CHAPTER

Signal Sampling and Quantization

2

CHAPTER OUTLINE

OBJECTIVES:

This chapter investigates the sampling process, sampling theory, and the signal reconstruction process. It also includes practical considerations for anti-aliasing and anti-image filters and signal quantization.

2.1 SAMPLING OF CONTINUOUS SIGNAL

As discussed in Chapter 1, Figure 2.1 describes a simplified block diagram of a digital signal processing (DSP) system. The analog filter processes the analog input to obtain the band-limited signal, which is sent to the analog-to-digital conversion (ADC) unit. The ADC unit samples the analog signal, quantizes the sampled signal, and encodes the quantized signal level to the digital signal.

Here we first develop concepts of sampling processing in the time domain. Figure 2.2 shows an analog (continuous-time) signal (solid line) defined at every point over the time axis (horizontal line) and amplitude axis (vertical line). Hence, the analog signal contains an infinite number of points.

It is impossible to digitize an infinite number of points. The infinite points cannot be processed by the digital signal (DS) processor or computer, since they require an infinite amount of memory and infinite amount of processing power for computations. Sampling can solve such a problem by taking samples at a fixed time interval as shown in Figure 2.2 and Figure 2.3, where the time T represents the sampling interval or sampling period in seconds.

As shown in Figure 2.3, each sample maintains its voltage level during the sampling interval T to give the ADC enough time to convert it. This process is called *sample and hold*. Since there exits one amplitude level for each sampling interval, we can sketch each sample amplitude level at its corresponding sampling time instant shown in Figure 2.2, where 14 samples at their sampling time instants are plotted, each using a vertical bar with a solid circle at its top.

Digital Signal Processing. http://dx.doi.org/10.1016/B978-0-12-415893-1.00002-0

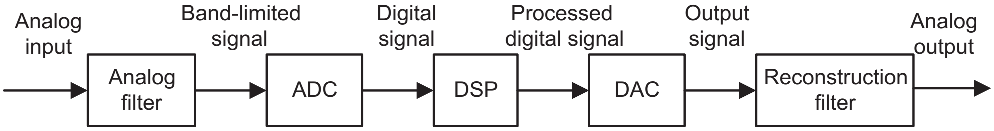


FIGURE 2.1

A digital signal processing scheme.

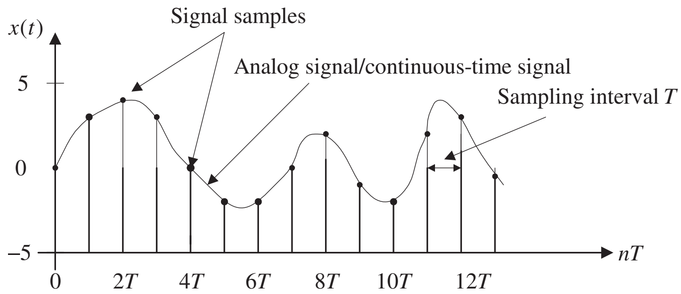


FIGURE 2.2

Display of the analog (continuous) signal and the digital samples versus the sampling time instants.

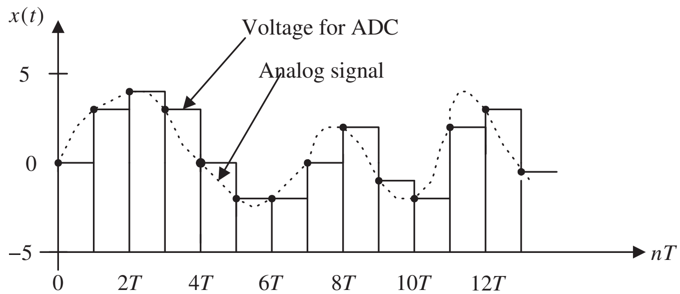


FIGURE 2.3

Sample-and-hold analog voltage for ADC.

For a given sampling interval T, which is defined as the time span between two sample points, the sampling rate is therefore given by

$$f_s = \frac{1}{T} \text{ samples per second (Hz)}$$

For example, if a sampling period is $T = 125$ microseconds, the sampling rate is $f_s = 1/125\mu s = 8,000$ samples per second (Hz).

After obtaining the sampled signal whose amplitude values are taken at the sampling instants, the processor is able to process the sample points. Next, we have to ensure that samples are collected at a rate high enough that the original analog signal can be reconstructed or recovered later. In other words, we are looking for a minimum sampling rate to acquire a complete reconstruction of the analog signal from its sampled version. If an analog signal is not appropriately sampled, *aliasing* will occur, which causes unwanted signals in the desired frequency band.

The sampling theorem guarantees that an analog signal can be in theory perfectly recovered as long as the sampling rate is at least twice as large as the highest-frequency component of the analog signal to be sampled. The condition is described as

$$f_s \geq 2f_{\max}$$

where $f_{\max}$ is the maximum-frequency component of the analog signal to be sampled. For example, to sample a speech signal containing frequencies up to 4 kHz, the minimum sampling rate is chosen to be at least 8 kHz, or 8,000 samples per second; to sample an audio signal possessing frequencies up to 20 kHz, at least 40,000 samples per second, or 40 kHz, of the audio signal are required.

Figure 2.4 illustrates sampling of two sinusoids, where the sampling interval between sample points is $T = 0.01$ seconds, and the sampling rate is thus $f_s = 100$ Hz. The first plot in the figure displays a sine wave with a frequency of 40 Hz and its sampled amplitudes. The sampling theorem condition is satisfied since $2f_{\max} = 80 < f_s$. The sampled amplitudes are labeled using the circles shown in the first

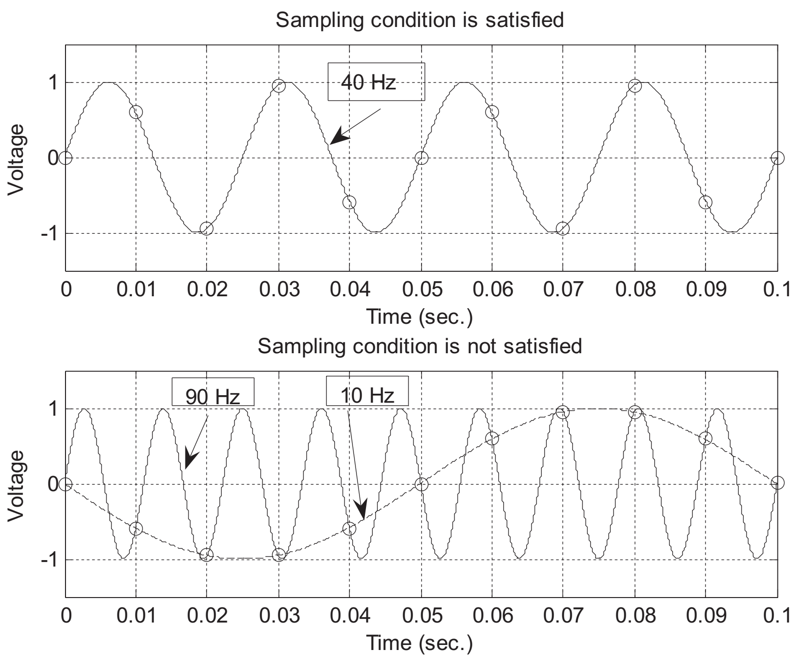


FIGURE 2.4

Plots of the appropriately sampled signals and nonappropriately sampled (aliased) signals.

plot. We notice that the 40-Hz signal is adequately sampled, since the sampled values clearly come from the analog version of the 40-Hz sine wave. However, as shown in the second plot, the sine wave with a frequency of 90 Hz is sampled at 100 Hz. Since the sampling rate of 100 Hz is relatively low compared with the 90-Hz sine wave, the signal is undersampled due to $2f_{\max} = 180 > f_s$. Hence, the condition of the sampling theorem is not satisfied. Based on the sample amplitudes labeled with the circles in the second plot, we cannot tell whether the sampled signal comes from sampling a 90-Hz sine wave (plotted using the solid line) or from sampling a 10-Hz sine wave (plotted using the dot-dash line). They are not distinguishable. Thus they are *aliases* of each other. We call the 10-Hz sine wave the aliasing noise in this case, since the sampled amplitudes actually come from sampling the 90-Hz sine wave.

Now let us develop the sampling theorem in frequency domain, that is, the minimum sampling rate requirement for sampling an analog signal. As we shall see, in practice this can help us design the anti-aliasing filter (a lowpass filter that will reject high frequencies that cause aliasing) that will be applied before sampling, and the anti-image filter (a reconstruction lowpass filter that will smooth the recovered sample-and-hold voltage levels to an analog signal) that will be applied after the digital-to-analog conversion (DAC).

Figure 2.5 depicts the sampled signal $x_s(t)$ obtained by sampling the continuous signal $x(t)$ at a sampling rate of f_s samples per second.

Mathematically, this process can be written as the product of the continuous signal and the sampling pulses (pulse train):

$$x_s(t) = x(t)p(t) \tag{2.1}$$

where $p(t)$ is the pulse train with a period $T = 1/f_s$. From spectral analysis, the original spectrum (frequency components) $X(f)$ and the sampled signal spectrum $X_s(f)$ in terms of Hz are related as

$$X_s(f) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X(f - nf_s) \tag{2.2}$$

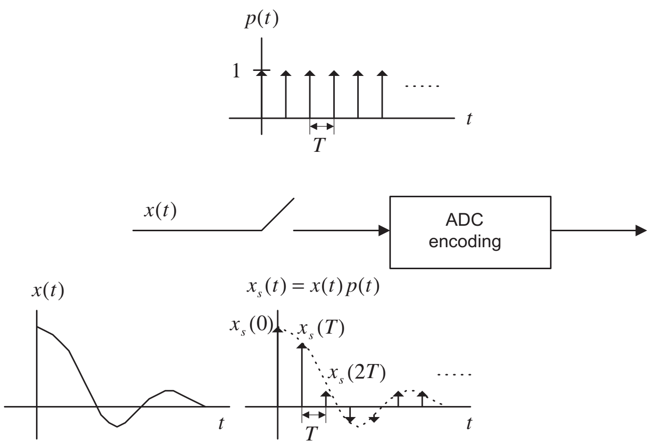


FIGURE 2.5

The simplified sampling process.

where $X(f)$ is assumed to be the original baseband spectrum while $X_s(f)$ is its sampled signal spectrum, consisting of the original baseband spectrum $X(f)$ and its replicas $X(f \pm nf_s)$. Since Equation (2.2) is a well-known formula, the derivation is omitted here and can be found in well-known texts (Ahmed and Nataranjan, 1983; Ambardar, 1999; Alkin, 1993; Oppenheim and Schafer, 1975; Proakis and Manolakis, 1996).

Expanding Equation (2.2) leads to the sampled signal spectrum in Equation (2.3):

$$X_s(f) = \cdots + \frac{1}{T}X(f+f_s) + \frac{1}{T}X(f) + \frac{1}{T}X(f-f_s) + \cdots \tag{2.3}$$

Equation (2.3) indicates that the sampled signal spectrum is the sum of the scaled original spectrum and copies of its shifted versions, called *replicas*. Three possible sketches based on Equation (2.3) can be obtained. Given the original signal spectrum $X(f)$ plotted in Figure 2.6(a), the sampled signal spectrum according to Equation (2.3) is plotted in Figure 2.6(b), where the replicas $\frac{1}{T}X(f)$, $\frac{1}{T}X(f-f_s)$, $\frac{1}{T}X(f+f_s)$, …, have separations between them. Figure 2.6(c) shows that the baseband spectrum and its replicas, $\frac{1}{T}X(f)$, $\frac{1}{T}X(f-f_s)$, $\frac{1}{T}X(f+f_s)$, …, are just connected, and finally, in Figure 2.6(d), the original

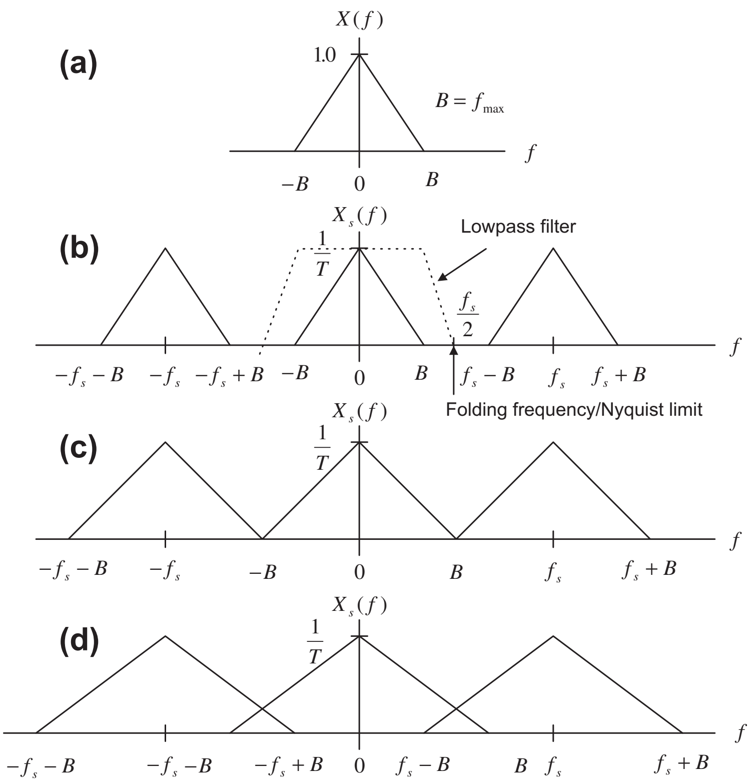


FIGURE 2.6

Plots of the sampled signal spectrum.

spectrum $\frac{1}{T}X(f)$ and its replicas $\frac{1}{T}X(f-f_s)$, $\frac{1}{T}X(f+f_s)$, ..., are overlapped; that is, there are many overlapping portions in the sampled signal spectrum.

From Figure 2.6, it is clear that the sampled signal spectrum consists of the scaled baseband spectrum centered at the origin, and its replicas centered at the frequencies of $\pm nf_s$ (multiples of the sampling rate) for each of $n = 1, 2, 3, \ldots$.

If applying a lowpass reconstruction filter to obtain exact reconstruction of the original signal spectrum, the following condition must be satisfied:

$$f_s - f_{\max} \geq f_{\max} \tag{2.4}$$

Solving Equation (2.4) gives

$$f_s \geq 2f_{\max} \tag{2.5}$$

In terms of frequency in radians per second, Equation (2.5) is equivalent to

$$\omega_s \geq 2\omega_{\max} \tag{2.6}$$

This fundamental conclusion is well known as the **Shannon sampling theorem**, which is formally described below:

> For a uniformly sampled DSP system, an analog signal can be perfectly recovered as long as the sampling rate is at least twice as large as the highest-frequency component of the analog signal to be sampled.

We summarize two key points here.

1. The sampling theorem establishes a minimum sampling rate for a given band-limited analog signal with highest-frequency component $f_{\max}$. If the sampling rate satisfies Equation (2.5), then the analog signal can be recovered via its sampled values using the lowpass filter, as described in Figure 2.6(b).
2. Half of the sampling frequency $f_s/2$ is usually called the *Nyquist frequency* (Nyquist limit) or *folding frequency*. The sampling theorem indicates that a DSP system with a sampling rate of f_s can ideally sample an analog signal with a maximum frequency that is up to half of the sampling rate without introducing spectral overlap (aliasing). Hence, the analog signal can be perfectly recovered from its sampled version.

Let us study the following example.

EXAMPLE 2.1

Suppose that an analog signal is given as

$$x(t) = 5\cos(2\pi \cdot 1{,}000t), \text{ for } t \geq 0$$

and is sampled at the rate 8,000 Hz.

a. Sketch the spectrum for the original signal.
b. Sketch the spectrum for the sampled signal from 0 to 20 kHz.

Solution:

a. Since the analog signal is sinusoid with a peak value of 5 and frequency of 1,000 Hz, we can write the sine wave using Euler's identity:

$$5\cos(2\pi \times 1{,}000t) = 5\cdot\left(\frac{e^{j2\pi\times 1{,}000t} + e^{-j2\pi\times 1{,}000t}}{2}\right) = 2.5e^{j2\pi\times 1{,}000t} + 2.5e^{-j2\pi\times 1{,}000t}$$

which is a Fourier series expansion for a continuous periodic signal in terms of the exponential form (see Appendix B). We can identify the Fourier series coefficients as

$$c_1 = 2.5 \text{ and } c_{-1} = 2.5$$

Using the magnitudes of the coefficients, we then plot the two-side spectrum as shown in Figure 2.7A.

b. After the analog signal is sampled at the rate of 8,000 Hz, the sampled signal spectrum and its replicas centered at the frequencies $\pm nf_s$, each with a scaled amplitude of $2.5/T$, are as shown in Figure 2.7B:

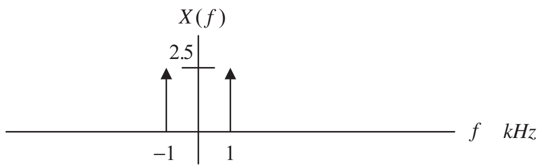


FIGURE 2.7A

Spectrum of the analog signal in Example 2.1.

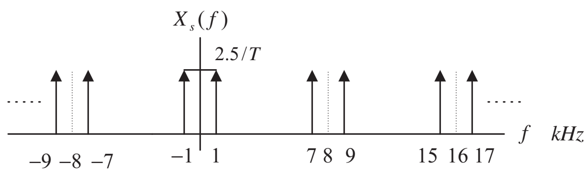


FIGURE 2.7B

Spectrum of the sampled signal in Example 2.1.

Notice that the spectrum of the sampled signal shown in Figure 2.7B contains the images of the original spectrum shown in Figure 2.7A; that the images repeat at multiples of the sampling frequency f_s (for our example, 8 kHz, 16kHz, 24kHz, ...); and that all images must be removed, since they convey no additional information.

2.2 SIGNAL RECONSTRUCTION

In this section, we investigate the recovery of analog signal from its sampled signal version. Two simplified steps are involved, as described in Figure 2.8. First, the digitally processed data $y(n)$ are converted to the ideal impulse train $y_s(t)$, in which each impulse has amplitude proportional to digital output $y(n)$, and two consecutive impulses are separated by a sampling period of T; second, the analog

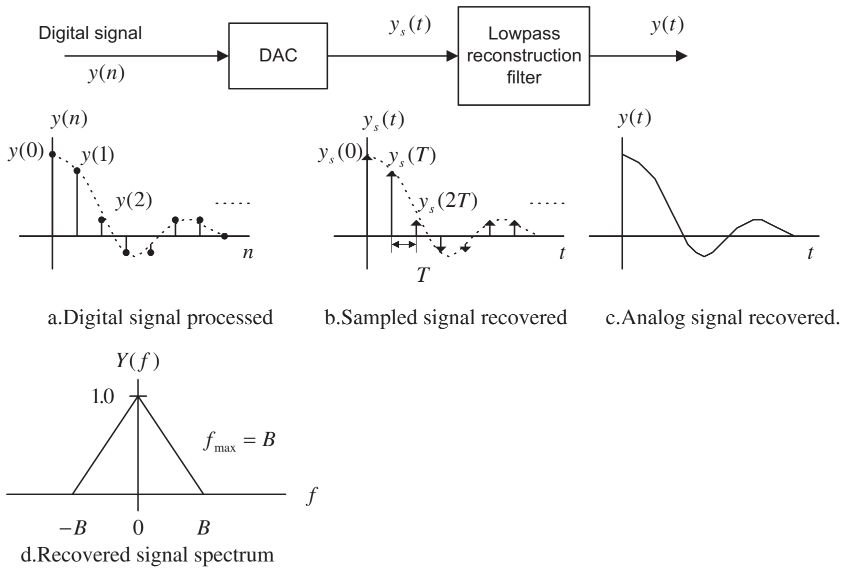


FIGURE 2.8

Signal notations at the reconstruction stage.

reconstruction filter is applied to the ideally recovered sampled signal $y_s(t)$ to obtain the recovered analog signal.

To study the signal reconstruction, we let $y(n) = x(n)$ for the case of no DSP, so that the reconstructed sampled signal and the input sampled signal are ensured to be the same; that is, $y_s(t) = x_s(t)$. Hence, the spectrum of the sampled signal $y_s(t)$ contains the same spectral content of the original spectrum $X(f)$, that is, $Y(f) = X(f)$, with a bandwidth of $f_{\max} = B$ Hz (described in Figure 2.8d) and the images of the original spectrum (scaled and shifted versions). The following three cases are discussed for recovery of the original signal spectrum $X(f)$.

Case 1: $f_s = 2f_{\max}$

As shown in Figure 2.9, where the Nyquist frequency is equal to the maximum frequency of the analog signal $x(t)$, an ideal lowpass reconstruction filter is required to recover the analog signal spectrum. This is an impractical case.

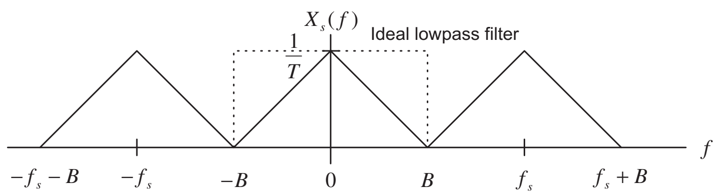


FIGURE 2.9

Spectrum of the sampled signal when $f_s = 2f_{\max}$.

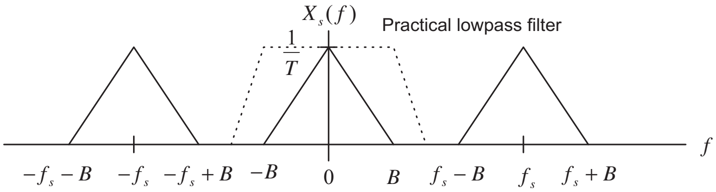


FIGURE 2.10

Spectrum of the sampled signal when $f_s > 2f_{\max}$.

Case 2: $f_s > 2f_{\max}$

In this case, as shown in Figure 2.10, there is a separation between the highest-frequency edge of the baseband spectrum and the lower edge of the first replica. Therefore, a practical lowpass reconstruction (anti-image) filter can be designed to reject all the images and achieve the original signal spectrum.

Case 3: $f_s < 2f_{\max}$

Case 3 violates the condition of the Shannon sampling theorem. As we can see, Figure 2.11 depicts the spectral overlapping between the original baseband spectrum and the spectrum of the first replica and so on. Even when we apply an ideal lowpass filter to remove these images, in the baseband there are still some foldover frequency components from the adjacent replica. This is aliasing, where the recovered baseband spectrum suffers spectral distortion, that is, it contains an aliasing noise spectrum; in the time domain, the recovered analog signal may consist of the aliasing noise frequency or frequencies. Hence, the recovered analog signal is incurably distorted.

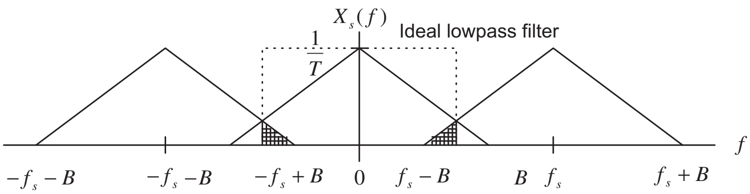


FIGURE 2.11

Spectrum of the sampled signal when $f_s < 2f_{\max}$.

Note that if an analog signal with a frequency f is undersampled, the aliasing frequency component f_{alias} in the baseband is simply given by the following expression:

$$f_{alias} = f_s - f$$

The following examples give a spectrum analysis of the signal recovery.

EXAMPLE 2.2

Assume that an analog signal is given by

$$x(t) = 5\cos(2\pi \cdot 2{,}000t) + 3\cos(2\pi \cdot 3{,}000t), \text{ for } t \geq 0$$

and is sampled at the rate of 8,000 Hz.

a. Sketch the spectrum of the sampled signal up to 20 kHz.
b. Sketch the recovered analog signal spectrum if an ideal lowpass filter with a cutoff frequency of 4 kHz is used to filter the sampled signal ($y(n) = x(n)$ in this case) to recover the original signal.

Solution:
a. Using Euler's identity, we get

$$x(t) = \frac{3}{2}e^{-j2\pi\cdot 3{,}000t} + \frac{5}{2}e^{-j2\pi\cdot 2{,}000t} + \frac{5}{2}e^{j2\pi\cdot 2{,}000t} + \frac{3}{2}e^{j2\pi\cdot 3{,}000t}$$

The two-sided amplitude spectrum for the sinusoid is displayed in Figure 2.12:

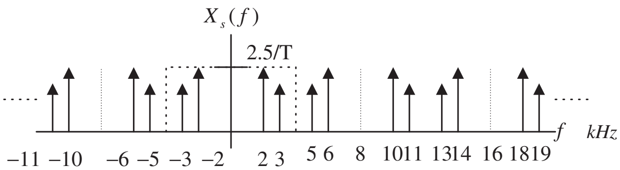


FIGURE 2.12

Spectrum of the sampled signal in Example 2.2.

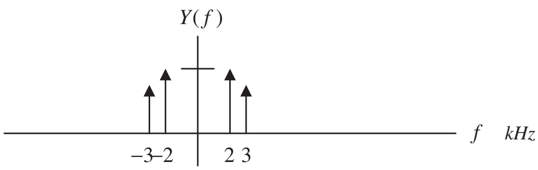


FIGURE 2.13

Spectrum of the recovered signal in Example 2.2.

b. Based on the spectrum in (a), the sampling theorem condition is satisfied; hence, we can recover the original spectrum using a reconstruction lowpass filter. The recovered spectrum is shown in Figure 2.13.

EXAMPLE 2.3

Assume an analog signal is given by

$$x(t) = 5\cos(2\pi \times 2{,}000t) + 1\cos(2\pi \times 5{,}000t), \text{ for } t \geq 0$$

and is sampled at a rate of 8,000 Hz.

a. Sketch the spectrum of the sampled signal up to 20 kHz.
b. Sketch the recovered analog signal spectrum if an ideal lowpass filter with a cutoff frequency of 4 kHz is used to recover the original signal ($y(n) = x(n)$ in this case).

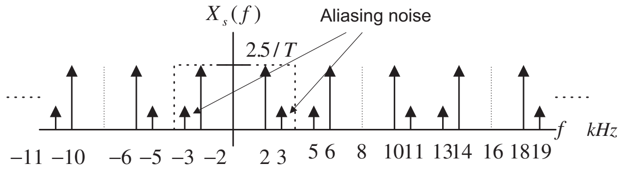


FIGURE 2.14

Spectrum of the sampled signal in Example 2.3.

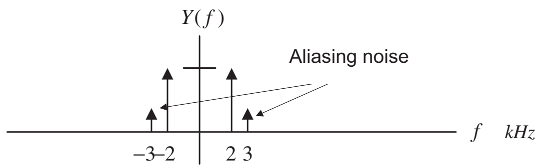


FIGURE 2.15

Spectrum of the recovered signal in Example 2.3.

Solution:

a. The spectrum for the sampled signal is sketched in Figure 2.14.

b. Since the maximum frequency of the analog signal is larger than that of the Nyquist frequency—that is, twice the maximum frequency of the analog signal is larger than the sampling rate—the sampling theorem condition is violated. The recovered spectrum is shown in Figure 2.15, where we see that aliasing noise occurs at 3 kHz.

2.2.1 Practical Considerations for Signal Sampling: Anti-Aliasing Filtering

In practice, the analog signal to be digitized may contain other frequency components whose frequencies are larger than the folding frequency, such as high-frequency noise. To satisfy the sampling theorem condition, we apply an anti-aliasing filter to limit the input analog signal, so that all the frequency components are less than the folding frequency (half of the sampling rate). Considering the worst case, where the analog signal to be sampled has a flat frequency spectrum, the band limited spectrum $X(f)$ and sampled spectrum $X_s(f)$ are depicted in Figure 2.16, where the shape of each replica in the sampled signal spectrum is the same as that of the anti-aliasing filter magnitude frequency response.

Due to nonzero attenuation of the magnitude frequency response of the anti-aliasing lowpass filter, the aliasing noise from the adjacent replica still appears in the baseband. However, the amount of aliasing noise is greatly reduced. We can also control the aliasing noise by either using a higher-order lowpass filter or increasing the sampling rate. For illustrative purpose, we use a Butterworth filter. The method can also be extended to other filter types such as the Chebyshev filter. The Butterworth magnitude frequency response with an order of n is given by

$$|H(f)| = \frac{1}{\sqrt{1 + \left(\frac{f}{f_c}\right)^{2n}}} \tag{2.7}$$

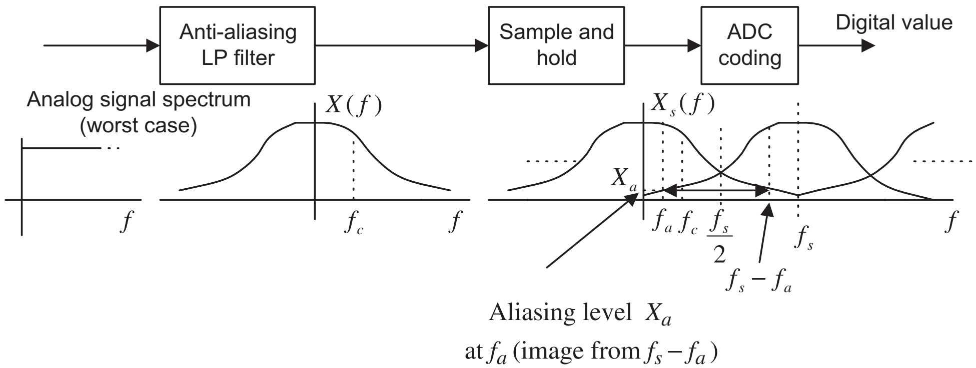


FIGURE 2.16

Spectrum of the sampled analog signal with a practical anti-aliasing filter.

For a second-order Butterworth lowpass filter with the unit gain, the transfer function (which will be discussed in Chapter 8) and its magnitude frequency response are given by

$$H(s) = \frac{(2\pi f_c)^2}{s^2 + 1.4141 \times (2\pi f_c)s + (2\pi f_c)^2} \tag{2.8}$$

$$|H(f)| = \frac{1}{\sqrt{1 + \left(\frac{f}{f_c}\right)^4}} \tag{2.9}$$

A unit gain second-order lowpass filter using a Sallen-Key topology is shown in Figure 2.17. Matching the coefficients of the circuit transfer function to that of the second-order Butterworth lowpass transfer function in Equation (2.10) gives the design formulas shown in Figure 2.17, where for a given cutoff

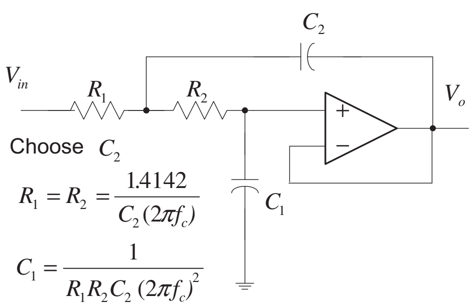


FIGURE 2.17

Second-order unit gain Sallen-Key lowpass filter.

frequency of f_c in Hz, and a capacitor value of C_2, we can determine the values for other elements using the formulas listed in the figure.

$$\frac{\frac{1}{R_1R_2C_1C_2}}{s^2+\left(\frac{1}{R_1C_2}+\frac{1}{R_2C_2}\right)s+\frac{1}{R_1R_2C_1C_2}}=\frac{(2\pi f_c)^2}{s^2+1.4141\times(2\pi f_c)s+(2\pi f_c)^2} \tag{2.10}$$

As an example, for a cutoff frequency of 3,400 Hz, and by selecting $C_2 = 0.01$ microfarad (μF), we get

$$R_1 = R_2 = 6,620\,\Omega,\ \text{and}\ C_1 = 0.005\,\mu F$$

Figure 2.18 shows the magnitude frequency response, where the absolute gain of the filter is plotted. As we can see, the absolute attenuation begins at the level of 0.7 at 3,400 Hz and reduces to 0.3 at 6,000 Hz. Ideally, we want the gain attenuation to be zero after 4,000 Hz if our sampling rate is 8,000 Hz. Practically speaking, aliasing will occur anyway with some degree. We will study achieving the higher-order analog filter via Butterworth and Chebyshev prototype function tables in Chapter 8. More details of the circuit realization for the analog filter can be found in Chen (1986).

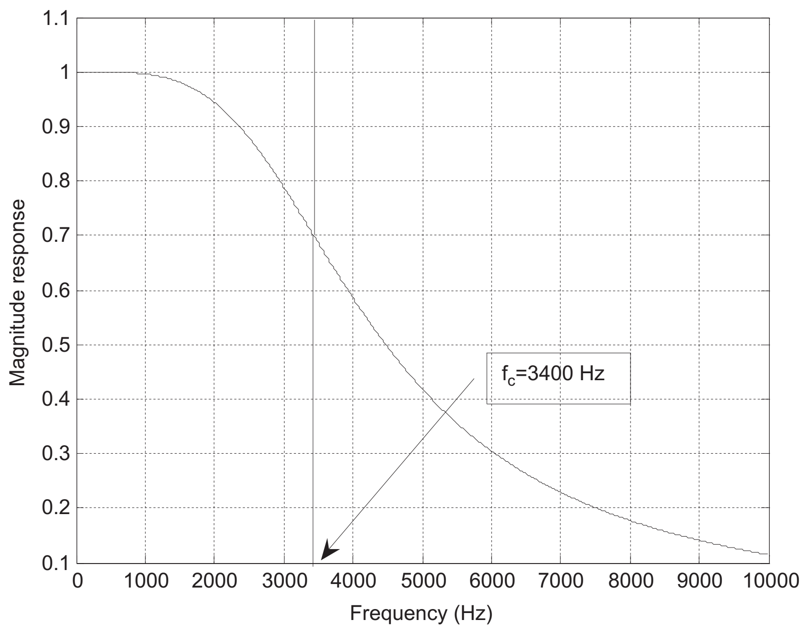


FIGURE 2.18

Magnitude frequency response of the second-order Butterworth lowpass filter.

According to Figure 2.16, we can derive the aliasing level percentage using the symmetry of the Butterworth magnitude function and its first replica. It follows that

$$\text{aliasing level } \% = \frac{X_a}{X(f)|_{f=f_a}} = \frac{|H(f)|_{f=f_s-f_a}}{|H(f)|_{f=f_a}} = \frac{\sqrt{1+\left(\frac{f_a}{f_c}\right)^{2n}}}{\sqrt{1+\left(\frac{f_s-f_a}{f_c}\right)^{2n}}} \quad \text{for} \quad 0 \le f \le f_c \tag{2.11}$$

With Equation (2.11), we can estimate the aliasing noise percentage, or choose a higher-order anti-aliasing filter to satisfy the requirement for the aliasing level percentage.

EXAMPLE 2.4

Given the DSP system shown in Figures 2.16 to 2.18, where a sampling rate of 8,000 Hz is used and the anti-aliasing filter is a second-order Butterworth lowpass filter with a cutoff frequency of 3.4 kHz, determine

a. the percentage of aliasing level at the cutoff frequency;
b. the percentage of aliasing level at a frequency of 1,000 Hz.

Solution:

$$f_s = 8{,}000, \; f_c = 3{,}400, \quad \text{and} \quad n = 2$$

a. Since $f_a = f_c = 3{,}400$ Hz, we compute

$$\text{aliasing level } \% = \frac{\sqrt{1+\left(\frac{3.4}{3.4}\right)^{2\times 2}}}{\sqrt{1+\left(\frac{8-3.4}{3.4}\right)^{2\times 2}}} = \frac{1.4142}{2.0858} = 67.8\%$$

b. With $f_a = 1{,}000$ Hz, we have

$$\text{aliasing level } \% = \frac{\sqrt{1+\left(\frac{1}{3.4}\right)^{2\times 2}}}{\sqrt{1+\left(\frac{8-1}{3.4}\right)^{2\times 2}}} = \frac{1.03007}{4.3551} = 23.05\%$$

Let us examine another example with an increased sampling rate.

EXAMPLE 2.5

Given the DSP system shown in Figures 2.16 to 2.18, where a sampling rate of 16,000 Hz is used and the anti-aliasing filter is a second-order Butterworth lowpass filter with a cutoff frequency of 3.4 kHz, determine the percentage of aliasing level at the cutoff frequency.

Solution:

$$f_s = 16{,}000, \; f_c = 3{,}400, \quad \text{and} \quad n = 2$$

Since $f_a = f_c = 3{,}400$ Hz, we have

$$\text{aliasing level \%} = \frac{\sqrt{1 + \left(\frac{3.4}{3.4}\right)^{2\times 2}}}{\sqrt{1 + \left(\frac{16 - 3.4}{3.4}\right)^{2\times 2}}} = \frac{1.4142}{13.7699} = 10.26\%$$

In comparison with the result in Example 2.4, increasing the sampling rate can reduce the aliasing level.

The following example shows how to choose the order of the anti-aliasing filter.

EXAMPLE 2.6

Given the DSP system shown in Figure 2.16, where a sampling rate of 40,000 Hz is used, the anti-aliasing filter is the Butterworth lowpass filter with a cutoff frequency 8 kHz, and the percentage of aliasing level at the cutoff frequency is required to be less than 1%, determine the order of the anti-aliasing lowpass filter.

Solution:

Using $f_s = 40{,}000$, $f_c = 8{,}000$, and $f_a = 8{,}000$ Hz, we start at order 1 and increase the filter order until the requirement is met.

$$n = 1, \quad \text{aliasing level \%} = \frac{\sqrt{1 + \left(\frac{8}{8}\right)^{2\times 1}}}{\sqrt{1 + \left(\frac{40 - 8}{8}\right)^{2\times 1}}} = \frac{1.4142}{\sqrt{1 + (4)^2}} = 34.30\%$$

$$n = 2, \quad \text{aliasing level \%} = \frac{1.4142}{\sqrt{1 + (4)^4}} = 8.82\%$$

$$n = 3, \quad \text{aliasing level \%} = \frac{1.4142}{\sqrt{1 + (4)^6}} = 2.21\%$$

$$n = 4, \quad \text{aliasing level \%} = \frac{1.4142}{\sqrt{1 + (4)^8}} = 0.55\% < 1\%$$

To satisfy the 1% aliasing level requirement, we choose $n = 4$.

2.2.2 Practical Considerations for Signal Reconstruction: Anti-Image Filter and Equalizer

The analog signal recovery for a practical DSP system is illustrated in Figure 2.19.

As shown in Figure 2.19, the DAC unit converts the processed digital signal $y(n)$ to a sampled signal $y_s(t)$, and then the hold circuit produces the sample-and-hold voltage $y_H(t)$. The transfer function of the hold circuit can be derived as

$$H_h(s) = \frac{1 - e^{-sT}}{sT} \tag{2.12}$$

We can obtain the frequency response of the DAC with the hold circuit by substituting $s = j\omega$ in Equation (2.12). It follows that

$$H_h(\omega) = e^{-j\omega T/2}\frac{\sin(\omega T/2)}{\omega T/2} \tag{2.13}$$

The magnitude and phase responses are given by

$$|H_h(\omega)| = \left|\frac{\sin(\omega T/2)}{\omega T/2}\right| = \left|\frac{\sin(x)}{x}\right| \tag{2.14}$$

$$\angle H_h(\omega) = -\omega T/2 \tag{2.15}$$

where $x = \omega T/2$. In terms of Hz, we have

$$|H_h(f)| = \left|\frac{\sin(\pi f T)}{\pi f T}\right| \tag{2.16}$$

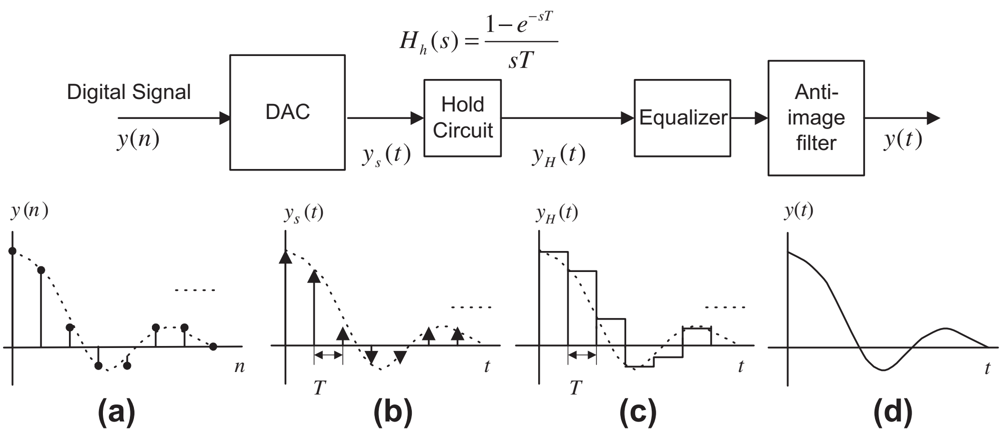


FIGURE 2.19

Signal notations at the practical reconstruction stage. (a) Processed digital signal. (b) Recovered ideal sampled signal. (c) Recovered sample-and-hold voltage. (d) Recovered analog signal.

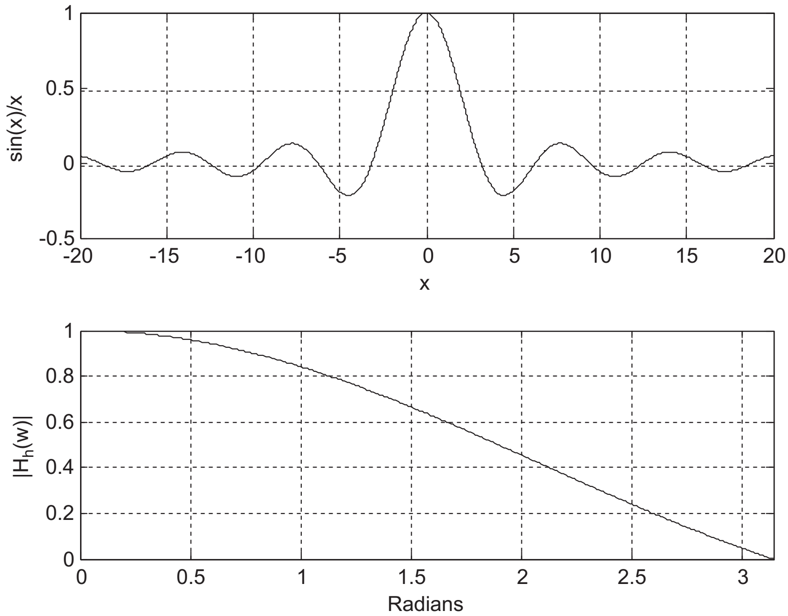


FIGURE 2.20

Sample-and-hold lowpass filtering effect.

$$\angle H_h(f) = -\pi f T \tag{2.17}$$

The plot of the magnitude effect is shown in Figure 2.20.

The magnitude frequency response acts like lowpass filtering and shapes the sampled signal spectrum of $Y_s(f)$. This shaping effect distorts the sampled signal spectrum $Y_s(f)$ in the desired frequency band, as illustrated in Figure 2.21. On the other hand, the spectral images are attenuated

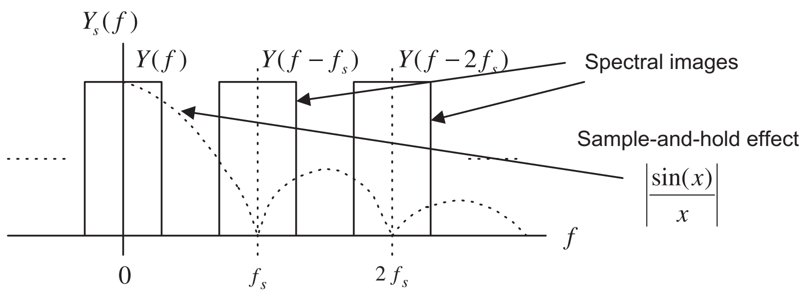


FIGURE 2.21

Sample-and-hold effect and distortion.

due to the lowpass effect of $\sin(x)/x$. This sample-and-hold effect can help us design the anti-image filter.

As shown in Figure 2.21, the percentage of distortion in the desired frequency band is given by

$$\begin{aligned} \text{distortion } \% &= (1 - |H_h(f)|) \times 100\% \\ &= \left(1 - \left|\frac{\sin(\pi f T)}{\pi f T}\right|\right) \times 100\% \end{aligned} \tag{2.18}$$

EXAMPLE 2.7

Given a DSP system with a sampling rate of 8,000 Hz and a hold circuit used after DAC, determine

a. the percentage of distortion at a frequency of 3,400 Hz;
b. the percentage of distortion at a frequency of 1,000 Hz.

Solution:

a. Since $fT = 3,400 \times 1/8,000 = 0.425$,

$$\text{distortion } \% = \left(1 - \left|\frac{\sin(0.425\pi)}{0.425\pi}\right|\right) \times 100\% = 27.17\%$$

b. Since $fT = 1,000 \times 1/8,000 = 0.125$,

$$\text{distortion } \% = \left(1 - \left|\frac{\sin(0.125\pi)}{0.125\pi}\right|\right) \times 100\% = 2.55\%$$

To overcome the sample-and-hold effect, the following methods can be applied.

1. We can compensate the sample-and-hold shaping effect using an equalizer whose magnitude response is opposite to the shape of the hold circuit magnitude frequency response, which is shown as the solid line in Figure 2.22.
2. We can increase the sampling rate using oversampling and interpolation methods when a higher sampling rate is available at the DAC. Using the interpolation will increase the sampling rate without affecting the signal bandwidth, so that the baseband spectrum and its images are separated further apart and a lower-order anti-aliasing filter can be used. This subject will be discussed in Chapter 12.
3. We can change the DAC configuration and perform digital pre-equalization using a flexible digital filter whose magnitude frequency response is against the spectral shape effect due to the hold circuit. Figure 2.23 shows a possible implementation. In this way, the spectral shape effect can be balanced before the sampled signal passes through the hold circuit. Finally, the anti-image filter will remove the rest of images and recover the desired analog signal.

The following practical example will illustrate the design of an anti-image filter using a higher sampling rate while making use of the sample-and-hold effect.

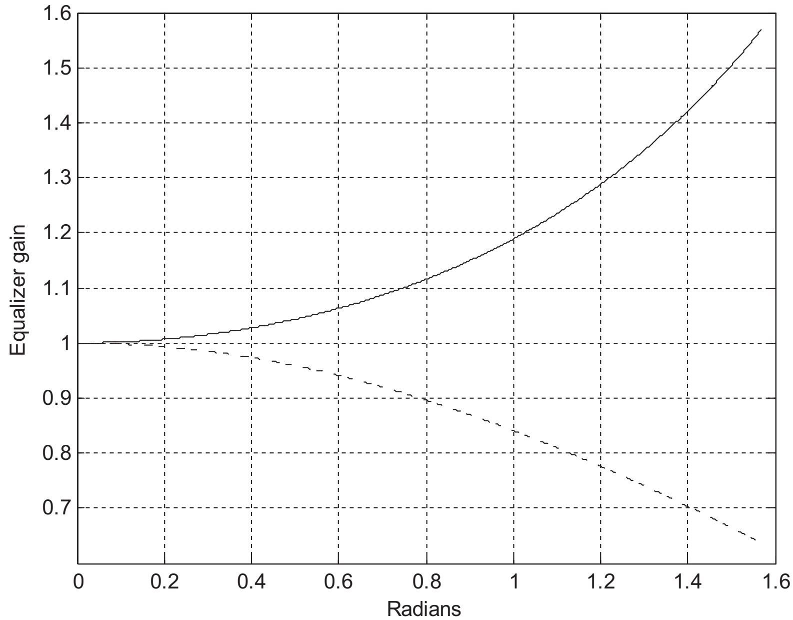


FIGURE 2.22

Ideal equalizer magnitude frequency response to overcome the distortion introduced by the sample-and-hold process.

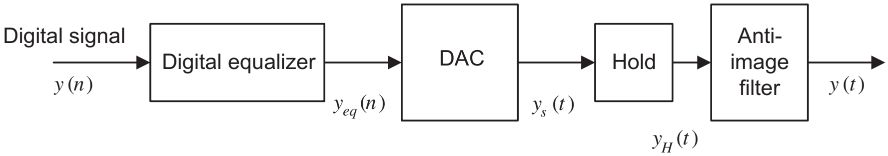


FIGURE 2.23

Possible implementation using a digital equalizer.

EXAMPLE 2.8

Determine the cutoff frequency and the order for the anti-image filter given a DSP system with a sampling rate of 16,000 Hz and specifications for the anti-image filter as shown in Figure 2.24.

Design requirements:

- Maximum allowable gain variation from 0 to 3,000 Hz = 2 dB
- 33 dB rejection at a frequency of 13,000 Hz
- Butterworth filter is assumed for the anti-image filter.

Solution:

We first determine the spectral shaping effects at $f = 3{,}000$ Hz and $f = 13{,}000$ Hz; that is,

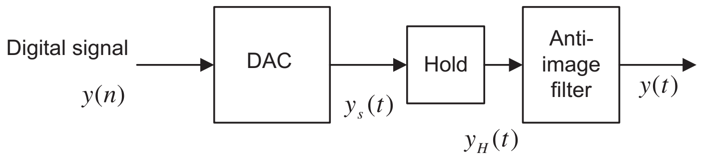


FIGURE 2.24

DSP recovery system for Example 2.8.

$$f = 3,000 \text{ Hz}, \; fT = 3,000 \times 1/16,000 = 0.1875$$

$$\text{gain} = \left|\frac{\sin(0.1875\pi)}{0.1875\pi}\right| = 0.9484 = -0.46 \text{ dB}$$

and

$$f = 13,000 \text{ Hz}, \; fT = 13,000 \times 1/16,000 = 0.8125$$

$$\text{gain} = \left|\frac{\sin(0.8125\pi)}{0.8125\pi}\right| = 0.2177 \approx -13 \text{ dB}$$

This gain would help the attenuation requirement.

Hence, the design requirements for the anti-image filter are

- Butterworth lowpass filter
- Maximum allowable gain variation from 0 to 3,000 Hz = 2−0.46 = 1.54 dB
- 33−13 = 20 dB rejection at frequency 13,000 Hz.

We set up equations using log operations of the Butterworth magnitude function as

$$20 \log(1 + (3,000/f_c)^{2n})^{1/2} \leq 1.54$$

$$20 \log(1 + (13,000/f_c)^{2n})^{1/2} \geq 20$$

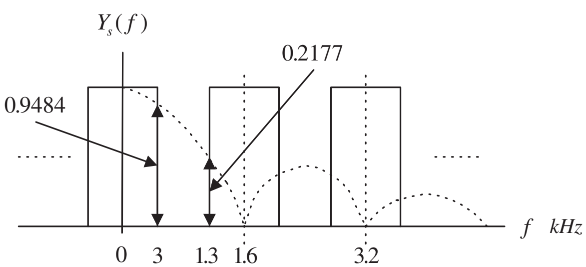


FIGURE 2.25

Spectral shaping by the sample-and-hold effect in Example 2.8.

From these two equations, we have to satisfy

$$(3{,}000/f_c)^{2n} = 10^{0.154} - 1$$

$$(13{,}000/f_c)^{2n} = 10^{2} - 1$$

Taking the ratio of these two equations yields

$$\left(\frac{13{,}000}{3{,}000}\right)^{2n} = \frac{10^2 - 1}{10^{0.154} - 1}$$

Then

$$n = \frac{1}{2}\log((10^2 - 1)/(10^{0.154} - 1))/\log(13{,}000/3{,}000) = 1.86 \approx 2$$

Finally, the cutoff frequency can be computed as

$$f_c = \frac{13{,}000}{(10^2 - 1)^{1/(2n)}} = \frac{13{,}000}{(10^2 - 1)^{1/4}} = 4{,}121.30 \text{ Hz}$$

$$f_c = \frac{3{,}000}{(10^{0.154} - 1)^{1/(2n)}} = \frac{3{,}000}{(10^{0.154} - 1)^{1/4}} = 3{,}714.23 \text{ Hz}$$

We choose the smaller one, that is,

$$f_c = 3{,}714.23 \text{ Hz}$$

With the filter order and cutoff frequency, we can realize the anti-image (reconstruction) filter using the second-order unit gain Sallen-Key lowpass filter described in Figure 2.17.

Note that the specifications for anti-aliasing filter designs are similar to anti-image (reconstruction) filters, except for their stopband edges. The anti-aliasing filter is designed to block the frequency components beyond the folding frequency before the ADC operation, while the reconstruction filter is designed to block the frequency components beginning at the lower edge of the first image after the DAC.

2.3 ANALOG-TO-DIGITAL CONVERSION, DIGITAL-TO-ANALOG CONVERSION, AND QUANTIZATION

During the ADC process, amplitudes of the analog signal to be converted have infinite precision. The continuous amplitude must be converted to digital data with finite precision, which is called *quantization*. Figure 2.26 shows quantization as a part of ADC.

There are several ways to implement ADC. The most common ones are

- Flash ADC
- Successive approximation ADC
- Sigma-delta ADC.

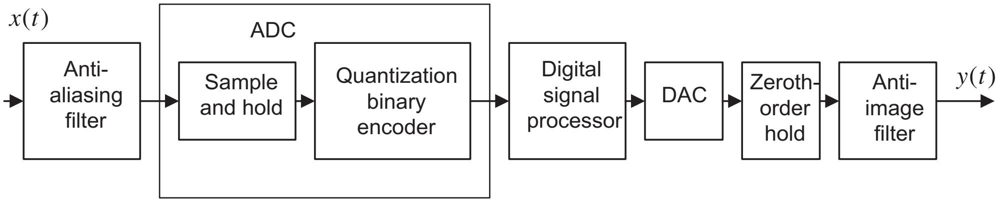


FIGURE 2.26

A block diagram for a DSP system.

In this chapter, we will focus on a simple 2-bit flash ADC unit, described in Figure 2.27, for illustrative purposes. Sigma-delta ADC will be studied in Chapter 12.

As shown in Figure 2.27, the 2-bit flash ADC unit consists of a serial reference voltage created by the equal value resistors, a set of comparators, and logic units. As an example, the reference voltages in the figure are 1.25 volts, 2.5 volts, 3.75 volts, and 5 volts, respectively. If an analog sample-and-hold voltage is $V_{in} = 3$ volts, then the lower two comparators will each output logic 1. Through the logic units, only the line labeled 10 is actively high, and the rest of lines are actively low. Hence, the encoding logic circuit outputs a 2-bit binary code of 10.

Flash ADC offers the advantage of high conversion speed, since all bits are acquired at the same time. Figure 2.28 illustrates a simple 2-bit DAC unit using an R-2R ladder. The DAC contains the R-2R ladder circuit, a set of single-throw switches, an adder, and a phase shifter. If a bit is logic 0, the switch connects a 2R resistor to ground. If a bit is logic 1, the corresponding 2R resistor is connected to the branch to the input of the operational amplifier (adder). When the operational amplifier operates in a linear range, the negative input is virtually equal to the positive input. The adder adds all the currents

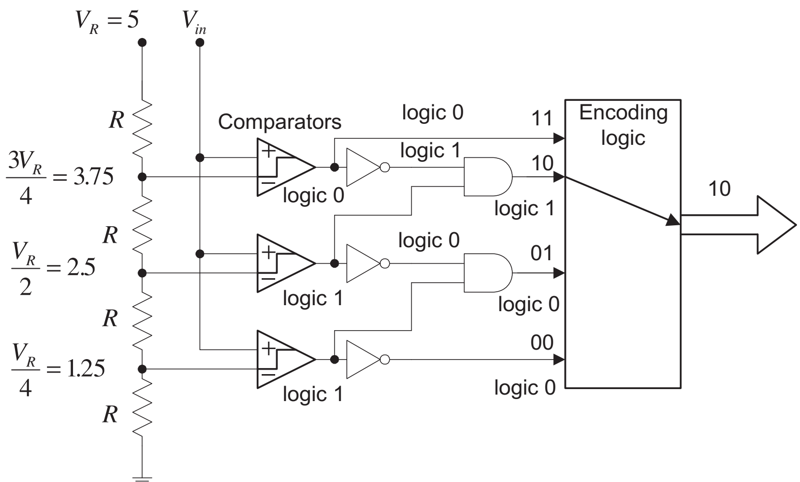


FIGURE 2.27

An example of a 2-bit flash ADC.

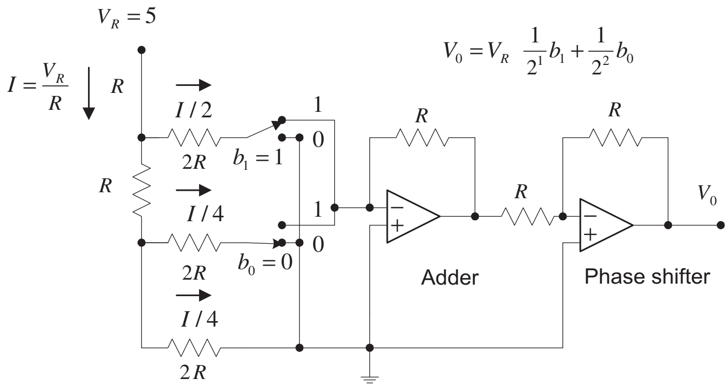


FIGURE 2.28

R-2R ladder DAC.

from all branches. The feedback resistor R in the adder provides overall amplification. The ladder network is equivalent to two 2R resistors in parallel. The entire network has a total current of

$$I = \frac{V_R}{R}$$

using Ohm's law, where V_R is the reference voltage, chosen to be 5 volts for our example. Hence, half of the total current flows into the b_1 branch, while the other half flows into the rest of the network. The halving process repeats for each branch successively to the lower bit branches to get lower bit weights. The second operational amplifier acts like a phase shifter to cancel the negative sign of the adder output. Using the basic electric circuit principle, we can determine the DAC output voltage as

$$V_0 = V_R \left(\frac{1}{2^1} b_1 + \frac{1}{2^2} b_0 \right)$$

where b_1 and b_0 are bits in the 2-bit binary code, with b_0 as the least significant bit (LSB).

In Figure 2.28, where we set $V_R = 5$ and $b_1 b_0 = 10$, the ADC output is expected to be

$$V_0 = 5 \times \left(\frac{1}{2^1} \times 1 + \frac{1}{2^2} \times 0 \right) = 2.5 \text{ volts}$$

As we can see, the recovered voltage of $V_0 = 2.5$ volts introduces voltage error as compared with $V_{in} = 3$ volts, discussed in the ADC stage. This is due to the fact that in the flash ADC unit, we use only four (i.e., finite) voltage levels to represent continuous (infinitely possible) analog voltage values. This is called *quantization error*, obtained by subtracting the original analog voltage from the recovered analog voltage. For our example, the quantization error is

$$V_0 - V_{in} = 2.5 - 3 = -0.5 \text{ volts}$$

Next, we focus on quantization development. The process of converting analog voltage with infinite precision to finite precision is called the *quantization process*. For example, if the digital processor has only a 3-bit word, the amplitudes can be converted into eight different levels.

A *unipolar quantizer* deals with analog signals ranging from 0 volt to a positive reference voltage, and a *bipolar quantizer* deals with analog signals ranging from a negative reference to a positive reference. The notations and general rules for quantization are as follows:

$$\Delta = \frac{(x_{\max} - x_{\min})}{L} \tag{2.19}$$

$$L = 2^m \tag{2.20}$$

$$i = round\left(\frac{x - x_{\min}}{\Delta}\right) \tag{2.21}$$

$$x_q = x_{\min} + i\Delta \quad i = 0, 1, \cdots, L-1 \tag{2.22}$$

where $x_{\max}$ and $x_{\min}$ are the maximum value and minimum values, respectively, of the analog input signal x. The symbol L denotes the number of quantization levels, which is determined by Equation (2.20), where m is the number of bits used in ADC. The symbol Δ is the step size of the quantizer or the ADC resolution. Finally, x_q indicates the quantization level, and i is an index corresponding to the binary code.

Figure 2.29 depicts a 3-bit unipolar quantizer and corresponding binary codes. From Figure 2.29, we see that $x_{\min} = 0$, $x_{\max} = 8\Delta$, and $m = 3$. Applying Equation (2.22) gives each quantization level as follows: $x_q = 0 + i\Delta$, $i = 0, 1, \cdots, L-1$, where $L = 2^3 = 8$ and i is the integer corresponding to the 3-bit binary code. Table 2.1 details quantization for each input signal subrange.

Similarly, a 3-bit bipolar quantizer and binary codes are shown in Figure 2.30, where we have $x_{\min} = -4\Delta$, $x_{\max} = 4\Delta$, and $m = 3$. The corresponding quantization table is given in Table 2.2.

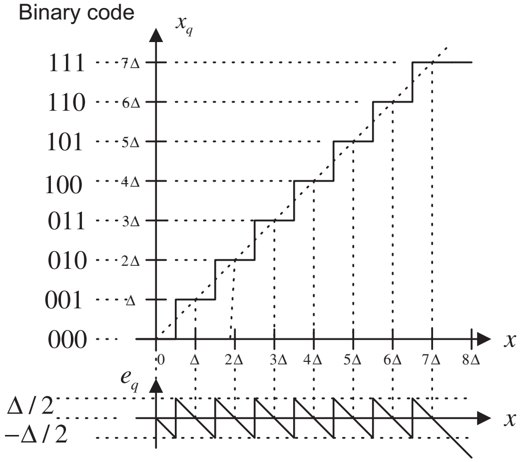


FIGURE 2.29

Characteristics of the unipolar quantizer.

Table 2.1 Quantization Table for the 3-Bit Unipolar Quantizer (step size $= \Delta = (x_{max} - x_{min})/2^3$, x_{max} = maximum voltage, and $x_{min} = 0$)

Binary Code	Quantization Level x_q (V)	Input Signal Subrange (V)
0 0 0	0	$0 \le x < 0.5\Delta$
0 0 1	Δ	$0.5\Delta \le x < 1.5\Delta$
0 1 0	2Δ	$1.5\Delta \le x < 2.5\Delta$
0 1 1	3Δ	$2.5\Delta \le x < 3.5\Delta$
1 0 0	4Δ	$3.5\Delta \le x < 4.5\Delta$
1 0 1	5Δ	$4.5\Delta \le x < 5.5\Delta$
1 1 0	6Δ	$5.5\Delta \le x < 6.5\Delta$
1 1 1	7Δ	$6.5\Delta \le x < 7.5\Delta$

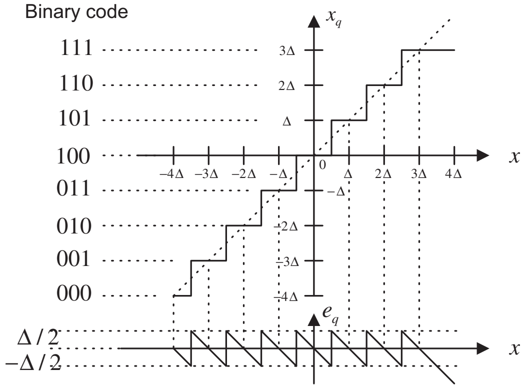


FIGURE 2.30

Characteristics for the bipolar quantizer.

EXAMPLE 2.9

Assuming that a 3-bit ADC channel accepts analog input ranging from 0 to 5 volts, determine

a. the number of quantization levels;
b. the step size of the quantizer or resolution;
c. the quantization level when the analog voltage is 3.2 volts;
d. the binary code produced by the ADC.

Solution:
Since the range is from 0 to 5 volts and a 3-bit ADC is used, we have

$$x_{min} = 0 \text{ volt}, \; x_{max} = 5 \text{ volts}, \quad \text{and} \quad m = 3 \text{ bits}$$

Table 2.2 Quantization Table for the 3-Bit Bipolar Quantizer (step size $= \Delta = (x_{max} - x_{min})/2^3$, x_{max} = maximum voltage, and $x_{min} = -x_{max}$)

Binary Code	Quantization Level x_q (V)	Input Signal Subrange (V)
0 0 0	-4Δ	$-4\Delta \le x < -3.5\Delta$
0 0 1	-3Δ	$-3.5\Delta \le x < -2.5\Delta$
0 1 0	-2Δ	$-2.5\Delta \le x < -1.5\Delta$
0 1 1	$-\Delta$	$-1.5\Delta \le x < -0.5\Delta$
1 0 0	0	$-0.5\Delta \le x < 0.5\Delta$
1 0 1	Δ	$0.5\Delta \le x < 1.5\Delta$
1 1 0	2Δ	$1.5\Delta \le x < 2.5\Delta$
1 1 1	3Δ	$2.5\Delta \le x < 3.5\Delta$

a. Using Equation (2.20), we get the number of quantization levels as

$$L = 2^m = 2^3 = 8$$

b. Applying Equation (2.19) yields

$$\Delta = \frac{5-0}{8} = 0.625 \text{ volts}$$

c. When $x = 3.2 \frac{\Delta}{0.625} = 5.12\Delta$, from Equation (2.21) we get

$$i = round\left(\frac{x - x_{min}}{\Delta}\right) = round(5.12) = 5$$

From Equation (2.22), we determine the quantization level as

$$x_q = 0 + 5\Delta = 5 \times 0.625 = 3.125 \text{ volts}$$

d. The binary code is determined as 101, either from Figure 2.29 or Table 2.1.

After quantizing the input signal x, the ADC produces binary codes, as illustrated in Figure 2.31.

The DAC process is shown in Figure 2.32. As shown in the figure, the DAC unit takes the binary codes from the DS processor. Then it converts the binary code using the zero-order hold circuit to reproduce the sample-and-hold signal. Assuming that the spectrum distortion due to sample-and-hold effect can be ignored for our illustration, the recovered sample-and-hold signal is further processed using the anti-image filter. Finally, the analog signal is produced.

When the DAC outputs the analog amplitude x_q with finite precision, it introduces quantization error defined as

$$e_q = x_q - x \tag{2.23}$$

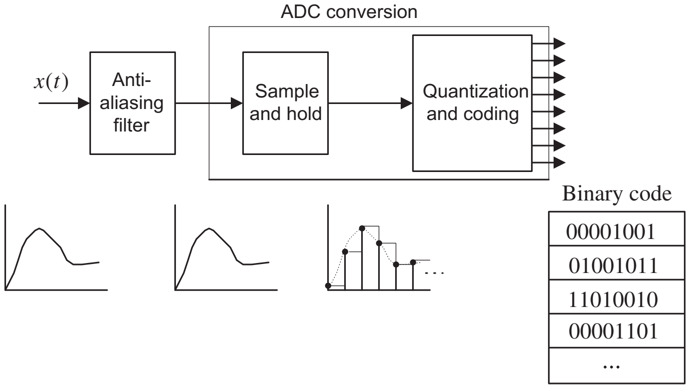


FIGURE 2.31

Typical ADC process.

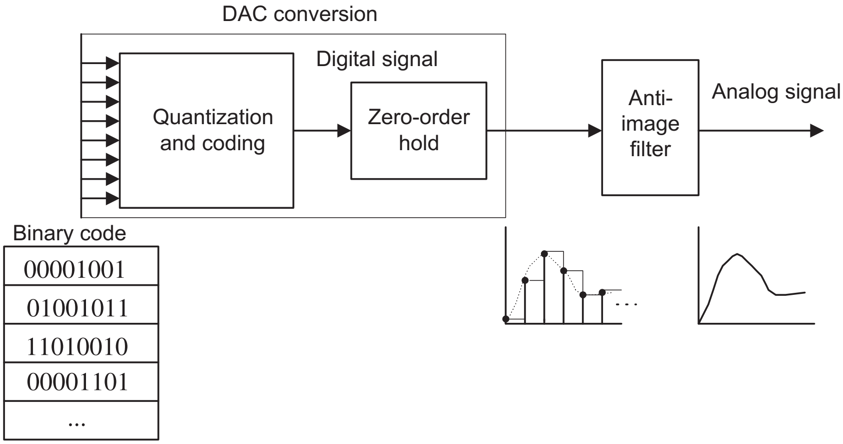


FIGURE 2.32

Typical DAC process.

The quantization error as shown in Figure 2.29 is bounded by half of the step size, that is,

$$-\frac{\Delta}{2} \leq e_q \leq \frac{\Delta}{2} \tag{2.24}$$

where Δ is the quantization step size, or the ADC resolution. We also refer to Δ as $V_{\min}$ (minimum detectable voltage) or the LSB value of the ADC.

EXAMPLE 2.10

Using Example 2.9, determine the quantization error when the analog input is 3.2 volts.

Solution:

Using Equation (2.23), we obtain

$$e_q = x_q - x = 3.125 - 3.2 = -0.075 \text{ volts}$$

Note that the quantization error is less than the half of the step size, that is,

$$|e_q| = 0.075 < \Delta/2 = 0.3125 \text{ volts}$$

In practice, we can empirically confirm that the quantization error appears in uniform distribution when the step size is much smaller than the dynamic range of the signal samples and we have a sufficiently large number of samples. Based on the theory of probability and random variables, the power of quantization noise is related to the quantization step and given by

$$E(e_q^2) = \frac{\Delta^2}{12} \tag{2.25}$$

where $E()$ is the expectation operator, which actually averages the squared values of the quantization error (the reader can get more information from the texts by Roddy and Coolen (1997); Tomasi (2004); and Stearns and Hush (1990)). The ratio of signal power to quantization noise power (SNR) can be expressed as

$$SNR = \frac{E(x^2)}{E(e_q^2)} \tag{2.26}$$

If we express the SNR in terms of decibels (dB), we have

$$SNR_{dB} = 10 \cdot \log_{10}(SNR) \text{ dB} \tag{2.27}$$

Substituting Equation (2.25) and $E(x^2) = x^2{}_{rms}$ into Equation (2.27), we achieve

$$SNR_{dB} = 10.79 + 20 \cdot \log_{10}\left(\frac{x_{rms}}{\Delta}\right) \tag{2.28}$$

where x_{rms} is the RMS (root mean squared) value of the signal to be quantized x.

Practically, the SNR can be calculated using the following formula:

$$SNR = \frac{\frac{1}{N}\sum_{n=0}^{N-1} x^2(n)}{\frac{1}{N}\sum_{n=0}^{N-1} e_q^2(n)} = \frac{\sum_{n=0}^{N-1} x^2(n)}{\sum_{n=0}^{N-1} e_q^2(n)} \tag{2.29}$$

where $x(n)$ is the nth sample amplitude and $e_q(n)$ the quantization error from quantizing $x(n)$.

EXAMPLE 2.11

If the analog signal to be quantized is a sinusoidal waveform, that is,

$$x(t) = A\sin(2\pi \times 1{,}000t)$$

and if the bipolar quantizer uses m bits, determine the SNR in terms of m bits.

Solution:

Since $x_{rms} = 0.707A$ and $\Delta = 2A/2^m$, substituting x_{rms} and Δ into Equation (2.28) leads to

$$\begin{aligned} SNR_{dB} &= 10.79 + 20\cdot\log_{10}\left(\frac{0.707A}{2A/2^m}\right) \\ &= 10.79 + 20\cdot\log_{10}(0.707/2) + 20m\cdot\log_{10}2 \end{aligned}$$

After simplifying the numerical values, we get

$$SNR_{dB} = 1.76 + 6.02m\,\text{dB} \tag{2.30}$$

EXAMPLE 2.12

For a speech signal, if a ratio of the RMS value over the absolute maximum value of the analog signal (Roddy and Coolen, 1997) is given, that is, $\left(\frac{x_{rms}}{|x|_{\max}}\right)$, and the ADC quantizer uses m bits, determine the SNR in terms of m bits.

Solution:

Since

$$\Delta = \frac{x_{\max} - x_{\min}}{L} = \frac{2|x|_{\max}}{2^m}$$

substituting Δ in Equation (2.28) achieves

$$\begin{aligned} SNR_{dB} &= 10.79 + 20\cdot\log_{10}\left(\frac{x_{rms}}{2|x|_{\max}/2^m}\right) \\ &= 10.79 + 20\cdot\log_{10}\left(\frac{x_{rms}}{|x|_{\max}}\right) + 20m\log_{10}2 - 20\log_{10}2 \end{aligned}$$

Thus, after numerical simplification, we have

$$SNR_{dB} = 4.77 + 20\cdot\log_{10}\left(\frac{x_{rms}}{|x|_{\max}}\right) + 6.02m \tag{2.31}$$

From Examples 2.11 and 2.12, we observed that increasing 1 bit of the ADC quantizer can improve SNR due to quantization by 6 dB.

EXAMPLE 2.13

Given a sinusoidal waveform with a frequency of 100 Hz,

$$x(t) = 4.5 \cdot \sin(2\pi \times 100t)$$

sampled at 8,000 Hz,

a. write a MATLAB program to quantize $x(t)$ using 4 bits to obtain and plot the quantized signal x_q, assuming the signal range is between -5 and 5 volts;

b. calculate the SNR due to quantization.

Solution:

a. Program 2.1. MATLAB program for Example 2.13.

```
%Example 2.13
clear all; close all
disp('Generate 0.02-second sine wave of 100 Hz and Vp=5');
fs=8000;                                          % Sampling rate
T=1/fs;                                           % Sampling interval
t=0:T:0.02;                                       % Duration of 0.02 second
sig = 4.5*sin(2*pi*100*t);                        % Generate sinusoids
bits = input('input number of bits =>');
lg = length(sig);                                 % Length of signal vector sig
for x=1:lg
  [Index(x) pq] = biquant(bits, -5,5, sig(x));   % Output quantized index
```

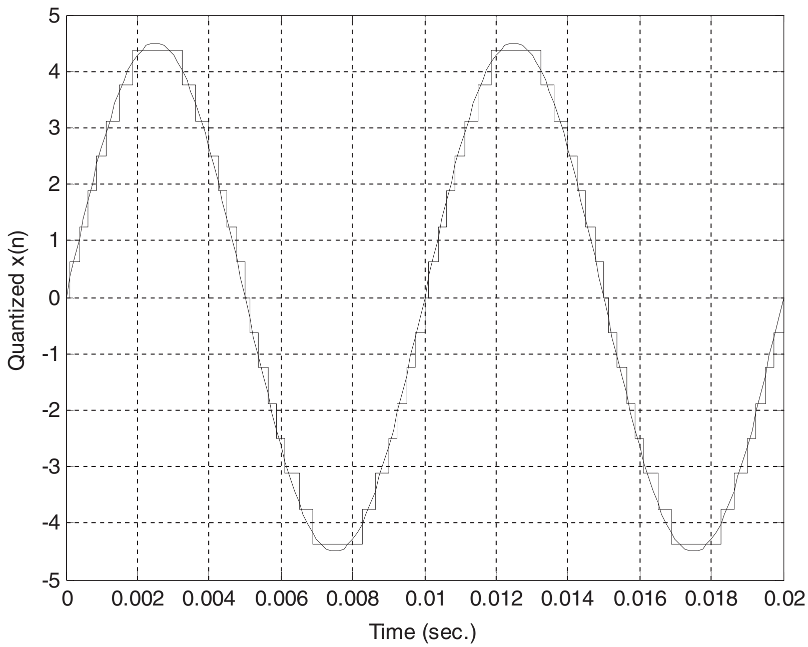


FIGURE 2.33

Comparison of the quantized signal and the original signal.

```
end
   % transmitted
   % received
   for x=1:lg
    qsig(x) = biqtdec(bits, -5,5, Index(x));       % Recover the quantized value
   end
    qerr = qsig-sig;                              % Calculate quantized error
   stairs(t,qsig); hold                           % Plot signal in staircase style
   plot(t,sig); grid;                             % Plot signal
   xlabel('Time (sec.)'); ylabel('Quantized x(n)')
   disp('Signal to noise ratio due to quantization noise')
   snr(sig,qsig);
```

b. Theoretically, applying Equation (2.30) gives

$$SNR_{dB} = 1.76 + 6.02 \cdot 4 = 25.84 \text{ dB}$$

Practically, using Equation (2.29), the simulated result is obtained as

$$SNR_{dB} = 25.78 \text{ dB}$$

It is clear from this example that the ratios of signal power to noise power due to quantization achieved from theory and from simulation are very close. Next, we look at an example for quantizing a speech signal.

EXAMPLE 2.14

Given the speech signal sampled at 8,000 Hz in the file *we.dat*,

a. write a MATLAB program to quantize $x(t)$ using 4-bit quantizers to obtain the quantized signal x_q, assuming the signal range is from -5 to 5 volts;
b. plot the original speech, quantized speech, and quantization error, respectively;
c. calculate the SNR due to quantization using the MATLAB program.

Solution:
a. Program 2.2 MATLAB program for Example 2.14.

```
%Example 2.14
clear all; close all
disp('load speech: We');
load we.dat                  % Load speech data at the current folder
sig = we;                    % Provided by the instructor
fs=8000;                     % Sampling rate
lg=length(sig);              % Length of signal vector
T=1/fs;                      % Sampling period
t=[0:1:lg-1]*T;              % Time instants in seconds
sig=4.5*sig/max(abs(sig));   % Normalizes speech in the range from -4.5 to 4.5
Xmax = max(abs(sig));        % Maximum amplitude
Xrms = sqrt( sum(sig .*
sig) / length(sig))          % RMS value
disp('Xrms/Xmax')
k=Xrms/Xmax
```

```
disp('20*log10(k)=>');
k = 20*log10(k)
bits = input('input number of bits =>');
lg = length(sig);
for x=1:lg
  [Index(x) pq] = biquant(bits, -5,5, sig(x)); % Output quantized index
end
% transmitted
% received
for x=1:lg
 qsig(x) = biqtdec(bits, -5,5, Index(x));      % Recover the quantized value
end
  qerr = sig-qsig;                             % Calculate the quantized error
subplot(3,1,1);plot(t,sig);
ylabel('Original speech');title('we.dat: we');
subplot(3,1,2);stairs(t, qsig);grid
ylabel('Quantized speech')
subplot(3,1,3);stairs(t, qerr);grid
ylabel('Quantized error')
xlabel('Time (sec.)');axis([0 0.25 -1 1]);
disp('signal to noise ratio due to quantization noise')
snr(sig,qsig);                                 % Signal to noise ratio in dB:
                                               % sig = original signal vector,
                                               % qsig =quantized signal vector
```

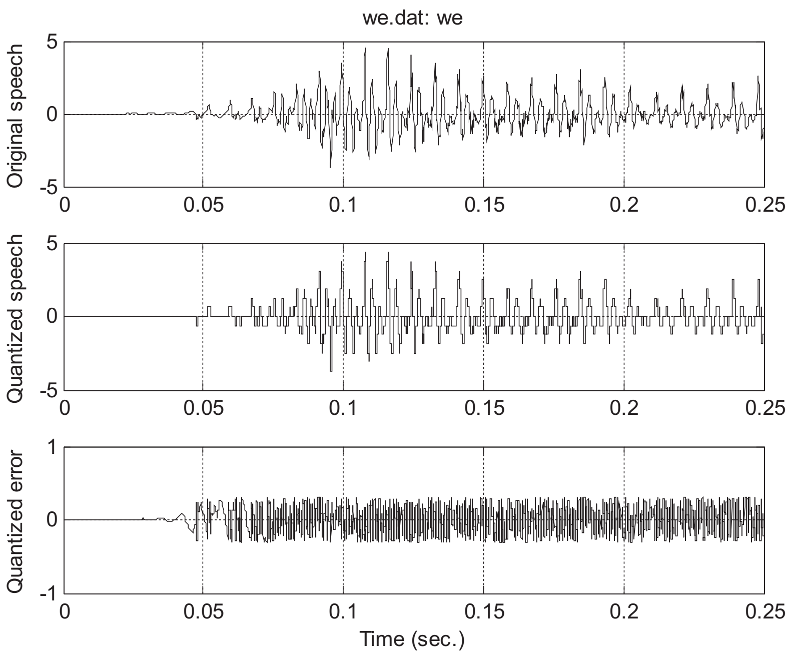


FIGURE 2.34

Original speech, quantized speech using the 4-bit bipolar quantizer, and quantization error.

b. In Figure 2.34, the top plot shows the speech wave to be quantized, while the middle plot displays the quantized speech signal using 4 bits. The bottom plot shows the quantization error. It also shows that the absolute value of quantization error is uniformly distributed in a range between -0.3125 and 0.3125.

c. From the MATLAB program, we have $\frac{x_{rms}}{|x|_{\max}} = 0.203$. Theoretically, from Equation (2.31), it follows that

$$SNR_{\mathrm{dB}} = 4.77 + 20\log_{10}\left(\frac{x_{rms}}{|x|_{\max}}\right) + 6.02 \cdot 4$$
$$= 4.77 + 20\log_{10}(0.203) + 6.02 \cdot 4 = 15 \text{ dB}$$

On the other hand, the simulated result using Equation (2.29) gives

$$SNR_{\mathrm{dB}} = 15.01 \text{ dB}$$

Results for SNRs from Equations (2.31) and (2.29) are very close in this example.

2.4 SUMMARY

1. Analog signal is sampled at a fixed time interval so the ADC will convert the sampled voltage level to the digital value; this is called the sampling process.
2. The fixed time interval between two samples is the sampling period, and the reciprocal of the sampling period is the sampling rate. Half of the sampling rate is the folding frequency (Nyquist limit).
3. The sampling theorem condition that the sampling rate must be larger than twice the highest frequency of the sampled analog signal must be met in order for the analog signal to be recovered.
4. The sampled spectrum is explained using the following well-known formula:

$$X_s(f) = \cdots + \frac{1}{T}X(f+f_s) + \frac{1}{T}X(f) + \frac{1}{T}X(f-f_s) + \cdots$$

   That is, the sampled signal spectrum is a scaled and shifted version of its analog signal spectrum and its replicas centered at the frequencies that are multiples of the sampling rate.
5. The analog anti-aliasing lowpass filter is used before ADC to remove frequency components higher than the folding frequency to avoid aliasing.
6. The reconstruction (analog lowpass) filter is adopted after DAC to remove the spectral images that exist in the sample-and-hold signal and obtain the smoothed analog signal. The sample-and-hold DAC effect may distort the baseband spectrum, but it also reduces image spectrum.
7. Quantization occurs when the ADC unit converts the analog signal amplitude with infinite precision to digital data with finite precision (a finite number of codes).
8. When the DAC unit converts a digital code to a voltage level, quantization error occurs. The quantization error is bounded by half of the quantization step size (ADC resolution), which is a ratio of the full range of the signal over the number of quantization levels (number of codes).
9. The performance of the quantizer in terms of the signal to quantization noise ratio (SNR), in dB, is related to the number of bits in ADC. Increasing each ADC code by 1 bit will improve SNR by 6 dB due to quantization.

2.5 MATLAB PROGRAMS

Program 2.3. MATLAB function for uniform quantization encoding.

```
function [ I, pq]= biquant(NoBits,Xmin,Xmax,value)
% function pq = biquant(NoBits, Xmin, Xmax, value)
% This routine is created for simulation of the uniform quantizer.
%
% NoBits: number of bits used in quantization
% Xmax: overload value
% Xmin: minimum value
% value: input to be quantized
% pq: output of quantized value
% I: coded integer index
 L=2^NoBits;
 delta=(Xmax-Xmin)/L;
 I=round((value-Xmin)/delta);
 if ( I==L)
  I=I-1;
 end
 if I<0
  I=0;
 end
 pq=Xmin+I*delta;
```

Program 2.4. MATLAB function for uniform quantization decoding.

```
function pq = biqtdec(NoBits,Xmin,Xmax,I)
% function pq = biqtdec(NoBits,Xmin, Xmax, I)
% This routine recovers the quantized value.
%
% NoBits: number of bits used in quantization
% Xmax: overload value
% Xmin: minimum value
% pq: output of quantized value
% I: coded integer index
 L=2^NoBits;
 delta=(Xmax-Xmin)/L;
 pq=Xmin+I*delta;
```

Program 2.5. MATLAB function for calculation of signal to quantization noise ratio.

```
function snr = calcsnr(speech, qspeech)
% function snr = calcsnr(speech, qspeech)
% This routine was created to calculate SNR.
%
% speech: original speech waveform
% qspeech: quantized speech
% snr: output SNR in dB
%
 qerr = speech-qspeech;
 snr = 10*log10(sum(speech.*speech)/sum(qerr.*qerr))
```

2.6 PROBLEMS

2.1. Given an analog signal

$$x(t) = 5\cos(2\pi \cdot 1,500t), \quad \text{for } t \geq 0$$

sampled at a rate of 8,000 Hz,

a. sketch the spectrum of the original signal;

b. sketch the spectrum of the sampled signal from 0 kHz up to 20 kHz.

2.2. Given an analog signal

$$x(t) = 5\cos(2\pi \cdot 2,500t) + 2\cos(2\pi \cdot 3,200t), \quad \text{for } t \geq 0$$

sampled at a rate of 8,000 Hz,

a. sketch the spectrum of the sampled signal up to 20 kHz;

b. sketch the recovered analog signal spectrum if an ideal lowpass filter with a cutoff frequency of 4 kHz is used to filter the sampled signal in order to recover the original signal.

2.3. Given an analog signal

$$x(t) = 3\cos(2\pi \cdot 1,500t) + 2\cos(2\pi \cdot 2,200t), \quad \text{for } t \geq 0$$

sampled at a rate of 8,000 Hz,

a. sketch the spectrum of the sampled signal up to 20 kHz;

b. sketch the recovered analog signal spectrum if an ideal lowpass filter with a cutoff frequency of 4 kHz is used to filter the sampled signal in order to recover the original signal.

2.4. Given an analog signal

$$x(t) = 3\cos(2\pi \cdot 1,500t) + 2\cos(2\pi \cdot 4,200t), \quad \text{for } t \geq 0$$

sampled at a rate of 8,000 Hz,

a. sketch the spectrum of the sampled signal up to 20 kHz;

b. sketch the recovered analog signal spectrum if an ideal lowpass filter with a cutoff frequency of 4 kHz is used to filter the sampled signal in order to recover the original signal.

2.5. Given an analog signal

$$x(t) = 5\cos(2\pi \cdot 2,500t) + 2\cos(2\pi \cdot 4,500t), \quad \text{for } t \geq 0$$

sampled at a rate of 8,000 Hz,

a. sketch the spectrum of the sampled signal up to 20 kHz;

b. sketch the recovered analog signal spectrum if an ideal lowpass filter with a cutoff frequency of 4 kHz is used to filter the sampled signal in order to recover the original signal;

c. determine the frequency/frequencies of aliasing noise.

2.6. Assuming a continuous signal is given as

$$x(t) = 10\cos(2\pi \cdot 5,500t) + 5\sin(2\pi \cdot 7,500t), \quad \text{for } t \geq 0$$

sampled at a rate of 8,000 Hz,

a. sketch the spectrum of the sampled signal up to 20 kHz;

b. sketch the recovered analog signal spectrum if an ideal lowpass filter with a cutoff frequency of 4 kHz is used to filter the sampled signal in order to recover the original signal;

c. determine the frequency/frequencies of aliasing noise.

2.7. Assuming a continuous signal is given as

$$x(t) = 8\cos(2\pi \cdot 5,000t) + 5\sin(2\pi \cdot 7,000t), \quad \text{for } t \geq 0$$

sampled at a rate of 8,000 Hz,

a. sketch the spectrum of the sampled signal up to 20 kHz;

b. sketch the recovered analog signal spectrum if an ideal lowpass filter with a cutoff frequency of 4 kHz is used to filter the sampled signal in order to recover the original signal;

c. determine the frequency/frequencies of aliasing noise.

2.8. Assuming a continuous signal is given as

$$x(t) = 10\cos(2\pi \cdot 5,000t) + 5\sin(2\pi \cdot 7,500t), \quad \text{for } t \geq 0$$

sampled at a rate of 8,000 Hz,

a. sketch the spectrum of the sampled signal up to 20 kHz;

b. sketch the recovered analog signal spectrum if an ideal lowpass filter with a cutoff frequency of 4 kHz is used to filter the sampled signal in order to recover the original signal;

c. determine the frequency/frequencies of aliasing noise.

2.9. Given a Butterworth type second-order anti-aliasing lowpass filter (Figure 2.35), determine the values of circuit elements if we want the filter to have a cutoff frequency of 1,000 Hz.

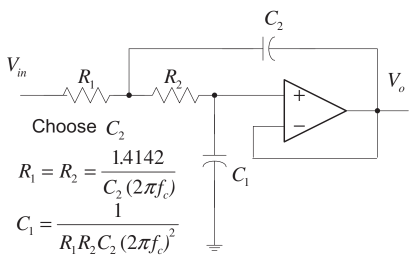


FIGURE 2.35

Filter circuit in Problem 2.9.

2.10. From Problem 2.9, determine the percentage of aliasing level at the frequency of 500 Hz, assuming that the sampling rate is 4,000 Hz.

2.11. Given a Butterworth type second-order anti-aliasing lowpass filter (Figure 2.36), determine the values of circuit elements if we want the filter to have a cutoff frequency of 800 Hz.

2.12. From Problem 2.11, determine the percentage of aliasing level at the frequency of 400 Hz, assuming that the sampling rate is 4,000 Hz.

2.13. Given a DSP system in which a sampling rate of 8,000 Hz is used and the anti-aliasing filter is a second-order Butterworth lowpass filter with a cutoff frequency of 3.2 kHz, determine

a. the percentage of aliasing level at the cutoff frequency;

b. the percentage of aliasing level at the frequency of 1,000 Hz.

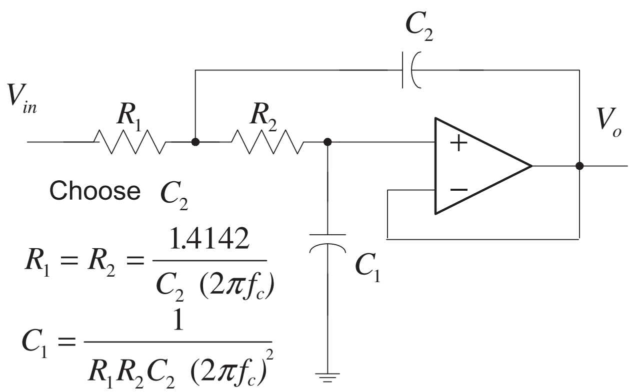
2.14. Given a DSP system in which a sampling rate of 8,000 Hz is used and the anti-aliasing filter is a Butterworth lowpass filter with a cutoff frequency 3.2 kHz, determine the order of the Butterworth lowpass filter required to make the percentage of aliasing level at the cutoff frequency less than 10%.

2.15. Given a DSP system in which a sampling rate of 8,000 Hz is used and the anti-aliasing filter is a second-order Butterworth lowpass filter with a cutoff frequency of 3.1 kHz, determine

a. the percentage of aliasing level at the cutoff frequency;

b. the percentage of aliasing level at a frequency of 900 Hz.

2.16. Given a DSP system in which a sampling rate of 8,000 Hz is used and the anti-aliasing filter is a Butterworth lowpass filter with a cutoff frequency 3.1 kHz, determine the order of the Butterworth lowpass filter required to make the percentage of aliasing level at the cutoff frequency less than 10%.



Choose C_2

$$R_1 = R_2 = \frac{1.4142}{C_2\ (2\pi f_c)}$$

$$C_1 = \frac{1}{R_1 R_2 C_2\ (2\pi f_c)^2}$$

FIGURE 2.36

Filter circuit in Problem 2.11.

2.17. Given a DSP system with a sampling rate of 8,000 Hz and assuming that the hold circuit is used after DAC, determine

a. the percentage of distortion at a frequency of 3,200 Hz;

b. the percentage of distortion at a frequency of 1,500 Hz.

2.18. A DSP system (Figure 2.37) is given with the following specifications:

Design requirements:

- Sampling rate 20,000 Hz
- Maximum allowable gain variation from 0 to 4,000 Hz = 2 dB
- 40 dB rejection at a frequency of 16,000 Hz
- Butterworth filter.

Determine the cutoff frequency and order for the anti-image filter.

2.19. Given a DSP system with a sampling rate of 8,000 Hz and assuming that the hold circuit is used after DAC, determine

a. the percentage of distortion at a frequency of 3,000 Hz;

b. the percentage of distortion at a frequency of 1,600 Hz.

2.20. A DSP system (Figure 2.38) is given with the following specifications:

Design requirements:

- Sampling rate 22,000 Hz
- Maximum allowable gain variation from 0 to 4,000 Hz = 2 dB

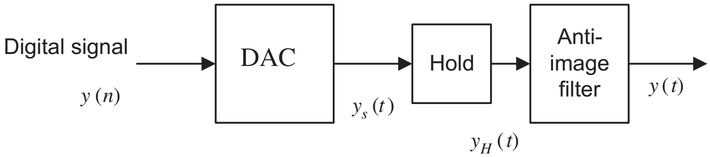


FIGURE 2.37

Analog signal reconstruction in Problem 2.18.

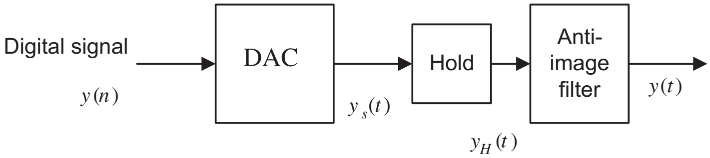


FIGURE 2.38

Analog signal reconstruction in Problem 2.20.

- 40 dB rejection at the frequency of 18,000 Hz
- Butterworth filter.

Determine the cutoff frequency and order for the anti-image filter.

2.21. Given the 2-bit flash ADC unit with an analog sample-and-hold voltage of 2 volts shown in Figure 2.39, determine the output bits.

2.22. Given the R-2R DAC unit with a 2-bit value defined as $b_1 b_0 = 01$ shown in Figure 2.40, determine the converted voltage.

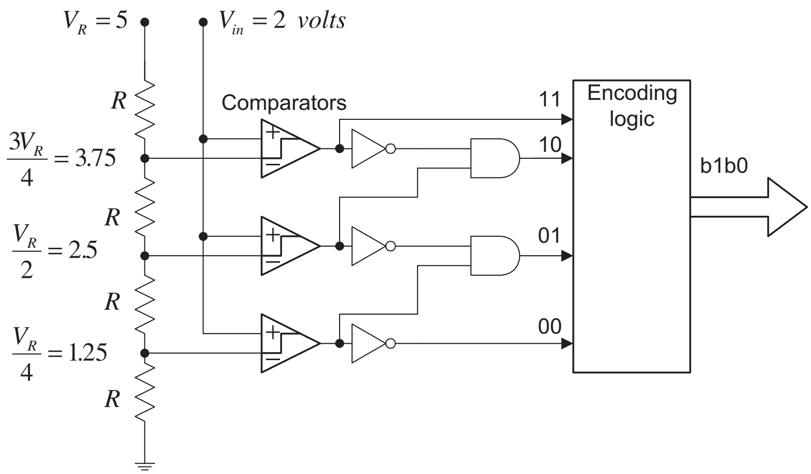


FIGURE 2.39

2-bit flash ADC in Problem 2.21.

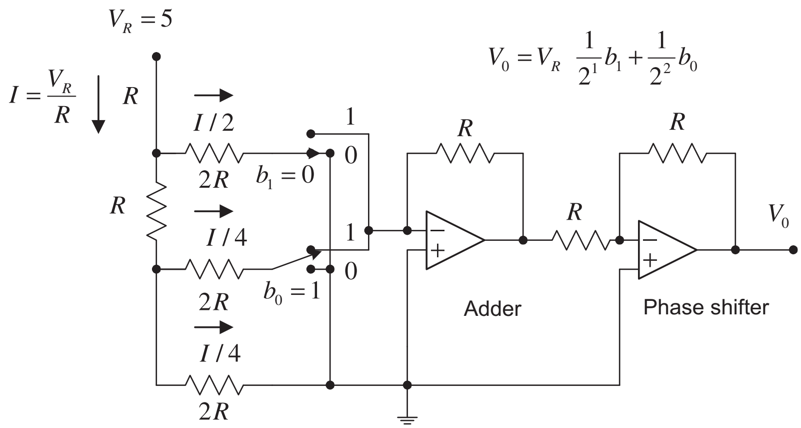


FIGURE 2.40

2-bit R-2R DAC in Problem 2.22.

2.23. Given the 2-bit flash ADC unit with an analog sample-and-hold voltage of 3.5 volts shown in Figure 2.39, determine the output bits.

2.24. Given the R-2R DAC unit with 2-bit values defined as $b_1 b_0 = 11$ and $b_1 b_0 = 10$ and shown in Figure 2.40, determine the converted voltages.

2.25. Assuming that a 4-bit ADC channel accepts analog input ranging from 0 to 5 volts, determine the following:

a. number of quantization levels;

b. step size of the quantizer or resolution;

c. quantization level when the analog voltage is 3.2 volts;

d. binary code produced by the ADC;

e. quantization error.

2.26. Assuming that a 5-bit ADC channel accepts analog input ranging from 0 to 4 volts, determine the following:

a. number of quantization levels;

b. step size of the quantizer or resolution;

c. quantization level when the analog voltage is 1.2 volts;

d. binary code produced by the ADC;

e. quantization error.

2.27. Assuming that a 3-bit ADC channel accepts analog input ranging from -2.5 to 2.5 volts, determine the following:

a. number of quantization levels;

b. step size of the quantizer or resolution;

c. quantization level when the analog voltage is -1.2 volts;

d. binary code produced by the ADC;

e. quantization error.

2.28. Assuming that a 8-bit ADC channel accepts analog input ranging from -2.5 to 2.5 volts, determine the following:

a. number of quantization levels;

b. step size of the quantizer or resolution;

c. quantization level when the analog voltage is 1.5 volts;

d. binary code produced by the ADC;

e. quantization error.

2.29. If the analog signal to be quantized is a sinusoidal waveform, that is,

$$x(t) = 9.5\sin(2,000 \times \pi t)$$

and if the bipolar quantizer uses 6 bits, determine

a. the number of quantization levels;

b. the quantization step size or resolution, Δ, assuming the signal range is from -10 to 10 volts;

c. the signal power to quantization noise power ratio.

2.30. For a speech signal, if the ratio of the RMS value over the absolute maximum value of the signal is given, that is, $\left(\frac{x_{rms}}{|x|_{\max}}\right) = 0.25$, and the ADC bipolar quantizer uses 6 bits, determine

a. the number of quantization levels;

b. the quantization step size or resolution, Δ, if the signal range is 5 volts;

c. the signal power to quantization noise power ratio.

2.6.1 Computer Problems with MATLAB

Use the MATLAB programs in Section 2.5 to solve the following problems.

2.31. Given a sinusoidal waveform of 100 Hz,

$$x(t) = 4.5\sin(2\pi \times 100t)$$

sample it at 8,000 samples per second and

a. write a MATLAB program to quantize $x(t)$ using a 6-bit bipolar quantizer to obtain the quantized signal x_q, assuming that the signal range is from -5 to 5 volts;

b. plot the original signal and quantized signal;

c. calculate the SNR due to quantization using the MATLAB program.

2.32. Given a signal waveform,

$$x(t) = 3.25\sin(2\pi \times 50t) + 1.25\cos(2\pi \times 100t + \pi/4)$$

sample it at 8,000 samples per second and

a. write a MATLAB program to quantize $x(t)$ using a 6-bit bipolar quantizer to obtain the quantized signal x_q, assuming that the signal range is from -5 to 5 volts;

b. plot the original signal and quantized signal;

c. calculate the SNR due to quantization using the MATLAB program.

2.33. Given a speech signal sampled at 8,000 Hz, as shown in Example 2.14,

a. write a MATLAB program to quantize $x(t)$ using a 6-bit bipolar quantizer to obtain the quantized signal x_q, assuming that the signal range is from -5 to 5 volts;

b. plot the original speech waveform, quantized speech, and quantization error;

c. calculate the SNR due to quantization using the MATLAB program.

2.6.2 MATLAB Projects

2.34. Performance evaluation of speech quantization:

Given an original speech segment "speech.dat" sampled at 8,000 Hz with each sample encoded in 16 bits, use Programs 2.3 to 2.5 and modify Program 2.2 to quantize the speech segment using 3 to 15 bits, respectively. The SNR in dB must be measured for each quantization. The MATLAB function: "sound(x/max(abs(x)),fs)" can be used to evaluate sound quality, where "x" is the speech segment while "fs" is the sampling rate of 8,000 Hz. In this project, create a plot of the measured SNR (dB) versus the number of bits and discuss the effect on the sound quality. For comparisons, plot the original speech and the quantized one using 3 bits, 8 bits, and 15 bits.

2.35. Performance evaluation of seismic data quantization:

The seismic signal, a measurement of the acceleration of ground motion, is required for applications in the study of geophysics. The seismic signal ("seismic.dat" provided by the US Geological Survey, Albuquerque Seismological Laboratory) has a sampling rate of 15 Hz with 6,700 data samples, and each sample is encoded using 32 bits. Quantizing each 32-bit sample down to the lower number of bits per sample can reduce the memory storage requirement with the trade-off of reduced signal quality. Use Programs 2.3 to 2.5 and modify Program 2.2 to quantize the seismic data using 13, 15, 17, …, 31 bits. The SNR in dB must be measured for each quantization. Create a plot of the measured SNR (dB) versus the number of bits. For comparison, plot the seismic data and the quantized one using 13 bits, 18 bits, 25 bits, and 31 bits.