qquad |\Omega| < \pi/T, \tag{4.58a}$$

$$Y_c(j\Omega) = H_c(j\Omega)X_c(j\Omega), \tag{4.58b}$$

$$Y(e^{j\omega}) = \frac{1}{T}Y_c\left(j\frac{\omega}{T}\right), \qquad |\omega| < \pi. \tag{4.58c}$$

Therefore, by substituting Eqs. (4.58a) and (4.58b) into Eq. (4.58c), it follows that the overall system behaves as a discrete-time system whose frequency response is

$$H(e^{j\omega}) = H_c\left(j\frac{\omega}{T}\right), \qquad |\omega| < \pi, \tag{4.59}$$

or equivalently, the overall frequency response of the system in Figure 4.15 will be equal to a given $H(e^{j\omega})$ if the frequency response of the continuous-time system is

$$H_c(j\Omega) = H(e^{j\Omega T}), \qquad |\Omega| < \pi/T. \tag{4.60}$$

Since $X_c(j\Omega) = 0$ for $|\Omega| \geq \pi/T$, $H_c(j\Omega)$ may be chosen arbitrarily above π/T. A convenient—but arbitrary—choice is $H_c(j\Omega) = 0$ for $|\Omega| \geq \pi/T$.

With this representation of a discrete-time system, we can focus on the equivalent effect of the continuous-time system on the bandlimited continuous-time signal $x_c(t)$. This is illustrated in Examples 4.7 and 4.8.

Example 4.7 Noninteger Delay

Let us consider a discrete-time system with frequency response

$$H(e^{j\omega}) = e^{-j\omega\Delta}, \qquad |\omega| < \pi. \tag{4.61}$$

When Δ is an integer, this system has a straightforward interpretation as a delay of Δ, i.e.,

$$y[n] = x[n - \Delta]. \tag{4.62}$$

When Δ is not an integer, Eq. (4.62) has no formal meaning, because we cannot shift the sequence $x[n]$ by a noninteger amount. However, with the use of the system of Figure 4.15, a useful time-domain interpretation can be applied to the system specified by Eq. (4.61). Let $H_c(j\Omega)$ in Figure 4.15 be chosen to be

$$H_c(j\Omega) = H(e^{j\Omega T}) = e^{-j\Omega T\Delta}. \tag{4.63}$$

Then, from Eq. (4.59), the overall discrete-time system in Figure 4.15 will have the frequency response given by Eq. (4.61), whether or not Δ is an integer. To interpret the system of Eq. (4.61), we note that Eq. (4.63) represents a time delay of $T\Delta$ seconds. Therefore,

$$y_c(t) = x_c(t - T\Delta). \tag{4.64}$$

Furthermore, $x_c(t)$ is the bandlimited interpolation of $x[n]$, and $y[n]$ is obtained by sampling $y_c(t)$. For example, if $\Delta = \frac{1}{2}$, $y[n]$ would be the values of the bandlimited interpolation halfway between the input sequence values. This is illustrated in

Figure 4.16. We can also obtain a direct convolution representation for the system defined by Eq. (4.61). From Eqs. (4.64) and (4.56), we obtain

$$\begin{aligned} y[n] &= y_c(nT) = x_c(nT - T\Delta) \\ &= \sum_{k=-\infty}^{\infty} x[k] \left. \frac{\sin[\pi(t - T\Delta - kT)/T]}{\pi(t - T\Delta - kT)/T} \right|_{t=nT} \\ &= \sum_{k=-\infty}^{\infty} x[k] \frac{\sin \pi(n - k - \Delta)}{\pi(n - k - \Delta)}, \end{aligned} \tag{4.65}$$

which is, by definition, the convolution of $x[n]$ with

$$h[n] = \frac{\sin \pi(n - \Delta)}{\pi(n - \Delta)}, \qquad -\infty < n < \infty.$$

When Δ is not an integer, $h[n]$ has infinite extent. However, when $\Delta = n_0$ is an integer, it is easily shown that $h[n] = \delta[n - n_0]$, which is the impulse response of the ideal integer delay system.

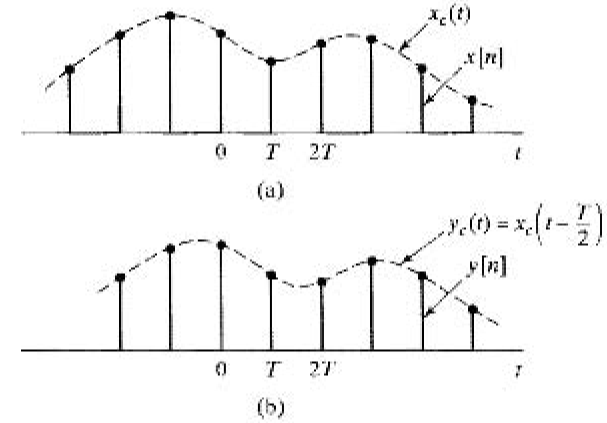


Figure 4.16 (a) Continuous-time processing of the discrete-time sequence (b) can produce a new sequence with a "half-sample" delay.

The noninteger delay represented by Eq. (4.65) has considerable practical significance, since such a factor often arises in the frequency-domain representation of systems. When this kind of term is found in the frequency response of a causal discrete-time system, it can be interpreted in the light of this example. This interpretation is illustrated in Example 4.8.

Example 4.8 Moving-Average System with Noninteger Delay

In Example 2.16, we considered the general moving-average system and obtained its frequency response. For the case of the causal $(M + 1)$-point moving-average system,

$M_1 = 0$ and $M_2 = M$, and the frequency response is

$$H(e^{j\omega}) = \frac{1}{(M+1)} \frac{\sin[\omega(M+1)/2]}{\sin(\omega/2)} e^{-j\omega M/2}, \qquad |\omega| < \pi. \tag{4.66}$$

This representation of the frequency response suggests the interpretation of the $(M + 1)$-point moving-average system as the cascade of two systems, as indicated in Figure 4.17. The first system imposes a frequency-domain amplitude weighting. The second system represents the linear-phase term in Eq. (4.66). If M is an even integer (meaning the moving average of an odd number of samples), then the linear-phase term corresponds to an integer delay, i.e.,

$$y[n] = w[n - M/2]. \tag{4.67}$$

However, if M is odd, the linear-phase term corresponds to a noninteger delay, specifically, an integer-plus-one-half sample interval. This noninteger delay can be interpreted in terms of the discussion in Example 4.7; i.e., $y[n]$ is equivalent to bandlimited interpolation of $w[n]$, followed by a continuous-time delay of $MT/2$ (where T is the assumed, but arbitrary, sampling period associated with the D/C interpolation of $w[n]$), followed by C/D conversion again with sampling period T. This fractional delay is illustrated in Figure 4.18. Figure 4.18(a) shows a discrete-time sequence $x[n] = \cos(0.25\pi n)$. This sequence is the input to a six-point ($M = 5$) moving-average filter. In this example, the input is "turned on" far enough in the past so that the output consists only of the steady-state response for the time interval shown. Figure 4.18(b) shows the corresponding output sequence, which is given by

$$\begin{aligned} y[n] &= H(e^{j0.25\pi})\frac{1}{2}e^{j0.25\pi n} + H(e^{-j0.25\pi})\frac{1}{2}e^{-j0.25\pi n} \\ &= \frac{1}{2}\frac{\sin[3(0.25\pi)]}{6\sin(0.125\pi)}e^{-j(0.25\pi)5/2}e^{j0.25\pi n} + \frac{1}{2}\frac{\sin[3(-0.25\pi)]}{6\sin(-0.125\pi)}e^{j(0.25\pi)5/2}e^{-j0.25\pi n} \\ &= 0.308\cos[0.25\pi(n - 2.5)]. \end{aligned}$$

Thus, the six-point moving-average filter reduces the amplitude of the cosine signal and introduces a phase shift that corresponds to 2.5 samples of delay. This is apparent in Figure 4.18, where we have plotted the continuous-time cosines that would be interpolated by the ideal D/C converter for both the input and the output sequence. Note in Figure 4.18(b) that the six-point moving-average filtering gives a sampled cosine signal such that the sample points have been shifted by 2.5 samples with respect to the sample points of the input. This can be seen from Figure 4.18 by comparing the positive peak at 8 in the interpolated cosine for the input to the positive peak at 10.5 in the interpolated cosine for the output. Thus, the six-point moving-average filter is seen to have a delay of $5/2 = 2.5$ samples.

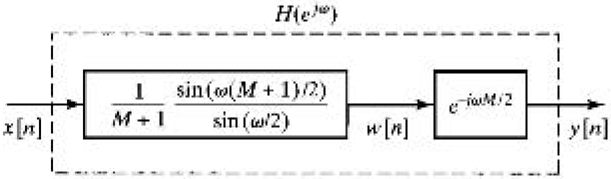


Figure 4.17 The moving-average system represented as a cascade of two systems.

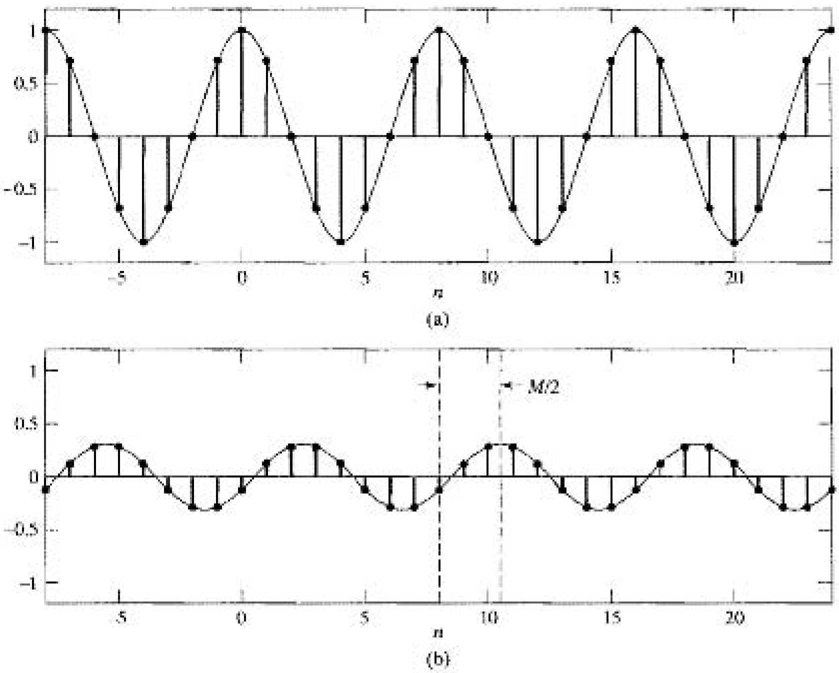


Figure 4.18 Illustration of moving-average filtering. (a) Input signal $x[n] = \cos(0.25\pi n)$. (b) Corresponding output of six-point moving-average filter.

4.6 CHANGING THE SAMPLING RATE USING DISCRETE-TIME PROCESSING

We have seen that a continuous-time signal $x_c(t)$ can be represented by a discrete-time signal consisting of a sequence of samples

$$x[n] = x_c(nT). \tag{4.68}$$

Alternatively, our previous discussion has shown that, even if $x[n]$ was not obtained originally by sampling, we can always use the bandlimited interpolation formula of Eq. (4.25) to reconstruct a continuous-time bandlimited signal $x_r(t)$ whose samples are $x[n] = x_r(nT) = x_c(nT)$, i.e., the samples of $x_c(t)$ and $x_r(t)$ are identical at the sampling times even when $x_r(t) \neq x_c(t)$.

It is often necessary to change the sampling rate of a discrete-time signal, i.e., to obtain a new discrete-time representation of the underlying continuous-time signal of the form

$$x_1[n] = x_c(nT_1), \tag{4.69}$$

where $T_1 \neq T$. This operation is often called *resampling*. Conceptually, $x_1[n]$ can be obtained from $x[n]$ by reconstructing $x_c(t)$ from $x[n]$ using Eq. (4.25) and then resampling $x_c(t)$ with period T_1 to obtain $x_1[n]$. However, this is not usually a practical approach,

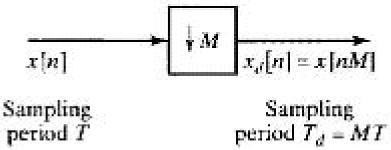


Figure 4.19 Representation of a compressor or discrete-time sampler.

because of the nonideal analog reconstruction filter, D/A converter, and A/D converter that would be used in a practical implementation. Thus, it is of interest to consider methods of changing the sampling rate that involve only discrete-time operations.

4.6.1 Sampling Rate Reduction by an Integer Factor

The sampling rate of a sequence can be reduced by "sampling" it, i.e., by defining a new sequence

$$x_d[n] = x[nM] = x_c(nMT). \tag{4.70}$$

Equation (4.70) defines the system depicted in Figure 4.19, which is called a *sampling rate compressor* (see Crochiere and Rabiner, 1983 and Vaidyanathan, 1993) or simply a *compressor*. From Eq. (4.70), it follows that $x_d[n]$ is identical to the sequence that would be obtained from $x_c(t)$ by sampling with period $T_d = MT$. Furthermore, if $X_c(j\Omega) = 0$ for $|\Omega| \geq \Omega_N$, then $x_d[n]$ is an exact representation of $x_c(t)$ if $\pi/T_d = \pi/(MT) \geq \Omega_N$. That is, the sampling rate can be reduced to π/M without aliasing if the original sampling rate is at least M times the Nyquist rate or if the bandwidth of the sequence is first reduced by a factor of M by discrete-time filtering. In general, the operation of reducing the sampling rate (including any prefiltering) is called *downsampling*.

As in the case of sampling a continuous-time signal, it is useful to obtain a frequency-domain relation between the input and output of the compressor. This time, however, it will be a relationship between DTFTs. Although several methods can be used to derive the desired result, we will base our derivation on the results already obtained for sampling continuous-time signals. First, recall that the DTFT of $x[n] = x_c(nT)$ is

$$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left[j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right]. \tag{4.71}$$

Similarly, the DTFT of $x_d[n] = x[nM] = x_c(nT_d)$ with $T_d = MT$ is

$$X_d(e^{j\omega}) = \frac{1}{T_d}\sum_{r=-\infty}^{\infty} X_c\left[j\left(\frac{\omega}{T_d} - \frac{2\pi r}{T_d}\right)\right]. \tag{4.72}$$

Now, since $T_d = MT$, we can write Eq. (4.72) as

$$X_d(e^{j\omega}) = \frac{1}{MT}\sum_{r=-\infty}^{\infty} X_c\left[j\left(\frac{\omega}{MT} - \frac{2\pi r}{MT}\right)\right]. \tag{4.73}$$

To see the relationship between Eqs. (4.73) and (4.71), note that the summation index r in Eq. (4.73) can be expressed as

$$r = i + kM, \tag{4.74}$$

where k and i are integers such that $-\infty < k < \infty$ and $0 \le i \le M-1$. Clearly, r is still an integer ranging from $-\infty$ to ∞, but now Eq. (4.73) can be expressed as

$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1}\left\{\frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left[j\left(\frac{\omega}{MT} - \frac{2\pi k}{T} - \frac{2\pi i}{MT}\right)\right]\right\}. \tag{4.75}$$

The term inside the square brackets in Eq. (4.75) is recognized from Eq. (4.71) as

$$X\,(e^{j(\omega-2\pi i)/M}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left[j\left(\frac{\omega-2\pi i}{MT} - \frac{2\pi k}{T}\right)\right]. \tag{4.76}$$

Thus, we can express Eq. (4.75) as

$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X\,(e^{j(\omega/M-2\pi i/M)}), \tag{4.77}$$

There is a strong analogy between Eqs. (4.71) and (4.77): Equation (4.71) expresses the Fourier transform of the sequence of samples, $x[n]$ (with period T), in terms of the Fourier transform of the continuous-time signal $x_c(t)$; Equation (4.77) expresses the Fourier transform of the discrete-time sampled sequence $x_d[n]$ (with sampling period M) in terms of the Fourier transform of the sequence $x[n]$. If we compare Eqs. (4.72) and (4.77), we see that $X_d(e^{j\omega})$ can be thought of as being composed of the superposition of either an infinite set of amplitude-scaled copies of $X_c(j\Omega)$, frequency scaled through $\omega = \Omega T_d$ and shifted by integer multiples of 2π [Eq. (4.72)], or M amplitude-scaled copies of the periodic Fourier transform $X\,(e^{j\omega})$, frequency scaled by M and shifted by integer multiples of 2π [Eq. (4.77)]. Either interpretation makes it clear that $X_d(e^{j\omega})$ is periodic with period 2π (as are all DTFTs) and that aliasing can be avoided by ensuring that $X\,(e^{j\omega})$ is bandlimited, i.e.,

$$X\,(e^{j\omega}) = 0, \qquad \omega_N \le |\omega| \le \pi, \tag{4.78}$$

and $2\pi/M \ge 2\omega_N$.

Downsampling is illustrated in Figure 4.20 for $M = 2$. Figure 4.20(a) shows the Fourier transform of a bandlimited continuous-time signal, and Figure 4.20(b) shows the Fourier transform of the impulse train of samples when the sampling period is T. Figure 4.20(c) shows $X\,(e^{j\omega})$ and is related to Figure 4.20(b) through Eq. (4.18). As we have already seen, Figures 4.20(b) and (c) differ only in a scaling of the frequency variable. Figure 4.20(d) shows the DTFT of the downsampled sequence when $M = 2$. We have plotted this Fourier transform as a function of the normalized frequency $\omega = \Omega T_d$. Finally, Figure 4.20(e) shows the DTFT of the downsampled sequence plotted as a function of the continuous-time frequency variable Ω. Figure 4.20(e) is identical to Figure 4.20(d), except for the scaling of the frequency axis through the relation $\Omega = \omega/T_d$.

In this example, $2\pi/T = 4\Omega_N$; i.e., the original sampling rate is exactly twice the minimum rate to avoid aliasing. Thus, when the original sampled sequence is downsampled by a factor of $M = 2$, no aliasing results. If the downsampling factor is more than 2 in this case, aliasing will result, as illustrated in Figure 4.21.

Figure 4.21(a) shows the continuous-time Fourier transform of $x_c(t)$, and Figure 4.21(b) shows the DTFT of the sequence $x[n] = x_c(nT)$, when $2\pi/T = 4\Omega_N$. Thus,

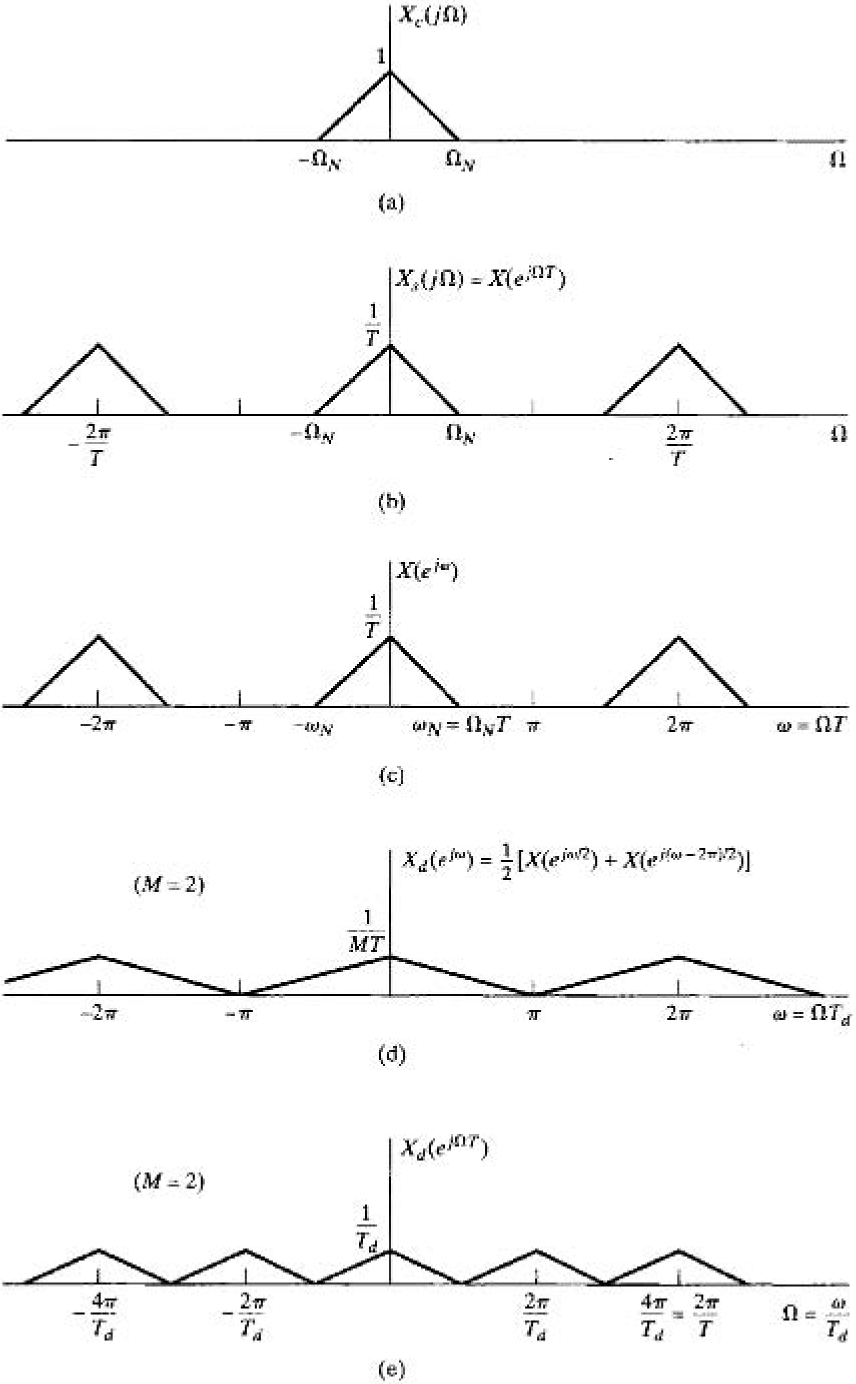


Figure 4.20 Frequency-domain illustration of downsampling.

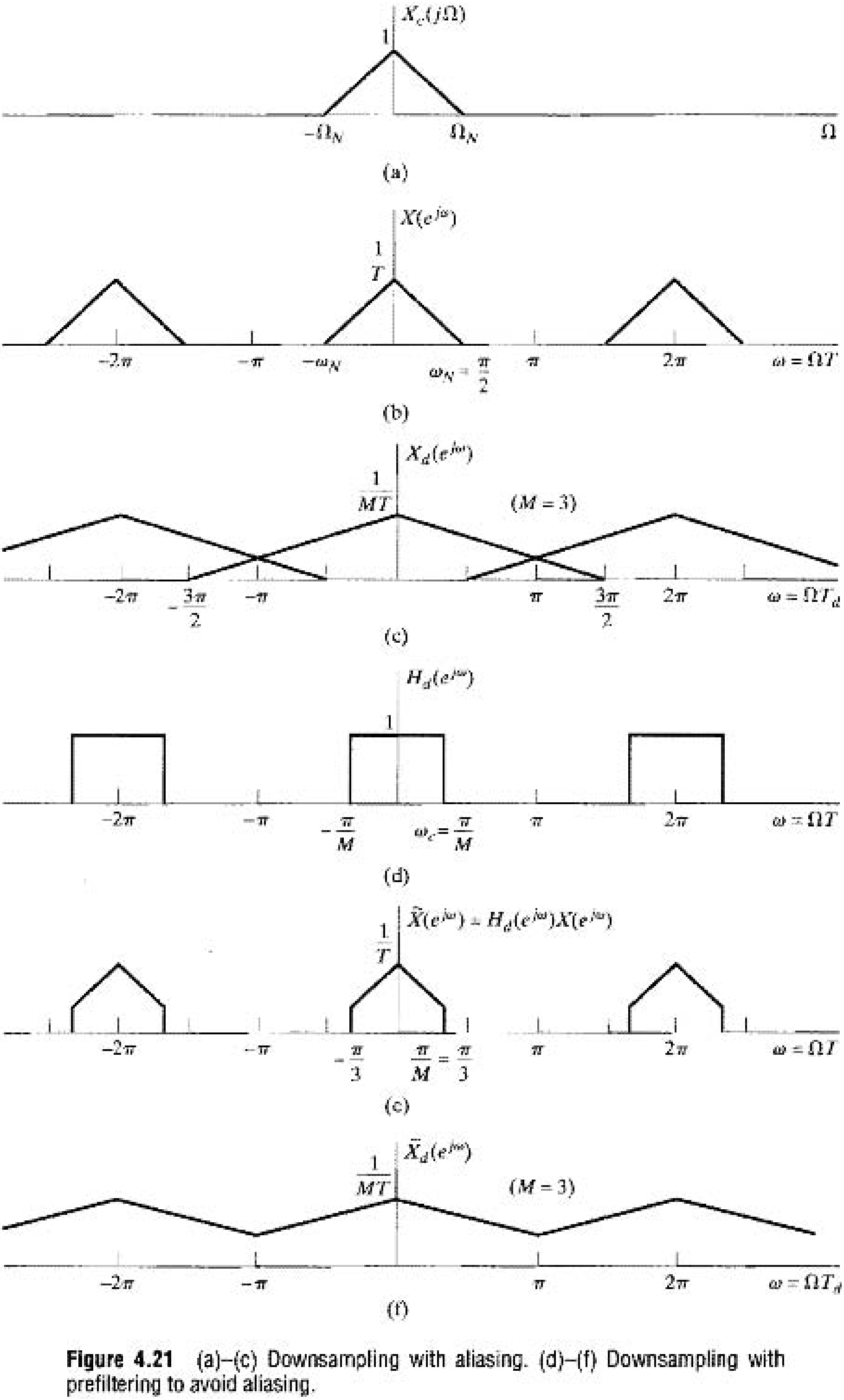


Figure 4.21 (a)–(c) Downsampling with aliasing. (d)–(f) Downsampling with prefiltering to avoid aliasing.

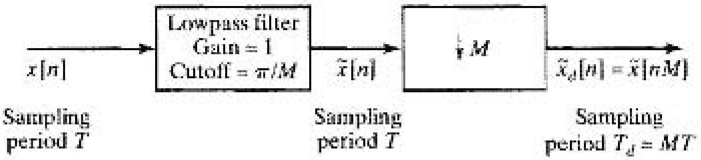


Figure 4.22 General system for sampling rate reduction by M.

$\omega_N = \Omega_N T = \pi/2$. Now, if we downsample by a factor of $M = 3$, we obtain the sequence $x_d[n] = x[3n] = x_c(n3T)$ whose DTFT is plotted in Figure 4.21(c) with normalized frequency $\omega = \Omega T_d$. Note that because $M\omega_N = 3\pi/2$, which is greater than π, aliasing occurs. In general, to avoid aliasing in downsampling by a factor of M requires that

$$\omega_N M \leq \pi \qquad \text{or} \qquad \omega_N \leq \pi/M. \tag{4.79}$$

If this condition does not hold, aliasing occurs, but it may be tolerable for some applications. In other cases, downsampling can be done without aliasing if we are willing to reduce the bandwidth of the signal $x[n]$ before downsampling. Thus, if $x[n]$ is filtered by an ideal lowpass filter with cutoff frequency π/M, then the output $\tilde{x}[n]$ can be downsampled without aliasing, as illustrated in Figures 4.21(d), (e), and (f). Note that the sequence $\tilde{x}_d[n] = \tilde{x}[nM]$ no longer represents the original underlying continuous-time signal $x_c(t)$. Rather, $\tilde{x}_d[n] = \tilde{x}_c(nT_d)$, where $T_d = MT$, and $\tilde{x}_c(t)$ is obtained from $x_c(t)$ by lowpass filtering with cutoff frequency $\Omega_c = \pi/T_d = \pi/(MT)$.

From the preceding discussion, we see that a general system for downsampling by a factor of M is the one shown in Figure 4.22. Such a system is called a *decimator*, and downsampling by lowpass filtering followed by compression has been termed *decimation* (Crochiere and Rabiner, 1983 and Vaidyanathan, 1993).

4.6.2 Increasing the Sampling Rate by an Integer Factor

We have seen that the reduction of the sampling rate of a discrete-time signal by an integer factor involves sampling the sequence in a manner analogous to sampling a continuous-time signal. Not surprisingly, increasing the sampling rate involves operations analogous to D/C conversion. To see this, consider a signal $x[n]$ whose sampling rate we wish to increase by a factor of L. If we consider the underlying continuous-time signal $x_c(t)$, the objective is to obtain samples

$$x_i[n] = x_c(nT_i), \tag{4.80}$$

where $T_i = T/L$, from the sequence of samples

$$x[n] = x_c(nT). \tag{4.81}$$

We will refer to the operation of increasing the sampling rate as *upsampling*.

From Eqs. (4.80) and (4.81), it follows that

$$x_i[n] = x[n/L] = x_c(nT/L), \qquad n = 0, \pm L, \pm 2L, \ldots. \tag{4.82}$$

Figure 4.23 shows a system for obtaining $x_i[n]$ from $x[n]$ using only discrete-time processing. The system on the left is called a *sampling rate expander* (see Crochiere and Rabiner, 1983 and Vaidyanathan, 1993) or simply an *expander*. Its output is

$$x_e[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \ldots, \\ 0, & \text{otherwise}, \end{cases} \tag{4.83}$$

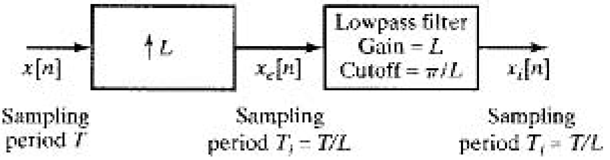


Figure 4.23 General system for sampling rate increase by L.

or equivalently,

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL]. \tag{4.84}$$

The system on the right is a lowpass discrete-time filter with cutoff frequency π/L and gain L. This system plays a role similar to the ideal D/C converter in Figure 4.9(b). First, we create a discrete-time impulse train $x_e[n]$, and we then use a lowpass filter to reconstruct the sequence.

The operation of the system in Figure 4.23 is most easily understood in the frequency domain. The Fourier transform of $x_e[n]$ can be expressed as

$$\begin{aligned} X_e(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x[k]\delta[n - kL] \right) e^{-j\omega n} \\ &= \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega Lk} = X(e^{j\omega L}). \end{aligned} \tag{4.85}$$

Thus, the Fourier transform of the output of the expander is a frequency-scaled version of the Fourier transform of the input; i.e., ω is replaced by ωL so that ω is now normalized by

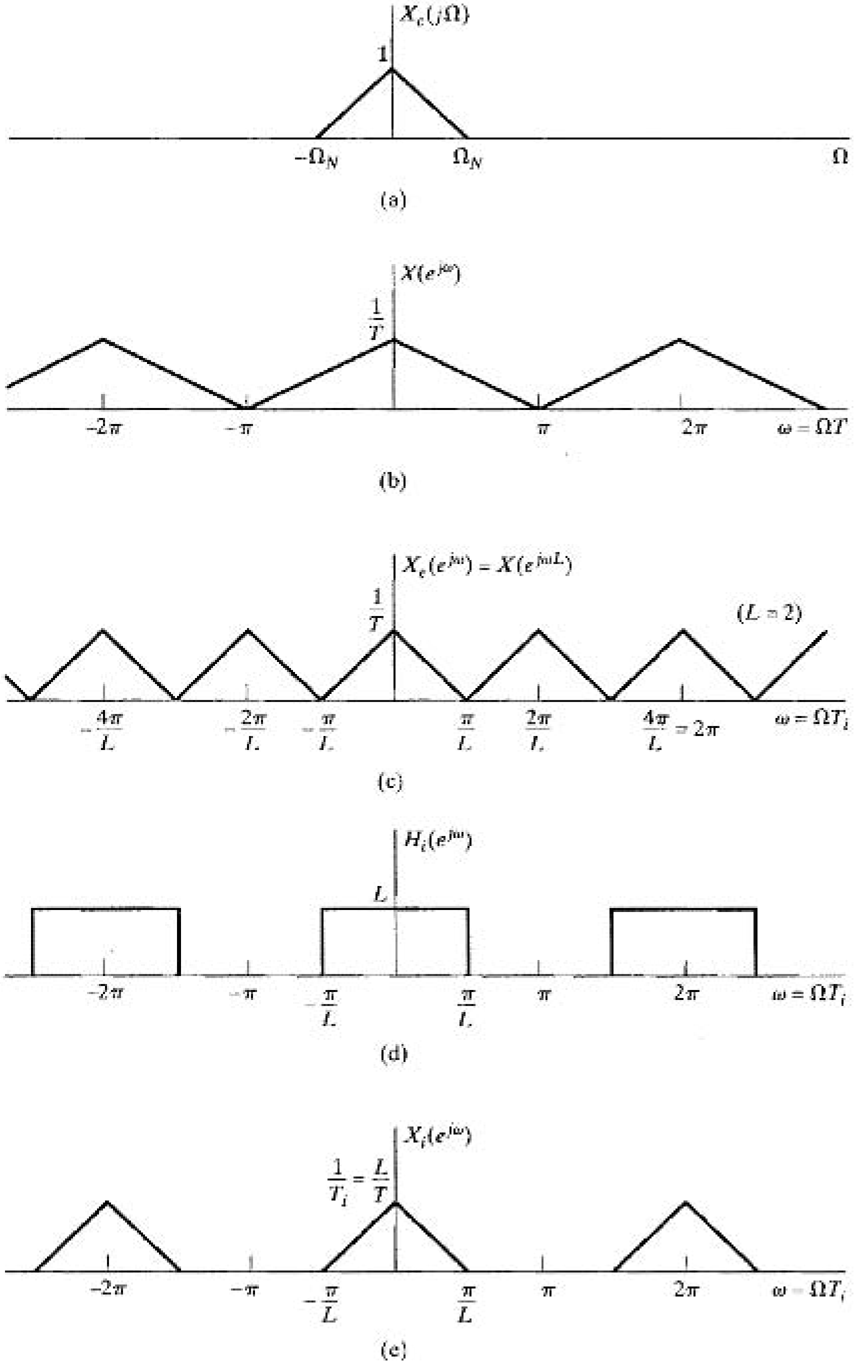
$$\omega = \Omega T_i. \tag{4.86}$$

This effect is illustrated in Figure 4.24. Figure 4.24(a) shows a bandlimited continuous-time Fourier transform, and Figure 4.24(b) shows the DTFT of the sequence $x[n] = x_c(nT)$, where $\pi/T = \Omega_N$. Figure 4.24(c) shows $X_e(e^{j\omega})$ according to Eq. (4.85), with $L = 2$, and Figure 4.24(e) shows the Fourier transform of the desired signal $x_i[n]$. We see that $X_i(e^{j\omega})$ can be obtained from $X_e(e^{j\omega})$ by correcting the amplitude scale from $1/T$ to $1/T_i$ and by removing all the frequency-scaled images of $X_c(j\Omega)$ except at integer multiples of 2π. For the case depicted in Figure 4.24, this requires a lowpass filter with a gain of 2 and cutoff frequency $\pi/2$, as shown in Figure 4.24(d). In general, the required gain would be L, since $L(1/T) = [1/(T/L)] = 1/T_i$, and the cutoff frequency would be π/L.

This example shows that the system of Figure 4.23 does indeed give an output satisfying Eq. (4.80) if the input sequence $x[n] = x_c(nT)$ was obtained by sampling without aliasing. Therefore, that system is called an *interpolator*, since it fills in the missing samples, and the operation of upsampling is consequently considered to be synonymous with *interpolation*.

As in the case of the D/C converter, it is possible to obtain an interpolation formula for $x_i[n]$ in terms of $x[n]$. First, note that the impulse response of the lowpass filter in Figure 4.23 is

$$h_i[n] = \frac{\sin(\pi n/L)}{\pi n/L}. \tag{4.87}$$



(a)

(b)

(c)

(d)

(e)

Figure 4.24 Frequency-domain illustration of interpolation.

Using Eq. (4.84), we obtain

$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k] \frac{\sin[\pi(n-kL)/L]}{\pi(n-kL)/L}. \tag{4.88}$$

The impulse response $h_i[n]$ has the properties

$$\begin{aligned} h_i[0] &= 1, \\ h_i[n] &= 0, \qquad n = \pm L, \pm 2L, \ldots. \end{aligned} \tag{4.89}$$

Thus, for the ideal lowpass interpolation filter, we have

$$x_i[n] = x[n/L] = x_c(nT/L) = x_c(nT_i), \qquad n = 0, \pm L, \pm 2L, \ldots, \tag{4.90}$$

as desired. The fact that $x_i[n] = x_c(nT_i)$ for all n follows from our frequency-domain argument.

4.6.3 Simple and Practical Interpolation Filters

Although ideal lowpass filters for interpolation cannot be implemented exactly, very good approximations can be designed using techniques to be discussed in Chapter 7. However, in some cases, very simple interpolation procedures are adequate or are forced on us by computational limitations. Since linear interpolation is often used (even though it is often not very accurate), it is worthwhile to examine this process within the general framework that we have just developed.

Linear interpolation corresponds to interpolation so that the samples between two original samples lie on a straight line connecting the two original sample values. Linear interpolation can be accomplished with the system of Figure 4.23 with the filter having the triangularly shaped impulse response

$$h_{\text{lin}}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise}, \end{cases} \tag{4.91}$$

as shown in Figure 4.25 for $L = 5$. With this filter, the interpolated output will be

$$x_{\text{lin}}[n] = \sum_{k=n-L+1}^{n+L-1} x_e[k] h_{\text{lin}}[n-k]. \tag{4.92}$$

Figure 4.26(a) depicts $x_e[k]$ (with the envelope of $h_{\text{lin}}[n-k]$ shown dashed for a particular value $n = 18$) and the corresponding output $x_{\text{lin}}[n]$ for the case $L = 5$. In this case, $x_{\text{lin}}[n]$ for $n = 18$ depends only on original samples $x[3]$ and $x[4]$. From this figure, we see that $x_{\text{lin}}[n]$ is identical to the sequence obtained by connecting the two original samples on either side of n by a straight line and then resampling at the $L - 1$ desired points in

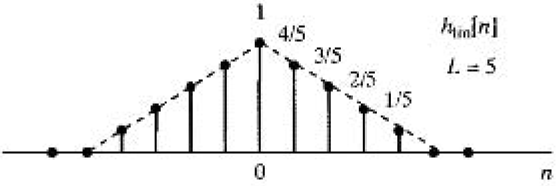
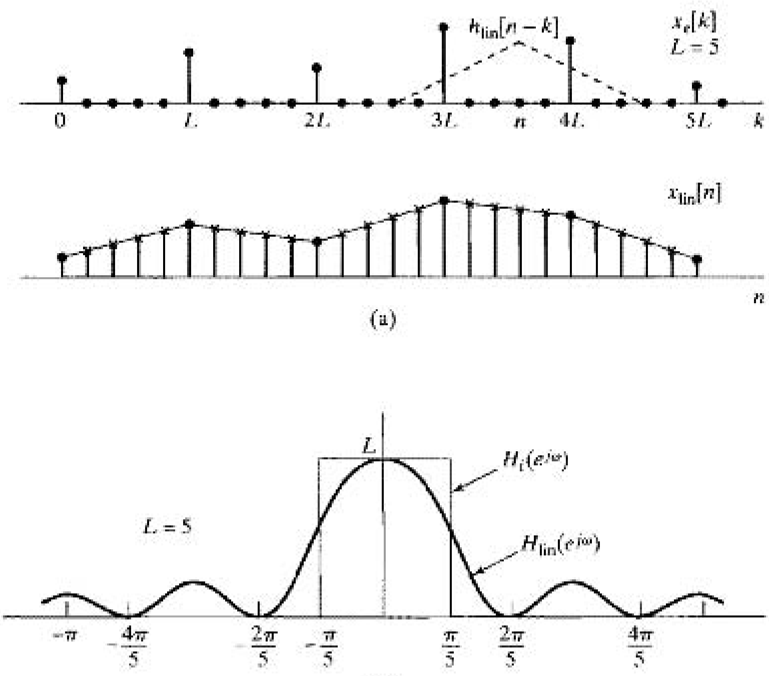


Figure 4.25 Impulse response for linear interpolation.



(a)

(b)

Figure 4.26 (a) Illustration of linear interpolation by filtering. (b) Frequency response of linear interpolator compared with ideal lowpass interpolation filter.

between. Also, note that the original sample values are preserved because $h_{\text{lin}}[0] = 1$ and $h_{\text{lin}}[n] = 0$ for $|n| \geq L$.

The nature of the distortion in the intervening samples can be better understood by comparing the frequency response of the linear interpolator with that of the ideal lowpass interpolator for a factor of L interpolation. It can be shown (see Problem 4.56) that

$$H_{\text{lin}}(e^{j\omega}) = \frac{1}{L}\left[\frac{\sin(\omega L/2)}{\sin(\omega/2)}\right]^2. \tag{4.93}$$

This function is plotted in Figure 4.26(b) for $L = 5$ together with the ideal lowpass interpolation filter. From the figure, we see that if the original signal is sampled at just the Nyquist rate, i.e., not oversampled, linear interpolation will not be very accurate, since the output of the filter will contain considerable energy in the band $\pi/L < |\omega| \leq \pi$ due to the frequency-scaled images of $X_c(j\Omega)$ at multiples of $2\pi/L$ that are not removed by the linear interpolation filter. However, if the original sampling rate is much higher than the Nyquist rate, then the linear interpolator will be more successful in removing these images because $H_{\text{lin}}(e^{j\omega})$ is small in a narrow region around these normalized frequencies, and at higher sampling rates, the increased frequency scaling causes the shifted copies of $X_c(j\Omega)$ to be more localized at multiples of $2\pi/L$. This is intuitively reasonable from a time domain perspective too, since, if the original sampling rate greatly exceeds the Nyquist rate, the signal will not vary significantly between samples, and thus, linear interpolation should be more accurate for oversampled signals.

Because of its double-sided infinite-length impulse response, the ideal bandlimited interpolator involves *all* of the original samples in the computation of each interpolated sample. In contrast, linear interpolation involves only *two* of the original samples in the computation of each interpolated sample. To better approximate ideal bandlimited interpolation, it is necessary to use filters with longer impulse responses. For this purpose FIR filters have many advantages. The impulse response $\tilde{h}_i[n]$ of an FIR filter for interpolation by a factor L usually is designed to have the following properties:

$$\tilde{h}_i[n] = 0 \qquad |n| \geq KL \tag{4.94a}$$

$$\tilde{h}_i[n] = \tilde{h}_i[-n] \qquad |n| \leq KL \tag{4.94b}$$

$$\tilde{h}_i[0] = 1 \qquad n = 0 \tag{4.94c}$$

$$\tilde{h}_i[n] = 0 \qquad n = \pm L, \pm 2L, \ldots, \pm KL. \tag{4.94d}$$

The interpolated output will therefore be

$$\tilde{x}_i[n] = \sum_{k=n-KL+1}^{n+KL-1} x_e[k]\tilde{h}_i[n-k]. \tag{4.95}$$

Note that the impulse response for linear interpolation satisfies Eqs. (4.94a)–(4.94d) with $K = 1$.

It is important to understand the motivation for the constraints of Eqs. (4.94a)–(4.94d). Equation (4.94a) states that the length of the FIR filter is $2KL - 1$ samples. Furthermore, this constraint ensures that only $2K$ original samples are involved in the computation of each sample of $\tilde{x}_i[n]$. This is because, even though $\tilde{h}_i[n]$ has $2KL - 1$ nonzero samples, the input $x_e[k]$ has only $2K$ nonzero samples within the region of support of $\tilde{h}_i[n-k]$ for any n between two of the original samples. Equation (4.94b) ensures that the filter will not introduce any phase shift into the interpolated samples since the corresponding frequency response is a real function of ω. The system could be made causal by introducing a delay of at least $KL - 1$ samples. In fact, the impulse response $\tilde{h}_i[n - KL]$ would yield an interpolated output delayed by KL samples, which would correspond to a delay of K samples at the original sampling rate. We might want to insert other amounts of delay so as to equalize delay among parts of a larger system that involves subsystems operating at different sampling rates. Finally, Eqs. (4.94c) and (4.94d) guarantee that the original signal samples will be preserved in the output, i.e.,

$$\tilde{x}_i[n] = x[n/L] \quad \text{at} \quad n = 0, \pm L, \pm 2L, \ldots. \tag{4.96}$$

Thus, if the sampling rate of $\tilde{x}_i[n]$ is subsequently reduced back to the original rate (with no intervening delay or a delay by a multiple of L samples) then $\tilde{x}_i[nL] = x[n]$; i.e., the original signal is recovered exactly. If this consistency is not required, the conditions of Eqs. (4.94c) and (4.94d) could be relaxed in the design of $\tilde{h}_i[n]$.

Figure 4.27 shows $x_e[k]$ and $\tilde{h}_i[n-k]$ with $K = 2$. The figure shows that each interpolated value depends on $2K = 4$ samples of the original input signal. Also note that computation of each interpolated sample requires only $2K$ multiplications and $2K - 1$ additions since there are always $L - 1$ zero samples in $x_e[k]$ between each of the original samples.

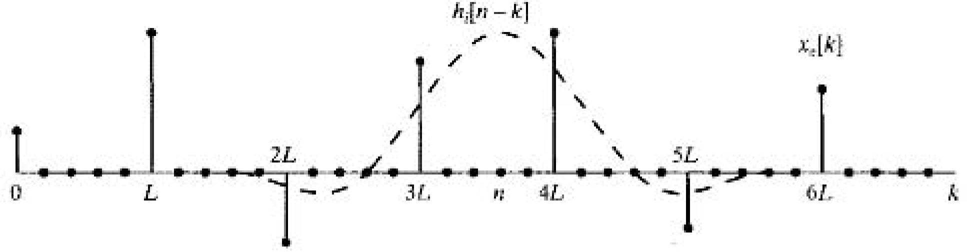


Figure 4.27 Illustration of interpolation involving $2K = 4$ samples when $L = 5$.

Interpolation is a much-studied problem in numerical analysis. Much of the development in this field is based on interpolation formulas that exactly interpolate polynomials of a certain degree. For example, the linear interpolator gives exact results for a constant signal and one whose samples vary along a straight line. Just as in the case of linear interpolation, higher-order Lagrange interpolation formulas (Schafer and Rabiner, 1973) and cubic spline interpolation formulas (Keys, 1981 and Unser, 2000) can be cast into our linear filtering framework to provide longer filters for interpolation. For example, the equation

$$\tilde{h}_i[n] = \begin{cases} (a+2)|n/L|^3 - (a+3)|n/L|^2 + 1 & 0 \le n \le L \\ a|n/L|^3 - 5|n/L|^2 + 8a|n/L| - 4a & L \le n \le 2L \\ 0 & \text{otherwise} \end{cases} \tag{4.97}$$

defines a convenient family of interpolation filter impulse responses that involve four ($K = 2$) original samples in the computation of each interpolated sample. Figure 4.28(a) shows the impulse response of a cubic filter for $a = -0.5$ and $L = 5$ along with the filter (dashed triangle) for linear ($K = 1$) interpolation. The corresponding frequency responses are shown in Figure 4.28(b) on a logarithmic amplitude (dB) scale. Note that the cubic filter has much wider regions around the frequencies $2\pi/L$ and $4\pi/L$ (0.4π and 0.8π in this case) but lower sidelobes than the linear interpolator, which is shown as the dashed line.

4.6.4 Changing the Sampling Rate by a Noninteger Factor

We have shown how to increase or decrease the sampling rate of a sequence by an integer factor. By combining decimation and interpolation, it is possible to change the sampling rate by a noninteger factor. Specifically, consider Figure 4.29(a), which

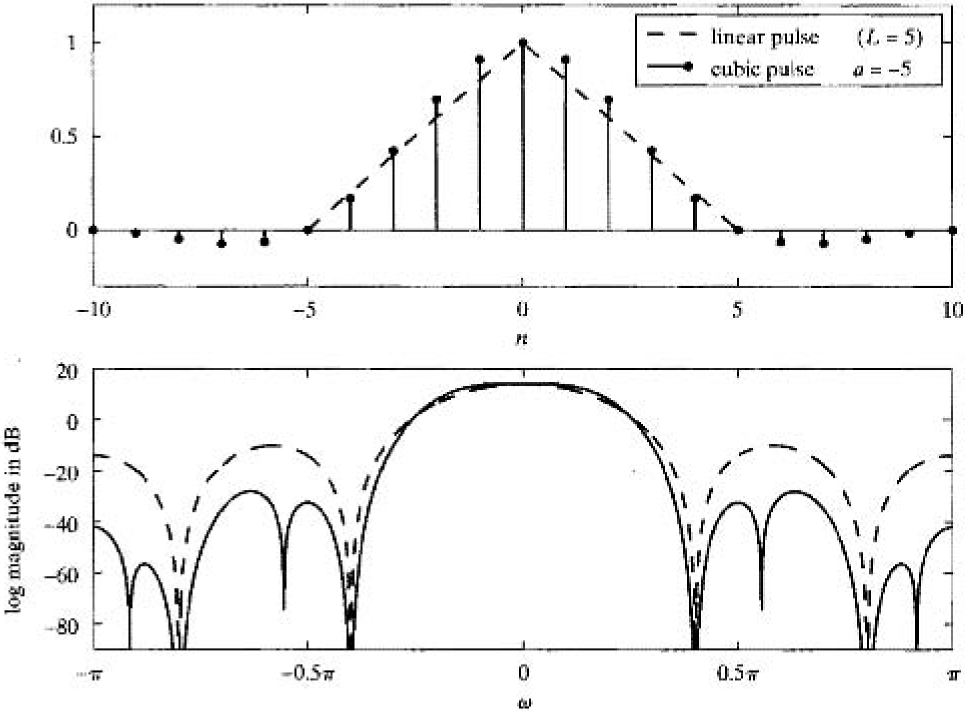


Figure 4.28 Impulse responses and frequency responses for linear and cubic interpolation.

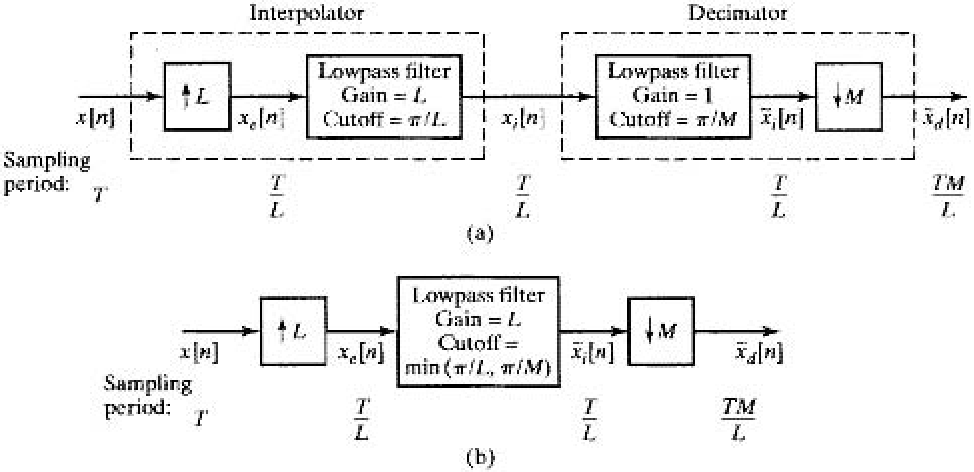


Figure 4.29 (a) System for changing the sampling rate by a noninteger factor. (b) Simplified system in which the decimation and interpolation filters are combined.

shows an interpolator that decreases the sampling period from T to T/L, followed by a decimator that increases the sampling period by M, producing an output sequence $\tilde{x}_d[n]$ that has an effective sampling period of (TM/L). By choosing L and M appropriately, we can approach arbitrarily close to any desired ratio of sampling periods. For example, if $L = 100$ and $M = 101$, then the effective sampling period is $1.01T$.

If $M > L$, there is a net increase in the sampling period (a decrease in the sampling rate), and if $M < L$, the opposite is true. Since the interpolation and decimation filters in Figure 4.29(a) are in cascade, they can be combined as shown in Figure 4.29(b) into one lowpass filter with gain L and cutoff equal to the minimum of π/L and π/M. If $M > L$, then π/M is the dominant cutoff frequency, and there is a net reduction in sampling rate. As pointed out in Section 4.6.1, if $x[n]$ was obtained by sampling at the Nyquist rate, the sequence $\tilde{x}_d[n]$ will correspond to a lowpass-filtered version of the original underlying bandlimited signal if we are to avoid aliasing. On the other hand, if $M < L$, then π/L is the dominant cutoff frequency, and there will be no need to further limit the bandwidth of the signal below the original Nyquist frequency.

Example 4.9 Sampling Rate Conversion by a Noninteger Rational Factor

Figure 4.30 illustrates sampling rate conversion by a rational factor. Suppose that a bandlimited signal with $X_c(j\Omega)$ as given in Figure 4.30(a) is sampled at the Nyquist rate; i.e., $2\pi/T = 2\Omega_N$. The resulting DTFT

$$X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right)$$

is plotted in Figure 4.30(b). An effective approach to changing the sampling period to $(3/2)T$, is to first interpolate by a factor $L = 2$ and then decimate by a factor of $M = 3$. Since this implies a net decrease in sampling rate, and the original signal was sampled at the Nyquist rate, we must incorporate additional bandlimiting to avoid aliasing.

Figure 4.30(c) shows the DTFT of the output of the $L = 2$ upsampler. If we were interested only in interpolating by a factor of 2, we could choose the lowpass filter to have a cutoff frequency of $\omega_c = \pi/2$ and a gain of $L = 2$. However, since the output of the filter will be decimated by $M = 3$, we must use a cutoff frequency of $\omega_c = \pi/3$, but the gain of the filter should still be 2 as in Figure 4.30(d). The Fourier transform $\tilde{X}_i(e^{j\omega})$ of the output of the lowpass filter is shown in Figure 4.30(e). The shaded regions indicate the part of the signal spectrum that is removed owing to the lower cutoff frequency for the interpolation filter. Finally, Figure 4.30(f) shows the DTFT of the output of the downsampler by $M = 3$. Note that the shaded regions show the aliasing that would have occurred if the cutoff frequency of the interpolation lowpass filter had been $\pi/2$ instead of $\pi/3$.

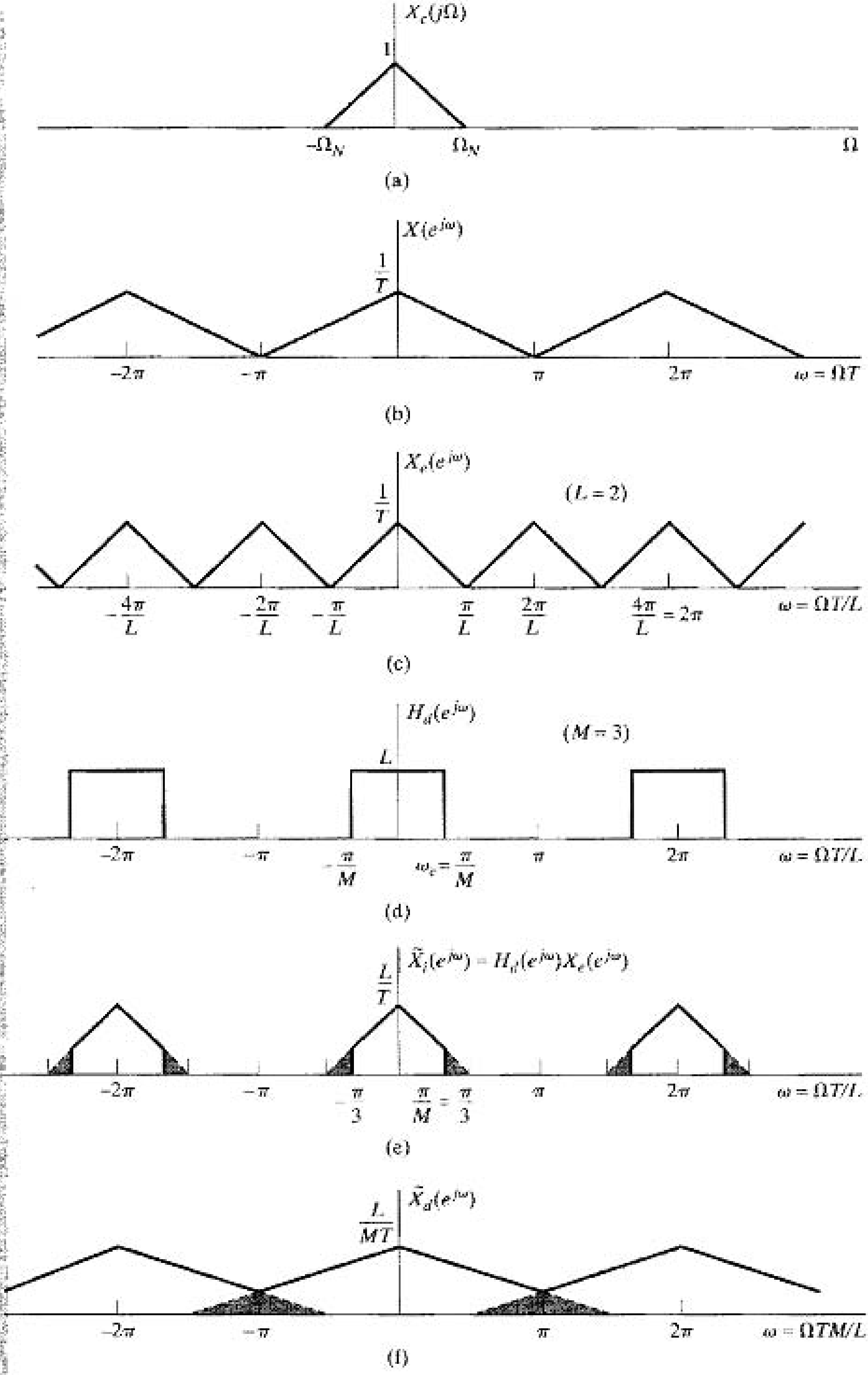


Figure 4.30 Illustration of changing the sampling rate by a noninteger factor.

4.7 MULTIRATE SIGNAL PROCESSING

As we have seen, it is possible to change the sampling rate of a discrete-time signal by a combination of interpolation and decimation. For example, if we want a new sampling period of $1.01T$, we can first interpolate by $L = 100$ using a lowpass filter that cuts off at $\omega_c = \pi/101$ and then decimate by $M = 101$. These large intermediate changes in sampling rate would require large amounts of computation for each output sample if we implement the filtering in a straightforward manner at the high intermediate sampling rate that is required. Fortunately, it is possible to greatly reduce the amount of computation required by taking advantage of some basic techniques broadly characterized as *multirate signal processing*. These multirate techniques refer in general to utilizing upsampling, downsampling, compressors, and expanders in a variety of ways to increase the efficiency of signal-processing systems. Besides their use in sampling rate conversion, they are exceedingly useful in A/D and D/A systems that exploit oversampling and noise shaping. Another important class of signal-processing algorithms that relies increasingly on multirate techniques is filter banks for the analysis and/or processing of signals.

Because of their widespread applicability, there is a large body of results on multirate signal processing techniques. In this section, we will focus on two basic results and show how a combination of these results can greatly improve the efficiency of sampling rate conversion. The first result is concerned with the interchange of filtering and downsampling or upsampling operations. The second is the polyphase decomposition. We shall also give two examples of how multirate techniques are used.

4.7.1 Interchange of Filtering with Compressor/Expander

First, we will derive two identities that aid in manipulating and understanding the operation of multirate systems. It is straightforward to show that the two systems in Figure 4.31 are equivalent. To see the equivalence, note that in Figure 4.31(b),

$$X_b(e^{j\omega}) = H(e^{j\omega M})X(e^{j\omega}), \tag{4.98}$$

and from Eq. (4.77),

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X_b(e^{j(\omega/M - 2\pi i/M)}). \tag{4.99}$$

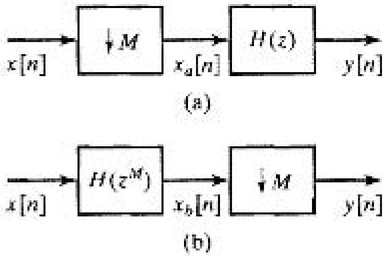


Figure 4.31 Two equivalent systems based on downsampling identities.

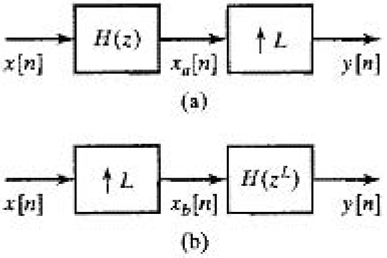


Figure 4.32 Two equivalent systems based on upsampling identities.

Substituting Eq. (4.98) into Eq. (4.99) gives

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j(\omega/M - 2\pi i/M)}\right) H(e^{j(\omega - 2\pi i)}). \tag{4.100}$$

Since $H(e^{j(\omega-2\pi i)}) = H(e^{j\omega})$, Eq. (4.100) reduces to

$$\begin{aligned} Y(e^{j\omega}) &= H(e^{j\omega})\frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j(\omega/M - 2\pi i/M)}\right) \\ &= H(e^{j\omega})X_a(e^{j\omega}), \end{aligned} \tag{4.101}$$

which corresponds to Figure 4.31(a). Therefore, the systems in Figure 4.31(a) and 4.31(b) are completely equivalent.

A similar identity applies to upsampling. Specifically, using Eq. (4.85) in Section 4.6.2, it is also straightforward to show the equivalence of the two systems in Figure 4.32. We have, from Eq. (4.85) and Figure 4.32(a),

$$\begin{aligned} Y(e^{j\omega}) &= X_a(e^{j\omega L}) \\ &= X(e^{j\omega L})H(e^{j\omega L}). \end{aligned} \tag{4.102}$$

Since, from Eq. (4.85),

$$X_b(e^{j\omega}) = X(e^{j\omega L}),$$

it follows that Eq. (4.102) is, equivalently,

$$Y(e^{j\omega}) = H(e^{j\omega L})X_b(e^{j\omega}),$$

which corresponds to Figure 4.32(b).

In summary, we have shown that the operations of linear filtering and downsampling or upsampling can be interchanged if we modify the linear filter.

4.7.2 Multistage Decimation and Interpolation

When decimation or interpolation ratios are large, it is necessary to use filters with very long impulse responses to achieve adequate approximations to the required lowpass filters. In such cases, there can be significant reduction in computation through the use of multistage decimation or interpolation. Figure 4.33(a) shows a two-stage decimation

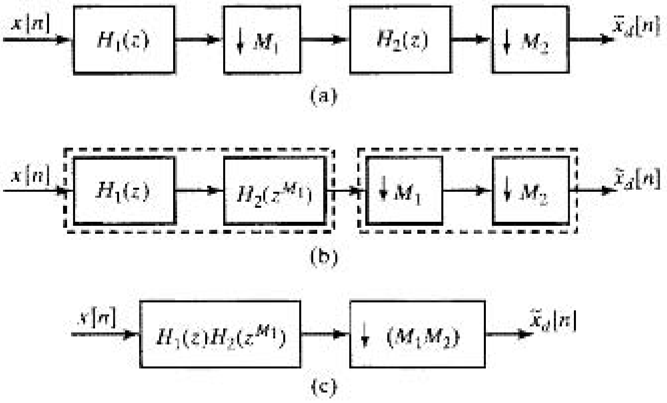


Figure 4.33 Multistage decimation: (a) Two-stage decimation system. (b) Modification of (a) using downsampling identity of Figure 4.31. (c) Equivalent one-stage decimation.

system where the overall decimation ratio is $M = M_1M_2$. In this case, two lowpass filters are required; $H_1(z)$ corresponds to a lowpass filter with nominal cutoff frequency π/M_1 and likewise, $H_2(z)$ has nominal cutoff frequency π/M_2. Note that for single-stage decimation, the required nominal cutoff frequency would be $\pi/M = \pi/(M_1M_2)$, which would be much smaller than that of either of the two filters. In Chapter 7 we will see that narrowband filters generally require high-order system functions to achieve sharp cutoff approximations to frequency-selective filter characteristics. Because of this effect, the two-stage implementation is often much more efficient than a single-stage implementation.

The single-stage system that is equivalent to Figure 4.33(a) can be derived using the downsampling identity of Figure 4.31. Figure 4.33(b) shows the result of replacing the system $H_2(z)$ and its preceding downsampler (by M_1) by the system $H_2(z^{M_1})$ followed by a downsampler by M_1. Figure 4.33(c) shows the result of combining the cascaded linear systems and cascaded downsamplers into corresponding single-stage systems. From this, we see that the system function of the equivalent single-stage lowpass filter is the product

$$H(z) = H_1(z)H_2(z^{M_1}). \tag{4.103}$$

This equation, which can be generalized to any number of stages if M has many factors, is a useful representation of the overall effective frequency response of the two-stage decimator. Since it explicitly shows the effects of the two filters, it can be used as an aid in designing effective multistage decimators that minimize computation. (See Crochiere and Rabiner, 1983, Vaidyanathan, 1993, and Bellanger, 2000.) The factorization in Eq. (4.103) has also been used directly to design lowpass filters (Neuvo et al., 1984). In this context, the filter with system function represented by Eq. (4.103) is called an *interpolated FIR filter*. This is because the corresponding impulse response can be seen to be the convolution of $h_1[n]$ with the second impulse response expanded by M_1; i.e.,

$$h[n] = h_1[n] * \sum_{k=-\infty}^{\infty} h_2[k]\delta[n - kM_1]. \tag{4.104}$$

The same multistage principles can be applied to interpolation, where, in this case, the upsampling identity of Figure 4.32 is used to relate the two-stage interpolator to an equivalent one-stage system. This is depicted in Figure 4.34.

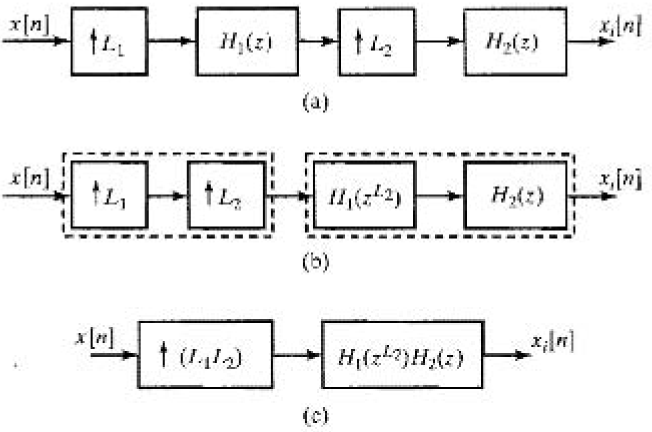


Figure 4.34 Multistage interpolation: (a) Two-stage interpolation system. (b) Modification of (a) using upsampling identity of Figure 4.32. (c) Equivalent one-stage interpolation.

4.7.3 Polyphase Decompositions

The polyphase decomposition of a sequence is obtained by representing it as a superposition of M subsequences, each consisting of every Mth value of successively delayed versions of the sequence. When this decomposition is applied to a filter impulse response, it can lead to efficient implementation structures for linear filters in several contexts. Specifically, consider an impulse response $h[n]$ that we decompose into M subsequences $h_k[n]$ with $k = 0, 1, \dots, M-1$ as follows:

$$h_k[n] = \begin{cases} h[n+k], & n = \text{integer multiple of } M, \\ 0, & \text{otherwise.} \end{cases} \tag{4.105}$$

By successively delaying these subsequences, we can reconstruct the original impulse response $h[n]$; i.e.,

$$h[n] = \sum_{k=0}^{M-1} h_k[n-k]. \tag{4.106}$$

This decomposition can be represented with the block diagram in Figure 4.35. If we create a chain of advance elements at the input and a chain of delay elements at the output, the block diagram in Figure 4.36 is equivalent to that of Figure 4.35. In the decomposition in Figures 4.35 and 4.36, the sequences $e_k[n]$ are

$$e_k[n] = h[nM+k] = h_k[nM] \tag{4.107}$$

and are referred to in general as the polyphase components of $h[n]$. There are several other ways to derive the polyphase components, and there are other ways to index them for notational convenience (Bellanger, 2000 and Vaidyanathan, 1993), but the definition in Eq. (4.107) is adequate for our purpose in this section.

Figures 4.35 and 4.36 are not realizations of the filter, but they show how the filter can be decomposed into M parallel filters. We see this by noting that Figures 4.35 and

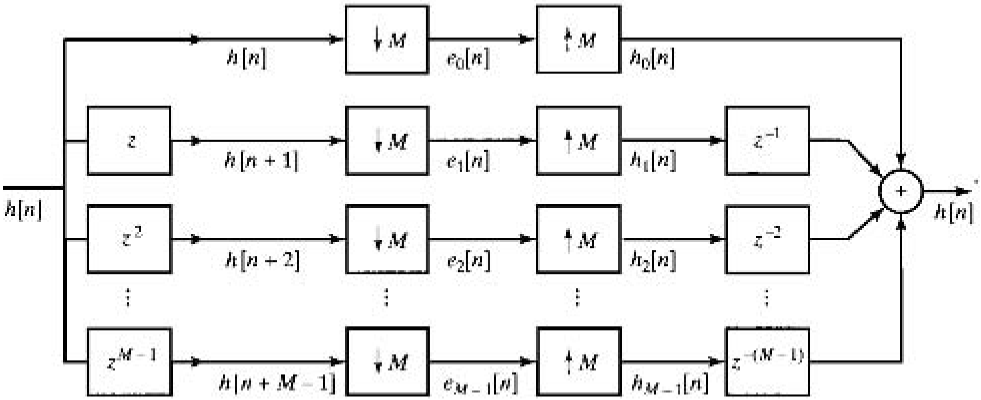


Figure 4.35 Polyphase decomposition of filter $h[n]$ using components $e_k[n]$.

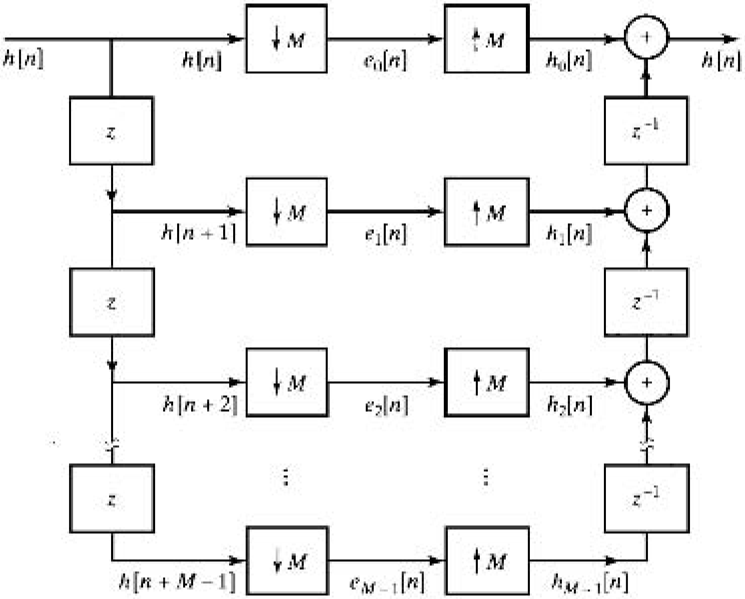


Figure 4.36 Polyphase decomposition of filter $h[n]$ using components $e_k[n]$ with chained delays.

4.36 show that, in the frequency or z-transform domain, the polyphase representation corresponds to expressing $H(z)$ as

$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}. \tag{4.108}$$

Equation (4.108) expresses the system function $H(z)$ as a sum of delayed polyphase component filters. For example, from Eq. (4.108), we obtain the filter structure shown in Figure 4.37.

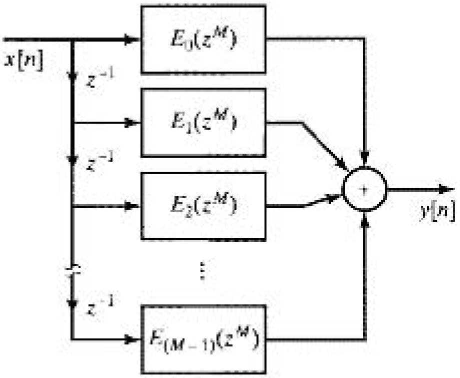


Figure 4.37 Realization structure based on polyphase decomposition of $h[n]$.

4.7.4 Polyphase Implementation of Decimation Filters

One of the important applications of the polyphase decomposition is in the implementation of filters whose output is then downsampled as indicated in Figure 4.38.

In the most straightforward implementation of Figure 4.38, the filter computes an output sample at each value of n, but then only one of every M output samples is retained. Intuitively, we might expect that it should be possible to obtain a more efficient implementation, which does not compute the samples that are thrown away.

To obtain a more efficient implementation, we can exploit a polyphase decomposition of the filter. Specifically, suppose we express $h[n]$ in polyphase form with polyphase components

$$e_k[n] = h[nM + k]. \tag{4.109}$$

From Eq. (4.108),

$$H(z) = \sum_{k=0}^{M-1} E_k(z^M)z^{-k}. \tag{4.110}$$

With this decomposition and the fact that downsampling commutes with addition, Figure 4.38 can be redrawn as shown in Figure 4.39. Applying the identity in Figure 4.31 to the system in Figure 4.39, we see that the latter then becomes the system shown in Figure 4.40.

To illustrate the advantage of Figure 4.40 compared with Figure 4.38, suppose that the input $x[n]$ is clocked at a rate of one sample per unit time and that $H(z)$ is an N-point FIR filter. In the straightforward implementation of Figure 4.38, we require N multiplications and $(N-1)$ additions per unit time. In the system of Figure 4.40, each of the filters $E_k(z)$ is of length N/M, and their inputs are clocked at a rate of 1 per M units of time. Consequently, each filter requires $\frac{1}{M}\left(\frac{N}{M}\right)$ multiplications per unit time and $\frac{1}{M}\left(\frac{N}{M}-1\right)$ additions per unit time. Since there are M polyphase components, the entire system therefore requires (N/M) multiplications and $\left(\frac{N}{M}-1\right)+(M-1)$ additions per unit time. Thus, we can achieve a significant savings for some values of M and N.

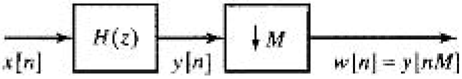


Figure 4.38 Decimation system.

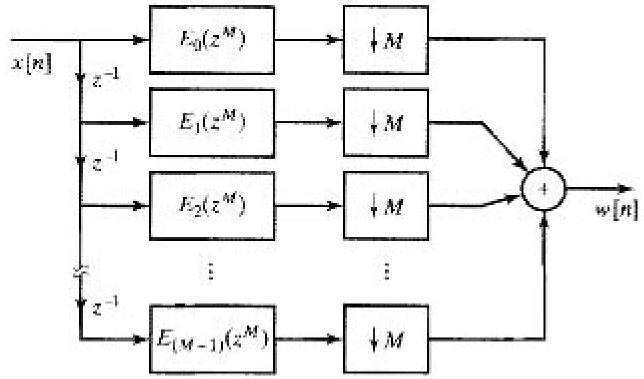


Figure 4.39 Implementation of decimation filter using polyphase decomposition.

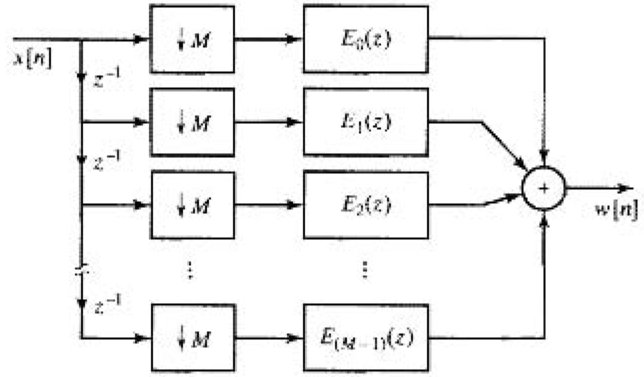


Figure 4.40 Implementation of decimation filter after applying the downsampling identity to the polyphase decomposition.

4.7.5 Polyphase Implementation of Interpolation Filters

A savings similar to that just discussed for decimation can be achieved by applying the polyphase decomposition to systems in which a filter is preceded by an upsampler as shown in Figure 4.41. Since only every Lth sample of $w[n]$ is nonzero, the most straightforward implementation of Figure 4.41 would involve multiplying filter coefficients by sequence values that are known to be zero. Intuitively, here again we would expect that a more efficient implementation was possible.

To implement the system in Figure 4.41 more efficiently, we again utilize the polyphase decomposition of $H(z)$. For example, we can express $H(z)$ as in the form of Eq. (4.110) and represent Figure 4.41 as shown in Figure 4.42. Applying the identity in Figure 4.32, we can rearrange Figure 4.42 as shown in Figure 4.43.

To illustrate the advantage of Figure 4.43 compared with Figure 4.41, we note that in Figure 4.41 if $x[n]$ is clocked at a rate of one sample per unit time, then $w[n]$ is clocked at a rate of L samples per unit time. If $H(z)$ is an FIR filter of length N, we then

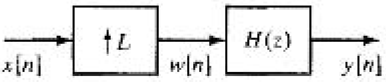


Figure 4.41 Interpolation system.

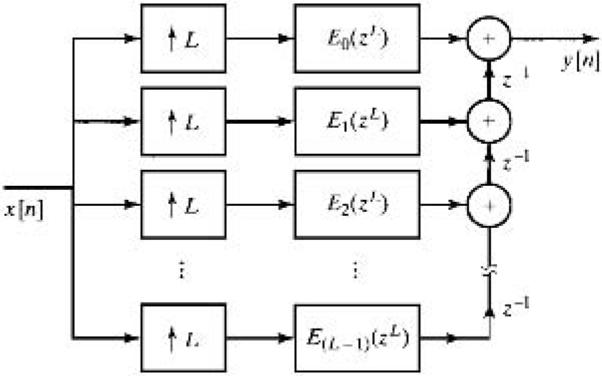


Figure 4.42 Implementation of interpolation filter using polyphase decomposition.

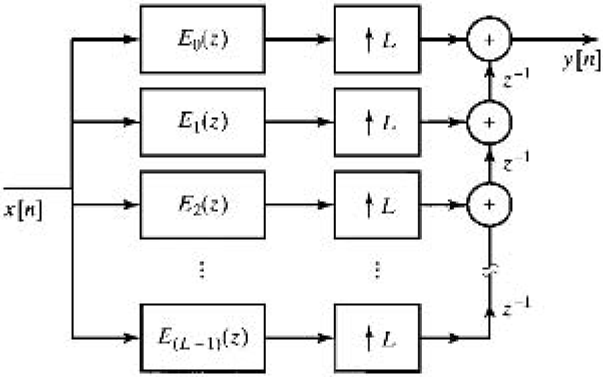


Figure 4.43 Implementation of interpolation filter after applying the upsampling identity to the polyphase decomposition.

require NL multiplications and $(NL - 1)$ additions per unit time. Figure 4.43, on the other hand, requires $L\,(N/L)$ multiplications and $L\,\left(\frac{N}{L} - 1\right)$ additions per unit time for the set of polyphase filters, plus $(L - 1)$ additions, to obtain $y[n]$. Thus, we again have the possibility of significant savings in computation for some values of L and N.

For both decimation and interpolation, gains in computational efficiency result from rearranging the operations so that the filtering is done at the low sampling rate. Combinations of interpolation and decimation systems for noninteger rate changes lead to significant savings when high intermediate rates are required.

4.7.6 Multirate Filter Banks

Polyphase structures for decimation and interpolation are widely used in filter banks for analysis and synthesis of audio and speech signals. For example, Figure 4.44 shows the block diagram of a two-channel analysis and synthesis filter bank commonly used in speech coding applications. The purpose of the analysis part of the system is to split the frequency spectrum of the input $x[n]$ into a lowpass band represented by the downsampled signal $v_0[n]$ and a highpass band represented by $v_1[n]$. In speech and audio coding applications, the channel signals are quantized for transmission and/or storage. Since the original band is nominally split into two equal parts of width $\pi/2$ radians, the

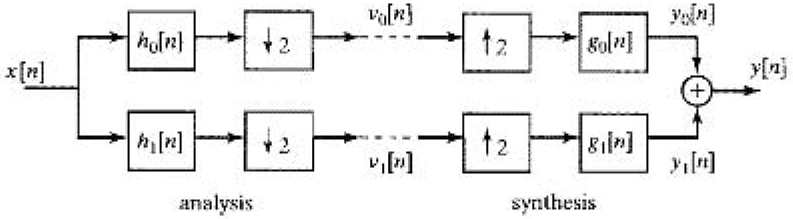


Figure 4.44 Two-channel analysis and synthesis filter bank.

sampling rates of the filter outputs can be 1/2 that of the input so that the total number of samples per second remains the same.[1] Note that downsampling the output of the lowpass filter expands the low-frequency band to the entire range $|\omega| < \pi$. On the other hand, downsampling the output of the highpass filter down-shifts the high-frequency band and expands it to the full range $|\omega| < \pi$.

The decomposition requires that $h_0[n]$ and $h_1[n]$ be impulse responses of lowpass and highpass filters respectively. A common approach is to derive the highpass filter from the lowpass filter by $h_1[n] = e^{j\pi n}h_0[n]$. This implies that $H_1(e^{j\omega}) = H_0(e^{j(\omega-\pi)})$ so that if $H_0(e^{j\omega})$ is a lowpass filter with nominal passband $0 \le |\omega| \le \pi/2$, then $H_1(e^{j\omega})$ will be a highpass filter with nominal passband $\pi/2 < |\omega| \le \pi$. The purpose of the righthand (synthesis) part of Figure 4.44 is to reconstitute an approximation to $x[n]$ from the two channel signals $v_0[n]$ and $v_1[n]$. This is achieved by upsampling both signals and passing them through a lowpass filter $g_0[n]$ and highpass filter $g_1[n]$ respectively. The resulting interpolated signals are added to produce the full-band output signal $y[n]$ sampled at the input sampling rate.

Applying the frequency-domain results for downsampling and upsampling to the system in Figure 4.44 leads to the following result:

$$Y(e^{j\omega}) = \frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j\omega}) + G_1(e^{j\omega})H_1(e^{j\omega})\right]X(e^{j\omega}) \tag{4.111a}$$

$$+\frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j(\omega-\pi)}) + G_1(e^{j\omega})H_1(e^{j(\omega-\pi)})\right]X(e^{j(\omega-\pi)}). \tag{4.111b}$$

If the analysis and synthesis filters are ideal so that they exactly split the band $0 \le |\omega| \le \pi$ into two equal segments without overlapping, then it is straightforward to verify that $Y(e^{j\omega}) = X(e^{j\omega})$; i.e., the synthesis filter bank reconstructs the input signal exactly. However, perfect or nearly perfect reconstruction also can be achieved with nonideal filters for which aliasing will occur in the downsampling operations of the analysis filter bank. To see this, note that the second term in the expression for $Y(e^{j\omega})$ (line labeled Eq. (4.111b)), which represents potential aliasing distortion from the downsampling operation, can be eliminated by choosing the filters such that

$$G_0(e^{j\omega})H_0(e^{j(\omega-\pi)}) + G_1(e^{j\omega})H_1(e^{j(\omega-\pi)}) = 0. \tag{4.112}$$

[1] Filter banks that conserve the total number of samples per second are termed *maximally decimated* filter banks.

This condition is called the *alias cancellation condition*. One set of conditions that satisfy Eq. (4.112) is

$$h_1[n] = e^{j\pi n}h_0[n] \Longleftrightarrow H_1(e^{j\omega}) = H_0(e^{j(\omega-\pi)}) \tag{4.113a}$$

$$g_0[n] = 2h_0[n] \Longleftrightarrow G_0(e^{j\omega}) = 2H_0(e^{j\omega}) \tag{4.113b}$$

$$g_1[n] = -2h_1[n] \Longleftrightarrow G_1(e^{j\omega}) = -2H_0(e^{j(\omega-\pi)}). \tag{4.113c}$$

The filters $h_0[n]$ and $h_1[n]$ are termed *quadrature mirror filters* since Eq. (4.113a) imposes mirror symmetry about $\omega = \pi/2$. Substituting these relations into Eq. (4.111a) leads to the relation

$$Y(e^{j\omega}) = \left[H_0^2(e^{j\omega}) - H_0^2(e^{j(\omega-\pi)})\right] X(e^{j\omega}), \tag{4.114}$$

from which it follows that perfect reconstruction (with possible delay of M samples) requires

$$H_0^2(e^{j\omega}) - H_0^2(e^{j(\omega-\pi)}) = e^{-j\omega M}. \tag{4.115}$$

It can be shown (Vaidyanathan, 1993) that the only computationally realizable filters satisfying Eq. (4.115) exactly are systems with impulse responses of the form $h_0[n] = c_0\delta[n-2n_0] + c_1\delta[n-2n_1-1]$ where n_0 and n_1 are arbitrarily chosen integers and $c_0c_1 = \frac{1}{4}$. Such systems cannot provide the sharp frequency selective properties needed in speech and audio coding applications, but to illustrate that such systems can achieve exact reconstruction, consider the simple two-point moving average lowpass filter

$$h_0[n] = \frac{1}{2}(\delta[n] + \delta[n-1]), \tag{4.116a}$$

which has frequency response

$$H_0(e^{j\omega}) = \cos(\omega/2)e^{-j\omega/2}. \tag{4.116b}$$

For this filter, $Y(e^{j\omega}) = e^{-j\omega}X(e^{j\omega})$ as can be verified by substituting Eq. (4.116b) into Eq. (4.114).

Either FIR or IIR filters can be used in the analysis/synthesis system of Figure 4.44 with the filters related as in Eq. (4.113a)–(4.113c) to provide nearly perfect reconstruction. The design of such filters is based on finding a design for $H_0(e^{j\omega})$ that is an acceptable lowpass filter approximation while satisfying Eq. (4.115) to within an acceptable approximation error. A set of such filters and an algorithm for their design was given by Johnston (1980). Smith and Barnwell (1984) and Mintzer (1985) showed that perfect reconstruction is possible with the two-channel filter bank of Figure 4.44 if the filters have a different relationship to one another than is specified by Eq. (4.113a)-(4.113c). The different relationship leads to filters called conjugate quadrature filters (CQF).

Polyphase techniques can be employed to save computation in the implementation of the analysis/synthesis system of Figure 4.44. Applying the polyphase downsampling result depicted in Figure 4.40 to the two channels leads to the block diagram in Figure 4.45(a), where

$$e_{00}[n] = h_0[2n] \tag{4.117a}$$

$$e_{01}[n] = h_0[2n+1] \tag{4.117b}$$

$$e_{10}[n] = h_1[2n] = e^{j2\pi n}h_0[2n] = e_{00}[n] \tag{4.117c}$$

$$e_{11}[n] = h_1[2n+1] = e^{j2\pi n}e^{j\pi}h_0[2n+1] = -e_{01}[n]. \tag{4.117d}$$

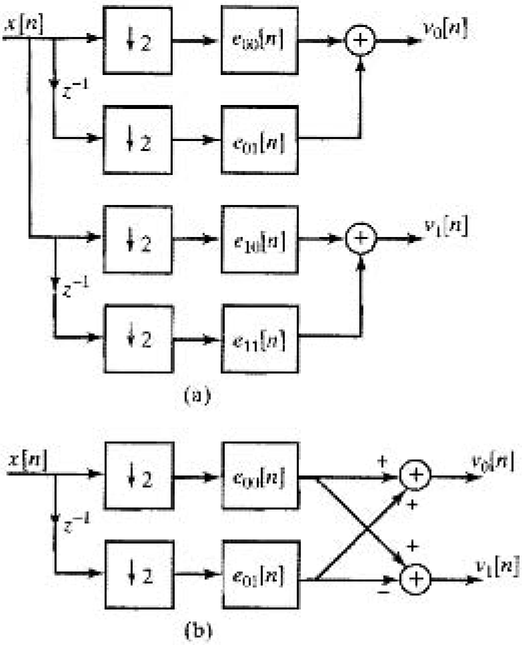


Figure 4.45 Polyphase representation of the two-channel analysis filter bank of Figure 4.44.

Equations (4.117c) and (4.117d) show that the polyphase filters for $h_1[n]$ are the same (except for sign) as those for $h_0[n]$. Therefore, only one set, $e_{00}[n]$ and $e_{01}[n]$ need be implemented. Figure 4.45(b) shows how both $v_0[n]$ and $v_1[n]$ can be formed from the outputs of the two polyphase filters. This equivalent structure, which requires only half the computation of Figure 4.45(a), is, of course, owing entirely to the simple relation between the two filters.

The polyphase technique can likewise be applied to the synthesis filter bank, by recognizing that the two interpolators can be replaced by their polyphase implementations and then the polyphase structures can be combined because $g_1[n] = -e^{j\pi n}g_0[n] = -e^{j\pi n}2h_0[n]$. The resulting polyphase synthesis system can be represented in terms of the polyphase filters $f_{00}[n] = 2e_{00}[n]$ and $f_{01}[n] = 2e_{01}[n]$ as in Figure 4.46. As in the case of the analysis filter bank, the synthesis polyphase filters can be shared between the two channels thereby halving the computation.

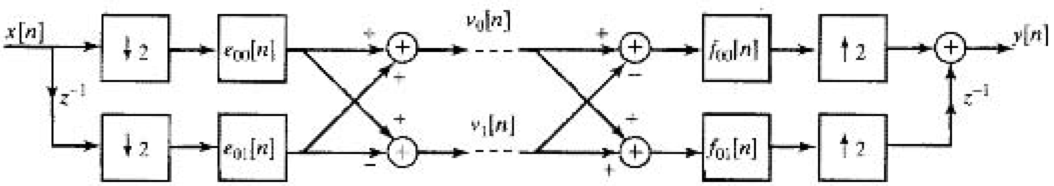


Figure 4.46 Polyphase representation of the two-channel analysis and synthesis filter bank of Figure 4.44.

This two-band analysis/synthesis system can be generalized to N equal width channels to obtain a finer decomposition of the spectrum. Such systems are used in audio coding, where they facilitate exploitation of the characteristics of human auditory perception in compression of the digital information rate. (See MPEG audio coding standard and Spanias, Painter, and Atti, 2007.) Also, the two-band system can be incorporated into a tree structure to obtain an analysis/synthesis system with either uniformly or nonuniformly spaced channels. When the CQF filters of Smith and Barnwell, and Mintzer are used, exact reconstruction is possible, and the resulting analysis synthesis system is essentially the discrete wavelet transform. (See Vaidyanathan, 1993 and Burrus, Gopinath and Guo, 1997.)

DIGITAL PROCESSING OF ANALOG SIGNALS

So far, our discussions of the representation of continuous-time signals by discrete-time signals have focused on idealized models of periodic sampling and bandlimited interpolation. We have formalized those discussions in terms of an idealized sampling system that we have called the *ideal continuous-to-discrete* (C/D) *converter* and an idealized bandlimited interpolator system called the *ideal discrete-to-continuous* (D/C) *converter*. These idealized conversion systems allow us to concentrate on the essential mathematical details of the relationship between a bandlimited signal and its samples. For example, in Section 4.4 we used the idealized C/D and D/C conversion systems to show that LTI discrete-time systems can be used in the configuration of Figure 4.47(a) to implement LTI continuous-time systems if the input is bandlimited and the sampling rate is at or above the Nyquist rate. In a practical setting, continuous-time signals are not precisely bandlimited, ideal filters cannot be realized, and the ideal C/D and D/C converters can only be approximated by devices that are called analog-to-digital (A/D) and digital-to-analog (D/A) converters, respectively. The block diagram of Figure 4.47(b) shows a

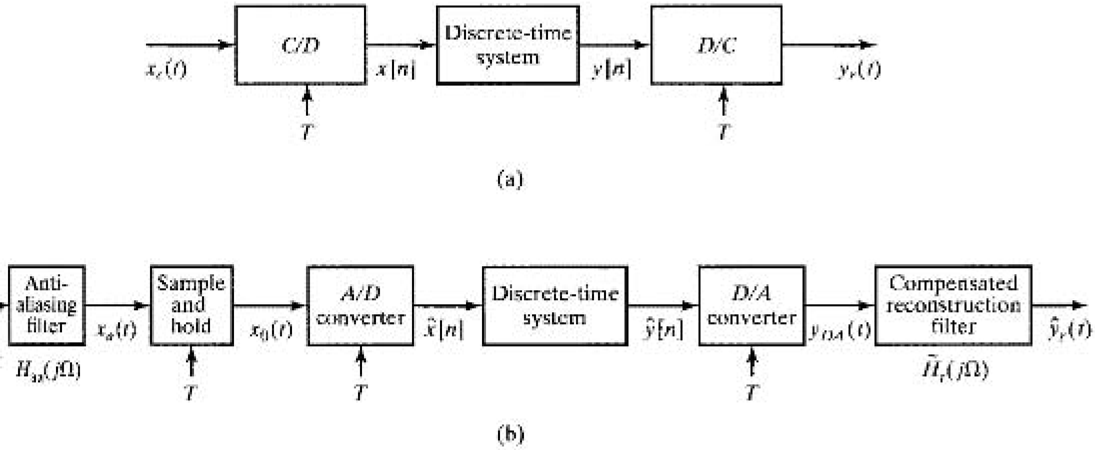


Figure 4.47 (a) Discrete-time filtering of continuous-time signals. (b) Digital processing of analog signals.

more realistic model for digital processing of continuous-time (analog) signals. In this section, we will examine some of the considerations introduced by each of the components of the system in Figure 4.47(b).

4.8.1 Prefiltering to Avoid Aliasing

In processing analog signals using discrete-time systems, it is generally desirable to minimize the sampling rate. This is because the amount of arithmetic processing required to implement the system is proportional to the number of samples to be processed. If the input is not bandlimited or if the Nyquist frequency of the input is too high, prefiltering may be necessary. An example of such a situation occurs in processing speech signals, where often only the low-frequency band up to about 3 to 4 kHz is required for intelligibility, even though the speech signal may have significant frequency content in the 4 kHz to 20 kHz range. Also, even if the signal is naturally bandlimited, wideband additive noise may fill in the higher frequency range, and as a result of sampling, these noise components would be aliased into the low-frequency band. If we wish to avoid aliasing, the input signal must be forced to be bandlimited to frequencies below one-half the desired sampling rate. This can be accomplished by lowpass filtering the continuous-time signal prior to C/D conversion, as shown in Figure 4.48. In this context, the lowpass filter that precedes the C/D converter is called an *antialiasing filter*. Ideally, the frequency response of the antialiasing filter would be

$$H_{\text{aa}}(j\Omega) = \begin{cases} 1, & |\Omega| < \Omega_c \leq \pi/T, \\ 0, & |\Omega| \geq \Omega_c. \end{cases} \tag{4.118}$$

From the discussion of Section 4.4.1, it follows that the overall system, from the output of the antialiasing filter $x_a(t)$ to the output $y_r(t)$, will always behave as an LTI system, since the input to the C/D converter, $x_a(t)$, is forced by the antialiasing filter to be bandlimited to frequencies below π/T radians/s. Thus, the overall effective frequency response of Figure 4.48 will be the product of $H_{\text{aa}}(j\Omega)$ and the effective frequency response from $x_a(t)$ to $y_r(t)$. Combining Eqs. (4.118) and (4.38) gives

$$H_{\text{eff}}(j\Omega) = \begin{cases} H(e^{j\Omega T}), & |\Omega| < \Omega_c, \\ 0, & |\Omega| \geq \Omega_c. \end{cases} \tag{4.119}$$

Thus, for an ideal lowpass antialiasing filter, the system of Figure 4.48 behaves as an LTI system with frequency response given by Eq. (4.119), even when $X_c(j\Omega)$ is not bandlimited. In practice, the frequency response $H_{\text{aa}}(j\Omega)$ cannot be ideally bandlimited, but $H_{\text{aa}}(j\Omega)$ can be made small for $|\Omega| > \pi/T$ so that aliasing is minimized. In this case, the overall frequency response of the system in Figure 4.48 should be approximately

$$H_{\text{eff}}(j\Omega) \approx H_{\text{aa}}(j\Omega)H(e^{j\Omega T}). \tag{4.120}$$

To achieve a negligibly small frequency response above π/T, it would be necessary for $H_{\text{aa}}(j\Omega)$ to begin to "roll off," i.e., begin to introduce attenuation, at frequencies below π/T. Eq. (4.120) suggests that the roll-off of the antialiasing filter (and other LTI distortions to be discussed later) could be at least partially compensated for by taking them into account in the design of the discrete-time system. This is illustrated in Problem 4.62.

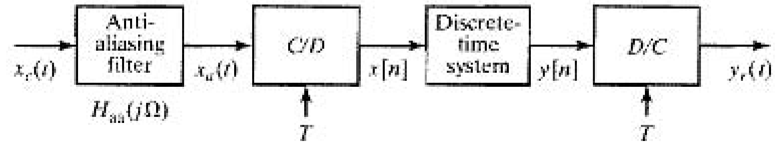


Figure 4.48 Use of prefiltering to avoid aliasing.

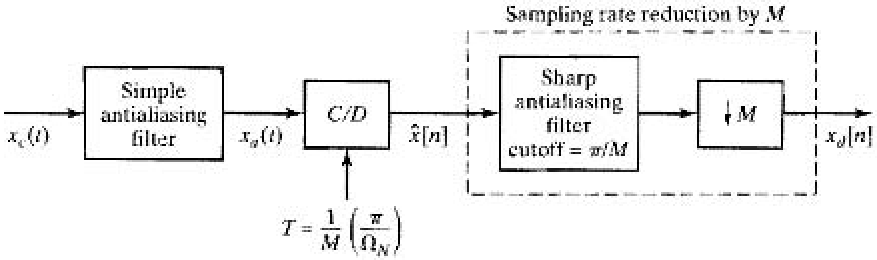

Sampling rate reduction by M
$x_c(t)$
Simple antialiasing filter
$x_a(t)$
C/D
$T = \frac{1}{M}\left(\frac{\pi}{\Omega_N}\right)$
$\hat{x}[n]$
Sharp antialiasing filter cutoff = π/M
$\downarrow M$
$x_d[n]$

Figure 4.49 Using oversampled A/D conversion to simplify a continuous-time antialiasing filter.

The preceding discussion requires sharp-cutoff antialiasing filters. Such sharp-cutoff analog filters can be realized using active networks and integrated circuits. However, in applications involving powerful, but inexpensive, digital processors, these continuous-time filters may account for a major part of the cost of a system for discrete-time processing of analog signals. Sharp-cutoff filters are difficult and expensive to implement, and if the system is to operate with a variable sampling rate, adjustable filters would be required. Furthermore, sharp-cutoff analog filters generally have a highly nonlinear phase response, particularly at the passband edge. Thus, it is desirable for several reasons to eliminate the continuous-time filters or simplify the requirements on them.

One approach is depicted in Figure 4.49. With Ω_N denoting the highest frequency component to eventually be retained after the antialiasing filtering is completed, we first apply a very simple antialiasing filter that has a gradual cutoff with significant attenuation at $M\Omega_N$. Next, implement the C/D conversion at a sampling rate much higher than $2\Omega_N$, e.g., at $2M\Omega_N$. After that, sampling rate reduction by a factor of M that includes sharp antialiasing filtering is implemented in the discrete-time domain. Subsequent discrete-time processing can then be done at the low sampling rate to minimize computation.

This use of oversampling followed by sampling rate conversion is illustrated in Figure 4.50. Figure 4.50(a) shows the Fourier transform of a signal that occupies the band $|\Omega| < \Omega_N$, plus the Fourier transform of what might correspond to high-frequency "noise" or unwanted components that we eventually want to eliminate with the antialiasing filter. Also shown (dotted line) is the frequency response of an antialiasing filter that does not cut off sharply but gradually falls to zero at frequencies above the frequency Ω_N. Figure 4.50(b) shows the Fourier transform of the output of this filter. If the signal $x_a(t)$ is sampled with period T such that $(2\pi/T - \Omega_c) \geq \Omega_N$, then the DTFT

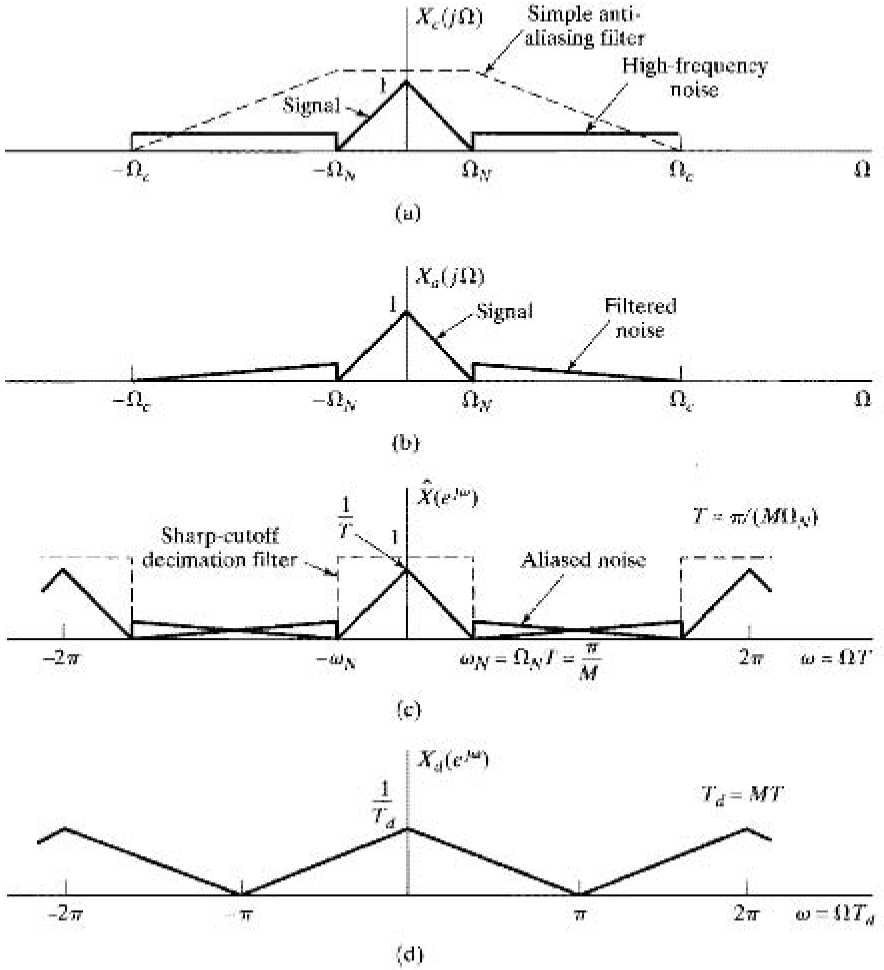


Figure 4.50 Use of oversampling followed by decimation in C/D conversion.

of the sequence $\hat{x}[n]$ will be as shown in Figure 4.50(c). Note that the "noise" will be aliased, but aliasing will not affect the signal band $|\omega| < \omega_N = \Omega_N T$. Now, if T and T_d are chosen so that $T_d = MT$ and $\pi/T_d = \Omega_N$, then $\hat{x}[n]$ can be filtered by a sharp-cutoff discrete-time filter (shown idealized in Figure 4.50(c)) with unity gain and cutoff frequency π/M. The output of the discrete-time filter can be downsampled by M to obtain the sampled sequence $x_d[n]$ whose Fourier transform is shown in Figure 4.50(d). Thus, all the sharp-cutoff filtering has been done by a discrete-time system, and only nominal continuous-time filtering is required. Since discrete-time FIR filters can have an exactly linear phase, it is possible using this oversampling approach to implement antialiasing filtering with virtually no phase distortion. This can be a significant advantage in situations where it is critical to preserve not only the frequency spectrum, but the waveshape as well.

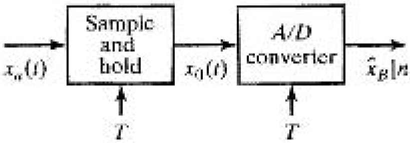


Figure 4.51 Physical configuration for A/D conversion.

4.8.2 A/D Conversion

An ideal C/D converter converts a continuous-time signal into a discrete-time signal, where each sample is known with infinite precision. As an approximation to this for digital signal processing, the system of Figure 4.51 converts a continuous-time (analog) signal into a digital signal, i.e., a sequence of finite-precision or quantized samples. The two systems in Figure 4.51 are available as physical devices. The A/D converter is a physical device that converts a voltage or current amplitude at its input into a binary code representing a quantized amplitude value closest to the amplitude of the input. Under the control of an external clock, the A/D converter can be caused to start and complete an A/D conversion every T seconds. However, the conversion is not instantaneous, and for this reason, a high-performance A/D system typically includes a sample-and-hold, as in Figure 4.51. The ideal sample-and-hold system is the system whose output is

$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n]h_0(t - nT), \tag{4.121}$$

where $x[n] = x_a(nT)$ are the ideal samples of $x_a(t)$ and $h_0(t)$ is the impulse response of the zero-order-hold system, i.e.,

$$h_0(t) = \begin{cases} 1, & 0 < t < T, \\ 0, & \text{otherwise.} \end{cases} \tag{4.122}$$

If we note that Eq. (4.121) has the equivalent form

$$x_0(t) = h_0(t) * \sum_{n=-\infty}^{\infty} x_a(nT)\delta(t - nT), \tag{4.123}$$

we see that the ideal sample-and-hold is equivalent to impulse train modulation followed by linear filtering with the zero-order-hold system, as depicted in Figure 4.52(a). The relationship between the Fourier transform of $x_0(t)$ and the Fourier transform of $x_a(t)$ can be worked out following the style of analysis of Section 4.2, and we will do a similar analysis when we discuss the D/A converter. However, the analysis is unnecessary at this point, since everything we need to know about the behavior of the system can be seen from the time-domain expression. Specifically, the output of the zero-order hold is a staircase waveform where the sample values are held constant during the sampling period of T seconds. This is illustrated in Figure 4.52(b). Physical sample-and-hold circuits are designed to sample $x_a(t)$ as nearly instantaneously as possible and to hold the sample value as nearly constant as possible until the next sample is taken. The purpose of this is to provide the constant input voltage (or current) required by the A/D converter. The details of the wide variety of A/D conversion processes and the details of sample-and-hold and A/D circuit implementations are outside the scope of this book. Many practical issues arise in obtaining a sample-and-hold that samples

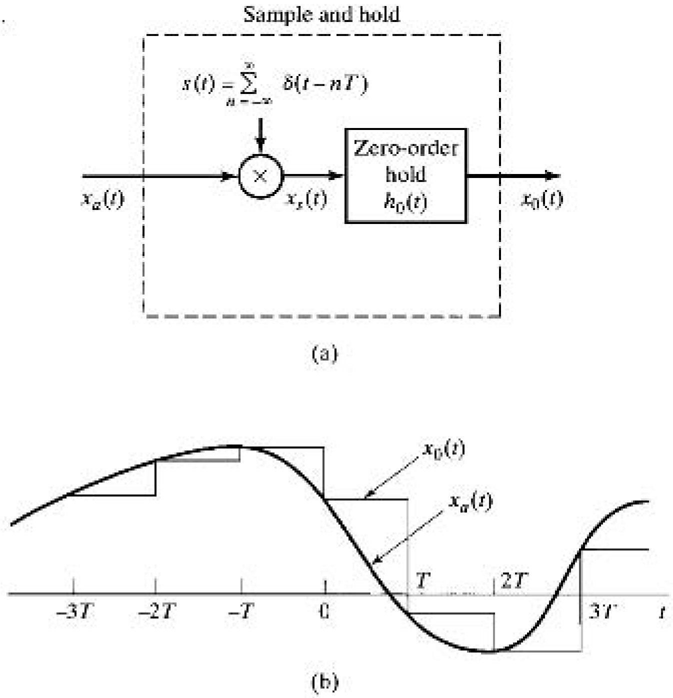


Figure 4.52 (a) Representation of an ideal sample-and-hold. (b) Representative input and output signals for the sample-and-hold.

quickly and holds the sample value constant with no decay or "glitches." Likewise, many practical concerns dictate the speed and accuracy of conversion of A/D converter circuits. Such questions are considered in Hnatek (1988) and Schmid (1976), and details of the performance of specific products are available in manufacturers' specification and data sheets. Our concern in this section is the analysis of the quantization effects in A/D conversion.

Since the purpose of the sample-and-hold in Figure 4.51 is to implement ideal sampling and to hold the sample value for quantization by the A/D converter, we can represent the system of Figure 4.51 by the system of Figure 4.53, where the ideal C/D converter represents the sampling performed by the sample-and-hold and, as we will describe later, the quantizer and coder together represent the operation of the A/D converter.

The quantizer is a nonlinear system whose purpose is to transform the input sample $x[n]$ into one of a finite set of prescribed values. We represent this operation as

$$\hat{x}[n] = Q(x[n]) \tag{4.124}$$

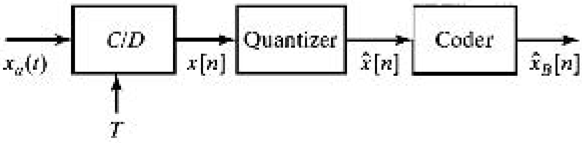


Figure 4.53 Conceptual representation of the system in Figure 4.51.

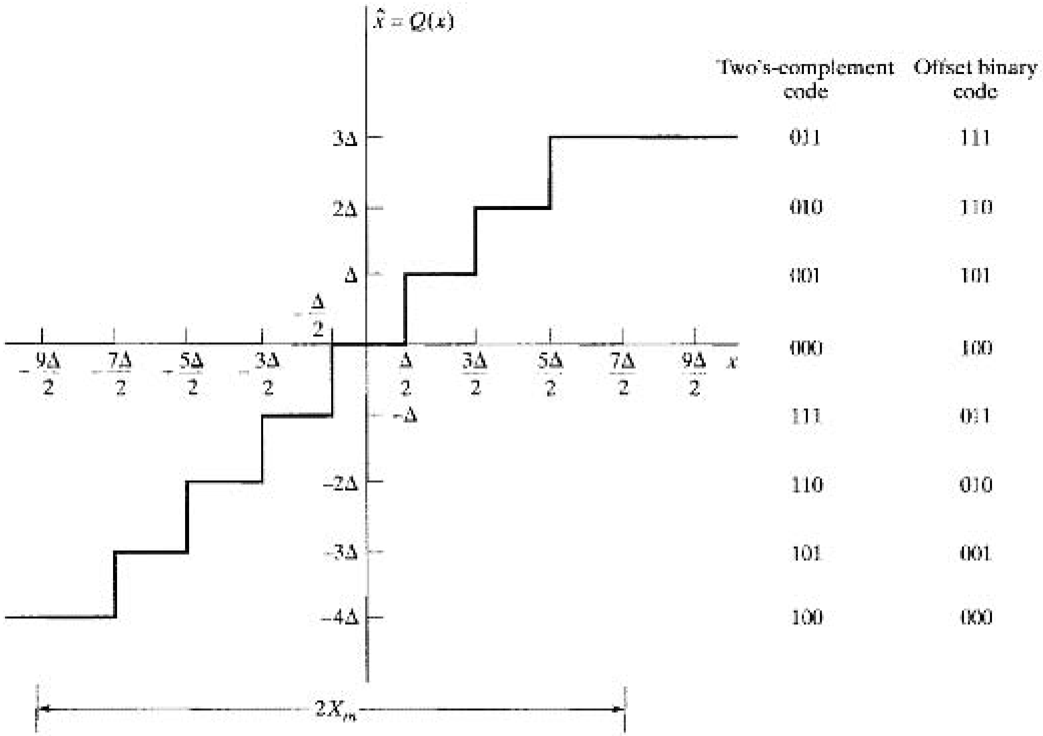


Figure 4.54 Typical quantizer for A/D conversion.

and refer to $\hat{x}[n]$ as the quantized sample. Quantizers can be defined with either uniformly or nonuniformly spaced quantization levels; however, when numerical calculations are to be done on the samples, the quantization steps usually are uniform. Figure 4.54 shows a typical uniform quantizer characteristic,[2] in which the sample values are *rounded* to the nearest quantization level.

Several features of Figure 4.54 should be emphasized. First, note that this quantizer would be appropriate for a signal whose samples are both positive and negative (bipolar). If it is known that the input samples are always positive (or negative), then a different distribution of the quantization levels would be appropriate. Next, observe that the quantizer of Figure 4.54 has an even number of quantization levels. With an even number of levels, it is not possible to have a quantization level at zero amplitude and also have an equal number of positive and negative quantization levels. Generally, the number of quantization levels will be a power of two, but the number will be much greater than eight, so this difference is usually inconsequential.

Figure 4.54 also depicts coding of the quantization levels. Since there are eight quantization levels, we can label them by a binary code of 3 bits. (In general, 2^{B+1} levels can be coded with a $(B+1)$-bit binary code.) In principle, any assignment of symbols

[2] Such quantizers are also called *linear* quantizers because of the linear progression of quantization steps.

can be used, and many binary coding schemes exist, each with its own advantages and disadvantages, depending on the application. For example, the right-hand column of binary numbers in Figure 4.54 illustrates the *offset binary* coding scheme, in which the binary symbols are assigned in numeric order, starting with the most negative quantization level. However, in digital signal processing, we generally wish to use a binary code that permits us to do arithmetic directly with the code words as scaled representations of the quantized samples.

The left-hand column in Figure 4.54 shows an assignment according to the two's complement binary number system. This system for representing signed numbers is used in most computers and microprocessors; thus, it is perhaps the most convenient labeling of the quantization levels. Note, incidentally, that the offset binary code can be converted to two's-complement code simply by complementing the most significant bit.

In the two's-complement system, the leftmost, or most significant, bit is considered as the sign bit, and we take the remaining bits as representing either binary integers or fractions. We will assume the latter; i.e., we assume a binary fraction point between the two most significant bits. Then, for the two's-complement interpretation, the binary symbols have the following meaning for $B = 2$:

Binary symbol	Numeric value, $\hat{x}_B$
$0_\diamond 1\,1$	3/4
$0_\diamond 1\,0$	1/2
$0_\diamond 0\,1$	1/4
$0_\diamond 0\,0$	0
$1_\diamond 1\,1$	−1/4
$1_\diamond 1\,0$	−1/2
$1_\diamond 0\,1$	−3/4
$1_\diamond 0\,0$	−1

In general, if we have a $(B + 1)$-bit binary two's-complement fraction of the form

$$a_{0\diamond}a_1a_2\ldots a_B,$$

then its value is

$$-a_0 2^0 + a_1 2^{-1} + a_2 2^{-2} + \cdots + a_B 2^{-B}.$$

Note that the symbol $\diamond$ denotes the "binary point" of the number. The relationship between the code words and the quantized signal levels depends on the parameter X_m in Figure 4.54. This parameter determines the full-scale level of the A/D converter. From Figure 4.54, we see that the step size of the quantizer would in general be

$$\Delta = \frac{2X_m}{2^{B+1}} = \frac{X_m}{2^B}. \tag{4.125}$$

The smallest quantization levels ($\pm\Delta$) correspond to the least significant bit of the binary code word. Furthermore, the numeric relationship between the code words and the quantized samples is

$$\hat{x}[n] = X_m \hat{x}_B[n], \tag{4.126}$$

since we have assumed that $\hat{x}_B[n]$ is a binary number such that $-1 \le \hat{x}_B[n] < 1$ (for two's complement). In this scheme, the binary coded samples $\hat{x}_B[n]$ are directly proportional

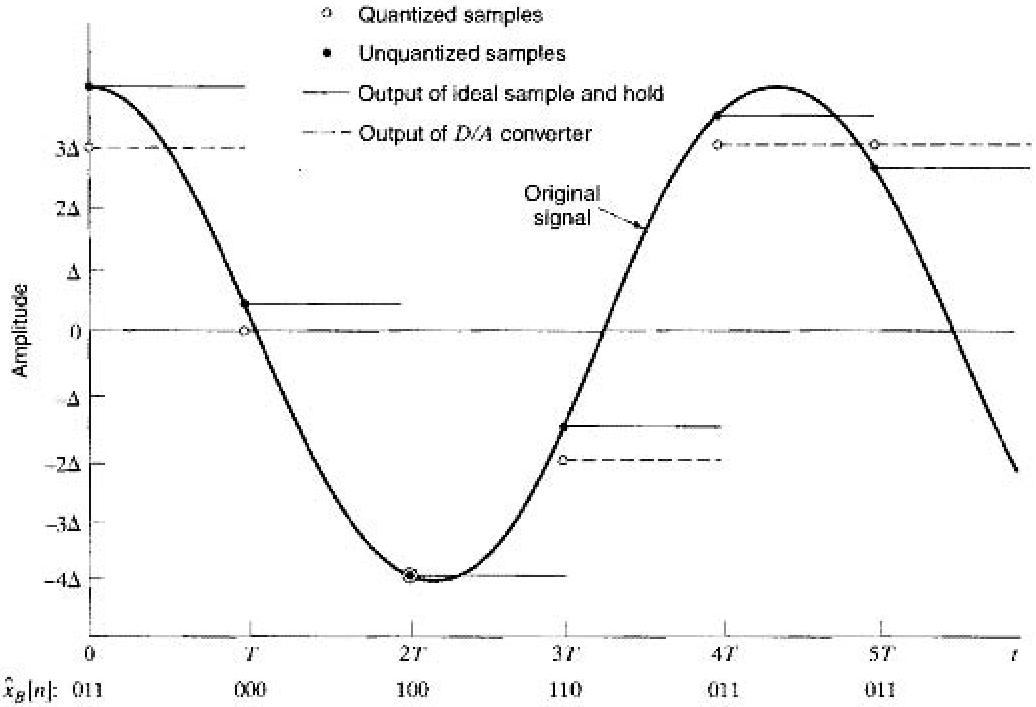


Figure 4.55 Sampling, quantization, coding, and D/A conversion with a 3-bit quantizer.

to the quantized samples (in two's-complement binary); therefore, they can be used as a numeric representation of the amplitude of the samples. Indeed, it is generally appropriate to assume that the input signal is normalized, so that the numeric values of $\hat{x}[n]$ and $\hat{x}_B[n]$ are identical and there is no need to distinguish between the quantized samples and the binary coded samples.

Figure 4.55 shows a simple example of quantization and coding of the samples of a sine wave using a 3-bit quantizer. The unquantized samples $x[n]$ are illustrated with solid dots, and the quantized samples $\hat{x}[n]$ are illustrated with open circles. Also shown is the output of an ideal sample-and-hold. The dotted lines labeled "output of D/A converter" will be discussed later. Figure 4.55 shows, in addition, the 3-bit code words that represent each sample. Note that, since the analog input $x_a(t)$ exceeds the full-scale value of the quantizer, some of the positive samples are "clipped."

Although much of the preceding discussion pertains to two's-complement coding of the quantization levels, the basic principles of quantization and coding in A/D conversion are the same regardless of the binary code used to represent the samples. A more detailed discussion of the binary arithmetic systems used in digital computing can be found in texts on computer arithmetic. (See, for example, Knuth, 1998.) We now turn to an analysis of the effects of quantization. Since this analysis does not depend on the assignment of binary code words, it will lead to rather general conclusions.

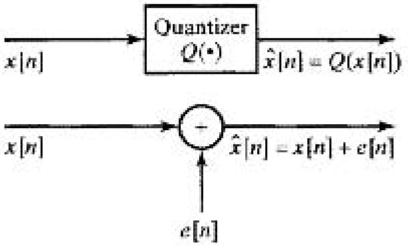


Figure 4.56 Additive noise model for quantizer.

4.8.3 Analysis of Quantization Errors

From Figures 4.54 and 4.55, we see that the quantized sample $\hat{x}[n]$ will generally be different from the true sample value $x[n]$. The difference between them is the *quantization error,* defined as

$$e[n] = \hat{x}[n] - x[n]. \tag{4.127}$$

For example, for the 3-bit quantizer of Figure 4.54, if $\Delta/2 < x[n] \leq 3\Delta/2$, then $\hat{x}[n] = \Delta$, and it follows that

$$-\Delta/2 \leq e[n] < \Delta/2. \tag{4.128}$$

In the case of Figure 4.54, Eq. (4.128) holds whenever

$$-9\Delta/2 < x[n] \leq 7\Delta/2. \tag{4.129}$$

In the general case of a $(B+1)$-bit quantizer with Δ given by Eq. (4.125), the quantization error satisfies Eq. (4.128) whenever

$$(-X_m - \Delta/2) < x[n] \leq (X_m - \Delta/2). \tag{4.130}$$

If $x[n]$ is outside this range, as it is for the sample at $t = 0$ in Figure 4.55, then the quantization error may be larger in magnitude than $\Delta/2$, and such samples are said to be *clipped*, and the quantizer is said to be *overloaded*.

A simplified, but useful, model of the quantizer is depicted in Figure 4.56. In this model, the quantization error samples are thought of as an additive noise signal. The model is exactly equivalent to the quantizer if we know $e[n]$. In most cases, however, $e[n]$ is not known, and a statistical model based on Figure 4.56 is then often useful in representing the effects of quantization. We will also use such a model in Chapters 6 and 9 to describe the effects of quantization in signal-processing algorithms. The statistical representation of quantization errors is based on the following assumptions:

1. The error sequence $e[n]$ is a sample sequence of a stationary random process.
2. The error sequence is uncorrelated with the sequence $x[n]$.[3]
3. The random variables of the error process are uncorrelated; i.e., the error is a white-noise process.
4. The probability distribution of the error process is uniform over the range of quantization error.

[3] This does not, of course, imply statistical independence, since the error is directly determined by the input signal.

As we will see, the preceding assumptions lead to a rather simple, but effective, analysis of quantization effects that can yield useful predictions of system performance. It is easy to find situations where these assumptions are not valid. For example, if $x_a(t)$ is a step function, the assumptions would not be justified. However, when the signal is a complicated signal, such as speech or music, where the signal fluctuates rapidly in a somewhat unpredictable manner, the assumptions are more realistic. Experimental measurements and theoretical analyses for random signal inputs have shown that, when the quantization step size (and therefore the error) is small and when the signal varies in a complicated manner, the measured correlation between the signal and the quantization error decreases, and the error samples also become uncorrelated. (See Bennett, 1948; Widrow, 1956, 1961; Sripad and Snyder, 1977; and Widrow and Kollár, 2008.) In a heuristic sense, the assumptions of the statistical model appear to be valid when the quantizer is not overloaded and when the signal is sufficiently complex, and the quantization steps are sufficiently small, so that the amplitude of the signal is likely to traverse many quantization steps from sample to sample.

Example 4.10 Quantization Error for a Sinusoidal Signal

As an illustration, Figure 4.57(a) shows the sequence of unquantized samples of the cosine signal $x[n] = 0.99\cos(n/10)$. Figure 4.57(b) shows the quantized sample sequence $\hat{x}[n] = Q\{x[n]\}$ for a 3-bit quantizer $(B + 1 = 3)$, assuming that $X_m = 1$. The dashed lines in this figure show the eight possible quantization levels. Figures 4.57(c) and 4.57(d) show the quantization error $e[n] = \hat{x}[n] - x[n]$ for 3- and 8-bit quantization, respectively. In each case, the scale of the quantization error is adjusted so that the range $\pm\Delta/2$ is indicated by the dashed lines.

Notice that in the 3-bit case, the error signal is highly correlated with the unquantized signal. For example, around the positive and negative peaks of the cosine, the quantized signal remains constant over many consecutive samples, so that the error has the shape of the input sequence during these intervals. Also, note that during the intervals around the positive peaks, the error is greater than $\Delta/2$ in magnitude because the signal level is too large for this setting of the quantizer parameters. On the other hand, the quantization error for 8-bit quantization has no apparent patterns.[4] Visual inspection of these figures supports the preceding assertions about the quantization-noise properties in the finely quantized (8-bit) case; i.e., the error samples appear to vary randomly, with no apparent correlation with the unquantized signal, and they range between $-\Delta/2$ and $+\Delta/2$.

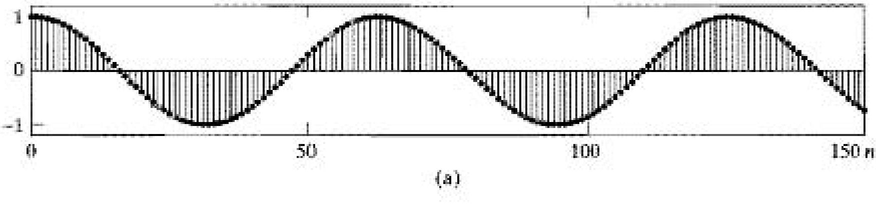


Figure 4.57 Example of quantization noise. (a) Unquantized samples of the signal $x[n] = 0.99\cos(n/10)$.

[4] For periodic cosine signals, the quantization error would, of course, be periodic, too; and therefore, its power spectrum would be concentrated at multiples of the frequency of the input signal. We used the frequency $\omega_0 = 1/10$ to avoid this case in the example.

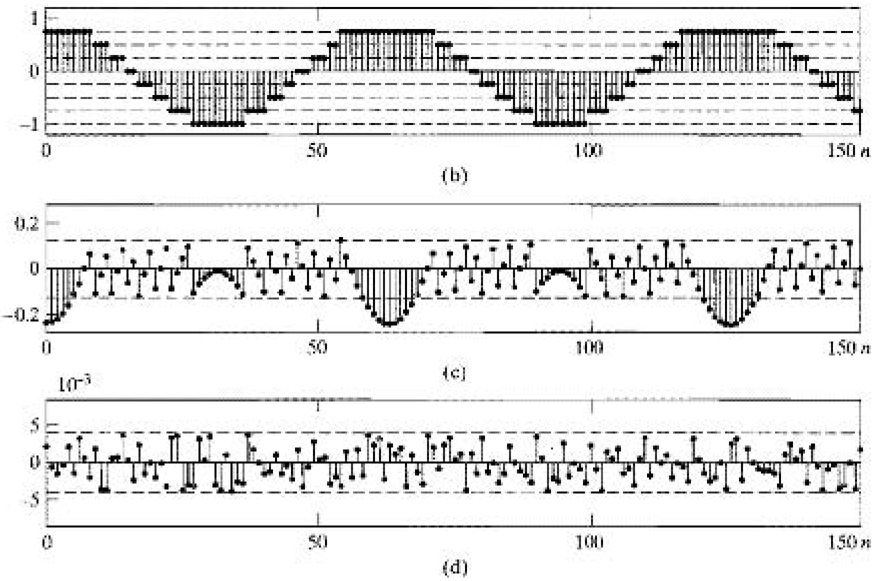


Figure 4.57 (*continued*) (b) Quantized samples of the cosine waveform in part (a) with a 3-bit quantizer. (c) Quantization error sequence for 3-bit quantization of the signal in (a). (d) Quantization error sequence for 8-bit quantization of the signal in (a).

For quantizers that round the sample value to the nearest quantization level, as shown in Figure 4.54, the amplitude of the quantization noise is in the range

$$-\Delta/2 \le e[n] < \Delta/2. \tag{4.131}$$

For small Δ, it is reasonable to assume that $e[n]$ is a random variable uniformly distributed from $-\Delta/2$ to $\Delta/2$. Therefore, the 1^{st}-order probability density assumed for the quantization noise is as shown in Figure 4.58. (If truncation rather than rounding is used in implementing quantization, then the error would always be negative, and we would assume a uniform probability density from $-\Delta$ to 0.) To complete the statistical model for the quantization noise, we assume that successive noise samples are uncorrelated with each other and that $e[n]$ is uncorrelated with $x[n]$. Thus, $e[n]$ is assumed to be a uniformly distributed white-noise sequence. The mean value of $e[n]$ is zero, and its

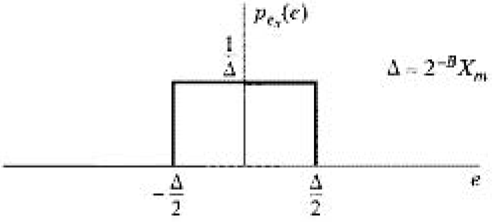


Figure 4.58 Probability density function of quantization error for a rounding quantizer such as that of Figure 4.54.

variance is

$$\sigma_e^2 = \int_{-\Delta/2}^{\Delta/2} e^2 \frac{1}{\Delta} de = \frac{\Delta^2}{12}. \tag{4.132}$$

For a $(B+1)$-bit quantizer with full-scale value X_m, the noise variance, or power, is

$$\sigma_e^2 = \frac{2^{-2B} X_m^2}{12}. \tag{4.133}$$

Equation (4.133) completes the white-noise model of quantization noise since the autocorrelation function would be $\phi_{ee}[m] = \sigma_e^2 \delta[m]$ and the corresponding power density spectrum would be

$$P_{ee}(e^{j\omega}) = \sigma_e^2 = \frac{2^{-2B} X_m^2}{12} \qquad |\omega| \leq \pi. \tag{4.134}$$

Example 4.11 Measurements of Quantization Noise

To confirm and illustrate the validity of the model for quantization noise, consider again quantization of the signal $x[n] = .99\cos(n/10)$ which can be computed with 64-bit floating-point precisions (for all practical purposes unquantized) and then quantized to $B+1$ bits. The quantization noise sequence can also be computed since we know both the input and the output of the quantizer. An amplitude histogram, which gives a count of the number of samples lying in each of a set of contiguous amplitude intervals or "bins," is often used as an estimate of the probability distribution of a random signal. Figure 4.59 shows histograms of the quantization noise for 16- and 8-bit quantization

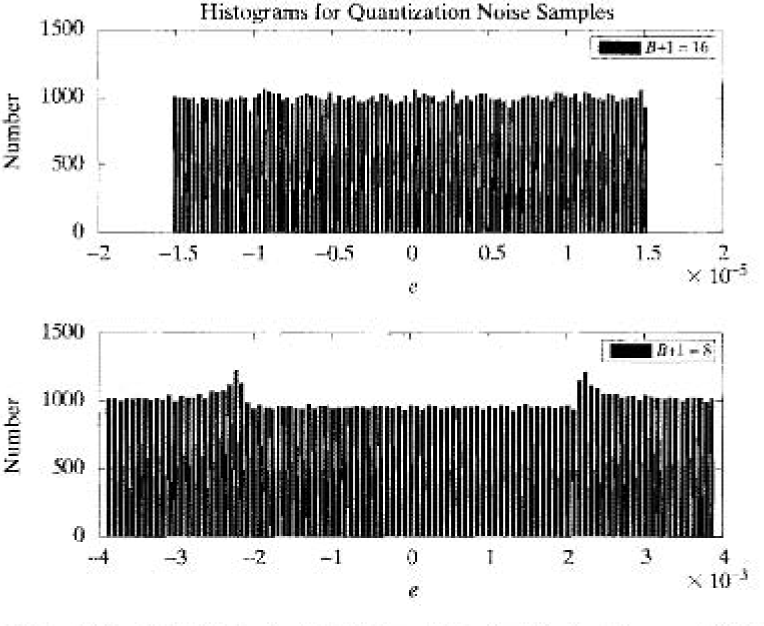


Figure 4.59 Histograms of quantization noise for (a) $B+1 = 16$ and (b) $B+1-8$.

with $X_m = 1$. Since the total number of samples was 101000, and the number of bins was 101, we should expect approximately 1000 samples in each bin if the noise is uniformly distributed. Furthermore the total range of samples should be $\pm 1/2^{16} = 1.53 \times 10^{-5}$ for 16-bit quantization and $\pm 1/2^8 = 3.9 \times 10^{-3}$ for 8-bit quantization. The histograms of Figure 4.59 are consistent with these values, although the 8-bit case shows some obvious deviation from the uniform distribution.

In Chapter 10, we show how to calculate estimates of the power density spectrum. Figure 4.60 shows such spectrum estimates for quantization noise signals where $B+1 = 16, 12, 8,$ and 4 bits. Observe that in this example, when the number of bits is 8 or greater, the spectrum is quite flat over the entire frequency range $0 \le \omega \le \pi$, and the spectrum level (in dB) is quite close to the value

$$10\log_{10}(P_{ee}(e^{j\omega})) = 10\log_{10}\left(\frac{1}{12(2^{2B})}\right) = -(10.79 + 6.02B),$$

which is predicted by the white-noise uniform-distribution model. Note that the curves for $B = 7, 11,$ and 15 differ at all frequencies by about 24 dB. Observe, however, that when $B + 1 = 4$, the model fails to predict the shape of the power spectrum of the noise.

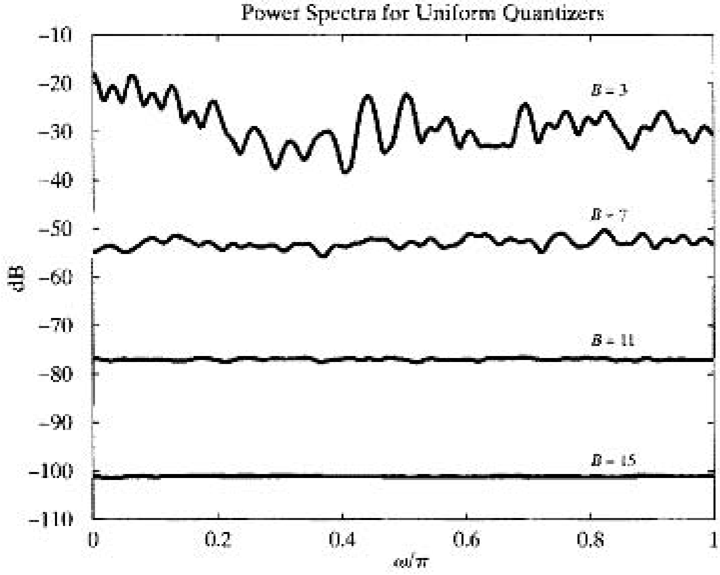


Figure 4.60 Spectra of quantization noise for several values of B.

This example demonstrates that the assumed model for quantization noise is useful in predicting the performance of uniform quantizers. A common measure of the amount of degradation of a signal by additive noise in general and quantization noise in particular is the signal-to-noise ratio (SNR), defined as the ratio of signal variance (power) to noise variance. Expressed in dB, the signal-to-quantization-noise ratio of a

$(B+1)$-bit uniform quantizer is

$$\begin{aligned} \mathrm{SNR}_Q &= 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) = 10\log_{10}\left(\frac{12\cdot 2^{2B}\sigma_x^2}{X_m^2}\right) \\ &= 6.02B + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right). \end{aligned} \tag{4.135}$$

From Eq. (4.135), we see that the SNR increases approximately 6 dB for each bit added to the word length of the quantized samples, i.e., for each doubling of the number of quantization levels. It is important to consider the term

$$-20\log_{10}\left(\frac{X_m}{\sigma_x}\right) \tag{4.136}$$

in Eq. (4.135). First, recall that X_m is a parameter of the quantizer, and it would usually be fixed in a practical system. The quantity σ_x is the rms value of the signal amplitude, and it would necessarily be less than the peak amplitude of the signal. For example, if $x_a(t)$ is a sine wave of peak amplitude X_p, then $\sigma_x = X_p/\sqrt{2}$. If σ_x is too large, the peak signal amplitude will exceed the full-scale amplitude X_m of the A/D converter. In this case Eq. (4.135) is no longer valid, and severe distortion results. On the other hand, if σ_x is too small, then the term in Eq. (4.136) will become large and negative, thereby decreasing the SNR in Eq. (4.135). In fact, it is easily seen that when σ_x is halved, the SNR decreases by 6 dB. Thus, it is very important that the signal amplitude be carefully matched to the full-scale amplitude of the A/D converter.

Example 4.12 SNR for Sinusoidal Signal

Using the signal $x[n] = A\cos(n/10)$, we can compute the quantization error for different values of $B+1$ with $X_m = 1$ and A varying. Figure 4.61 shows estimates of SNR as a function of X_m/σ_x obtained by computing the average power over many samples of the signal and dividing by the corresponding estimate of the average power of the noise; i.e.,

$$\mathrm{SNR}_Q = 10\log_{10}\left(\frac{\dfrac{1}{N}\displaystyle\sum_{n=0}^{N-1}(x[n])^2}{\dfrac{1}{N}\displaystyle\sum_{n=0}^{N-1}(e[n])^2}\right),$$

where in the case of Figure 4.61, $N = 101000$.

Observe that the curves in Figure 4.61 closely follow Eq. (4.135) over a wide range of values of B. In particular, the curves are straight lines as a function of $\log(X_m/\sigma_x)$, and they are offset from one another by 12 dB because the values of B differ by 2. SNR increases as X_m/σ_x decreases since increasing σ_x with X_m fixed

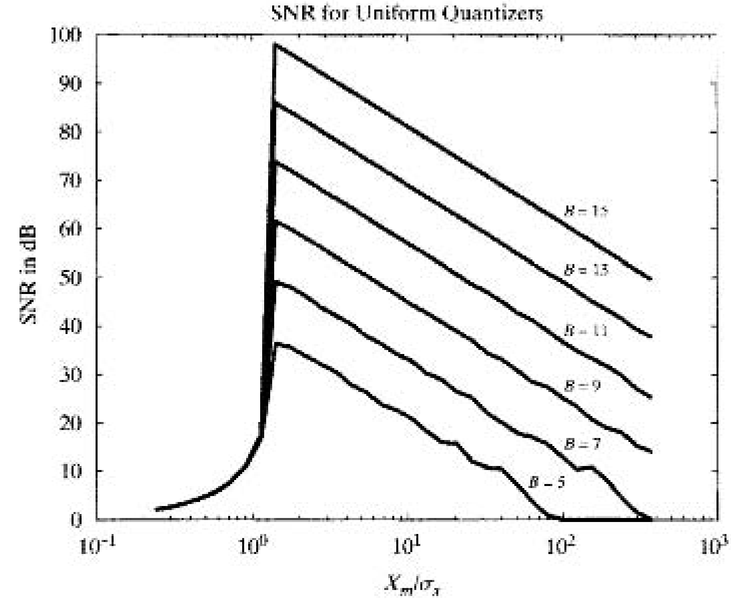


Figure 4.61 Signal-to-quantization-noise ratio as a function of X_m/σ_x for several values of B.

means that the signal uses more of the available quantization levels. However, note the precipitous fall of the curves as $X_m/\sigma_x \rightarrow 1$. Since $\sigma_x = .707A$ for a sine wave, this means that the amplitude A becomes greater than $X_m = 1$ and severe clipping occurs. Thus, the SNR decreases rapidly after the amplitude exceeds X_m.

For analog signals such as speech or music, the distribution of amplitudes tends to be concentrated about zero and falls off rapidly with increasing amplitude. In such cases, the probability that the magnitude of a sample will exceed three or four times the rms value is very low. For example, if the signal amplitude has a Gaussian distribution, only 0.064 percent of the samples would have an amplitude greater than $4\sigma_x$. Thus, to avoid clipping the peaks of the signal (as is assumed in our statistical model), we might set the gain of filters and amplifiers preceding the A/D converter so that $\sigma_x = X_m/4$. Using this value of σ_x in Eq. (4.135) gives

$$\text{SNR}_Q \approx 6B - 1.25 \text{ dB}. \tag{4.137}$$

For example, obtaining a SNR of about 90–96 dB for use in high-quality music recording and playback requires 16-bit quantization, but it should be remembered that such performance is obtained only if the input signal is carefully matched to the full-scale range of the A/D converter.

This trade-off between peak signal amplitude and absolute size of the quantization noise is fundamental to any quantization process. We will see its importance again in Chapter 6 when we discuss round-off noise in implementing discrete-time linear systems.

4.8.4 D/A Conversion

In Section 4.3, we discussed how a bandlimited signal can be reconstructed from a sequence of samples using ideal lowpass filtering. In terms of Fourier transforms, the reconstruction is represented as

$$X_r(j\Omega) = X(e^{j\Omega T})H_r(j\Omega), \tag{4.138}$$

where $X(e^{j\omega})$ is the DTFT of the sequence of samples and $X_r(j\Omega)$ is the Fourier transform of the reconstructed continuous-time signal. The ideal reconstruction filter is

$$H_r(j\Omega) = \begin{cases} T, & |\Omega| < \pi/T, \\ 0, & |\Omega| \geq \pi/T. \end{cases} \tag{4.139}$$

For this choice of $H_r(j\Omega)$, the corresponding relation between $x_r(t)$ and $x[n]$ is

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]\frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}. \tag{4.140}$$

The system that takes the sequence $x[n]$ as input and produces $x_r(t)$ as output is called the *ideal D/C converter*. A physically realizable counterpart to the ideal D/C converter is a *digital-to-analog converter* (D/A converter) followed by an analog lowpass filter. As depicted in Figure 4.62, a D/A converter takes a sequence of binary code words $\hat{x}_B[n]$ as its input and produces a continuous-time output of the form

$$\begin{aligned} x_{DA}(t) &= \sum_{n=-\infty}^{\infty} X_m \hat{x}_B[n] h_0(t-nT) \\ &= \sum_{n=-\infty}^{\infty} \hat{x}[n] h_0(t-nT), \end{aligned} \tag{4.141}$$

where $h_0(t)$ is the impulse response of the zero-order hold given by Eq. (4.122). The dotted lines in Figure 4.55 show the output of a D/A converter for the quantized examples of the sine wave. Note that the D/A converter holds the quantized sample for one sample period in the same way that the sample-and-hold holds the unquantized input sample. If we use the additive-noise model to represent the effects of quantization, Eq. (4.141) becomes

$$x_{DA}(t) = \sum_{n=-\infty}^{\infty} x[n]h_0(t-nT) + \sum_{n=-\infty}^{\infty} e[n]h_0(t-nT). \tag{4.142}$$

To simplify our discussion, we define

$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n]h_0(t-nT), \tag{4.143}$$

$$e_0(t) = \sum_{n=-\infty}^{\infty} e[n]h_0(t-nT). \tag{4.144}$$

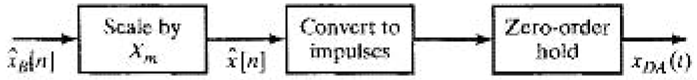


Figure 4.62 Block diagram of D/A converter.

so that Eq. (4.142) can be written as

$$x_{DA}(t) = x_0(t) + e_0(t). \tag{4.145}$$

The signal component $x_0(t)$ is related to the input signal $x_a(t)$, since $x[n] = x_a(nT)$. The noise signal $e_0(t)$ depends on the quantization-noise samples $e[n]$ in the same way that $x_0(t)$ depends on the unquantized signal samples. The Fourier transform of Eq. (4.143) is

$$\begin{aligned} X_0(j\Omega) &= \sum_{n=-\infty}^{\infty} x[n]H_0(j\Omega)e^{-j\Omega nT} \\ &= \left(\sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega Tn}\right) H_0(j\Omega) \\ &= X(e^{j\Omega T})H_0(j\Omega). \end{aligned} \tag{4.146}$$

Now, since

$$X(e^{j\Omega T}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_a\left(j\left(\Omega - \frac{2\pi k}{T}\right)\right), \tag{4.147}$$

it follows that

$$X_0(j\Omega) = \left[\frac{1}{T}\sum_{k=-\infty}^{\infty} X_a\left(j\left(\Omega - \frac{2\pi k}{T}\right)\right)\right] H_0(j\Omega). \tag{4.148}$$

If $X_a(j\Omega)$ is bandlimited to frequencies below π/T, the shifted copies of $X_a(j\Omega)$ do not overlap in Eq. (4.148), and if we define a compensated reconstruction filter as

$$\tilde{H}_r(j\Omega) = \frac{H_r(j\Omega)}{H_0(j\Omega)}, \tag{4.149}$$

then the output of the filter will be $x_a(t)$ if the input is $x_0(t)$. The frequency response of the zero-order-hold filter is easily shown to be

$$H_0(j\Omega) = \frac{2\sin(\Omega T/2)}{\Omega}e^{-j\Omega T/2}. \tag{4.150}$$

Therefore, the compensated reconstruction filter is

$$\tilde{H}_r(j\Omega) = \begin{cases} \dfrac{\Omega T/2}{\sin(\Omega T/2)}e^{j\Omega T/2}, & |\Omega| < \pi/T, \\ 0, & |\Omega| \geq \pi/T. \end{cases} \tag{4.151}$$

Figure 4.63(a) shows $|H_0(j\Omega)|$ as given by Eq. (4.150), compared with the magnitude of the ideal interpolation filter $|H_r(j\Omega)|$ as given by Eq. (4.139). Both filters have a gain of T at $\Omega = 0$, but the zero-order-hold, although lowpass in nature, does not cut off sharply at $\Omega = \pi/T$. Figure 4.63(b) shows the magnitude of the frequency response of the ideal compensated reconstruction filter to be used following a zero-order-hold reconstruction system such as a D/A converter. The phase response would ideally correspond to an advance time shift of $T/2$ seconds to compensate for the delay of that amount introduced by the zero-order hold. Since such a time advance cannot be realized in practical real-time approximations to the ideal compensated reconstruction

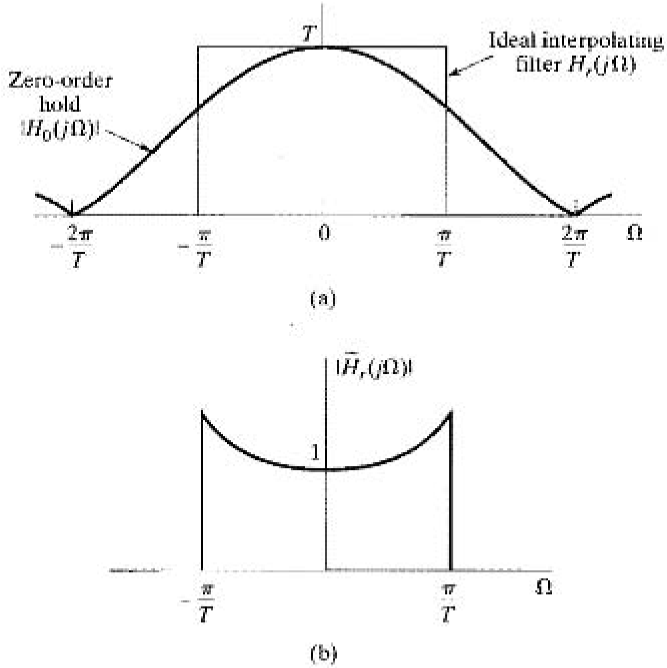


Figure 4.63 (a) Frequency response of zero-order hold compared with ideal interpolating filter. (b) Ideal compensated reconstruction filter for use with a zero-order-hold output.

filter, only the magnitude response would normally be compensated, and often even this compensation is neglected, since the gain of the zero-order hold drops only to $2/\pi$ (or -4 dB) at $\Omega = \pi/T$.

Figure 4.64 shows a D/A converter followed by an ideal compensated reconstruction filter. As can be seen from the preceding discussion, with the ideal compensated reconstruction filter following the D/A converter, the reconstructed output signal would be

$$\begin{aligned} \hat{x}_r(t) &= \sum_{n=-\infty}^{\infty} \hat{x}[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T} \\ &= \sum_{n=-\infty}^{\infty} x[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T} + \sum_{n=-\infty}^{\infty} e[n] \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}. \end{aligned} \tag{4.152}$$

In other words, the output would be

$$\hat{x}_r(t) = x_a(t) + e_a(t), \tag{4.153}$$

where $e_a(t)$ would be a bandlimited white-noise signal.

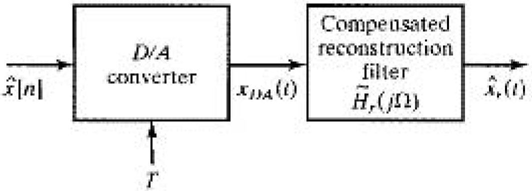


Figure 4.64 Physical configuration for D/A conversion.

Returning to a consideration of Figure 4.47(b), we are now in a position to understand the behavior of systems for digital processing of analog signals. If we assume that the output of the antialiasing filter is bandlimited to frequencies below π/T, that $\tilde{H}_r(j\Omega)$ is similarly bandlimited, and that the discrete-time system is linear and time invariant, then the output of the overall system will be of the form

$$\hat{y}_r(t) = y_a(t) + e_a(t), \tag{4.154}$$

where

$$TY_a(j\Omega) = \tilde{H}_r(j\Omega)H_0(j\Omega)H(e^{j\Omega T})H_{\text{aa}}(j\Omega)X_c(j\Omega), \tag{4.155}$$

in which $H_{\text{aa}}(j\Omega)$, $H_0(j\Omega)$, and $\tilde{H}_r(j\Omega)$ are the frequency responses of the antialiasing filter, the zero-order hold of the D/A converter, and the reconstruction lowpass filter, respectively. $H(e^{j\Omega T})$ is the frequency response of the discrete-time system. Similarly, assuming that the quantization noise introduced by the A/D converter is white noise with variance $\sigma_e^2 = \Delta^2/12$, it can be shown that the power spectrum of the output noise is

$$P_{e_a}(j\Omega) = |\tilde{H}_r(j\Omega)H_0(j\Omega)H(e^{j\Omega T})|^2\sigma_e^2, \tag{4.156}$$

i.e., the input quantization noise is changed by the successive stages of discrete- and continuous-time filtering. From Eq. (4.155), it follows that, under the model for the quantization error and the assumption of negligible aliasing, the overall effective frequency response from $x_c(t)$ to $\hat{y}_r(t)$ is

$$TH_{\text{eff}}(j\Omega) = \tilde{H}_r(j\Omega)H_0(j\Omega)H(e^{j\Omega T})H_{\text{aa}}(j\Omega). \tag{4.157}$$

If the antialiasing filter is ideal, as in Eq. (4.118), and if the compensation of the reconstruction filter is ideal, as in Eq. (4.151), then the effective frequency response is as given in Eq. (4.119). Otherwise Eq. (4.157) provides a reasonable model for the effective response. Note that Eq. (4.157) suggests that compensation for imperfections in any of the four terms can, in principle, be included in any of the other terms; e.g., the discrete-time system can include appropriate compensation for the antialiasing filter or the zero-order hold or the reconstruction filter or all of these.

In addition to the filtering supplied by Eq. (4.157), Eq. (4.154) reminds us that the output will also be contaminated by the filtered quantization noise. In Chapter 6 we will see that noise can be introduced as well in the implementation of the discrete-time linear system. This internal noise will, in general, be filtered by parts of the discrete-time system implementation, by the zero-order hold of the D/A converter, and by the reconstruction filter.

4.9 OVERSAMPLING AND NOISE SHAPING IN A/D AND D/A CONVERSION

In Section 4.8.1, we showed that oversampling can make it possible to implement sharp-cutoff antialiasing filtering by incorporating digital filtering and decimation. As we discuss in Section 4.9.1, oversampling and subsequent discrete-time filtering and downsampling also permit an increase in the step size Δ of the quantizer or, equivalently, a reduction in the number of bits required in the A/D conversion. In Section 4.9.2 we show how the step size can be reduced even further by using oversampling together with quantization-noise feedback, and in Section 4.9.3 we show how the oversampling principle can be applied in D/A conversion.

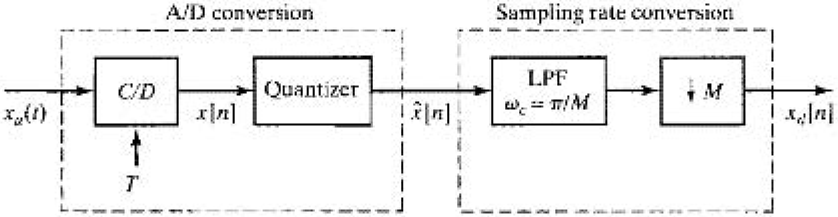


Figure 4.65 Oversampled A/D conversion with simple quantization and downsampling.

4.9.1 Oversampled A/D Conversion with Direct Quantization

To explore the relation between oversampling and the quantization step size, we consider the system in Figure 4.65. To analyze the effect of oversampling in this system, we consider $x_a(t)$ to be a zero-mean, wide-sense-stationary, random process with power-spectral density denoted by $\Phi_{x_a x_a}(j\Omega)$ and autocorrelation function denoted by $\phi_{x_a x_a}(\tau)$. To simplify our discussion, we assume initially that $x_a(t)$ is already bandlimited to Ω_N, i.e.,

$$\Phi_{x_a x_a}(j\Omega) = 0, \qquad |\Omega| \geq \Omega_N, \tag{4.158}$$

and we assume that $2\pi/T = 2M\Omega_N$. The constant M, which is assumed to be an integer, is called the *oversampling ratio*. Using the additive noise model discussed in detail in Section 4.8.3, we can replace Figure 4.65 by Figure 4.66. The decimation filter in Figure 4.66 is an ideal lowpass filter with unity gain and cutoff frequency $\omega_c = \pi/M$. Because the entire system of Figure 4.66 is linear, its output $x_d[n]$ has two components, one due to the signal input $x_a(t)$ and one due to the quantization noise input $e[n]$. We denote these components by $x_{da}[n]$ and $x_{de}[n]$, respectively.

Our goal is to determine the ratio of signal power $\mathcal{E}\{x_{da}^2[n]\}$ to quantization-noise power $\mathcal{E}\{x_{de}^2[n]\}$ in the output $x_d[n]$ as a function of the quantizer step size Δ and the oversampling ratio M. Since the system of Figure 4.66 is linear, and since the noise is assumed to be uncorrelated with the signal, we can treat the two sources separately in computing the respective powers of the signal and noise components at the output.

First, we will consider the signal component of the output. We begin by relating the power spectral density, autocorrelation function, and signal power of the sampled signal $x[n]$ to the corresponding functions for the continuous-time analog signal $x_a(t)$.

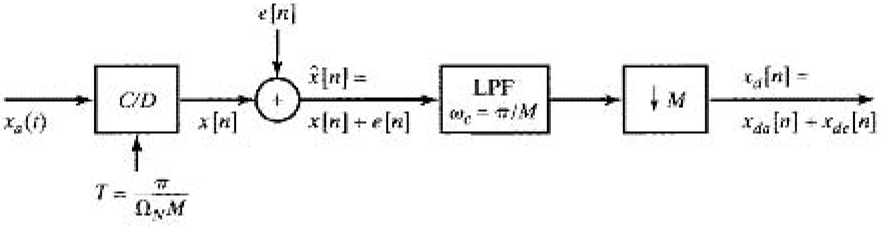


Figure 4.66 System of Figure 4.65 with quantizer replaced by linear noise model.

Let $\phi_{xx}[m]$ and $\Phi_{xx}(e^{j\omega})$ respectively denote the autocorrelation and power spectral density of $x[n]$. Then, by definition, $\phi_{xx}[m] = \mathcal{E}\{x[n+m]x[n]\}$, and since $x[n] = x_a(nT)$ and $x[n+m] = x_a(nT + mT)$,

$$\mathcal{E}\{x[n+m]x[n]\} = \mathcal{E}\{x_a((n+m)T)x_a(nT)\}. \tag{4.159}$$

Therefore,

$$\phi_{xx}[m] = \phi_{x_a x_a}(mT); \tag{4.160}$$

i.e., the autocorrelation function of the sequence of samples is a sampled version of the autocorrelation function of the corresponding continuous-time signal. In particular the wide-sense-stationarity assumption implies that $\mathcal{E}\{x_a^2(t)\}$ is a constant independent of t. It then follows that

$$\mathcal{E}\{x^2[n]\} = \mathcal{E}\{x_a^2(nT)\} = \mathcal{E}\{x_a^2(t)\} \qquad \text{for all } n \text{ or } t. \tag{4.161}$$

Since the power spectral densities are the Fourier transforms of the autocorrelation functions, as a consequence of Eq. (4.160),

$$\Phi_{xx}(e^{j\Omega T}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} \Phi_{x_a x_a}\left[j\left(\Omega - \frac{2\pi k}{T}\right)\right]. \tag{4.162}$$

Assuming that the input is bandlimited as in Eq. (4.158), and assuming oversampling by a factor of M so that $2\pi/T = 2M\Omega_N$, we obtain, by substituting $\Omega = \omega/T$ into Eq. (4.162)

$$\Phi_{xx}(e^{j\omega}) = \begin{cases} \dfrac{1}{T}\Phi_{x_a x_a}\left(j\dfrac{\omega}{T}\right), & |\omega| < \pi/M, \\ 0, & \pi/M < \omega \le \pi. \end{cases} \tag{4.163}$$

For example, if $\Phi_{x_a x_a}(j\Omega)$ is as depicted in Figure 4.67(a), and if we choose the sampling rate to be $2\pi/T = 2M\Omega_N$, then $\Phi_{xx}(e^{j\omega})$ will be as depicted in Figure 4.67(b).

It is instructive to demonstrate that Eq. (4.161) is true by utilizing the power spectrum. The total power of the original analog signal is given by

$$\mathcal{E}\{x_a^2(t)\} = \frac{1}{2\pi}\int_{-\Omega_N}^{\Omega_N} \Phi_{x_a x_a}(j\Omega)d\Omega.$$

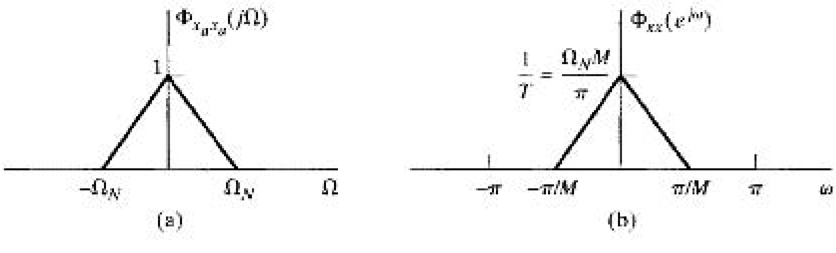


Figure 4.67 Illustration of frequency and amplitude scaling between $\Phi_{x_a x_a}(j\Omega)$ and $\Phi_{xx}(e^{j\omega})$.

From Eq. (4.163), the total power of the sampled signal is

$$\mathcal{E}\{x^2[n]\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_{xx}(e^{j\omega})d\omega \tag{4.164}$$

$$= \frac{1}{2\pi}\int_{-\pi/M}^{\pi/M} \frac{1}{T}\Phi_{x_ax_a}\left(j\frac{\omega}{T}\right)d\omega. \tag{4.165}$$

Using the fact that $\Omega_N T = \pi/M$ and making the substitution $\Omega = \omega/T$ in Eq. (4.165) gives

$$\mathcal{E}\{x^2[n]\} = \frac{1}{2\pi}\int_{-\Omega_N}^{\Omega_N} \Phi_{x_ax_a}(j\Omega)d\Omega = \mathcal{E}\{x_a^2(t)\}.$$

Thus, the total power of the sampled signal and the total power of the original analog signal are exactly the same as was also shown in Eq. (4.161). Since the decimation filter is an ideal lowpass filter with cutoff $\omega_c = \pi/M$, the signal $x[n]$ passes unaltered through the filter. Therefore, the downsampled signal component at the output, $x_{da}[n] = x[nM] = x_a(nMT)$, also has the same total power. This can be seen from the power spectrum by noting that, since $\Phi_{xx}(e^{j\omega})$ is bandlimited to $|\omega| < \pi/M$,

$$\Phi_{x_{da}x_{da}}(e^{j\omega}) = \frac{1}{M}\sum_{k=0}^{M-1} \Phi_{xx}(e^{j(\omega-2\pi k)/M})$$

$$= \frac{1}{M}\Phi_{xx}(e^{j\omega/M}) \qquad |\omega| < \pi. \tag{4.166}$$

Using Eq. (4.166), we obtain

$$\mathcal{E}\{x_{da}^2[n]\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_{x_{da}x_{da}}(e^{j\omega})d\omega$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{M}\Phi_{xx}(e^{j\omega/M})d\omega$$

$$= \frac{1}{2\pi}\int_{-\pi/M}^{\pi/M} \Phi_{xx}(e^{j\omega})\,d\omega = \mathcal{E}\{x^2[n]\},$$

which shows that the power of the signal component stays the same as it traverses the entire system from the input $x_a(t)$ to the corresponding output component $x_{da}[n]$. In terms of the power spectrum, this occurs because, for each scaling of the frequency axis that results from sampling, we have a counterbalancing inverse scaling of the amplitude, so that the area under the power spectrum remains the same as we go from $\Phi_{x_ax_a}(j\Omega)$ to $\Phi_{xx}(e^{j\omega})$ to $\Phi_{x_{da}x_{da}}(e^{j\omega})$ by sampling.

Now let us consider the noise component that is generated by quantization. According to the model in Section 4.8.3, we assume that $e[n]$ is a wide-sense-stationary white-noise process with zero mean and variance[5]

$$\sigma_e^2 = \frac{\Delta^2}{12}.$$

[5] Since the random process has zero mean, the average power and the variance are the same.

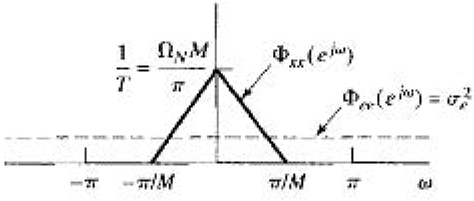


Figure 4.68 Power spectral density of signal and quantization noise with an oversampling factor of M.

Consequently, the autocorrelation function and power density spectrum for $e[n]$ are, respectively,

$$\phi_{ee}[m] = \sigma_e^2 \delta[m] \tag{4.167}$$

and

$$\Phi_{ee}(e^{j\omega}) = \sigma_e^2 \qquad |\omega| < \pi. \tag{4.168}$$

In Figure 4.68, we show the power density spectrum of $e[n]$ and of $x[n]$. The power density spectrum of the quantized signal $\hat{x}[n]$ is the sum of these, since the signal and quantization-noise samples are assumed to be uncorrelated in our model.

Although we have shown that the power in either $x[n]$ or $e[n]$ does not depend on M, we note that as the oversampling ratio M increases, less of the quantization-noise spectrum overlaps with the signal spectrum. It is this effect of the oversampling that lets us improve the signal-to-quantization-noise ratio by sampling-rate reduction. Specifically, the ideal lowpass filter removes the quantization noise in the band $\pi/M < |\omega| \leq \pi$, while it leaves the signal component unaltered. The noise power at the output of the ideal lowpass filter is

$$\mathcal{E}\{e^2[n]\} = \frac{1}{2\pi}\int_{-\pi/M}^{\pi/M} \sigma_e^2 d\omega = \frac{\sigma_e^2}{M}.$$

Next, the lowpass filtered signal is downsampled, and, as we have seen, the signal power in the downsampled output remains the same. In Figure 4.69, we show the resulting power density spectrum of both $x_{da}[n]$ and $x_{de}[n]$. Comparing Figures 4.68 and 4.69, we can see that the area under the power density spectrum for the signal has not changed, since the frequency axis and amplitude axis scaling have been inverses of each other. On the other hand, the noise power in the decimated output is the same as at the output of the lowpass filter; i.e.,

$$\mathcal{E}\{x_{de}^2[n]\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\sigma_e^2}{M} d\omega = \frac{\sigma_e^2}{M} = \frac{\Delta^2}{12M}. \tag{4.169}$$

Thus, the quantization-noise power $\mathcal{E}\{x_{de}^2[n]\}$ has been reduced by a factor of M through the filtering and downsampling, while the signal power has remained the same.

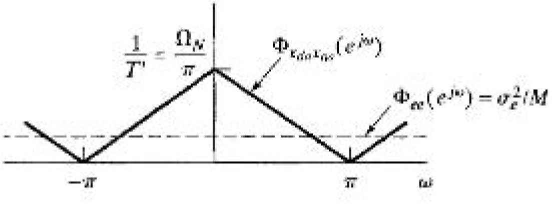


Figure 4.69 Power spectral density of signal and quantization noise after downsampling.

From Eq. (4.169), we see that for a given quantization noise power, there is a clear trade-off between the oversampling factor M and the quantizer step size Δ. Equation (4.125) states that for a quantizer with $(B+1)$ bits and maximum input signal level between plus and minus X_m, the step size is

$$\Delta = X_m/2^B,$$

and therefore,

$$\mathcal{E}\{x_{de}^2[n]\} = \frac{1}{12M}\left(\frac{X_m}{2^B}\right)^2. \tag{4.170}$$

Equation (4.170) shows that for a fixed quantizer, the noise power can be decreased by increasing the oversampling ratio M. Since the signal power is independent of M, increasing M will increase the signal-to-quantization-noise ratio. Alternatively, for a fixed quantization noise power $P_{de} = \mathcal{E}\{x_{de}^2[n]\}$, the required value for B is

$$B = -\frac{1}{2}\log_2 M - \frac{1}{2}\log_2 12 - \frac{1}{2}\log_2 P_{de} + \log_2 X_m. \tag{4.171}$$

From Eq. (4.171), we see that for every doubling of the oversampling ratio M, we need 1/2 bit less to achieve a given signal-to-quantization-noise ratio, or, in other words, if we oversample by a factor $M = 4$, we need one less bit to achieve a desired accuracy in representing the signal.

4.9.2 Oversampled A/D Conversion with Noise Shaping

In the previous section, we showed that oversampling and decimation can improve the signal-to-quantization-noise ratio. This seems to be a somewhat remarkable result. It implies that we can, in principle, use very crude quantization in our initial sampling of the signal, and if the oversampling ratio is high enough, we can still obtain an accurate representation of the original samples by doing digital computation on the noisy samples. The problem with what we have seen so far is that, to make a significant reduction in the required number of bits, we need very large oversampling ratios. For example, to reduce the number of bits from 16 to 12 would require $M = 4^4 = 256$. This seems to be a rather high cost. However, the basic oversampling principle can lead to much higher gains if we combine it with the concept of noise spectrum shaping by feedback.

As was indicated in Figure 4.68, with direct quantization the power density spectrum of the quantization noise is constant over the entire frequency band. The basic concept in noise shaping is to modify the A/D conversion procedure so that the power density spectrum of the quantization noise is no longer uniform, but rather, is shaped such that most of the noise power is outside the band $|\omega| < \pi/M$. In that way, the subsequent filtering and downsampling removes more of the quantization-noise power.

The noise-shaping quantizer, generally referred to as a sampled-data Delta-Sigma modulator, is shown in Figure 4.70. (See Candy and Temes, 1992 and Schreier and Temes, 2005.) Figure 4.70(a) shows a block diagram of how the system is implemented with integrated circuits. The integrator is a switched-capacitor discrete-time integrator. The A/D converter can be implemented in many ways, but generally, it is a simple 1-bit quantizer or comparator. The D/A converter converts the digital output back to an analog pulse that is subtracted from the input signal at the input to the integrator. This system can

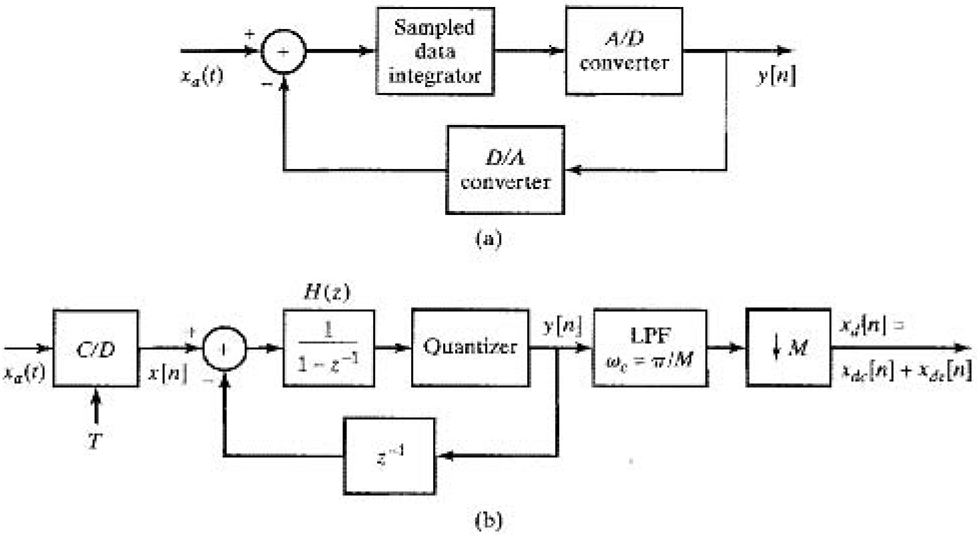


Figure 4.70 Oversampled quantizer with noise shaping.

be represented by the discrete-time equivalent system shown in Figure 4.70(b). The switched-capacitor integrator is represented by an accumulator system, and the delay in the feedback path represents the delay introduced by the D/A converter.

As before, we model the quantization error as an additive noise source so that the system in Figure 4.70 can be replaced by the linear model in Figure 4.71. In this system, the output $y[n]$ is the sum of two components: $y_x[n]$ due to the input $x[n]$ alone and $\hat{e}[n]$ due to the noise $e[n]$ alone.

We denote the transfer function from $x[n]$ to $y[n]$ as $H_x(z)$ and from $e[n]$ to $y[n]$ as $H_e(z)$. These transfer functions can both be calculated in a straightforward manner and are

$$H_x(z) = 1, \tag{4.172a}$$

$$H_e(z) = (1 - z^{-1}). \tag{4.172b}$$

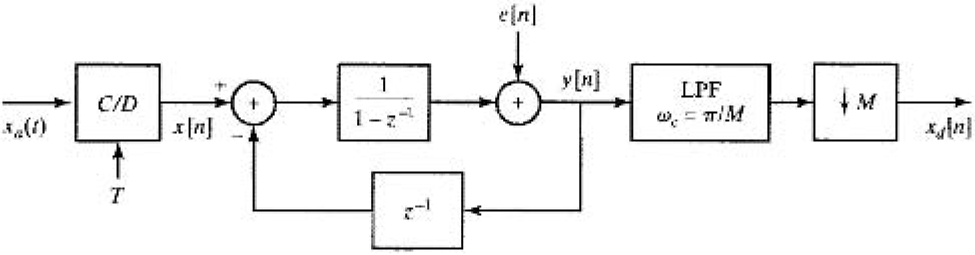


Figure 4.71 System of Figure 4.70 from $x_a(t)$ to $x_d[n]$ with quantizer replaced by a linear noise model.

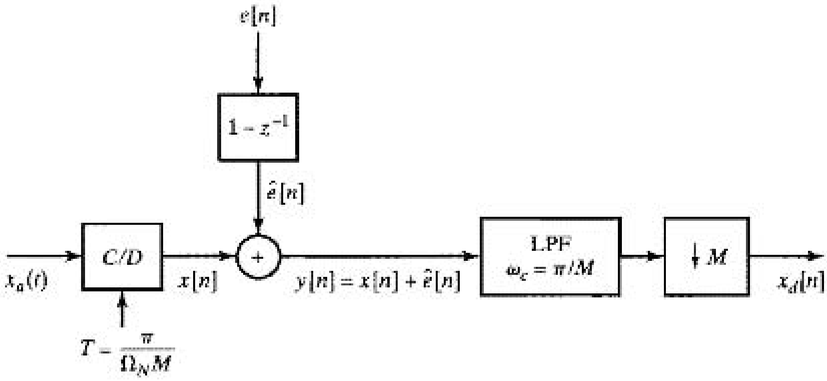


Figure 4.72 Equivalent representation of Figure 4.71.

Consequently,

$$y_x[n] = x[n], \tag{4.173a}$$

and

$$\hat{e}[n] = e[n] - e[n-1]. \tag{4.173b}$$

Therefore, the output $y[n]$ can be represented equivalently as $y[n] = x[n] + \hat{e}[n]$, where $x[n]$ appears unmodified at the output and the quantization noise $e[n]$ is modified by the first-difference operator $H_e(z)$. This is depicted in the block diagram in Figure 4.72. With the power density spectrum for $e[n]$ given by Eq. (4.168), the power density spectrum of the quantization noise $\hat{e}[n]$ that is present in $y[n]$ is

$$\begin{aligned}\Phi_{\hat{e}\hat{e}}(e^{j\omega}) &= \sigma_e^2 |H_e(e^{j\omega})|^2 \\ &= \sigma_e^2 [2\sin(\omega/2)]^2.\end{aligned} \tag{4.174}$$

In Figure 4.73, we show the power density spectrum of $\hat{e}[n]$, the power spectrum of $e[n]$, and the same signal power spectrum that was shown in Figure 4.67(b) and Figure 4.68. It is interesting to observe that the *total* noise power is increased from $\mathcal{E}\{e^2[n]\} = \sigma_e^2$

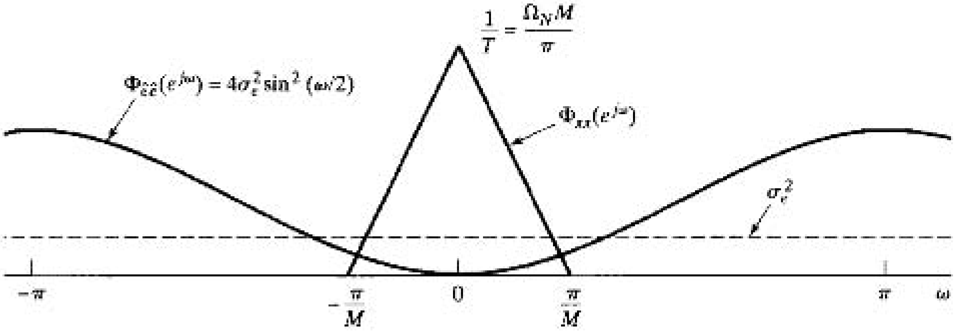


Figure 4.73 The power spectral density of the quantization noise and the signal.

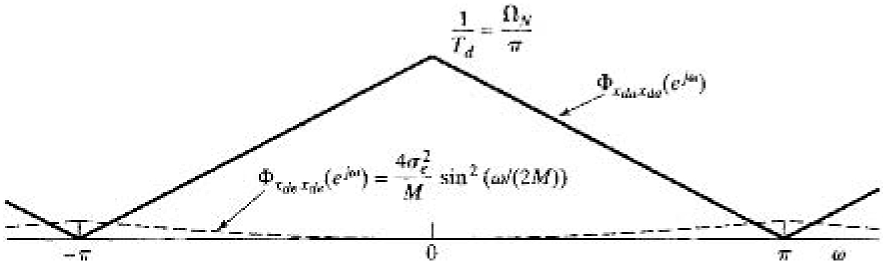


Figure 4.74 Power spectral density of the signal and quantization noise after downsampling.

at the quantizer to $\mathcal{E}\{\hat{e}^2[n]\} = 2\sigma_e^2$ at the output of the noise-shaping system. However, note that in comparison with Figure 4.68, the quantization noise has been shaped in such a way that more of the noise power is outside the signal band $|\omega| < \pi/M$ than in the direct oversampled case, where the noise spectrum is flat.

In the system of Figure 4.70, this out-of-band noise power is removed by the lowpass filter. Specifically, in Figure 4.74 we show the power density spectrum of $\Phi_{x_{da}x_{da}}(e^{j\omega})$ superimposed on the power density spectrum of $\Phi_{x_{de}x_{de}}(e^{j\omega})$. Since the downsampler does not remove any of the signal power, the signal power in $x_{da}[n]$ is

$$P_{da} = \mathcal{E}\{x_{da}^2[n]\} = \mathcal{E}\{x^2[n]\} = \mathcal{E}\{x_a^2(t)\}.$$

The quantization-noise power in the final output is

$$P_{de} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\Phi_{x_{de}x_{de}}(e^{j\omega})d\omega = \frac{1}{2\pi}\frac{\Delta^2}{12M}\int_{-\pi}^{\pi}\left(2\sin\left(\frac{\omega}{2M}\right)\right)^2 d\omega. \tag{4.175}$$

To compare this approximately with the results in Section 4.9.1, assume that M is sufficiently large so that

$$\sin\left(\frac{\omega}{2M}\right) \approx \frac{\omega}{2M}.$$

With this approximation, Eq. (4.175) is easily evaluated to obtain

$$P_{de} = \frac{1}{36}\frac{\Delta^2\pi^2}{M^3}. \tag{4.176}$$

From Eq. (4.176), we see again a trade-off between the oversampling ratio M and the quantizer step size Δ. For a $(B+1)$-bit quantizer and maximum input signal level between plus and minus X_m, $\Delta = X_m/2^B$. Therefore, to achieve a given quantization-noise power P_{de}, we must have

$$B = -\frac{3}{2}\log_2 M + \log_2(\pi/6) - \frac{1}{2}\log_2 P_{de} + \log_2 X_m. \tag{4.177}$$

Comparing Eq. (4.177) with Eq. (4.171), we see that, whereas with direct quantization a doubling of the oversampling ratio M gained 1/2 bit in quantization, the use of noise shaping results in a gain of 1.5 bits.

Table 4.1 gives the equivalent savings in quantizer bits over direct quantization with no oversampling ($M = 1$) for (a) direct quantization with oversampling, as discussed in Section 4.9.1, and (b) oversampling with noise shaping, as examined in this section.

TABLE 4.1 EQUIVALENT SAVINGS IN QUANTIZER BITS RELATIVE TO $M = 1$ FOR DIRECT QUANTIZATION AND 1st-ORDER NOISE SHAPING

M	Direct quantization	Noise shaping
4	1	2.2
8	1.5	3.7
16	2	5.1
32	2.5	6.6
64	3	8.1

The noise-shaping strategy in Figure 4.70 can be extended by incorporating a second stage of accumulation as shown in Figure 4.75. In this case, with the quantizer again modeled as an additive noise source $e[n]$, it can be shown that

$$y[n] = x[n] + \hat{e}[n]$$

where, in the two-stage case, $\hat{e}[n]$ is the result of processing the quantization noise $e[n]$ through the transfer function

$$H_e(z) = (1 - z^{-1})^2. \tag{4.178}$$

The corresponding power density spectrum of the quantization noise now present in $y[n]$ is

$$\Phi_{\hat{e}\hat{e}}(e^{j\omega}) = \sigma_e^2[2\sin(\omega/2)]^4, \tag{4.179}$$

with the result that, although the total noise power at the output of the two-stage noise-shaping system is greater than for the one-stage case, even more of the noise lies outside the signal band. More generally, p stages of accumulation and feedback can be used, with corresponding noise shaping given by

$$\Phi_{\hat{e}\hat{e}}(e^{j\omega}) = \sigma_e^2[2\sin(\omega/2)]^{2p}. \tag{4.180}$$

In Table 4.2, we show the equivalent reduction in quantizer bits as a function of the order p of the noise shaping and the oversampling ratio M. Note that with $p = 2$ and $M = 64$, we obtain almost 13 bits of increase in accuracy, suggesting that a 1-bit quantizer could achieve about 14-bit accuracy at the output of the decimator.

Although multiple feedback loops such as the one shown in Figure 4.75 promise greatly increased noise reduction, they are not without problems. Specifically, for large values of p, there is an increased potential for instability and oscillations to occur. An alternative structure known as multistage noise shaping (MASH) is considered in Problem 4.68.

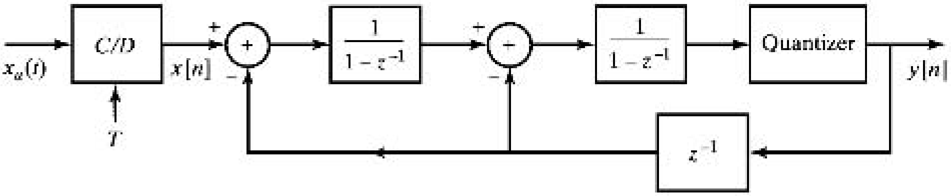


Figure 4.75 Oversampled quantizer with 2nd-order noise shaping.

TABLE 4.2 REDUCTION IN QUANTIZER BITS AS ORDER p OF NOISE SHAPING

Quantizer order p	Oversampling factor M				
	4	8	16	32	64
0	1.0	1.5	2.0	2.5	3.0
1	2.2	3.7	5.1	6.6	8.1
2	2.9	5.4	7.9	10.4	12.9
3	3.5	7.0	10.5	14.0	17.5
4	4.1	8.5	13.0	17.5	22.0
5	4.6	10.0	15.5	21.0	26.5

4.9.3 Oversampling and Noise Shaping in D/A Conversion

In Sections 4.9.1 and 4.9.2, we discussed the use of oversampling to simplify the process of A/D conversion. As we mentioned, the signal is initially oversampled to simplify antialias filtering and improve accuracy, but the final output $x_d[n]$ of the A/D converter is sampled at the Nyquist rate for $x_a(t)$. The minimum sampling rate is, of course, highly desirable for digital processing or for simply representing the analog signal in digital form, as in the CD audio recording system. It is natural to apply the same principles in reverse to achieve improvements in the D/A conversion process.

The basic system, which is the counterpart to Figure 4.65, is shown in Figure 4.76. The sequence $y_d[n]$, which is to be converted to a continuous-time signal, is first upsampled to produce the sequence $\hat{y}[n]$, which is then requantized before sending it to a D/A converter that accepts binary samples with the number of bits produced by the requantization process. We can use a simple D/A converter with few bits if we can be assured that the quantization noise does not occupy the signal band. Then the noise can be removed by inexpensive analog filtering.

In Figure 4.77, we show a structure for the quantizer that shapes the quantization noise in a similar manner to the 1^{st}-order noise shaping provided by the system in Figure 4.70. In our analysis we assume that $y_d[n]$ is effectively unquantized or so finely quantized relative to $y[n]$ that the primary source of quantizer error is the quantizer in Figure 4.76. To analyze the system in Figures 4.76 and 4.77, we replace the quantizer in Figure 4.77 by an additive white-noise source $e[n]$, so that Figure 4.77 is replaced by Figure 4.78. The transfer function from $\hat{y}[n]$ to $y[n]$ is unity, i.e., the upsampled signal

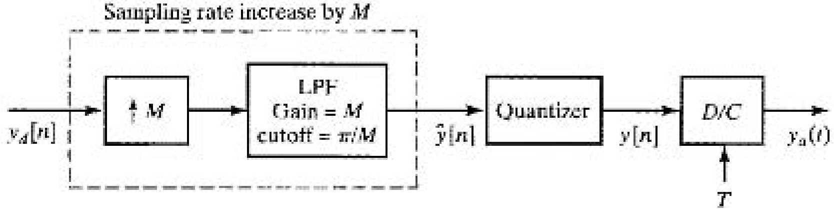


Figure 4.76 Oversampled D/A conversion.

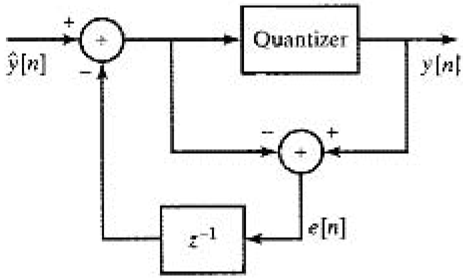


Figure 4.77 1st-order noise-shaping system for oversampled D/A quantization.

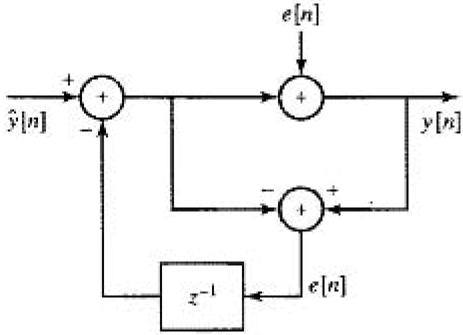


Figure 4.78 System of Figure 4.77 with quantizer replaced by linear noise model.

$\hat{y}[n]$ appears at the output unaltered. The transfer function $H_e(z)$ from $e[n]$ to $y[n]$ is

$$H_e(z) = 1 - z^{-1}.$$

Therefore, the quantization noise component $\tilde{e}[n]$ that appears at the output of the noise-shaping system in Figure 4.78 has the power density spectrum

$$\Phi_{\tilde{e}\tilde{e}}(e^{j\omega}) = \sigma_e^2(2\sin\omega/2)^2, \tag{4.181}$$

where, again, $\sigma_e^2 = \Delta^2/12$.

An illustration of this approach to D/A conversion is given in Figure 4.79. Figure 4.79(a) shows the power spectrum $\Phi_{y_d y_d}(e^{j\omega})$ of the input $y_d[n]$ in Figure 4.76. Note that we assume that the signal $y_d[n]$ is sampled at the Nyquist rate. Figure 4.79(b) shows the corresponding power spectrum at the output of the upsampler (by M), and Figure 4.79(c) shows the quantization noise spectrum at the output of the quantizer/noise-shaper system. Finally, Figure 4.79(d) shows the power spectrum of the signal component superimposed on the power spectrum of the noise component at the analog output of the D/C converter of Figure 4.76. In this case, we assume that the D/C converter has an ideal lowpass reconstruction filter with cutoff frequency $\pi/(MT)$, which will remove as much of the quantization noise as possible.

In a practical setting, we would like to avoid sharp-cutoff analog reconstruction filters. From Figure 4.79(d), it is clear that if we can tolerate somewhat more quantization noise, then the D/C reconstruction filter need not roll off so sharply. Furthermore, if we use multistage techniques in the noise shaping, we can obtain an output noise spectrum of the form

$$\Phi_{\tilde{e}\tilde{e}}(e^{j\omega}) = \sigma_e^2(2\sin\omega/2)^{2p},$$

which would push more of the noise to higher frequencies. In this case, the analog reconstruction filter specifications could be relaxed even further.

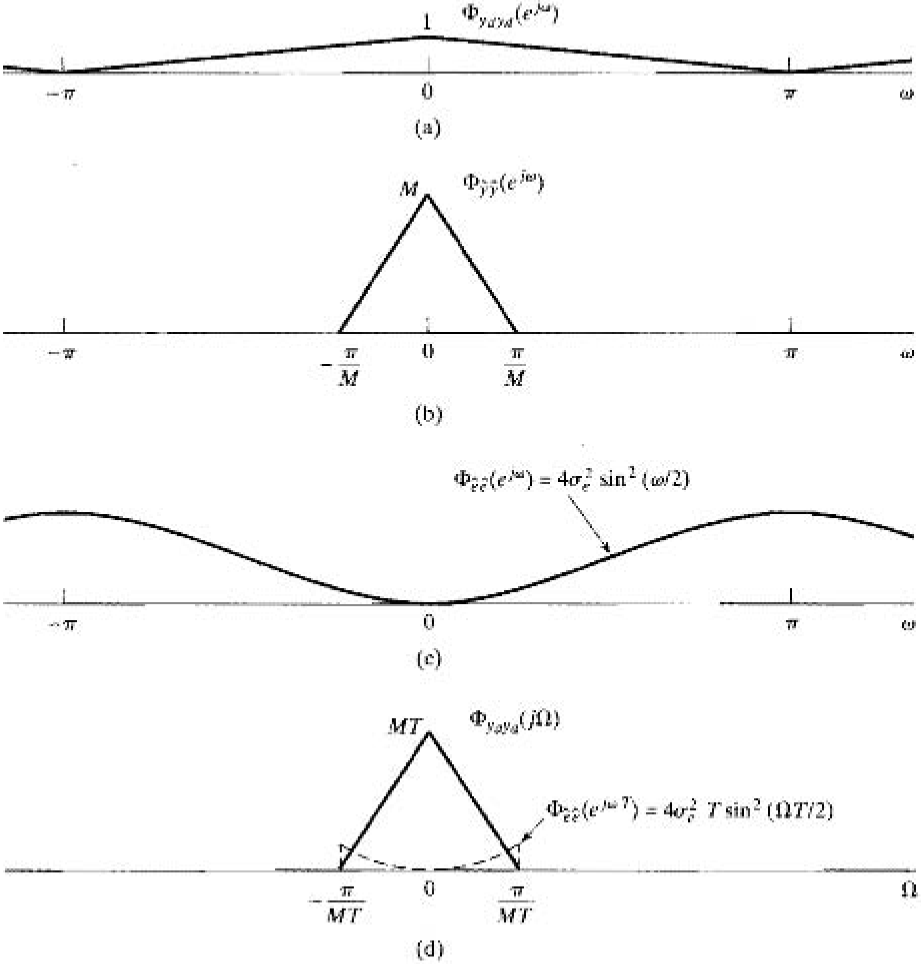


Figure 4.79 (a) Power spectral density of signal $y_d[n]$. (b) Power spectral density of signal $\tilde{y}[n]$. (c) Power spectral density of quantization noise. (d) Power spectral density of the continuous-time signal and the quantization noise.

4.10 SUMMARY

In this chapter, we developed and explored the relationship between continuous-time signals and the discrete-time sequences obtained by periodic sampling. The fundamental theorem that allows the continuous-time signal to be represented by a sequence of samples is the Nyquist-Shannon theorem, which states that, for a bandlimited signal, periodic samples are a sufficient representation, as long as the sampling rate is sufficiently high relative to the highest frequency in the continuous-time signal. Under this condition, the continuous-time signal can be reconstructed by lowpass filtering from knowledge of only the original bandwidth, the sampling rate and the samples. This corresponds to bandlimited interpolation. If the sampling rate is too low relative to the bandwidth of the signal, then aliasing distortion occurs and the original signal cannot be reconstructed by bandlimited interpolation.

The ability to represent signals by sampling permits the discrete-time processing of continuous-time signals. This is accomplished by first sampling, then applying discrete-time processing, and, finally, reconstructing a continuous-time signal from the result. Examples given were lowpass filtering and differentiation.

Sampling rate changes are an important class of digital signal processing operations. Downsampling a discrete-time signal corresponds in the frequency domain to an amplitude-scaled replication of the discrete-time spectrum and rescaling of the frequency axis, which may require additional bandlimiting to avoid aliasing. Upsampling corresponds to effectively increasing the sampling rate and is also represented in the frequency domain by a rescaling of the frequency axis. By combining upsampling and downsampling by integer amounts, noninteger sampling rate conversion can be achieved. We also showed how this can be efficiently done using multirate techniques.

In the final sections of the chapter, we explored a number of practical considerations associated with the discrete-time processing of continuous-time signals, including the use of prefiltering to avoid aliasing, quantization error in A/D conversion, and some issues associated with the filtering used in sampling and reconstructing the continuous-time signals. Finally, we showed how discrete-time decimation and interpolation and noise shaping can be used to simplify the analog side of A/D and D/A conversion.

The focus of this chapter has been on periodic sampling as a process for obtaining a discrete representation of a continuous-time signal. While such representations are by far the most common and are the basis for almost all of the topics to be discussed in the remainder of this text, there are other approaches to obtaining discrete representations that may lead to more compact representations for signals where other information (besides bandwidth) is known about the signal. Some examples can be found in Unser (2000).

Problems

Basic Problems with Answers

4.1. The signal

$$x_c(t) = \sin(2\pi(100)t)$$

was sampled with sampling period $T = 1/400$ second to obtain a discrete-time signal $x[n]$. What is the resulting sequence $x[n]$?

4.2. The sequence

$$x[n] = \cos\left(\frac{\pi}{4}n\right), \qquad -\infty < n < \infty,$$

was obtained by sampling the continuous-time signal

$$x_c(t) = \cos(\Omega_0 t), \qquad -\infty < t < \infty,$$

at a sampling rate of 1000 samples/s. What are two possible positive values of Ω_0 that could have resulted in the sequence $x[n]$?

4.3. The continuous-time signal

$$x_c(t) = \cos(4000\pi t)$$

is sampled with a sampling period T to obtain the discrete-time signal

$$x[n] = \cos\left(\frac{\pi n}{3}\right).$$

(a) Determine a choice for T consistent with this information.

(b) Is your choice for T in part (a) unique? If so, explain why. If not, specify another choice of T consistent with the information given.

4.4. The continuous-time signal

$$x_c(t) = \sin(20\pi t) + \cos(40\pi t)$$

is sampled with a sampling period T to obtain the discrete-time signal

$$x[n] = \sin\left(\frac{\pi n}{5}\right) + \cos\left(\frac{2\pi n}{5}\right).$$

(a) Determine a choice for T consistent with this information.

(b) Is your choice for T in part (a) unique? If so, explain why. If not, specify another choice of T consistent with the information given.

4.5. Consider the system of Figure 4.10, with the discrete-time system an ideal lowpass filter with cutoff frequency $\pi/8$ radians/s.

(a) If $x_c(t)$ is bandlimited to 5 kHz, what is the maximum value of T that will avoid aliasing in the C/D converter?

(b) If $1/T = 10$ kHz, what will the cutoff frequency of the effective continuous-time filter be?

(c) Repeat part (b) for $1/T = 20$ kHz.

4.6. Let $h_c(t)$ denote the impulse response of an LTI continuous-time filter and $h_d[n]$ the impulse response of an LTI discrete-time filter.

(a) If

$$h_c(t) = \begin{cases} e^{-at}, & t \geq 0, \\ 0, & t < 0, \end{cases}$$

where a is a positive real constant, determine the continuous-time filter frequency response and sketch its magnitude.

(b) If $h_d[n] = Th_c(nT)$ with $h_c(t)$ as in part (a), determine the discrete-time filter frequency response and sketch its magnitude.

(c) For a given value of a, determine, as a function of T, the minimum magnitude of the discrete-time filter frequency response.

4.7. A simple model of a multipath communication channel is indicated in Figure P4.7-1. Assume that $s_c(t)$ is bandlimited such that $S_c(j\Omega) = 0$ for $|\Omega| \geq \pi/T$ and that $x_c(t)$ is sampled with a sampling period T to obtain the sequence

$$x[n] = x_c(nT).$$

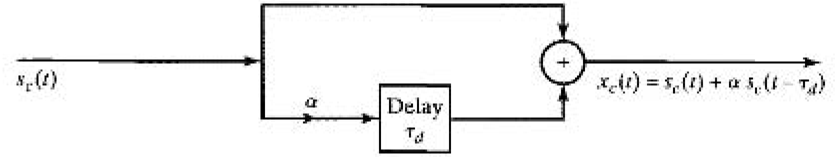


Figure P4.7-1

(a) Determine the Fourier transform of $x_c(t)$ and the Fourier transform of $x[n]$ in terms of $S_c(j\Omega)$.

(b) We want to simulate the multipath system with a discrete-time system by choosing $H(e^{j\omega})$ in Figure P4.7-2 so that the output $r[n] = x_c(nT)$ when the input is $s[n] = s_c(nT)$. Determine $H(e^{j\omega})$ in terms of T and τ_d.

(c) Determine the impulse response $h[n]$ in Figure P4.7 when (i) $\tau_d = T$ and (ii) $\tau_d = T/2$.

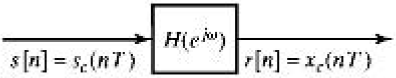


Figure P4.7-2

4.8. Consider the system in Figure P4.8 with the following relations:

$$X_c(j\Omega) = 0, \qquad |\Omega| \geq 2\pi \times 10^4,$$

$$x[n] = x_c(nT),$$

$$y[n] = T \sum_{k=-\infty}^{n} x[k].$$

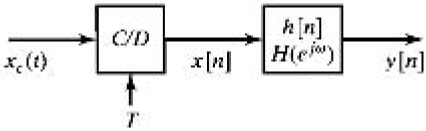


Figure P4.8

(a) For this system, what is the maximum allowable value of T if aliasing is to be avoided, i.e., so that $x_c(t)$ can be recovered from $x[n]$.

(b) Determine $h[n]$.

(c) In terms of $X(e^{j\omega})$, what is the value of $y[n]$ for $n \to \infty$?

(d) Determine whether there is any value of T for which

$$y[n]\Big|_{n=\infty} = \int_{-\infty}^{\infty} x_c(t)dt. \qquad \text{(P4.8-1)}$$

If there is such a value for T, determine the maximum value. If there is not, explain and specify how T would be chosen so that the equality in Eq. (P4.8-1) is best approximated.

4.9. Consider a stable discrete-time signal $x[n]$ whose discrete-time Fourier transform $X(e^{j\omega})$ satisfies the equation

$$X(e^{j\omega}) = X\left(e^{j(\omega-\pi)}\right)$$

and has even symmetry, i.e., $x[n] = x[-n]$.

(a) Show that $X(e^{j\omega})$ is periodic with a period π.

(b) Find the value of $x[3]$. (*Hint*: Find values for all odd-indexed points.)

(c) Let $y[n]$ be the decimated version of $x[n]$, i.e., $y[n] = x[2n]$. Can you reconstruct $x[n]$ from $y[n]$ for all n. If yes, how? If no, justify your answer.

4.10. Each of the following continuous-time signals is used as the input $x_c(t)$ for an ideal C/D converter as shown in Figure 4.1 with the sampling period T specified. In each case, find the resulting discrete-time signal $x[n]$.

(a) $x_c(t) = \cos(2\pi(1000)t)$, $T = (1/3000)$ sec
(b) $x_c(t) = \sin(2\pi(1000)t)$, $T = (1/1500)$ sec
(c) $x_c(t) = \sin(2\pi(1000)t)\,/\,(\pi t)$, $T = (1/5000)$ sec

4.11. The following continuous-time input signals $x_c(t)$ and corresponding discrete-time output signals $x[n]$ are those of an ideal C/D as shown in Figure 4.1. Specify a choice for the sampling period T that is consistent with each pair of $x_c(t)$ and $x[n]$. In addition, indicate whether your choice of T is unique. If not, specify a second possible choice of T consistent with the information given.

(a) $x_c(t) = \sin(10\pi t)$, $x[n] = \sin(\pi n/4)$
(b) $x_c(t) = \sin(10\pi t)/(10\pi t)$, $x[n] = \sin(\pi n/2)/(\pi n/2)$.

4.12. In the system of Figure 4.10, assume that

$$H(e^{j\omega}) = j\omega/T, \quad -\pi \le \omega < \pi,$$

and $T = 1/10$ sec.

(a) For each of the following inputs $x_c(t)$, find the corresponding output $y_c(t)$.

(i) $x_c(t) = \cos(6\pi t)$.
(ii) $x_c(t) = \cos(14\pi t)$.

(b) Are the outputs $y_c(t)$ those you would expect from a differentiator?

4.13. In the system shown in Figure 4.15, $h_c(t) = \delta(t - T/2)$.

(a) Suppose the input $x[n] = \sin(\pi n/2)$ and $T = 10$. Find $y[n]$.
(b) Suppose you use the same $x[n]$ as in part (a), but halve T to be 5. Find the resulting $y[n]$.
(c) In general, how does the continuous-time LTI system $h_c(t)$ limit the range of the sampling period T that can be used without changing $y[n]$?

4.14. Which of the following signals can be downsampled by a factor of 2 using the system in Figure 4.19 without any loss of information?

(a) $x[n] = \delta[n - n_0]$, for n_0 some unknown integer
(b) $x[n] = \cos(\pi n/4)$
(c) $x[n] = \cos(\pi n/4) + \cos(3\pi n/4)$
(d) $x[n] = \sin(\pi n/3)\,/(\pi n/3)$
(e) $x[n] = (-1)^n \sin(\pi n/3)\,/(\pi n/3)$.

4.15. Consider the system shown in Figure P4.15. For each of the following input signals $x[n]$, indicate whether the output $x_r[n] = x[n]$.

(a) $x[n] = \cos(\pi n/4)$
(b) $x[n] = \cos(\pi n/2)$
(c)

$$x[n] = \left[\frac{\sin(\pi n/8)}{\pi n}\right]^2$$

Hint: Use the modulation property of the Fourier transform to find $X(e^{j\omega})$.

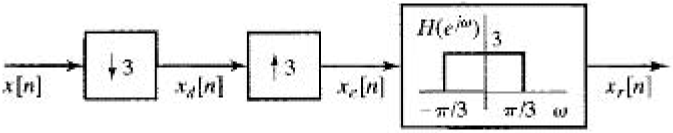


Figure P4.15

4.16. Consider the system in Figure 4.29. The input $x[n]$ and corresponding output $\tilde{x}_d[n]$ are given for a specific choice of M/L in each of the following parts. Determine a choice for M/L based on the information given, and specify whether your choice is unique.

(a) $x[n] = \sin(\pi n/3)/(\pi n/3)$, $\tilde{x}_d[n] = \sin(5\pi n/6)/(5\pi n/6)$

(b) $x[n] = \cos(3\pi n/4)$, $\tilde{x}_d[n] = \cos(\pi n/2)$.

4.17. Each of the following parts lists an input signal $x[n]$ and the upsampling and downsampling rates L and M for the system in Figure 4.29. Determine the corresponding output $\tilde{x}_d[n]$.

(a) $x[n] = \sin(2\pi n/3)/\pi n$, $L = 4$, $M = 3$

(b) $x[n] = \sin(3\pi n/4)$, $L = 6$, $M = 7$.

4.18. For the system shown in Figure 4.29, $X(e^{j\omega})$, the Fourier transform of the input signal $x[n]$, is shown in Figure P4.18. For each of the following choices of L and M, specify the maximum possible value of ω_0 such that $\tilde{X}_d(e^{j\omega}) = aX(e^{jM\omega/L})$ for some constant a.

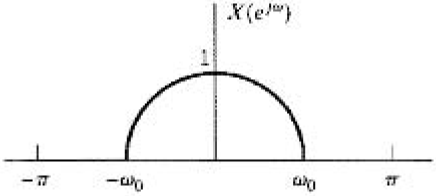


Figure P4.18

(a) $M = 3$, $L = 2$

(b) $M = 5$, $L = 3$

(c) $M = 2$, $L = 3$.

4.19. The continuous-time signal $x_c(t)$ with the Fourier transform $X_c(j\Omega)$ shown in Figure P4.19-1 is passed through the system shown in Figure P4.19-2. Determine the range of values for T for which $x_r(t) = x_c(t)$.

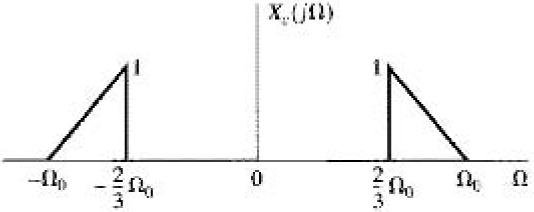


Figure P4.19-1

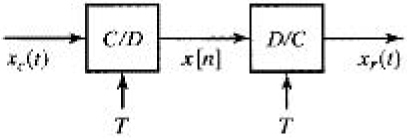


Figure P4.19-2

4.20. Consider the system in Figure 4.10. The input signal $x_c(t)$ has the Fourier transform shown in Figure P4.20 with $\Omega_0 = 2\pi(1000)$ radians/second. The discrete-time system is an ideal lowpass filter with frequency response

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| < \omega_c, \\ 0, & \text{otherwise.} \end{cases}$$

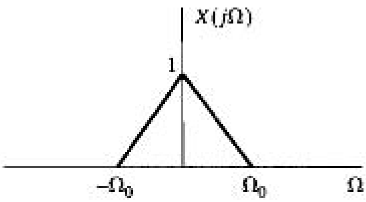


Figure P4.20

(a) What is the minimum sampling rate $F_s = 1/T$ such that no aliasing occurs in sampling the input?

(b) If $\omega_c = \pi/2$, what is the minimum sampling rate such that $y_r(t) = x_c(t)$?

Basic Problems

4.21. Consider a continuous-time signal $x_c(t)$ with Fourier transform $X_c(j\Omega)$ shown in Figure P4.21-1.

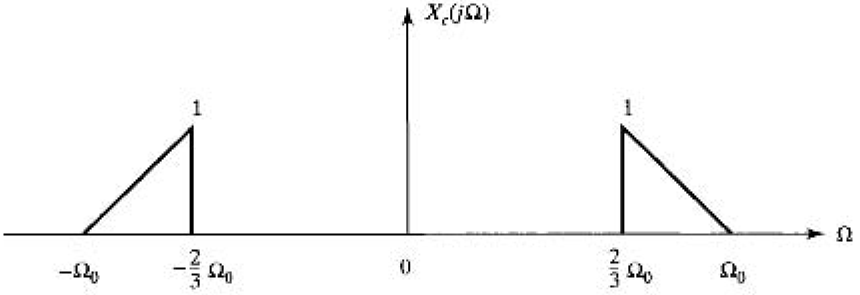


Figure P4.21-1 Fourier transform $X_c(j\Omega)$

(a) A continuous-time signal $x_r(t)$ is obtained through the process shown in Figure P4.21-2. First, $x_c(t)$ is multiplied by an impulse train of period T_1 to produce the waveform $x_s(t)$, i.e.,

$$x_s(t) = \sum_{n=-\infty}^{+\infty} x[n]\delta(t - nT_1).$$

Next, $x_s(t)$ is passed through a low pass filter with frequency response $H_r(j\Omega)$. $H_r(j\Omega)$ is shown in Figure P4.21-3.

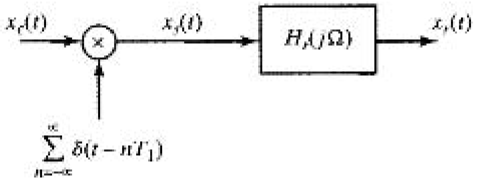


Figure P4.21-2 Conversion system for part (a)

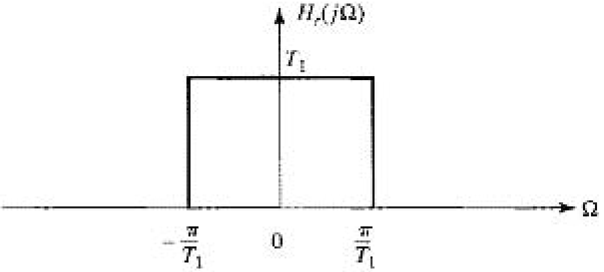


Figure P4.21-3 Frequency response $H_r(j\Omega)$

Determine the range of values for T_1 for which $x_r(t) = x_c(t)$.

(b) Consider the system in Figure P4.21-4. The system in this case is the same as the one in part (a), except that the sampling period is now T_2. The system $H_s(j\Omega)$ is some continuous-time ideal LTI filter. We want $x_o(t)$ to be equal to $x_c(t)$ for all t, i.e., $x_o(t) = x_c(t)$ for some choice of $H_s(j\Omega)$. Find all values of T_2 for which $x_o(t) = x_c(t)$ is possible. For the largest T_2 you determined that would still allow recovery of $x_c(t)$, choose $H_s(j\Omega)$ so that $x_o(t) = x_c(t)$. Sketch $H_s(j\Omega)$.

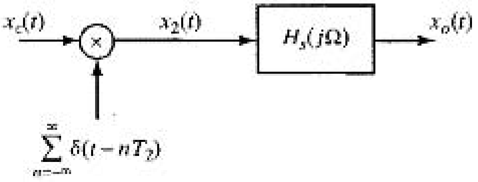


Figure P4.21-4 Conversion system for part (b)

4.22. Suppose that the bandlimited differentiator of Example 4.4 has input $x_c(t) = \cos(\Omega_0 t)$ with $\Omega_0 < \pi/T$. In this problem, we wish to verify that the continuous-time signal reconstructed from the output of the bandlimited differentiator is indeed the derivative of $x_c(t)$.

(a) The sampled input will be $x[n] = \cos(\omega_0 n)$, where $\omega_0 = \Omega_0 T < \pi$. Determine an expression for $X(e^{j\omega})$ that is valid for $|\omega| \leq \pi$.

(b) Now use Eq. (4.46) to determine the DTFT of $Y(e^{j\omega})$, the output of the discrete-time system.

(c) From Eq. (4.32) determine $Y_r(j\Omega)$, the continuous-time Fourier transform of the output of the D/C converter.

(d) Use the result of (c) to show that

$$y_r(t) = -\Omega_0 \sin(\Omega_0 t) = \frac{d}{dt}[x_c(t)].$$

4.23. Figure P4.23-1 shows a continuous-time filter that is implemented using an LTI discrete-time filter ideal lowpass filter with frequency response over $-\pi \leq \omega \leq \pi$ as

$$H(e^{j\omega}) = \begin{cases} 1 & |\omega| < \omega_c \\ 0 & \omega_c < |\omega| \leq \pi. \end{cases}$$

(a) If the continuous-time Fourier transform of $x_c(t)$, namely $X_c(j\Omega)$, is as shown in Figure P4.23-2 and $\omega_c = \frac{\pi}{5}$, sketch and label $X(e^{j\omega})$, $Y(e^{j\omega})$ and $Y_c(j\Omega)$ for each of the following cases:

(i) $1/T_1 = 1/T_2 = 2 \times 10^4$
(ii) $1/T_1 = 4 \times 10^4,\ \ 1/T_2 = 10^4$
(iii) $1/T_1 = 10^4,\ \ 1/T_2 = 3 \times 10^4$.

(b) For $1/T_1 = 1/T_2 = 6 \times 10^3$, and for input signals $x_c(t)$ whose spectra are bandlimited to $|\Omega| < 2\pi \times 5 \times 10^3$ (but otherwise unconstrained), what is the maximum choice of the cutoff frequency ω_c of the filter $H(e^{j\omega})$ for which the overall system is LTI? For this maximum choice of ω_c, specify $H_c(j\Omega)$.

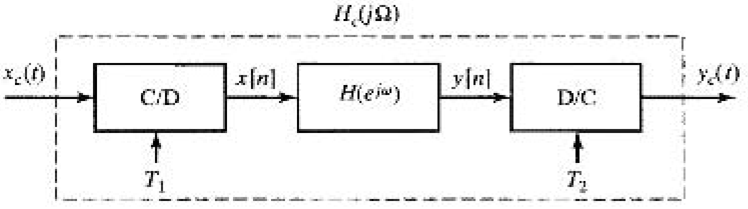


Figure P4.23-1

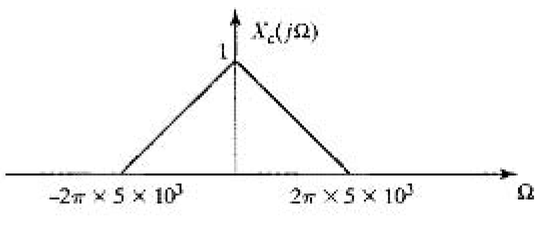


Figure P4.23-2

4.24. Consider the system shown in Figure P4.24-1.

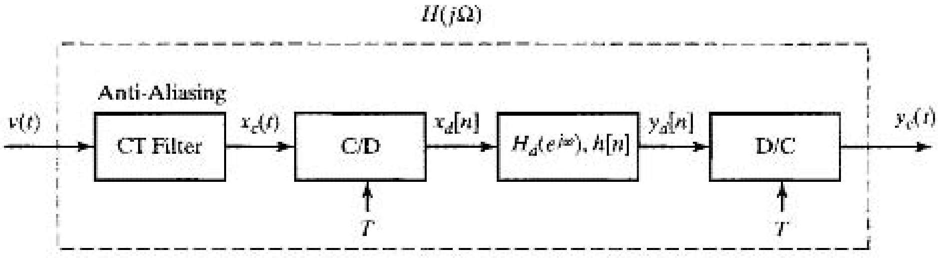


Figure P4.24-1

The anti-aliasing filter is a continuous-time filter with the frequency response $L(j\Omega)$ shown in Figure P4.24-2.

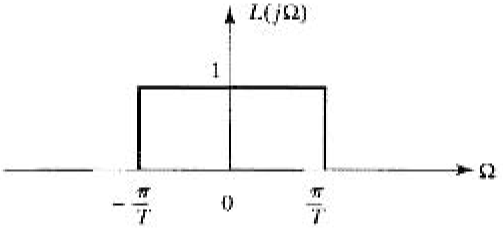


Figure P4.24-2

The frequency response of the LTI discrete-time system between the converters is given by:

$$H_d(e^{j\omega}) = e^{-j\frac{\omega}{3}}, \quad |\omega| < \pi$$

(a) What is the effective continuous-time frequency response of the overall system, $H(j\Omega)$?

(b) Choose the most accurate statement:

(i) $y_c(t) = \frac{d}{dt}x_c(3t)$.

(ii) $y_c(t) = x_c(t - \frac{T}{3})$.

(iii) $y_c(t) = \frac{d}{dt}x_c(t - 3T)$.

(iv) $y_c(t) = x_c(t - \frac{1}{3})$.

(a) Express $y_d[n]$ in terms of $y_c(t)$.

(b) Determine the impulse response $h[n]$ of the discrete-time LTI system.

4.25. Two bandlimited signals, $x_1(t)$ and $x_2(t)$, are multiplied, producing the product signal $w(t) = x_1(t)x_2(t)$. This signal is sampled by a periodic impulse train yielding the signal

$$w_p(t) = w(t)\sum_{n=-\infty}^{\infty}\delta(t - nT) = \sum_{n=-\infty}^{\infty} w(nT)\delta(t - nT).$$

Assume that $x_1(t)$ is bandlimited to Ω_1, and $x_2(t)$ is bandlimited to Ω_2; that is,

$$X_1(j\Omega) = 0, \quad |\Omega| \geq \Omega_1$$

$$X_2(j\Omega) = 0, \quad |\Omega| \geq \Omega_2.$$

Determine the *maximum* sampling interval T such that $w(t)$ is recoverable from $w_p(t)$ through the use of an ideal lowpass filter.

4.26. The system of Figure P4.26 is to be used to filter continuous time music signals using a sampling rate of 16kHz.

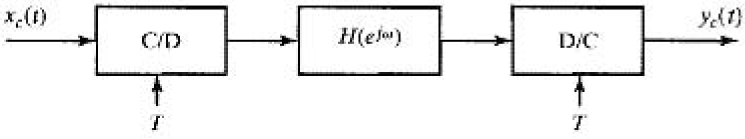


Figure P4.26

$H(e^{j\omega})$ is an ideal lowpass filter with a cutoff of $\pi/2$. If the input has been bandlimited such that $X_c(j\Omega) = 0$ for $|\Omega| > \Omega_c$, how should Ω_c be chosen so that the overall system in Figure P4.26 is LTI?

4.27. The system shown in Figure P4.27 is intended to approximate a differentiator for bandlimited continuous-time input waveforms.

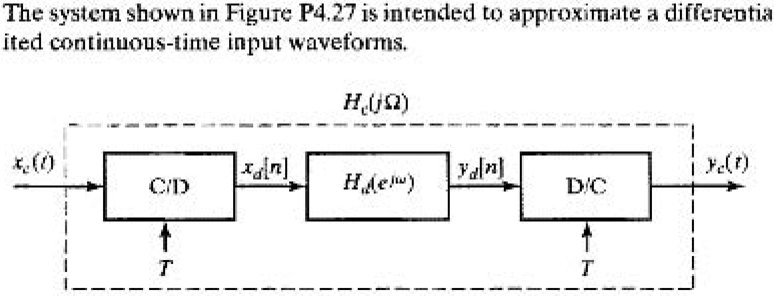


Figure P4.27

- The continuous-time input signal $x_c(t)$ is bandlimited to $|\Omega| < \Omega_M$.
- The C/D converter has sampling rate $T = \dfrac{\pi}{\Omega_M}$, and produces the signal $x_d[n] = x_c(nT)$.
- The discrete-time filter has frequency response

$$H_d(e^{j\omega}) = \frac{e^{j\omega/2} - e^{-j\omega/2}}{T}, \qquad |\omega| \le \pi .$$

- The ideal D/C converter is such that $y_d[n] = y_c(nT)$.

(a) Find the continuous-time frequency response $H_c(j\Omega)$ of the end-to-end system.

(b) Find $x_d[n]$, $y_c(t)$, and $y_d[n]$, when the input signal is

$$x_c(t) = \frac{\sin(\Omega_M t)}{\Omega_M t} .$$

4.28. Consider the representation of the process of sampling followed by reconstruction shown in Figure P4.28.

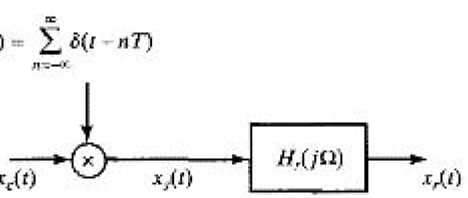


Figure P4.28

Assume that the input signal is

$$x_c(t) = 2\cos(100\pi t - \pi/4) + \cos(300\pi t + \pi/3) \qquad -\infty < t < \infty$$

The frequency response of the reconstruction filter is

$$H_r(j\Omega) = \begin{cases} T & |\Omega| \le \pi/T \\ 0 & |\Omega| > \pi/T \end{cases}$$

(a) Determine the continuous-time Fourier transform $X_c(j\Omega)$ and plot it as a function of Ω.

(b) Assume that $f_s = 1/T = 500$ samples/sec and plot the Fourier transform $X_s(j\Omega)$ as a function of Ω for $-2\pi/T \le \Omega \le 2\pi/T$. What is the output $x_r(t)$ in this case? (You should be able to give an exact equation for $x_r(t)$.)

(c) Now, assume that $f_s = 1/T = 250$ samples/sec. Repeat part (b) for this condition.

(d) Is it possible to choose the sampling rate so that

$$x_r(t) = A + 2\cos(100\pi t - \pi/4)$$

where A is a constant? If so, what is the sampling rate $f_s = 1/T$, and what is the numerical value of A?

4.29. In Figure P4.29, assume that $X_c(j\Omega) = 0$, $|\Omega| \geq \pi/T_1$. For the general case in which $T_1 \neq T_2$ in the system, express $y_c(t)$ in terms of $x_c(t)$. Is the basic relationship different for $T_1 > T_2$ and $T_1 < T_2$?

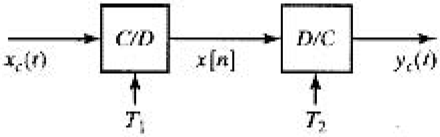


Figure P4.29

4.30. In the system of Figure P4.30, $X_c(j\Omega)$ and $H(e^{j\omega})$ are as shown. Sketch and label the Fourier transform of $y_c(t)$ for each of the following cases:

(a) $1/T_1 = 1/T_2 = 10^4$
(b) $1/T_1 = 1/T_2 = 2 \times 10^4$
(c) $1/T_1 = 2 \times 10^4$, $1/T_2 = 10^4$
(d) $1/T_1 = 10^4$, $1/T_2 = 2 \times 10^4$.

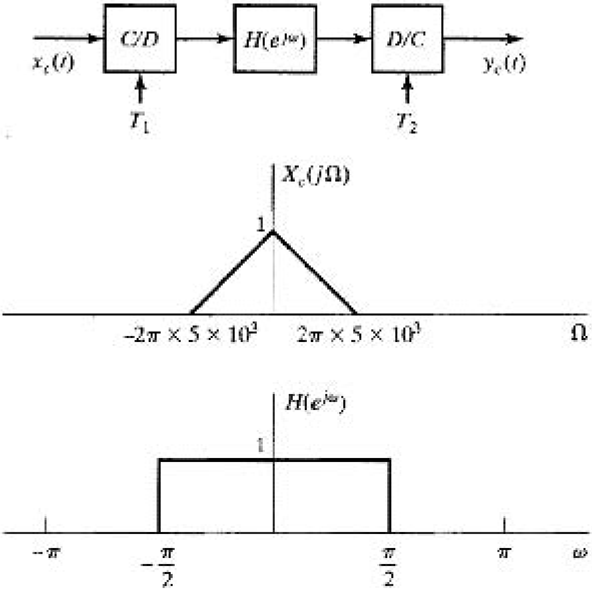


Figure P4.30

4.31. Figure P4.31-1 shows the overall system for filtering a continuous-time signal using a discrete-time filter. The frequency responses of the reconstruction filter $H_r(j\Omega)$ and the discrete-time filter $H(e^{j\omega})$ are shown in Figure P4.31-2.

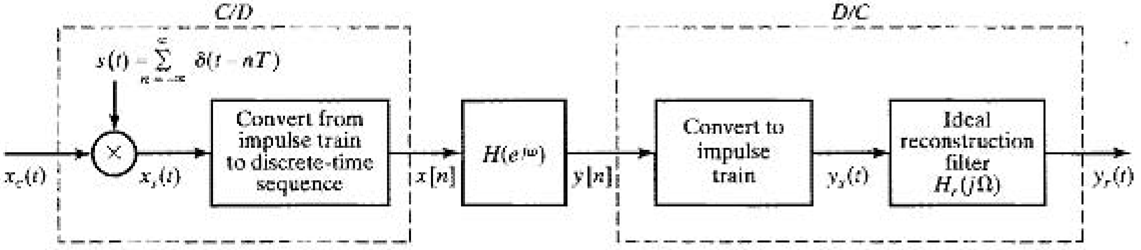

C/D
$s(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT)$
$x_c(t)$
$x_s(t)$
Convert from impulse train to discrete-time sequence
$x[n]$
$H(e^{j\omega})$
$y[n]$
D/C
Convert to impulse train
$y_s(t)$
Ideal reconstruction filter $H_r(j\Omega)$
$y_r(t)$

Figure P4.31-1

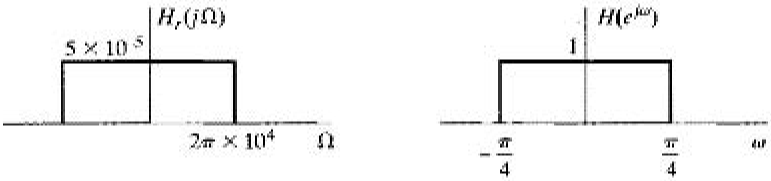


Figure P4.31-2

(a) For $X_c(j\Omega)$ as shown in Figure P4.31-3 and $1/T=20$ kHz, sketch $X_s(j\Omega)$ and $X(e^{j\omega})$.

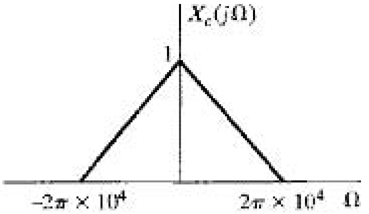


Figure P4.31-3

For a certain range of values of T, the overall system, with input $x_c(t)$ and output $y_c(t)$, is equivalent to a continuous-time lowpass filter with frequency response $H_{eff}(j\Omega)$ sketched in Figure P4.31-4.

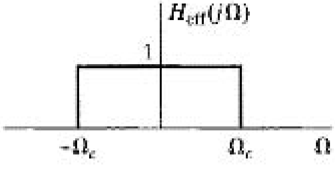


Figure P4.31-4

(b) Determine the range of values of T for which the information presented in (a) is true when $X_c(j\Omega)$ is bandlimited to $|\Omega|\le 2\pi\times10^4$ as shown in Figure P4.31-3.

(c) For the range of values determined in (b), sketch Ω_c as a function of $1/T$.

Note: This is one way of implementing a variable-cutoff continuous-time filter using fixed continuous-time and discrete-time filters and a variable sampling rate.

4.32. Consider the discrete-time system shown in Figure P4.32-1

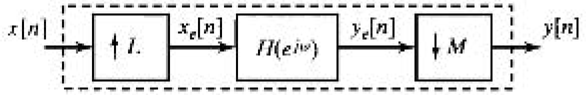


Figure P4.32-1

where

(i) L and M are positive integers.

(ii) $x_e[n] = \begin{cases} x[n/L] & n = kL, \quad k \text{ is any integer} \\ 0 & \text{otherwise.} \end{cases}$

(iii) $y[n] = y_e[nM]$.

(iv) $H(e^{j\omega}) = \begin{cases} M & |\omega| \le \frac{\pi}{4} \\ 0 & \frac{\pi}{4} < |\omega| \le \pi \end{cases}$.

(a) Assume that $L = 2$ and $M = 4$, and that $X(e^{j\omega})$, the DTFT of $x[n]$, is real and is as shown in Figure P4.32-2. Make an appropriately labeled sketch of $X_e(e^{j\omega})$, $Y_e(e^{j\omega})$, and $Y(e^{j\omega})$, the DTFTs of $x_e[n]$, $y_e[n]$, and $y[n]$, respectively. Be sure to clearly label salient amplitudes and frequencies.

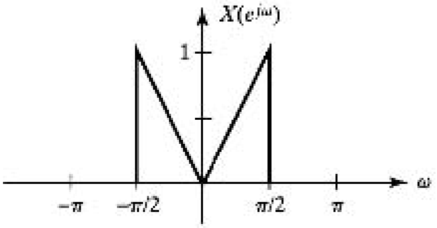


Figure P4.32-2

(b) Now assume $L = 2$ and $M = 8$. Determine $y[n]$ in this case.
Hint: See which diagrams in your answer to part (a) change.

4.33. For the system shown in Figure P4.33, find an expression for $y[n]$ in terms of $x[n]$. Simplify the expression as much as possible.

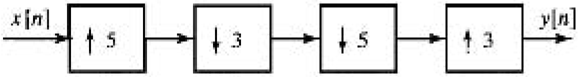


Figure P4.33

Advanced Problems

4.34. In the system shown in Figure P4.34, the individual blocks are defined as indicated.

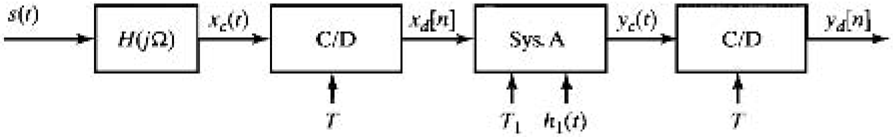


Figure P4.34

$$H(j\Omega): \quad H(j\Omega) = \begin{cases} 1, & |\Omega| < \pi \cdot 10^{-3} \text{ rad/sec} \\ 0, & |\Omega| > \pi \cdot 10^{-3} \text{ rad/sec} \end{cases}$$

$$\text{System A:} \quad y_c(t) = \sum_{k=-\infty}^{\infty} x_d[k] h_1(t - kT_1)$$

$$\text{Second C/D:} \quad y_d[n] = y_c(nT)$$

(a) Specify a choice for T, T_1, and $h_1(t)$ so that $y_c(t)$ and $x_c(t)$ are guaranteed to be equal for any choice of $s(t)$.

(b) State whether your choice in (a) is unique or whether there are other choices for T, T_1, and $h_1(t)$ that will guarantee that $y_c(t)$ and $x_c(t)$ are equal. As usual, clearly show your reasoning.

(c) For this part, we are interested in what is often referred to as *consistent resampling*. Specifically, the system A constructs a continuous-time signal $y_c(t)$ from $x_d[n]$ the sequence of samples of $x_c(t)$ and is then resampled to obtain $y_d[n]$. The resampling is referred to as consistent if $y_d[n] = x_d[n]$. Determine the most general conditions you can on T, T_1, and $h_1(t)$ so that $y_d[n] = x_d[n]$.

4.35. Consider the system shown in Figure P4.35-1.

For parts (a) and (b) only, $X_c(j\Omega) = 0$ for $|\Omega| > 2\pi \times 10^3$ and $H(e^{j\omega})$ is as shown in Figure P4.35-2 (and of course periodically repeats).

(a) Determine the most general condition on T, if any, so that the overall continuous-time system from $x_c(t)$ to $y_c(t)$ is LTI.

(b) Sketch and clearly label the overall equivalent continuous-time frequency response $H_{\text{eff}}(j\Omega)$ that results when the condition determined in (a) holds.

(c) **For this part only** assume that $X_c(j\Omega)$ in Figure P4.35-1 is bandlimited to avoid aliasing, i.e., $X_c(j\Omega) = 0$ for $|\Omega| \geq \frac{\pi}{T}$. For a general sampling period T, we would like to choose the system $H(e^{j\omega})$ in Figure P4.35-1 so that the overall continuous-time system from $x_c(t)$ to $y_c(t)$ is LTI for any input $x_c(t)$ bandlimited as above. Determine the most general conditions on $H(e^{j\omega})$, if any, so that the overall CT system is LTI. Assuming that these conditions hold, also specify in terms of $H(e^{j\omega})$ the overall equivalent continuous-time frequency response $H_{\text{eff}}(j\Omega)$.

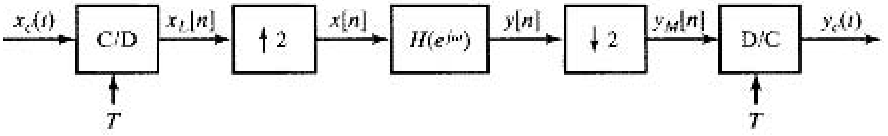


Figure P4.35-1

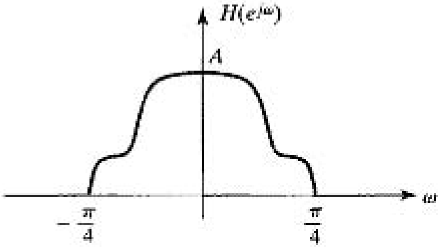


Figure P4.35-2

4.36. We have a discrete-time signal, $x[n]$, arriving from a source at a rate of $\frac{1}{T_1}$ samples per second. We want to digitally resample it to create a signal $y[n]$ that has $\frac{1}{T_2}$ samples per second, where $T_2 = \frac{3}{5}T_1$.

(a) Draw a block diagram of a discrete-time system to perform the resampling. Specify the input/output relationship for all the boxes in the Fourier domain.

(b) For an input signal $x[n] = \delta[n] = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise}, \end{cases}$ determine y[n].

4.37. Consider the decimation filter structure shown in Figure P4.37-1:

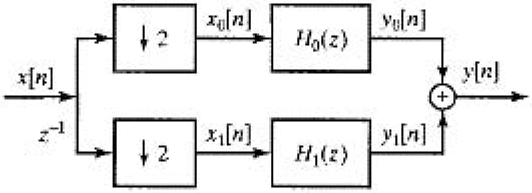


Figure P4.37-1

where $y_0[n]$ and $y_1[n]$ are generated according to the following difference equations:

$$y_0[n] = \frac{1}{4}y_0[n-1] - \frac{1}{3}x_0[n] + \frac{1}{8}x_0[n-1]$$

$$y_1[n] = \frac{1}{4}y_1[n-1] + \frac{1}{12}x_1[n]$$

(a) How many multiplies per output sample does the implementation of the filter structure require? Consider a divide to be equivalent to a multiply.

The decimation filter can also be implemented as shown in Figure P4.37-2,

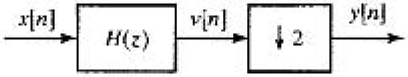


Figure P4.37-2

where $v[n] = av[n-1] + bx[n] + cx[n-1]$.

(b) Determine a, b, and c.

(c) How many multiplies per output sample does this second implementation require?

4.38. Consider the two systems of Figure P4.38.

(a) For $M = 2$, $L = 3$, and any arbitrary $x[n]$, will $y_A[n] = y_B[n]$? If your answer is yes, justify your answer. If your answer is no, clearly explain or give a counterexample.

(b) How must M and L be related to guarantee $y_A[n] = y_B[n]$ for arbitrary $x[n]$?

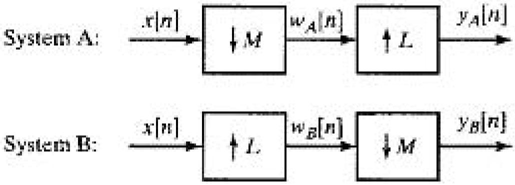


Figure P4.38

4.39. In system A, a continuous-time signal $x_c(t)$ is processed as indicated in Figure P4.39-1.

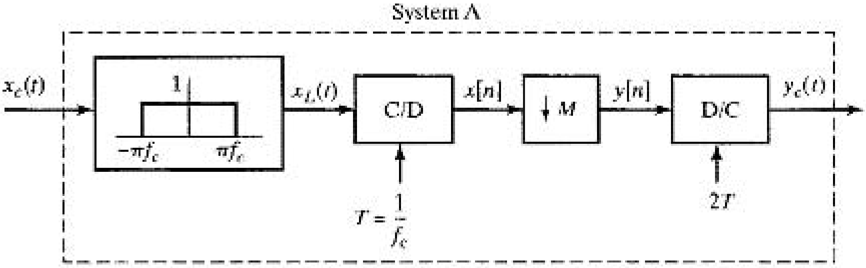


Figure P4.39-1

(a) If $M = 2$ and $x_c(t)$ has the Fourier transform shown in Figure P4.39-2, determine $y[n]$. Clearly show your work on this part.

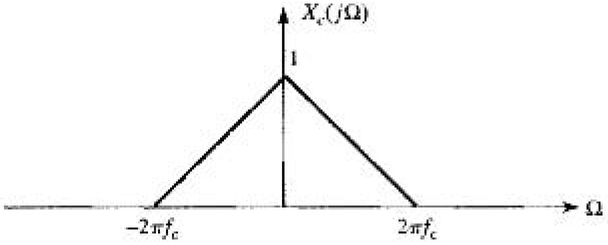


Figure P4.39-2

We would now like to modify system A by appropriately placing additional processing modules in the cascade chain of system A (i.e., blocks can be added at any point in the cascade chain—at the beginning, at the end, or even in between existing blocks). All of the current blocks in system A must be kept. We would like the modified system to be an ideal LTI lowpass filter, as indicated in Figure P4.39-3.

$x_c(t)$
$H(j\Omega)$
$y_c(t)$

Figure P4.39-3

$$H(j\Omega) = \begin{cases} 1 & |\Omega| < \frac{2\pi f_c}{5} \\ 0 & \text{otherwise} \end{cases}$$

We have available an unlimited number of the six modules specified in the table given in Figure P4.39-4. The per unit cost for each module is indicated, and we would like the final cost to be as low as possible. **Note that the D/C converter is running at a rate of "$2T$".**

(b) Design the lowest-cost modified system if $M = 2$ in System A. Specify the parameters for all the modules used.

(c) Design the lowest-cost modified system if $M = 4$ in System A. Specify the parameters for all the modules used.

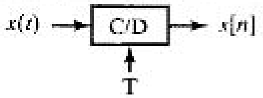
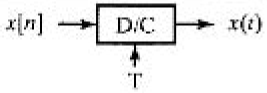
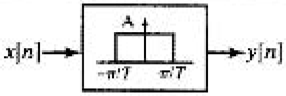
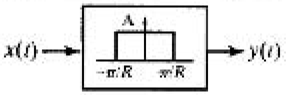
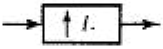
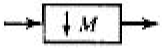
	**Continuous to Discrete-Time Converter** Parameters: T Cost : 10
	**Discrete to Continuous Time-Converter** Parameters: T Cost : 10
	**Discrete-Time Lowpass Filter** Parameters: A, T Cost : 10
	**Continuous-Time Lowpass Filter** Parameters: A, R Cost : 20
	**Expander** Parameters: L Cost : 5
	**Compressor** Parameters: M Cost : 5

Figure P4.39-4

4.40. Consider the discrete-time system shown in Figure P4.40-1.

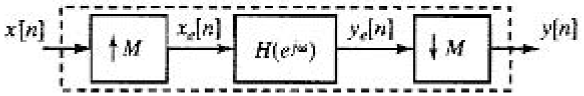


Figure P4.40-1

where

(i) M is an integer.

(ii) $x_e[n] = \begin{cases} x[n/M] & n = kM, \quad k \text{ is any integer} \\ 0 & \text{otherwise.} \end{cases}$

(iii) $y[n] = y_e[nM]$.

(iv) $H(e^{j\omega}) = \begin{cases} M & |\omega| \leq \frac{\pi}{4} \\ 0 & \frac{\pi}{4} < |\omega| \leq \pi\ . \end{cases}$

(a) Assume that $M = 2$ and that $X(e^{j\omega})$, the DTFT of $x[n]$, is real and is as shown in Figure P4.40-2. Make an appropriately labeled sketch of $X_e(e^{j\omega})$, $Y_e(e^{j\omega})$, and $Y(e^{j\omega})$, the DTFTs of $x_e[n]$, $y_e[n]$, and $y[n]$, respectively. Be sure to clearly label salient amplitudes and frequencies.

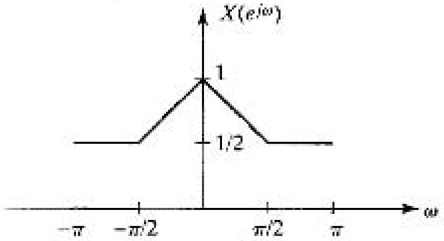


Figure P4.40-2

(b) For $M = 2$ and $X(e^{j\omega})$ as given in Figure P4.40-2, find the value of

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

(c) For $M = 2$, the overall system is LTI. Determine and sketch the magnitude of the frequency response of the overall system $|H_{\text{eff}}(e^{j\omega})|$.

(d) For $M = 6$, the overall system is still LTI. Determine and sketch the magnitude of the overall system's frequency response $|H_{\text{eff}}(e^{j\omega})|$.

4.41. **(a)** Consider the system in Figure P4.41-1 where a filter $H(z)$ is followed by a compressor. Suppose that $H(z)$ has an impulse response given by:

$$h[n] = \begin{cases} (\frac{1}{2})^n, & 0 \le n \le 11 \\ 0, & \text{otherwise.} \end{cases} \tag{P4.41-1}$$

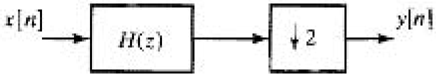


Figure P4.41-1

The efficiency of this system can be improved by implementing the filter $H(z)$ and the compressor using a polyphase decomposition. Draw an efficient polyphase structure for this system with two polyphase components. Please specify the filters you use.

(b) Now consider the system in Figure P4.41-2 where a filter $H(z)$ is preceded by an expander. Suppose that $H(z)$ has the impulse response as given in Eq. (P4.41-1).

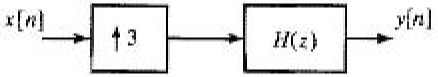
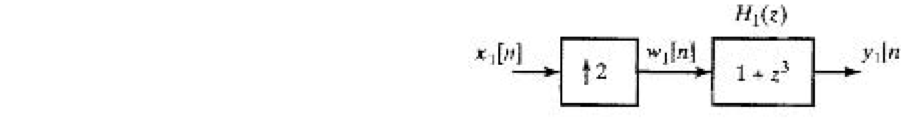


Figure P4.41-2

The efficiency of this system can be improved by implementing the expander and filter $H(z)$ using a polyphase decomposition. Draw an efficient polyphase structure for this system with three polyphase components. Please specify the filters you use.

4.42. For the systems shown in Figure P4.42-1 and Figure P4.42-2, determine whether or not it is possible to specify a choice for $H_2(z)$ in System 2 so that $y_2[n] = y_1[n]$ when $x_2[n] = x_1[n]$ and $H_1(z)$ is as specified. If it is possible, specify $H_2(z)$. If it is not possible, clearly explain.

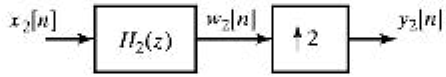
System 1:



$$w_1[n] = \begin{cases} x_1[n/2] & , \ n/2 \text{ integer} \\ 0 & , \ \text{otherwise} \end{cases}$$

Figure P4.42-1

System 2:



$$y_2[n] = \begin{cases} w_2[n/2] & , \ n/2 \text{ integer} \\ 0 & , \ \text{otherwise} \end{cases}$$

Figure P4.42-2

4.43. The block diagram in Figure P4.43 represents a system that we would like to implement. Determine a block diagram of an equivalent system consisting of a cascade of LTI systems, compressor blocks, and expander blocks which results in the minimum number of multiplications per output sample.

Note: By "equivalent system," we mean that it produces the same output sequence for any given input sequence.

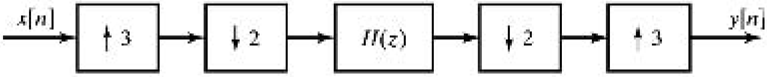
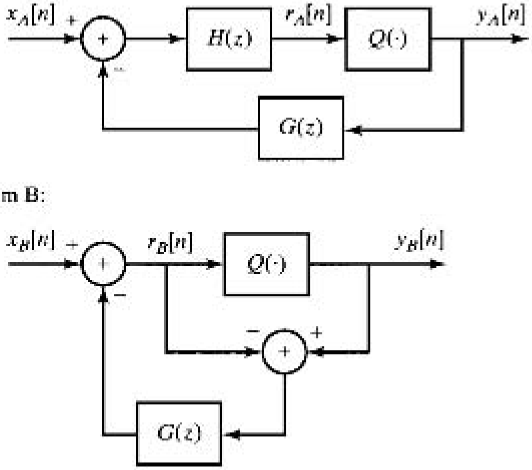


Figure P4.43

$$H(z) = \frac{z^{-6}}{7 + z^{-6} - 2z^{-12}}$$

4.44. Consider the two systems shown in Figure P4.44.

System A:



System B:

Figure P4.44

where $Q(\cdot)$ represents a quantizer which is the same in both systems. For any given $G(z)$, can $H(z)$ always be specified so that the two systems are equivalent (i.e., $y_A[n] = y_B[n]$ when $x_A[n] = x_B[n]$) for any arbitrary quantizer $Q(\cdot)$? If so, specify $H(z)$. If not, clearly explain your reasoning.

4.45. The quantizer $Q(\cdot)$ in the system S_1 (Figure P4.45-1) can be modeled with an additive noise. Figure P4.45-2 shows system S_2, which is a model for system S_1

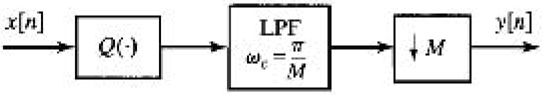


Figure P4.45-1 System S_1

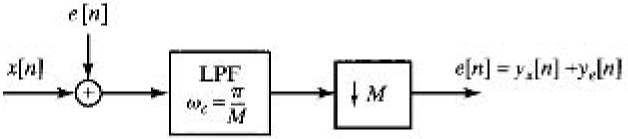


Figure P4.45-2 System S_2

The input $x[n]$ is a zero-mean, wide-sense stationary random process with power spectral density $\Phi_{xx}(e^{j\omega})$ which is bandlimited to π/M and we have $E\left[x^2[n]\right] = 1$. The additive noise $e[n]$ is wide-sense stationary white noise with zero mean and variance σ_e^2. Input and additive noise are uncorrelated. The frequency response of the low-pass filter in all the diagrams has a unit gain.

(a) For system S_2 find the signal to noise ratio: $SNR = 10 \log \frac{E[y_x^2[n]]}{E[y_e^2[n]]}$. Note that $y_x[n]$ is the output due to $x[n]$ alone and $y_e[n]$ is the output due to $e[n]$ alone.

(b) To improve the SNR owing to quantization, the system of Figure P4.45-3 is proposed:

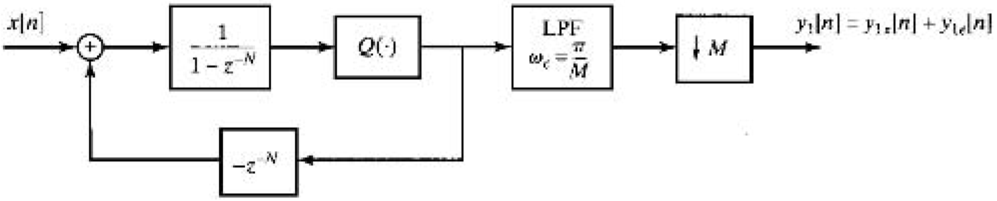


Figure P4.45-3

where $N > 0$ is an integer such that $\pi N << M$. Replace the quantizer with the additive model, as in Figure P4.45-4. Express $y_{1x}[n]$ in terms of $x[n]$ and $y_{1e}[n]$ in terms of $e[n]$.

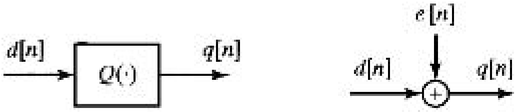


Figure P4.45-4

(c) Assume that $e[n]$ is a zero mean wide-sense stationary white noise that is uncorrelated with input $x[n]$. Is $y_{1e}[n]$ a wide-sense stationary signal? How about $y_1[n]$? Explain.

(d) Is the proposed method in part (b) improving the SNR? For which value of N is the SNR of the system in part (b) maximized?

4.46. The following are three proposed identities involving compressors and expanders. For each, state whether or not the proposed identity is valid. If your answer is that it is valid, explicitly show why. If your answer is no, explicitly give a simple counterexample.

(a) Proposed identity (a):

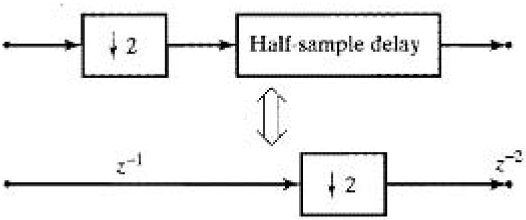


Figure P4.46-1

(b) Proposed identity (b):

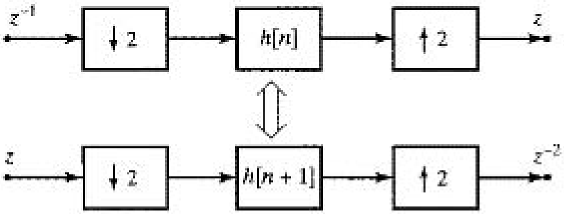


Figure P4.46-2

(c) Proposed identity (c):

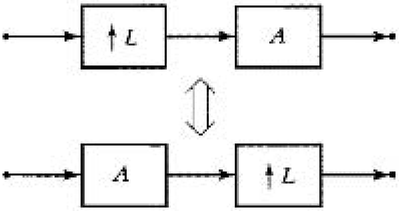
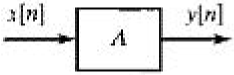


Figure P4.46-3

where L is a positive integer, and A is defined in terms of $X(e^{j\omega})$ and $Y(e^{j\omega})$ (the respective DTFTs of A's input and output) as:



$$Y(e^{j\omega}) = \left(X(e^{j\omega})\right)^L$$

Figure P4.46-4

4.47. Consider the system shown in Figure P4.47-1 for discrete-time processing of the continuous-time input signal $g_c(t)$.

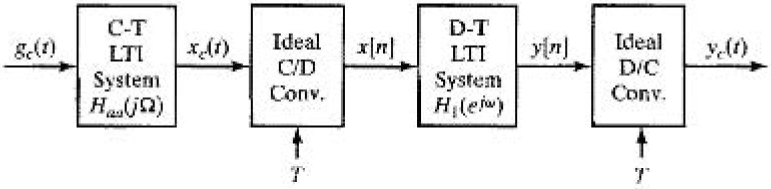


Figure P4.47-1

The continuous-time input signal to the overall system is of the form $g_c(t) = f_c(t) + e_c(t)$ where $f_c(t)$ is considered to be the "signal" component and $e_c(t)$ is considered to be an "additive noise" component. The Fourier transforms of $f_c(t)$ and $e_c(t)$ are as shown in Figure P4.47-2.

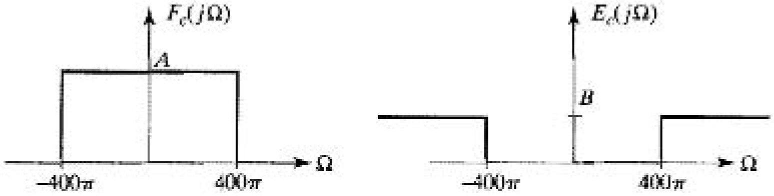


Figure P4.47-2

Since the total input signal $g_c(t)$ does not have a bandlimited Fourier transform, a zero-phase continuous-time antialiasing filter is used to combat aliasing distortion. Its frequency response is given in Figure P4.47-3.

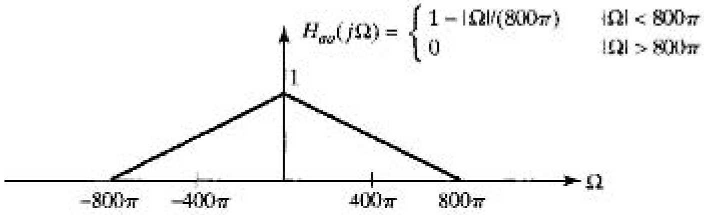


Figure P4.47-3

(a) If in Figure P4.47-1 the sampling rate is $2\pi/T = 1600\pi$, and the discrete-time system has frequency response

$$H_1(e^{j\omega}) = \begin{cases} 1 & |\omega| < \pi/2 \\ 0 & \pi/2 < |\omega| \le \pi \end{cases}$$

sketch the Fourier transform of the continuous-time output signal for the input whose Fourier transform is defined in Figure P4.47-2.

(b) If the sampling rate is $2\pi/T = 1600\pi$, determine the magnitude and phase of $H_1(e^{j\omega})$ (the frequency response of the discrete-time system) so that the output of the system in Figure P4.47-1 is $y_c(t) = f_c(t - 0.1)$. You may use any combination of equations or carefully labeled plots to express your answer.

(c) It turns out that since we are only interested in obtaining $f_c(t)$ at the output, we can use a lower sampling rate than $2\pi/T = 1600\pi$ while still using the antialiasing filter in Figure P4.47-3. Determine the minimum sampling rate that will avoid aliasing distortion of $F_c(j\Omega)$ and determine the frequency response of the filter $H_1(e^{j\omega})$ that can be used so that $y_c(t) = f_c(t)$ at the output of the system in Figure P4.47-1.

(d) Now consider the system shown in Figure P4.47-4, where $2\pi/T = 1600\pi$, and the input signal is defined in Figure P4.47-2 and the antialiasing filter is as shown in Figure P4.47-3.

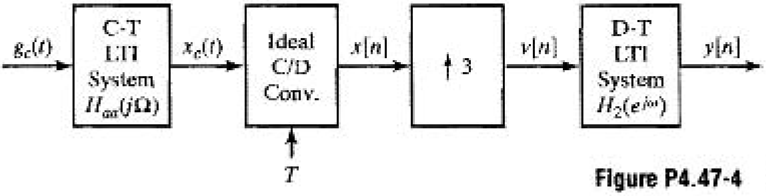

$g_c(t)$
C-T LTI System $H_{aa}(j\Omega)$
$x_c(t)$
Ideal C/D Conv.
T
$x[n]$
$\uparrow 3$
$v[n]$
D-T LTI System $H_2(e^{j\omega})$
$y[n]$

Figure P4.47-4 Another System Block Diagram

where

$$v[n] = \begin{cases} x[n/3] & n = 0, \pm 3, \pm 6, \ldots \\ 0 & \text{otherwise} \end{cases}$$

What should $H_2(e^{j\omega})$ be if it is desired that $y[n] = f_c(nT/3)$?

4.48. **(a)** A finite sequence $b[n]$ is such that:

$$B(z) + B(-z) = 2c, \quad c \neq 0.$$

Explain the structure of $b[n]$. Is there any constraint on the length of $b[n]$?

(b) Is it possible to have $B(z) = H(z)H(z^{-1})$? Explain.

(c) A length-N filter $H(z)$ is such that,

$$H(z)H(z^{-1}) + H(-z)H(-z^{-1}) = c. \qquad \text{(P4.48-1)}$$

Find $G_0(z)$ and $G_1(z)$ such that the filter shown in Figure P4.48 is LTI:

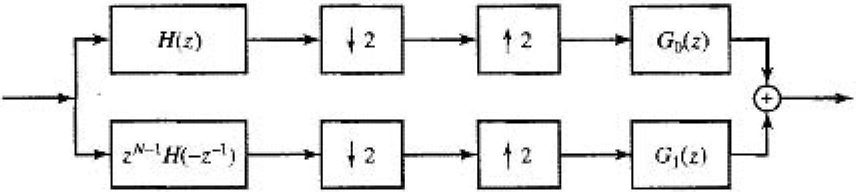


Figure P4.48

(d) For $G_0(z)$ and $G_1(z)$ given in part (c), does the overall system perfectly reconstruct the input? Explain.

4.49. Consider the multirate system shown in Figure P4.49-1 with input $x[n]$ and output $y[n]$:

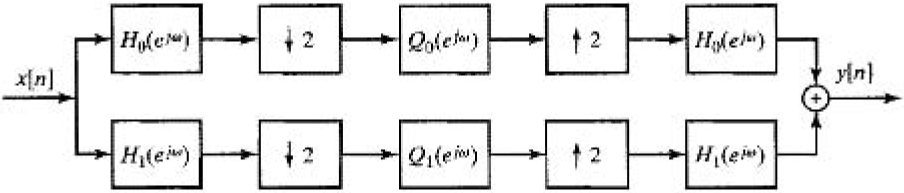


Figure P4.49-1

where $Q_0(e^{j\omega})$ and $Q_1(e^{j\omega})$ are the frequency responses of two LTI systems. $H_0(e^{j\omega})$ and $H_1(e^{j\omega})$ are ideal lowpass and highpass filters, respectively, with cutoff frequency at $\pi/2$ as shown in Figure P4.49-2:

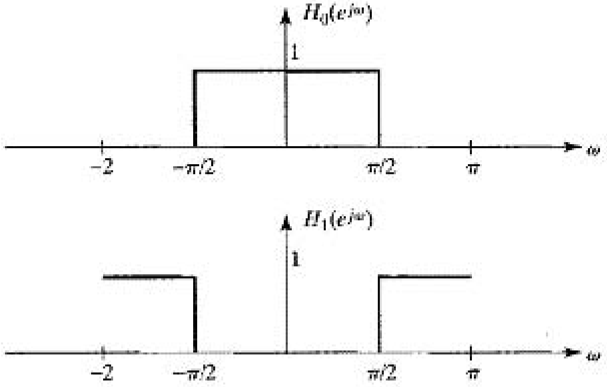


Figure P4.49-2

The overall system is LTI if $Q_0(e^{j\omega})$ and $Q_1(e^{j\omega})$ are as shown in Figure P4.49-3:

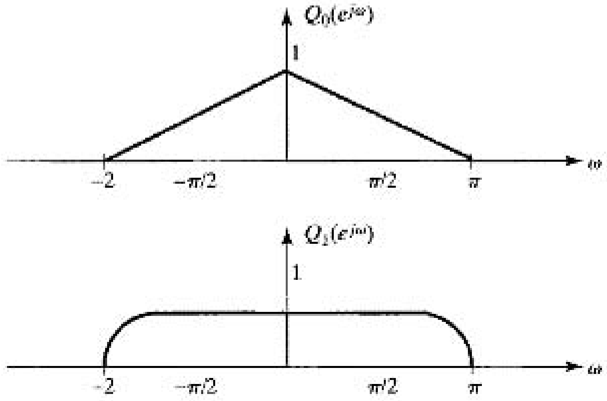


Figure P4.49-3

For these choices of $Q_0(e^{j\omega})$ and $Q_1(e^{j\omega})$, sketch the frequency response

$$G(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})}$$

of the overall system.

4.50. Consider the QMF filterbank shown in Figure P4.50:

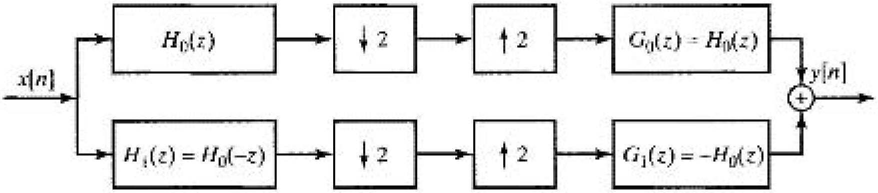


Figure P4.50

The input–output relationship is $Y(z) = T(z)X(z)$, where

$$T(z) = \frac{1}{2}(H_0^2(z) - H_0^2(-z)) = 2z^{-1}E_0(z^2)E_1(z^2)$$

and $E_0(z^2)$, $E_1(z^2)$ are the polyphase components of $H_0(z)$.

Parts (a) and (b) are independent.

(a) Explain whether the following two statements are correct:

(a1) If $H_0(z)$ is linear phase, then $T(z)$ is linear phase.

(a2) If $E_0(z)$ and $E_1(z)$ are linear phase, then $T(z)$ is linear phase.

(b) The prototype filter is known, $h_0[n] = \delta[n] + \delta[n-1] + \frac{1}{4}\delta[n-2]$:

(b1) What are $h_1[n]$, $g_0[n]$ and $g_1[n]$?

(b2) What are $e_0[n]$ and $e_1[n]$?

(b3) What are $T(z)$ and $t[n]$?

4.51. Consider the system in Figure 4.10 with $X_c(j\Omega) = 0$ for $|\Omega| \geq 2\pi(1000)$ and the discrete-time system a squarer, i.e., $y[n] = x^2[n]$. What is the largest value of T such that $y_c(t) = x_c^2(t)$?

4.52. In the system of Figure P4.52,

$$X_c(j\Omega) = 0, \qquad |\Omega| \geq \pi/T,$$

and

$$H(e^{j\omega}) = \begin{cases} e^{-j\omega}, & |\omega| < \pi/L, \\ 0, & \pi/L < |\omega| \leq \pi. \end{cases}$$

How is $y[n]$ related to the input signal $x_c(t)$?

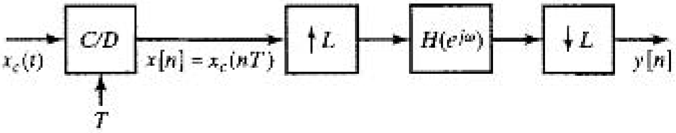


Figure P4.52

Extension Problems

4.53. In many applications, discrete-time random signals arise through periodic sampling of continuous-time random signals. We are concerned in this problem with a derivation of the sampling theorem for random signals. Consider a continuous-time, stationary, random process defined by the random variables $\{x_a(t)\}$, where t is a continuous variable. The autocorrelation function is defined as

$$\phi_{x_c x_c}(\tau) = \mathcal{E}\{x(t)x^*(t+\tau)\},$$

and the power density spectrum is

$$P_{x_c x_c}(\Omega) = \int_{-\infty}^{\infty} \phi_{x_c x_c}(\tau)e^{-j\Omega\tau}d\tau.$$

A discrete-time random process obtained by periodic sampling is defined by the set of random variables $\{x[n]\}$, where $x[n] = x_a(nT)$ and T is the sampling period.

(a) What is the relationship between $\phi_{xx}[n]$ and $\phi_{x_c x_c}(\tau)$?

(b) Express the power density spectrum of the discrete-time process in terms of the power density spectrum of the continuous-time process.

(c) Under what condition is the discrete-time power density spectrum a faithful representation of the continuous-time power density spectrum?

4.54. Consider a continuous-time random process $x_c(t)$ with a bandlimited power density spectrum $P_{x_c x_c}(\Omega)$ as depicted in Figure P4.54-1. Suppose that we sample $x_c(t)$ to obtain the discrete-time random process $x[n] = x_c(nT)$.

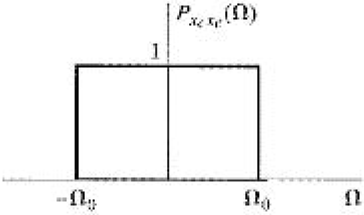


Figure P4.54-1

(a) What is the autocorrelation sequence of the discrete-time random process?

(b) For the continuous-time power density spectrum in Figure P4.54-1, how should T be chosen so that the discrete-time process is white, i.e., so that the power spectrum is constant for all ω?

(c) If the continuous-time power density spectrum is as shown in Figure P4.54-2, how should T be chosen so that the discrete-time process is white?

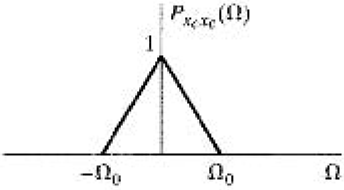


Figure P4.54-2

(d) What is the general requirement on the continuous-time process and the sampling period such that the discrete-time process is white?

4.55. This problem explores the effect of interchanging the order of two operations on a signal, namely, sampling and performing a memoryless nonlinear operation.

(a) Consider the two signal-processing systems in Figure P4.55-1, where the C/D and D/C converters are ideal. The mapping $g[x] = x^2$ represents a memoryless nonlinear device. For the two systems in the figure, sketch the signal spectra at points 1, 2, and 3 when the sampling rate is selected to be $1/T = 2f_m$ Hz and $x_c(t)$ has the Fourier transform shown in Figure P4.55-2. Is $y_1(t) = y_2(t)$? If not, why not? Is $y_1(t) = x^2(t)$? Explain your answer.

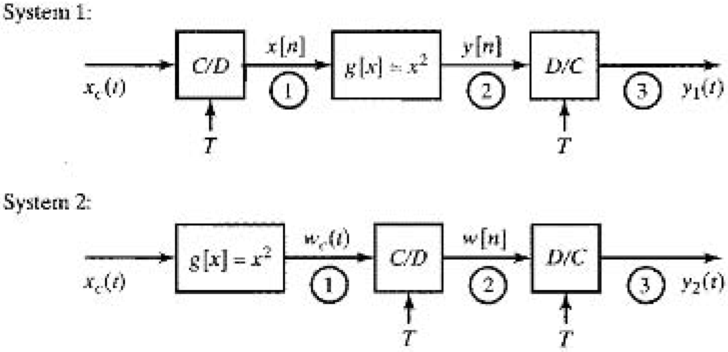


Figure P4.55-1

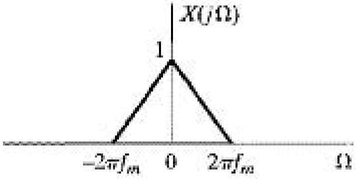


Figure P4.55-2

(b) Consider System 1, and let $x(t) = A\cos(30\pi t)$. Let the sampling rate be $1/T = 40$ Hz. Is $y_1(t) = x_c^2(t)$? Explain why or why not.

(c) Consider the signal-processing system shown in Figure P4.55-3, where $g[x] = x^3$ and $g^{-1}[v]$ is the (unique) inverse, i.e., $g^{-1}[g(x)] = x$. Let $x(t) = A\cos(30\pi t)$ and $1/T = 40$ Hz. Express $v[n]$ in terms of $x[n]$. Is there spectral aliasing? Express $y[n]$ in terms of $x[n]$. What conclusion can you reach from this example? You may find the following identity helpful:

$$\cos^3 \Omega_0 t = \tfrac{3}{4}\cos \Omega_0 t + \tfrac{1}{4}\cos 3\Omega_0 t.$$

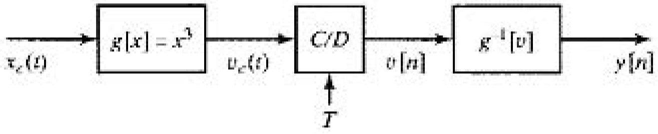


Figure P4.55-3

(d) One practical problem is that of digitizing a signal having a large dynamic range. Suppose we compress the dynamic range by passing the signal through a memoryless nonlinear device prior to A/D conversion and then expand it back after A/D conversion. What is the impact of the nonlinear operation prior to the A/D converter in our choice of the sampling rate?

4.56. Figure 4.23 depicts a system for interpolating a signal by a factor of L, where

$$x_e[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \text{etc}\ldots, \\ 0, & \text{otherwise,} \end{cases}$$

and the lowpass filter interpolates between the nonzero values of $x_e[n]$ to generate the upsampled or interpolated signal $x_i[n]$. When the lowpass filter is ideal, the interpolation is referred to as bandlimited interpolation. As indicated in Section 4.6.3, simple interpolation procedures are adequate in many applications. Two simple procedures often used are zero-order-hold and linear interpolation. For zero-order-hold interpolation, each value of $x[n]$ is simply repeated L times; i.e.,

$$x_i[n] = \begin{cases} x_e[0], & n = 0,\ 1, \ldots, L-1, \\ x_e[L], & n = L,\ L+1, \ldots, 2L-1, \\ x_e[2L], & n = 2L,\ 2L+1, \ldots, \\ \vdots & \end{cases}$$

Linear interpolation is described in Section 4.6.2.

(a) Determine an appropriate choice for the impulse response of the lowpass filter in Figure 4.23 to implement zero-order-hold interpolation. Also, determine the corresponding frequency response.

(b) Equation (4.91) specifies the impulse response for linear interpolation. Determine the corresponding frequency response. (You may find it helpful to use the fact that $h_{\text{lin}}[n]$ is triangular and consequently corresponds to the convolution of two rectangular sequences.)

(c) Sketch the magnitude of the filter frequency response for zero-order-hold and linear interpolation. Which is a better approximation to ideal bandlimited interpolation?

4.57. We wish to compute the autocorrelation function of an upsampled signal, as indicated in Figure P4.57-1. It is suggested that this can equivalently be accomplished with the system of Figure P4.57-2. Can $H_2(e^{j\omega})$ be chosen so that $\phi_3[m] = \phi_1[m]$? If not, why not? If so, specify $H_2(e^{j\omega})$.

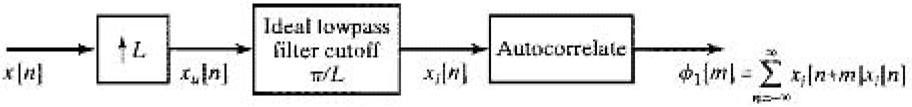

$x[n]$
$\uparrow L$
$x_u[n]$
Ideal lowpass filter cutoff π/L
$x_i[n]$
Autocorrelate
$\phi_1[m] = \sum_{n=-\infty}^{\infty} x_i[n+m]x_i[n]$

Figure P4.57-1

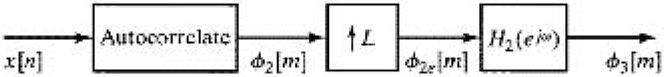


Figure P4.57-2

4.58. We are interested in upsampling a sequence by a factor of 2, using a system of the form of Figure 4.23. However, the lowpass filter in that figure is to be approximated by a five-point filter with impulse response $h[n]$ indicated in Figure P4.58-1. In this system, the output $y_1[n]$ is obtained by direct convolution of $h[n]$ with $w[n]$.

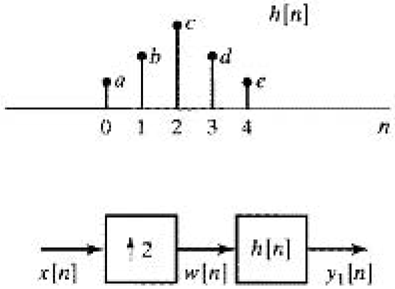


Figure P4.58-1

(a) A proposed implementation of the system with the preceding choice of $h[n]$ is shown in Figure P4.58-2. The three impulse responses $h_1[n]$, $h_2[n]$, and $h_3[n]$ are all restricted to be zero outside the range $0 \leq n \leq 2$. Determine and clearly justify a choice for $h_1[n]$, $h_2[n]$, and $h_3[n]$ so that $y_1[n] = y_2[n]$ for any $x[n]$, i.e., so that the two systems are identical.

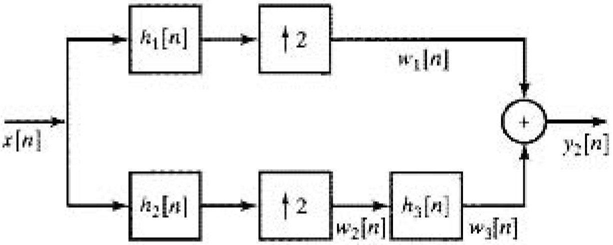


Figure P4.58-2

(b) Determine the number of multiplications per output point required in the system of Figure P4.58-1 and in the system of Figure P4.58-2. You should find that the system of Figure P4.58-2 is more efficient.

4.59. Consider the analysis–synthesis system shown in Figure P4.59-1. The lowpass filter $h_0[n]$ is identical in the analyzer and synthesizer, and the highpass filter $h_1[n]$ is identical in the analyzer and synthesizer. The Fourier transforms of $h_0[n]$ and $h_1[n]$ are related by

$$H_1(e^{j\omega}) = H_0(e^{j(\omega+\pi)}).$$

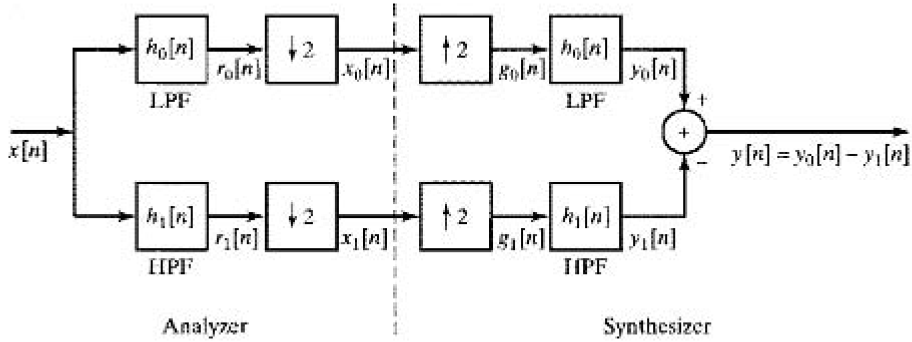


Figure P4.59-1

(a) If $X(e^{j\omega})$ and $H_0(e^{j\omega})$ are as shown in Figure P4.59-2, sketch (to within a scale factor) $X_0(e^{j\omega})$, $G_0(e^{j\omega})$, and $Y_0(e^{j\omega})$.

(b) Write a general expression for $G_0(e^{j\omega})$ in terms of $X(e^{j\omega})$ and $H_0(e^{j\omega})$. Do *not* assume that $X(e^{j\omega})$ and $H_0(e^{j\omega})$ are as shown in Figure P4.59-2.

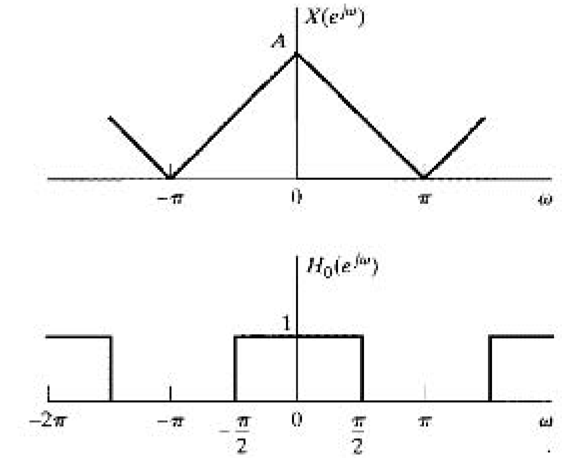


Figure P4.59-2

(c) Determine a set of conditions on $H_0(e^{j\omega})$ that is as general as possible and that will guarantee that $|Y(e^{j\omega})|$ is proportional to $|X(e^{j\omega})|$ for any stable input $x[n]$.

Note: Analyzer–synthesizer filter banks of the form developed in this problem are very similar to quadrature mirror filter banks. (For further reading, see Crochiere and Rabiner (1983), pp. 378–392.)

4.60. Consider a real-valued sequence $x[n]$ for which

$$X(e^{j\omega}) = 0, \qquad \frac{\pi}{3} \leq |\omega| \leq \pi.$$

One value of $x[n]$ may have been corrupted, and we would like to approximately or exactly recover it. With $\hat{x}[n]$ denoting the corrupted signal,

$$\hat{x}[n] = x[n] \text{ for } n \neq n_0,$$

and $\hat{x}[n_0]$ is real but not related to $x[n_0]$. In each of the following three cases, specify a practical algorithm for exactly or approximately recovering $x[n]$ from $\hat{x}[n]$:

(a) The value of n_0 is known.

(b) The exact value of n_0 is *not* known, but we know that n_0 is an even number.

(c) Nothing about n_0 is known.

4.61. Communication systems often require conversion from time-division multiplexing (TDM) to frequency-division multiplexing (FDM). In this problem, we examine a simple example of such a system. The block diagram of the system to be studied is shown in Figure P4.61-1. The TDM input is assumed to be the sequence of interleaved samples

$$w[n] = \begin{cases} x_1[n/2] & \text{for } n \text{ an even integer,} \\ x_2[(n-1)/2] & \text{for } n \text{ an odd integer.} \end{cases}$$

Assume that the sequences $x_1[n] = x_{c1}(nT)$ and $x_2[n] = x_{c2}(nT)$ have been obtained by sampling the continuous-time signals $x_{c1}(t)$ and $x_{c2}(t)$, respectively, without aliasing.

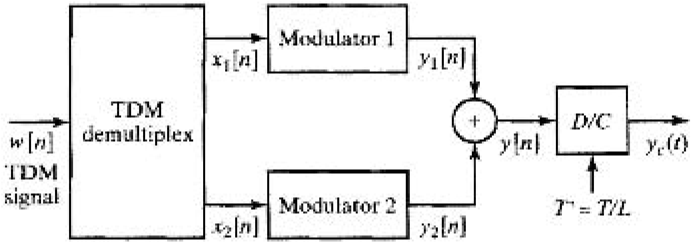


Figure P4.61-1

Assume also that these two signals have the same highest frequency, Ω_N, and that the sampling period is $T = \pi/\Omega_N$.

(a) Draw a block diagram of a system that produces $x_1[n]$ and $x_2[n]$ as outputs; i.e., obtain a system for demultiplexing a TDM signal using simple operations. State whether or not your system is linear, time invariant, causal, and stable.

The k^{th} modulator system ($k = 1$ or 2) is defined by the block diagram in Figure P4.61-2. The lowpass filter $H_i(e^{j\omega})$, which is the same for both channels, has gain L and cutoff frequency π/L, and the highpass filters $H_k(e^{j\omega})$ have unity gain and cutoff frequency ω_k. The modulator frequencies are such that

$$\omega_2 = \omega_1 + \pi/L \qquad \text{and} \qquad \omega_2 + \pi/L \le \pi \qquad (\text{assume } \omega_1 > \pi/2).$$

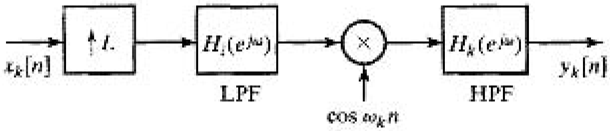


Figure P4.61-2

(b) Assume that $\Omega_N = 2\pi \times 5 \times 10^3$. Find ω_1 and L so that, after ideal D/C conversion with sampling period T/L, the Fourier transform of $y_c(t)$ is zero, except in the band of frequencies

$$2\pi \times 10^5 \le |\omega| \le 2\pi \times 10^5 + 2\Omega_N.$$

(c) Assume that the continuous-time Fourier transforms of the two original input signals are as sketched in Figure P4.61-3. Sketch the Fourier transforms at each point in the system.

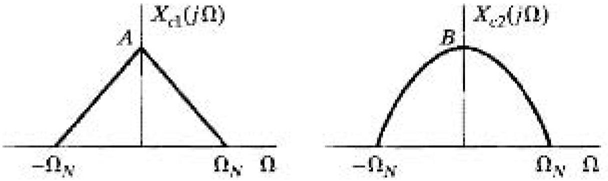


Figure P4.61-3

(d) Based on your solution to parts (a)–(c), discuss how the system could be generalized to handle M equal-bandwidth channels.

4.62. In Section 4.8.1, we considered the use of prefiltering to avoid aliasing. In practice, the antialiasing filter cannot be ideal. However, the nonideal characteristics can be at least partially compensated for with a discrete-time system applied to the sequence $x[n]$ that is the output of the C/D converter.

Consider the two systems in Figure P4.62-1. The antialiasing filters $H_{\text{ideal}}(j\Omega)$ and $H_{\text{aa}}(j\Omega)$ are shown in Figure P4.62-2. $H(e^{j\omega})$ in Figure P4.62-1 is to be specified to compensate for the nonideal characteristics of $H_{\text{aa}}(j\Omega)$.

Sketch $H(e^{j\omega})$ so that the two sequences $x[n]$ and $w[n]$ are identical.

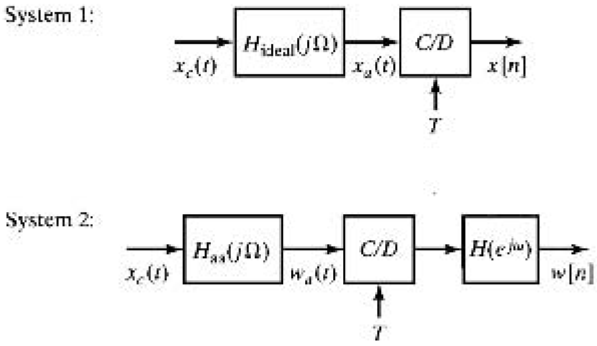


Figure P4.62-1

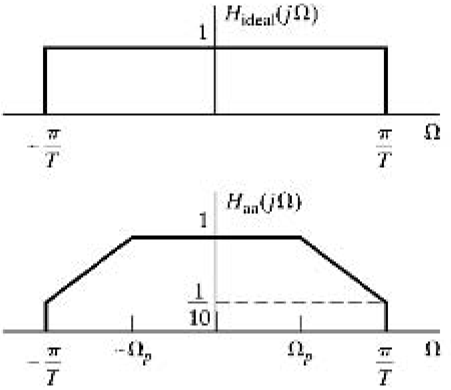


Figure P4.62-2

4.63. As discussed in Section 4.8.2, to process sequences on a digital computer, we must quantize the amplitude of the sequence to a set of discrete levels. This quantization can be expressed in terms of passing the input sequence $x[n]$ through a quantizer $Q(x)$ that has an input–output relation as shown in Figure 4.54.

As discussed in Section 4.8.3, if the quantization interval Δ is small compared with changes in the level of the input sequence, we can assume that the output of the quantizer is of the form

$$y[n] = x[n] + e[n],$$

where $e[n] = Q(x[n]) - x[n]$ and $e[n]$ is a stationary random process with a 1st-order probability density uniform between $-\Delta/2$ and $\Delta/2$, uncorrelated from sample to sample and uncorrelated with $x[n]$, so that $\mathcal{E}\{e[n]x[m]\} = 0$ for all m and n.

Let $x[n]$ be a stationary white-noise process with zero mean and variance σ_x^2.

(a) Find the mean, variance, and autocorrelation sequence of $e[n]$.

(b) What is the signal-to-quantizing-noise ratio σ_x^2/σ_e^2?

(c) The quantized signal $y[n]$ is to be filtered by a digital filter with impulse response $h[n] = \frac{1}{2}[a^n + (-a)^n]u[n]$. Determine the variance of the noise produced at the output due to the input quantization noise, and determine the SNR at the output.

In some cases we may want to use nonlinear quantization steps, for example, logarithmically spaced quantization steps. This can be accomplished by applying uniform quantization to the logarithm of the input as depicted in Figure P4.63, where $Q[\cdot]$ is a uniform quantizer as specified in Figure 4.54. In this case, if we assume that Δ is small compared with changes in the sequence $\ln(x[n])$, then we can assume that the output of the quantizer is

$$\ln(y[n]) = \ln(x[n]) + e[n].$$

Thus,

$$y[n] = x[n] \cdot \exp(e[n]).$$

For small e, we can approximate $\exp(e[n])$ by $(1 + e[n])$, so that

$$y[n] \approx x[n](1 + e[n]) = x[n] + f[n]. \qquad \text{(P4.63-1)}$$

This equation will be used to describe the effect of logarithmic quantization. We assume $e[n]$ to be a stationary random process, uncorrelated from sample to sample, independent of the signal $x[n]$, and with 1st-order probability density uniform between $\pm\Delta/2$.

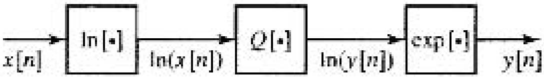


Figure P4.63

(d) Determine the mean, variance, and autocorrelation sequence of the *additive* noise $f[n]$ defined in Eq. (4.57).

(e) What is the signal-to-quantizing-noise ratio σ_x^2/σ_f^2? Note that in this case σ_x^2/σ_f^2 is independent of σ_x^2. Within the limits of our assumption, therefore, the signal-to-quantizing-noise ratio is independent of the input signal level, whereas, for linear quantization, the ratio σ_x^2/σ_e^2 depends directly on σ_x^2.

(f) The quantized signal $y[n]$ is to be filtered by means of a digital filter with impulse response $h[n] = \frac{1}{2}[a^n + (-a)^n]u[n]$. Determine the variance of the noise produced at the output due to the input quantization noise, and determine the SNR at the output.

4.64. Figure P4.64-1 shows a system in which two continuous-time signals are multiplied and a discrete-time signal is then obtained from the product by sampling the product at the Nyquist rate; i.e., $y_1[n]$ is samples of $y_c(t)$ taken at the Nyquist rate. The signal $x_1(t)$ is bandlimited to 25 kHz ($X_1(j\Omega) = 0$ for $|\Omega| \geq 5\pi \times 10^4$), and $x_2(t)$ is limited to 2.5 kHz ($X_2(j\Omega) = 0$ for $|\Omega| \geq (\pi/2) \times 10^4$).

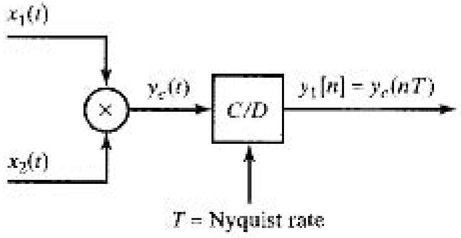


Figure P4.64-1

In some situations (digital transmission, for example), the continuous-time signals have already been sampled at their individual Nyquist rates, and the multiplication is to be carried out in the discrete-time domain, perhaps with some additional processing before and after multiplication, as indicated in Figure P4.64-2. Each of the systems A, B, and C either is an identity or can be implemented using one or more of the modules shown in Figure P4.64-3.

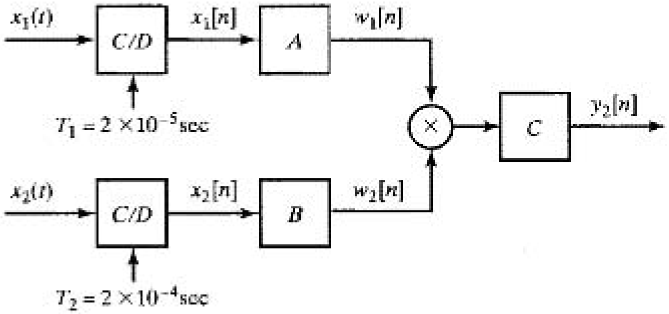


Figure P4.64-2

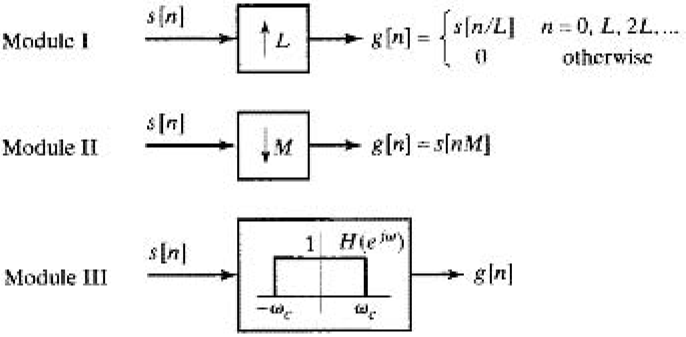


Figure P4.64-3

For each of the three systems A, B, and C, either specify that the system is an identity system or specify an appropriate interconnection of one or more of the modules shown in Figure P4.64-3. Also, specify all relevant parameters L, M, and ω_c. The systems A, B, and C should be constructed such that $y_2[n]$ is proportional to $y_1[n]$, i.e.,

$$y_2[n] = ky_1[n] = ky_c(nT) = kx_1(nT) \times x_2(nT).$$

and these samples are at the Nyquist rate, i.e., $y_2[n]$ does not represent oversampling or undersampling of $y_c(t)$.

4.65. Suppose $s_c(t)$ is a speech signal with the continuous-time Fourier transform $S_c(j\Omega)$ shown in Figure P4.65-1. We obtain a discrete-time sequence $s_r[n]$ from the system shown in Figure P4.65-2, where $H(e^{j\omega})$ is an ideal discrete-time lowpass filter with cutoff frequency ω_c and a gain of L throughout the passband, as shown in Figure 4.29(b). The signal $s_r[n]$ will be used as an input to a speech coder, which operates correctly only on discrete-time samples representing speech sampled at an 8-kHz rate. Choose values of L, M, and ω_c that produce the correct input signal $s_r[n]$ for the speech coder.

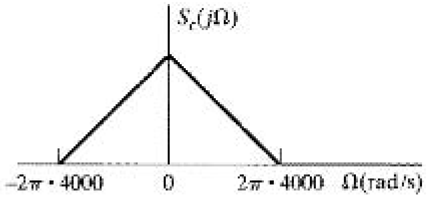


Figure P4.65-1

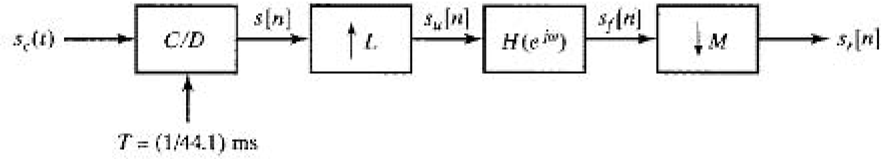


Figure P4.65-2

4.66. In many audio applications, it is necessary to sample a continuous-time signal $x_c(t)$ at a sampling rate $1/T = 44$ kHz. Figure P4.66-1 shows a straightforward system, including a continuous-time antialias filter $H_{a0}(j\Omega)$, to acquire the desired samples. In many applications, the "4x oversampling" system shown in Figure P4.66-2 is used instead of the conventional system shown in Figure P4.66-1. In the system in Figure P4.66-2,

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| \le \pi/4, \\ 0, & \text{otherwise}, \end{cases}$$

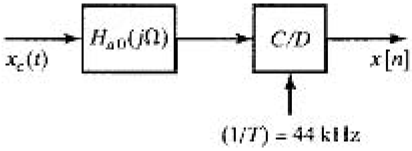


Figure P4.66-1

is an ideal lowpass filter, and

$$H_{a1}(j\Omega) = \begin{cases} 1, & |\Omega| \le \Omega_p, \\ 0, & |\Omega| > \Omega_s, \end{cases}$$

for some $0 \le \Omega_p \le \Omega_s \le \infty$.

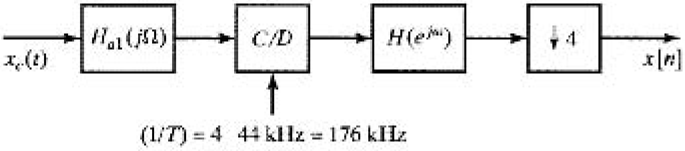


Figure P4.66-2

Assuming that $H(e^{j\omega})$ is ideal, find the minimal set of specifications on the antialias filter $H_{a1}(j\Omega)$, i.e., the smallest Ω_p and the largest Ω_s, such that the overall system of Figure P4.66-2 is equivalent to the system in Figure P4.66-1.

4.67. In this problem, we will consider the "double integration" system for quantization with noise shaping shown in Figure P4.67. In this system,

$$H_1(z) = \frac{1}{1 - z^{-1}} \quad \text{and} \quad H_2(z) = \frac{z^{-1}}{1 - z^{-1}},$$

and the frequency response of the decimation filter is

$$H_3(e^{j\omega}) = \begin{cases} 1, & |\omega| < \pi/M, \\ 0, & \pi/M \leq |\omega| \leq \pi. \end{cases}$$

The noise source $e[n]$, which represents a quantizer, is assumed to be a zero-mean white-noise (constant power spectrum) signal that is uniformly distributed in amplitude and has noise power $\sigma_e^2 = \Delta^2/12$.

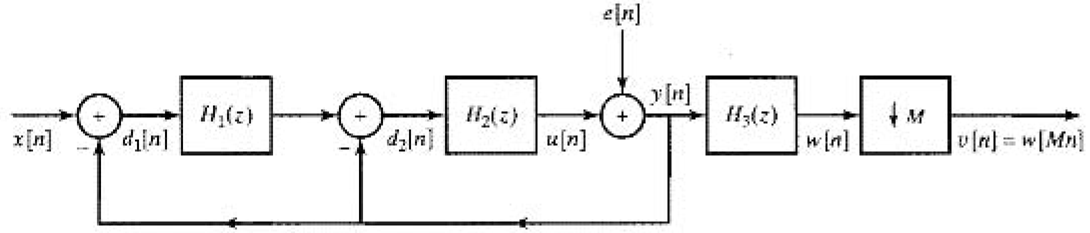


Figure P4.67

(a) Determine an equation for $Y(z)$ in terms of $X(z)$ and $E(z)$. Assume for this part that $E(z)$ exists. From the z-transform relation, show that $y[n]$ can be expressed in the form $y[n] = x[n-1] + f[n]$, where $f[n]$ is the output owing to the noise source $e[n]$. What is the time-domain relation between $f[n]$ and $e[n]$?

(b) Now assume that $e[n]$ is a white-noise signal as described prior to part (a). Use the result from part (a) to show that the power spectrum of the noise $f[n]$ is

$$P_{ff}(e^{j\omega}) = 16\sigma_e^2 \sin^4(\omega/2).$$

What is the *total* noise power (σ_f^2) in the noise component of the signal $y[n]$? On the same set of axes, sketch the power spectra $P_{ee}(e^{j\omega})$ and $P_{ff}(e^{j\omega})$ for $0 \leq \omega \leq \pi$.

(c) Now assume that $X(e^{j\omega}) = 0$ for $\pi/M < \omega \leq \pi$. Argue that the output of $H_3(z)$ is $w[n] = x[n-1] + g[n]$. State in words what $g[n]$ is.

(d) Determine an expression for the noise power σ_g^2 at the output of the decimation filter. Assume that $\pi/M \ll \pi$, i.e., M is large, so that you can use a small-angle approximation to simplify the evaluation of the integral.

(e) After the decimator, the output is $v[n] = w[Mn] = x[Mn-1] + q[n]$, where $q[n] = g[Mn]$. Now suppose that $x[n] = x_c(nT)$ (i.e., $x[n]$ was obtained by sampling a continuous-time signal). What condition must be satisfied by $X_c(j\Omega)$ so that $x[n-1]$ will pass through the filter unchanged? Express the "signal component" of the output $v[n]$ in terms of $x_c(t)$. What is the total power σ_q^2 of the noise at the output? Give an expression for the power spectrum of the noise at the output, and, on the same set of axes, sketch the power spectra $P_{ee}(e^{j\omega})$ and $P_{qq}(e^{j\omega})$ for $0 \leq \omega \leq \pi$.

4.68. For sigma-delta oversampled A/D converters with high-order feedback loops, stability becomes a significant consideration. An alternative approach referred to as multi-stage noise shaping (MASH) achieves high-order noise shaping with only 1st-order feedback. The structure for 2^{nd}-order MASH noise shaping is shown in Figure P4.68-2 and analyzed in this problem.

Figure P4.68-1 is a 1^{st}-order sigma-delta $(\Sigma - \Delta)$ noise shaping system, where the effect of the quantizer is represented by the additive noise signal $e[n]$. The noise $e[n]$ is explicitly shown in the diagram as a second output of the system. Assume that the input $x[n]$ is a zero-mean wide-sense stationary random process. Assume also that $e[n]$ is zero-mean, white, wide-sense stationary, and has variance σ_e^2. $e[n]$ is uncorrelated with $x[n]$.

(a) For the system in Figure P4.68-1, the output $y[n]$ has a component $y_x[n]$ due only to $x[n]$ and a component $y_e[n]$ due only to $e[n]$, i.e., $y[n] = y_x[n] + y_e[n]$.

(i) Determine $y_x[n]$ in terms of $x[n]$.

(ii) Determine $P_{y_e}(\omega)$, the power spectral density of $y_e[n]$.

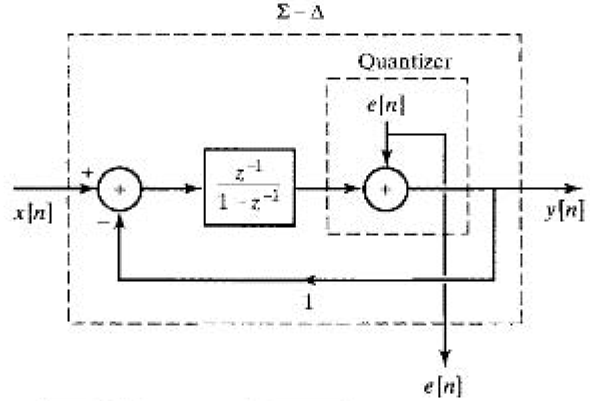


Figure P4.68-1

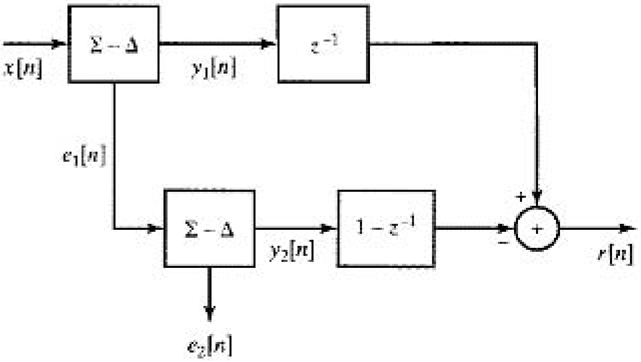


Figure P4.68-2

(a) The system of Figure P4.68 is now connected in the configuration shown in Figure P4.68, which shows the structure of the MASH system. Notice that $e_1[n]$ and $e_2[n]$ are the noise signals resulting from the quantizers in the sigma-delta noise shaping systems. The output of the system $r[n]$ has a component $r_x[n]$ owing only to $x[n]$, and a component $r_e[n]$ due only to the quantization noise, i.e., $r[n] = r_x[n] + r_e[n]$. Assume that $e_1[n]$ and $e_2[n]$ are zero-mean, white, wide-sense stationary, each with variance σ_e^2. Also assume that $e_1[n]$ is uncorrelated with $e_2[n]$.

(i) Determine $r_x[n]$ in terms of $x[n]$.

(ii) Determine $P_{r_e}(\omega)$, the power spectral density of $r_e[n]$.