Copyrighted Materials
Copyright © 2019 Elsevier Retrieved from app.knovel.com

CHAPTER 2

SIGNAL SAMPLING AND QUANTIZATION

CHAPTER OUTLINE

2.1 Sampling of Continuous Signal 13
2.2 Signal Reconstruction 20
2.2.1 Practical Considerations for Signal Sampling: Anti-Aliasing Filtering 24
2.2.2 Practical Considerations for Signal Reconstruction: Anti-Image Filter and Equalizer 28
2.3 Analog-to-Digital Conversion, Digital-to-Analog Conversion, and Quantization 35
2.4 Summary 47
2.5 MATLAB Programs 48
2.6 Problems 49

2.1 SAMPLING OF CONTINUOUS SIGNAL

As discussed in Chapter 1, Fig. 2.1 describes a simplified block diagram of a digital signal processing (DSP) system. The analog filter processes 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 time domain. Fig. 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. Furthermore, the infinite points are not appropriate to be processed by a digital signal processor or a computer, since they require infinite amount of memory and infinite amount of processing power for computations. Sampling can solve such a problem by taking samples at the fixed time interval as shown in Figs. 2.2 and 2.3, where the time T represents the sampling interval or sampling period in seconds.

As shown in Fig. 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 Fig. 2.2, where 14 sampled samples at their sampling time instants are plotted, each using a vertical bar with a solid circle at its top.

© 2019 Elsevier Inc. All rights reserved.

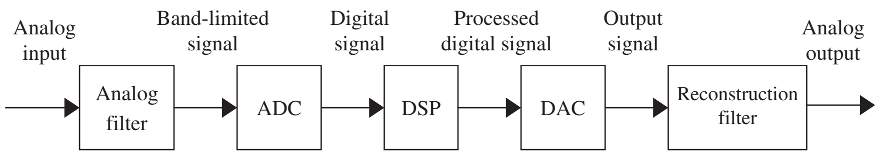

FIG. 2.1

A digital signal processing scheme.

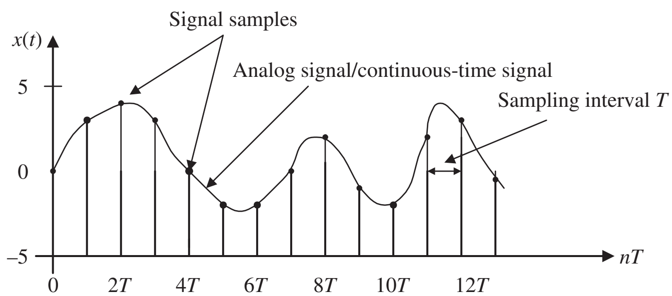

FIG. 2.2

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

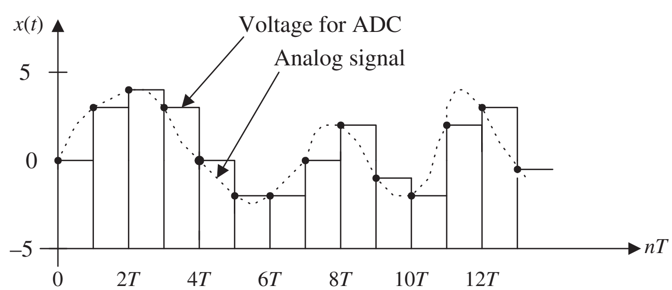

FIG. 2.3

Sample-and-hold analog voltage for ADC.

For a given sampling interval T, which is defined as the time span between two neighboring 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 $T = 125\ \mu s$, the sampling rate is determined as $f_s = 1/125\ \mu s = 8000$ samples per second (Hz).

After the analog signal is sampled, we obtain the sampled signal whose amplitude values are taken at the sampling instants, thus the processor is able to handle 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 of 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 8000 samples per second; to sample an audio signal with frequencies up to 20 kHz, a sampling rate with at least 40,000 samples per second or 40 kHz is required.

Fig. 2.4 illustrates sampling of two sinusoids, where the sampling interval between sample points is $T=0.01$ s, thus the sampling rate is $f_s=100$ 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 plot. We note 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

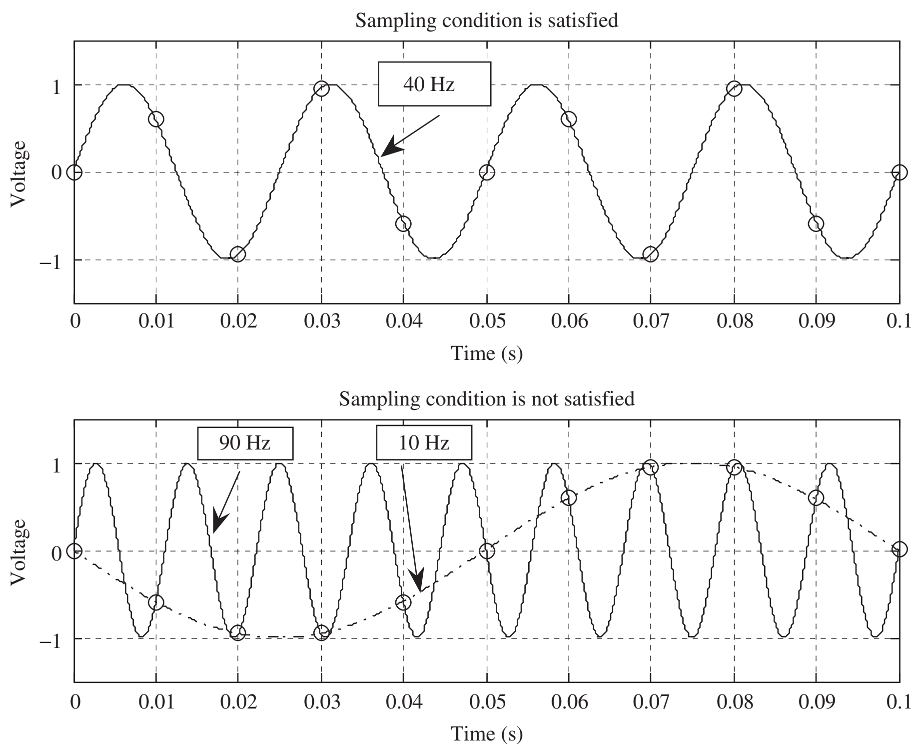

FIG. 2.4

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

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 the 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) to 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) to be applied after the digital-to-analog conversion (DAC).

Fig. 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 $T = 1/f_s$. The pulse train can be expressed as

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{2.2}$$

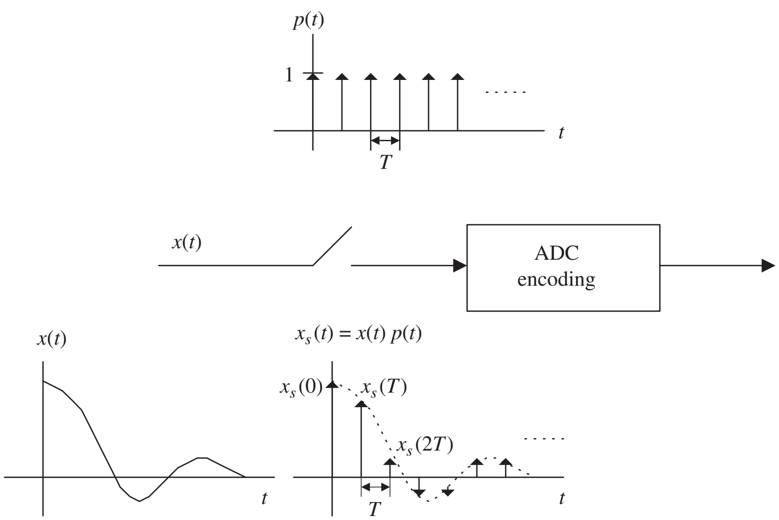

FIG. 2.5

The simplified sampling process.

with a fundamental frequency of $\omega_0 = 2\pi/T = 2\pi f_s$ rad/s. $p(t)$ can be expanded by the Fourier series (see Appendix B), that is,

$$p(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t}, \tag{2.3}$$

where a_k are the Fourier coefficients which can be determined by

$$a_k = \frac{1}{T}\int_{-\infty}^{\infty} \delta(t) e^{-jk\omega_0 t} dt = \frac{1}{T}. \tag{2.4}$$

Substituting Eq. (2.4) into Eq. (2.3), the pulse train is given by

$$p(t) = \sum_{k=-\infty}^{\infty} \frac{1}{T} e^{jk\omega_0 t}. \tag{2.5}$$

Again, substituting Eq. (2.5) into Eq. (2.1) leads to

$$x_s(t) = \sum_{k=-\infty}^{\infty} \frac{1}{T} x(t) e^{jk\omega_0 t}. \tag{2.6}$$

Applying Fourier transform (see Appendix B) on Eq. (2.6), we yield the following:

$$X_s(f) = FT\left\{\sum_{k=-\infty}^{\infty} \frac{1}{T} x(t) e^{jk\omega_0 t}\right\} = \sum_{k=-\infty}^{\infty} \frac{1}{T} FT\left\{x(t) e^{jk\omega_0 t}\right\} = \sum_{k=-\infty}^{\infty} \frac{1}{T} \int_{-\infty}^{\infty} \left\{x(t) e^{jk\omega_0 t}\right\} e^{-j\omega t} dt, \tag{2.7}$$

that is,

$$X_s(f) = \sum_{k=-\infty}^{\infty} \frac{1}{T} \int_{-\infty}^{\infty} x(t) e^{-j(\omega - k\omega_0)t} dt = \sum_{k=-\infty}^{\infty} \frac{1}{T} \int_{-\infty}^{\infty} x(t) e^{-j2\pi(f - kf_s)t} dt. \tag{2.8}$$

From the definition of Fourier transform, we note that

$$X(f) = \sum_{k=-\infty}^{\infty} \frac{1}{T} \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt. \tag{2.9}$$

Using Eq. (2.9), the original spectrum (frequency components) $X(f)$ and the sampled signal spectrum $X_s(f)$ in terms of Hz are related as

$$X_s(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X(f - kf_s), \tag{2.10}$$

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 $X(f \pm kf_s)$. The derivation for Eq. (2.10) can also be found in well-known texts (Ahmed and Natarajan, 1983; Ambardar, 1999; Alkin, 1993; Oppenheim and Shafer, 1975; Proakis and Manolakis, 2007).

Expanding Eq. (2.10) leads to the sampled signal spectrum in Eq. (2.11)

$$X_s(f) = \cdots + \frac{1}{T} X(f + f_s) + \frac{1}{T} X(f) + \frac{1}{T} X(f - f_s) + \cdots. \tag{2.11}$$

Eq. (2.11) indicates that the sampled signal spectrum is the sum of the scaled original spectrum and copies of its shifted versions, called *replicas*. The sketch of Eq. (2.11) is given in Fig. 2.6, where three possible sketches are classified. Given the original signal spectrum $X(f)$ plotted in Fig. 2.6A, the sampled signal spectrum according to Eq. (2.11) is plotted in Fig. 2.6B, 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. Fig. 2.6C 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 Fig. 2.6D, the original 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.

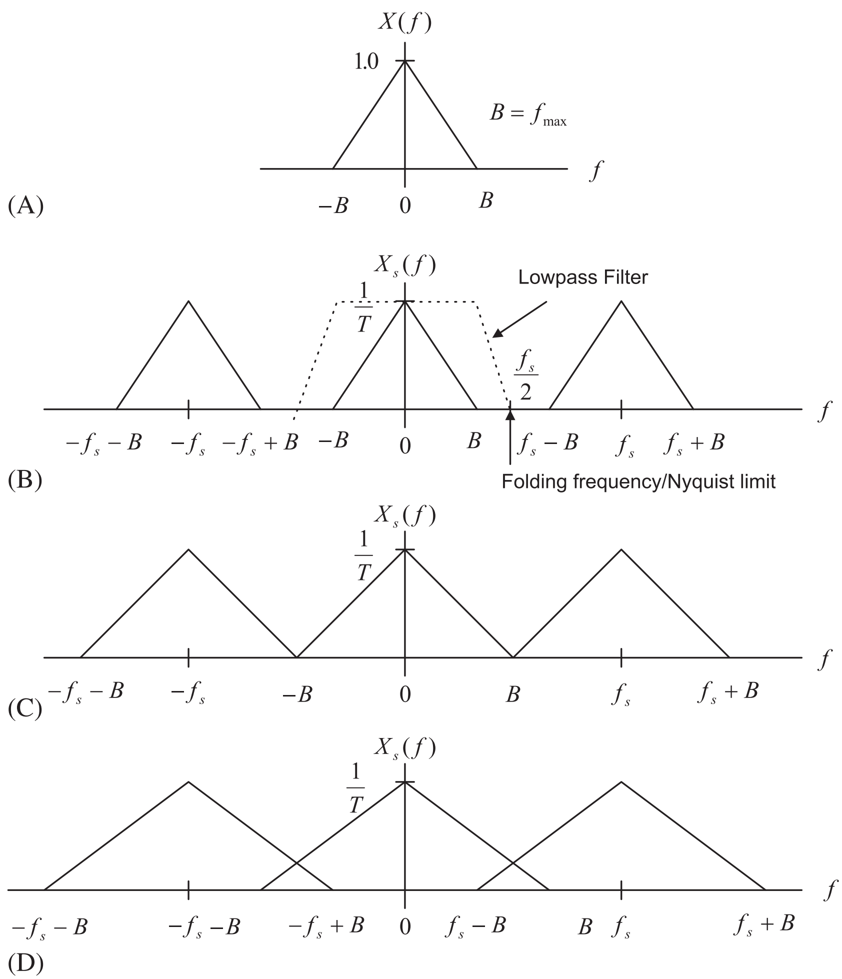

FIG. 2.6

Plots of the sampled signal spectrum. (A) Original signal spectrum. (B) Sampled signal spectrum for $f_s > 2B$. (C) Sampled signal spectrum for $f_s = 2B$. (D) Sampled signal spectrum for $f_s < 2B$.

From Fig. 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 $\pm kf_s$(multiples of the sampling rate) for each of $k=1, 2, 3, \ldots$.

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

$$f_s - f_{\max} \geq f_{\max}. \tag{2.12}$$

Solving Eq. (2.12) gives

$$f_s \geq 2f_{\max}. \tag{2.13}$$

In terms of frequency in radians per second, Eq. (2.13) is equivalent to

$$\omega_s \geq 2\omega_{\max}. \tag{2.14}$$

This fundamental conclusion is well known as the *Shannon sampling theorem*, which is formally described as 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) Sampling theorem establishes a minimum sampling rate for a given band-limited analog signal with the highest-frequency component $f_{\max}$. If the sampling rate satisfies Eq. (2.13), then the analog signal can be recovered via its sampled values using a lowpass filter, as described in Fig. 2.6B.

(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 its highest frequency 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

$x(t) = 5\cos(2\pi \times 1000t)$, for $t \geq 0$,

and is sampled at the rate 8000 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 1000 Hz, we can write the sine wave using Euler's identity:

$$5\cos(2\pi \times 1000t) = 5 \cdot \left(\frac{e^{j2\pi \times 1000t} + e^{-j2\pi \times 1000t}}{2}\right) = 2.5e^{j2\pi \times 1000t} + 2.5e^{-j2\pi \times 1000t},$$

Continued

EXAMPLE 2.1—CONT'D

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

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

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

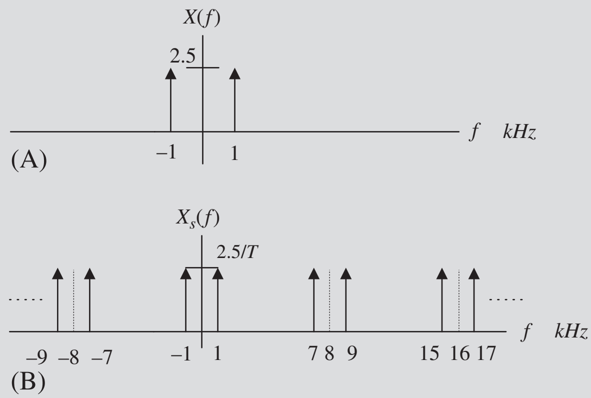

FIG. 2.7

Spectrum of the analog signal (A) and sampled signal (B) in Example 2.1.

(b) After the analog signal is sampled at the rate of 8000 Hz, the sampled signal spectrum and its replicas centered at the frequencies $\pm kf_s$, each with the scaled amplitude being $2.5/T$, are as shown in Fig. 2.7B:

Note that the spectrum of the sampled signal shown in Fig. 2.7B contains the images of the original spectrum shown in Fig. 2.7A; that the images repeat at multiples of the sampling frequency f_s (e.g., 8, 16, 24 kHz, etc.); 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 Fig. 2.8. First, the digitally processed data $y(n)$ are converted to the ideal impulse train $y_s(t)$, in which each impulse has its amplitude proportional to digital output $y(n)$, and two consecutive impulses are separated by a sampling period of T; second, the analog 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

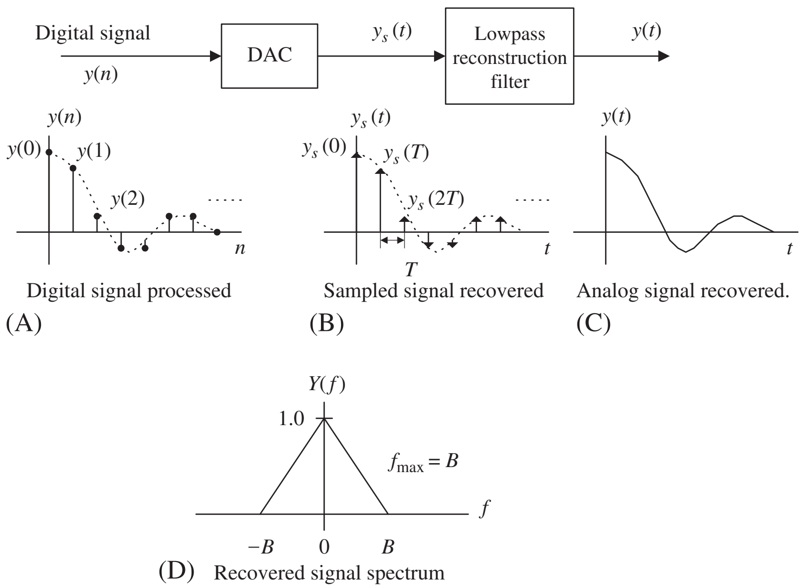

FIG. 2.8

Signal notations at its reconstruction stage: (A) Digital signal processed, (B) Sampled signal recovered, (C) Analog signal recovered, (D) Recovered signal spectrum.

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 Fig. 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 Fig. 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.

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

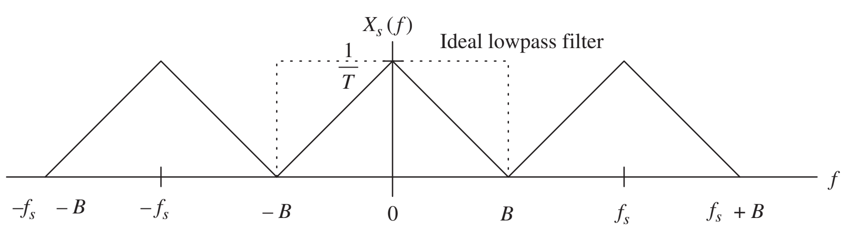

FIG. 2.9

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

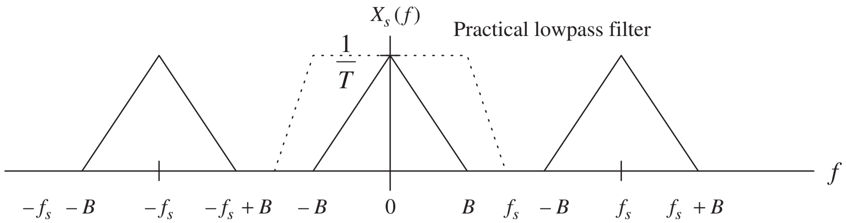

FIG. 2.10

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

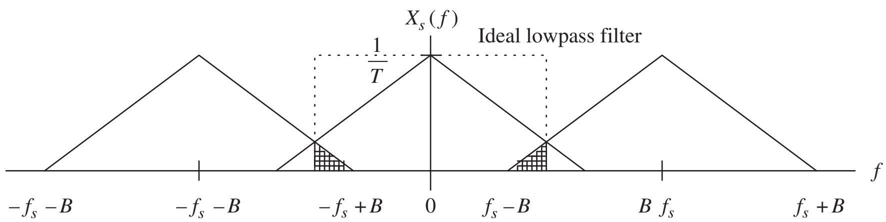

FIG. 2.11

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

In this case, as shown in Fig. 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, Fig. 2.11 depicts the spectral overlapping between the original baseband spectrum and the spectrum of the first replica and so on. Even we apply an ideal lowpass filter to remove these images, in the baseband there still some foldover frequency components from the adjacent replica. This is aliasing, where the recovered baseband spectrum suffers spectral distortion, that is, contains aliasing noise spectrum; in time domain, the recovered analog signal may consist of the aliasing noise frequency or frequencies. Hence, the recovered analog signal is incurably distorted.

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

Assuming that an analog signal is given by

$$x(t) = 5\cos(2\pi \times 2000t) + 3\cos(2\pi \times 3000t), \text{ for } t \geq 0$$

and it is sampled at the rate of 8000 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

$$x(t)=\frac{3}{2}e^{-j2\pi\times3000t}+\frac{5}{2}e^{-j2\pi\times2000t}+\frac{5}{2}e^{j2\pi\times2000t}+\frac{3}{2}e^{j2\pi\times3000t}.$$

The two-sided amplitude spectrum for the sinusoid is displayed in Fig. 2.12.

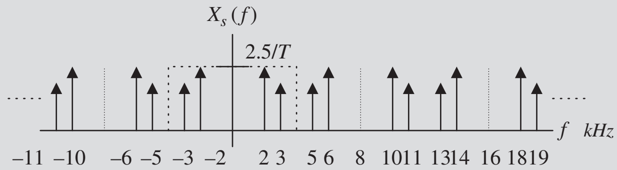

FIG. 2.12

Spectrum of the sampled 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 Fig. 2.13.

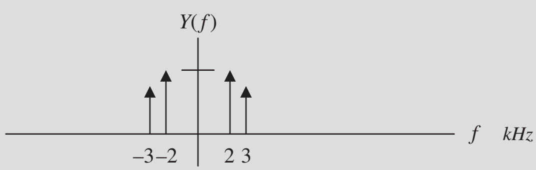

FIG. 2.13

Spectrum of the recovered signal in Example 2.2.

EXAMPLE 2.3

Given an analog signal

$x(t)=5\cos(2\pi\times2000t)+1\cos(2\pi\times5000t)$, for $t\geq0$,

which is sampled at a rate of 8000 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).

Solution:

(a) The spectrum for the sampled signal is sketched in Fig. 2.14:

Continued

EXAMPLE 2.3—CONT'D

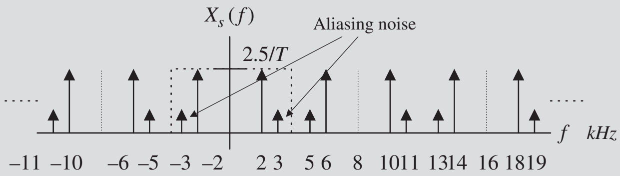

FIG. 2.14

Spectrum of the sampled signal in Example 2.3.

(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 Fig. 2.15, where we see that aliasing noise occurs at 3 kHz.

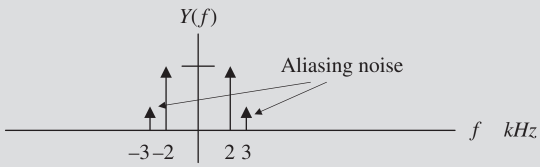

FIG. 2.15

Spectrum of the recovered signal in Example 2.3.

2.2.1 PRACTICAL CONSIDERATIONS FOR SIGNAL SAMPLING: ANTI-ALIASING FILTERING

In practice, the analog signal to be digitized may contain the 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 Fig. 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 level of the aliasing noise is greatly reduced. We can also control the aliasing noise level 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

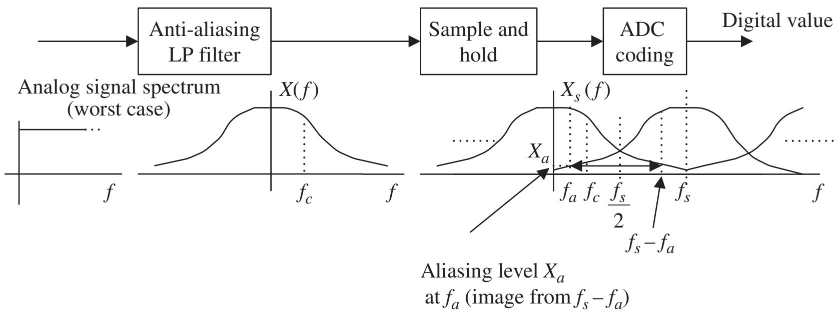

FIG. 2.16

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

$$|H(f)| = \frac{1}{\sqrt{1 + (f/f_c)^{2n}}}. \tag{2.15}$$

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

$$H(s) = \frac{(2\pi f_c)^2}{s^2 + 1.4141 \times (2\pi f_c)s + (2\pi f_c)^2}, \tag{2.16}$$

$$|H(f)| = \frac{1}{\sqrt{1 + (f/f_c)^4}}. \tag{2.17}$$

A unit gain second-order lowpass using a Sallen-Key topology is shown in Fig. 2.17. Matching the coefficients of the circuit transfer function to that of the second-order Butterworth lowpass transfer function in Eq. (2.16) as depicted in Eq. (2.18) gives the design formulas shown Fig. 2.17, where for a given cutoff 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.

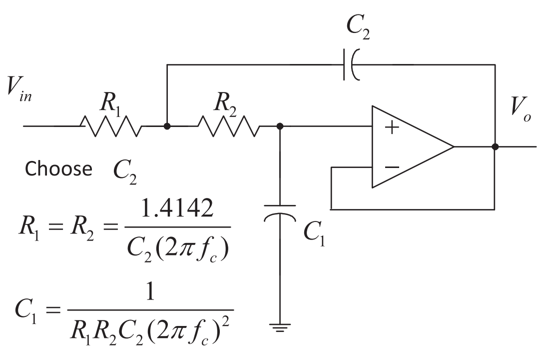

FIG. 2.17

Second-order unit-gain Sallen-Key lowpass filter.

$$\frac{\dfrac{1}{R_1R_2C_1C_2}}{s^2+\left(\dfrac{1}{R_1C_2}+\dfrac{1}{R_2C_2}\right)s+\dfrac{1}{R_1R_2C_1C_2}}=\frac{(2\pi f_c)^2}{s^2+1.4141\times(2\pi f_c)s+(2\pi f_c)^2}. \tag{2.18}$$

As an example, for a cutoff frequency of 3400 Hz, and by selecting $C_2=0.01$ microfarad (μF), we can get $R_1=R_2=6620\Omega$, and $C_1=0.005\mu F$.

Fig. 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 3400 Hz and reduces to 0.3 at 6000 Hz. Ideally, we want the gain attenuation to be zero after 4000 Hz if our sampling rate is 8000 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).

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

$$\begin{aligned}\text{Aliasing level }\% &= \frac{X_a}{X(f)|_{f=f_a}}=\frac{|H(f)|_{f=f_s-f_a}}{|H(f)|_{f=f_a}}\\ &= \frac{\sqrt{1+\left(\dfrac{f_a}{f_c}\right)^{2n}}}{\sqrt{1+\left(\dfrac{f_s-f_a}{f_c}\right)^{2n}}},\ \text{for } 0\le f\le f_c.\end{aligned} \tag{2.19}$$

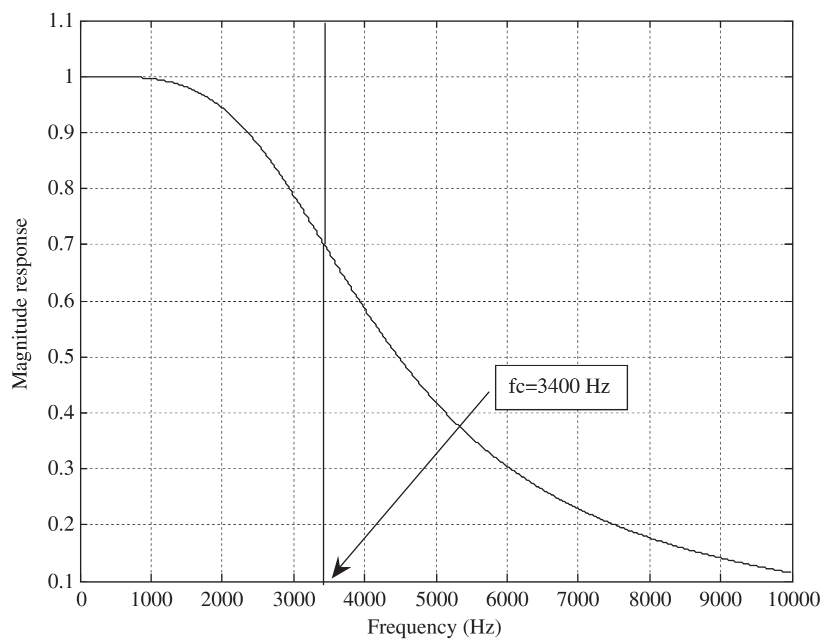

FIG. 2.18

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

With the Eq. (2.19), we can estimate the aliasing level, or choose a higher-order anti-aliasing filter to satisfy requirement for the percentage of aliasing level.

EXAMPLE 2.4

Given the DSP system shown in Figs. 2.16–2.18, where a sampling rate of 8000 Hz is used and the anti-aliasing filter is a second-order Butterworth lowpass filter with a cutoff frequency of 3.4 kHz,

(a) Determine the percentage of aliasing level at the cutoff frequency.

(b) Determine the percentage of aliasing level at the frequency of 1000 Hz.

Solution:

$$f_s = 8000,\ f_c = 3400,\ \text{ and }\ n = 2.$$

(a) Since $f_a = f_c = 3400$ 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 $f_a = 1000$ 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 Figs. 2.16–2.18, 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:

$$f_s = 16000, f_c = 3400, \text{and}\ n = 2.$$

Since $f_a = f_c = 3400$ Hz, we have

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

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

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

EXAMPLE 2.6

Given the DSP system shown in Fig. 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 $f_s = 40,\ 000$, $f_c = 8000$, and $f_a = 8000$ Hz, we try each of the following filters with the increasing number of the filter order.

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

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

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

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

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

2.2.2 PRACTICAL CONSIDERATIONS FOR SIGNAL RECONSTRUCTION: ANTI-IMAGE FILTER AND EQUALIZER

Consider that a unit impulse function $\delta(t)$ is passed through the hold transfer function, $H_h(s)$. The impulse response $h(t)$ is expected in Fig. 2.19 and can be expressed as

$$h(t) = u(t) - u(t-T), \tag{2.20}$$

where T is the sampling period and $u(t)$ is the unit step function, that is,

$$u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}. \tag{2.21}$$

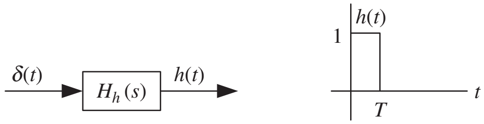

FIG. 2.19

Impulse response of the hold unit.

The transfer function $H_h(s)$ can be obtained by Laplace transform of the impulse response $h(t)$, that is,

$$H_h(s) = L\{h(t)\} = L\{u(t) - u(t-T)\} = \frac{1}{s} - \frac{1}{s}e^{-sT} = \frac{1-e^{-sT}}{s}. \tag{2.22}$$

The analog signal recovery for a practical DSP system is illustrated as shown in Fig. 2.20.

As shown in Fig. 2.20, 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 is derived as

$$H_h(s) = \frac{1-e^{-sT}}{s}. \tag{2.23}$$

We can obtain the frequency response of the DAC with the hold circuit by substituting $s = j\omega$ to Eq. (2.23). It follows that

$$H_h(j\omega) = \frac{1-e^{-j\omega T}}{j\omega} = e^{-j\omega T/2}\frac{\left(e^{j\omega T/2} - e^{-j\omega T/2}\right)}{j\omega} = Te^{-j\omega T/2}\frac{\sin(\omega T/2)}{\omega T/2}. \tag{2.24}$$

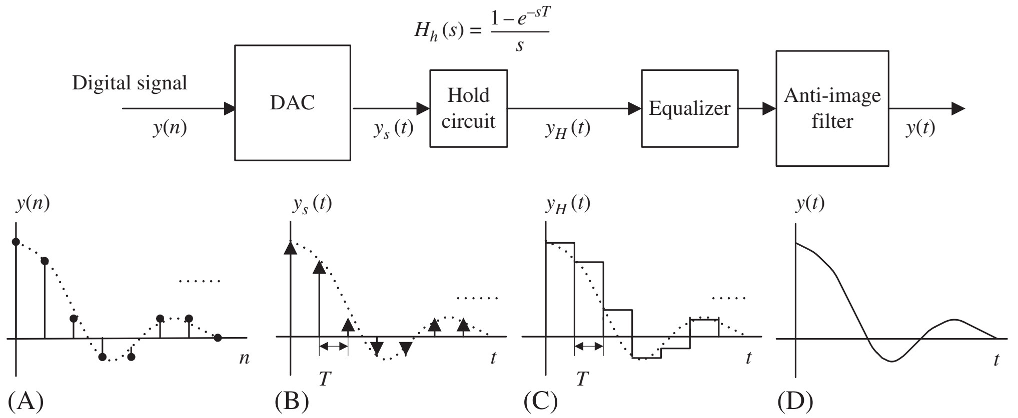

FIG. 2.20

Signal notations at the practical reconstruction stage: (A) Processed digital signal, (B) Recovered ideal sampled signal, (C) Recovered sample-and-hold voltage, and (D) Recovered analog signal.

The magnitude and phase responses are given by

$$|H_h(\omega)| = T\left|\frac{\sin(\omega T/2)}{\omega T/2}\right| = T\left|\frac{\sin(x)}{x}\right|, \tag{2.25}$$

$$\angle H_h(\omega) = -\omega T/2, \tag{2.26}$$

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

$$|H_h(f)| = T\left|\frac{\sin(\pi f T)}{\pi f T}\right|, \tag{2.27}$$

$$\angle H_h(f) = -\pi f T. \tag{2.28}$$

The plot of the magnitude effect is shown in Fig. 2.21.
Fig. 2.22A describes the recovery stage.

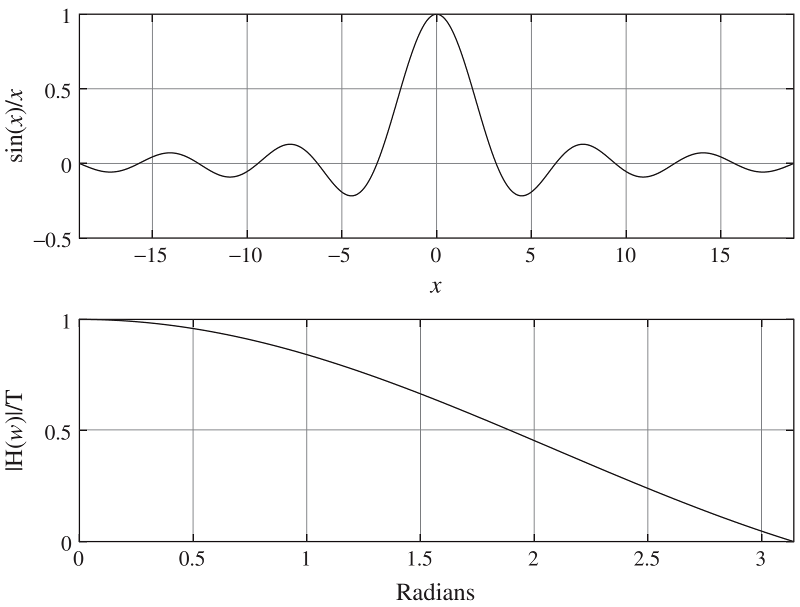

FIG. 2.21

Sample-and-hold lowpass filtering effect.

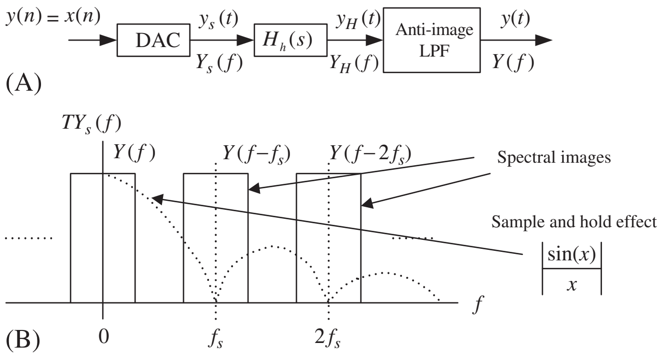

FIG. 2.22

(A) Signal recovery stage and (B) Sample-and-hold effect and distortion.

Assuming that there is no digital processing, that is, $y_s(t) = x_s(t)$, then

$$Y_s(f) = X_s(f) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X(f - kf_s). \tag{2.29}$$

As shown in Fig. 2.22A, $y_H(t)$ is the sample-and-hold signal in time domain while $Y_H(f)$ is the spectrum of the sampled and hold signal, which is given by

$$Y_H(f) = H_h(f)Y_s(f) = e^{-j\pi fT}T\frac{\sin(\pi fT)}{\pi fT}Y_s(f). \tag{2.30}$$

The magnitude frequency response of the sampled and hold signal becomes

$$|Y_H(f)| = T\left|\frac{\sin(\pi fT)}{\pi fT}\right||Y_s(f)|. \tag{2.31}$$

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 Fig. 2.22B. On the other hand, the spectral images are attenuated due to the lowpass effect of $\sin(x)/x$. This sample-and-hold effect can help us design the anti-image filter.

Since the magnitude frequency response of the sampled signal using an ideal sampler is $T|Y_s(f)|$, therefore, the spectral distortion at the recovery stage can be derived as

$$\text{Distortion} = \frac{T|Y_s(f)| - |Y_H(f)|}{T|Y_s(f)|} = 1 - \frac{|Y_H(f)|}{T|Y_s(f)|} = 1 - \left|\frac{\sin(\pi fT)}{\pi fT}\right|. \tag{2.32}$$

The percentage of distortion in the desired frequency band is given by

$$\text{Distortion\%} = \left(1 - \left|\frac{\sin(\pi fT)}{\pi fT}\right|\right) \times 100\%. \tag{2.33}$$

Let us look at Example 2.7

EXAMPLE 2.7

Given a DSP system with a sampling rate of 8000 Hz and a hold circuit used after DAC,
(a) Determine the percentage of distortion at the frequency of 3400 Hz,
(b) Determine the percentage of distortion at the frequency of 1000 Hz.

Solution:

(a) Since $fT = 3400 \times 1/8000 = 0.425$,

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

(b) Since $fT = 1000 \times 1/8000 = 0.125$,

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

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 Fig. 2.23.

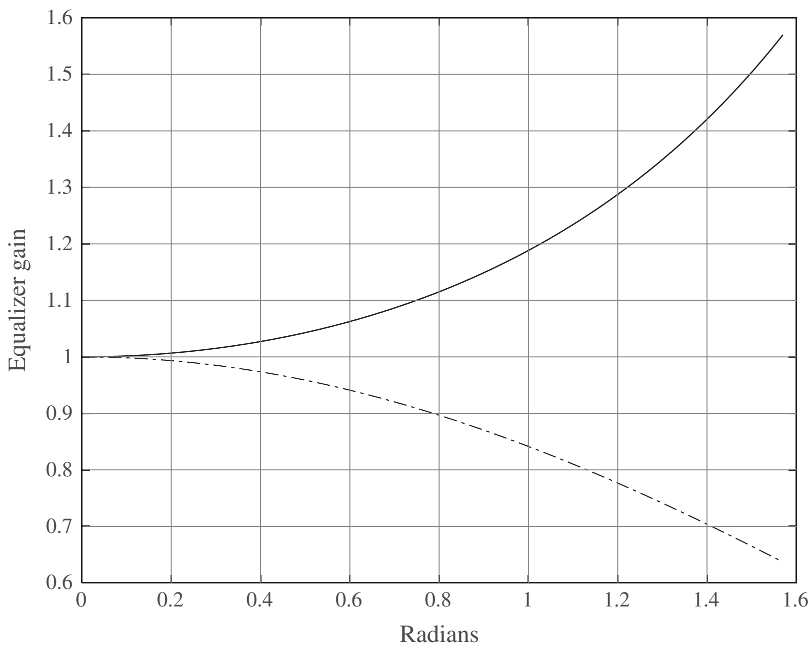

FIG. 2.23

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

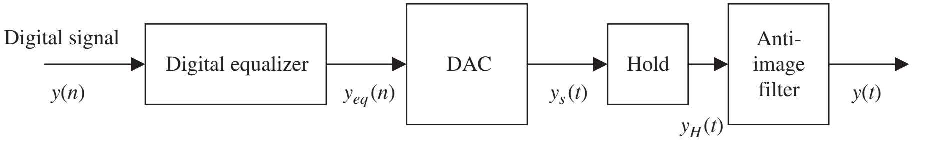

FIG. 2.24

Possible implementation using the digital equalizer.

(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 11.

(3) We can change the DAC configuration and perform digital pre-equalization using the flexible digital filter whose magnitude frequency response is against the spectral shape effect due to the hold circuit. Fig. 2.24 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 illustrates the design of an anti-image filter using a higher sampling rate while making use of the sample-and-hold effect.

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 Fig. 2.25.

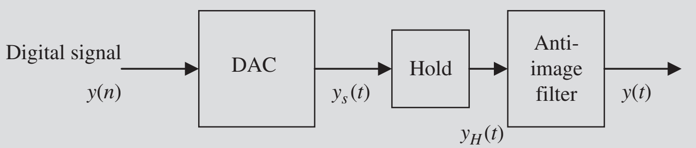

FIG. 2.25

DSP recovery system for Example 2.8.

Design requirements:

- Maximum allowable gain variation from 0 to 3000 Hz = 2 dB
- 33 dB rejection at the frequency of 13,000 Hz
- Butterworth filter is assumed for the anti-image filter

Solution:

We first determine the spectral shaping effects at $f=3000$ Hz, and $f=13,000$ Hz; that is,

$$f = 3000\,\text{Hz}, fT = 3000 \times 1/16,000 = 0.1875,$$

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

and

Continued

EXAMPLE 2.8—CONT'D

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

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

This gain would help the attenuation requirement as shown in Fig. 2.26.

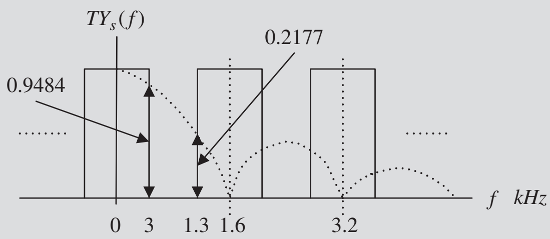

FIG. 2.26

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

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

- Butterworth lowpass filter
- Maximum allowable gain variation from 0 to 3000 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

$$20\log\left(1 + (3000/f_c)^{2n}\right)^{1/2} \le 1.54$$

$$20\log\left(1 + (13{,}000/f_c)^{2n}\right)^{1/2} \ge 20.$$

From these two equations, we have to satisfy

$$(3000/f_c)^{2n} = 10^{0.154} - 1$$

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

Taking the ratio of these two equations yields

$$\left(\frac{13{,}000}{3000}\right)^{2n} = \frac{10^2 - 1}{10^{0.154} - 1}.$$

Then

$$n = \frac{1}{2}\log\left(\left(10^2 - 1\right)/\left(10^{0.154} - 1\right)\right)/\log(13{,}000/3000) = 1.86 \approx 2.$$

Finally, the cutoff frequency can be computed as

$$f_c = \frac{13{,}000}{\left(10^2 - 1\right)^{1/(2n)}} = \frac{13{,}000}{\left(10^2 - 1\right)^{1/4}} = 4121.30\,\text{Hz}$$

$$f_c = \frac{3000}{\left(10^{0.154} - 1\right)^{1/(2n)}} = \frac{3000}{\left(10^{0.154} - 1\right)^{1/4}} = 3714.23\,\text{Hz}.$$

We choose the smaller one, that is,

$$f_c = 3714.23\,\text{Hz}.$$

With the filter order and cutoff frequency, we can realize the anti-image (reconstruction) filter using a second-order unit-gain Sallen-Key lowpass filter described in Fig. 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 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 a digital data with finite precision, which is called the *quantization*. Fig. 2.27 shows that quantization is a part of ADC.

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

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

In this chapter, we will focus on a simple 2-bit flash ADC unit, described in Fig. 2.28, for illustrative purpose. Sigma-delta ADC is studied in Chapter 11.

As shown in Fig. 2.28, 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.

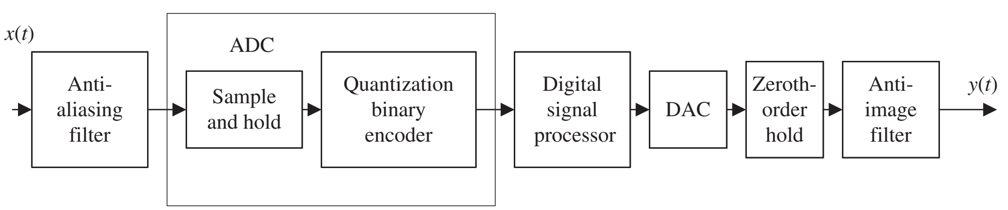

FIG. 2.27

A block diagram for a DSP system.

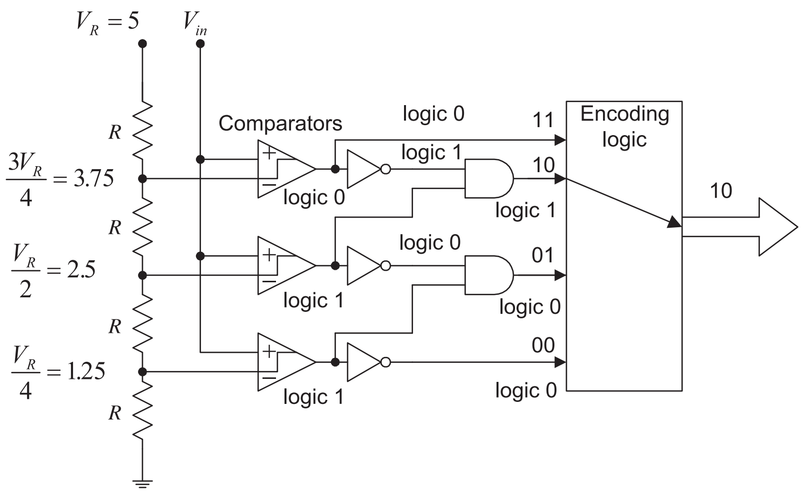

FIG. 2.28

An example of a 2-bit flash ADC.

As an example, the reference voltages in the figure are 1.25, 2.5, 3.75, and 5 V, respectively. If an analog sample-and-hold voltage is $V_{\text{in}} = 3$ V, then the lower two comparators will have output logic 1 of each. Through the logic units, only the line labeled 10 is actively high, and the rest of lines are active 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. Fig. 2.29 illustrates a simple 2-bit DAC unit using an R-2R ladder. The DAC contains the R-2R

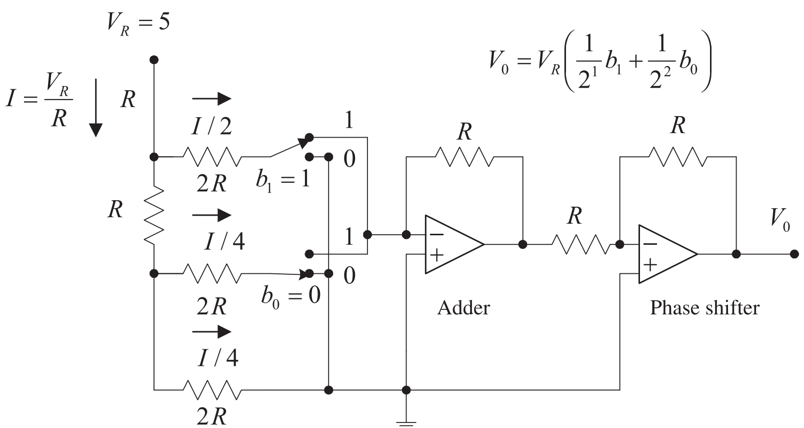

FIG. 2.29

R-2R ladder DAC.

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 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 V, for 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).

As an example shown in Fig. 2.29, where we set $V_R = 5$ and $b_1b_0 = 10$, the ADC output is expected to be

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

As we can see, the recovered voltage of $V_0 = 2.5$ V introduces voltage error as compared with $V_{in} = 3$ V, as 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. The voltage error is called *quantization error*, obtained by subtracting the original analog voltage from the recovered analog voltage. For example, we have the quantization error as

$$V_0 - V_{in} = 2.5 - 3 = -0.5\,\text{V}.$$

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 V to a positive reference voltage, and a *bipolar quantizer* has an analog signal range from a negative reference to a positive reference. The notations and general rules for quantization are:

$$\Delta = \frac{(x_{max} - x_{min})}{L} \tag{2.34}$$

$$L = 2^m \tag{2.35}$$

$$i = \text{round}\left(\frac{x - x_{min}}{\Delta}\right) \tag{2.36}$$

$$x_q = x_{min} + i\Delta \quad i = 0, 1, \ldots, L-1, \tag{2.37}$$

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 Eq. (2.35), where m is the number of bits used in ADC. The symbol Δ is the step size of the quantizer or the ADC

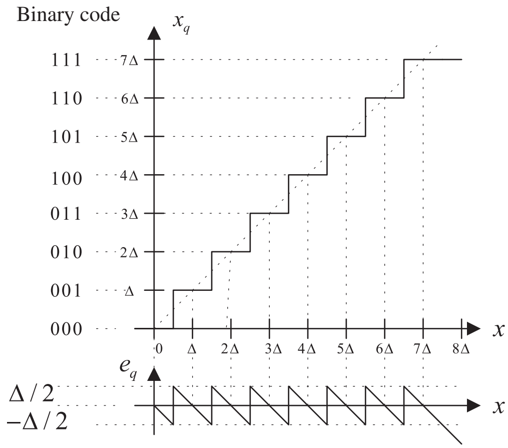

FIG. 2.30

Characteristics for a unipolar quantizer.

resolution. Finally, x_q indicates the quantization level, and i is an index corresponding to the binary code.

Fig. 2.30 depicts a 3-bit unipolar quantizer and corresponding binary codes. From Fig. 2.30, we see that $x_{\min}=0$, $x_{\max}=8\Delta$, and $m=3$. Applying Eq. (2.37) gives each quantization level as follows: $x_q=0+i\Delta$, $i=0, 1, \ldots, L-1$, 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 Fig. 2.31, where we have

$$x_{\min}=-4\Delta, x_{\max}=4\Delta, \text{and } m=3.$$

The corresponding quantization table is given in Table 2.2.

Table 2.1 Quantization Table for a 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	$6.5\Delta\le x<7.5\Delta$

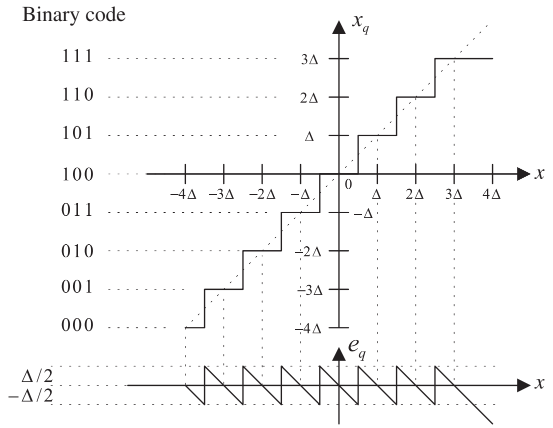

FIG. 2.31

Characteristics for a bipolar quantizer.

Table 2.2 Quantization Table for a 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	$2.5\Delta \le x < 3.5\Delta$

EXAMPLE 2.9

Assuming that a 3-bit ADC channel accepts analog input ranging from 0 to 5 V, determine the following:

(a) Number of quantization levels
(b) Step size of quantizer or resolution
(c) Quantization level when the analog voltage is 3.2 V
(d) Binary code produced by the ADC.

Solution:

Since the range is from 0 to 5 V and the 3-bit ADC is used, we have

$$x_{min} = 0 \text{ V}, x_{max} = 5 \text{ V}, \text{and } m = 3 \text{ bits}.$$

Continued

EXAMPLE 2.9—CONT'D

(a) Using Eq. (2.35), we get the number of quantization levels as

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

(b) Applying Eq. (2.34) yields

$$\Delta = \frac{5-0}{8} = 0.625\,\text{V}.$$

(c) When $x = 3.2\frac{\Delta}{0.625} = 5.12\Delta$, from Eq. (2.36) we get

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

(d) From Eq. (2.37), we determine the quantization level as

$$x_q = 0 + 5\Delta = 5 \times 0.625 = 3.125\,\text{V}.$$

The binary code is determined as 101, either from Fig. 2.30 or Table 2.1.

After quantizing the input signal x, the ADC produces binary codes, as illustrated in Fig. 2.32.

The DAC process is shown in Fig. 2.33. As shown in the figure, the DAC unit takes the binary codes from the digital signal 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 yielded.

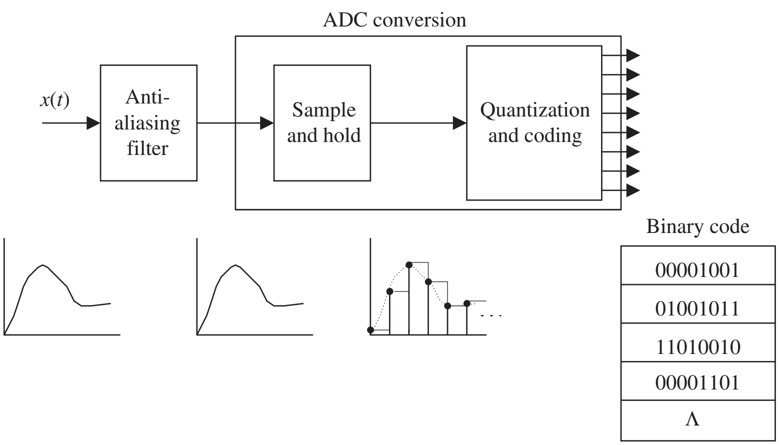

FIG. 2.32

Typical ADC process.

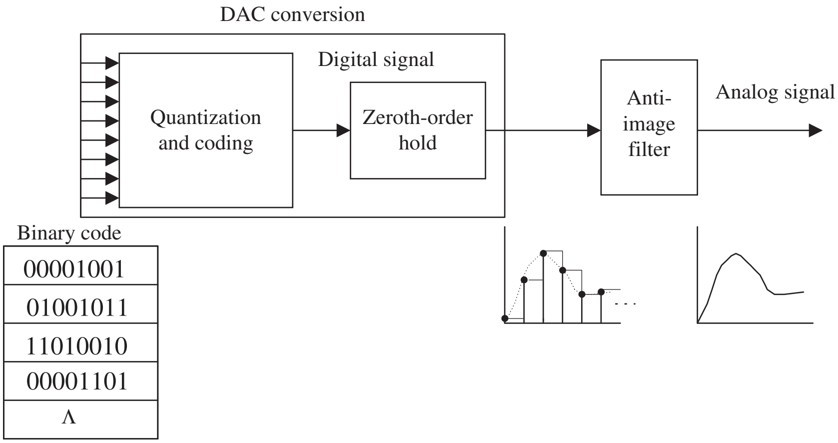

FIG. 2.33

Typical DAC process.

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

$$e_q = x_q - x. \tag{2.38}$$

The quantization error as shown in Fig. 2.30 is bounded by half of the step size, that is,

$$-\frac{\Delta}{2} < e_q \le \frac{\Delta}{2}, \tag{2.39}$$

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 V.

Solution:

Using Eq. (2.38), we obtain

$$e_q = x_q - x = 3.125 - 3.2 = -0.075\,\text{V}.$$

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

$$|e_q| = 0.075 < \frac{\Delta}{2} = 0.3125\,\text{V}.$$

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 there are a sufficiently large number of samples. Assuming that e_q is a uniformly distributed random variable which has a range within a quantized interval Δ and has the following probability density function, $f(e_q)$, as shown in Fig. 2.34.

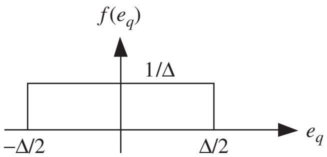

FIG. 2.34

Quantized noise distribution.

The quantized noise power can be derived as

$$E\left\{e_q^2\right\} = \int_{-\Delta/2}^{\Delta/2} e_q^2 f\left(e_q\right) de_q = \int_{-\Delta/2}^{\Delta/2} e_q^2 \frac{1}{\Delta} de_q = \left.\frac{e_q^3}{3\Delta}\right|_{-\Delta/2}^{\Delta/2} = \frac{\Delta^2}{12}. \tag{2.40}$$

Hence, the power of quantization noise is related to the quantization step and given by

$$E\left(e_q^2\right) = \frac{\Delta^2}{12}, \tag{2.41}$$

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) due to quantization can be expressed as

$$SNR = \frac{E(x^2)}{E\left(e_q^2\right)}. \tag{2.42}$$

If we express the SNR in terms of decibels (dB), it follows that

$$SNR_{dB} = 10 \times \log_{10}(SNR)\,\text{dB}. \tag{2.43}$$

Substituting Eq. (2.41) and $E(x^2) = x^2_{rms}$ into Eq. (2.43), we achieve

$$SNR_{dB} = 10.79 + 20 \times \log_{10}\left(\frac{x_{rms}}{\Delta}\right), \tag{2.44}$$

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:

$$SNR = \frac{\frac{1}{N}\sum_{n=0}^{N-1} x^2(n)}{\frac{1}{N}\sum_{n=0}^{N-1} e_q^2(n)} = \frac{\sum_{n=0}^{N-1} x^2(n)}{\sum_{n=0}^{N-1} e_q^2(n)}, \tag{2.45}$$

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,

$$x(t) = A\sin(2\pi \times 1000t),$$

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 Eq. (2.44) leads to

$$SNR_{dB} = 10.79 + 20 \times \log_{10}\left(\frac{0.707A}{2A/2^m}\right)$$
$$= 10.79 + 20 \times \log_{10}(0.707/2) + 20m \times \log_{10}2.$$

After simplifying the numerical values, we get

$$SNR_{dB} = 1.76 + 6.02m\,\text{dB}. \tag{2.46}$$

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 Eq. (2.44) achieves

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

Thus, after numerical simplification, we have

$$SNR_{dB} = 4.77 + 20 \times \log_{10}\left(\frac{x_{rms}}{|x|_{\max}}\right) + 6.02m. \tag{2.47}$$

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,

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

sampled at 8000 Hz,

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

(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
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 stair case 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)
```

Fig. 2.35 shows plots of the quantized signal and the original signal.

(b) Theoretically, applying Eq. (2.46) leads to

$$SNR_{dB} = 1.76 + 6.02 \times 4 = 25.84\,\text{dB}.$$

Practically, using Eq. (2.45), the simulated result is obtained as

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

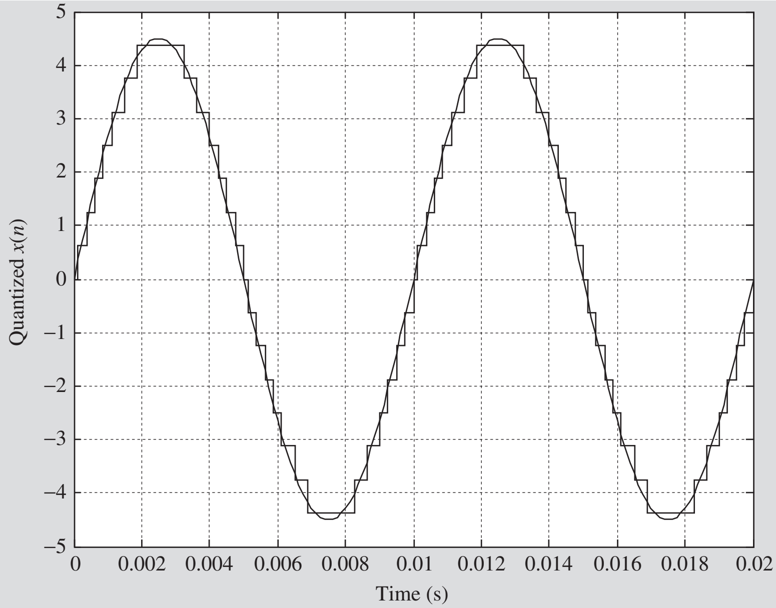

FIG. 2.35

Comparison of the quantized signal and the original signal.

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 the quantizing a speech signal.

EXAMPLE 2.14

Given a speech signal sampled at 8000 Hz given in the file *we.dat*,

(a) Write a MATLAB program to quantize the $x(t)$ using 4-bits quantizers to obtain the quantized signal x_q, assuming the signal range is from −5 to 5 V.

(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
```

Continued

EXAMPLE 2.14—CONT'D

```
t=[0:1:lg-1]*T;                                   % Time instants in second
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
```

(b) In Fig. 2.36, 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 Eq. (2.47), it follows that

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

$$= 4.77 + 20\log_{10}(0.203) + 6.02 \times 4 = 15\text{dB}.$$

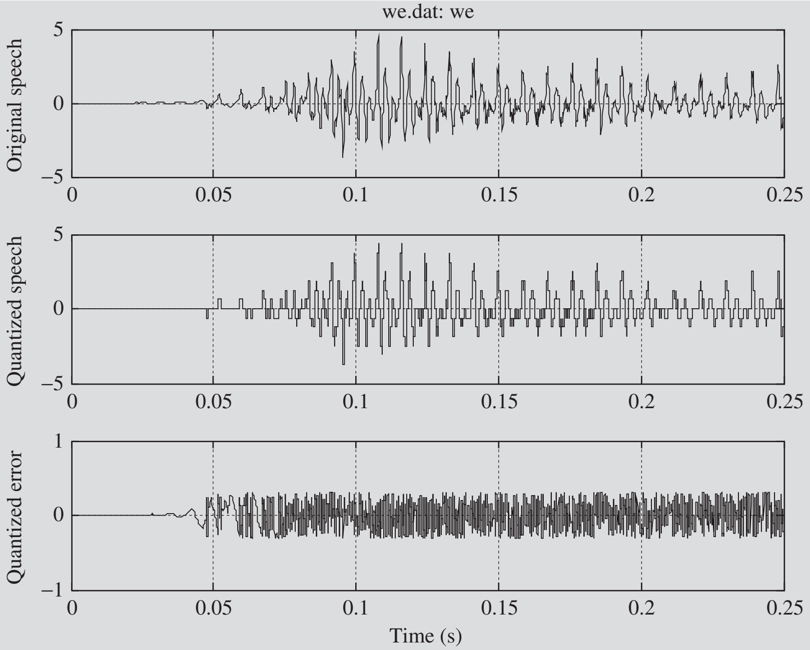

FIG. 2.36

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

On the other hand, the simulated result using Eq. (2.45) gives

$$SNR_{dB} = 15.01\,\text{dB}.$$

Results for SNRs from Eqs. (2.47) and (2.45) 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 the sampling period is the sampling rate. The half of sampling rate is the folding frequency (Nyquist limit).
3. The sampling theorem condition that the sampling rate be larger than twice of the highest frequency of the analog signal to be sampled, must be met in order to have the analog signal 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 sampled-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 means 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 the quantization levels (number of the 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 1 bit used in each ADC code will improve 6-dB SNR 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 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 recover 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 is created for calculation of 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

$$x(t) = 5\cos(2\pi \times 1500t), \text{ for } t \geq 0,$$

sampled at a rate of 8000 Hz,

(a) sketch the spectrum of the original signal;
(b) sketch the spectrum of the sampled signal from 0 up to 20 kHz.

2.2. Given an analog signal

$$x(t) = 5\cos(2\pi \times 2500t) + 2\cos(2\pi \times 3200t), \text{ for } t \geq 0,$$

sampled at a rate of 8000 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

$$x(t) = 3\cos(2\pi \times 1500t) + 2\cos(2\pi \times 2200t), \text{ for } t \geq 0,$$

sampled at a rate of 8000 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

$$x(t) = 3\cos(2\pi \times 1500t) + 2\cos(2\pi \times 4200t), \text{ for } t \geq 0,$$

sampled at a rate of 8000 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

$$x(t) = 5\cos(2\pi \times 2500t) + 2\cos(2\pi \times 4500t), \text{ for } t \geq 0,$$

sampled at a rate of 8000 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

$$x(t) = 10\cos(2\pi \times 5500t) + 5\sin(2\pi \times 7500t), \text{ for } t \geq 0,$$

sampled at a rate of 8000 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

$$x(t) = 8\cos(2\pi \times 5000t) + 5\sin(2\pi \times 7000t), \text{ for } t \geq 0,$$

sampled at a rate of 8000 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

$$x(t) = 10\cos(2\pi \times 5000t) + 5\sin(2\pi \times 7500t), \text{ for } t \geq 0,$$

sampled at a rate of 8000 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 the following second-order anti-aliasing lowpass filter (Fig. 2.37) which is a Butterworth type, determine the values of circuit elements if we want the filter to have a cutoff frequency of 1000 Hz.
2.10. From Problem 2.9, determine the percentage of aliasing level at the frequency of 500 Hz, assuming that the sampling rate is 4000 Hz.
2.11. Given the following second-order anti-aliasing lowpass filter (Fig. 2.38) which is a Butterworth type, 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 4000 Hz.
2.13. Given a DSP system in which a sampling rate of 8000 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 1000 Hz.

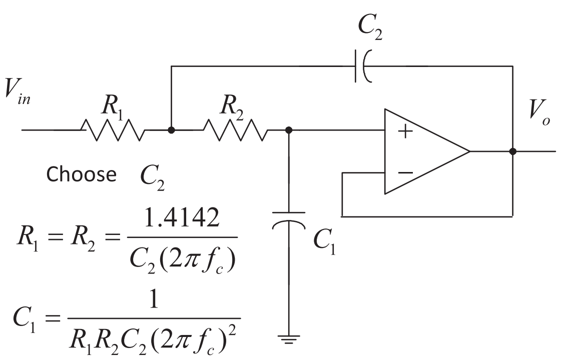

FIG. 2.37

Filter Circuit in Problem 2.9.

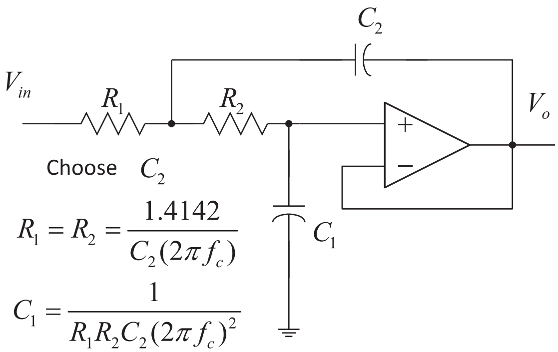

FIG. 2.38

Filter circuit in Problem 2.11.

2.14. Given a DSP system in which a sampling rate of 8000 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 for the percentage of aliasing level at the cutoff frequency required to be less than 10%.

2.15. Given a DSP system in which a sampling rate of 8000 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 the frequency of 900 Hz.

2.16. Given a DSP system in which a sampling rate of 8000 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 for the percentage of aliasing level at the cutoff frequency required to be less than 10%.

2.17. Given a DSP system (Fig. 2.39) with a sampling rate of 8000 Hz and assuming that the hold circuit is used after DAC, determine

(a) the percentage of distortion at the frequency of 3200 Hz;
(b) the percentage of distortion at the frequency of 1500 Hz.

2.18. A DSP system is given with the following specifications:

Design requirements:

Sampling rate 20,000 Hz;
Maximum allowable gain variation from 0 to 4000 Hz = 2 dB;

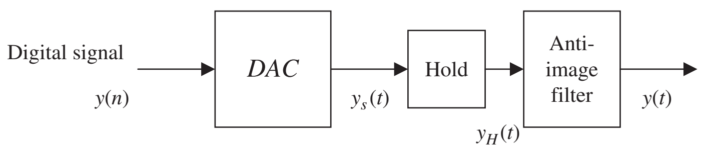

FIG. 2.39

Analog signal reconstruction in Problem 2.18.

40 dB rejection at the frequency of 16,000 Hz; and
Butterworth filter assumed.

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

2.19. Given a DSP system with a sampling rate of 8000 Hz and assuming that the hold circuit is used after DAC, determine

(a) the percentage of distortion at the frequency of 3000 Hz;
(b) the percentage of distortion at the frequency of 1600 Hz.

2.20. A DSP system (Fig. 2.40) is given with the following specifications:
Design requirements:

Sampling rate 22,000 Hz;
Maximum allowable gain variation from 0 to 4000 Hz = 2 dB;
40 dB rejection at the frequency of 18,000 Hz; and
Butterworth filter assumed

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 V shown in Fig. 2.41, determine the output bits.

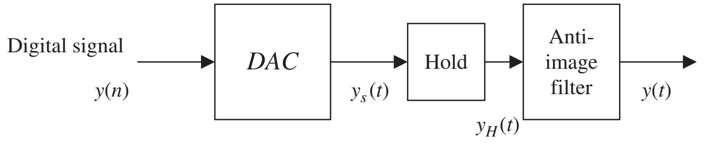

FIG. 2.40

Analog signal reconstruction in Problem 2.20.

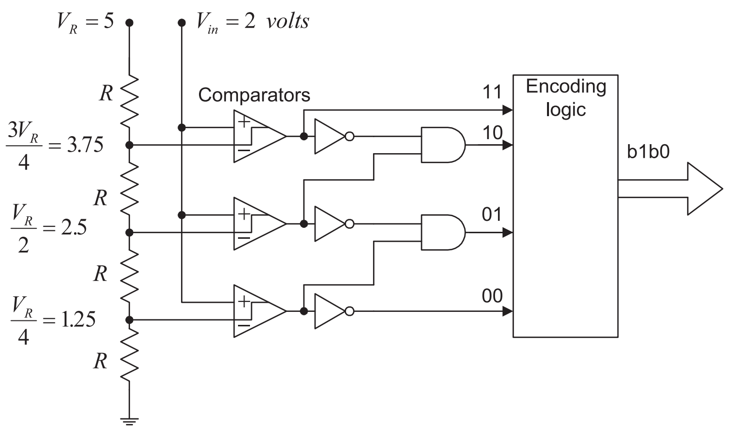

FIG. 2.41

A 2-Bit flash ADC in Problem 2.21.

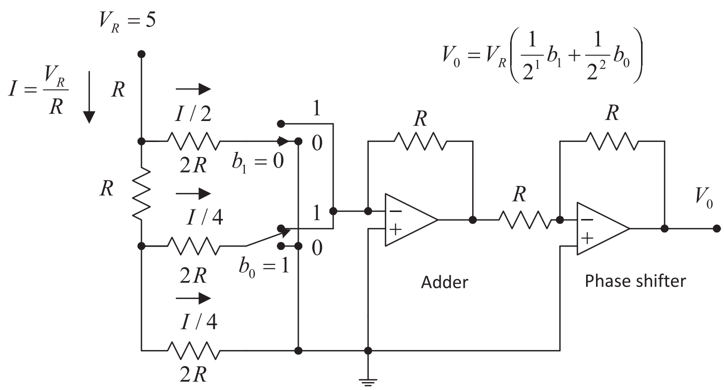

FIG. 2.42

A 2-Bit R-2R DAC in Problem 2.22.

2.22. Given the R-2R DAC unit with a 2-bit value as $b_1b_0 = 01$ shown in Fig. 2.42, determine the converted voltage.

2.23. Given the 2-bit flash ADC unit with an analog sample-and-hold voltage of 3.5 V shown in Fig. 2.41, determine the output bits.

2.24. Given the R-2R DAC unit with 2-bit values as $b_1b_0 = 11$ and $b_1b_0 = 10$, respectively, and shown in Fig. 2.42, determine the converted voltages.

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

(a) Number of quantization levels;
(b) Step size of quantizer or resolution;
(c) Quantization level when the analog voltage is 3.2 V;
(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 V, determine the following:

(a) Number of quantization levels;
(b) Step size of quantizer or resolution;
(c) Quantization level when the analog voltage is 1.2 V;
(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 V, determine the following:

(a) Number of quantization levels;
(b) Step size of quantizer or resolution;

(c) Quantization level when the analog voltage is $-1.2\,\text{V}$;
(d) Binary code produced by the ADC;
(e) Quantization error.

2.28. Assuming that an 8-bit ADC channel accepts analog input ranging from -2.5 to $2.5\,\text{V}$, determine the following:

(a) Number of quantization levels;
(b) Step size of quantizer or resolution;
(c) Quantization level when the analog voltage is $1.5\,\text{V}$;
(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, $x(t) = 9.5\sin(2000 \times \pi t)$, and if a bipolar quantizer uses 6 bits, determine

(a) Number of quantization levels;
(b) Quantization step size or resolution, Δ, assuming the signal range is from -10 to $10\,\text{V}$;
(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) Number of quantization levels;
(b) Quantization step size or resolution, Δ, if the signal range is $5\,\text{V}$;
(c) The signal power-to-quantization noise power ratio.

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,

$$x(t) = 4.5\sin(2\pi \times 100t), \text{ for } t \ge 0,$$

sample it at 8000 samples per second and

(a) Write a MATLAB program to quantize $x(t)$ using a 6-bits bipolar quantizer to obtain the quantized signal x_q, assuming that the signal range to be from -5 to $5\,\text{V}$;
(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,

$$x(t) = 3.25\sin(2\pi \times 50t) + 1.25\cos(2\pi \times 100t + \pi/4), \text{ for } t \ge 0,$$

sample it at 8000 samples per second and

(a) Write a MATLAB program to quantize $x(t)$ using a 6-bits bipolar quantizer to obtain the quantized signal x_q, assuming that the signal range to be from -5 to $5\,\text{V}$;
(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 8000 Hz, as shown in Example 2.14,

(a) Write a MATLAB program to quantize $x(t)$ using a 6-bits bipolar quantizer to obtain the quantized signal x_q, assuming that the signal range is from -5 to 5 V;
(b) Plot the original speech waveform, quantized speech, and quantization error;
(c) Calculate the SNR due to quantization using the MATLAB program.

MATLAB Projects

2.34. Performance evaluation of speech quantization:

Given an original speech segment "speech.dat" sampled at 8000 Hz with each sample encoded in 16 bits, use Programs 2.3–2.5 and modify Program 2.2 to quantize the speech segment using 3–15 bits, respectively. The SNR in dB must be measured for each quantization. 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 8000 Hz. In this project, create a plot of the measured SNR (dB) versus the number of bits and discuss the effect of the sound quality. For comparisons, plot the original speech and the quantized one using 3, 8, 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 USGS Albuquerque Seismological Laboratory) has a sampling rate of 15 Hz with 6700 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 reduced signal quality. Use Programs 2.3–2.5 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, 18, 25, and 31 bits.

Advanced Problems

2.36–2.38. Given the following sampling system (see Fig. 2.43), $x_s(t) = x(t)p(t)$

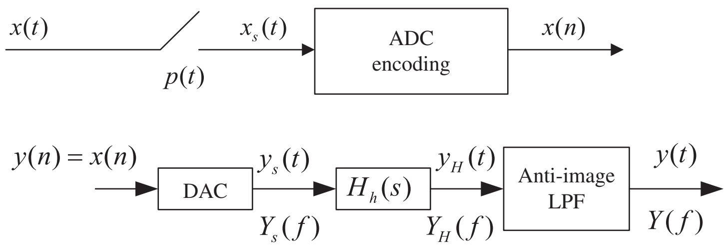

FIG. 2.43

Analog signal reconstruction in Problems 2.36–2.38.

2.36. If the pulse train used is depicted in Fig. 2.44

(a) Determine the Fourier series expansion for $p(t) = \sum_{k=-\infty}^{\infty} a_k e^{jkw_0 t}$;
(b) Determine $X_s(f)$ in terms of $X_s(f)$ using Fourier transform, that is,

$$X_s(f) = FT\{x_s(t)\} = FT\{x(t)p(t)\};$$

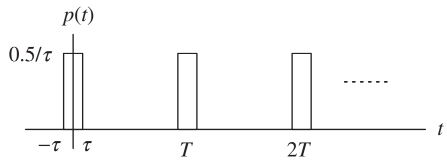

FIG. 2.44

Pulse train in Problem 2.36.

(c) Determine spectral distortion referring to $X_s(f)$ for $-f_s/2<f<f_s/2$.

2.37. If the pulse train used is depicted in Fig. 2.45

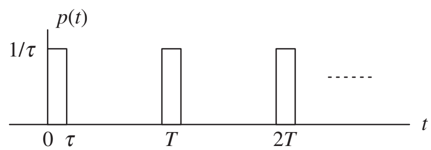

FIG. 2.45

Pulse train in Problem 2.37.

(a) Determine the Fourier series expansion for $p(t)=\sum_{k=-\infty}^{\infty} a_k e^{jkw_0 t}$;
(b) Determine $X_s(f)$ in terms of $X(f)$ using Fourier transform, that is,
$X_s(f)=FT\{x_s(t)\}=FT\{x(t)p(t)\}$;
(c) Determine spectral distortion referring to $X(f)$ for $-f/2<f<f_s/2$.

2.38. If the pulse train used is depicted in Fig. 2.46

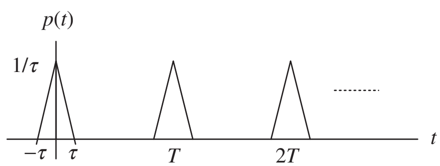

FIG. 2.46

Pulse train in Problem 2.38.

(a) Determine the Fourier series expansion for $p(t)=\sum_{k=-\infty}^{\infty} a_k e^{jkw_0 t}$;
(b) Determine $X_s(f)$ in terms of $X(f)$ using Fourier transform, that
$X_s(f)=FT\{x_s(t)\}=FT\{x(t)p(t)\}$;
(c) Determine spectral distortion referring to $X(f)$ for $-f_s/2<f<f_s/2$.

2.39. In Fig. 2.16, a Chebyshev lowpass filter is chosen to serve as anti-aliasing lowpass filter, where the magnitude frequency response of the Chebyshev filter with an order n is given by

$$|H(f)|=\frac{1}{\sqrt{1+\varepsilon^2 C_n^2(f/f_c)}},$$

where ε is the absolute ripple, and

$$C_n(x) = \begin{cases} \cos\left[n\cos^{-1}(x)\right] & x < 1 \\ \cosh\left[n\cosh^{-1}(x)\right] & x > 1 \end{cases} \text{ and } \cosh^{-1}(x) = \ln\left(x + \sqrt{x^2 - 1}\right).$$

(a) Derive the formula for the aliasing level; and

(b) When the sampling frequency is 8 kHz, a cutoff frequency is 3.4 kHz, ripple is 1 dB, and order equals $n=4$, determine the aliasing level at frequency of 1 kHz.

2.40. Given the following signal

$$x(t) = \sum_{i=1}^{N} A_i \cos(\omega_i t + \phi_i)$$

with the signals ranging from $-\sum_{i=1}^{N} A_i$ to $\sum_{i=1}^{N} A_i$, determine the signal to quantization noise power ratio using m bits.

2.41. Given the following modulated signal

$$x(t) = A_1 \cos(\omega_t t + \phi_1) \times A_2 \cos(\omega_2 t + \phi_2)$$

with the signal ranging from $-A_1A_2$ to A_1A_2, determine the signal to quantization noise power ratio using m bits.

2.42. Assume that truncation of the continuous signal $x(n)$ in Problem 2.40 is defined as:

$$x_q(n) = x(n) + e_q(n),$$

where $-\Delta < e_q(n) \leq 0$ and $\Delta = 2\sum_{i=1}^{N} A_i/2^m$. The quantized noise has the distribution given in Fig. 2.47, determine the SNR using m bits.

2.43. Assume that truncation of the continuous signal $x(n)$ in Problem 2.41 is defined as

$$x_q(n) = x(n) + e_q(n),$$

where $0 \leq e_q(n) < \Delta$ and $\Delta = 2A_1A_2/2^m$. The quantized noise has the distribution given in Fig. 2.48, determine the SNR using m bits.

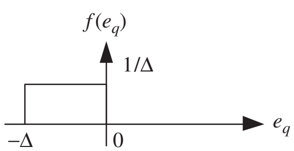

FIG. 2.47

The truncated error distribution in Problem 2.42.

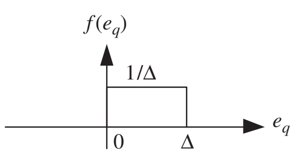

FIG. 2.48

The truncated error distribution in Problem 2.43.