CHAPTER

INFINITE IMPULSE RESPONSE FILTER DESIGN

8

CHAPTER OUTLINE

Digital Signal Processing. https://doi.org/10.1016/B978-0-12-815071-9.00008-7

8.1 INFINITE IMPULSE RESPONSE FILTER FORMAT

In this chapter, we will study several methods for infinite impulse response (IIR) filter design. An IIR filter is described using the difference equation, as discussed in Chapter 6:

$$y(n) = b_0 x(n) + b_1 x(n-1) + \cdots + b_M x(n-M) \\ - a_1 y(n-1) - \cdots - a_N y(n-N).$$

Chapter 6 also gives the IIR filter transfer function as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}},$$

where b_i and a_i are the $(M+1)$ numerator and N denominator coefficients, respectively. $Y(z)$ and $X(z)$ are the z-transform functions of the input $x(n)$ and output $y(n)$. To become familiar with the form of the IIR filter, let us look at the following example.

EXAMPLE 8.1

Given the following IIR filter:

$$y(n) = 0.2x(n) + 0.4x(n-1) + 0.5y(n-1),$$

determine the transfer function, nonzero coefficients, and impulse response.

Solution:

Applying the z-transform and solving for a ratio of the z-transform output over input, we have

$$H(z) = \frac{Y(z)}{X(z)} = \frac{0.2 + 0.4z^{-1}}{1 - 0.5z^{-1}}.$$

We also identify the nonzero numerator coefficients and denominator coefficient as

$$b_0 = 0.2, b_1 = 0.4, \text{and } a_1 = -0.5.$$

To solve the impulse response, we rewrite the transfer function as

$$H(z) = \frac{0.2}{1 - 0.5z^{-1}} + \frac{0.4z^{-1}}{1 - 0.5z^{-1}}.$$

Using the inverse z-transform and shift theorem, we obtain the impulse response as

$$h(n) = 0.2(0.5)^n u(n) + 0.4(0.5)^{n-1} u(n-1).$$

The obtained impulse response has an infinite number of terms, where the first several terms are calculated as

$$h(0) = 0.2, h(1) = 0.5, h(2) = 0.25, \cdots$$

At this point, we can make the following remarks:

1. The IIR filter output $y(n)$ depends not only on the current input $x(n)$ and past inputs $x(n-1)$, …, but also on the past output(s) $y(n-1)$ …, (recursive terms). Its transfer function is a ratio of the numerator polynomial over the denominator polynomial, and its impulse response has an infinite number of terms.
2. Since the transfer function has the denominator polynomial, the pole(s) of a designed IIR filter must be inside the unit circle on the z-plane to ensure its stability.
3. Comparing with the finite impulse response (FIR) filter (see Chapter 7), the IIR filter offers a much smaller filter size. Hence, the filter operation requires a fewer number of computations, but the linear phase is not easily obtained. The IIR filter is preferred when a small filter size is called for but the application does not require a linear phase.

The objective of IIR filter design is to determine the filter numerator and denominator coefficients to satisfy filter specifications such as passband gain and stopband attenuation, as well as cutoff frequency/frequencies for the lowpass, highpass, bandpass, and bandstop filters.

We will first focus on bilinear transformation (BLT) design method. Then we will introduce other design methods such as the impulse invariant design and the pole-zero placement design.

8.2 BILINEAR TRANSFORMATION DESIGN METHOD

Fig. 8.1 illustrates a flow chart of the BLT design used in this book. The design procedure includes the following steps: (1) transforming digital filter specifications into analog filter specifications, (2) performing analog filter design, and (3) applying BLT (which will be introduced in the next section) and verifying its frequency response.

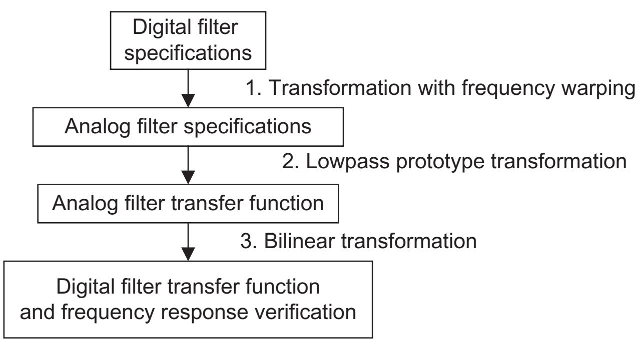

FIG. 8.1

General procedure for IIR filter design using BLT.

8.2.1 ANALOG FILTERS USING LOWPASS PROTOTYPE TRANSFORMATION

Before we begin to develop the BLT design, let us review analog filter design using *lowpass prototype transformation*. This method converts the analog lowpass filter with a cutoff frequency of 1 rad/s, called the lowpass prototype, into practical analog lowpass, highpass, bandpass, and bandstop filters with their frequency specifications.

Letting $H_P(s)$ be a transfer function of the lowpass prototype, the transformation of the lowpass prototype into a lowpass filter is shown in Fig. 8.2.

As shown in Fig. 8.2, $H_{LP}(s)$ designates the analog lowpass filter with a cutoff frequency ω_c rad/s. The lowpass prototype to lowpass filter transformation substitutes s in the lowpass prototype function $H_P(s)$ with s/ω_c, where v is the normalized frequency of the lowpass prototype and ω_c is the cutoff frequency of the lowpass filter to be designed. Let us consider the following first-order lowpass prototype:

$$H_P(s) = \frac{1}{s+1}. \tag{8.1}$$

Its frequency response is obtained by substituting $s = jv$ into Eq. (8.1), that is,

$$H_P(j\nu) = \frac{1}{jv+1}$$

with the magnitude gain given in Eq. (8.2):

$$|H_P(j\nu)| = \frac{1}{\sqrt{1+\nu^2}}. \tag{8.2}$$

We compute the gains at $v=0$, $v=1$, $v=100$, and $v=10,000$ to obtain 1, $1/\sqrt{2}$, 0.0995, and 0.01, respectively. The cutoff frequency gain at $v=1$ equals $1/\sqrt{2}$, which is equivalent to -3 dB, and the direct-current (DC) gain is 1. The gain approaches zero when the frequency tends to $v=+\infty$. This verifies that the lowpass prototype is a normalized lowpass filter with a normalized cutoff frequency of 1. Applying the prototype transformation $s = s/\omega_c$ in Fig. 8.2, we get an analog lowpass filter with a cutoff frequency of ω_c as

$$H(s) = \frac{1}{s/\omega_c+1} = \frac{\omega_c}{s+\omega_c}. \tag{8.3}$$

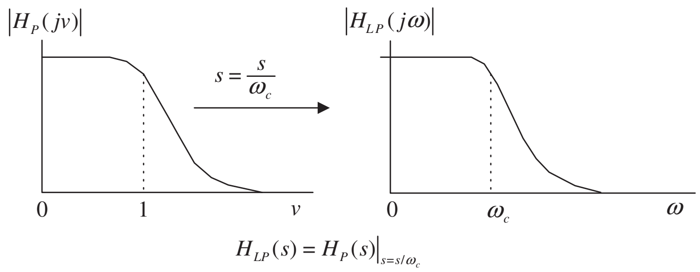

FIG. 8.2

Analog lowpass prototype transformation into a lowpass filter.

We can obtain the analog frequency response by substituting $s = j\omega$ into Eq. (8.3), that is,

$$H(j\omega) = \frac{1}{j\omega/\omega_c + 1}.$$

The magnitude response is determined by

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^2}}. \tag{8.4}$$

Similarly, we verify the gains at $\omega = 0$, $\omega = \omega_c$, $\omega = 100\omega_c$, and $\omega = 10,000\omega_c$ to be 1, $1/\sqrt{2}$, 0.0995, and 0.01, respectively. The filter gain at the cutoff frequency ω_c equals $1/\sqrt{2}$, and the DC gain is 1. The gain approaches zero when $\omega = +\infty$. We note that filter gains do not change but the filter frequency is scaled up by a factor of ω_c. This verifies that the prototype transformation converts the lowpass prototype to the analog lowpass filter with the specified cutoff frequency of ω_c without an effect on the filter gain.

This first-order prototype function is used here for an illustrative purpose. We will obtain general functions for Butterworth and Chebyshev lowpass prototypes in a later section.

The highpass, bandpass, and bandstop filters using the specified lowpass prototype transformation can be easily verified. We review them in Figs. 8.3–8.5, respectively.

The transformation from the lowpass prototype to the highpass filter $H_{HP}(s)$ with a cutoff frequency ω_c rad/s is shown in Fig. 8.3, where $s = \omega_c/s$ in the lowpass prototype transformation.

The transformation of the lowpass prototype function to a bandpass filter with a center frequency ω_0, a lower cutoff frequency ω_l, and an upper cutoff frequency ω_h in the passband is depicted in Fig. 8.4, where $s = (s^2 + \omega_0^2)/(sW)$ is substituted into the lowpass prototype.

As shown in Fig. 8.4, ω_0 is the geometric center frequency, which is defined as $\omega_0 = \sqrt{\omega_l \omega_h}$ while the passband bandwidth is given by $W = \omega_h - \omega_l$. Similarly, the transformation from the lowpass prototype to a bandstop (bandreject) filter is illustrated in Fig. 8.5 with $s = sW/(s^2 + \omega_0^2)$ substituted into the lowpass prototype.

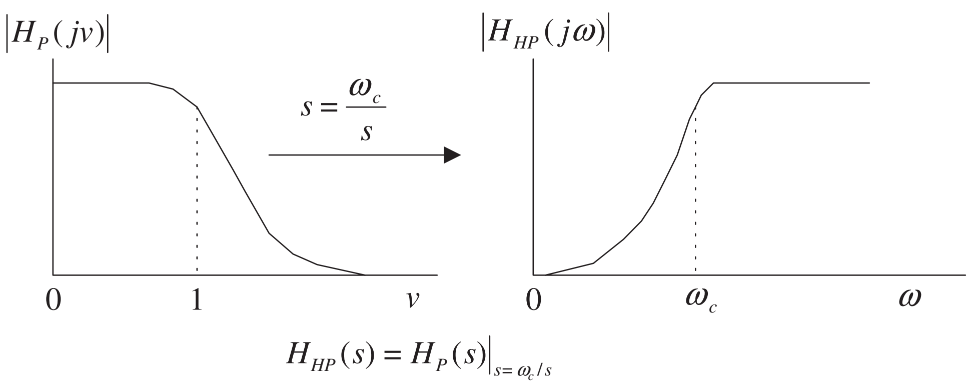

FIG. 8.3

Analog lowpass prototype transformation to the highpass filter.

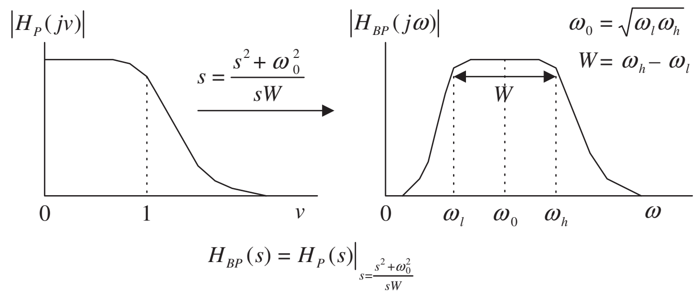

FIG. 8.4

Analog lowpass prototype transformation to the bandpass filter.

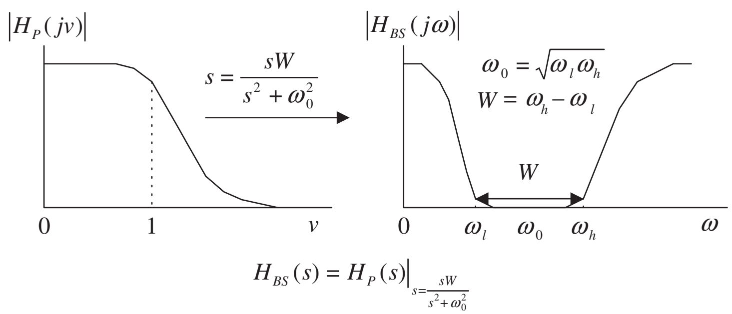

FIG. 8.5

Analog lowpass prototype transformation to a bandstop filter.

Finally, the lowpass prototype transformations are summarized in Table 8.1.

MATLAB function **freqs ()** can be used to plot analog filter frequency responses for verification with the following syntax:

H = freqs(B,A,W)

B = the vector containing the numerator coefficients

A = the vector containing the denominator coefficients

W = the vector containing the specified analog frequency points (rad/s)

H = the vector containing the frequency response.

The following example verifies the lowpass prototype transformation.

Table 8.1 Analog Lowpass Prototype Transformations

Filter Type	Prototype Transformation
Lowpass	$\frac{s}{\omega_c}$, ω_c is the cutoff frequency
Highpass	$\frac{\omega_c}{s}$, ω_c is the cutoff frequency
Bandpass	$\frac{s^2+\omega_0^2}{sW}$, $\omega_0 = \sqrt{\omega_l\omega_h}$, $W = \omega_h - \omega_l$
Bandstop	$\frac{sW}{s^2+\omega_0^2}$, $\omega_0 = \sqrt{\omega_l\omega_h}$, $W = \omega_h - \omega_l$

EXAMPLE 8.2

Given a lowpass prototype

$$H_P(s) = \frac{1}{s+1},$$

Determine each of the following analog filters and plot their magnitude responses from 0 to 200 rad/s.

1. The highpass filter with a cutoff frequency of 40 rad/s.
2. The bandpass filter with a center frequency of 10 rad/ s and bandwidth of 20 rad/s.

Solution:

(1) Applying the lowpass prototype transformation by substituting $s = 40/s$ into the lowpass prototype, we have an analog highpass filter as

$$H_{HP}(s) = \frac{1}{\frac{40}{s}+1} = \frac{s}{s+40}.$$

(2) Similarly, substituting the lowpass to bandpass transformation $s = (s^2+100)/(20s)$ into the lowpass prototype leads to

$$H_{BP}(s) = \frac{1}{\frac{s^2+100}{20s}+1} = \frac{20s}{s^2+20s+100}.$$

The program for plotting the magnitude responses for highpass filter and bandpass filter is shown in Program 8.1, and Fig. 8.6 displays the magnitude responses for the highpass filter and bandpass filter, respectively.

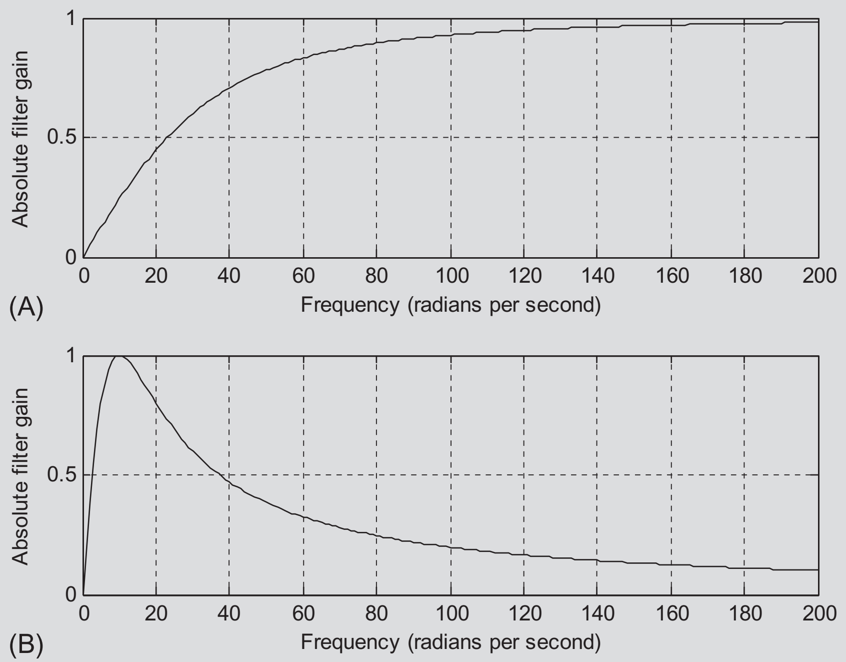

FIG. 8.6

Magnitude responses for the analog highpass filter and bandpass filter in Example 8.2.

Program 8.1. MATLAB program in Example 8.2.

```
W=0:1:200; %Aanalog frequency points for computing the filter gains
Ha=freqs([1 0 ][1 1 4 0],W);                  % Frequency response for the highpass filter
Hb=freqs([2 0 0][1 2 0 1 0 0],W);             % Frequency response for the bandpass filter
subplot(2,1,1);plot(W, abs(Ha),'k');grid % The filter gain plot for highpass filter
xlabel('(a) Frequency (radians per second)')
ylabel('Absolute filter gain');
subplot(2,1,2);plot(W,abs(Hb),'k');grid  % The filter gain plot for bandpass filter
xlabel('(b) Frequency (radians per second)')
ylabel('Absolute filter gain');
```

Fig. 8.6 confirms the transformation of the lowpass prototype into a highpass filter and a bandpass filter, respectively. To obtain the transfer function of an analog filter, we always begin with lowpass prototype and apply the corresponding lowpass prototype transformation. To transfer from a lowpass prototype to a bandpass or bandstop filter, the resultant order of the analog filter is twice the lowpass prototype order.

8.2.2 BILINEAR TRANSFORMATION AND FREQUENCY WARPING

In this section, we develop the BLT, which converts an analog filter to a digital filter. We begin by finding the area under a curve using the integration of calculus and the numerical recursive method in order to determine the BLT. The area under the curve is a common problem in early calculus course. As shown in Fig. 8.7, the area under the curve can be determined using the following integration:

$$y(t) = \int_0^t x(t)dt, \tag{8.5}$$

where $y(t)$ (area under the curve) and $x(t)$ (curve function) are the output and input of the analog integrator, respectively, and t is the upper limit of the integration.

Applying Laplace transform in Eq. (8.5), we have

$$Y(s) = \frac{X(s)}{s} \tag{8.6}$$

and find the Laplace transfer function as

$$G(s) = \frac{Y(s)}{X(s)} = \frac{1}{s}. \tag{8.7}$$

Now we examine the numerical integration method shown in Fig. 8.7 to approximate the integration of Eq. (8.5) using the following difference equation:

$$y(n) = y(n-1) + \frac{x(n) + x(n-1)}{2}T, \tag{8.8}$$

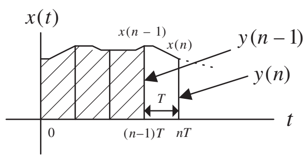

FIG. 8.7

Digital integration method to calculate the area under the curve.

where T denotes the sampling period. $y(n)=y(nT)$ is the output sample that is the whole area under the curve, while $y(n-1)=y(nT-T)$ is the previous output sample from the integrator indicating the previously computed area under the curve (the shaded area in Fig. 8.7). Note that $x(n)=x(nT)$ and $x(n-1)=x(nT-T)$, sample amplitudes from the curve, are the current input sample and the previous input sample in Eq. (8.8). Applying the z-transform on both sides of Eq. (8.8) leads to

$$Y(z)=z^{-1}Y(z)+\frac{T}{2}\left(X(z)+z^{-1}X(z)\right).$$

Solving for the ratio of $Y(z)/X(z)$, we achieve the z-transfer function as

$$H(z)=\frac{Y(z)}{X(z)}=\frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}}. \tag{8.9}$$

Next, on comparing Eq. (8.9) with Eq. (8.7), it follows that

$$\frac{1}{s}=\frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}}=\frac{T}{2}\frac{z+1}{z-1}. \tag{8.10}$$

Solving for s in Eq. (8.10) gives the BLT

$$s=\frac{2}{T}\frac{z-1}{z+1}. \tag{8.11}$$

The BLT method is a mapping or transformation of points on the s-plane to the z-plane. Eq. (8.11) can be alternatively written as

$$z=\frac{1+sT/2}{1-sT/2}. \tag{8.12}$$

The general mapping properties are summarized as follows:

1. The left half of the s-plane is mapped onto the inside area of the unit circle of the z-plane.
2. The right half of the s-plane is mapped onto the outside area of the unit circle of the z-plane.
3. The positive $j\omega$ axis portion in the s-plane is mapped onto the positive half circle (the dashed line arrow in Fig. 8.8) on the unit circle, while the negative $j\omega$ axis is mapped onto the negative half circle (the dotted line arrow in Fig. 8.8) on the unit circle.

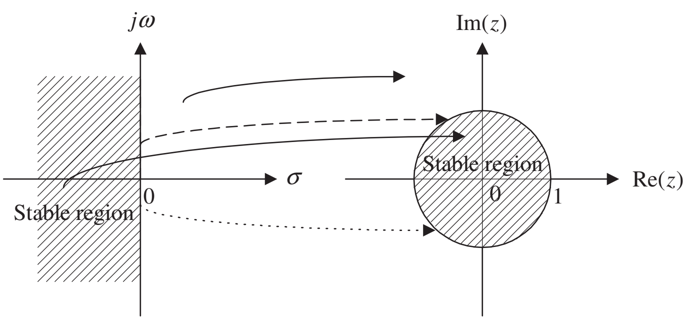

FIG. 8.8

Mapping between the s-plane and the z-plane by the BLT.

To verify these features, let us look at the following illustrative example:

EXAMPLE 8.3

Assuming that $T=2$ s in Eq. (8.12), and given the following points:
(1) $s=-1+j$, on the left half of s-plane
(2) $s=1-j$, on the right half of s-plane
(3) $s=j$, on the positive $j\omega$ on the s-pane
(4) $s=-j$, on the negative $j\omega$ on the s-plane

Convert each of the points in the s-plane to the z-plane, and verify the mapping properties (1) to (3).

Solution:

Substituting $T=2$ into Eq. (8.12) leads to

$$z=\frac{1+s}{1-s}.$$

We can carry out mapping for each point as follows:

(1) $z=\frac{1+(-1+j)}{1-(-1+j)}=\frac{j}{2-j}=\frac{1\angle 90^\circ}{\sqrt{5}\angle -26.57^\circ}=0.4472\angle 116.57^\circ$,

since $|z|=0.4472<1$, which is inside the unit circle on the z-plane.

(2) $z=\frac{1+(1-j)}{1-(1-j)}=\frac{2-j}{j}=\frac{\sqrt{5}\angle -26.57^\circ}{1\angle 90^\circ}=2.2361\angle -116.57^\circ$,

since $|z|=2.2361>1$, which is outside the unit circle on the z-plane.

(3) $z=\frac{1+j}{1-j}=\frac{\sqrt{2}\angle 45^\circ}{\sqrt{2}\angle -45^\circ}=1\angle 90^\circ$, since $|z|=1$ and $\theta=90^\circ$, which is on the positive half circle on the unit circle in the z-plane.

(4) $z=\frac{1-j}{1-(-j)}=\frac{1-j}{1+j}=\frac{\sqrt{2}\angle -45^\circ}{\sqrt{2}\angle 45^\circ}=1\angle -90^\circ$,

since $|z|=1$ and $\theta=-90^\circ$, which is on the negative half circle on the unit circle in the z-plane.

As shown in Example (8.3), the BLT offers conversion of an analog transfer function to a digital transfer function. Example (8.4) shows how to perform the BLT.

EXAMPLE 8.4

Given an analog filter whose transfer function is

$$H(s)=\frac{10}{s+10},$$

convert it to the digital filter transfer function and difference equation, respectively, when a sampling period is given as $T=0.01$ s.

Solution:

Applying the BLT, we have

$$H(z)=H(s)\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}}=\frac{10}{s+10}\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}}.$$

Substituting $T=0.01$, it follows that

$$H(z)=\frac{10}{\frac{200(z-1)}{z+1}+10}=\frac{0.05}{\frac{z-1}{z+1}+0.05}=\frac{0.05(z+1)}{z-1+0.05(z+1)}=\frac{0.05z+0.05}{1.05z-0.95}.$$

Finally, we get

$$H(z)=\frac{(0.05z+0.05)/(1.05z)}{(1.05z-0.95)/(1.05z)}=\frac{0.0476+0.0476z^{-1}}{1-0.9048z^{-1}}.$$

Applying the technique in Chapter 6, we yield the difference equation as

$$y(n)=0.0476x(n)+0.0476x(n-1)+0.9048y(n-1).$$

Next, we examine frequency mapping between the s-plane and the z-plane. As illustrated in Fig. 8.9, the analog frequency ω_a is marked on the $j\omega$ axis on the s-plane, whereas ω_d is the digital frequency labeled on the unit circle in the z-plane.

We substitute $s=jw_a$ and $z=e^{j\omega T}$ into the BLT in Eq. (8.11) to get

$$j\omega_a=\frac{2}{T}\frac{e^{j\omega_d T}-1}{e^{j\omega_d T}+1}. \tag{8.13}$$

Simplifying Eq. (8.13) leads to

$$\omega_a=\frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right). \tag{8.14}$$

Eq. (8.14) explores the relation between the analog frequency on $j\omega$ axis and the corresponding digital frequency ω_d on the unit circle. We can also write its inverse as

$$\omega_d=\frac{2}{T}\tan^{-1}\left(\frac{\omega_a T}{2}\right). \tag{8.15}$$

The range of the digital frequency ω_d is from 0 rad/s to the folding frequency $\omega_s/2$ rad/ s, where ω_s is the sampling frequency in terms of radians per second. We make a plot of Eq. (8.14) in Fig. 8.10.

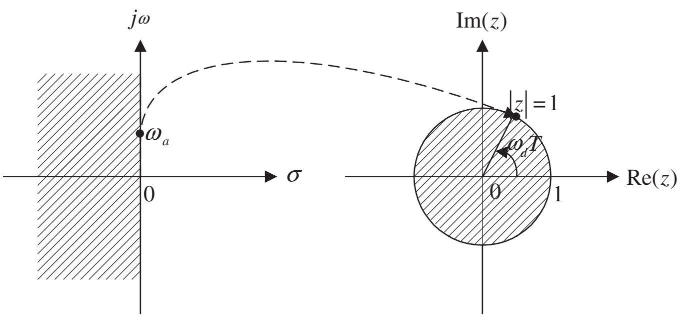

FIG. 8.9

Frequency mapping from the analog domain to the digital domain.

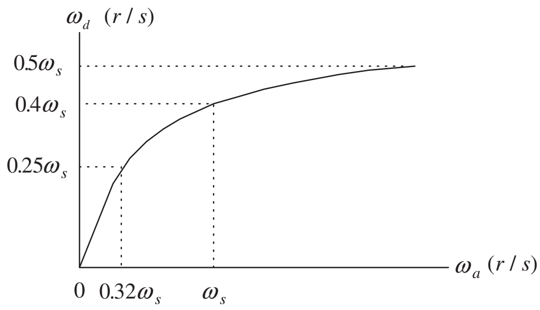

FIG. 8.10

Frequency warping from BLT.

From Fig. 8.10, when the digital frequency range $0 \le \omega_d \le 0.25\omega_s$ is mapped to the analog frequency range $0 \le \omega_a \le 0.32\omega_s$, the transformation appears to be linear; however, the digital frequency range $0.25\omega_s \le \omega_d \le 0.5\omega_s$ is mapped to the analog frequency range for $\omega_a > 0.32\omega_s$, the transformation is nonlinear. The analog frequency range for $\omega_a > 0.32\omega_s$ is compressed into the digital frequency range $0.25\omega_s \le \omega_d \le 0.5\omega_s$. This nonlinear frequency mapping effect is called *frequency warping*. We must incorporate the frequency warping into the IIR filter design.

The following example will illustrate the frequency warping effect in the BLT.

EXAMPLE 8.5

Assume the following analog frequencies:

$\omega_a = 10$ rad/s

$\omega_a = \omega_s/4 = 50\pi = 157$ rad/s

$\omega_a = \omega_s/2 = 100\pi = 314$ rad/s.

Find their digital frequencies using the BLT with a sampling period of 0.01 s, given the analog filter in Example 8.4 and the developed digital filter.

Solution:

From Eq. (8.15), we can calculate digital frequency ω_d as follows:

When $\omega_a = 10$ (rad/s) and $T = 0.01$ (s)

$$\omega_d = \frac{2}{T}\tan^{-1}\left(\frac{\omega_a T}{2}\right) = \frac{2}{0.01}\tan^{-1}\left(\frac{10 \times 0.01}{2}\right) = 9.99\,(\text{rad/s}),$$

which is close to the analog frequency of 10 rad/s. When

$\omega_a = 157$ (rad/s) and $T = 0.01$ (s),

$$\omega_d = \frac{2}{0.01}\tan^{-1}\left(\frac{157 \times 0.01}{2}\right) = 133.11\,(\text{rad/s}),$$

which has an error as compared with the desired value 157. When

$\omega_a = 314$ (rad/s) and $T = 0.01$ (s),

$$\omega_d = \frac{2}{0.01}\tan^{-1}\left(\frac{314 \times 0.01}{2}\right) = 252.5\,(\text{rad/s}),$$

which gives a bigger error compared with the digital folding frequency of 314 rad/s.

Fig. 8.11 shows how to correct the frequency warping error. First, given the digital frequency specification, we prewarp the digital frequency specification to the analog frequency specification by Eq. (8.14).

Second, we obtain the analog lowpass filter $H(s)$ using the prewarped analog frequency ω_a and the lowpass prototype. For the lowpass analog filter, we have

$$H(s) = H_P(s)\big|_{s=\frac{s}{\omega_a}} = H_P\left(\frac{s}{\omega_a}\right). \tag{8.16}$$

Finally, substituting the BLT Eq. (8.11) into Eq. (8.16) yields the digital filter as

$$H(z) = H(s)\big|_{s=\frac{2}{T}\frac{z-1}{z+1}}. \tag{8.17}$$

Similarly, this approach can be extended to other types of filter designs.

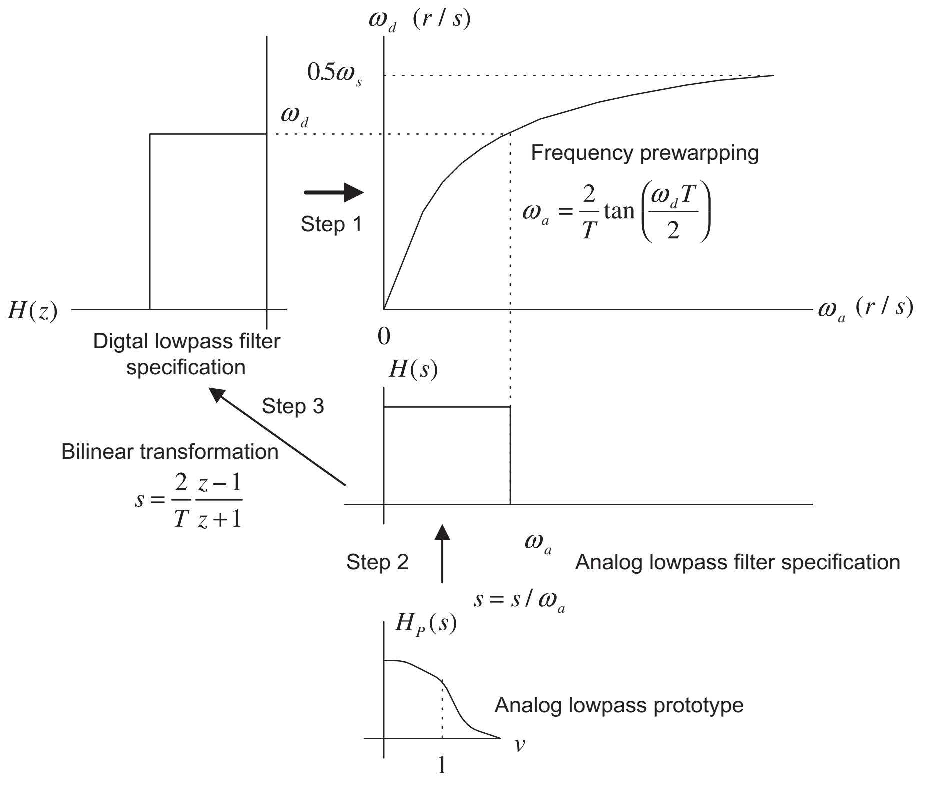

FIG. 8.11

Graphical representation of IIR filter design using the BLT.

8.2.3 BILINEAR TRANSFORMATION DESIGN PROCEDURE

Now we can summarize the BLT design procedure.

1. Given the digital filter frequency specifications, prewarp the digital frequency specifications to the analog frequency specifications.

   For the lowpass filter and the highpass filter:

$$\omega_a = \frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right). \tag{8.18}$$

   For the bandpass filter and the bandstop filter :

$$\omega_{al} = \frac{2}{T}\tan\left(\frac{\omega_l T}{2}\right), \omega_{ah} = \frac{2}{T}\tan\left(\frac{\omega_h T}{2}\right), \tag{8.19}$$

   and $\omega_0 = \sqrt{\omega_{al}\omega_{ah}}$, $W = \omega_{ah} - \omega_{al}$.

2. Perform the prototype transformation using the lowpass prototype $H_P(s)$.

$$\text{From lowpass to lowpass}: H(s) = H_P(s)\big|_{s=\frac{s}{\omega_a}} \tag{8.20}$$

$$\text{From lowpass to highpass}: H(s) = H_P(s)\big|_{s=\frac{\omega_a}{s}} \tag{8.21}$$

$$\text{From lowpass to bandpass}: H(s) = H_P(s)\big|_{s=\frac{s^2+\omega_0^2}{sW}} \tag{8.22}$$

$$\text{From lowpass to bandstop}: H(s) = H_P(s)\big|_{s=\frac{sW}{s^2+\omega_0^2}} \tag{8.23}$$

3. Substitute the BLT to obtain the digital filter

$$H(z) = H(s)\big|_{s=\frac{2}{T}\frac{z-1}{z+1}}. \tag{8.24}$$

Table 8.2 lists MATLAB functions for the BLT design.

Table 8.2 MATLAB Functions for the Bilinear Transformation Design

Lowpass to lowpass: $H(s) = H_P(s)\big|_{s=\frac{s}{\omega_a}}$
>>[B,A] = lp2lp(Bp,Ap,wa)
Lowpass to highpass: $H(s) = H_P(s)\big|_{s=\frac{\omega_a}{s}}$
>>[B,A] = lp2hp(Bp,Ap,wa)
Lowpass to bandpass: $H(s) = H_P(s)\big|_{s=\frac{s^2+\omega_0^2}{sW}}$

Table 8.2 MATLAB Functions for the Bilinear Transformation Design—cont'd

>>[B,A] = lp2bp(Bp,Ap,w0,W)
Lowpass to bandstop: $H(s) = H_P(s)\|_{s=\frac{sW}{s^2+\omega_0^2}}$
>>[B,A] = lp2bs(Bp,Ap,w0,W)
Bilinear transformation to achieve the digital filter:
>>[b,a] = bilinear(B,A,fs)
Plot of the magnitude and phase frequency responses of the digital filter:
>>**freqz(b,a,512,fs)**
Definitions of design parameters:
Bp = vector containing the numerator coefficients of the lowpass prototype.
Ap = vector containing the denominator coefficients of the lowpass prototype.
wa = cutoff frequency for the lowpass or highpass analog filter (rad/s).
w0 = center frequency for the bandpass or bandstop analog filter (rad/s).
W = bandwidth for the bandpass or banstop analog filter (rad/s).
B = vector containing the numerator coefficients of the analog filter.
A = vector containing the denominator coefficients of the analog filter.
b = vector containing the numerator coefficients of the digital filter.
a = vector containing the denominator coefficients of the digital filter.
fs = sampling rate (samples/s).

We illustrate the lowpass filter design procedure in Example 8.6. Other types of filters, such as highpass, bandpass, and bandstop will be illustrated in the next section.

EXAMPLE 8.6

The normalized lowpass filter with a cutoff frequency of 1 rad/s is given as

$$H_P(s) = \frac{1}{s+1}.$$

(a) Use the given $H_P(s)$ and the BLT to design a corresponding digital IIR lowpass filter with a cutoff frequency of 15 Hz and a sampling rate of 90 Hz.
(b) Use MATLAB to plot the magnitude response and phase response of $H(z)$.

Solution:

(a) First, we obtain the digital frequency as
$\omega_d = 2\pi f = 2\pi(15) = 30\pi$ (rad/s), and $T = \frac{1}{f_s} = \frac{1}{90}$ (s).
We then follow the design procedure:

1. First calculate the prewarped analog frequency as

$$\omega_a = \frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right) = \frac{2}{1/90}\tan\left(\frac{30\pi/90}{2}\right),$$

that is, $\omega_a = 180 \times \tan(\pi/6) = 180 \times \tan(30^\circ) = 103.92$ (rad/s).

Continued

EXAMPLE 8.6—CONT'D

2. Then perform the prototype transformation (lowpass to lowpass) as follows:

$$H(s) = H_P(s)\Big|_{s=\frac{s}{\omega_a}} = \frac{1}{\frac{s}{\omega_a}+1} = \frac{\omega_a}{s+\omega_a},$$

which yields an analog filter:

$$H(s) = \frac{103.92}{s+103.92}.$$

3. Applying the BLT yields

$$H(z) = \frac{103.92}{s+103.92}\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}}.$$

We simplify the algebra by dividing both the numerator and the denominator by 180:

$$H(z) = \frac{103.92}{180\times\frac{z-1}{z+1}+103.92} = \frac{103.92/180}{\frac{z-1}{z+1}+103.92/180} = \frac{0.5773}{\frac{z-1}{z+1}+0.5773}.$$

then we multiply both the numerator and the denominator by $(z+1)$ to obtain

$$H(z) = \frac{0.5773(z+1)}{\left(\frac{z-1}{z+1}+0.5773\right)(z+1)} = \frac{0.5773z+0.5773}{(z-1)+0.5773(z+1)} = \frac{0.5773z+0.5773}{1.5773z-0.4227}.$$

Finally, we divide both the numerator and the denominator by $1.5773z$ to get the transfer function in the standard format:

$$H(z) = \frac{(0.5773z+0.5773)/(1.5773z)}{(1.5773z-0.4227)/(1.5773z)} = \frac{0.3660+0.3660z^{-1}}{1-0.2679z^{-1}}.$$

(b) The corresponding MATLAB design is listed in Program 8.2. Fig. 8.12 shows the magnitude and phase frequency responses.

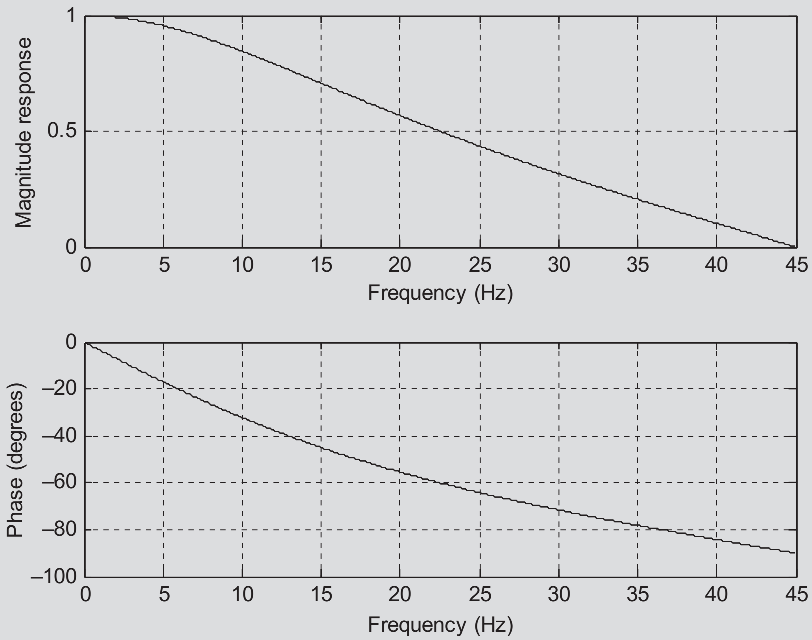

FIG. 8.12

Frequency responses of the designed digital filter for Example 8.6.

Program 8.2. MATLAB program for Example 8.6.

```
%Example 8.6.
% Plot the magnitude and phase responses
 fs=90;   % Sampling rate (Hz)
 [B, A]=lp2lp([1][1 1],103.92);
 [b,a]=bilinear(B,A,fs)
 % b= [0.3660 0.3660] numerator coefficients of the digital filter from MATLAB
 % a= [1 -0.2679]  denominator coefficients of the digital filter from MATLAB
[hz, f]=freqz([0.3660 0.3660][1 -0.2679],512,fs); %the frequency response
phi=180*unwrap(angle(hz))/pi;
subplot(2,1,1), plot(f, abs(hz)),grid;
axis([0 fs/2 0 1]);
xlabel('Frequency (Hz)'); ylabel('Magnitude Response')
subplot(2,1,2), plot(f, phi); grid;
axis([0 fs/2 -100 0]);
xlabel('Frequency (Hz)'); ylabel('Phase (degrees)')
```

8.3 DIGITAL BUTTERWORTH AND CHEBYSHEV FILTER DESIGNS

In this section, we design various types of digital Butterworth and Chebyshev filters using the BLT design method discussed in the previous section.

8.3.1 LOWPASS PROTOTYPE FUNCTION AND ITS ORDER

As described in Section 8.2 (The BLT Design Procedure), BLT design requires obtaining the analog filter with prewarped frequency specifications. These analog filter design requirements include the ripple specification at the passband frequency edge, the attenuation specification at the stopband frequency edge, and the type of lowpass prototype (which we shall discuss) and its order.

Table 8.3 lists the Butterworth prototype functions with 3-dB passband ripple specification. Tables 8.4 and 8.5 contain the Chebyshev prototype functions (type I) with 1- and 0.5-dB passband ripple specifications, respectively. Other lowpass prototypes with different ripple specifications and order can be computed using the methods described in Appendix C.

Table 8.3 3-dB Butterworth Lowpass Prototype Transfer Functions ($\varepsilon = 1$)

n	$H_P(s)$
1	$\frac{1}{s+1}$
2	$\frac{1}{s^2+1.4142s+1}$
3	$\frac{1}{s^3+2s^2+2s+1}$
4	$\frac{1}{s^4+2.6131s^3+3.4142s^2+2.6131s+1}$
5	$\frac{1}{s^5+3.2361s^4+5.2361s^3+5.2361s^2+3.2361s+1}$
6	$\frac{1}{s^6+3.8637s^5+7.4641s^4+9.1416s^3+7.4641s^2+3.8637s+1}$

Table 8.4 Chebyshev Lowpass Prototype Transfer Functions with 0.5 dB Ripple ($\varepsilon = 0.3493$)

n	$H_P(s)$
1	$\frac{2.8628}{s+2.8628}$
2	$\frac{1.4314}{s^2+1.4256s+1.5162}$
3	$\frac{0.7157}{s^3+1.2529s^2+1.5349s+0.7157}$
4	$\frac{0.3579}{s^4+1.1974s^3+1.7169s^2+1.0255s+0.3791}$
5	$\frac{0.1789}{s^5+1.1725s^4+1.9374s^3+1.3096s^2+0.7525s+0.1789}$
6	$\frac{0.0895}{s^6+1.1592s^5+2.1718s^4+1.5898s^3+1.1719s^2+0.4324s+0.0948}$

Table 8.5 Chebyshev Lowpass Prototype Transfer Functions with 1 dB Ripple ($\varepsilon = 0.5088$)

n	$H_P(s)$
1	$\frac{1.9652}{s+1.9652}$
2	$\frac{0.9826}{s^2+1.0977s+1.1025}$
3	$\frac{0.4913}{s^3+0.9883s^2+1.2384s+0.4913}$
4	$\frac{0.2456}{s^4+0.9528s^3+1.4539s^2+0.7426s+0.2756}$
5	$\frac{0.1228}{s^5+0.9368s^4+1.6888s^3+0.9744s^2+0.5805s+0.1228}$
6	$\frac{0.0614}{s^6+0.9283s^5+1.9308s^4+1.20121s^3+0.9393s^2+0.3071s+0.0689}$

In this section, we will focus on Chebyshev type I filter. The Chebyshev type II filter design can be found in Proakis and Manolakis (2007) and Porat (1997).

The magnitude response function of the Butterworth lowpass prototype with order n is shown in Fig. 8.13, where the magnitude response $|H_P(v)|$ vs. the normalized frequency v is given by Eq. (8.25):

$$|H_P(v)| = \frac{1}{\sqrt{1+\varepsilon^2 v^{2n}}}. \tag{8.25}$$

With the given passband ripple A_P dB at the normalized passband frequency edge $v_p = 1$, and the stopband attenuation A_s dB at the normalized stopband frequency edge v_s, the following two equations must be satisfied to determine the prototype filter order:

$$A_P\text{dB} = -20 \times \log_{10}\left(\frac{1}{\sqrt{1+\varepsilon^2}}\right), \tag{8.26}$$

$$A_s\text{dB} = -20 \times \log_{10}\left(\frac{1}{\sqrt{1+\varepsilon^2 v_s^{2n}}}\right). \tag{8.27}$$

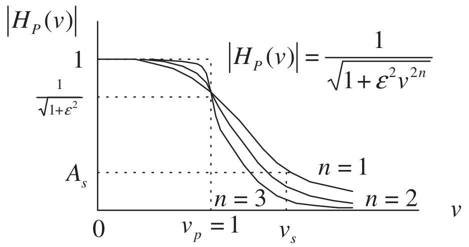

FIG. 8.13

Normalized Butterworth magnitude response function.

Solving Eqs. (8.26) and (8.27), we determine the lowpass prototype order as

$$\varepsilon^2 = 10^{0.1A_p} - 1, \tag{8.28}$$

$$n \geq \frac{\log_{10}\left(\frac{10^{0.1A_s} - 1}{\varepsilon^2}\right)}{[2 \times \log_{10}(v_s)]}, \tag{8.29}$$

where ε is the absolute ripple specification.

The magnitude response function of Chebyshev lowpass prototype with order n is shown in Fig. 8.14, where the magnitude response $|H_P(v)|$ vs. the normalized frequency v is given by

$$|H_P(v)| = \frac{1}{\sqrt{1 + \varepsilon^2 C_n^2(v)}}, \tag{8.30}$$

where

$$C_n(v_s) = \cosh\left[n\cosh^{-1}(v_s)\right] \tag{8.31}$$

$$\cosh^{-1}(v_s) = \ln\left(v_s + \sqrt{v_s^2 - 1}\right). \tag{8.32}$$

As shown in Fig. 8.14, the magnitude response for the Chebyshev lowpass prototype with an odd-numbered order begins with a filter DC gain of 1. In the case of a Chebyshev lowpass prototype an even-numbered order, the magnitude starts at a filter DC gain of $1/\sqrt{1+\varepsilon^2}$. For both cases, the filter gain at the normalized cutoff frequency $v_p = 1$ is $1/\sqrt{1+\varepsilon^2}$.

Similarly, Eqs. (8.33) and (8.34) must be satisfied:

$$A_P \mathrm{dB} = -20 \times \log_{10}\left(\frac{1}{\sqrt{1+\varepsilon^2}}\right), \tag{8.33}$$

$$A_s \mathrm{dB} = -20 \times \log_{10}\left(\frac{1}{\sqrt{1 + \varepsilon^2 C_n^2(v_s)}}\right). \tag{8.34}$$

The lowpass prototype order can be solved in Eqs. (8.35a) and (8.35b):

$$\varepsilon^2 = 10^{0.1A_p} - 1 \tag{8.35a}$$

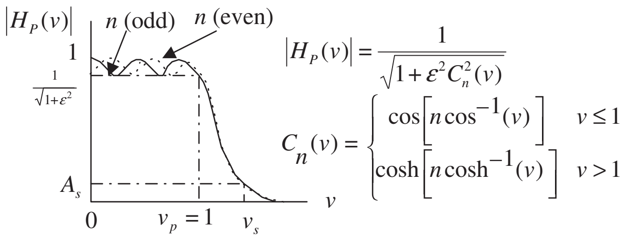

FIG. 8.14

Normalized Chebyshev magnitude response function.

Table 8.6 Conversion from Analog Filter Specifications to Lowpass Prototype Specifications

Analog Filter Specifications	Lowpass Prototype Specifications
Lowpass: ω_{ap}, ω_{as}	$v_p = 1,\ v_s = \omega_{as}/\omega_{ap}$
Highpass: ω_{ap}, ω_{as}	$v_p = 1,\ v_s = \omega_{ap}/\omega_{as}$
Bandpass: ω_{apl}, ω_{aph}, ω_{asl}, ω_{ash}, $\omega_0 = \sqrt{\omega_{apl}\omega_{aph}}$, $\omega_0 = \sqrt{\omega_{asl}\omega_{ash}}$	$v_p = 1,\ v_s = \dfrac{\omega_{ash} - \omega_{asl}}{\omega_{aph} - \omega_{apl}}$
Bandstop: ω_{apl}, ω_{aph}, ω_{asl}, ω_{ash}, $\omega_0 = \sqrt{\omega_{apl}\omega_{aph}}$, $\omega_0 = \sqrt{\omega_{asl}\omega_{ash}}$	$v_p = 1,\ v_s = \dfrac{\omega_{aph} - \omega_{apl}}{\omega_{ash} - \omega_{asl}}$

ω_{ap}, *passband frequency edge;* ω_{as}, *stopband frequency edge;*
ω_{apl}, *lower cutoff frequency in passband;* ω_{aph}, *upper cutoff frequency in passband;*
ω_{asl}, *lower cutoff frequency in stopband;* ω_{ash}, *upper cutoff frequency in stopband;*
ω_0, *geometric center frequency.*

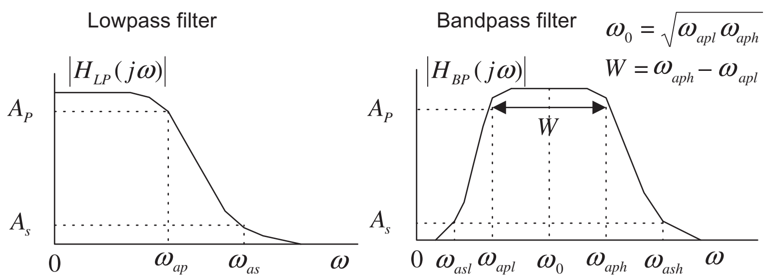

FIG. 8.15

Specifications for analog lowpass and bandpass filters.

$$n \ge \frac{\cosh^{-1}\left[\left(\dfrac{10^{0.1A_s} - 1}{\varepsilon^2}\right)^{0.5}\right]}{\cosh^{-1}(v_s)}, \tag{8.35b}$$

where $\cosh^{-1}(x) = \ln\left(x + \sqrt{x^2 - 1}\right)$, and ε is the absolute ripple parameter.

The normalized stopband frequency v_s can be determined from the frequency specifications of an analog filter illustrated in Table 8.6. Then the order of the lowpass prototype can be determined by Eqs. (8.29) for the Butterworth function and by Eq. (8.35b) for the Chebyshev function. Fig. 8.15 gives frequency edge notations for analog lowpass and bandpass filters. The notations for analog highpass and bandstop filters can be defined correspondingly.

8.3.2 LOWPASS AND HIGHPASS FILTER DESIGN EXAMPLES

The following examples illustrate various designs for the Butterworth and Chebyshev lowpass and highpass filters.

EXAMPLE 8.7

(a) Design a digital lowpass Butterworth filter with the following specifications:
 (1) 3-dB attenuation at the passband frequency of 1.5 kHz.
 (2) 10-dB stopband attenuation at the frequency of 3 kHz.
 (3) Sampling frequency at 8000 Hz.

(b) Use MATLAB to plot the magnitude and phase responses.

Solution:

(a) First, we obtain the digital frequencies in radians per second:

$$\omega_{dp} = 2\pi f = 2\pi(1500) = 3000\pi \text{ (rad/s)}$$

$$\omega_{ds} = 2\pi f = 2\pi(3000) = 6000\pi \text{ (rad/s)}$$

$$T = \frac{1}{f_s} = \frac{1}{8000} \text{ (s)}$$

Following the steps of the design procedure,

(1) We apply the warping equation as

$$\omega_{ap} = \frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right) = 16{,}000 \times \tan\left(\frac{3000\pi/8000}{2}\right) = 1.0691 \times 10^4 \text{ (rad/s)}.$$

$$\omega_{as} = \frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right) = 16{,}000 \times \tan\left(\frac{6000\pi/8000}{2}\right) = 3.8627 \times 10^4 \text{ (rad/s)}.$$

We then find the lowpass prototype specifications using Table 8.6 as follows:

$$v_s = \frac{\omega_{as}}{\omega_{ap}} = \frac{3.8627 \times 10^4}{1.0691 \times 10^4} = 3.6130 \text{ and } A_s = 10 \text{ dB}.$$

The filter order is computed as

$$\varepsilon^2 = 10^{0.1\times 3} - 1 = 1$$

$$n = \frac{\log_{10}\left(10^{0.1\times 10} - 1\right)}{2 \times \log_{10}(3.6130)} = 0.8553.$$

(2) Rounding n up, we choose $n = 1$ for the lowpass prototype. From Table 8.3, we have

$$H_P(s) = \frac{1}{s+1}.$$

Applying the prototype transformation (lowpass to lowpass) yields the analog filter

$$H(s) = H_P(s)\Big|_{\frac{s}{\omega_{ap}}} = \frac{1}{\frac{s}{\omega_{ap}} + 1} = \frac{\omega_{ap}}{s + \omega_{ap}} = \frac{1.0691 \times 10^4}{s + 1.0691 \times 10^4}.$$

(3) Finally, using the BLT, we have

$$H(z) = \frac{1.0691 \times 10^4}{s + 1.0691 \times 10^4}\bigg|_{s = 16000(z-1)/(z+1)}.$$

Substituting the BLT leads to

$$H(z) = \frac{1.0691 \times 10^4}{\left(16,000\frac{z-1}{z+1}\right) + 1.0691 \times 10^4}.$$

To simplify the algebra, we divide both numerator and denominator by 16,000 to get

$$H(z) = \frac{0.6682}{\left(\frac{z-1}{z+1}\right) + 0.6682}.$$

Then multiplying both numerator and denominator by $(z+1)$ leads to

$$H(z) = \frac{0.6682(z+1)}{(z-1) + 0.6682(z+1)} = \frac{0.6682z + 0.6682}{1.6682z - 0.3318}.$$

Dividing both numerator and denominator by $(1.6682 \times z)$ leads to

$$H(z) = \frac{0.4006 + 0.4006z^{-1}}{1 - 0.1989z^{-1}}.$$

(b) Steps 2 and 3 can be carried out using the MATLAB Program 8.3, as shown in the first three lines of the MATLAB codes. Fig. 8.16 describes the filter frequency responses.

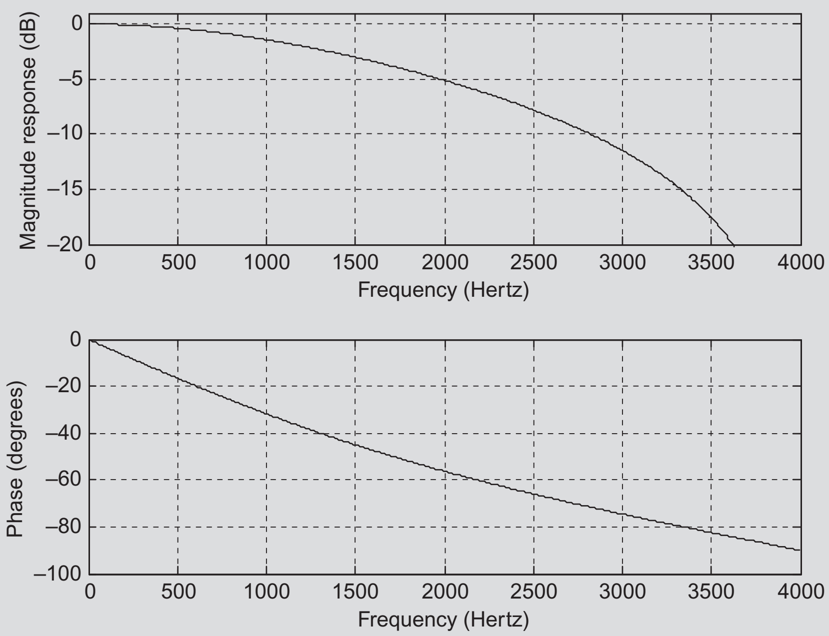

FIG. 8.16

Frequency responses of the designed digital filter for Example 8.7.

Program 8.3. MATLAB program for Example 8.7.

```
%Example 8.7
% Design of the digital lowpass Butterworth filter
format long
fs=8000;                           % Sampling rate
[B A]=lp2lp([1][1 1], 1.0691*10^4) % Complete step 2
[b a]=bilinear(B,A,fs) % Complete step 3
% Plot the magnitude and phase responses |H(z)|
% b=[0.4005 0.4005]; numerator coefficients from MATLAB
% a=[1 -0.1989]; denominator coefficients from MATLAB
freqz(b,a,512,fs);
axis([0 fs/2 -20 1])
```

EXAMPLE 8.8

(a) Design a first-order highpass digital Chebyshev filter with a cutoff frequency of 3 kHz and 1 dB ripple on passband using a sampling frequency of 8000 Hz.
(b) Use MATLAB to plot the magnitude and phase responses.

Solution:

(a) First, we obtain the digital frequency in radians per second:

$$\omega_d = 2\pi f = 2\pi(3000) = 6000\pi \text{ (rad/s), and } T = \frac{1}{f_s} = \frac{1}{8000} \text{ (s)}.$$

Following the steps of the design procedure, we have

1. $\omega_a = \frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right) = 16{,}000 \times \tan\left(\frac{6000\pi/8000}{2}\right) = 3.8627 \times 10^4 \text{ (rad/s)}.$
2. Since the filter order is given as 1, we select the first-order lowpass prototype from Table 8.5 as

$$H_P(s) = \frac{1.9652}{s + 1.9652}.$$

Applying the prototype transformation (lowpass to highpass), we obtain

$$H(s) = H_P(s)\Big|_{\frac{\omega_a}{s}} = \frac{1.9652}{\frac{\omega_a}{s} + 1.9652} = \frac{1.9652s}{1.9652s + 3.8627 \times 10^4}.$$

Dividing both numerator and denominator by 1.9652 gives

$$H(s) = \frac{s}{s + 1.9656 \times 10^4}.$$

3. Using the BLT, we have

$$H(z) = \frac{s}{s + 1.9656 \times 10^4}\Bigg|_{s=16{,}000(z-1)/(z+1)}.$$

Algebra work is demonstrated as follows:

$$H(z) = \frac{16{,}000\dfrac{z-1}{z+1}}{16{,}000\dfrac{z-1}{z+1} + 1.9656 \times 10^4}.$$

Simplifying the transfer function yields

$$H(z) = \frac{0.4487 - 0.4487z^{-1}}{1 + 0.1025z^{-1}}.$$

(b) Steps 2 and 3 and frequency response plots shown in Fig. 8.17 can be carried out using the MATLAB Program 8.4.

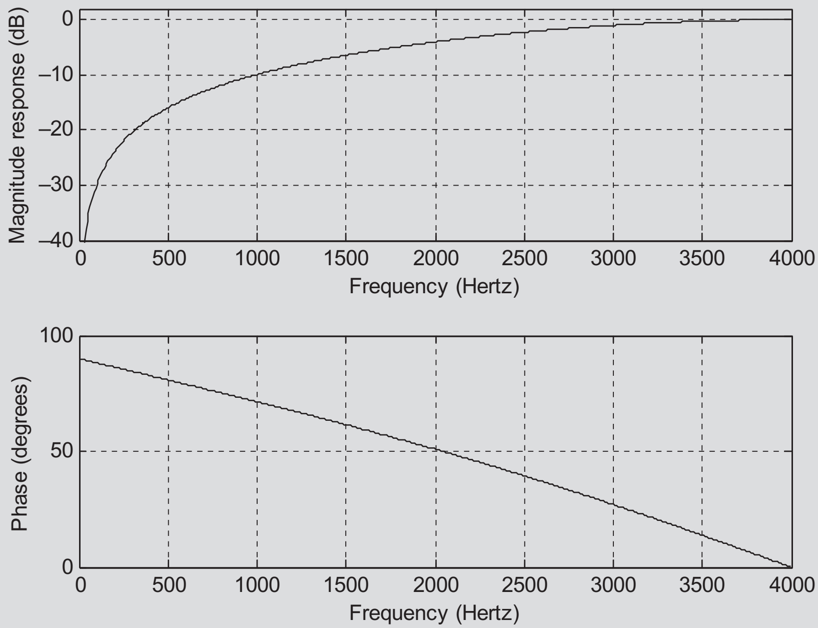

FIG. 8.17

Frequency responses of the designed digital filter for Example 8.8.

Program 8.4. MATLAB program for Example 8.8

```
%Example 8.8
% Design of the digital highpass Butterworth filter
format long
fs=8000; % Sampling rate
[B A]=lp2hp([1.9652][1 1.9652], 3.8627*10^4) % Complete step 2
```

```
[b a]=bilinear(B,A,fs) % Complete step 3
% Plot the magnitude and phase responses |H(z)|
% b=[0.4487 -0.4487 ]; numerator coefficients from MATLAB
% a=[1 0.1025]; denominator coefficients from MATLAB
freqz(b,a,512,fs);
axis([0 fs/2 -40 2])
```

EXAMPLE 8.9

(a) Design a second-order lowpass digital Butterworth filter with a cutoff frequency of 3.4 kHz at a sampling frequency of 8000 Hz.
(b) Use MATLAB to plot the magnitude and phase responses.

Solution:

(a) First, we obtain the digital frequency in radians per second:

$$\omega_d = 2\pi f = 2\pi(3400) = 6800\pi \text{ (rad/s), and } T = \frac{1}{f_s} = \frac{1}{8000} \text{ (s)}.$$

Following the steps of the design procedure, we compute the prewarped analog frequencies as

1.
$$\omega_a = \frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right) = 16{,}000 \times \tan\left(\frac{6800\pi/8000}{2}\right) = 6.6645 \times 10^4 \text{ (rad/s)}.$$

2. Since the order of 2 is given in the specification, we directly pick the second-order lowpass prototype from Table 8.3:

$$H_P(s) = \frac{1}{s^2 + 1.4142s + 1}.$$

After applying the prototype transformation (lowpass to lowpass), we have

$$H(s) = H_P(s)\Big|_{\frac{s}{\omega_a}} = \frac{4.4416 \times 10^9}{s^2 + 9.4249 \times 10^4 s + 4.4416 \times 10^9}.$$

3. Carrying out the BLT yields

$$H(z) = \left.\frac{4.4416 \times 10^9}{s^2 + 9.4249 \times 10^4 s + 4.4416 \times 10^9}\right|_{s=16000(z-1)/(z+1)}.$$

Let us work on algebra:

$$H(z) = \frac{4.4416 \times 10^9}{\left(16{,}000\frac{z-1}{z+1}\right)^2 + 9.4249 \times 10^4\left(16{,}000\frac{z-1}{z+1}\right) + 4.4416 \times 10^9}.$$

To simplify, we divide both numerator and denominator by $(16,000)^2$ to get

$$H(z) = \frac{17.35}{\left(\frac{z-1}{z+1}\right)^2 + 5.8906\left(\frac{z-1}{z+1}\right) + 17.35}.$$

Then multiplying both numerator and denominator by $(z+1)^2$ leads to

$$H(z) = \frac{17.35(z+1)^2}{(z-1)^2 + 5.8906(z-1)(z+1) + 17.35(z+1)^2}.$$

Using identities, we have

$$H(z) = \frac{17.35(z^2+2z+1)}{(z^2-2z+1) + 5.8906(z^2-1) + 17.35(z^2+2z+1)} = \frac{17.35z^2 + 34.7z + 17.35}{24.2406z^2 + 32.7z + 12.4594}.$$

Dividing both numerator and denominator by $(24.2406z^2)$ leads to

$$H(z) = \frac{0.7157 + 1.4314z^{-1} + 0.7151z^{-2}}{1 + 1.3490z^{-1} + 0.5140z^{-2}}.$$

(b) Steps 2 and 3 require a certain amount of algebra work and can be verified using MATLAB Program 8.5, as shown in the first three lines of the code. Fig. 8.18 plots the filter magnitude and phase frequency responses.

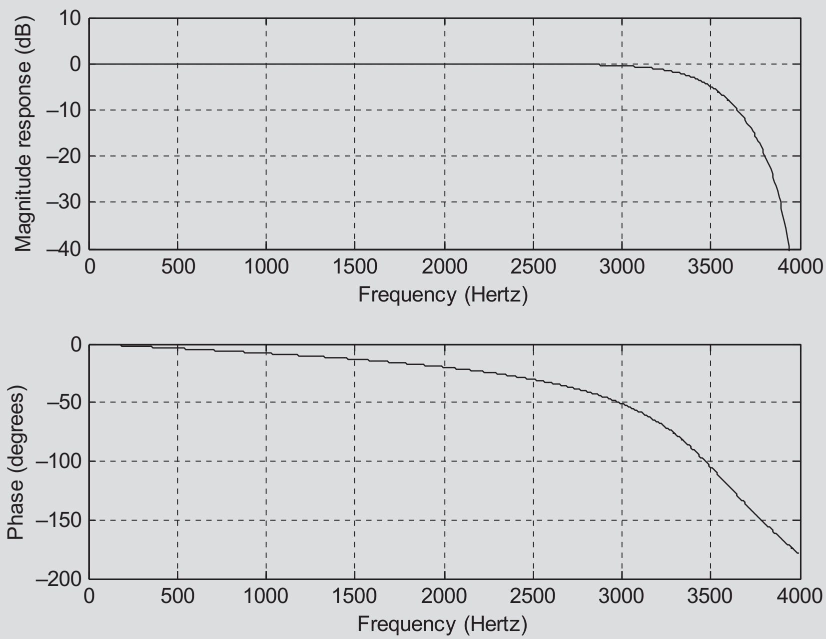

FIG. 8.18

Frequency responses of the designed digital filter for Example 8.9.

Program 8.5. MATLAB program for Example 8.9.

```
%Example 8.9
% Design of the digital lowpass Butterworth filter
format long
fs=8000;   % Sampling rate
[B A]=lp2lp([1][1 1.4142 1], 6.6645*10^4) % Complete step 2
[b a]=bilinear(B,A,fs) % Complete step 3
% Plot the magnitude and phase responses |H(z)|
% b=[0.7157 1.4315 0.7157]; numerator coefficients from MATLAB
%a=[1 1.3490 0.5140]; denominator coefficients from MATLAB
freqz(b,a,512,fs);
axis([0 fs/2 -40 10])
```

EXAMPLE 8.10

(a) Design a highpass digital Chebyshev filter with the following specifications:
1. 0.5-dB ripple on passband at frequency of 3000 Hz.
2. 25-dB attenuation at frequency of 1000 Hz.
3. Sampling frequency at 8000 Hz.

(b) Use MATLAB to plot the magnitude and phase responses.

Solution:

(a) From the specifications, the digital frequencies are

$$\omega_{dp} = 2\pi f = 2\pi(3000) = 6000\pi \text{ (rad/s)}$$

$$\omega_{ds} = 2\pi f = 2\pi(1000) = 2000\pi \text{ (rad/s)}$$

$$\text{and } T = \frac{1}{f_s} = \frac{1}{8000} \text{ (s)}.$$

Using the design procedure, it follows that

$$\omega_{ap} = \frac{2}{T}\tan\left(\frac{\omega_{dp}T}{2}\right) = 16{,}000 \times \tan\left(\frac{6000\pi/8000}{2}\right) = 3.8627 \times 10^4 \text{ (rad/s)}$$

$$\omega_{as} = 16{,}000 \times \tan\left(\frac{\omega_{ds}T}{2}\right) = 16{,}000 \times \tan\left(\frac{2000\pi/8000}{2}\right) = 6.6274 \times 10^3 \text{ (rad/s)}.$$

We find the lowpass prototype specification as follows:

$$v_s = \frac{\omega_{ps}}{\omega_{sp}} = \frac{3.8627 \times 10^4}{6.6274 \times 10^3} = 5.8284 \text{ and } A_s = 25 \text{ dB},$$

then the filter order is computed as

$$\varepsilon^2 = 10^{0.1\times 0.5} - 1 = 0.1220$$

$$\frac{\left(10^{0.1\times 25} - 1\right)}{0.1220} = 2583.8341$$

$$n = \frac{\cosh^{-1}\left[(2583.8341)^{0.5}\right]}{\cosh^{-1}(5.8284)} = \frac{\ln\left(50.8314 + \sqrt{50.8314^2 - 1}\right)}{\ln\left(5.8284 + \sqrt{5.8284^2 - 1}\right)} = 1.8875.$$

We select $n = 2$ for the lowpass prototype function. Following the steps of the design procedure, it follows that

1. $\omega_p = 3.8627 \times 10^4$ (rad/s).
2. Performing the prototype transformation (lowpass to lowpass) using the prototype filter illustrated in Table 8.4, we have

$$H_P(s) = \frac{1.4314}{s^2 + 1.4256s + 1.5162} \text{ and}$$

$$H(s) = H_P(s)\big|_{\frac{s}{\omega_a}} = \frac{1.4314}{\left(\frac{\omega_p}{s}\right)^2 + 1.4256\left(\frac{\omega_p}{s}\right) + 1.5162} = \frac{0.9441s^2}{s^2 + 3.6319 \times 10^4 s + 9.8407 \times 10^8}.$$

3. Hence, applying the BLT, we convert the analog filter to the digital filter as

$$H(z) = \left.\frac{0.9441s^2}{s^2 + 3.6319 \times 10^4 s + 9.8407 \times 10^8}\right|_{s=16,000(z-1)/(z+1)}.$$

After algebraic simplification, it follows that

$$H(z) = \frac{0.1327 - 0.2654z^{-1} + 0.1327z^{-2}}{1 + 0.7996z^{-1} + 0.3618z^{-2}}.$$

(b) MATLAB Program 8.6 is listed for this example, and frequency responses are given in Fig. 8.19.

Program 8.6. MATLAB program for Example 8.10.

```
%Example 8.10
% Design of the digital highpass Chebyshev filter
format long
 fs=8000;  % Sampling rate
 % BLT design
[B A]=lp2hp([1.4314][1 1.4256 1.5162], 3.8627*10^4) % Complete step 2
[b a]=bilinear(B,A,fs) % Complete step 3
% Plot the magnitude and phase responses |H(z)|
% b=[0.1327 -0.2654 0.1327]; numerator coefficients from MATLAB
% a=[1 0.7996 0.3618]; denominator coefficients from MATLAB
freqz(b,a,512,fs);
axis([0 fs/2 -40 10])
```

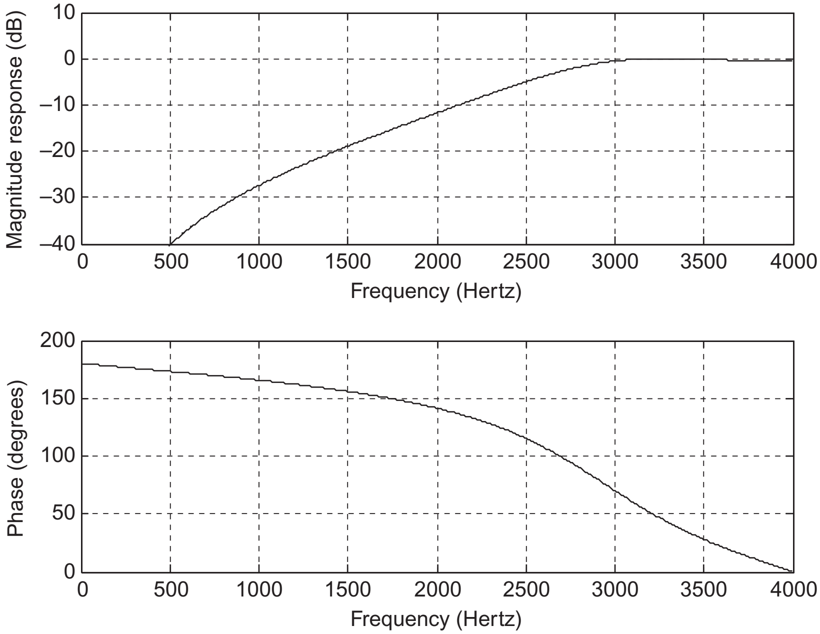

FIG. 8.19

Frequency responses of the designed digital filter for Example 8.10.

8.3.3 BANDPASS AND BANDSTOP FILTER DESIGN EXAMPLES

EXAMPLE 8.11

(a) Design a second-order digital bandpass Butterworth filter with the following specifications:
- an upper cutoff frequency of 2.6kHz,
- a lower cutoff frequency of 2.4kHz, and
- a sampling frequency of 8000Hz.

(b) Use MATLAB to plot the magnitude and phase responses.

Solution:

(a) Let us find the digital frequencies in radians per second:
- $\omega_h = 2\pi f_h = 2\pi(2600) = 5200\pi$ (rad/s)
- $\omega_l = 2\pi f_l = 2\pi(2400) = 4800\pi$ (rad/s), and $T = \frac{1}{f_s} = \frac{1}{8000}$ (s) s.

Following the steps of the design procedure, we have the following:

1. $\omega_{ah} = \frac{2}{T}\tan\left(\frac{\omega_h T}{2}\right) = 16{,}000 \times \tan\left(\frac{5200\pi/8000}{2}\right) = 2.6110 \times 10^4$ (rad/s)

$$\omega_{al} = 16{,}000 \times \tan\left(\frac{\omega_l T}{2}\right) = 16{,}000 \times \tan(0.3\pi) = 2.2022 \times 10^4 \text{ (rad/s)}$$

$$W = \omega_{ah} - \omega_{al} = 26{,}110 - 22{,}022 = 4088 \text{ (rad/s)}$$

$$\omega_0^2 = \omega_{ah} \times \omega_{al} = 5.7499 \times 10^8.$$

2. We perform the prototype transformation (lowpass to bandpass) to obtain $H(s)$. From Table 8.3, we pick the lowpass prototype with order of 1 to produce a bandpass filter with order of 2 as

$$H_P(s) = \frac{1}{s+1},$$

and applying the lowpass to bandpass transformation, it follows that

$$H(s) = H_P(s)\Big|_{\frac{s^2+\omega_0^2}{sW}} = \frac{Ws}{s^2 + Ws + \omega_0^2} = \frac{4088s}{s^2 + 4088s + 5.7499 \times 10^8}.$$

3. Hence we apply the BLT to yield

$$H(z) = \frac{4088s}{s^2 + 4088s + 5.7499 \times 10^8}\Bigg|_{s=16{,}000(z-1)/(z+1)}.$$

Via algebra work, we obtain the digital filter as

$$H(z) = \frac{0.0730 - 0.0730z^{-2}}{1 + 0.7117z^{-1} + 0.8541z^{-2}}.$$

(b) MATLAB Program 8.7 is given for this example, and the corresponding frequency response plots are illustrated in Fig. 8.20.

Program 8.7. MATLAB program for Example 8.11

```
%Example 8.11
% Design of the digital bandpass Butterworth filter
format long
fs=8000;
[B A]=lp2bp([1][1 1],sqrt(5.7499*10ˆ8),4088) % Complete step 2
[b a]=bilinear(B,A,fs) % Complete step 3
% Plot the magnitude and phase responses |H(z)|
% b=[0.0730 0 -0.0730]; numerator coefficients from MATLAB
% a=[1 0.7117 0.8541]; denominator coefficients form MATLAB
freqz(b, a,512,fs);
axis([0 fs/2 -40 10])
```

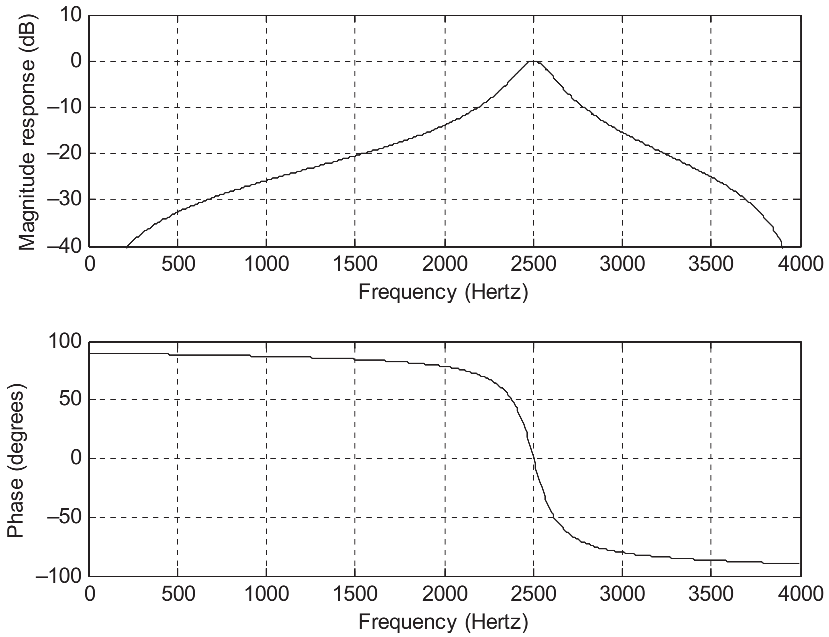

FIG. 8.20

Frequency responses of the designed digital filter for Example 8.11.

EXAMPLE 8.12

Now let us examine the bandstop Butterworth filter design.

(a) Design a digital bandstop Butterworth filter with the following specifications:
- Center frequency of 2.5 kHz
- Passband width of 200 Hz and ripple of 3 dB
- Stopband width of 50 Hz and attenuation of 10 dB
- Sampling frequency of 8000 Hz

(b) Use MATLAB to plot the magnitude and phase responses.

Solution:

(a) The digital frequencies of the digital filter are as follows:

$$\omega_h = 2\pi f_h = 2\pi(2600) = 5200\pi \text{ (rad/s)},$$

$$\omega_l = 2\pi f_l = 2\pi(2400) = 4800\pi \text{ (rad/s)},$$

$$\omega_{d0} = 2\pi f_0 = 2\pi(2500) = 5000\pi \text{ (rad/s)}, \text{and } T = 1/f_s = 1/8000 \text{ (s)}.$$

Applying the steps of the IIR filter design approach, it follows that

1.

$$\omega_{ah} = \frac{2}{T}\tan\left(\frac{\omega_h T}{2}\right) = 16{,}000 \times \tan\left(\frac{5200\pi/8000}{2}\right) = 2.6110 \times 10^4 \text{ (rad/s)}$$

$$\omega_{al} = 16{,}000 \times \tan\left(\frac{\omega_l T}{2}\right) = 16{,}000 \times \tan(0.3\pi) = 2.2022 \times 10^4 \text{ (rad/s)}$$

$$\omega_0 = 16{,}000 \times \tan\left(\frac{\omega_{d0} T}{2}\right) = 16{,}000 \times \tan(0.3125\pi) = 2.3946 \times 10^4 \text{ (rad/s)}$$

$$\omega_{sh} = \frac{2}{T}\tan\left(\frac{2525 \times 2\pi/8000}{2}\right) = 16{,}000 \times \tan\left(56.8125^\circ\right) = 2.4462 \times 10^4 \text{ (rad/s)}$$

$$\omega_{sl} = 16{,}000 \times \tan\left(\frac{2475 \times 2\pi/8000}{2}\right) = 16{,}000 \times \tan\left(55.6875^\circ\right) = 2.3444 \times 10^4 \text{ (rad/s)}$$

To adjust the unit passband gain at the center frequency of 2500 Hz, we perform the following:
Fixing $\omega_{al} = 2.2022 \times 10^4$, we compute $\omega_{ah} = \frac{\omega_0^2}{\omega_{al}} = \frac{\left(2.3946 \times 10^4\right)^2}{2.2022 \times 10^4} = 2.6037 \times 10^4$
and the passband bandwidth: $W = \omega_{ah} - \omega_{al} = 4015$
Fixing $\omega_{sl} = 2.3444 \times 10^4$, $\omega_{sh} = \omega_0^2/\omega_{sl} = \frac{\left(2.3946 \times 10^4\right)^2}{2.3444 \times 10^4} = 2.4459 \times 10^4$
and the stopband bandwidth: $W_s = \omega_{sh} - \omega_{sl} = 1015$
Again, fixing $\omega_{ah} = 2.6110 \times 10^4$, we get $\omega_{al} = \frac{\omega_0^2}{\omega_{ah}} = \frac{\left(2.3946 \times 10^4\right)^2}{2.6110 \times 10^4} = 2.1961 \times 10^4$
and the passband bandwidth: $W = \omega_{ah} - \omega_{al} = 4149$
Fixing $\omega_{sh} = 2.4462 \times 10^4$, $\omega_{sl} = \frac{\omega_0^2}{\omega_{sh}} = \frac{\left(2.3946 \times 10^4\right)^2}{2.4462 \times 10^4} = 2.3441 \times 10^4$
and the stopband bandwidth: $W_s = \omega_{sh} - \omega_{sl} = 1021$
For an aggressive bandstop design, we choose $\omega_{al} = 2.6110 \times 10^4$, $\omega_{ah} = 2.1961 \times 10^4$, $\omega_{sl} = 2.3441 \times 10^4$, $\omega_{sh} = 2.4462 \times 10^4$, and $\omega_0 = 2.3946 \times 10^4$ to satisfy a larger bandwidth.
Thus we develop the prototype specification

$$v_s = \frac{26{,}110 - 21{,}916}{24{,}462 - 23{,}441} = 4.0177$$

$n = \left(\frac{\log_{10}\left(10^{0.1 \times 10} - 1\right)}{2 \times \log_{10}(4.0177)}\right) = 0.7899$, choose $n = 1$.
$W = \omega_{ah} - \omega_{al} = 26,\ 110 - 21,\ 961 = 4149 \text{ (rad/s)}$, $\omega_0^2 = 5.7341 \times 10^8$.

2. Then, carrying out the prototype transformation (lowpass to bandstop) using the first-order lowpass prototype filter given by

$$H_P(s) = \frac{1}{s+1},$$

it follows that

$$H(s) = H_P(s)\Big|_{\frac{sW}{s^2+\omega_0^2}} = \frac{\left(s^2 + \omega_0^2\right)}{s^2 + Ws + \omega_0^2}.$$

Substituting the values of ω_0^2 and W yields

$$H(s) = \frac{s^2 + 5.7341 \times 10^8}{s^2 + 4149s + 5.7341 \times 10^8}.$$

Continued

EXAMPLE 8.12—CONT'D

3. Hence, applying the BLT leads to

$$H(z) = \left. \frac{s^2 + 5.7341 \times 10^8}{s^2 + 4149s + 5.73411 \times 10^8} \right|_{s=16,000(z-1)/(z+1)}.$$

After BLT, we get

$$H(z) = \frac{0.9259 + 0.7078z^{-1} + 0.9249z^{-2}}{1 + 0.7078z^{-1} + 0.8518z^{-2}}.$$

(b) MATLAB Program 8.8 includes the design steps. Fig. 8.21 shows the filter frequency responses.

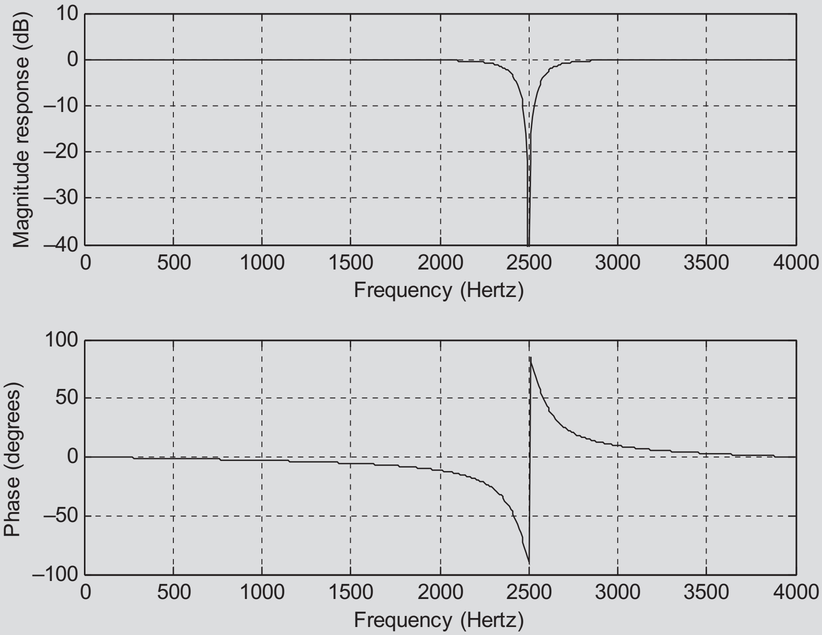

FIG. 8.21

Frequency responses of the designed digital filter for Example 8.12.

Program 8.8. MATLAB program for Example 8.12.

```
%Example 8.12
% Design of the digital bandstop Butterworth filter
format long
```

```
fs=8000; % Sampling rate
[B A]=lp2bs([1][1 1],sqrt(5.7341*10^8),4149) % Complete step 2
[b a]=bilinear(B,A,fs) % Complete step 3
% Plot the magnitude and phase responses |H(z)|
% b=[0.9259 0.7078 0.9259]; numerator coefficients from MATLAB
% a=[1 0.7078 0.8518]; denominator coefficients from MATLAB
freqz(b,a,512,fs);
axis([0 fs/2 -40 10])
```

EXAMPLE 8.13

(a) Design a digital bandpass Chebyshev filter with the following specifications:
- Center frequency of 2.5 kHz
- Passband bandwidth of 200 Hz, 0.5 dB ripple on passband
- Lower stop frequency of 1.5 kHz, upper stop frequency of 3.5 kHz
- Stopband attenuation of 10 dB
- Sampling frequency of 8000 Hz

(b) Use MATLAB to plot the magnitude and phase responses.

Solution:

(a) The digital frequencies are given as

$$\omega_{dph} = 2\pi f_{dph} = 2\pi(2600) = 5200\pi \text{ (rad/s)},$$

$$\omega_{dpl} = 2\pi f_{dpl} = 2\pi(2400) = 4800\pi \text{ (rad/s)},$$

$$\omega_{d0} = 2\pi f_0 = 2\pi(2500) = 5000\pi \text{ (rad/s)}, \text{and } T = 1/f_s = 1/8000 \text{ (s)}.$$

Applying the frequency prewarping equation, it follows that

$$\omega_{aph} = \frac{2}{T}\tan\left(\frac{\omega_{dph}T}{2}\right) = 16{,}000 \times \tan\left(\frac{5200\pi/8000}{2}\right) = 2.6110 \times 10^4 \text{ (rad/s)}$$

$$\omega_{apl} = 16{,}000 \times \tan\left(\frac{\omega_{dpl}T}{2}\right) = 16{,}000 \times \tan(0.3\pi) = 2.2022 \times 10^4 \text{ (rad/s)}$$

$$\omega_0 = 16{,}000 \times \tan\left(\frac{\omega_{d0}T}{2}\right) = 16{,}000 \times \tan(0.3125\pi) = 2.3946 \times 10^4 \text{ (rad/s)}$$

$$\omega_{ash} = 16{,}000 \times \tan\left(\frac{3500 \times 2\pi/8000}{2}\right) = 16{,}000 \times \tan\left(78.75^{\circ}\right) = 8.0437 \times 10^4 \text{ (rad/s)}$$

$$\omega_{asl} = 16{,}000 \times \tan\left(\frac{1500 \times 2\pi/8000}{2}\right) = 1.0691 \times 10^4 \text{ (rad/s)}$$

Now, adjusting the unit gain for the center frequency of 2500 Hz leads to the following:

Fixing $\omega_{apl} = 2.2022 \times 10^4$, we have $\omega_{aph} = \frac{\omega_0^2}{\omega_{apl}} = \frac{\left(2.3946 \times 10^4\right)^2}{2.2022 \times 10^4} = 2.6038 \times 10^4$

and the passband bandwidth: $W = \omega_{aph} - \omega_{apl} = 4016$

Continued

EXAMPLE 8.13—CONT'D

Fixing $\omega_{asl} = 1.0691 \times 10^4$, $\omega_{ash} = \frac{\omega_0^2}{\omega_{asl}} = \frac{(2.3946 \times 10^4)^2}{2.10691 \times 10^4} = 5.3635 \times 10^4$

and the stopband bandwidth: $W_s = \omega_{ash} - \omega_{asl} = 42,\ 944$

Again, fixing $\omega_{aph} = 2.6110 \times 10^4$, we have $\omega_{apl} = \frac{\omega_0^2}{\omega_{aph}} = \frac{(2.3946 \times 10^4)^2}{2.6110 \times 10^4} = 2.1961 \times 10^4$

and the passband bandwidth: $W = \omega_{aph} - \omega_{apl} = 4149$

Fixing $\omega_{ash} = 8.0437 \times 10^4$, $\omega_{asl} = \frac{\omega_0^2}{\omega_{ash}} = \frac{(2.3946 \times 10^4)^2}{8.0437 \times 10^4} = 0.7137 \times 10^4$

and the stopband bandwidth: $W_s = \omega_{ash} - \omega_{asl} = 73,\ 300$

For an aggressive bandpass design, we select $\omega_{apl} = 2.2022 \times 10^4$, $\omega_{aph} = 2.6038 \times 10^4$, $\omega_{asl} = 1.0691 \times 10^4$, $\omega_{ash} = 5.3635 \times 10^4$ for a smaller bandwidth for passband.
Thus, we obtain the prototype specifications:

$$v_s = \frac{53{,}635 - 10{,}691}{26{,}038 - 22{,}022} = 10.6932$$

$$\varepsilon^2 = 10^{0.1 \times 0.5} - 1 = 0.1220$$

$$\frac{\left(10^{0.1 \times 10} - 1\right)}{0.1220} = 73.7705$$

$$n = \frac{\cosh^{-1}\left[(73.7705)^{0.5}\right]}{\cosh^{-1}(10.6932)} = \frac{\ln\left(8.5890 + \sqrt{8.5890^2 - 1}\right)}{\ln\left(10.6932 + \sqrt{10.6932^2 - 1}\right)} = 0.9280$$

rounding up n leads to $n = 1$.
Applying the design steps leads to

1. $\omega_{aph} = 2.6038 \times 10^4\,(\text{rad/s})$, $\omega_{apl} = 2.2022 \times 10^4(\text{rad/s})$

$$W = 4016\,(\text{rad/s}), \omega_0^2 = 5.7341 \times 10^8$$

2. Performing the prototype transformation (lowpass to bandpass), we obtain

$$H_P(s) = \frac{2.8628}{s + 2.8628}$$

and

$$H(s) = H_P(s)\Big|_{s = \frac{s^2 + \omega_0^2}{sW}} = \frac{2.8628Ws}{s^2 + 2.8628Ws + \omega_0^2} = \frac{1.1497 \times 10^4 s}{s^2 + 1.1497 \times 10^4 s + 5.7341 \times 10^8}.$$

3. Applying the BLT, the analog filter is converted to a digital filter as follows:

$$H(z) = \frac{1.1497 \times 10^4 s}{s^2 + 1.1497 \times 10^4 s + 5.7341 \times 10^8}\Bigg|_{s = 16{,}000(z-1)/(z+1)},$$

which is simplified and arranged to be

$$H(z) = \frac{0.1815 - 0.1815z^{-2}}{1 + 0.6264z^{-1} + 0.6396z^{-2}}.$$

(a) Program 8.9 lists the MATLAB details. Fig. 8.22 displays the frequency responses.

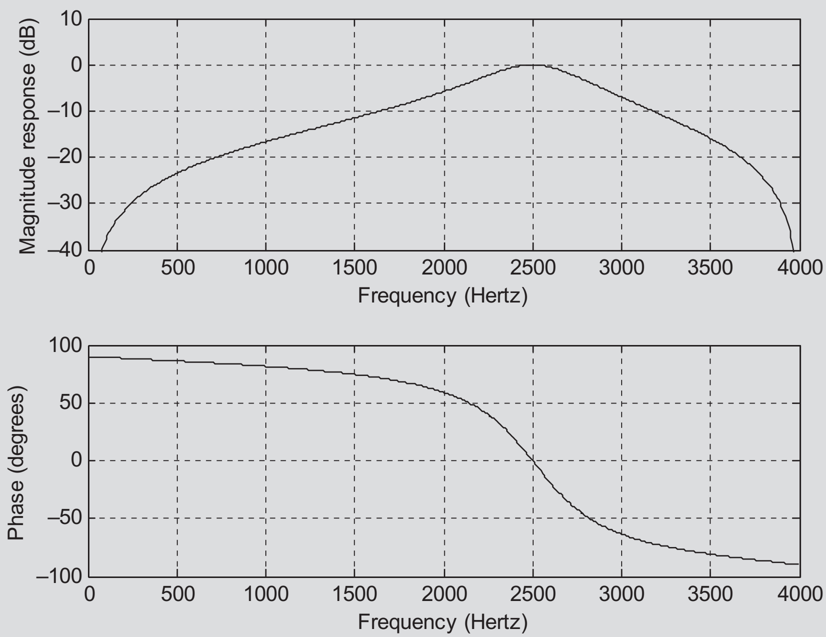

FIG. 8.22

Frequency responses of the designed digital filter for Example 8.13.

Program 8.9. MATLAB program for Example 8.13.

```
%Example 8.13
% Design of the digital bandpass Chebyshev filter
format long
fs=8000;
[B A]=lp2bp([2.8628][1 2.8628],sqrt(5.7341*10^8),4016) % Complete step 2
[b a]=bilinear(B,A,fs) % Complete step 3
% Plot the magnitude and phase responses |H(z)|
% b=[0.1815 0.0 -0.1815]; numerator coefficients from MATLAB
% a=[1 0.6264 0.6396]; denominator coefficients from MATLAB
freqz(b,a,512,fs);
axis([0 fs/2 -40 10])
```

8.4 HIGHER-ORDER INFINITE IMPULSE RESPONSE FILTER DESIGN USING THE CASCADE METHOD

For higher-order IIR filter design, use of a cascade transfer function is preferred. The factored forms for the lowpass prototype transfer functions for Butterworth and Chebyshev filters are provided in Tables 8.7–8.9. A Butterworth filter design example will be provided and the similar procedure can be adopted for Chebyshev filters.

Table 8.7 3 dB Butterworth Prototype Functions in the Cascade Form

n	$H_P(s)$
3	$\dfrac{1}{(s+1)(s^2+s+1)}$
4	$\dfrac{1}{(s^2+0.7654s+1)(s^2+1.8478s+1)}$
5	$\dfrac{1}{(s+1)(s^2+0.6180s+1)(s^2+1.6180s+1)}$
6	$\dfrac{1}{(s^2+0.5176s+1)(s^2+1.4142s+1)(s^2+1.9319s+1)}$

Table 8.8 Chebyshev Prototype Functions in the Cascade Form with 0.5 dB Ripple ($\varepsilon = 0.3493$)

n	$H_P(s)$ 0.5 dB Ripple ($\varepsilon = 0.3493$)
3	$\dfrac{0.7157}{(s+0.6265)(s^2+0.6265s+1.1425)}$
4	$\dfrac{0.3579}{(s^2+0.3507s+1.0635)(s^2+0.8467s+0.3564)}$
5	$\dfrac{0.1789}{(s+0.3623)(s^2+0.2239s+1.0358)(s^2+0.5862s+0.4768)}$
6	$\dfrac{0.0895}{(s^2+0.1553s+1.0230)(s^2+0.4243s+0.5900)(s^2+0.5796s+0.1570)}$

Table 8.9 Chebyshev Prototype Functions in the Cascade Form with 1 dB Ripple ($\varepsilon = 0.5088$)

n	$H_P(s)$ 1 dB Ripple ($\varepsilon = 0.5088$)
3	$\dfrac{0.4913}{(s+0.4942)(s^2+0.4942s+0.9942)}$
4	$\dfrac{0.2456}{(s^2+0.2791s+0.9865)(s^2+0.6737s+0.2794)}$
5	$\dfrac{0.1228}{(s+0.2895)(s^2+0.1789s+0.9883)(s^2+0.4684s+0.4293)}$
6	$\dfrac{0.0614}{(s^2+0.1244s+0.9907)(s^2+0.3398s+0.5577)(s^2+0.4641s+0.1247)}$

EXAMPLE 8.14

(a) Design a fourth-order digital lowpass Butterworth filter with a cutoff frequency of 2.5 kHz at a sampling frequency of 8000 Hz.
(b) Use MATLAB to plot the magnitude and phase responses.

Solution:

(a) First, we obtain the digital frequency in radians per second:

$$\omega_d = 2\pi f = 2\pi(2500) = 5000\pi \text{ (rad/s), and } T = \frac{1}{f_s} = \frac{1}{8000} \text{ (s)}.$$

Following the design steps, we compute the specifications for the analog filter.

1.

$$\omega_a = \frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right) = 16{,}000 \times \tan\left(\frac{5000\pi/8000}{2}\right) = 2.3946 \times 10^4 \text{ (rad/s)}$$

2. From Table 8.7, we have the fourth-order factored prototype transfer function as

$$H_P(s) = \frac{1}{(s^2 + 0.7654s + 1)(s^2 + 1.8478s + 1)}.$$

Applying the prototype transformation, we obtain

$$H(s) = H_P(s)\Big|_{\frac{s}{\omega_a}} = \frac{\omega_a^2 \times \omega_a^2}{\left(s^2 + 0.7654\omega_a s + \omega_a^2\right)\left(s^2 + 1.8478\omega_a s + \omega_a^2\right)}.$$

Substituting $\omega_a = 2.3946 \times 10^4$ (rad/s) yields

$$H(s) = \frac{\left(5.7340 \times 10^8\right) \times \left(5.7340 \times 10^8\right)}{\left(s^2 + 1.8328s + 5.7340 \times 10^8\right)\left(s^2 + 4.4247 \times 10^4 s + 5.7340 \times 10^8\right)}.$$

3. Hence, after applying BLT, we have

$$H(z) = \frac{\left(5.7340 \times 10^8\right) \times \left(5.7340 \times 10^8\right)}{\left(s^2 + 1.8328s + 5.7340 \times 10^8\right)\left(s^2 + 4.4247 \times 10^4 s + 5.7340 \times 10^8\right)}\Bigg|_{s=16{,}000(z-1)/(z+1)}.$$

Simplifying the algebra, we have the digital filter as

$$H(z) = \frac{0.5108 + 1.0215z^{-1} + 0.5108z^{-2}}{1 + 0.5654z^{-1} + 0.4776z^{-2}} \times \frac{0.3730 + 0.7460z^{-1} + 0.3730z^{-2}}{1 + 0.4129z^{-1} + 0.0790z^{-2}}.$$

(b) A MATLAB program is better to carry out the algebra and is listed in Program 8.10. Fig. 8.23 shows the filter magnitude and phase frequency responses.

Continued

EXAMPLE 8.14—CONT'D

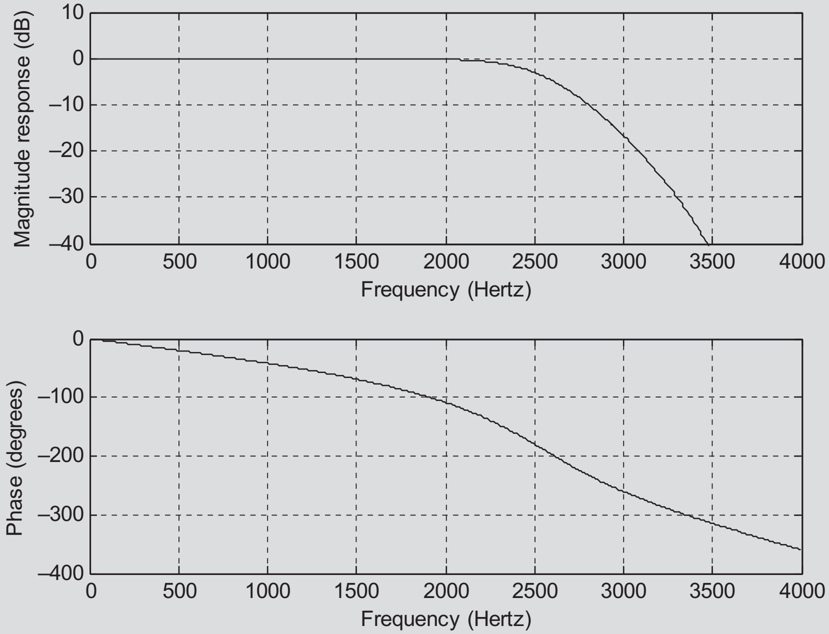

FIG. 8.23

Frequency responses of the designed digital filter for Example 8.14.

Program 8.10. MATLAB program for Example 8.14.

```
%Example 8.14
% Design of the fourth-order digital lowpass Butterworth filter
% in the cascade form
format long
fs=8000;   % Sampling rate
[B1 A1]=lp2lp([1][1 0.7654 1], 2.3946*10^4) % Complete step 2
[b1 a1]=bilinear(B1,A1,fs) % complete step 3
[B2 A2]=lp2lp([1][1 1.8478 1], 2.3946*10^4) % Complete step 2
[b2 a2]=bilinear(B2,A2,fs) % complete step 3
% Plot the magnitude and phase responses |H(z)|
% b1=[0.5108 1.0215 0.5108]; a1=[1 0.5654 0.4776]; coefficients from MATLAB
% b2=[0.3730 0.7460 0.3730]; a2=[1 0.4129 0.0790]; coefficients from MATLAB
freqz(conv(b1,b2),conv(a1,a2),512,fs); % Combined filter responses
axis([0 fs/2 -40 10]);
```

The higher-order bandpass, highpass, and bandstop filters using the cascade form can be designed similarly.

8.5 APPLICATION: DIGITAL AUDIO EQUALIZER

In this section, the design of a digital audio equalizer is introduced. For an audio application such as the CD player, the digital audio equalizer is used to make the sound as one desires by changing filter gains for different audio frequency bands. Other applications include adjusting the sound source to take room acoustics into account, removing undesired noise, and boosting the desired signal in the specified passband. The simulation is based on the consumer digital audio processor—such as CD player—handling the 16-bit digital samples with a sampling rate of 44.1 kHz and an audio signal bandwidth at 22.05 kHz. A block diagram of the digital audio equalizer is depicted in Fig. 8.24.

A seven-band audio equalizer is adopted for discussion. The center frequencies are listed in Table 8.10. The 3-dB bandwidth for each bandpass filter is chosen to be 50% of the center frequency. As shown in Fig. 8.24, g_0 through g_6 are the digital gains for each bandpass filter output and can be adjusted to make sound effects, while $y_0(n)$ through $y_6(n)$ are the digital amplified bandpass filter outputs. Finally, the equalized signal is the sum of the amplified bandpass filter outputs and itself. By changing the digital gains of the equalizer, many sound effects can be produced.

To complete the design and simulation, second-order IIR bandpass Butterworth filters are chosen for the audio equalizer, the coefficients are achieved using the BLT method, and are provided in Table 8.11.

The magnitude response for each filter bank is plotted in Fig. 8.25 for design verification. As shown in Fig. 8.25, after careful examination, the magnitude response of each filter band meets the design specification. We will perform simulation next.

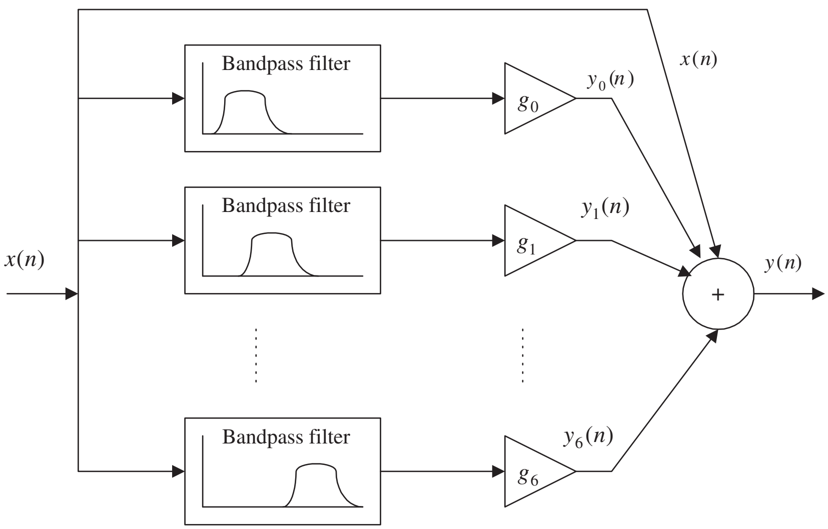

FIG. 8.24

Simplified block diagram of the audio equalizer.

Table 8.10 Specifications for an Audio Equalizer to be Designed

Center frequency (Hz)	100	200	400	1000	2500	6000	15,000
Bandwidth (Hz)	50	100	200	500	1250	3000	7500

Table 8.11 Designed Filter Banks

Filter Banks	Coefficients for the Numerator	Coefficients for the Denominator
Bandpass filter 0	0.0031954934, 0, –0.0031954934	1, –1.9934066716, 0.9936090132
Bandpass filter 1	0.0063708102, 0, –0.0063708102	1, –1.9864516324, 0.9872583796
Bandpass filter 2	0.0126623878, 0, –0.0126623878	1, –1.9714693192, 0.9746752244
Bandpass filter 3	0.0310900413, 0, –0.0310900413	1, –1.9181849043, 0.9378199174
Bandpass filter 4	0.0746111954, 0, –0.0746111954	1, –1.7346085867, 0.8507776092
Bandpass filter 5	0.1663862883, 0, –0.1663862884	1, –1.0942477187, 0.6672274233
Bandpass filter 6	0.3354404899, 0, –0.3354404899	1, 0.7131366534, 0.3291190202

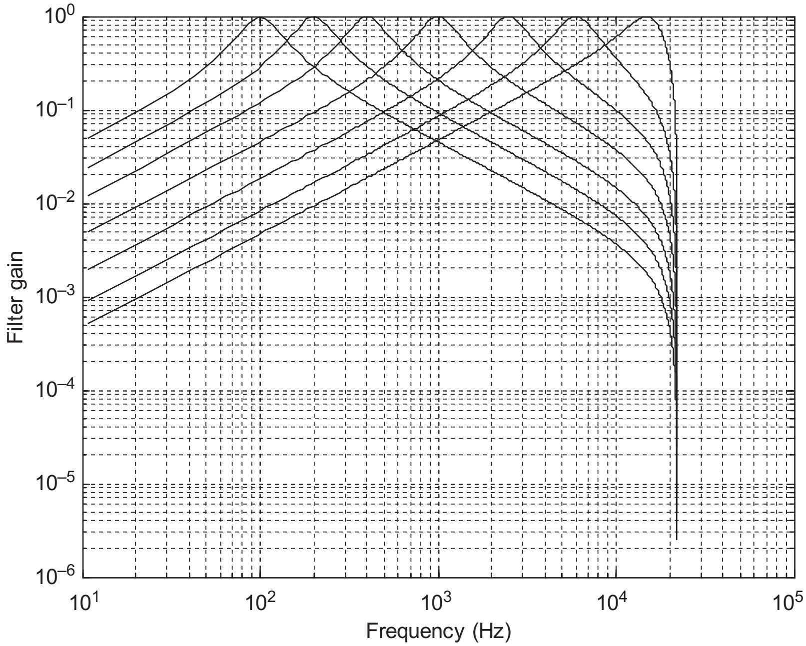

FIG. 8.25

Magnitude frequency responses for the audio equalizer.

Simulation in the MATLAB environment is based on the following setting. The audio test signal having frequency components of 100, 200, 400, 1000, 2500, 6000, and 15,000 Hz is generated from Eq. (8.36):

$$\begin{aligned} x(n) = {} & \sin(200\pi n/44,100) + \sin(400\pi n/44,100 + \pi/14) \\ & + \sin(800\pi n/44,100 + \pi/7) + \sin(2,000\pi n/44,100 + 3\pi/14) \\ & + \sin(5,000\pi n/44,100 + 2\pi/7) + \sin(12,000\pi n/44,100 + 5\pi/14) \\ & + \sin(30,000\pi n/44,100 + 3\pi/7). \end{aligned} \tag{8.36}$$

The gains set for the filter banks are as follows:

$$g_0 = 10,\ g_1 = 10,\ g_2 = 0,\ g_3 = 0,\ g_4 = 0,\ g_5 = 10,\ g_6 = 10.$$

After simulation, we note that the frequency components at 100, 200, 6000, and 15,000 Hz will be boosted by $20 \times \log_{10} 10 = 20$ (dB). The top plot in Fig. 8.26 shows the spectrum for the audio test signal, while the bottom plot depicts the spectrum for the equalized audio test signal. As shown in the plots, before audio digital equalization, the spectral peaks at all bands are at the same level; after audio digital equalization, the frequency components at bank 0, bank 1, bank 5, and bank 6 are amplified. Therefore, as we expected, the operation of the digital equalizer boosts the low-frequency and high-frequency components. The MATLAB list for the simulation is shown in Program 8.11.

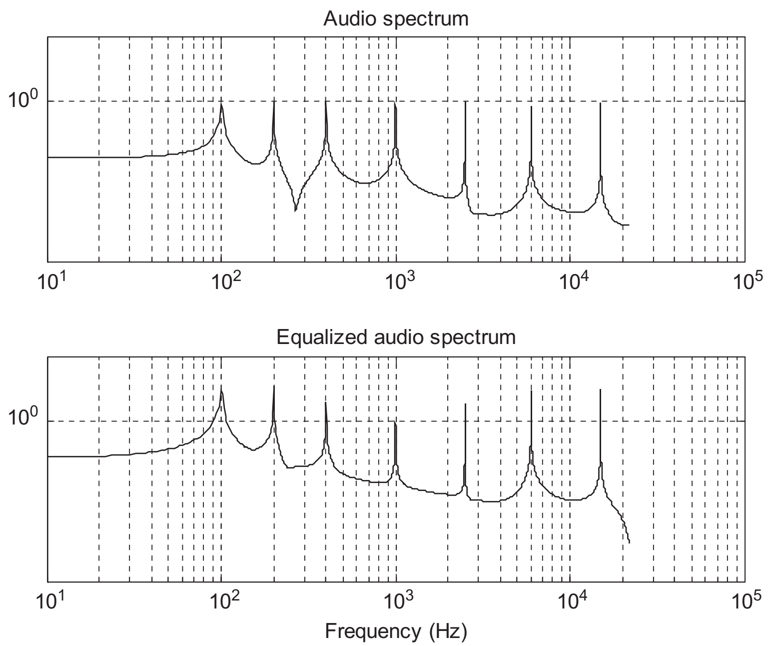

FIG. 8.26

Audio spectrum and equalized audio spectrum.

Program 8.11. MATLAB program for the digital audio equalizer.

```
close all; clear all
% Filter Coefficients (Butterworth type designed using the BLT)
B0=[0.0031954934 0 -0.0031954934]; A0=[1.0000000000 -1.9934066716 0.9936090132];
B1=[0.0063708102 0 -0.0063708102]; A1=[1.0000000000 -1.9864516324 0.9872583796];
B2=[0.0126623878 0 -0.0126623878]; A2=[1.0000000000 -1.9714693192 0.9746752244];
B3=[0.0310900413 0 -0.0310900413]; A3=[ 1.0000000000 -1.9181849043 0.9378199174];
B4=[ 0.0746111954 0.000000000 -0.0746111954];
A4=[1.0000000000 -1.7346085867 0.8507776092];
B5=[0.1663862883 0.0000000000 -0.1663862884];
A5=[1.0000000000 -1.0942477187 0.6672274233];
B6=[0.3354404899 0.0000000000 -0.3354404899];
A6=[1.0000000000 0.7131366534 0.3291190202];
[h0,f]=freqz(B0,A0,2048,44100);
[h1,f]=freqz(B1,A1,2048,44100);
[h2,f]=freqz(B2,A2,2048,44100);
[h3,f]=freqz(B3,A3,2048,44100);
[h4,f]=freqz(B4,A4,2048,44100);
[h5,f]=freqz(B5,A5,2048,44100);
[h6,f]=freqz(B6,A6,2048,44100);
loglog(f,abs(h0),f,abs(h1), f,abs(h2), ...
f,abs(h3),f,abs(h4),f,abs(h5),f,abs(h6));
xlabel('Frequency (Hz)');
ylabel('Filter Gain');grid
axis([10 10^5 10^(-6) 1]);
figure(2)
g0=10;g1=10;g2=0;g3=0;g4=0;g5=10;g6=10;
p0=0;p1=pi/14;p2=2*p1;p3=3*p1;p4=4*p1;p5=5*p1;p6=6*p1;
n=0:1:20480;                % Indices of samples
fs=44100;                   % Sampling rate
x=sin(2*pi*100*n/fs)+sin(2*pi*200*n/fs+p1)+...
 sin(2*pi*400*n/fs+p2)+sin(2*pi*1000*n/fs+p3)+...
 sin(2*pi*2500*n/fs+p4)+sin(2*pi*6000*n/fs+p5)+...
 sin(2*pi*15000*n/fs+p6); % Generate test audio signals
y0=filter(B0,A0,x);         % Bandpass filter 0
y1=filter(B1,A1,x);         % Bandpass filter 1
y2=filter(B2,A2,x);         % Bandpass filter 2
y3=filter(B3,A3,x);         % Bandpass filter 3
y4=filter(B4,A4,x);         % Bandpass filter 4
y5=filter(B5,A5,x);         % Bandpass filter 5
y6=filter(B6,A6,x);         % Bandpass filter 6
y=g0.*y0+g1.*y1+g2.*y2+g3.*y3+g4.*y4+g5.*y5+g6.*y6+x; % Equalizer output
N=length(x);
Axk=2*abs(fft(x))/N;Axk(1)=Axk(1)/2; % One-sided amplitude spectrum of the input
f=[0:N/2]*fs/N;
subplot(2,1,1);loglog(f,Axk(1:N/2+1));
title('Audio spectrum');
axis([10 100000 0.00001 100]);grid;
Ayk=2*abs(fft(y))/N; Ayk(1)=Ayk(1)/2; % One-sided amplitude spectrum of the output
subplot(2,1,2);loglog(f,Ayk(1:N/2+1));
xlabel('Frequency (Hz)');
title('Equalized audio spectrum');
axis([10 100000 0.00001 100]);grid;
```

8.6 IMPULSE INVARIANT DESIGN METHOD

We illustrate the concept of the impulse invariant design shown in Fig. 8.27. Given the transfer function of a designed analog filter, an analog impulse response can be easily found by the inverse Laplace transform of the transfer function. To replace the analog filter by equivalent digital filter, we apply an approximation in time domain in which the digital filter impulse response must be equivalent to the analog impulse response. Therefore, we can sample the analog impulse response to get the digital impulse response, and take the z-transform of the sampled analog impulse response to obtain the transfer function of the digital filter.

The analog impulse response can be achieved by taking the inverse Laplace transform of the analog filter $H(s)$, that is,

$$h(t) = L^{-1}(H(s)). \tag{8.37}$$

Now, if we sample the analog impulse response with a sampling interval of T and use T as a scale factor, it follows that

$$T \times h(n) = T \times h(t)|_{t=nT}, n \geq 0. \tag{8.38}$$

Taking the z-transform on both sides of Eq. (8.38) yields the digital filter as

$$H(z) = Z[T \times h(n)]. \tag{8.39}$$

The effect of the scale factor T in Eq. (8.38) can be explained as follows. We approximate the area under the curve specified by the analog impulse function $h(t)$ using a digital sum given by

$$\text{Area} = \int_0^\infty h(t)dt \approx T \times h(0) + T \times h(1) + T \times h(2) + \cdots. \tag{8.40}$$

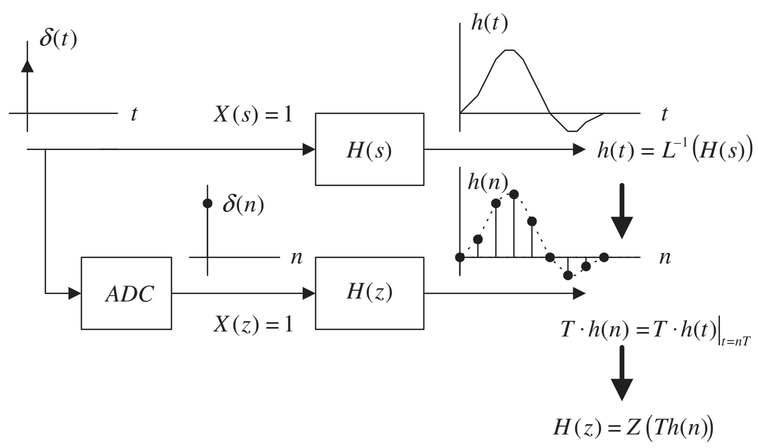

FIG. 8.27

Impulse invariant design method.

Note that the area under the curve indicates the DC gain of the analog filter while the digital sum in Eq. (8.40) is the DC gain of the digital filter.

The rectangular approximation is used, since each sample amplitude is multiplied by the sampling interval T. Due to the interval size for approximation in practice, we cannot guarantee that the digital sum has exactly the same value as the one from the integration unless the sampling interval T in Eq. (8.40) approaches zero. This means that the higher the sampling rate, that is, the smaller the sampling interval, the more accurately the digital filter gain matches the analog filter gain. Hence, in practice, we need to further apply gain scaling for adjustment if it is a requirement. We look at the following examples.

EXAMPLE 8.15

Consider the following Laplace transfer function:

$$H(s) = \frac{2}{s+2}.$$

(a) Determine $H(z)$ using the impulse invariant method if the sampling rate $f_s = 10\,(\text{Hz})$.
(b) Use MATLAB to plot

1. The magnitude response $|H(f)|$ and phase response $\phi(f)$ with respect to $H(s)$ for the frequency range from 0 to $f_s/2\,(\text{Hz})$.
2. The magnitude response $|H(e^{j\Omega})| = |H(e^{j2\pi fT})|$ and phase response $\phi(f)$ with respect to $H(z)$ for the frequency range from 0 to $f_s/2\,(\text{Hz})$.

Solution:

(a) Taking the inverse Laplace transform the analog transfer function, the impulse response therefore is found to be

$$h(t) = L^{-1}\left[\frac{2}{s+2}\right] = 2e^{-2t}u(t).$$

Sampling the impulse response $h(t)$ with $T = 1/f_s = 0.1\,(\text{s})$, we have

$$Th(n) = T2e^{-2nT}u(n) = 0.2e^{-0.2n}u(n).$$

Using the z-transform table in Chapter 5, we have

$$Z[e^{-an}u(n)] = \frac{z}{z - e^{-a}}.$$

and noting that $e^{-a} = e^{-0.2} = 0.8187$, the digital filter transfer function $H(z)$ is finally given by

$$H(z) = \frac{0.2z}{z - 0.8187} = \frac{0.2}{1 - 0.8187z^{-1}}.$$

(b) The MATLAB list is given in Program 8.12. The first and third plots in Fig. 8.28 show comparisons of the magnitude and phase frequency responses. The shape of the magnitude response (first plot) closely matches that of the analog filter, while the phase response (third plot) differs from the analog phase response in this example.

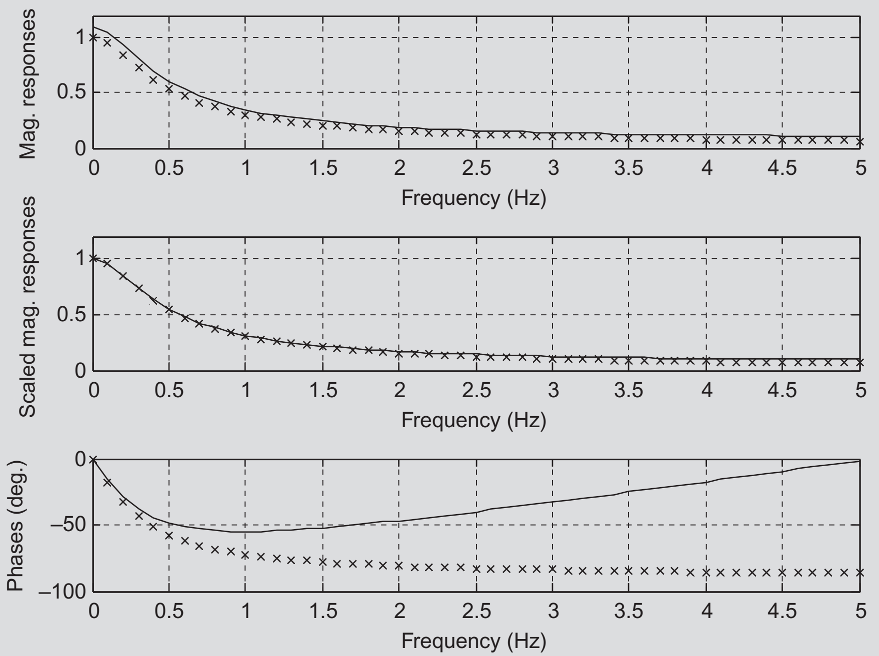

FIG. 8.28

Frequency responses. Line of *x*'s, frequency responses of the analog filter; solid line, frequency responses of the designed digital filter.

Program 8.12. MATLAB program for Example 8.15.

```
%Example 8.15.
% Plot the magnitude responses |H(s)| and |H(z)|
% For the Laplace transfer function H(s)
f=0:0.1:5;T=0.1            % Frequency range and sampling interval
w=2*pi*f;                  % Frequency range in rad/sec
hs=freqs([2], [1 2],w);    % Analog frequency response
phis=180*angle(hs)/pi;
% For the z-transfer function H(z)
hz=freqz([0.2][1 -0.8187],length(w));  % Digital frequency response
hz_scale=freqz([0.1813][1 -0.8187],length(w)); % Scaled digital mag. response
 phiz=180*angle(hz)/pi;
 %Plot magnitude and phase responses.
subplot(3,1,1), plot(f,abs(hs),'kx',f, abs(hz),'k-'),grid; axis([0 5 0 1.2]);
xlabel('Frequency (Hz)'); ylabel('Mag. Responses')
subplot(3,1,2), plot(f,abs(hs),'kx',f, abs(hz_scale),'k-'),grid; axis([0 5 0 1.2]);
xlabel('Frequency (Hz)'); ylabel('Scaled Mag. Responses')
subplot(3,1,3), plot(f,phis,'kx',f, phiz,'k-'); grid;
xlabel('Frequency (Hz)'); ylabel('Phases (deg.)');
```

The filter DC gain is given by

$$H\left(e^{j\Omega}\right)\big|_{\Omega=0} = H(1) = 1.1031.$$

we can further scale the filter to have a unit gain of

$$H(z) = \frac{1}{1.1031}\frac{0.2}{1-0.8187z^{-1}} = \frac{0.1813}{1-0.8187z^{-1}}.$$

The scaled magnitude frequency response is shown in the middle plot along with that of analog filter in Fig. 8.28, where the magnitudes are matched very well below 1.8 Hz.

Example 8.15 demonstrates the design procedure using the impulse invariant design. The filter performance depends on the sampling interval (Lynn and Fuerst, 1999). As shown in Fig. 8.27, the analog impulse response $h(t)$ is not a band-limited signal whose frequency components are generally larger than the Nyquist limit (folding frequency); hence, sampling $h(t)$ could cause aliasing. Fig. 8.29A shows the analog impulse response $Th(t)$ in Example 8.15 and its sampled version $Th(nT)$, where the sampling

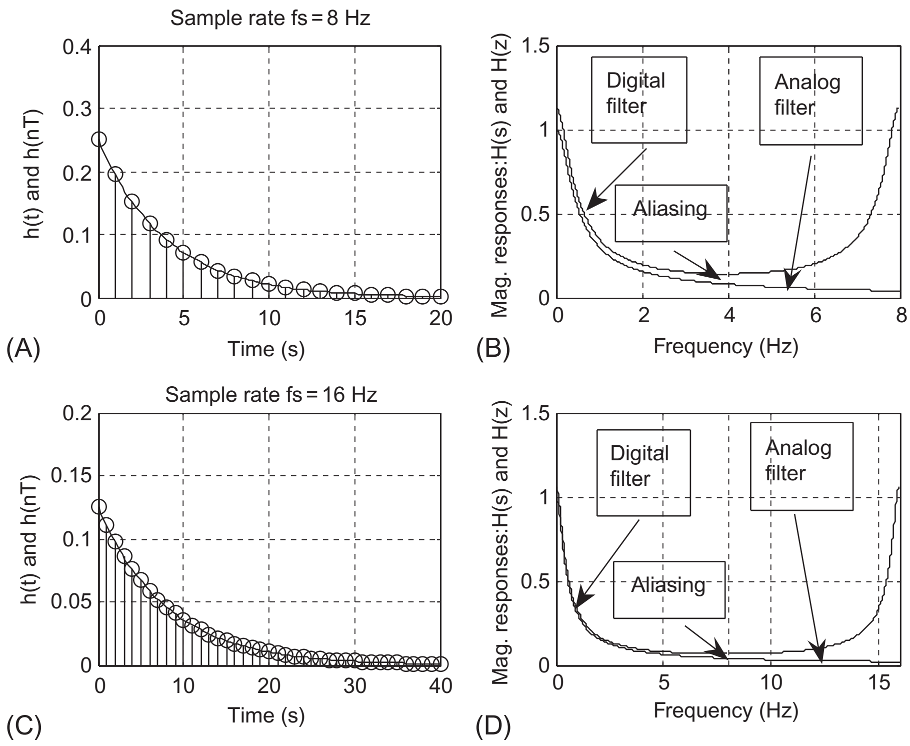

FIG. 8.29

Sampling interval effect in the impulse invariant IIR filter design. (A) Analog impulse response and its sampled version at a sampling rate of 8 Hz. (B) Magnitude frequency responses of the analog filter and the digital filter at a sampling rate of 8 Hz. (C) Analog impulse response and its sampled version at a sample rate of 16 Hz. (D) Magnitude frequency responses of the analog filter and the digital filter at a sampling rate of 16 Hz.

interval is 0.125 s. The magnitude frequency responses of the analog and digital filters are plotted in Fig. 8.29B. The aliasing occurs, since the impulse response contains the frequency components beyond the Nyquist limit, that is, 4 Hz, in this case. Furthermore, using lower sampling rate of 8 Hz causes less accuracy in the magnitude frequency response of the digital filter, so more aliasing develops.

Fig. 8.29C shows the analog impulse response and its sampled version using a higher sampling rate of 16 Hz. Fig. 8.29D displays the more accurate magnitude frequency response of the digital filter. Hence, we can obtain a reduced aliasing level. Note that the aliasing cannot be avoided, due to sampling of the analog impulse response. The only way to reduce the aliasing is to use a higher sampling frequency or design a filter with a very low cutoff frequency to reduce the aliasing to a minimum level.

Investigation of the sampling interval effect leads us to the following conclusions. Note that the analog impulse response for a highpass filter or a bandstop filter contains frequency components at the maximum level at the Nyquist limit (folding frequency), even assuming that the sampling rate is much higher than the cutoff frequency of a highpass filter or the upper cutoff frequency of a bandstop filter. Hence, sampling the analog impulse response always produces the maximum aliasing level. Without using an additional anti-aliasing filter or the advanced method (Nelatury, 2007) employing the frequency prewarping technique, the impulse invariant method alone are not suggested for designing the highpass filter or bandstop filter.

Instead, in practice, we should apply the BLT design method. The impulse invariant design method is only appropriate for designing a lowpass filter or bandpass filter with a sampling rate much larger than the lower cutoff frequency of the lowpass filter or the upper cutoff frequency of the bandpass filter.

Next, let us focus on the second-order filter design via Example 8.16.

EXAMPLE 8.16

Consider the following Laplace transfer function:

$$H(s) = \frac{s}{s^2 + 2s + 5}.$$

(a) Determine $H(z)$ using the impulse invariant method if the sampling rate $f_s = 10$ Hz.

(b) Use MATLAB to plot:

1. The magnitude response $|H(f)|$ and phase response $\phi(f)$ with respect to $H(s)$ for the frequency range from 0 to $f_s/2$ (Hz).
2. The magnitude response $|H(e^{j\Omega})| = |H(e^{j2\pi fT})|$ and phase response $\phi(f)$ with respect to $H(z)$ for the frequency range from 0 to $f_s/2$ (Hz).

Solution:

(a) Since $H(s)$ has complex poles located at $s = -1 \pm 2j$, we can write it in a quadratic form as

$$H(s) = \frac{s}{s^2 + 2s + 5} = \frac{s}{(s+1)^2 + 2^2}.$$

We can further write the transfer function as

$$H(s) = \frac{(s+1) - 1}{(s+1)^2 + 2^2} = \frac{(s+1)}{(s+1)^2 + 2^2} - 0.5 \times \frac{2}{(s+1)^2 + 2^2}.$$

Continued

EXAMPLE 8.16—CONT'D

From the Laplace transform table (Appendix B), the analog impulse response can easily be found as

$$h(t) = e^{-t}\cos(2t)u(t) - 0.5e^{-t}\sin(2t)u(t).$$

Sampling the impulse response $h(t)$ using a sampling interval $T = 0.1$ and using the scale factor of $T = 0.1$, we have

$$Th(n) = Th(t)|_{t=nT} = 0.1e^{-0.1n}\cos(0.2n)u(n) - 0.05e^{-0.1n}\sin(0.2n)u(n).$$

Applying the z-transform (Chapter 5) leads to

$$\begin{aligned} H(z) &= Z[0.1e^{-0.1n}\cos(0.2n)u(n) - 0.05e^{-0.1n}\sin(0.2n)u(n)] \\ &= \frac{0.1z(z - e^{-0.1}\cos(0.2))}{z^2 - 2e^{-0.1}\cos(0.2)z + e^{-0.2}} - \frac{0.05e^{-0.1}\sin(0.2)z}{z^2 - 2e^{-0.1}\cos(0.2)z + e^{-0.2}}. \end{aligned}$$

After algebraic simplification, we obtain the second-order digital filter as

$$H(z) = \frac{0.1 - 0.09767z^{-1}}{1 - 1.7735z^{-1} + 0.8187z^{-2}}.$$

(b) The magnitude and phase frequency responses are shown in Fig. 8.30 and MATLAB Program 8.13 is given. The passband gain of the digital filter is higher than that of the analog filter, but their shapes are same.

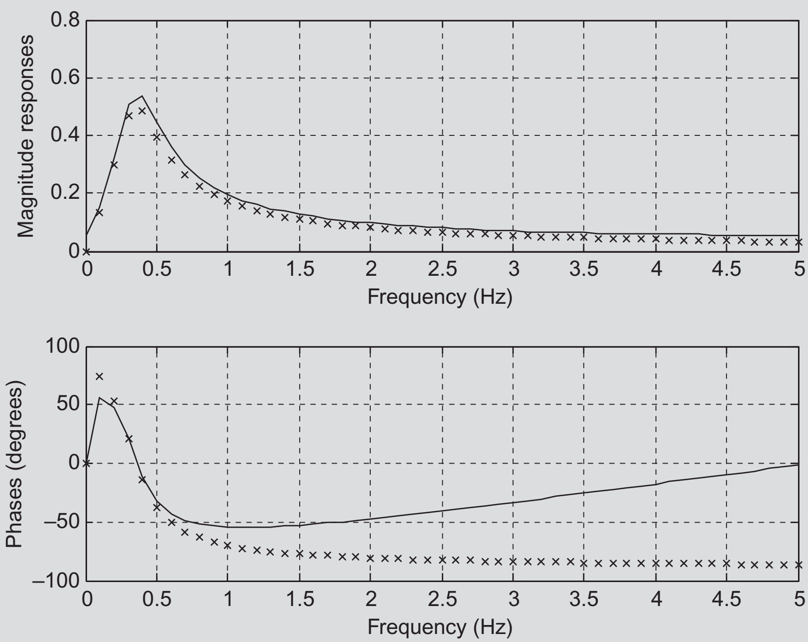

FIG. 8.30

Frequency responses. Line of x's, frequency responses of the analog filter; solid line, frequency responses of the designed digital filter.

Program 8.13. MATLAB program for Example 8.16.

```
%Example 8.16
% Plot the magnitude responses |H(s)| and |H(z)|
% For the Laplace transfer function H(s)
f=0:0.1:5;T=0.1;  % Initialize analog frequency range in Hz and sampling interval
w=2*pi*f;         % Convert the frequency range to radians/second
hs=freqs([1 0], [1 2 5],w);    % Calculate analog filter frequency responses
phis=180*angle(hs)/pi;
% For the z-transfer function H(z)
% Calculate digital filter frequency responses
hz=freqz([0.1 -0.09766][1 -1.7735 0.8187],length(w));
 phiz=180*angle(hz)/pi;
 % Plot magnitude and phase responses
subplot(2,1,1), plot(f,abs(hs),'x',f, abs(hz),'-'),grid;
xlabel('Frequency (Hz)'); ylabel('Magnitude Responses')
subplot(2,1,2), plot(f,phis,'x',f, phiz,'-'); grid;
xlabel('Frequency (Hz)'); ylabel('Phases (degrees)')
```

8.7 POLE-ZERO PLACEMENT METHOD FOR SIMPLE INFINITE IMPULSE RESPONSE FILTERS

This section introduces a pole-zero placement method for a simple IIR filter design. Let us first examine the effects of the pole-zero placement on the magnitude response in the z-plane shown in Fig. 8.31.

In the z-plane, when we place a pair of complex conjugate zeros at a given point on the unit circle with an angle θ (usually we do), we will have a numerator factor of $(z-e^{j\theta})(z-e^{-j\theta})$ in the transfer function. Its magnitude contribution to the frequency response at $z=e^{j\Omega}$ is $(e^{j\Omega}-e^{j\theta})(e^{j\Omega}-e^{-j\theta})$. When $\Omega=\theta$,

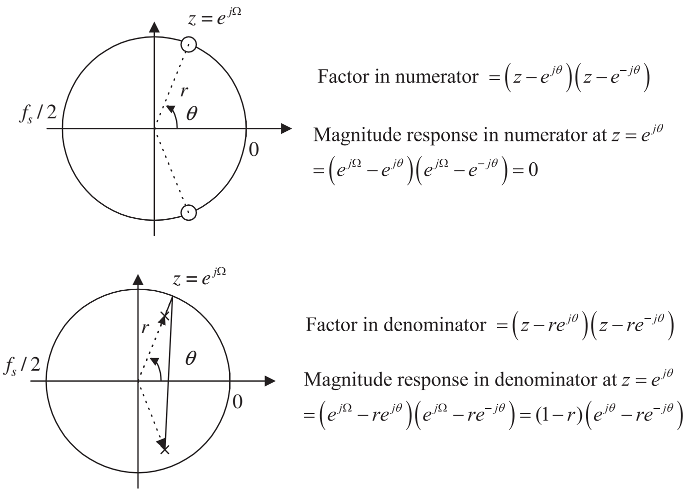

FIG. 8.31

Effects of the pole-zero placement on the magnitude response.

the magnitude will reach zero, since the first factor $(e^{j\theta} - e^{j\theta}) = 0$ contributes zero magnitude. When a pair of complex conjugate poles are placed at a given point within the unit circle, we have a denominator factor of $(z - re^{j\theta})(z - re^{-j\theta})$, where r is the radius chosen to be less than and close to 1 to place the poles inside the unit circle. The magnitude contribution to the frequency response at $\Omega = \theta$ will rise to a large magnitude, since the first factor $(e^{j\theta} - re^{j\theta}) = (1 - r)e^{j\theta}$ gives a small magnitude of $1 - r$, which is the length between the pole and the unit circle at the angle $\Omega = \theta$. Note that the magnitude of $e^{j\theta}$ is 1.

Therefore, we can reduce the magnitude response using zero placement, while we increase the magnitude response using pole placement. Placing a combination of poles and zeros will result in different frequency responses, such as lowpass, highpass, bandpass, and bandstop. The method is intuitive and approximate. However, it is easy to compute filter coefficients for simple IIR filters. Here, we describe the design procedures for second-order bandpass and bandstop filters, as well first-order lowpass and highpass filters. Practically, the pole-zero placement method has a good performance when the bandpass and bandstop filters have very narrow bandwidth requirements and the lowpass and highpass filters have either very low cutoff frequency close to DC or very high cutoff frequency close to the folding frequency (Nyquist limit).

8.7.1 SECOND-ORDER BANDPASS FILTER DESIGN

Typical pairs of poles and zeros for a bandpass filter are placed in Fig. 8.32. Poles are complex conjugate, with the magnitude r controlling the bandwidth and the angle θ controlling the center frequency. The zeros are placed at $z = 1$ corresponding to DC, and at $z = -1$ corresponding to the folding frequency.

The poles will raise the magnitude response at the center frequency while the zeros will cause zero gains at DC (zero frequency) and at the folding frequency.

From Fig. 8.32A, we see that the angles for pole locations are $\pm\theta$. The angle can be determined by

$$\theta = \left(\frac{f_0}{f_s}\right) \times 360^\circ \tag{8.41}$$

Fig. 8.32B describes how the radius of complex conjugate poles is determined.

As shown in Fig. 8.32B, the magnitude at the center frequency can be approximated by

$$\left|H\left(e^{j\theta}\right)\right| = \frac{ab}{(1-r)r_1}$$

where a and b are the distances from $z = e^{j\theta}$ to the locations of zeros while $(1 - r)$ and r_1 are the distances from $z = e^{j\theta}$ to the locations of two complex conjugate poles, respectively. r is the desired radius to be determined. Now, we can approximate the magnitude at the 3-dB cutoff frequency as

$$\left|H\left(e^{j\theta_{3dB}}\right)\right| = \frac{a'b'}{r_2 r_3}$$

where a' and b' are the distances from $z = e^{j\theta_{3dB}}$ to the locations of zeros; and r_2 and r_3 are the distances from $z = e^{j\theta_{3dB}}$ to the locations of two complex conjugate poles, respectively. For r is close to 1, assuming that $a \approx a'$, $b \approx b'$, $r_1 \approx r_2$, and $\left|H\left(e^{j\theta_{3dB}}\right)\right| / \left|H\left(e^{j\theta}\right)\right| = 1/\sqrt{2}$ due to 3-dB magnitude attenuation, we obtain the relation $r_3 = \sqrt{2(1-r)}$, which indicates 45° between r_3 and 1–r. The bandwidth is then approximated to 2(1–r). Normalizing BW_{3dB} in Hz to radians, it follows that

$$2\pi \times BW_{3dB}T = 2(1 - r).$$

Solving this equation, we obtain the desired radius for complex conjugate poles, that is,

$$r = 1 - \pi \times \frac{BW_{3dB}}{f_s} \tag{8.42}$$

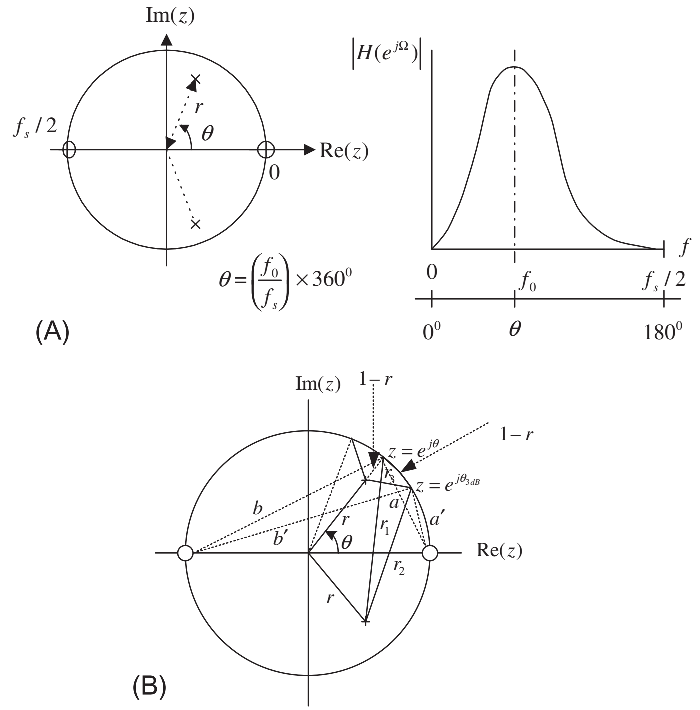

FIG. 8.32

(A) Pole-zero placement for a second-order narrow bandpass filter. (B) Radius of the complex conjugate poles for a second-order narrow bandpass filter.

The design equations for a bandpass filter using pole-zero placement are summarized as

$$\theta = \left(\frac{f_0}{f_s}\right) \times 360^\circ \tag{8.43}$$

$$r \approx 1 - (BW_{3dB}/f_s) \times \pi, \text{ good for } 0.9 \le r < 1 \tag{8.44}$$

$$H(z) = \frac{K(z-1)(z+1)}{(z - re^{j\theta})(z - re^{-j\theta})} = \frac{K(z^2-1)}{(z^2 - 2rz\cos\theta + r^2)}, \tag{8.45}$$

where K is a scale factor to adjust the bandpass filter to have a unit passband gain given by

$$K = \frac{(1-r)\sqrt{1 - 2r\cos 2\theta + r^2}}{2|\sin\theta|}. \tag{8.46}$$

EXAMPLE 8.17

A second-order bandpass filter is required to satisfy the following specifications:

- Sampling rate = 8000 Hz
- 3-dB bandwidth: $BW = 200$ Hz
- Narrow passband centered at $f_0 = 1000$ Hz

zero gain at 0 and 4000 Hz

Continued

EXAMPLE 8.17—CONT'D

Find the transfer function using the pole-zero placement method.

Solution:

First, we calculate the required magnitude of the poles.

$$r = 1 - \frac{200}{8000} \times \pi = 0.9215,$$

which is a good approximation. Use the center frequency to obtain the angle of the pole location:

$$\theta = \left(\frac{1000}{8000}\right) \times 360 = 45^\circ.$$

Compute the unit-gain scale factor as

$$K = \frac{(1 - 0.9215)\sqrt{1 - 2 \times 0.9215 \times \cos 2 \times 45^\circ + 0.9215^2}}{2\left|\sin 45^\circ\right|} = 0.0755$$

Finally, the transfer function is given by

$$H(z) = \frac{0.0755(z^2 - 1)}{\left(z^2 - 2 \times 0.9215 z \cos 45^\circ + 0.9215^2\right)} = \frac{0.0755 - 0.0755 z^{-2}}{1 - 1.3031 z^{-1} + 0.8491 z^{-2}}.$$

8.7.2 SECOND-ORDER BANDSTOP (NOTCH) FILTER DESIGN

For this type of filter, the pole placement is the same as the bandpass filter (Fig. 8.33). The zeros are placed on the unit circle with the same angles with respect to poles. This will improve passband performance. The magnitude and the angle of the complex conjugate poles determine the 3-dB bandwidth and center frequency, respectively.

Through a similar derivation, the design formulas for bandstop filters can be obtained via the following equations:

$$\theta = \left(\frac{f_0}{f_s}\right) \times 360^\circ \tag{8.47}$$

$$r \approx 1 - (BW_{3dB}/f_s) \times \pi, \text{ good for } 0.9 \le r < 1 \tag{8.48}$$

$$H(z) = \frac{K\left(z - e^{j\theta}\right)\left(z + e^{-j\theta}\right)}{(z - re^{j\theta})(z - re^{-j\theta})} = \frac{K(z^2 - 2z\cos\theta + 1)}{(z^2 - 2rz\cos\theta + r^2)}. \tag{8.49}$$

The scale factor to adjust the bandstop filter to have a unit passband gain is given by

$$K = \frac{(1 - 2r\cos\theta + r^2)}{(2 - 2\cos\theta)}. \tag{8.50}$$

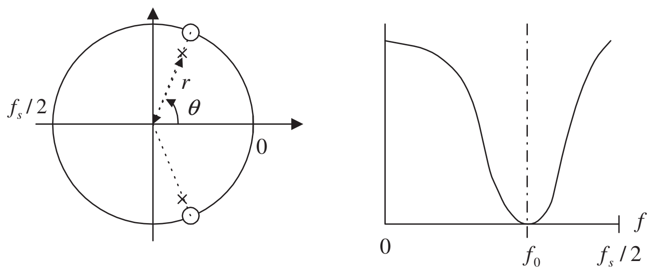

FIG. 8.33

Pole-zero placement for a second-order notch filter.

EXAMPLE 8.18

A second-order notch filter is required to satisfy the following specifications:

- sampling rate $= 8000\,\text{Hz}$
- 3-dB bandwidth: $BW = 100\,\text{Hz}$
- narrow passband centered at $f_0 = 1500\,\text{Hz}$

Find the transfer function using the pole-zero placement approach.

Solution:

We first calculate the required magnitude of the poles

$$r \approx 1 - \left(\frac{100}{8000}\right) \times \pi = 0.9607,$$

which is a good approximation. We use the center frequency to obtain the angle of the pole location:

$$\theta = \left(\frac{1500}{8000}\right) \times 360^\circ = 67.5^\circ.$$

The unit-gain scale factor is calculated as

$$K = \frac{\left(1 - 2 \times 0.9607 \cos 67.5^\circ + 0.9607^2\right)}{\left(2 - 2\cos 67.5^\circ\right)} = 0.9620.$$

Finally, we obtain the transfer function:

$$H(z) = \frac{0.9620\left(z^2 - 2z\cos 67.5^\circ + 1\right)}{\left(z^2 - 2 \times 0.9607 z \cos 67.5^\circ + 0.9607^2\right)} = \frac{0.9620 - 0.7363z^{-1} + 0.9620z^{-2}}{1 - 0.7353z^{-1} + 0.9229}.$$

8.7.3 FIRST-ORDER LOWPASS FILTER DESIGN

The first-order pole-zero placement can be operated in two cases. The first situation is when the cutoff frequency is less than $f_s/4$. Then the pole-zero placement is as shown in Fig. 8.34A.

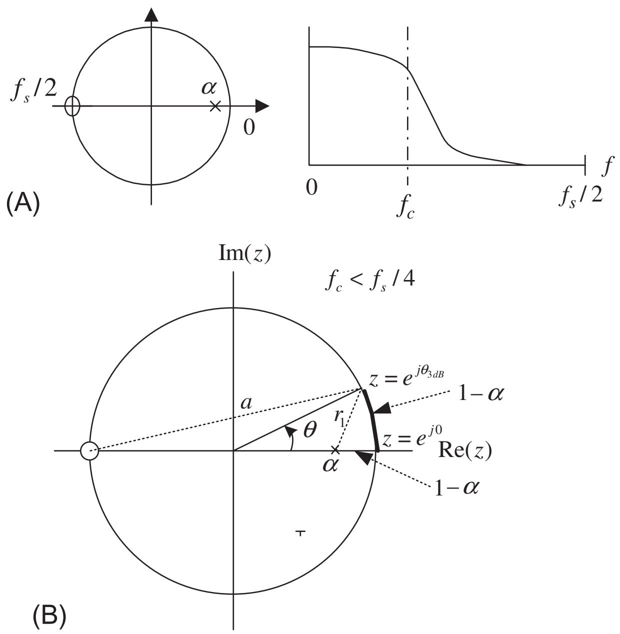

FIG. 8.34

(A) Pole-zero placement for the first-order lowpass filter with $f_c < f_s/4$. (B) Determination of pole location for the first-order lowpass filter with $f_c < f_s/4$.

As shown in Fig. 8.34A, the pole $z = \alpha$ is placed in real axis. The zero is placed at $z = -1$ to ensure zero gain at the folding frequency (Nyquist limit). Fig. 8.34B illustrates how to determine α.

As depicted in Fig. 8.34B, the DC magnitude response is given by

$$\left|H\left(e^{j0}\right)\right| = \frac{\text{distance from } e^{j0} \text{ to } z = -1}{\text{distance from } e^{j0} \text{ to } z = \alpha} = \frac{2}{1-\alpha}$$

Magnitude at the 3-dB cutoff frequency can be approximated by

$$\left|H\left(e^{j\theta_{3dB}}\right)\right| = \frac{\text{distance from } e^{j\theta_{3dB}} \text{ to } z = -1}{\text{distance from } e^{j\theta_{3dB}} \text{ to } z = \alpha} = \frac{a}{r_1}$$

For r is close to 1, we assume $a \approx 2$. Then $\left|H\left(e^{j\theta_{3dB}}\right)\right| / \left|H\left(e^{j\theta}\right)\right| = 1/\sqrt{2}$, that is, $r_1 \approx \sqrt{2}(1-\alpha)$, which indicates 45° between r_1 and Re(z) axis. We can approximate the bandwidth can be $(1-\alpha)$. Normalizing the bandwidth f_c in Hz to radians, we have

$$2\pi \times f_c T = 1 - \alpha$$

Therefore, the pole location can be solved as

$$\alpha = 1 - 2\pi \times \frac{f_c}{f_s}$$

When the cutoff frequency is above $f_s/4$, the pole-zero placement is adopted as shown in Fig. 8.35. The similar derivation can be performed to achieve the pole location.

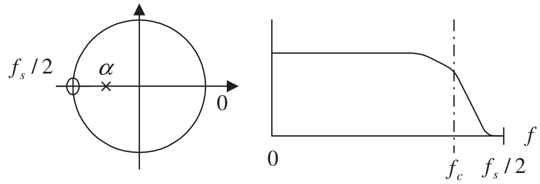

FIG. 8.35

Pole-zero placement for the first-order lowpass filter with $f_c > f_s/4$.

Design formulas for lowpass filters using the pole-zero placement are given in the following equations:

$$\text{When } f_c < f_s/4, \alpha \approx 1 - 2 \times (f_c/f_s) \times \pi, \text{ good for } 0.9 \leq \alpha < 1 \tag{8.51}$$

$$\text{When } f_c > f_s/4, \alpha \approx -(1 - \pi + 2 \times (f_c/f_s) \times \pi), \text{ good for } -1 < \alpha \leq -0.9 \tag{8.52}$$

The transfer function is

$$H(z) = \frac{K(z+1)}{(z-\alpha)}, \tag{8.53}$$

and the unit passband gain scale factor is given by

$$K = \frac{(1-\alpha)}{2}. \tag{8.54}$$

EXAMPLE 8.19

A first-order lowpass filter is required to satisfy the following specifications:

- sampling rate = 8000 Hz
- 3-dB cutoff frequency: $f_c = 100$ Hz
- zero gain at 4000 Hz

Find the transfer function using the pole-zero placement method.

Solution:

Since the cutoff frequency of 100 Hz is much less than $f_s/4 = 2000$ Hz, we determine the pole as

$$\alpha \approx 1 - 2 \times \left(\frac{100}{8000}\right) \times \pi = 0.9215,$$

which is above 0.9. Hence, we have a good approximation. The unit-gain scale factor is calculated by

$$K = \frac{(1 - 0.9215)}{2} = 0.03925.$$

Last, we can develop the transfer function as

$$H(z) = \frac{0.03925(z+1)}{(z-0.9215)} = \frac{0.03925 + 0.03925z^{-1}}{1 - 0.9215z^{-1}}.$$

Continued

EXAMPLE 8.19—CONT'D

Note that we can also determine the unit-gain factor K by substituting $z = e^{j0} = 1$ to the transfer function $H(z) = \frac{(z+1)}{(z-\alpha)}$, then find a DC gain. Set the scale factor to be a reciprocal of the DC gain. This can be easily done, that is,

$$\text{DCgain} = \left.\frac{z+1}{z-0.9215}\right|_{z=1} = \frac{1+1}{1-0.9215} = 25.4777.$$

Hence, $K = 1/25.4777 = 0.03925$.

8.7.4 FIRST-ORDER HIGHPASS FILTER DESIGN

Similar to the lowpass filter design, the pole-zero placements for the first-order highpass filters in two cases are shown in Fig. 8.36A and B.

Formulas for designing highpass filters using the pole-zero placement are listed in the following equations:

$$\text{When } f_c < f_s/4, \alpha \approx 1 - 2 \times (f_c/f_s) \times \pi, \text{ good for } 0.9 \le \alpha < 1. \tag{8.55}$$

$$\text{When } f_c > f_s/4, \alpha \approx -(1 - \pi + 2 \times (f_c/f_s) \times \pi), \text{ good for } -1 < \alpha \le -0.9. \tag{8.56}$$

$$H(z) = \frac{K(z-1)}{(z-\alpha)} \tag{8.57}$$

$$K = \frac{(1+\alpha)}{2}. \tag{8.58}$$

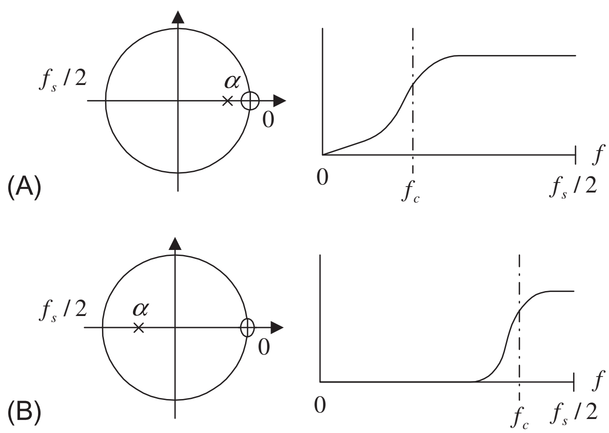

FIG. 8.36

(A) Pole-zero placement for the first-order highpass filter with $f_c < f_s/4$. (B) Pole-zero placement for the first-order highpass filter with $f_c > f_s/4$.

EXAMPLE 8.20

A first-order highpass filter is required to satisfy the following specifications:

- sampling rate = 8000 Hz
- 3-dB cutoff frequency: $f_c = 3800$ Hz
- zero gain at 0 Hz

Find the transfer function using the pole-zero placement method.

Solution:

Since the cutoff frequency of 3800 Hz is much larger than $f_s/4 = 2000$ Hz, we determine the pole as

$$\alpha \approx -\left(1 - \pi + 2 \times \left(\frac{3800}{8000}\right) \times \pi\right) = -0.8429,$$

The unit-gain scale factor and transfer function are obtained as

$$K = \frac{(1 - 0.8429)}{2} = 0.07854$$

$$H(z) = \frac{0.07854(z-1)}{(z+0.8429)} = \frac{0.07854 - 0.07854z^{-1}}{1 + 0.8429z^{-1}}.$$

Note that we can also determine the unit-gain scale factor K by substituting $z = e^{j180^0} = -1$ to the transfer function $H(z) = \frac{(z-1)}{(z-\alpha)}$, finding a passband gain at the Nyquist limit $f_s/2 = 4000$ Hz. We then set the scale factor to be a reciprocal of the passband gain, that is,

$$\text{Passband gain} = \left.\frac{z-1}{z+0.8429}\right|_{z=1} = \frac{-1-1}{-1+0.8429} = 12.7307.$$

Hence, $K = 1/12.7307 = 0.07854$.

8.8 REALIZATION STRUCTURES OF INFINITE IMPULSE RESPONSE FILTERS

In this section, we will realize the designed IIR filter using direct-form I as well as direct-form II. We will then realize a higher-order IIR filter using a cascade form.

8.8.1 REALIZATION OF INFINITE IMPULSE RESPONSE FILTERS IN DIRECT-FORM I AND DIRECT-FORM II

EXAMPLE 8.21

Realize the first-order digital highpass Butterworth filter

$$H(z) = \frac{0.1936 - 0.1936z^{-1}}{1 + 0.6128z^{-1}}$$

using a direct-form I.

Continued

EXAMPLE 8.21—CONT'D

Solution:

From the transfer function, we can identify

$$b_0 = 0.1936, b_1 = -0.1936, \text{and } a_1 = 0.6128.$$

Applying the direct-form I developed in Chapter 6 results in the diagram in Fig. 8.37. The digital signal processing (DSP) equation for implementation is then given by

$$y(n) = -0.6128y(n-1) + 0.1936x(n) - 0.1936x(n-1).$$

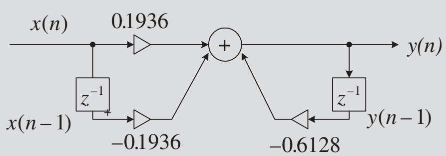

FIG. 8.37

Realization of IIR filter in Example 8.21 in direct-form I.

Program 8.14 lists the MATLAB implementation.

Program 8.14. *m*-File for Example 8.21.

```
%Sample MATLAB code
sample = 2:2:20; %Input test array
x=[ 0 0 ]; % Input buffer [x(n) x(n-1) ...]
y=[ 0 0]; % Output buffer [y(n) y(n-1) ... ]
b=[0.1936 -0.1936]; % Numerator coefficients [b0 b1 ... ]
a=[1 0.6128]; % Denominator coefficients [1 a0 a1 ...]
for n=1:1:length(sample) % Processing loop
  for k=2:-1:2
     x(k)=x(k-1);     % Shift input by one sample
     y(k)=y(k-1);     % Shift output by one sample
  end
  x(1)=sample(n);  % Get new sample
  y(1)=0;      % Digital filtering
  for k=1:1:2
     y(1)=y(1)+x(k)*b(k);
  end
  for k=2:2
     y(1)=y(1)-a(k)*y(k);
  end
  out(n)=y(1); %Output the filtered sample to output array
end
  out
```

EXAMPLE 8.22

Realize the following digital filter using a direct-form II.

$$H(z) = \frac{0.7157 + 1.4314z^{-1} + 0.7157z^{-2}}{1 + 1.3490z^{-1} + 0.5140z^{-2}}.$$

Solution:

First, we can identify

$$b_0 = 0.7157, b_1 = 1.4314, b_2 = 0.7157$$

$$\text{and } a_1 = 1.3490, a_2 = 0.5140.$$

Applying the direct-form II developed in Chapter 6 leads to Fig. 8.38.

There are two difference equations required for implementation:

$$w(n) = x(n) - 1.3490w(n-1) - 0.5140w(n-2)$$

$$y(n) = 0.7157w(n) + 1.4314w(n-1) + 0.7157w(n-2).$$

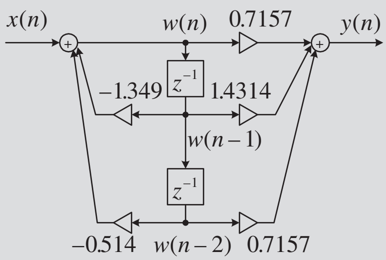

FIG. 8.38

Realization of IIR filter in Example 8.22 in direct-form II.

The MATLAB implementation is listed in Program 8.15.

Program 8.15. *m*-File for Example 8.22.

```
%Sample MATLAB code
sample =2:2:20; % Input test array
x=[0];                          %Input buffer [x(n) ]
y=[0];                          %Output buffer [y(n)]
w=[0 0 0]; % Buffer for w(n) [w(n) w(n-1) ...]
b=[0.7157 1.4314 0.7157];   % Numerator coefficients [b0 b1 ...]
a=[1 1.3490 0.5140]; % Denominator coefficients [1 a1 a2 ...]
for n=1:1:length(sample)    % Processing loop
  for k=3:-1:2
    w(k)=w(k-1);                %Shift w(n) by one sample
  end
  x(1)=sample(n);    % Get new sample
  w(1)=x(1);         % Perform IIR filtering
```

```
    for k=2:1:3
      w(1)=w(1)-a(k)*w(k);
    end
    y(1)=0;              % Perform FIR filtering
    for k=1:1:3
      y(1)=y(1)+b(k)*w(k);
    end
    out(n)=y(1);         % Send the filtered sample to output array
  end
    out
```

8.8.2 REALIZATION OF HIGHER-ORDER INFINITE IMPULSE RESPONSE FILTERS VIA THE CASCADE FORM

EXAMPLE 8.23

Given a fourth-order filter transfer function designed as

$$H(z) = \frac{0.5108z^2 + 1.0215z + 0.5108}{z^2 + 0.5654z + 0.4776} \times \frac{0.3730z^2 + 0.7460z + 0.3730}{z^2 + 0.4129z + 0.0790},$$

realize the digital filter using the cascade (series) form via second-order sections.

Solution:

Since the filter is designed using the cascade form, we have two sections of the second-order filters, whose transfer functions are

$$H_1(z) = \frac{0.5108z^2 + 1.0215z + 0.5108}{z^2 + 0.5654z + 0.4776} = \frac{0.5180 + 1.0215z^{-1} + 0.5108z^{-2}}{1 + 0.5654z^{-1} + 0.4776z^{-2}}$$

and

$$H_2(z) = \frac{0.3730z^2 + 0.7460z + 0.3730}{z^2 + 0.4129z + 0.0790} = \frac{0.3730 + 0.7460z^{-1} + 0.3730z^{-2}}{1 + 0.4129z^{-1} + 0.0790z^{-2}}.$$

Each filter section is developed using the direct-form I, shown in Fig. 8.39.

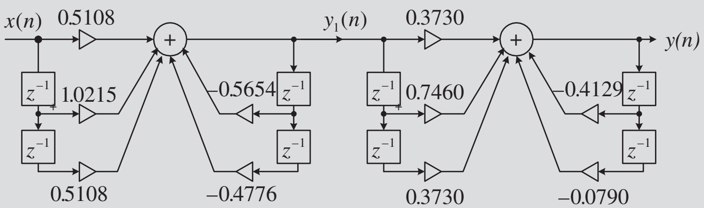

FIG. 8.39

Cascade realization of IIR filter in Example 8.23 in direct-form I.

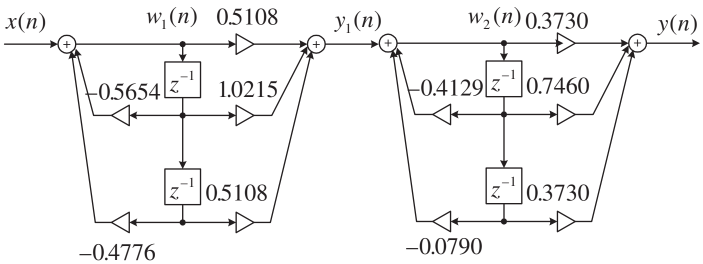

FIG. 8.40

Cascade realization of IIR filter in Example 8.23 in direct-form II.

There are two sets of DSP equations for the implementation of the first and second sections, respectively.

First section:

$$\begin{aligned} y_1(n) = &-0.5654y_1(n-1) - 0.4776y_1(n-2) \\ &+0.5108x(n) + 1.0215x(n-1) + 0.5108x(n-2) \end{aligned}$$

Second section:

$$\begin{aligned} y(n) = &-0.4129y(n-1) - 0.0790y(n-2) \\ &+0.3730y_1(n) + 0.7460y_1(n-1) + 0.3730y_1(n-2). \end{aligned}$$

Again, after we use the direct-form II for realizing each second-order filter, the realization shown in Fig. 8.40 is developed.

The difference equations for the implementation of the first section are:

$$w_1(n) = x(n) - 0.5654w_1(n-1) - 0.4776w_1(n-2)$$

$$y_1(n) = 0.5108w_1(n) + 1.0215w_1(n-1) + 0.5108w_1(n-2).$$

The difference equations for the implementation of the second section are:

$$w_2(n) = y_1(n) - 0.4129w_2(n-1) - 0.0790w_2(n-2)$$

$$y(n) = 0.3730w_2(n) + 0.7460w_2(n-1) + 0.3730w_2(n-2).$$

Note that for both direct-form I and direct-form II, the output from the first filter section becomes the input for the second filter section.

8.9 APPLICATION: 60-HZ HUM ELIMINATOR AND HEART RATE DETECTION USING ELECTROCARDIOGRAPHY

Hum noise created by poor power suppliers, transformers, or electromagnetic interference sourced by a main power supply is characterized by a frequency of 60-Hz and its harmonics. If this noise interferes with a desired audio or biomedical signal [e.g., in electrocardiography (ECG)], the desired signal could be corrupted. It is sufficient to eliminate the 60-Hz hum frequency with its second and third harmonics in most practical applications. We can complete this by cascading with notch filters with notch frequencies of 60, 120, and 180 Hz, respectively. Fig. 8.41 depicts the functional block diagram.

Now let us apply the 60-Hz hum eliminator to an ECG recording system. The ECG is a small electrical signal captured from an ECG sensor. The ECG signal is produced by the activity of the human heart, thus can be used for heart rate detection, fetal monitoring, and diagnostic purposes. The single pulse of the ECG is depicted in Fig. 8.42, which shows that an ECG signal is characterized by five peaks and valleys, labeled P, Q, R, S, and T. The highest positive wave is the R wave. Shortly before and after the R wave are negative waves called Q wave and S wave. The P wave comes before the Q wave, while the T wave comes after the S wave. The Q, R, and S waves together are called the QRS complex.

The properties of the QRS complex, with its rate of occurrence and times, highs, and widths, provide information to cardiologists concerning various pathological conditions of the heart. The reciprocal of the time period between R wave peaks (in milliseconds) multiplied by 60,000 gives instantaneous heart rate in beats per minute. On a modern ECG monitor, the acquired ECG signal is displayed for the diagnostic purpose.

However, a major source of frequent interference is the electric-power system. Such interference appears on the recorded ECG data due to electrical-field coupling between the power lines and the electrocardiograph or patient, which is the cause of the electrical field surrounding mains power lines. Another cause is magnetic induction in the power line, whereby current in the power line generates a magnetic field around the line. Sometimes, the harmonics of 60-Hz hum exist due to the nonlinear sensor and signal amplifier effects. If such interference is severe, the recorded ECG data becomes useless.

In this application, we focus on ECG enhancement for heart rate detection. To significantly reduce the 60-Hz interference, we apply signal enhancement to the ECG recording system, as shown in Fig. 8.43.

The 60-Hz eliminator removes the 60-Hz interference and has the capability to reduce its second harmonic of 120 Hz and third harmonic of 180 Hz.

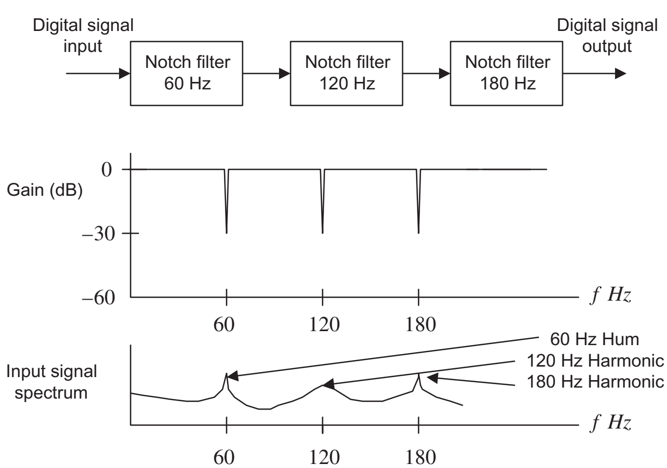

FIG. 8.41

(Top) 60-Hz Hum eliminator; (middle) the filter frequency response of the eliminator; (bottom) the input signal spectrum corrupted by the 60-Hz hum and its second and third harmonics.

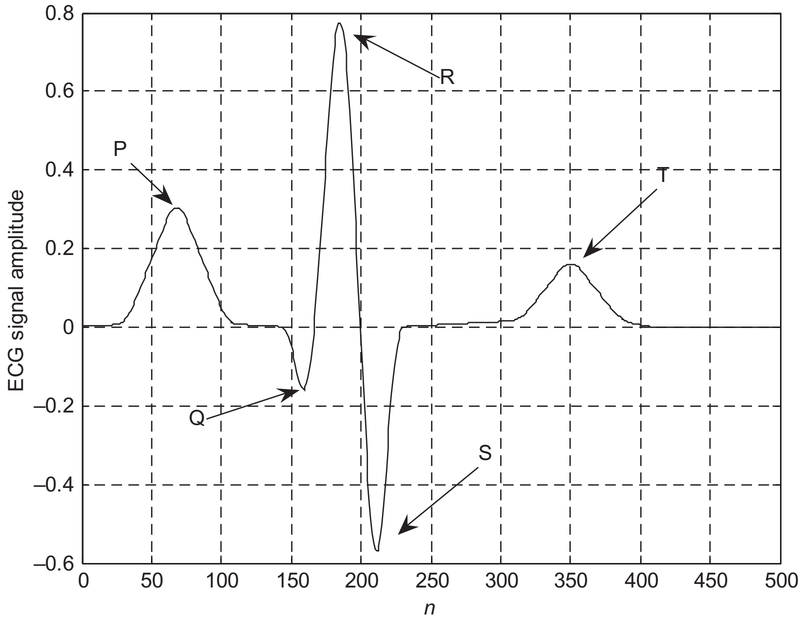

FIG. 8.42

Characteristics of the ECG pulse.

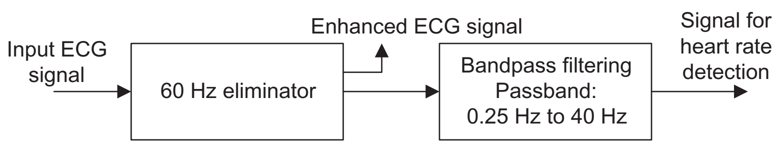

FIG. 8.43

ECG signal enhancement system.

The next objective is to detect the heart rate using the enhanced ECG signal. We need to remove DC drift and filter muscle noise, which may occur at approximately 40 Hz or more. If we consider the lowest heart rate as 30 beats per minute, the corresponding frequency is $30/60 = 0.5$ Hz. Choosing the lower cutoff frequency of 0.25 Hz should be reasonable.

Thus, a bandpass filter with a passband from 0.25 to 40 Hz [range from 0.67 to 40 Hz, discussed in Webster (2009)], either FIR or IIR type, can be designed to reduce such effects. The resultant ECG signal is valid only for the detection of heart rate. Note that the ECG signal after bandpass filtering with a passband from 0.25 to 40 Hz is no longer valid for general ECG applications, since the original ECG signal occupies the frequency range from 0.01 to 250 Hz (diagnostic-quality ECG), as discussed in Carr and Brown (2001) and Webster (2009). The enhanced ECG signal from the 60-Hz hum eliminator can serve for general ECG signal analysis (which is beyond the scope of this book). We summarize the design specifications for the heart rate detection application as follows:

System outputs: Enhanced ECG signal with the 60-Hz elimination
Processed ECG signal for heart rate detection

60-Hz eliminator:

Harmonics to be removed:	60 Hz (fundamental) 120 Hz (second harmonic) 180 Hz (third harmonic)
3-dB bandwidth for each filter:	4 Hz
Sampling rate:	600 Hz
Notch filter type:	Second-order IIR
Design method:	Pole-zero placement

Bandpass filter:

Passband frequency range:	0.25–40 Hz
Passband ripple:	0.5 dB
Filter type:	Chebyshev fourth order
Design method:	BLT method
DSP sampling rate:	600 Hz

Let us carry out the 60-Hz eliminator design and determine the transfer function and difference equation for each notch filter and bandpass filter. For the first section with the notch frequency of 60 Hz, applying Eqs. (8.47)–(8.50) leads to

$$r = 1 - \left(\frac{4}{600}\right) \times \pi = 0.9791$$

$$\theta = \left(\frac{60}{600}\right) \times 360^{\circ} = 36^{\circ}.$$

We calculate $2\cos(36^{\circ}) = 1.6180$, $2r\cos(36^{\circ}) = 1.5842$, and

$$K = \frac{(1 - 2r\cos\theta + r^2)}{(2 - 2\cos\theta)} = 0.9803.$$

Hence it follows that

$$H_1(z) = \frac{0.9803 - 1.5862z^{-1} + 0.9803z^{-2}}{1 - 1.5842z^{-1} + 0.9586z^{-2}}$$

$$y_1(n) = 0.9803x(n) - 1.5862x(n-1) + 0.9802x(n-2) + 1.5842y_1(n-1) - 0.9586y_1(n-2).$$

Similarly, we yield the transfer functions and difference equations for the second section and the third section as

Second section:

$$H_2(z) = \frac{0.9794 - 0.6053z^{-1} + 0.9794z^{-2}}{1 - 0.6051z^{-1} + 0.9586z^{-2}}$$

$$y_2(n) = 0.9794y_1(n) - 0.6053y_1(n-1) + 0.9794y_1(n-2) + 0.6051y_2(n-1) - 0.9586y_2(n-2)$$

Third section:

$$H_3(z) = \frac{0.9793 + 0.6052z^{-1} + 0.9793z^{-2}}{1 + 0.6051z^{-1} + 0.9586z^{-2}}$$

$$y_3(n) = 0.9793y_2(n) + 0.6052y_2(n-1) + 0.9793y_2(n-2) - 0.6051y_3(n-1) - 0.9586y_3(n-2).$$

The cascaded frequency responses are plotted in Fig. 8.44. As we can see, the rejection for each notch frequency is below 50 dB.

The second-stage design using the BLT gives the bandpass filter transfer function and difference equation

$$H_4(z) = \frac{0.0464 - 0.0927z^{-2} + 0.0464z^{-4}}{1 - 3.3523z^{-1} + 4.2557z^{-2} - 2.4540z^{-3} + 0.5506z^{-4}}$$

$$\begin{aligned} y_4(n) &= 0.0464y_3(n) - 0.0927y_3(n-2) + 0.0464y_3(n-4) \\ &+ 3.3523y_4(n-1) - 4.25571y_4(n-2) + 2.4540y_4(n-3) - 0.5506y_4(n-4). \end{aligned}$$

Fig. 8.45 depicts the processed results at each stage. In Fig. 8.45, plot A shows the initial corrupted ECG data, which includes 60-Hz interference and its 120 and 180 Hz harmonics, along with muscle noise. Plot B shows that the interferences of 60 Hz and its harmonics of 120 and 180 Hz have been removed. Finally, plot C displays the result after the bandpass filter. As we have expected, the muscle noise has been removed; and the enhanced ECG signal has been observed.

MATLAB simulation is listed in Program 8.16.

With the processed ECG signal, a simple zero-cross algorithm can be designed to detect the heart rate. Based on plot C in Fig. 8.45, we use a threshold value of 0.5 and continuously compare each of two

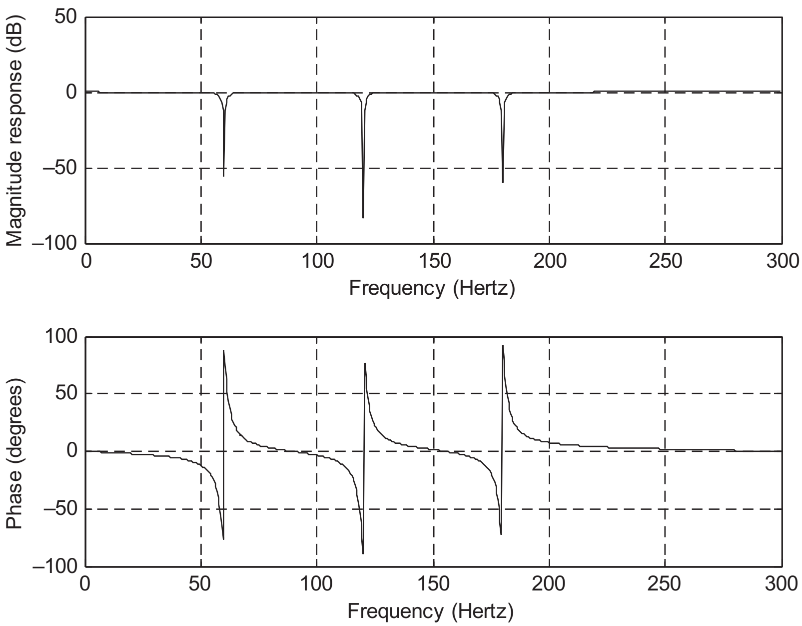

FIG. 8.44

Frequency responses of three cascaded notch filters.

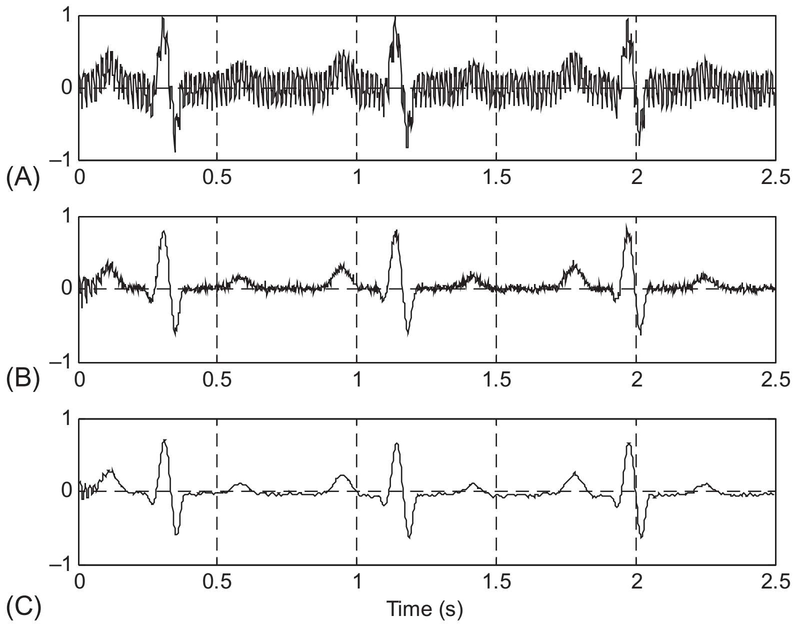

FIG. 8.45

Results of ECG signal processing: (A) initial corrupted ECG data; (B) ECG data enhanced by removing 60 Hz interference; (C) ECG data with DC blocking and noise removal for heart rate detection.

consecutive samples with the threshold. If both results are opposite, then a zero cross is detected. Each zero-crossing measure is given by

$$\text{Zero crossing} = \frac{|cur\_sign - pre\_sign|}{2},$$

where *cur_sign* and *pre_sign* are determined based on the current input $x(n)$, the past input $x(n-1)$, and the threshold value, given as

$$\begin{aligned}&\text{if } x(n) \geq threshold\ cur\_sign = 1 \text{ else } cur\_sign = -1\\&\text{if } x(n-1) \geq threshold\ pre\_sign = 1 \text{ else } pre\_sign = -1.\end{aligned}$$

Fig. 8.46 summarizes the algorithm.

After detecting the total number of zero crossings, the number of the peaks will be half the number of the zero crossings. The heart rate in terms of pulses per minute can be determined by

$$\text{Heart rate} = \frac{60}{\left(\dfrac{\text{Number of enhanced } ECG \text{ data}}{f_s}\right)} \times \left(\frac{\text{zero crossing number}}{2}\right).$$

In our simulation, we have detected six zero-crossing points using 1500 captured data at a sampling rate of 600 samples per second. Hence,

$$\text{Heart rate} = \frac{60}{\left(\dfrac{1500}{600}\right)} \times \left(\frac{6}{2}\right) = 72 \text{ pulses per minute}.$$

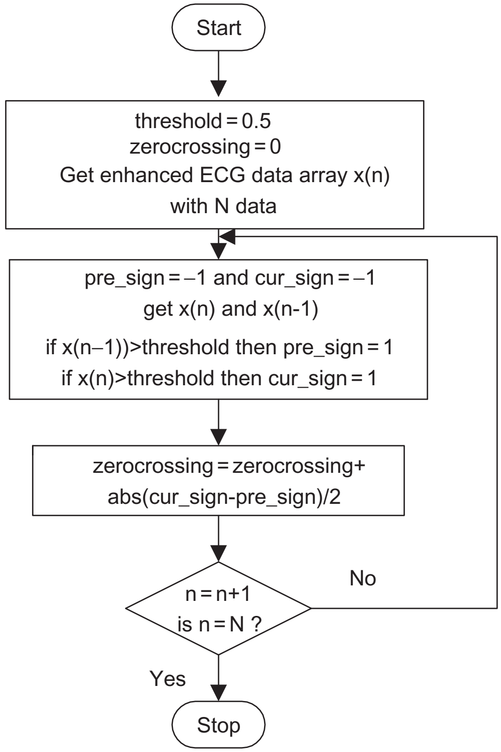

FIG. 8.46

A simple zero-crossing algorithm.

The MATLAB implementation of the zero-crossing detection can be found in the last part in Program 8.16.

Program 8.16. MATLAB program for heart rate detection using an ECG signal.

```
load ecgbn.dat; % Load noisy ECG recording
b1=[0.9803 -1.5862 0.9803];    %Notch filter with a notch frequency of 60 Hz
a1=[1 -1.5842 0.9586];
b2=[0.9794 -0.6053 0.9794];    % Notch filter with a notch frequency 120 Hz
a2=[1 -0.6051 0.9586];
b3=[0.9793 0.6052 0.9793];     % Notch filter with a notch frequency of 180 Hz
a3=[1 0.6051 0.9586];
y1=filter(b1,a1,ecgbn);  % First section filtering
y2=filter(b2,a2,y1);     % Second section filtering
y3=filter(b3,a3,y2);     % Third section filtering
%Bandpass filter
fs=600;               % Sampling rate
T=1/600;              % Sampling interval
```

```
% BLT design
wd1=2*pi*0.25;
wd2=2*pi*40;
wa1=(2/T)*tan(wd1*T/2);
wa2=(2/T)*tan(wd2*T/2);
[B,A]=lp2bp([1.4314], [1 1.4652 1.5162],sqrt(wa1*wa2),wa2-wa1);
[b,a]=bilinear(B,A,fs);
b =[ 0.046361 0 -0.092722 0 0.046361]
a =[1 -3.352292 4.255671 -2.453965 0.550587]
y4=filter(b,a,y3);              %Bandpass filtering
t=0:T:1499*T;                   % Recover time
subplot(3,1,1);plot(t,ecgbn);grid;ylabel('(a)');
subplot(3,1,2);plot(t,y3);grid;ylabel('(b)');
subplot(3,1,3);plot(t,y4);grid;ylabel('(c)');
xlabel('Time (sec.)');
%Zero cross algorithm
zcross=0.0;threshold=0.5
for n=2:length(y4)
  pre_sign=-1;cur_sign=-1;
 if y4(n-1)>threshold
   pre_sign=1;
 end
 if y4(n)>threshold
   cur_sign=1;
 end
zcross=zcross+abs(cur_sign-pre_sign)/2;
end
zcross                               % Output the number of zero crossings
rate=60*zcross/(2*length(y4)/600)    % Output the heart rate
```

8.10 COEFFICIENT ACCURACY EFFECTS ON INFINITE IMPULSE RESPONSE FILTERS

In practical applications, the IIR filter coefficients with infinite precision may be quantized due to the finite word length. Quantization of infinite precision filter coefficients changes the locations of the zeros and poles of the designed filter transfer function, hence changes the filter frequency responses. Since analysis of filter coefficient quantization for the IIR filter is very complicated and beyond the scope of this textbook, we pick only a couple of simple cases for discussion. Filter coefficient quantization for specific processors such as the fixed-point DSP processor and floating-point processor will be included in Chapter 14. To illustrate this effect, we look at the following first-order IIR filter transfer function having filter coefficients with infinite precision,

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}. \tag{8.59}$$

After filter coefficient quantization, we have the quantized digital IIR filter transfer function

$$H^q(z) = \frac{b_0^q + b_1^q z^{-1}}{1 + a_1^q z^{-1}}. \tag{8.60}$$

Solving for pole and zero, we achieve

$$z_1 = -\frac{b_1^q}{b_0^q} \tag{8.61}$$

$$p_1 = -a_1^q. \tag{8.62}$$

Now considering a second-order IIR filter transfer function as

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \tag{8.63}$$

and its quantized IIR filter transfer function

$$H^q(z) = \frac{b_0^q + b_1^q z^{-1} + b_2^q z^{-2}}{1 + a_1^q z^{-1} + a_2^q z^{-2}}, \tag{8.64}$$

solving for poles and zeros finds:

$$z_{1,2} = -0.5 \times \frac{b_1^q}{b_0^q} \pm j\left(\frac{b_2^q}{b_0^q} - 0.25 \times \left(\frac{b_1^q}{b_0^q}\right)^2\right)^{\frac{1}{2}} \tag{8.65}$$

$$p_{1,2} = -0.5 \times a_1^q \pm j\left(a_2^q - 0.25 \times \left(a_1^q\right)^2\right)^{\frac{1}{2}}. \tag{8.66}$$

With the developed Eqs. (8.61) and (8.62) for the first-order IIR filter, and Eqs. (8.65) and (8.66) for the second-order IIR filter, we can study the effects of the location changes of the poles and zeros, and the frequency responses due to filter coefficient quantization.

EXAMPLE 8.24

Given the following first-order IIR filter

$$H(z) = \frac{1.2341 + 0.2126 z^{-1}}{1 - 0.5126 z^{-1}},$$

and assuming that we use 1 sign bit and 6 bits for encoding the magnitude of the filter coefficients, find the quantized transfer function and pole-zero locations.

Solution:

Let us find the pole and zero for infinite precision filter coefficients:
Solving $1.2341z + 0.2126 = 0$ leads a zero location $z_1 = -0.17227$.
Solving $z - 0.5126 = 0$ gives a pole location $p_1 = 0.5126$.
Now let us quantize the filter coefficients. Quantizing 1.2341 can be illustrated as
$1.2341 \times 2^5 = 39.4912 = 39$(rounded to integer).
Since the maximum magnitude of the filter coefficients is 1.2341, which is between 1 and 2, we scale all coefficient magnitudes by a factor of 2^5 and round off each value to an integer whose magnitude is encoded using 6 bits. As shown in the quantization, 6 bits are required to encode the integer 39. When the coefficient integer is scaled back by the same scale factor, the corresponding quantized coefficient with finite precision (7 bits, including the sign bit) is found to be

Continued

EXAMPLE 8.24—CONT'D

$$b_0^q = \frac{39}{2^5} = 1.21875.$$

Following the same procedure, we can obtain

$$b_1^q = 0.1875$$

and

$$a_1^q = -0.5.$$

Thus we achieve the quantized transfer function

$$H^q(z) = \frac{1.21875 + 0.1875z^{-1}}{1 - 0.5z^{-1}}.$$

Solving for pole and zero leads to

$$p_1 = 0.5$$

and

$$z_1 = -0.1538.$$

It is clear that the pole and zero locations change after the filter coefficients are quantized. This effect can change the frequency response of the designed filter as well. In Example 8.25, we study the quantization of the filter coefficients for the second-order IIR filter and examine the pole/zero location changes and magnitude/phase frequency responses.

EXAMPLE 8.25

A second-order digital lowpass Chebyshev filter with a cutoff frequency of 3.4 kHz and 0.5-dB ripple on passband at a sampling frequency of 8000 Hz is designed. Assume that we use 1 sign bit and 7 bits for encoding the magnitude of each filter coefficient. The z-transfer function is given by

$$H(z) = \frac{0.7434 + 1.4865z^{-1} + 0.7434z^{-2}}{1 + 1.5149z^{-1} + 0.6346z^{-2}}.$$

(a) Find the quantized transfer function and pole and zero locations.
(b) Plot the magnitude and phase responses, respectively.

Solution:

(a) Since the maximum magnitude of the filter coefficients is between 1 and 2, the scale factor for quantization is chosen to be 2^6, so that the coefficient integer can be encoded using 7 bits.

After performing filter coefficient encoding, we have

$$H^q(z) = \frac{0.7500 + 1.484375z^{-1} + 0.7500z^{-2}}{1 + 1.515625z^{-1} + 0.640625z^{-2}}.$$

For comparison, the uncoded zeros and encoded zeros of the transfer function $H(z)$ are as follows:
Uncoded zeros: –1, –1;
Coded zeros: –0.9896 + 0.1440i, –0.9896 –0.1440i.
Similarly, the uncoded poles and coded poles of the transfer function $H^q(z)$ are as follows:
Uncoded poles: –0.7574 + 0.2467i, –0.7574 –0.2467i;
Coded poles: –0.7578 + 0.2569i, –0.7578 –0.2569i.

(b) The comparisons for the magnitude responses and phase responses are listed in Program 8.17 and plotted in Fig. 8.47.

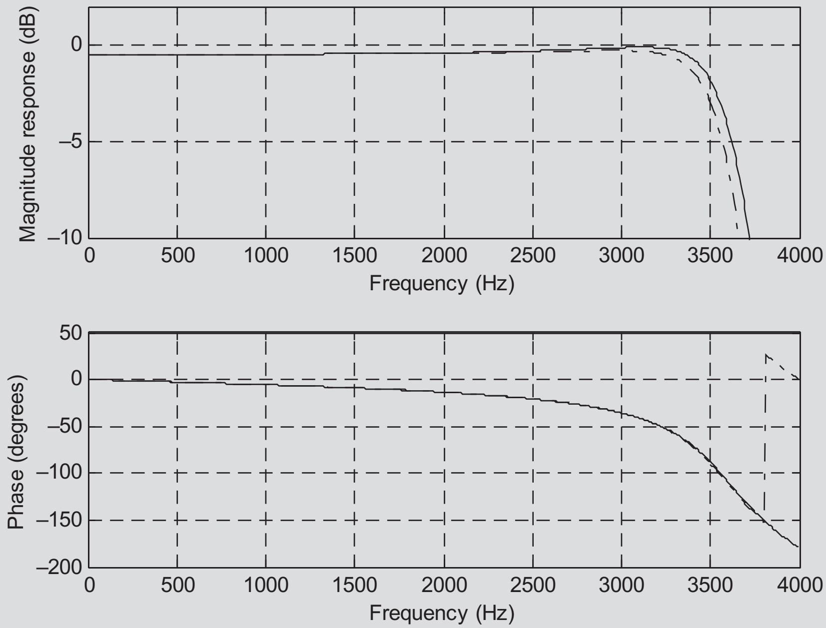

FIG. 8.47

Frequency responses (dash-dotted line, quantized coefficients; solid line, unquantized coefficients).

Program 8.17. MATLAB *m*-file for Example 8.25.

```
%Example 8.25
% Plot the magnitude and phase responses
 fs=8000;                         % Sampling rate
 B=[0.7434 1.4868 0.7434];
 A=[1 1.5149 0.6346];
 [hz,f]=freqz(B,A,512,fs);   % Calculate responses without coefficient quantization
 phi=180*unwrap(angle(hz))/pi;
 Bq=[0.750 1.4834375 0.75000];
 Aq=[1 1.515625 0.640625];
```

```
[hzq,f]=freqz(Bq,Aq,512,fs); % Calculate responses with coefficient quantization
phiq=180*unwrap(angle(hzq))/pi;
subplot(2,1,1), plot(f,20*log10(abs(hz)),f,20*log10(abs(hzq)),'-.');grid;
axis([0 4000 -10 2])
xlabel('Frequency (Hz)');
ylabel('Magnitude Response (dB)');
subplot(2,1,2), plot(f, phi, f, phiq,'-.'); grid;
xlabel('Frequency (Hz)');
ylabel('Phase (degrees)');
```

From Fig. 8.47, we observe that the quantization of IIR filter coefficients has more effect on magnitude response and less effect on the phase response in the passband. In practice, one needs to verify this effect to make sure that the magnitude frequency response meets the filter specifications.

8.11 APPLICATION: GENERATION AND DETECTION OF DTMF TONES USING THE GOERTZEL ALGORITHM

In this section, we study an application of the digital filters for the generation and detection of dual-tone multifrequency (DTMF) signals used for telephone touch keypads. In our daily life, DTMF touch tones produced by telephone keypads on handsets are applied to dial telephone numbers routed to telephone companies, where the DTMF tones are digitized and processed and the detected dialed telephone digits are used for the telephone switching system to ring the party to be called. A telephone touch keypad is shown in Fig. 8.48, where each key is represented by two tones with their specified frequencies. For example, if the key "7" is pressed, the DTMF signal containing the designated frequencies of 852 and 1209 Hz is generated, which is sent to the central office at the telephone company for processing. At the

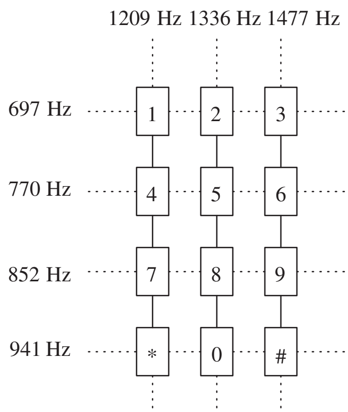

FIG. 8.48

DTMF tone specifications.

central office, the received DTMF tones are detected through the digital filters and some logic operations are used to decode the dialed signal consisting of 852 and 1209 Hz as key "7." The frequencies defined for each key are in Fig. 8.48.

8.11.1 SINGLE-TONE GENERATOR

Now, let us look at a digital tone generator whose transfer function is obtained from the z-transform function of a sinusoidal sequence $\sin(n\Omega_0)$ as

$$H(z) = \frac{z\sin\Omega_0}{z^2 - 2z\cos\Omega_0 + 1} = \frac{z^{-1}\sin\Omega_0}{1 - 2z^{-1}\cos\Omega_0 + z^{-2}}, \tag{8.67}$$

where Ω_0 is the normalized digital frequency. Given the sampling rate of the DSP system and the frequency of the tone to be generated, we have the relationship

$$\Omega_0 = \frac{2\pi f_0}{f_s}. \tag{8.68}$$

Applying the inverse z-transform to the transfer function leads to the difference equation

$$y(n) = \sin\Omega_0 x(n-1) + 2\cos\Omega_0 y(n-1) - y(n-2), \tag{8.69}$$

since

$$Z^{-1}(H(z)) = Z^{-1}\left(\frac{z\sin\Omega_0}{z^2 - 2z\cos\Omega_0 + 1}\right) = \sin(\Omega_0 n) = \sin\left(\frac{2\pi f_0 n}{f_s}\right),$$

which is the impulse response. Hence, to generate a pure tone with the amplitude of A, an impulse function $x(n) = A\delta(n)$ must be used as the input to the digital filter, as illustrated in Fig. 8.49.

Now, we illustrate implementation. Assuming that the sampling rate of the DSP system is 8000 Hz, we need to generate a digital tone of 1 kHz. Then we compute

$$\Omega_0 = 2\pi \times \frac{1000}{8000} = \frac{\pi}{4},\ \sin\Omega_0 = 0.707107, \text{ and } 2\cos\Omega_0 = 1.414214.$$

The required filter transfer function is determined as

$$H(z) = \frac{0.707107z^{-1}}{1 - 1.414214z^{-1} + z^{-2}}.$$

The MATLAB simulation using the input $x(n) = \delta(n)$ is displayed in Fig. 8.50, where the top plot is the generated tone of 1 kHz, and the bottom plot shows its spectrum. The corresponding MATLAB list is given in Program 8.18.

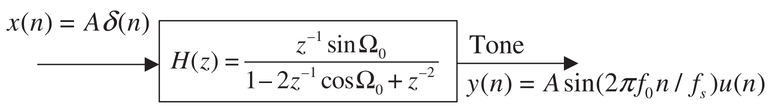

FIG. 8.49

Single-tone generator.

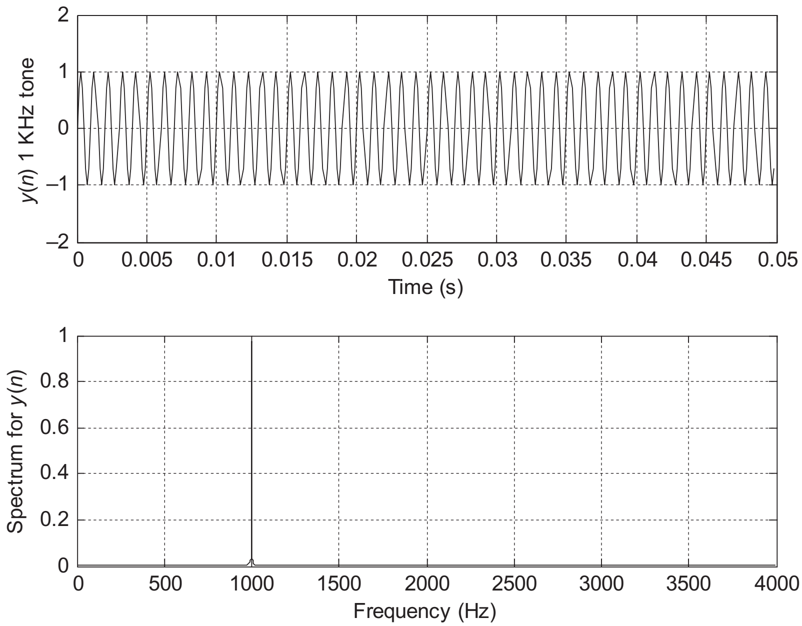

FIG. 8.50

Plots of a generated single tone of 1000 Hz and its spectrum.

Note that if we replace the filter $H(z)$ with the z-transform of other sequences such as a cosine function and use the impulse sequence as the filter input, the filter will generate the corresponding digital wave such as the digital cosine wave.

Program 8.18. MATLAB program for generating a sinusoid.

```
fs=8000;                                    % Sampling rate
t=0:1/fs:1;                                 % Time vector for 1 s
x=zeros(1,length(t));                       % Initialize input to be zero
x(1)=1;                                     % Set up impulse function
y=filter([0 0.707107][1 -1.414214 1],x);    % Perform filtering
subplot(2,1,1);plot(t(1:400),y(1:400));grid
ylabel('y(n) 1 kHz tone'); xlabel('time (second)')
Ak=2*abs(fft(y))/length(y);Ak(1)=Ak(1)/2; % One-sides amplitude spectrum
f=[0:1:(length(y)-1)/2]*fs/length(y);      % Indices to frequencies (Hz) for plot
subplot(2,1,2);plot(f,Ak(1:(length(y)+1)/2));grid
ylabel('Spectrum for y(n)'); xlabel('frequency (Hz)')
```

8.11.2 DUAL-TONE MULTIFREQUENCY TONE GENERATOR

Now that the principle of a single-tone generator is illustrated, we can extent it to develop the DTMF tone generator using two digital filters in parallel. The DTMF tone generator for key "7" is depicted in Fig. 8.51.

Here we generate the DTMF tone for key "7" for a duration of 1 s, assuming the sampling rate of 8000 Hz. The generated tone and its spectrum are plotted in Fig. 8.52 for verification, while the MATLAB implementation is given in Program 8.19.

Program 8.19. MATLAB program for DTMF tone generation.

```
close all; clear all
fs=8000;                 % Sampling rate
t=0:1/fs:1;              % 1-s time vector
x=zeros(1,length(t));    % Initialize input to be zero
x(1)=1;                  % Set-up impulse function
% Generate 852-Hz tone
y852=filter([0 sin(2*pi*852/fs)][1 -2*cos(2*pi*852/fs) 1],x);
% Generate 1209-Hz tone
y1209=filter([0 sin(2*pi*1209/fs) ][1 -2*cos(2*pi*1209/fs) 1],x); % Filtering
y7=y852+y1209;           % Generate DTMF tone
subplot(2,1,1);plot(t(1:400),y7(1:400));grid
ylabel('y(n) DTMF: number 7');
xlabel('time (second)')
Ak=2*abs(fft(y7))/length(y7);Ak(1)=Ak(1)/2; % One-sided amplitude spectrum
f=[0:1:(length(y7)-1)/2]*fs/length(y7); % Map indices to frequencies (Hz) for plot
subplot(2,1,2);plot(f,Ak(1:(length(y7)+1)/2));grid
ylabel('Spectrum for y7(n)');
xlabel('frequency (Hz)');
```

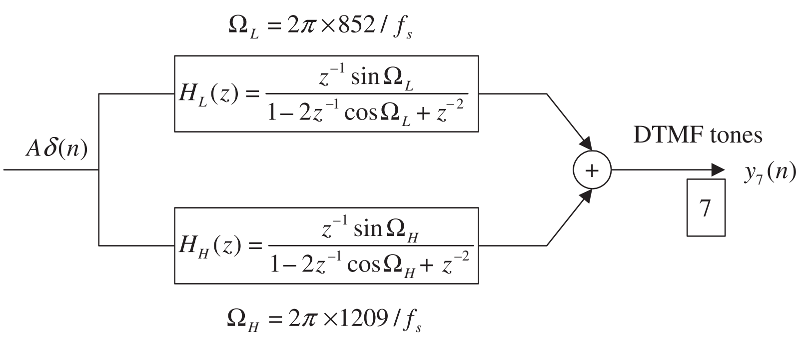

FIG. 8.51

Digital DTMF tone generator for the keypad digit "7."

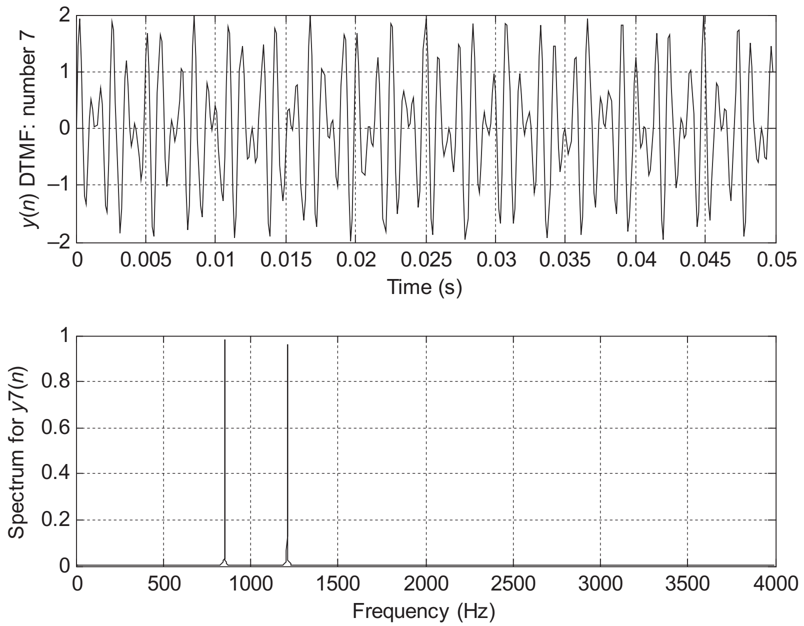

FIG. 8.52

Plots of the generated DTMF tone of "7" and its spectrum.

8.11.3 GOERTZEL ALGORITHM

In practice, the DTMF tone detector is designed using the Goertzel algorithm. This is a special and powerful algorithm used for computing discrete Fourier transform (DFT) coefficients and signal spectra using a digital filtering method. The modified Goertzel algorithm can be used for computing signal spectra without involving complex algebra like the DFT algorithm.

To devise the Goertzel algorithm, we begin with applying DFT to a sequence $x(n)$ as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} = e^{j\frac{2\pi kN}{N}} \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \tag{8.70}$$

Note that $W_N^k = e^{-2\pi k/N}$ and adding $e^{j2\pi kN/N} = 1$ does not affect Eq. (8.70). We rewrite Eq. (8.70) in the convolution format. It follows that

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{j\frac{2\pi k}{N}(N-n)} \tag{8.71}$$

In Eq. (8.71), if we define $h_k(n) = e^{j2\pi kn/N}$ and note that $h_k(n)$ is periodic of N, we can express $X(k)$ as

$$X(k) = \sum_{n=0}^{N-1} x(n) h_k(N-n) = x(0)h_k(N) + x(1)h_k(N-1) + \cdots + x(N-1)h_k(1). \tag{8.72}$$

The results of Eq. (8.72) will be used later. If we treat $h_k(n)$ as the FIR filter coefficients, its output can be obtained by convolution sum as shown below:

$$y_k(m) = \sum_{n=0}^{m} x(n) h_k(m-n), \tag{8.73}$$

where $x(n)$ is a causal sequence. The convolution is depicted below:

After filtering $N+1$ times, we obtain

$$y_k(0) = \sum_{n=0}^{0} x(n) h_k(0-n) = x(0) h_k(0),$$

$$y_k(1) = \sum_{n=0}^{1} x(n) h_k(1-n) = x(0) h_k(1) + x(1) h_k(0),$$

$$y_k(2) = \sum_{n=0}^{2} x(n) h_k(2-n) = x(0) h_k(2) + x(1) h_k(1) + x(2) h_k(0),$$

$$\cdots\cdot$$

$$\begin{aligned} y_k(N) &= \sum_{n=0}^{N} x(n) h_k(N-n) \\ &= x(0) h_k(N) + x(1) h_k(N-1) + \cdots + x(N-1) h_k(1) + x(N) h_k(0). \end{aligned} \tag{8.74}$$

Now, if we set $x(N)=0$, then

$$y_k(N) = x(0) h_k(N) + x(1) h_k(N-1) + \cdots + x(N-1) h_k(1) = X(k). \tag{8.75}$$

In general, the convolution form is expressed as

$$y_k(n) = x(n) * h_k(n). \tag{8.76}$$

But we can find a recursive solution by considering $h_k(n) = e^{j2\pi kn/N}$ and $n \to \infty$. We see that

$$\begin{aligned} H_k(z) &= h_k(0) + h_k(1) z^{-1} + h_k(2) z^{-2} + \cdots \\ &= \left(e^{j\frac{2\pi k}{N}}\right)^0 + \left(e^{j\frac{2\pi k}{N}}\right)^1 z^{-1} + \left(e^{j\frac{2\pi k}{N}}\right)^2 z^{-2} + \cdots. \end{aligned} \tag{8.77}$$

For $|e^{j\frac{2\pi k}{N}} z^{-1}| < 1$, we achieve the filter transfer function as

$$H_k(z) = \frac{1}{1 - e^{j\frac{2\pi k}{N}} z^{-1}} = \frac{1}{1 - W_N^{-k} z^{-1}}. \tag{8.78}$$

Through simple manipulations, we yield the Goertzel filter as follows:

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} \frac{1 - W_N^k z^{-1}}{1 - W_N^k z^{-1}} = \frac{1 - W_N^k z^{-1}}{1 - 2\cos\left(\frac{2\pi k}{N}\right) z^{-1} + z^{-2}}. \tag{8.79}$$

Note that the Goertzel filter $H_k(z)$ is a marginally stable filter and has a resonance response at $\Omega = 2\pi k/N$.

Comparing Eqs. (8.72), (8.74), and (8.75), we can compute DFT coefficient $X(k)$ at the specified frequency bin k with the given input data $x(n)$ for $n=0, 1, \ldots, N-1$, and the last element set to be $x(N)=0$ using Goertzel filter $H_k(z)$. We can process the data sequence $N+1$ times to achieve the filter output as $y_k(n)$ for $n=0, 1, \ldots, N$. The DFT coefficient $X(k)$ is the last datum from the Goertzel filter, that is,

$$X(k)=y_k(N). \tag{8.80}$$

The implementation of the Goertzel filter is presented by direct-form II realization in Fig. 8.53.

According to the direct-form II realization, we can write the Goertzel algorithm as

$$x(N)=0 \tag{8.81}$$

$$\text{for } n=0,1,\cdots,N$$

$$v_k(n)=2\cos\left(\frac{2\pi k}{N}\right)v_k(n-1)-v_k(n-2)+x(n). \tag{8.82}$$

$$y_k(n)=v_k(n)-W_N^k v_k(n-1) \tag{8.83}$$

with initial conditions: $v_k(-2)=0$, $v_k(-1)=0$.

Then the DFT coefficient $X(k)$ is given as

$$X(k)=y_k(N) \tag{8.84}$$

The squared magnitude of $x(k)$ is computed as

$$|X(k)|^2=v_k^2(N)+v_k^2(N-1)-2\cos\left(\frac{2\pi k}{N}\right)v_k(N)v_k(N-1). \tag{8.85}$$

We show the derivation of Eq. (8.85) as follows. Note that Eq. (8.83) involves complex algebra, since the equation contains only one complex number, a factor

$$W_N^k=e^{-j\frac{2\pi k}{N}}=\cos\left(\frac{2\pi k}{N}\right)-j\sin\left(\frac{2\pi k}{N}\right)$$

discussed in Chapter 4. If our objective is to compute the spectrum value, we can substitute $n=N$ into Eq. (8.83) to obtain $X(k)$ and multiply $X(k)$ by its conjugate $X^*(k)$ to achieve the squared magnitude of the DFT coefficient. It follows (Ifeachor and Jervis, 2002) that

$$|X(k)|^2=X(k)X^*(k).$$

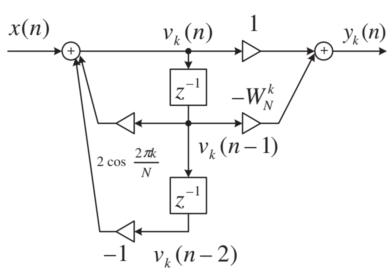

FIG. 8.53

Second-order Goertzel IIR filter.

Since $X(k) = y_k(N) - W_N^k v_k(N-1)$,

$$X^*(k) = y_k(N) - W_N^{-k} v_k(N-1)$$

then

$$\begin{aligned} |X(k)|^2 &= \left(y_k(N) - W_N^k y_k(N-1)\right)\left(y_k(N) - W_N^{-k} y_k(N-1)\right) \\ &= y_k^2(N) + y_k^2(N-1) - \left(W_N^k + W_N^{-k}\right) y_k(N) y_k(N-1). \end{aligned} \quad (8.86)$$

Using Euler's identity yields

$$W_N^k + W_N^{-k} = e^{-j\frac{2\pi}{N}k} + e^{j\frac{2\pi}{N}k} = 2\cos\left(\frac{2\pi k}{N}\right). \quad (8.87)$$

Substituting Eq. (8.87) into Eq. (8.86) leads to Eq. (8.85).

We can see that the DSP equation for $v_k(k)$ and computation of the squared magnitude of the DFT coefficient $|X(k)|^2$ do not involve any complex algebra. Hence, we will use this advantage for later development. To illustrate the algorithm, let us consider Example 8.26.

EXAMPLE 8.26

Given the digital data sequence of length 4 as $x(0) = 1$, $x(1) = 2$, $x(2) = 3$, and $x(3) = 4$, use the Goertzel algorithm to compute DFT coefficient $X(1)$ and the corresponding spectral amplitude at the frequency bin $k = 1$.

Solution:

We have $k = 1$, $N = 4$, $x(0) = 1$, $x(1) = 2$, $x(2) = 3$, and $x(3) = 4$. Note that

$$2\cos\left(\frac{2\pi}{4}\right) = 0 \text{ and } W_4^1 = e^{-j\frac{2\pi\times 1}{4}} = \cos\left(\frac{\pi}{2}\right) - j\sin\left(\frac{\pi}{2}\right) = -j.$$

We first write the simplified difference equations:

$$x(4) = 0$$

$$\text{for } n = 0, 1, \cdots, 4$$

$$v_1(n) = -v_1(n-2) + x(n)$$

$$y_1(n) = v_1(n) + jv_1(n-1)$$

then

$$X(1) = y_1(4)$$

$$|X(1)|^2 = v_1^2(4) + v_1^2(3).$$

The digital filter process is demonstrated in the following:

$$v_1(0) = -v_1(-2) + x(0) = 0 + 1 = 1$$

$$y_1(0) = v_1(0) + jv_1(-1) = 1 + j \times 0 = 1$$

$$v_1(1) = -v_1(-1) + x(1) = 0 + 2 = 2$$

$$y_1(1) = v_1(1) + jv_1(0) = 2 + j \times 1 = 2 + j$$

Continued

EXAMPLE 8.26—CONT'D

$$v_1(2) = -v_1(0) + x(2) = -1 + 3 = 2$$

$$y_1(2) = v_1(2) + jv_1(1) = 2 + j \times 2 = 2 + j2$$

$$v_1(3) = -v_1(1) + x(3) = -2 + 4 = 2$$

$$y_1(3) = v_1(3) + jv_1(2) = 2 + j \times 2 = 2 + j2$$

$$v_1(4) = -v_1(2) + x(4) = -2 + 0 = -2$$

$$y_1(4) = v_1(4) + jv_1(3) = -2 + j \times 2 = -2 + j2.$$

Then the DFT coefficient and its squared magnitude are determined as

$$X(1) = y_1(4) = -2 + j2$$

$$|X(1)|^2 = v_1^2(4) + v_1^2(3) = (-2)^2 + (2)^2 = 8.$$

Thus, the two-side amplitude spectrum is computed as

$$A_1 = \frac{1}{4}\sqrt{\left(|X(1)|^2\right)} = 0.7071$$

and the corresponding single-sided amplitude spectrum is $A_1 = 2 \times 0.707 = 1.4141$.

From this simple illustrative example, we see that the Goertzel algorithm has the following advantages:

1. We can apply the algorithm for computing the DFT coefficient $X(k)$ at a specified frequency bin k; unlike the fast Fourier transform (FFT) algorithm, all the DFT coefficients are computed once it is applied.
2. If we want to compute the spectrum at frequency bin k, that is, $|X(k)|$, Eq. (8.82) shows that we need to process $v_k(n)$ for $N+1$ times and then compute $|X(k)|^2$. The operations avoid complex algebra.

If we use the modified Goertzel filter in Fig. 8.54, then the corresponding transfer function is given by

$$G_k(z) = \frac{V_k(z)}{X(z)} = \frac{1}{1 - 2\cos\left(\frac{2\pi k}{N}\right)z^{-1} + z^{-2}}. \tag{8.88}$$

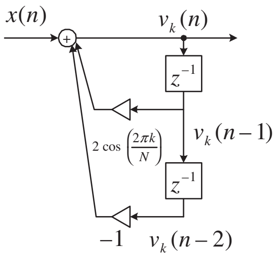

FIG. 8.54

Modified second-order Goertzel IIR filter.

The modified Goertzel algorithm becomes

$$x(N)=0$$

$$\text{for}\, n=0,1,\cdots,N,$$

$$v_k(n)=2\cos\left(\frac{2\pi k}{N}\right)v_k(n-1)-v_k(n-2)+x(n)$$

with initial conditions: $v_k(-2)=0$, and $v_k(-1)=0$.

Then the squared magnitude of the DFT coefficient is given by

$$|X(k)|^2=v_k^2(N)+v_k^2(N-1)-2\cos\left(\frac{2\pi k}{N}\right)v_k(N)v_k(N-1).$$

EXAMPLE 8.27

Given the digital data sequence of length 4 as $x(0)=1$, $x(1)=2$, $x(2)=3$, and $x(3)=4$, use the Goertzel algorithm to compute the spectral amplitude at the frequency bin $k=0$.

Solution:

$$k=0, N=4, x(0)=1, x(1)=2, x(2)=3, \text{and}\, x(3)=4.$$

Using the modified Goertzel algorithm and noting that $2\cdot\cos\left(\frac{2\pi}{4}\times 0\right)=2$, we get the simplified difference equations as

$$x(4)=0$$

$$\text{for}\, n=0,1,\cdots,4$$

$$v_0(n)=2v_0(n-1)-v_0(n-2)+x(n)$$

$$\text{then}\, |X(0)|^2=v_0^2(4)+v_0^2(3)-2v_0(4)v_0(3).$$

The digital filtering is performed as

$$v_0(0)=2v_0(-1)-v_0(-2)+x(0)=0+0+1=1$$

$$v_0(1)=2v_0(0)-v_0(-1)+x(1)=2\times 1+0+2=4$$

$$v_0(2)=2v_0(1)-v_0(0)+x(2)=2\times 4-1+3=10$$

$$v_0(3)=2v_0(2)-v_0(1)+x(3)=2\times 10-4+4=20$$

$$v_0(4)=2v_0(3)-v_0(2)+x(4)=2\times 20-10+0=30.$$

Then the squared magnitude is determined by

$$|X(0)|^2=v_0^2(4)+v_0^2(3)-2v_0(4)v_0(3)=(30)^2+(20)^2-2\times 30\times 20=100.$$

Thus, the amplitude spectrum is computed as

$$A_0=\frac{1}{4}\sqrt{\left(|X(0)|^2\right)}=2.5.$$

We can write a MATLAB function for the Geortzel algorithm shown in Program 8.20:

Program 8.20. MATLAB function for Geortzel Algorithm.

```
function [ Xk, Ak] = galg(x,k)
% Geortzel Algorithm
% [ Xk, Ak] = galg(x,k)
% x=input vetcor; k=frequency index
% Xk= kth DFT coeficient; Ak=magnitude of the kth DFT coefficient
N=length(x); x=[x 0];
vk=zeros(1,N+3);
for n=1:N+1
   vk(n+2)=2*cos(2*pi*k/N)*vk(n+1)-vk(n)+x(n);
end
Xk=vk(N+3)-exp(-2*pi*j*k/N)*vk(N+2);
Ak=vk(N+3)*vk(N+3)+vk(N+2)*vk(N+2)-2*cos(2*pi*k/N)*vk(N+3)*vk(N+2);
Ak=sqrt(Ak)/N;
End
```

EXAMPLE 8.28

Use MATLAB function (Program 8.20) to verify the results in Examples 8.26 and 8.27.

Solution:

(a) For Example 8.26, we obtain

```
>> x=[1 2 3 4]
x = 1 2 3 4
>> [X1, A1]=galg(x,1)
X1 = –2.0000 + 2.0000i
A1 =0.7071
```

(b) For Example 8.27, we obtain

```
>> x=[1 2 3 4]
x = 1 2 3 4
>> [X0, A0]=galg(x,1)
X0 = 10
A0 = 2.5000
```

8.11.4 DUAL-TONE MULTIFREQUENCY TONE DETECTION USING THE MODIFIED GOERTZEL ALGORITHM

Based on the specified frequencies of each DTMF tone shown in Fig. 8.48 and the modified Goertzel algorithm, we can develop the following design principles for DTMF tone detection:

1. When a digitized DTMF tone $x(n)$ is received, it has two nonzero frequency components from the following seven numbers: 697, 770, 852, 941, 1209, 1336, and 1477 Hz.
2. We can apply the modified Goertzel algorithm to compute seven spectral values, which correspond to the seven frequencies in (1). The single-sided amplitude spectrum is computed as

$$A_k = \frac{2}{N}\sqrt{|X(k)|^2}. \tag{8.89}$$

Table 8.12 DTMF Frequencies and Their Frequency Bins

DTMF Frequency (Hz)	Frequency bin: $k = \frac{f}{f_s} \times N$
697	18
770	20
852	22
941	24
1209	31
1336	34
1477	38

Since the modified Goertzel algorithm is used, there is no complex algebra involved.

3. Ideally, there are two nonzero spectral components. We will use these two nonzero spectral components to determine which key is pressed.

4. The frequency bin number (frequency index) can be determined based on the sampling rate f_s, and the data size of N via the following relation:

$$k = \frac{f}{f_s} \times N \text{ (round off to an integer)}. \tag{8.90}$$

Given the key frequency specification in Table 8.12, we can determine the frequency bin k for each DTMF frequency with $f_s = 8000$ Hz and $N = 205$. The results are summarized in Table 8.12. The DTMF detector block diagram is shown in Fig. 8.55.

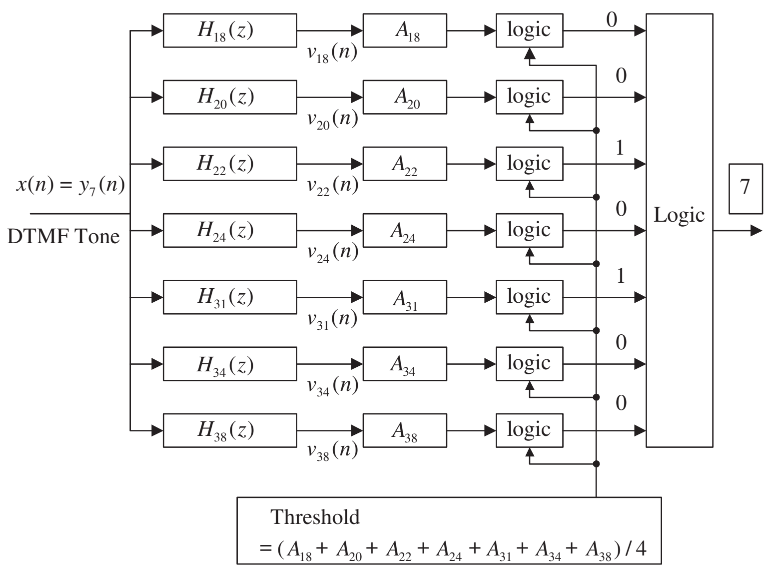

FIG. 8.55

DTMF detector using the Goertzel algorithm.

5. The threshold value can be the sum of all seven spectral values divided by a factor of 4. Note that there are only two nonzero spectral values and the others are zeros; hence, the threshold value should ideally be half of the individual nonzero spectral value. If the detected spectral value is larger than the threshold value, then the logic operation outputs logic 1; otherwise, it outputs logic 0. Finally, the logic operation at the last stage is to decode the key information based on the 7-bit binary pattern.

EXAMPLE 8.29

Given a DSP system with $f_s = 8000\,\text{Hz}$ and data size $N = 205$, seven Goertzel IIR filters are implemented for DTMF tone detection.

Determine the following for the frequencies corresponding to key 7:

1. Frequency bin numbers
2. The Goertzel filter transfer functions and DSP equations
3. Equations for calculating amplitude spectral values

Solution:

For key 7, we have $f_L = 852\,\text{Hz}$ and $f_H = 1209\,\text{Hz}$.

1. Using Eq. (8.90), we get

$$k_L = \frac{852}{8{,}000} \times 205 \approx 22, \text{and } k_H = \frac{1{,}209}{8{,}000} \times 205 \approx 31.$$

2. Since $2\cos\left(\frac{2\pi \times 22}{205}\right) = 1.5623$, and $2\cos\left(\frac{2\pi \times 31}{205}\right) = 1.1631$, it follows that

$$H_{22}(z) = \frac{1}{1 - 1.5623z^{-1} + z^{-2}}$$

and

$$H_{31}(z) = \frac{1}{1 - 1.1631z^{-1} + z^{-2}}.$$

The DSP equations are therefore given by

$$v_{22}(n) = 1.5623v_{22}(n-1) - v_{22}(n-2) + x(n) \text{ with } x(205) = 0, \text{ for } n = 0, 1, \cdots, 205$$
$$v_{31}(n) = 1.1631v_{31}(n-1) - v_{31}(n-2) + x(n) \text{ with } x(205) = 0, \text{ for } n = 0, 1, \cdots, 205.$$

3. The amplitude spectral values are determined by

$$|X(22)|^2 = (v_{22}(205))^2 + (v_{22}(204))^2 - 1.5623(v_{22}(205)) \times (v_{22}(204))$$

$$A_{22} = \frac{2\sqrt{|X(22)|^2}}{205}$$

and

$$|X(31)|^2 = (v_{31}(205))^2 + (v_{31}(204))^2 - 1.1631(v_{31}(205)) \times (v_{31}(204))$$

$$A_{31} = \frac{2\sqrt{|X(31)|^2}}{205}.$$

The MATLAB simulation for decoding the key 7 is shown in Program 8.21. Fig. 8.56A shows the frequency responses of the second-order Goertzel bandpass filters. The input is generated as shown in Fig. 8.52. After filtering, the calculated spectral values and threshold value for decoding key 7 are displayed in Fig. 8.56B, where only two spectral values corresponding to the frequencies of 770 and 1209 Hz are above the threshold, and are encoded as logic 1. According to the key information in Fig. 8.55, the final logic operation decodes the key as 7.

The principle can easily be extended to transmit the ASCII (American Standard Code for Information Interchange) code or other types of code using the parallel Goertzel filter bank.

If the calculated spectral value is larger than the threshold value, then the logic operation outputs logic 1; otherwise, it outputs logic 0. Finally, the logic operation at the last stage decodes the key information based on the 7-bit binary pattern.

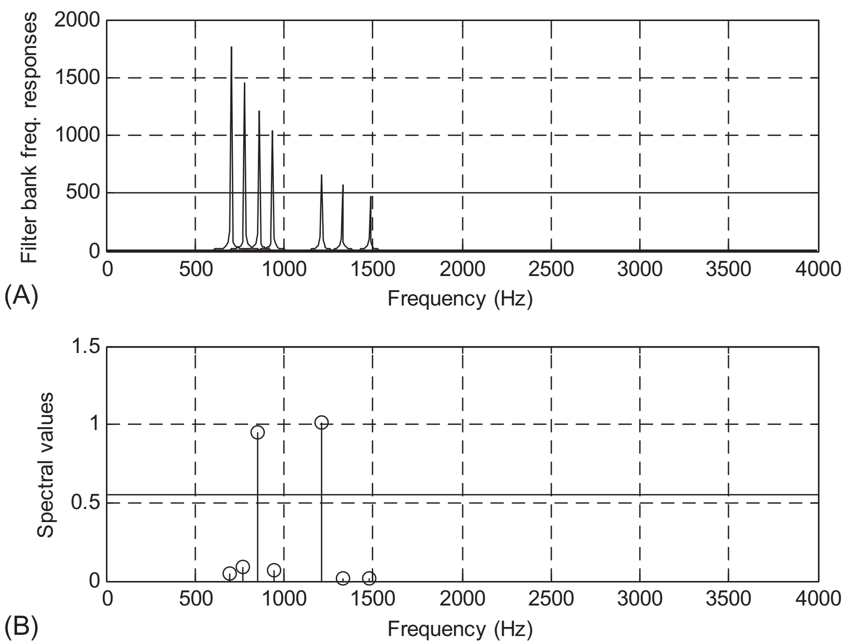

FIG. 8.56

(A) Goertzel filter bank frequency responses; (B) Display of spectral values and threshold for key 7.

Program 8.21. DTMF detection using the Goertzel algorithm.

```
close all;clear all;
% DTMF tone generator
N=205;
fs=8000; t=[0:1:N-1]/fs;            % Sampling rate and time vector
x=zeros(1,length(t));x(1)=1;        % Generate the impulse function
%Generation of tones
y697=filter([0 sin(2*pi*697/fs)][1 -2*cos(2*pi*697/fs) 1],x);
y770=filter([0 sin(2*pi*770/fs)][1 -2*cos(2*pi*770/fs) 1],x);
y852=filter([0 sin(2*pi*852/fs)][1 -2*cos(2*pi*852/fs) 1],x);
y941=filter([0 sin(2*pi*941/fs)][1 -2*cos(2*pi*941/fs) 1],x);
y1209=filter([0 sin(2*pi*1209/fs) ][1 -2*cos(2*pi*1209/fs) 1],x);
y1336=filter([0 sin(2*pi*1336/fs)][1 -2*cos(2*pi*1336/fs) 1],x);
y1477=filter([0 sin(2*pi*1477/fs)][1 -2*cos(2*pi*1477/fs) 1],x);
key=input('input of the following keys: 1,2,3,4,5,6,7,8,9,*,0,# =>','s');
yDTMF=[];
if key=='1' yDTMF=y697+y1209; end
if key=='2' yDTMF=y697+y1336; end
if key=='3' yDTMF=y697+y1477; end
if key=='4' yDTMF=y770+y1209; end
if key=='5' yDTMF=y770+y1336; end
if key=='6' yDTMF=y770+y1477; end
if key=='7' yDTMF=y852+y1209; end
if key=='8' yDTMF=y852+y1336; end
if key=='9' yDTMF=y852+y1477; end
if key=='*' yDTMF=y941+y1209; end
if key=='0' yDTMF=y941+y1336; end
if key=='#' yDTMF=y941+y1477; end
if size(yDTMF)==0 disp('Invalid input key'); return; end
yDTMF=[yDTMF 0];        % DTMF signal appended with a zero
% DTMF detector (use Goertzel algorithm)
a697=[1 -2*cos(2*pi*18/N) 1];
a770=[1 -2*cos(2*pi*20/N) 1];
a852=[1 -2*cos(2*pi*22/N) 1];
a941=[1 -2*cos(2*pi*24/N) 1];
a1209=[1 -2*cos(2*pi*31/N) 1];
a1336=[1 -2*cos(2*pi*34/N) 1];
a1477=[1 -2*cos(2*pi*38/N) 1];
% Filter bank frequency responses
[w1, f]=freqz(1,a697,512,fs);
[w2, f]=freqz(1,a770,512,fs);
[w3, f]=freqz(1,a852,512,fs);
[w4, f]=freqz(1,a941,512,fs);
[w5, f]=freqz(1,a1209,512,fs);
[w6, f]=freqz(1,a1336,512,fs);
[w7, f]=freqz(1,a1477,512,fs);
subplot(2,1,1);plot(f,abs(w1),f,abs(w2),f,abs(w3), ...
f,abs(w4),f,abs(w5),f,abs(w6),f,abs(w7));grid
xlabel('Frequency (Hz)'); ylabel('(a) Filter bank freq. responses');
% Filter bank bandpass filtering
```

```
y697=filter(1,a697,yDTMF);
y770=filter(1,a770,yDTMF);
y852=filter(1,a852,yDTMF);
y941=filter(1,a941,yDTMF);
y1209=filter(1,a1209,yDTMF);
y1336=filter(1,a1336,yDTMF);
y1477=filter(1,a1477,yDTMF);
% Determine the absolute magnitude of DFT coefficents
m(1)=sqrt(y697(206)^2+y697(205)^2- ...
  2*cos(2*pi*18/205)*y697(206)*y697(205));
m(2)=sqrt(y770(206)^2+y770(205)^2- ...
  2*cos(2*pi*20/205)*y770(206)*y770(205));
m(3)=sqrt(y852(206)^2+y852(205)^2- ...
  2*cos(2*pi*22/205)*y852(206)*y852(205));
m(4)=sqrt(y941(206)^2+y941(205)^2- ...
  2*cos(2*pi*24/205)*y941(206)*y941(205));
m(5)=sqrt(y1209(206)^2+y1209(205)^2- ...
  2*cos(2*pi*31/205)*y1209(206)*y1209(205));
m(6)=sqrt(y1336(206)^2+y1336(205)^2- ...
  2*cos(2*pi*34/205)*y1336(206)*y1336(205));
m(7)=sqrt(y1477(206)^2+y1477(205)^2- ...
  2*cos(2*pi*38/205)*y1477(206)*y1477(205));
% Convert the magnitude of DFT coefficients to the single-side spectrum
m=2*m/205;
% Determine the threshold
th=sum(m)/4;
% Plot the DTMF spectrum with the threshold
f=[ 697 770 852 941 1209 1336 1477];
f1=[0 fs/2];
th=[ th th];
subplot(2,1,2);stem(f,m);grid;hold; plot(f1,th);
xlabel('Frequency (Hz)'); ylabel(' (b) Spectral values');
m=round(m); % Round to the binary pattern
if m== [ 1 0 0 0 1 0 0] disp('Detected Key 1'); end
if m== [ 1 0 0 0 0 1 0] disp('Detected Key 2'); end
if m== [ 1 0 0 0 0 0 1] disp('Detected Key 3'); end
if m== [ 0 1 0 0 1 0 0] disp('Detected Key 4'); end
if m== [ 0 1 0 0 0 1 0] disp('Detected Key 5'); end
if m== [ 0 1 0 0 0 0 1] disp('Detected Key 6'); end
if m== [ 0 0 1 0 1 0 0] disp('Detected Key 7'); end
if m== [ 0 0 1 0 0 1 0] disp('Detected Key 8'); end
if m== [ 0 0 1 0 0 0 1] disp('Detected Key 9'); end
if m== [ 0 0 0 1 1 0 0] disp('Detected Key *'); end
if m== [ 0 0 0 1 0 1 0] disp('Detected Key 0'); end
if m== [ 0 0 0 1 0 0 1] disp('Detected Key #'); end
```

8.12 SUMMARY OF INFINITE IMPULSE RESPONSE (IIR) DESIGN PROCEDURES AND SELECTION OF THE IIR FILTER DESIGN METHODS IN PRACTICE

In this section, we first summarize the design procedures of the BLT design, impulse invariant design, and pole-zero placement design methods, and then discuss the selection of the particular filter for typical applications.

The BLT design method:

1. Given the digital filter frequency specifications, prewarp each of the digital frequency edge to the analog frequency edge using Eqs. (8.18) and (8.19).
2. Determine the prototype filter order using Eq. (8.29) for the Butterworth filter or Eq. (8.35b) for the Chebyshev filter, and perform lowpass prototype transformation using the lowpass prototype in Table 8.3 (Butterworth function) or Tables 8.4 and 8.5 (Chebyshev function) using Eqs. (8.20)–(8.23).
3. Apply the BLT to the analog filter using Eq. (8.24) and output the transfer function.
4. Verify the frequency responses, and output the difference equation.

The impulse invariant design method:

1. Given the lowpass or bandpass filter frequency specifications, perform analog filter design. For the highpass or bandstop filter design, quit this method and use the BLT.
   (a) Determine the prototype filter order using Eq. (8.29) for the Butterworth filter or Eq. (8.35b) for the Chebyshev filter.
   (b) Perform lowpass prototype transformation using the lowpass prototype in Table 8.3 (Butterworth function) or Tables 8.4 and 8.5 (Chebyshev functions) using Eqs. (8.20)–(8.23).
   (c) Skip step (1) if the analog filter transfer function is given to begin with.
2. Determine the impulse response by applying the partial fraction expansion technique to the analog transfer function and inverse Laplace transform using Eq. (8.37).
3. Sample the analog impulse response using Eq. (8.38) and apply the z-transform to the digital impulse function to obtain the digital filter transfer function.
4. Verify the frequency response, and output the difference equation. If the frequency specifications are not net, quit the design method and use the BLT.

The pole-zero placement method:

1. Given the filter cutoff frequency specifications, determine the pole-zero locations using the corresponding equations:
   (a) Second-order bandpass filter: Eqs. (8.43) and (8.44).
   (b) Second-order notch filter: Eqs. (8.47) and (8.48).
   (c) First-order lowpass filter: Eqs. (8.51) or (8.52).
   (d) First-order highpass filter: Eqs. (8.55) or (8.56).
2. Apply the corresponding equation and scale factor to obtain the digital filter transfer function:
   (a) Second-order bandpass filter: Eqs. (8.45) and (8.46).
   (b) Second-order notch filter: Eqs. (8.49) and (8.50).
   (c) First-order lowpass filter: Eqs. (8.53) and (8.54).
   (d) First-order highpass filter: Eqs. (8.57) and (8.58).

3. Verify the frequency response, and output the difference equation. If the frequency specifications are not net, quit the design method and use BLT.

Table 8.13 compares the design parameters of the three design methods.

BLT = bilinear transformation; LPF = lowpass filter; BPF = bandpass filter; HPF = highpass filter.

Performance comparisons using the three methods are displayed in Fig. 8.57, where the bandpass filter is designed using the following specifications:

Passband ripple = −3 dB
Center frequency = 400 Hz
Bandwidth = 200 Hz
Sampling rate = 2000 Hz
Butterworth IIR filter = second-order

As expected, the BLT method satisfies the design requirement, and the pole-zero placement method has little performance degradation because $r = 1-(f_0/f_s)\pi = 0.6858 < 0.9$, and this effect will also cause the center frequency to be shifted. For the bandpass filter designed using the impulse invariant method, the gain at the center frequency is scaled to 1 for a frequency response shape comparison. The performance of the impulse invariant method is satisfied in passband. However, it has significant performance degradation in stopband when compared with the other two methods. This is due to aliasing when sampling the analog impulse response in time domain. The advanced method (Nelatury, 2007), which incorporates the frequency prewarping technique, can be applied to correct the aliasing effect.

Table 8.13 Comparisons of Three IIR Design Methods

	Design Method		
	Bilinear Transformation	Impulse Invariant	Pole-zero Placement
Filter type	Lowpass, highpass, bandpass, bandstop	Appropriate for lowpass and bandpass	Second-order for bandpass and bandstop; first-order for lowpass and highpass
Linear phase	No	No	No
Ripple and stopband specifications	Used for determining the filter order	Used for determining the filter order	Not required; 3 dB on passband offered.
Special requirement	None	Very high sampling relative to the cutoff frequency (LPF) or to upper cutoff frequency (BPF)	narrow band for BPF or Notch filter; lower cutoff frequency or higher cutoff frequency for LPF or HPF.
Algorithm complexity	High: Frequency prewarping, analog filter design, BLT	Moderate: Analog filter design determining digital impulse response. Apply z-transform	Simple: Simple design equations
Minimum design tool	Calculator, algebra	Calculator, algebra	Calculator

BLT, bilinear transformation; *LPF*, *lowpass filter*; *BPF*, *band pass filter*; *HPF*, *highpass filter*.

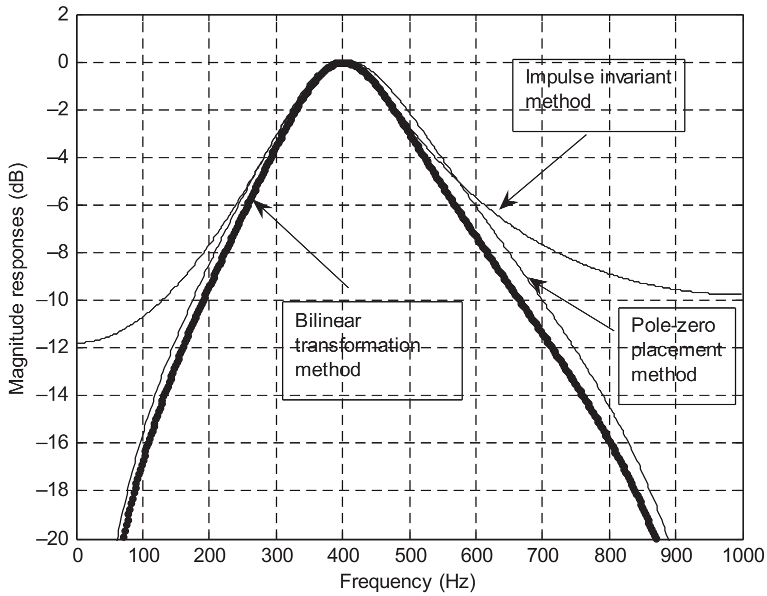

FIG. 8.57

Performance comparisons for the BLT, pole-zero placement, and impulse invariant methods.

Improvement in using the pole-zero placement and impulse invariant methods can be achieved by using a very high sampling rate. Example 8.30 describes the possible selection of the design method by a DSP engineer to solve the real-world problem.

EXAMPLE 8.30

Determine an appropriate IIR filter design method for each of the following DSP applications. As described in the previous section, we apply a notch filter to remove the 60-Hz interference and cascade a bandpass filter to remove noise in an ECG signal for heart rate detection. The following specifications are required:

Notch filter:
Harmonics to be removed = 60 Hz
3-dB bandwidth for the notch filter = 4 Hz
Bandpass filter:
Passband frequency range = 0.25–40 Hz
Passband ripple = 0.5 dB
Sampling rate = 600 Hz.

The pole-zero placement method is the best choice, since the notch filter to be designed has a very narrow 3-dB bandwidth of 4 Hz. This simple design gives a quick solution. Since the bandpass filter requires a passband ripple of 0.5 dB from 0.25 to 40 Hz, the BLT can also be an appropriate choice. Even though the impulse invariant method could work for this case, since the sampling rate of 600 Hz is much larger than 40 Hz, aliasing cannot be prevented completely. Hence, the BLT is a preferred design method for the bandpass filter.

8.13 SUMMARY

1. The BLT method is able to transform the transfer function of an analog filter to the transfer function of the corresponding digital filter in general.
2. The BLT maps the left half of an s-plane to the inside unit circle of the z-plane. Stability of mapping is guaranteed.
3. The BLT causes analog frequency warping. The analog frequency range from 0 Hz to infinite is warped to a digital frequency range from 0 Hz to the folding frequency.
4. Given the digital frequency specifications, analog filter frequency specifications must be developed using the frequency warping equation before designing the corresponding analog filter and applying the BLT.
5. An analog filter transfer function can be obtained by lowpass prototype, which can be selected from the Butterworth and Chebyshev functions.
6. The higher-order IIR filter can be designed using a cascade form.
7. The impulse invariant design method maps the analog impulse response to the digital equivalent impulse response. The method works for lowpass and bandpass filter design with a very high sampling rate. It is not appropriate for the highpass and bandstop filter design.
8. The pole-zero placement method can be applied for a simple IIR filter design such as the second-order bandpass and bandstop filters with narrow band specifications, first-order lowpass and highpass filters with the cutoff frequency close to either DC or the folding frequency.
9. Quantizing IIR filter coefficients explore the fact that the quantization of the filter coefficients has more effect on the magnitude frequency response than on the phase frequency response. It may cause the quantized IIR filter to be unstable.
10. A simple audio equalizer uses bandpass IIR filter banks to create sound effects.
11. The 60-Hz interference eliminator is designed to enhance biomedical ECG signals for heart rate detection. It can also be adapted for audio humming noise elimination.
12. A single tone or a DTMF tone can be generated using the IIR filter with the impulse sequence as the filter input.
13. The Goertzel algorithm is derived and applied for DTMF tone detection. This is an important application in the telecommunication industry.
14. The procedures for the BLT, impulse invariant, and pole-zero placement design methods were summarized, and their design feasibilities were compared, including the filter type, linear phase, ripple and stopband specifications, special requirements, algorithm complexity, and design tool(s).

8.14 PROBLEMS

8.1 Given an analog filter with the transfer function

$$H(s) = \frac{1000}{s + 1000},$$

convert it to the digital filter transfer function and difference equation using the BLT if the DSP system has a sampling period of $T = 0.001$ s.

8.2 The lowpass filter with a cutoff frequency of 1 rad/s is given as

$$H_P(s) = \frac{1}{s+1}.$$

(a) Use $H_P(s)$ and the BLT to obtain a corresponding IIR digital lowpass filter with a cutoff frequency of 30 Hz, assuming a sampling rate of 200 Hz.
(b) Use MATLAB to plot the magnitude and phase frequency responses of $H(z)$.

8.3 The normalized lowpass filter with a cutoff frequency of 1 rad/s is given as

$$H_P(s) = \frac{1}{s+1}.$$

(a) Use $H_P(s)$ and the BLT to obtain a corresponding IIR digital highpass filter with a cutoff frequency of 30 Hz, assuming a sampling rate of 200 Hz.
(b) Use MATLAB to plot the magnitude and phase frequency responses of $H(z)$.

8.4 Consider the normalized lowpass filter with a cutoff frequency of 1 rad/s:

$$H_P(s) = \frac{1}{s+1}.$$

(a) Use $H_P(s)$ and the BLT to design a corresponding IIR digital notch (bandstop) filter with a lower cutoff frequency of 20 Hz, an upper cutoff frequency of 40 Hz, and a sampling rate of 120 Hz.
(b) Use MATLAB to plot the magnitude and phase frequency responses of $H(z)$.

8.5 Consider the following normalized lowpass filter with a cutoff frequency of 1 rad/s:

$$H_P(s) = \frac{1}{s+1}.$$

(a) Use $H_P(s)$ and the BLT to design a corresponding IIR digital bandpass filter with a lower cutoff frequency of 15 Hz, an upper cutoff frequency of 25 Hz, and a sampling rate of 120 Hz.
(b) Use MATLAB to plot the magnitude and phase frequency responses of $H(z)$.

8.6 Design a first-order digital lowpass Butterworth filter with a cutoff frequency of 1.5 kHz and a passband ripple of 3 dB at a sampling frequency of 8000 Hz.
(a) Determine the transfer function and difference equation.
(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.7 Design a second-order digital lowpass Butterworth filter with a cutoff frequency of 1.5 kHz and a passband ripple of 3 dB at a sampling frequency of 8000 Hz.
(a) Determine the transfer function and difference equation.
(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.8. Design a third-order digital highpass Butterworth filter with a cutoff frequency of 2 kHz and a passband ripple of 3 dB at a sampling frequency of 8000 Hz.
(a) Determine the transfer function and difference equation.
(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.9 Design a second-order digital bandpass Butterworth filter with a lower cutoff frequency of 1.9 kHz, an upper cutoff frequency 2.1 kHz, and a passband ripple of 3 dB at a sampling frequency of 8000 Hz.
(a) Determine the transfer function and difference equation.
(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.10 Design a second-order digital bandstop Butterworth filter with a center frequency of 1.8 kHz, a bandwidth of 200 Hz, and a passband ripple of 3 dB at a sampling frequency of 8000 Hz.

(a) Determine the transfer function and difference equation.

(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.11 Design a first-order digital lowpass Chebyshev filter with a cutoff frequency of 1.5 kHz and 1 dB ripple on passband at a sampling frequency of 8000 Hz.

(a) Determine the transfer function and difference equation.

(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.12 Design a second-order digital lowpass Chebyshev filter with a cutoff frequency of 1.5 kHz and 0.5 dB ripple on passband at a sampling frequency of 8000 Hz. Use MATLAB to plot the magnitude and phase frequency responses.

8.13 Design a third-order digital highpass Chebyshev filter with a cutoff frequency of 2 kHz and 1 dB ripple on the passband at a sampling frequency of 8000 Hz.

(a) Determine the transfer function and difference equation.

(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.14 Design a second-order digital bandpass Chebyshev filter with the following specifications:
Center frequency of 1.5 kHz
Bandwidth of 200 Hz
0.5-dB ripple on passband
Sampling frequency of 8000 Hz.

(a) Determine the transfer function and difference equation.

(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.15 Design a second-order bandstop digital Chebyshev filter with the following specifications:
Center frequency of 2.5 kHz
Bandwidth of 200 Hz
1-dB ripple on stopband
Sampling frequency of 8000 Hz.

(a) Determine the transfer function and difference equation.

(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.16 Design a fourth-order lowpass digital Butterworth filter with a cutoff frequency of 2 kHz, and the passband ripple of 3 dB at a sampling frequency at 8000 Hz.

1. Determine transfer function and difference equation.

2. Use MATLAB to plot the magnitude and phase frequency responses.

8.17 Design a fourth-order digital lowpass Chebyshev filter with a cutoff frequency of 1.5 kHz and a 0.5-dB ripple at a sampling frequency of 8000 Hz.

(a) Determine the transfer function and difference equation.

(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.18 Design a fourth-order digital bandpass Chebyshev filter with a center frequency of 1.5 kHz, a bandwidth of 200 Hz, and a 0.5-dB ripple at a sampling frequency of 8000 Hz.

(a) Determine the transfer function and difference equation.

(b) Use MATLAB to plot the magnitude and phase frequency responses.

8.19 Consider the following Laplace transfer function:

$$H(s) = \frac{10}{s+10}.$$

(a) Determine $H(z)$ and the difference equation using the impulse invariant method if the sampling rate $f_s = 10\,\text{Hz}$.

(b) Use MATLAB to plot the magnitude frequency response $|H(f)|$ and the phase frequency response $\phi(f)$ with respect to $H(s)$ for the frequency range from 0 to $f_s/2\,\text{Hz}$.

(c) Use MATLAB to plot the magnitude frequency response $|H(e^{j\Omega})| = |H(e^{j2\pi fT})|$ and the phase frequency response $\phi(f)$ with respect to $H(z)$ for the frequency range from 0 to $f_s/2\,\text{Hz}$.

8.20 Consider the following Laplace transfer function:

$$H(s) = \frac{1}{s^2 + 3s + 2}.$$

(a) Determine $H(z)$ and the difference equation using the impulse invariant method if the sampling rate $f_s = 10\,\text{Hz}$.

(b) Use MATLAB to plot the magnitude frequency response $|H(f)|$ and the phase frequency response $\phi(f)$ with respect to $H(s)$ for the frequency range from 0 to $f_s/2\,\text{Hz}$.

(c) Use MATLAB to plot the magnitude frequency response $|H(e^{j\Omega})| = |H(e^{j2\pi fT})|$ and the phase frequency response $\phi(f)$ with respect to $H(z)$ for the frequency range from 0 to $f_s/2\,\text{Hz}$.

8.21 Consider the following Laplace transfer function:

$$H(s) = \frac{s}{s^2 + 4s + 5}.$$

(a) Determine $H(z)$ and the difference equation using the impulse invariant method if the sampling rate $f_s = 10\,\text{Hz}$.

(b) Use MATLAB to plot the magnitude frequency response $|H(f)|$ and the phase frequency response $\phi(f)$ with respect to $H(s)$ for the frequency range from 0 to $f_s/2\,\text{Hz}$;

(c) Use MATLAB to plot the magnitude frequency response $|H(e^{j\Omega})| = |H(e^{j2\pi fT})|$ and the phase frequency response $\phi(f)$ with respect to $H(z)$ for the frequency range from 0 to $f_s/2\,\text{Hz}$.

8.22 A second-order bandpass filter is required to satisfy the following specifications:
Sampling rate = 8000 Hz
3-dB bandwidth: $BW = 100\,\text{Hz}$ Hz
Narrow passband centered at $f_0 = 2000\,\text{Hz}$
Zero gain at 0 and 4000 Hz.
Find the transfer function and difference equation by the pole-zero placement method.

8.23 A second-order notch filter is required to satisfy the following specifications:
Sampling rate = 8000 Hz
3-dB bandwidth: $BW = 200\,\text{Hz}$
Narrow passband centered at $f_0 = 1000\,\text{Hz}$.
Find the transfer function and difference equation by the pole-zero placement method.

8.24 A first-order lowpass filter is required to satisfy the following specifications:
Sampling rate = 8000 Hz
3-dB cutoff frequency: $f_c = 200\,\text{Hz}$
Zero gain at 4000 Hz.
Find the transfer function and difference equation using the pole-zero placement method.

8.25 A first-order lowpass filter is required to satisfy the following specifications:
Sampling rate = 8000 Hz
3-dB cutoff frequency: $f_c = 3800\,\text{Hz}$

Zero gain at 4000 Hz.
Find the transfer function and difference equation by the pole-zero placement method.

8.26 A first-order highpass filter is required to satisfy the following specifications:
Sampling rate = 8000 Hz
3-dB cutoff frequency: $f_c = 3850$ Hz
Zero gain at 0 Hz.
Find the transfer function and difference equation by the pole-zero placement method.

8.27 A first-order highpass filter is required to satisfy the following specifications:
Sampling rate = 8000 Hz
3-dB cutoff frequency: $f_c = 100$ Hz
Zero gain at 0 Hz.
Find the transfer function and difference equation by the pole-zero placement method.

8.28 Given a filter transfer function

$$H(z) = \frac{0.3430z^2 + 0.6859z + 0.3430}{z^2 + 0.7075z + 0.7313},$$

(a) Realize the digital filter using direct-form I and direct-form II.
(b) Determine the difference equations for each implementation.

8.29 Given a fourth-order filter transfer function

$$H(z) = \frac{0.3430z^2 + 0.6859z + 0.3430}{z^2 + 0.7075z + 0.7313} \times \frac{0.4371z^2 + 0.8742z + 0.4371}{z^2 - 0.1316z + 0.1733},$$

(a) Realize the digital filter using the cascade (series) form via second-order sections using the direct-form II.
(b) Determine the difference equations for implementation.

8.30 Given a DSP system with a sampling rate of 1000 Hz, develop a 200-Hz single tone generator using the digital IIR filter by completing the following steps:
(a) Determine the digital IIR filter transfer function.
(b) Determine the DSP equation (difference equation).

8.31 Given a DSP system with a sampling rate of 8000 Hz, develop a 250-Hz single tone generator using the digital IIR filter by completing the following steps:
(a) Determine the digital IIR filter transfer function.
(b) Determine the DSP equation (difference equation).

8.32 Given a DSP system with a sampling rate of 8000 Hz, develop a DTMF tone generator for key 9 using the digital IIR filters by completing the following steps:
(a) Determine the digital IIR filter transfer functions.
(b) Determine the DSP equations (difference equation).

8.33 Given a DSP system with a sampling rate 8000 Hz, develop a DTMF tone generator for key 3 using the digital IIR filters by completing the following steps:
(a) Determine the digital IIR filter transfer functions.
(b) Determine the DSP equations (difference equation).

8.34 Given $x(0) = 1$, $x(1) = 2$, $x(2) = 0$, $x(3) = -1$, using the Goertzel algorithm to compute the following DFT coefficients and their amplitude spectrum:

(a) $X(0)$
(b) $|X(0)|^2$
(c) A_0 (single side)
(d) $X(1)$
(e) $|X(1)|^2$
(f) A_1 (single side)

8.35 Repeat Problem 8.34 for $X(1)$ and $X(3)$.

8.36 Given the digital data sequence of length 4 as $x(0)=4$, $x(1)=3$, $x(2)=2$, and $x(3)=1$, use the modified Goertzel algorithm to compute the spectral amplitude at the frequency bin $k=0$ and $k=2$.

8.37 Repeat Problem 8.36 for $X(1)$ and $X(3)$.

Use **MATLAB** to solve Problems 8.38–8.50.

8.38 A speech sampled at 8000 Hz is corrupted by a sine wave of 360 Hz. Design a notch filter to remove the noise with the following specifications:
Chebyshev notch filter
Center frequency: 360 Hz
Bandwidth: 60 Hz
Passband ripple: 0.5 dB
Stopband attenuation: 5 dB at 355 and 365 Hz, respectively
Determine the transfer function and difference equation.

8.39 In Problem 8.38, if the speech is corrupted by a sine wave of 360 Hz and its third harmonics, cascading two notch filters can be applied to remove noise signals. The possible specifications are given as follows:
Chebyshev notch filter 1
Center frequency: 360 Hz
Bandwidth: 60 Hz
Passband ripple: 0.5 dB
Stopband attenuation: 5 dB at 355 and 365 Hz, respectively
Chebyshev notch filter 2
Center frequency: 1080 Hz
Bandwidth: 60 Hz
Passband and ripple: 0.5 dB
Stopband attenuation: 5 dB at 1075 and 1085 Hz, respectively
Determine the transfer function and difference equation for each filter (Fig. 8.58).

8.40 In a speech recording system with a sampling frequency of 10,000 Hz, the speech is corrupted by random noise. To remove the random noise while preserving speech information, the following specifications are given:

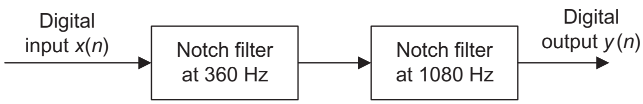

FIG. 8.58

Cascaded notch filter in Problem 8.39.

Speech frequency range: 0–3000 Hz
Stopband range: 4000–5000 Hz
Passband ripple: 3 dB
Stopband attenuation: 25 dB
Butterworth IIR filter
Determine the filter order and transfer function.

8.41 In Problem 8.40, if we use a Chebyshev IIR filter with the following specifications:
Speech frequency range: 0–3000 Hz
Stopband range: 4000–5000 Hz
Passband ripple: 1 dB
Stopband attenuation: 35 dB
Chebyshev IIR filter
Determine the filter order and transfer function.

8.42 Given a speech equalizer to compensate midrange frequency loss of hearing (Fig. 8.59) and the following specifications:
Sampling rate: 8000 Hz
Second-order bandpass IIR filter
Frequency range to be emphasized: 1500–2000 Hz
Passband ripple: 3 dB
Pole-zero placement design method
Determine the transfer function.

8.43 In Problem 8.42, if we use an IIR filter with following specifications:
Sampling rate: 8000 Hz
Butterworth IIR filter
Frequency range to be emphasized: 1500–2000 Hz
Lower stopband: 0–1000 Hz
Upper stopband: 2500–4000 Hz
Passband ripple: 3 dB
Stopband attenuation: 20 dB
Determine the filter order and filter transfer function.

8.44 A digital crossover can be designed as shown in Fig. 8.60.
Given audio specifications as
Sampling rate: 44,100 Hz
Crossover frequency: 1000 Hz
Highpass filter: third-order Butterworth type at a cutoff frequency of 1000 Hz
Lowpass filter: third-order Butterworth type at a cutoff frequency of 1000 Hz

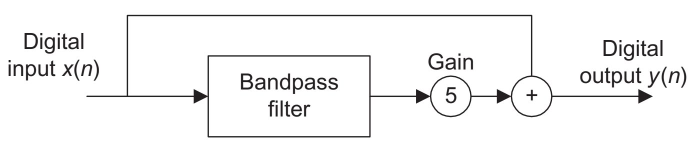

FIG. 8.59

Speech equalizer in Problem 8.42.

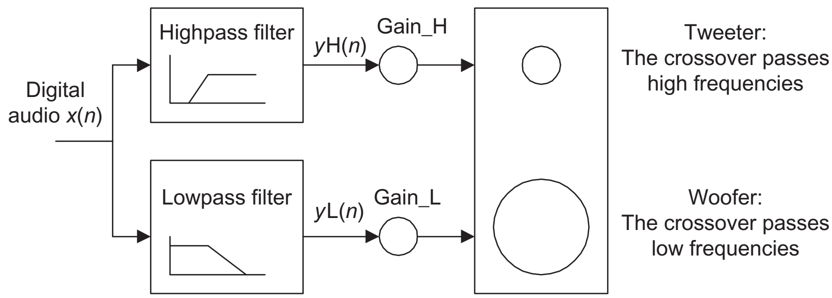

FIG. 8.60

Two-band digital crossover system in Problem 8.44.

Use the MATLAB BLT design method to determine:

(a) the transfer functions and difference equations for the highpass and lowpass filters;

(b) frequency responses for the highpass filter and the lowpass filter;

(c) combined frequency response for both filters.

8.45 Given a DSP system with a sampling rate of 8000 Hz, develop an 800 Hz single-tone generator using a digital IIR filter by completing the following steps:

(a) Determine the digital IIR filter transfer function.

(b) Determine the DSP equation (difference equation).

(c) Write a MATLAB program using the MATLAB function filter() to generate and plot the 800-Hz tone for a duration of 0.01 s.

8.46 Given a DSP system with a sampling rate set up to be 8000 Hz, develop a DTMF tone generator for key "5" using digital IIR filters by completing the following steps:

(a) Determine the digital IIR filter transfer functions.

(b) Determine the DSP equations (difference equation).

(c) Write a MATLAB program using the MATLAB function filter() to generate and plot the DTMF tone for key 5 for 205 samples.

8.47 Given $x(0)=1$, $x(1)=1$, $x(2)=0$, $x(3)=-1$, using the Goertzel algorithm to compute the following DFT coefficients and their amplitude spectra:

(a) $X(0)$

(b) $|X(0)|^2$

(c) A_0 (single sided)

(d) $X(1)$

(e) $|X(1)|^2$

(f) A_1 (single sided)

8.48 Repeat Problem 8.47 for single-side spectra: A_2 and A_3.

8.49 Given a DSP system with a sampling rate set up to be 8000 Hz and data size of 205 ($N=205$), seven Goertzel IIR filters are implemented for DTMF tone detection. For the frequencies corresponding to key 5, determine:

(a) the modified Goertzel filter transfer functions

(b) the filter DSP equations for $v_k(n)$

(c) the DSP equations for the squared magnitudes

$$|X(k)|^2 = |y_k(205)|^2$$

(d) using the data generated in Problem 8.46 (c), Write a program using the MATLAB function filter() and Goertzel algorithm to detect the spectral values of the DTMF tone for key 5.

8.50 Given an input data sequence:

$$x(n) = 1.2 \cdot \sin(2\pi(1000)n/10,000)) - 1.5 \cdot \cos(2\pi(4000)n/10,000)$$

assuming a sampling frequency of 10 kHz, implement the designed IIR filter in Problem 8.41 to filter 500 data points of $x(n)$ with the following specified method, and plot the 500 samples of the input and output data.

(a) Direct-form I implementation

(b) Direct-form II implementation.

MATLAB Projects

8.51 The 60-Hz hum eliminator with harmonics and heart rate detection

Given the recorded ECG data (ecgbn.dat) which is corrupted by 60-Hz interference with its harmonics and the sampling rate is 600 Hz, plot its spectrum and determine the harmonics. With the harmonic frequency information, design a notch filter to enhance the ECG signal. Then use the designed notch filter to process the given ECG signal and apply the zero-cross algorithm to determine the heart rate.

8.52 Digital speech and audio equalizer

Design a seven-band audio equalizer using fourth-order bandpass filters with a sampling rate of 44.1 kHz. The center frequencies are listed in Table 8.14.

In this project, use the designed equalizer to process a stereo audio ("No9seg.wav").

Plot the magnitude response for each filter bank.

Listen and evaluate the processed audio with the following gain settings:

(a) each filter bank gain = 0 (no equalization)

(b) lowpass filtered

(c) bandpass filtered

(d) highpass filtered

8.53 DTMF tone generation and detection

Implement the DTMF tone generation and detection according to Section 8.11 with the following specifications:

(a) Input keys: 1,2,3,4,5,6,7,8,9,*,0,#, A, B, C, D (key frequencies are given in Fig. 8.61).

(b) Sampling frequency is 8000 Hz.

(c) Program will respond each input key with its DTMF tone and display the detected key.

Table 8.14 Specification for Center Frequencies and Bandwidths

Center Frequency (Hz)	160	320	640	1280	2560	5120	10,240
Bandwidth (Hz)	80	160	320	640	1280	2560	5120

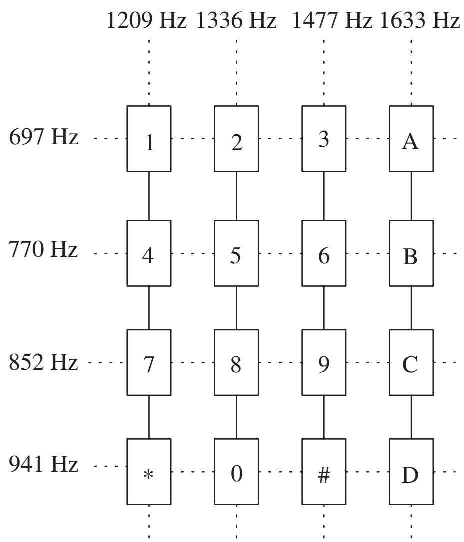

FIG. 8.61

DTMF key frequencies.

Advanced Problems

8.54 For the second-order IIR notch filter design using the pole-placement method show that the pole placement for 3-dB bandwidth is

$$\theta = \frac{f_0}{f_s} \times 360^{\circ}$$
$$r = 1 - \left(\frac{BW_{3dB}}{f_s}\right)\pi \text{ for } 0.9 < r < 1.$$

8.55 For the first-order IIR lowpass filter design with $f_s/4 < f_c < f_s/2$ using the pole-placement method show that the pole placement for 3-dB bandwidth is

$$\alpha = -\left(1 - \pi + \frac{2\pi f_c}{f_s}\right) \text{ for } -1 < \alpha < -0.9.$$

8.56 For the first-order IIR highpass filter design with $0 < f_c < f_s/4$ using the pole-placement method show that the pole placement for 3-dB bandwidth is

$$\alpha = 1 - 2\pi f_c / f_s \text{ for } 0.9 < \alpha < 1.$$

8.57 For the first-order IIR highpass filter design with $f_s/4 < f_c < f_s/2$ using the pole-placement method show that the pole placement for 3-dB bandwidth is

$$\alpha = -\left(1 - \pi + \frac{2\pi f_c}{f_s}\right) \text{ for } -1 < \alpha < -0.9.$$

8.58 Filter design by Padé approximation
Given the desired impulse response $h_d(n)$, $n \geq 0$ for an IIR filter, the IIR filter to be designed can be written as

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} = \sum_{k=0}^{\infty} h(k) z^{-k}.$$

Let $x(n) = \delta(n)$, the impulse response $y(n) = h(n)$ can be expressed as

$$h(n) = -a_1 h(n-1) - a_2 h(n-2) - \cdots - a_N h(n-N) + b_0 \delta(n) + b_1 \delta(n-1) + \cdots + b_M \delta(n-M)$$

(a) Show that for $0 \leq n \leq M$

$$h(n) = -a_1 h(n-1) - a_2 h(n-2) - \cdots - a_N h(n-N) + b_n$$

(b) Show that for $n > M$

$$h(n) = -a_1 h(n-1) - a_2 h(n-2) - \cdots - a_N h(n-N)$$

(c) Let $h(n)$ match $h_d(n)$ for $0 \leq n \leq N+M$, that is, $h(n) = h_d(n)$, for $0 \leq n \leq N+M$. Show that

$$\begin{bmatrix} -h_d(M) & -h_d(M-1) & \cdots & -h_d(M+1-N) \\ -h_d(M+1) & -h_d(M) & \cdots & -h_d(M+2-N) \\ \cdots & \cdots & \ddots & \cdots \\ -h_d(M+N-1) & -h_d(M+N-2) & \cdots & -h_d(M) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} h_d(M+1) \\ h_d(M+2) \\ \vdots \\ h_d(M+N) \end{bmatrix}$$

$$\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_M \end{bmatrix} = \begin{bmatrix} h_d(0) & 0 & \cdots & 0 \\ h_d(1) & h_d(0) & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots \\ h_d(M) & h_d(M-1) & \cdots & h_d(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_M \end{bmatrix}.$$

8.59 Assume the desired unit impulse response is

$$h_d(n) = \frac{1}{n+1} u(n).$$

Design an IIR filter using the Pade approximation method with the following form:

$$H(z) = \frac{\sum_{k=0}^{5} b_k z^{-k}}{1 + \sum_{k=1}^{5} a_k z^{-k}}.$$

8.60 Shanks methods for least-mean squares filter design

Given the desired impulse response $h_d(n)$, $n \geq 0$ for an IIR filter, the IIR filter to be designed can be written as

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}} = \sum_{k=0}^{\infty} h(k) z^{-k}.$$

Find b_k and a_k such that the sum of squared errors between $h_d(n)$ and $h(n)$ is minimized.

(a) From Padé approximation, we can set

$$\tilde{h}_d(n) = -\sum_{k=1}^{N} a_k h_d(n-k).$$

Show that the equations determine a_k by minimizing the following squared errors:

$$E_1 = \sum_{n=M+1}^{\infty} \left[h_d(n) - \tilde{h}_d(n)\right]^2,$$

That is, for $m = 1, 2, \cdots, N$

$$\sum_{k=1}^{N} a_k \sum_{n=M+1}^{\infty} h_d(n-k) h_d(n-m) = -\sum_{n=M+1}^{\infty} h_d(n) h_d(n-m).$$

(b) To determine b_k, we first split $H(z)$ into $H_1(z)$ and $H_2(z)$, that is, $H(z) = H_1(z)H_2(z)$ where

$$H_1(z) \frac{1}{1 + \sum_{k=1}^{N} \tilde{a}_k z^{-k}} \text{ and } H_2(z) = \sum_{k=0}^{M} b_k z^{-k}.$$

Let $v(n)$ be the impulse response of $H_1(z)$, that is,

$$v(n) = -\sum_{k=1}^{N} \tilde{a}_k v(n-k) + \delta(n).$$

Then

$$\hat{h}_d(n) = \sum_{k=0}^{M} b_k v(n-k).$$

After minimizing the sum of squared error

$$E_2 = \sum_{n=0}^{\infty} \left[h_d(n) - \hat{h}_d(n)\right]^2,$$

Show that for $m = 0, 1, \cdots, M$

$$\sum_{k=0}^{M} b_k \sum_{n=0}^{\infty} v(n-k)v(n-m) = \sum_{n=0}^{\infty} h_d(n)v(n-m).$$

8.61 Assume that the desired unit impulse response is

$$h_d(n) = 2\left(\frac{1}{2}\right)^n u(n).$$

Design an IIR filter using the Shanks method with the following form:

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}.$$