Design of infinite impulse response (IIR) digital filters

8

8.1	Introduction: summary of the basic features of IIR filters	455
8.2	Design stages for digital IIR filters	456
8.3	Performance specification	457
8.4	Coefficient calculation methods for IIR filters	459
8.5	Pole–zero placement method of coefficient calculation	459
8.6	Impulse invariant method of coefficient calculation	463
8.7	Matched *z*-transform (MZT) method of coefficient calculation	468
8.8	Bilinear *z*-transform (BZT) method of coefficient calculation	471
8.9	Use of BZT and classical analog filters to design IIR filters	482
8.10	Calculating IIR filter coefficients by mapping *s*-plane poles and zeros	500
8.11	Using IIR filter design programs	508
8.12	Choice of coefficient calculation methods for IIR filters	509
8.13	Realization structures for IIR digital filters	517
8.14	Finite wordlength effects in IIR filters	524
8.15	Implementation of IIR filters	529
8.16	A detailed design example of an IIR digital filter	530
8.17	Summary	535
8.18	Application examples in digital audio and instrumentation	536
8.19	Application examples in telecommunication	538
	Problems	549
	References	554
	Bibliography	555
	Appendices	557

This chapter presents practical design methods for digital infinite impulse response (IIR) filters, including popular methods which permit analog filters to be converted into equivalent digital filters. A simple but general step-by-step guide for designing digital IIR filters, from specifications to implementation, is described. Several fully worked examples are given to illustrate various aspects of digital IIR filter design, including the analysis of the effects of finite precision arithmetic on filter performance and real-time implementation.

A number of MATLAB and C language programs are provided to enable users to calculate filter coefficients and to carry out finite wordlength analysis. The reader is referred to Chapter 6 for a description of a general framework for filter design, comparison between IIR and FIR, and between digital and analog filters. In this chapter we will concentrate on IIR filter design and applications.

8.1 Introduction: summary of the basic features of IIR filters

Realizable IIR digital filters are characterized by the following recursive equation:

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^{N} b_k x(n-k) - \sum_{k=1}^{M} a_k y(n-k) \tag{8.1}$$

where $h(k)$ is the impulse response of the filter which is theoretically infinite in duration, b_k and a_k are the coefficients of the filter, and $x(n)$ and $y(n)$ are the input and output to the filter. The transfer function for the IIR filter is given by

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}}{1 + a_1 z^{-1} + \ldots + a_M z^{-M}} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}} \tag{8.2}$$

An important part of the IIR filter design process is to find suitable values for the coefficients b_k and a_k such that some aspect of the filter characteristic, such as frequency response, behaves in a desired manner. Equations 8.1 and 8.2 are the characteristic equations for IIR filters.

Note that, in Equation 8.1, the current output sample, $y(n)$, is a function of past outputs, $y(n-k)$, as well as present and past input samples, $x(n-k)$, that is the IIR filter is a feedback system of some sort. The strength of IIR filters comes from the flexibility the feedback arrangement provides. For example, an IIR filter normally requires fewer coefficients than an FIR filter for the same set of specifications, which is why IIR filters are used when sharp cutoff and high throughput are the important requirements. The price for this is that the IIR filter can become unstable or its performance significantly degraded if adequate care is not taken in its design.

The transfer function of the IIR filter, $H(z)$, given in Equation 8.2 can be factored as

$$H(z) = \frac{K(z - z_1)(z - z_2)\ldots(z - z_N)}{(z - p_1)(z - p_2)\ldots(z - p_M)} \tag{8.3}$$

where $z_1, z_2, \ldots$ are the zeros of $H(z)$, that is those values of z for which $H(z)$ becomes zero, and $p_1, p_2, \ldots$ are the poles of $H(z)$, that is values of z for which $H(z)$ is infinite.

A plot of the poles and zeros of the transfer function is known as the pole–zero diagram and provides a very useful way of representing and analyzing the filter in the complex z-plane; see Chapter 3 for details. For the filter to be stable, all its poles must lie inside the unit circle (or be coincident with zeros on the unit circle). There is no restriction on the zero locations.

8.2 Design stages for digital IIR filters

The design of IIR filters can be conveniently broken down into five main stages.

(1) Filter specification, at which stage the designer gives the function of the filter (for example, lowpass) and the desired performance.

(2) Approximation or coefficient calculation, where we select one of a number of methods and calculate the values of the coefficients, b_k and a_k, in the transfer function, $H(z)$, such that the specifications given in stage 1 are satisfied.

(3) Realization, which is simply converting the transfer function into a suitable filter structure. Typical structures for IIR filters are parallel and cascade of second and/or first-order filter sections.

(4) Analysis of errors that would arise from representing the filter coefficients and carrying out the arithmetic operations involved in filtering with only a finite number of bits.

(5) Implementation, which involves building the hardware and/or writing the software codes, and carrying out the actual filtering operation.

These stages are summarized in Figure 8.1. As indicated in the figure, the five stages are not independent and they are not always performed in the order given. In fact, techniques are now available that combine the second, third and fourth steps. However, the approach discussed here will ensure a successful design. To arrive at an efficient filter, it may be necessary to iterate a few times within and/or between the stages.

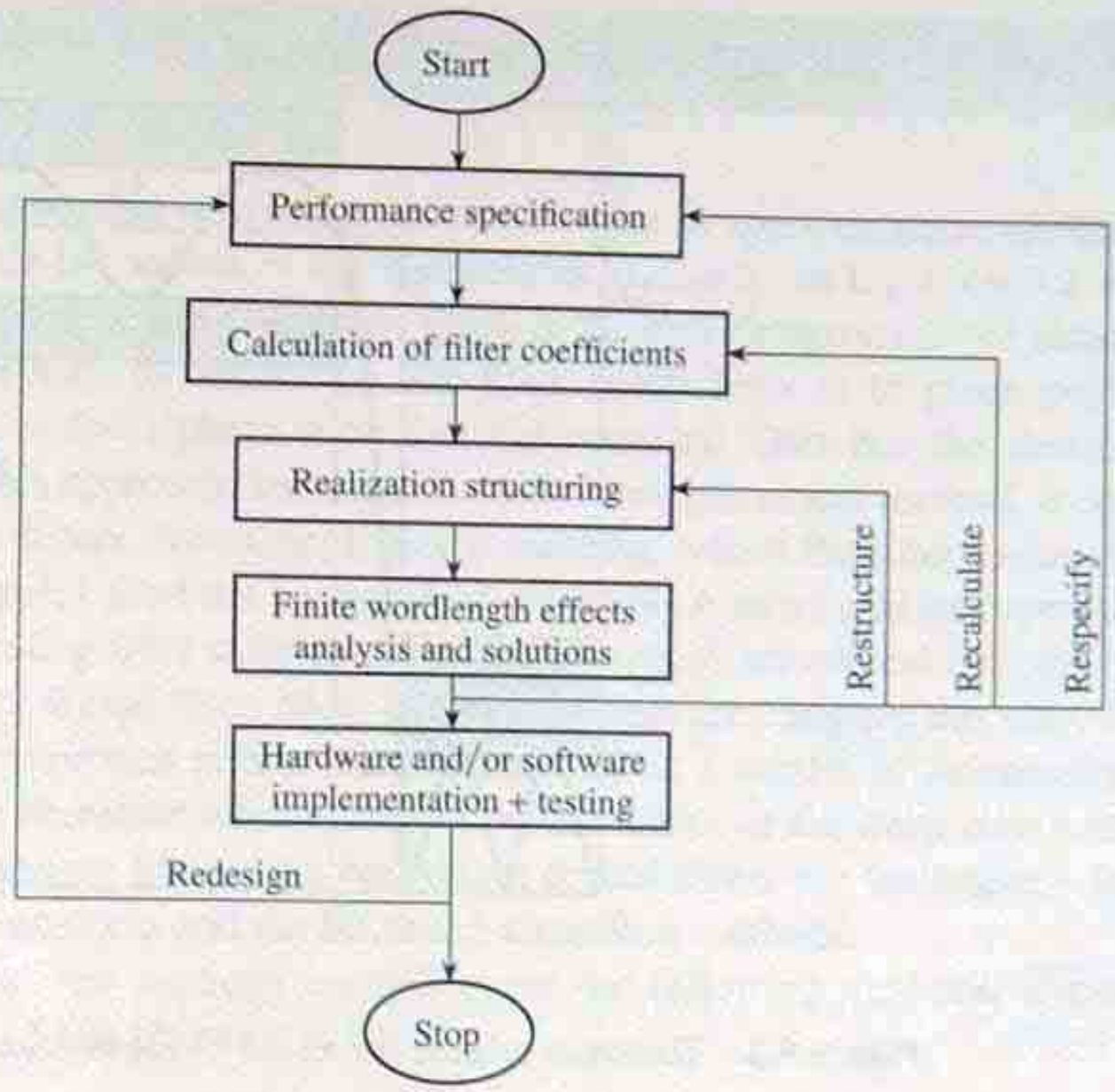


Figure 8.1 Summary of design stages for digital filters.

8.3 Performance specification

As with most other engineering problems, the design of digital IIR filters starts with an explicit specification of the performance requirements. These should include (i) signal characteristics (types of signal sources and sinks, I/O interface, data rates and wordlengths, and frequencies of interest), (ii) the frequency response characteristics of the filter (the desired amplitude and/or phase responses and their tolerances (if any), the speed of operation), (iii) the manner of implementation (for example, as a high level language routine in a computer or as a DSP processor-based system, choice of signal processor, modes of filtering (real-time or batch)), and (iv) other design constraints (such as costs and permissible signal degradation through the filter). In general, most of the above requirements are application dependent. The designer may not have enough information to specify the filter completely at the outset, but as many of the filter requirements as possible should be specified to simplify the design process.

For frequency selective filters, such as lowpass and bandpass filters, the frequency response specifications are often in the form of a tolerance scheme. Figure 8.2 depicts such a scheme for a bandpass IIR filter. The shaded horizontal lines indicate the tolerance limits. The following parameters are normally used to specify the frequency response.

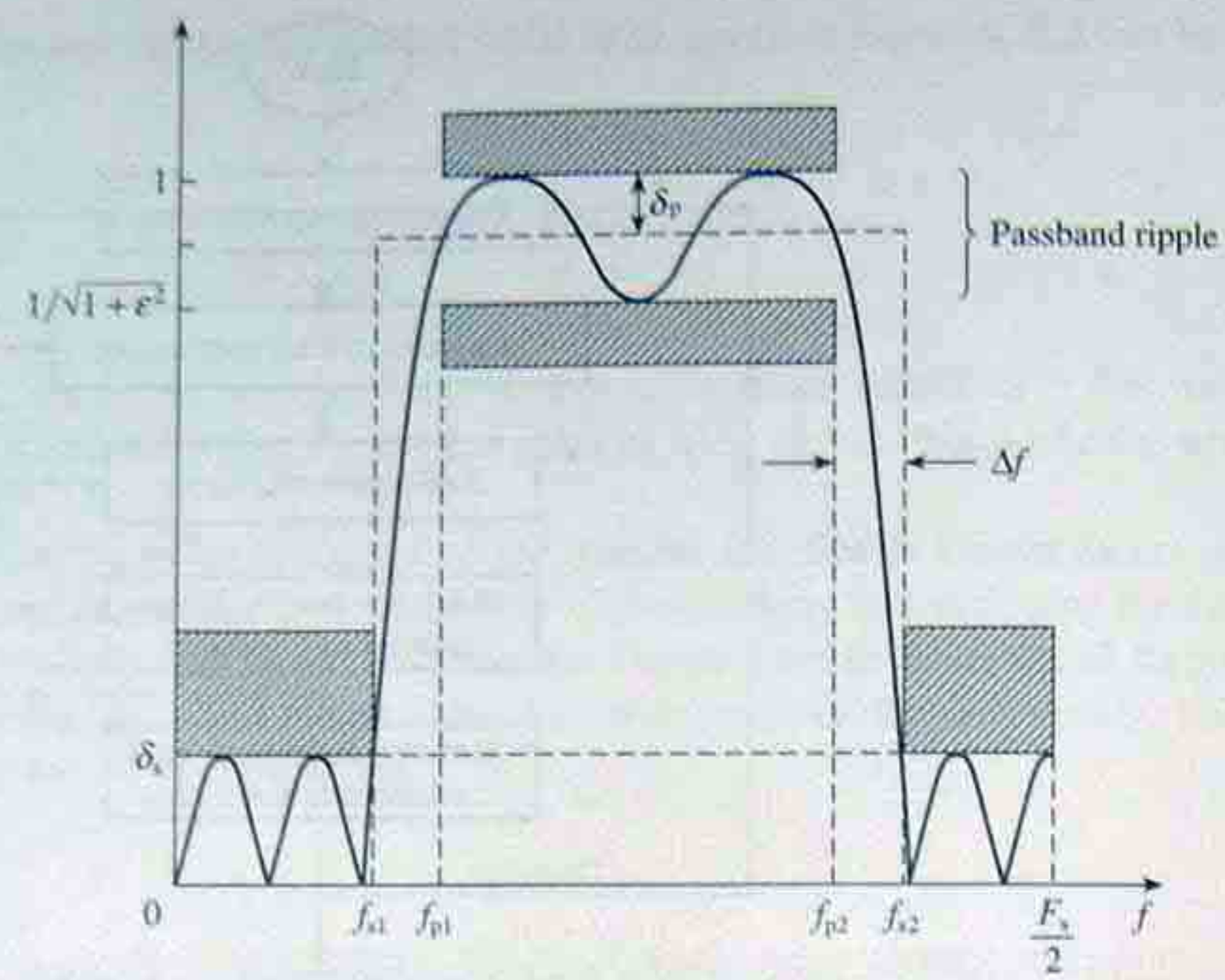


Figure 8.2 Tolerance scheme for an IIR bandpass filter.

ε^2	passband ripple parameter
δ_p	passband deviation
δ_s	stopband deviation
f_{p1} and f_{p2}	passband edge frequencies
f_{s1} and f_{s2}	stopband edge frequencies

The bandedge frequencies are sometimes given in normalized form, that is a fraction of the sampling frequency (f/F_s), but we shall specify them in standard frequency units of hertz or kilohertz as these are less confusing, especially to the inexperienced designer. Passband and stopband deviations may be expressed as ordinary numbers or in decibels: the passband ripple in decibels is

$$A_p = 10 \log_{10} (1 + \varepsilon^2) = -20 \log_{10} (1 - \delta_p) \tag{8.4a}$$

and the stopband attenuation in decibels is

$$A_s = -20 \log_{10} (\delta_s) \tag{8.4b}$$

As discussed in Chapter 6 and is evident in Figure 8.2, for IIR filters the passband ripple is the difference between the minimum and maximum deviation in the passband. For FIR filters, the passband ripple is the difference between the ideal response and the maximum (or minimum) deviation in the passband. Thus, for IIR, when we say passband ripple, what is meant is the peak-to-peak passband ripple.

8.4 Coefficient calculation methods for IIR filters

The task at this stage is to select one of a number of approximation methods and to use it to calculate the values of the coefficients, a_k and b_k, in Equation 8.2, such that the frequency response specifications given in the first design stage are satisfied.

A simple way to obtain the IIR filter coefficients is to place poles and zeros judiciously in the z-plane such that the resulting filter has the desired frequency response. This approach, known as the pole–zero placement method, is only useful for very simple filters, for example notch filtering, where the filter parameters (such as passband ripple) need not be specified precisely. A more efficient approach is first to design an analog filter satisfying the desired specifications and then to convert it into an equivalent digital filter. Most IIR digital filters are designed this way. The rationale behind this approach is that there already exists a wealth of information on analog filters in the literature which can be utilized. Three of the most common methods of converting analog filters into equivalent digital filters are the impulse invariant, the matched z-transform and the bilinear z-transform methods.

In the next few sections we will cover the following methods of calculating the coefficients of IIR filters:

- pole–zero placement;
- impulse invariant;
- matched z-transform;
- bilinear z-transform.

8.5 Pole–zero placement method of coefficient calculation

8.5.1 Basic concepts and illustrative design examples

When a zero is placed at a given point on the z-plane, the frequency response will be zero at the corresponding point. A pole on the other hand produces a peak at the corresponding frequency point; see Figure 8.3. Poles that are close to the unit circle give rise to large peaks, whereas zeros close to or on the circle produce troughs or minima. Thus, by strategically placing poles and zeros on the z-plane, we can obtain simple lowpass, or other frequency selective, filters. Lynn and Fuerst (1989) provide a more detailed discussion of filters of this type.

An important point to bear in mind is that for the coefficients of the filter to be real, the poles and zeros must either be real (that is lie on the positive or negative real axes) or occur in complex conjugate pairs. We will illustrate the method with examples.

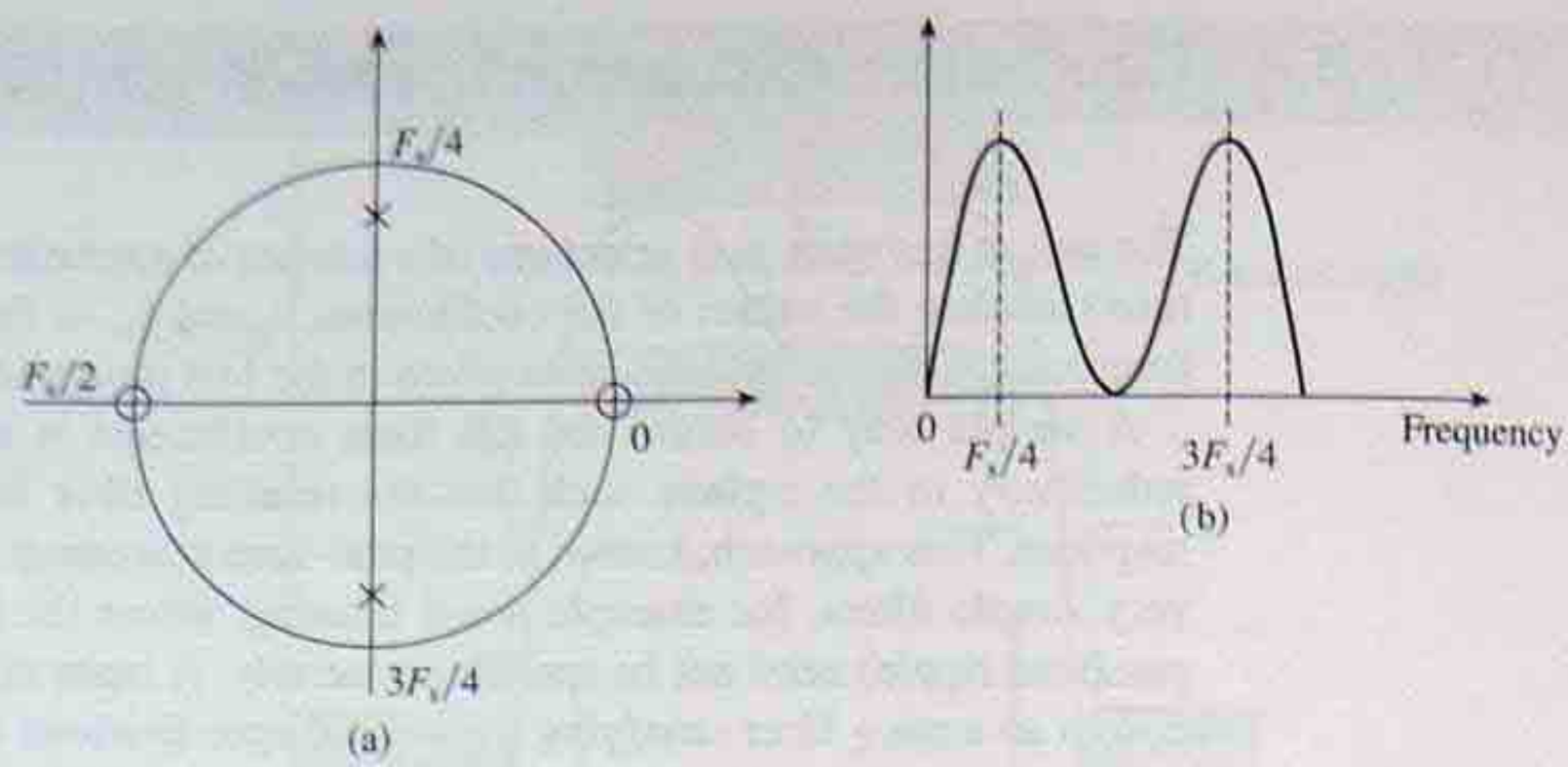


Figure 8.3 (a) Pole–zero diagram of a simple filter, and (b) a sketch of its frequency response.

Example 8.1 *Illustrating the simple pole–zero method of calculating filter coefficients* A bandpass digital filter is required to meet the following specifications:

(1) complete signal rejection at dc and 250 Hz;
(2) a narrow passband centred at 125 Hz;
(3) a 3 dB bandwidth of 10 Hz.

Assuming a sampling frequency of 500 Hz, obtain the transfer function of the filter, by suitably placing z-plane poles and zeros, and its difference equations.

Solution First, we must determine where to place the poles and zeros on the z-plane. Since a complete rejection is required at 0 and 250 Hz, we need to place zeros at corresponding points on the z-plane. These are at angles of 0° and $360° \times 250/500 = 180°$ on the unit circle. To have the passband centred at 125 Hz requires us to place poles at $\pm 360° \times 125/500 = \pm 90°$. To ensure that the coefficients are real, it is necessary to have a complex conjugate pole pair.

The radius, r, of the poles is determined by the desired bandwidth. An approximate relationship between r, for $r > 0.9$, and bandwidth, bw, is given by

$$r \simeq 1 - (\mathrm{bw}/F_s)\pi \tag{8.5}$$

For the problem, bw = 10 Hz and $F_s = 500$ Hz, giving $r = 1 - (10/500)\pi = 0.937$. The pole–zero diagram is given in Figure 8.4(a). From the pole–zero diagram, the transfer function can be written down by inspection:

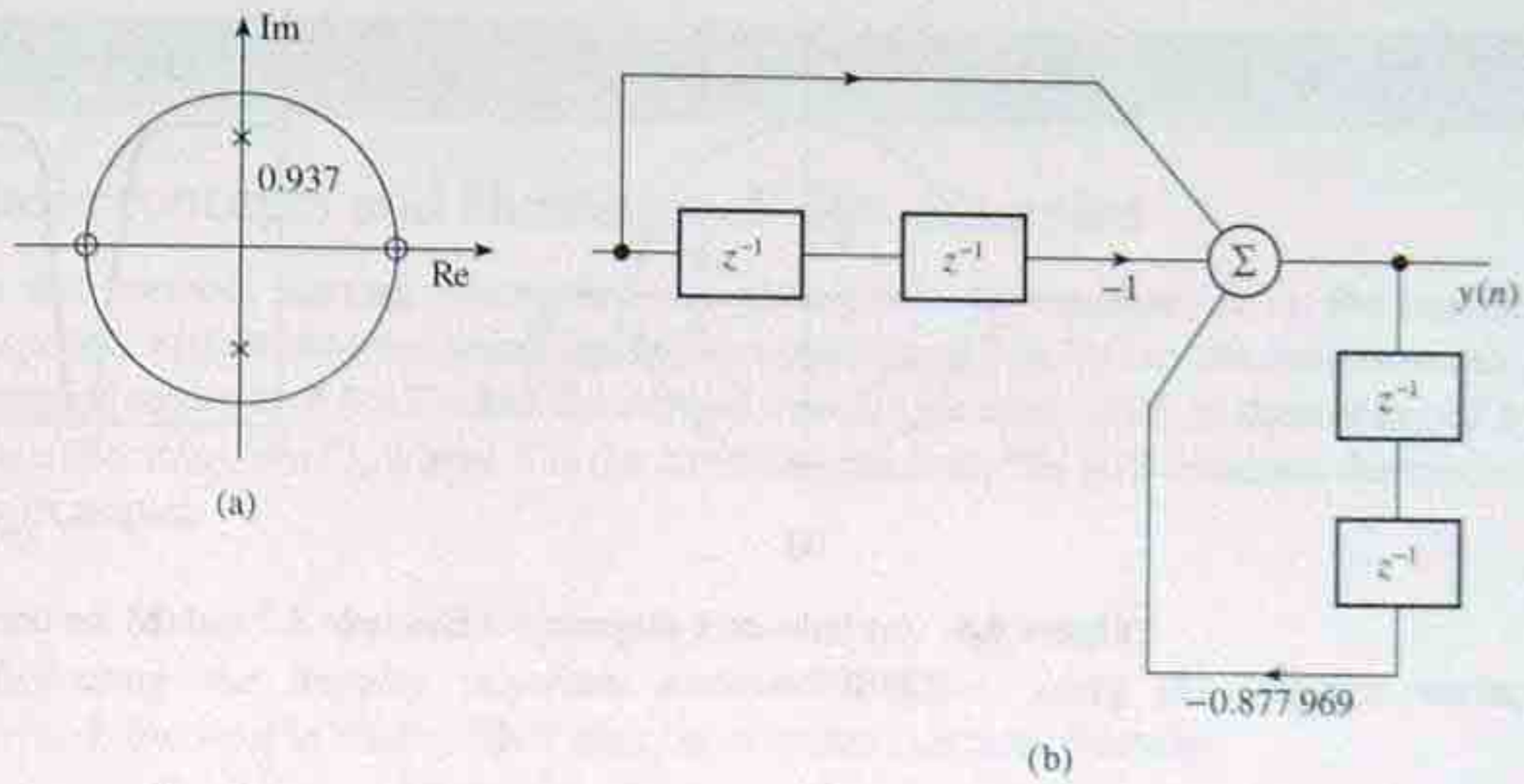


Figure 8.4 (a) Pole–zero diagram. (b) Block diagram representation of filter.

$$H(z) = \frac{(z-1)(z+1)}{(z - re^{j\pi/2})(z - re^{-j\pi/2})}$$

$$= \frac{z^2 - 1}{z^2 + 0.877\,969} = \frac{1 - z^{-2}}{1 + 0.877\,969 z^{-2}}$$

The difference equation is

$$y(n) = -0.877\,969 y(n-2) + x(n) - x(n-2)$$

Comparing the transfer function, $H(z)$, with the general IIR equation (Equation 8.2), we find that the filter is a second-order section with the following coefficients:

$b_0 = 1$ $\quad a_1 = 0$

$b_1 = 0$ $\quad a_2 = 0.877\,969$

$b_2 = -1$

Example 8.2 *Using the pole–zero placement method to calculate coefficients of a notch filter*
Obtain, by the pole–zero placement method, the transfer function and the difference equation of a simple digital notch filter that meets the following specifications:

notch frequency	50 Hz
3 dB width of notch	±5 Hz
sampling frequency	500 Hz

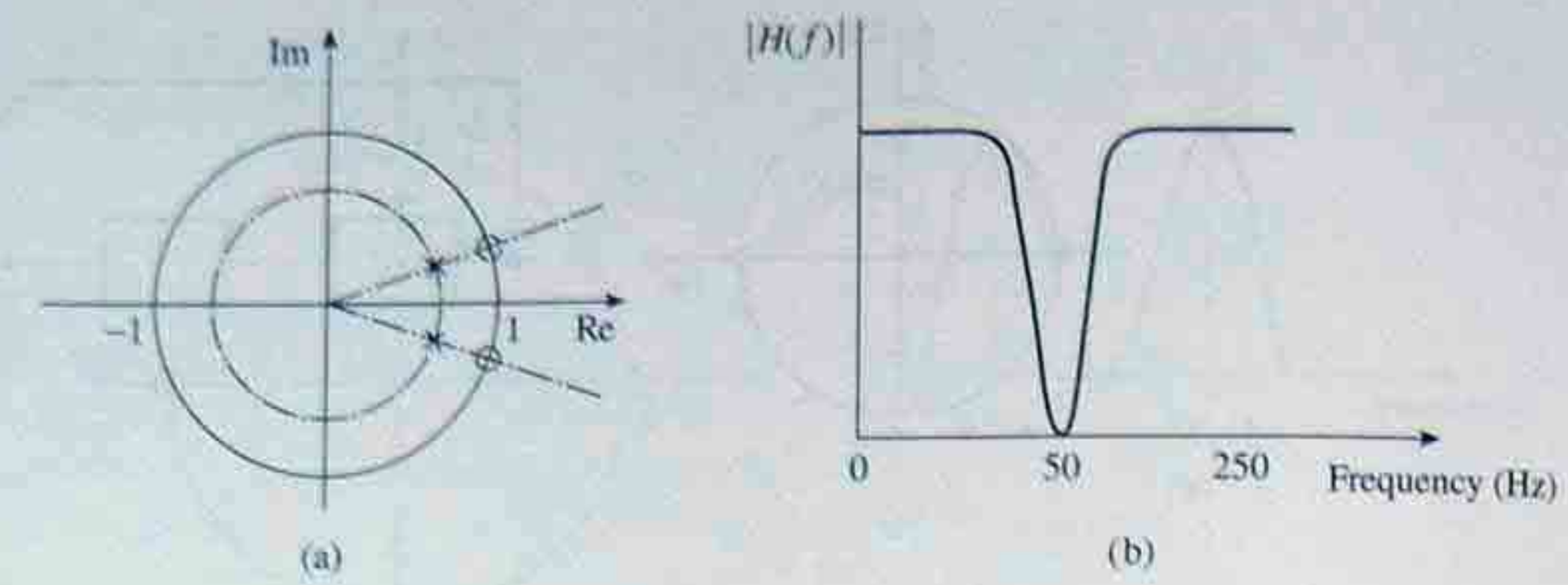


Figure 8.5 (a) Pole–zero diagram for Example 8.2 and (b) the corresponding frequency response.

Solution

- To reject the component at 50 Hz, we place a pair of complex zeros at points on the unit circle corresponding to 50 Hz, that is at angles of $360° \times 50/500 = \pm 36°$.
- To achieve a sharp notch filter and improved amplitude response on either side of the notch frequency, a pair of complex conjugate poles are placed at a radius $r < 1$. The width of the notch is determined by the locations of the poles. The relationship between the bandwidth and the radius of Example 8.1 is applicable. Thus the radius of the poles is 0.9372.
- The pole–zero diagram is given in Figure 8.5(a). From the figure, the transfer function of the filter is given by

$$H(z) = \frac{[z - \exp(-j36°)][z - \exp(j36°)]}{[z - 0.937\exp(-36°)][z - 0.9372\exp(36°)]}$$

$$= \frac{z^2 - 1.6180z + 1}{z^2 - 1.5164z + 0.8783} = \frac{1 - 1.6180z^{-1} + z^{-2}}{1 - 1.5164z^{-1} + 0.8783z^{-2}}$$

The difference equation is

$$y(n) = x(n) - 1.6180x(n-1) + x(n-2) + 1.5164y(n-1) - 0.8783y(n-2)$$

Comparing $H(z)$ with Equation 8.2 shows that the coefficients for the notch filter are

$b_0 = 1$ $\quad a_1 = -1.5164$

$b_1 = -1.6180$ $\quad a_2 = 0.8783$

$b_2 = 1$

8.6 Impulse invariant method of coefficient calculation

8.6.1 Basic concepts and illustrative design examples

In this method, starting with a suitable analog transfer function, $H(s)$, the impulse response, $h(t)$, is obtained using the Laplace transform. The $h(t)$ so obtained is suitably sampled to produce $h(nT)$, and the desired transfer function, $H(z)$, is then obtained by z-transforming $h(nT)$, where T is the sampling interval. We will illustrate the method by examples.

Example 8.3 *Illustrating the impulse invariant method* Digitize, using the impulse variant method, the simple analog filter with the transfer function given by

$$H(s) = \frac{C}{s-p} \tag{8.6}$$

Solution The impulse response, $h(t)$, is given by the inverse Laplace transform:

$$h(t) = L^{-1}[H(s)] = L^{-1}\left(\frac{C}{s-p}\right) = Ce^{pt}$$

where L^{-1} symbolizes the inverse Laplace transform. According to the impulse invariant method, the impulse response of the equivalent digital filter, $h(nT)$, is equal to $h(t)$ at the discrete times $t = nT$, $n = 0, 1, 2, \ldots$, that is

$$h(nT) = h(t)|_{t=nT} = Ce^{pnT}$$

The transfer function of $H(z)$ is obtained by z-transforming $h(nT)$:

$$H(z) = \sum_{n=0}^{\infty} h(nT)z^{-n} = \sum_{n=0}^{\infty} Ce^{pnT}z^{-1}$$

$$= \frac{C}{1 - e^{pT}z^{-1}}$$

Thus, from the result above, we can write

$$\frac{C}{s-p} \rightarrow \frac{C}{1 - e^{pT}z^{-1}} \tag{8.7}$$

To apply the impulse invariant method to a high-order (for example, Mth-order) IIR filter with simple poles, the transfer function, $H(s)$, is first expanded using partial fractions as the sum of single-pole filters:

$$H(s) = \frac{C_1}{s - p_1} + \frac{C_2}{s - p_2} + \ldots + \frac{C_M}{s - p_M}$$
$$= \sum_{K=1}^{M} \frac{C_K}{s - p_K} \tag{8.8}$$

where the p_K are the poles of $H(s)$. Each term on the right-hand side of Equation 8.8 has the same form as Equation 8.6 and so the transformation given in Equation 8.8 is applicable. Thus:

$$\sum_{K=1}^{M} \frac{C_K}{s - p_K} \rightarrow \sum_{K=1}^{M} \frac{C_K}{1 - e^{p_K T} z^{-1}} \tag{8.9}$$

High-order IIR filters are normally realized as cascades or parallel combinations of standard second-order filter sections. Thus the case when $M = 2$ is of particular interest. In this case the transform of Equation 8.9 becomes

$$\frac{C_1}{s - p_1} + \frac{C_2}{s - p_2} \rightarrow \frac{C_1}{1 - e^{p_1 T} z^{-1}} + \frac{C_2}{1 - e^{p_2 T} z^{-1}}$$
$$= \frac{C_1 + C_2 - (C_1 e^{p_2 T} + C_2 e^{p_1 T}) z^{-1}}{1 - (e^{p_1 T} + e^{p_2 T}) z^{-1} + e^{(p_1 + p_2) T} z^{-2}} \tag{8.10}$$

If the poles, p_1 and p_2, are complex conjugates, then C_1 and C_2 will also be complex conjugates and Equation 8.10 reduces to

$$\frac{C_1}{1 - e^{p_1 T} z^{-1}} + \frac{C_1^*}{1 - e^{p_1^* T} z^{-1}} = \frac{2C_r - [C_r \cos(p_i T) + C_i \sin(p_i T)] 2 e^{p_r T} z^{-1}}{1 - 2 e^{p_r T} \cos(p_i T) z^{-1} + e^{2 p_r T} z^{-2}} \tag{8.11}$$

where C_r and C_i are the real and imaginary parts of C_1, p_r and p_i are the real and imaginary parts of p_1, and * symbolizes a complex conjugate.

For most practical impulse invariant IIR filters, the transformations given in Equations 8.7, 8.10 and/or 8.11 are the only transformations required to obtain the coefficients of the transfer function. A C language program for computing the coefficients of impulse invariant filters is given in the appendix. We will illustrate the use of the transformations by an example.

Example 8.4 *Applying the impulse invariant method to filter design* It is required to design a digital filter to approximate the following normalized analog transfer function:

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

Using the impulse invariant method obtain the transfer function, $H(z)$, of the digital filter, assuming a 3 dB cutoff frequency of 150 Hz and a sampling frequency of 1.28 kHz.

Solution Before applying the impulse invariant method, we need to frequency scale the normalized transfer function. This is achieved by replacing s by s/α, where $\alpha = 2\pi \times 150 = 942.4778$, to ensure that the resulting filter has the desired response. Thus

$$H'(s) = H(s)|_{s=s/\alpha} = \frac{\alpha^2}{s^2 + \sqrt{2}\,\alpha s + \alpha^2} = \frac{C_1}{s - p_1} + \frac{C_2}{s - p_2}$$

where

$$p_1 = \frac{-\sqrt{2}\,\alpha(1 - \mathrm{j})}{2} = -666.4324(1 - \mathrm{j}),\ p_2 = p_1^*$$

$$C_1 = -\frac{\alpha}{\sqrt{2}}\mathrm{j} = -666.4324\mathrm{j};\ C_2 = C_1^*$$

Since the poles are complex conjugates, the transformation in Equation 8.11 is used to obtain the discrete-time transfer function, $H(z)$. For the problem, $C_r = 0$, $C_i = -666.4324$, $p_iT = 0.5207$, $p_rT = -0.5207$, $e^{p_rT} = 0.5941$, $\sin(p_iT) = 0.4974$, $\cos(p_iT) = 0.8675$, and $e^{p_rT} = 0.3530$. Substituting these values into Equation 8.11, we obtain $H(z)$:

$$H(z) = \frac{393.9264z^{-1}}{1 - 1.0308z^{-1} + 0.3530z^{-2}}$$

If we substitute $z = e^{j\omega T}$ in the equation above, the value of $H(z)$ at $\omega = 0$ is 1223, approximately equal to the sampling frequency. Such a large gain is characteristic of impulse invariant filters. In general, the gain of the transfer function obtained by this method is equal to the sampling frequency, that is $1/T$, and results from sampling the impulse response. To keep the gain down and to avoid overflows when the filter is implemented, it is common practice to multiply $H(z)$ by T (or equivalently to divide it by the sampling frequency). Thus, for the problem, the transfer function becomes

$$H(z) = \frac{0.3078z^{-1}}{1 - 1.0308z^{-1} + 0.3530z^{-2}}$$

Thus we have

$$b_0 = 0 \qquad a_1 = -1.0308$$
$$b_1 = 0.3078 \qquad a_2 = 0.3530$$

An alternative method of removing the effect of the sampling frequency on the filter gain is to work with normalized frequencies. Thus in the last example we would use $T = 1$ and $\alpha = 2\pi \times 150/1280 = 0.7363$. Using these values in Equation 8.11 leads

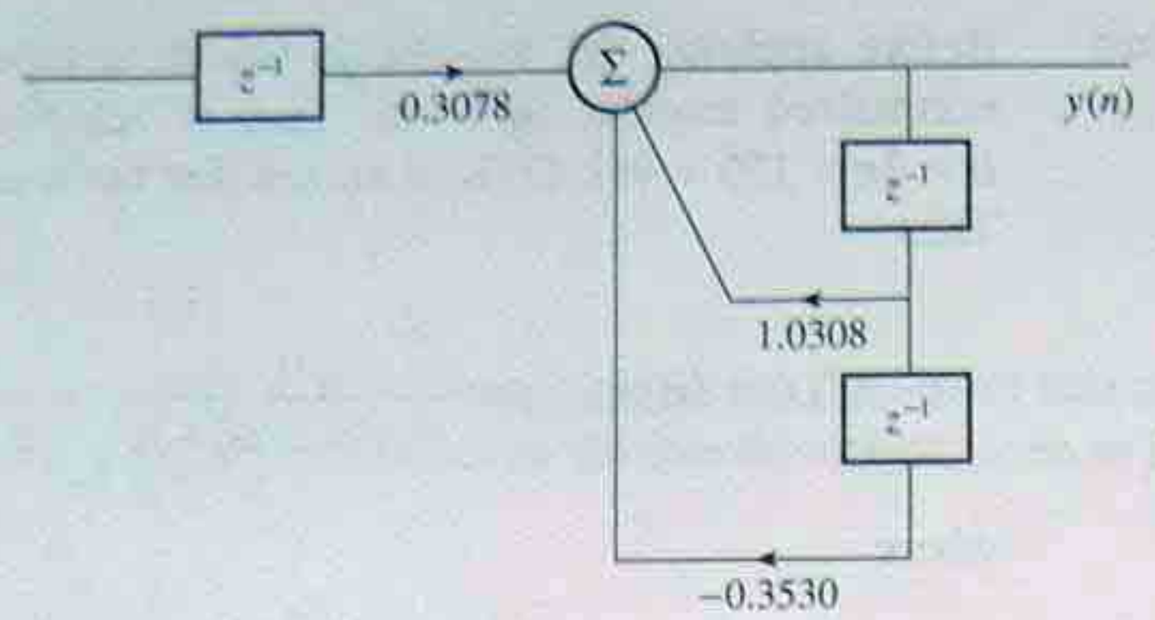


Figure 8.6 A block diagram representation of the filter in Example 8.4.

directly to the desired transfer function above. An important advantage of working with normalized frequencies is that the numbers involved are much simpler. It also means that the results can be generalized. The filter is represented in the form of a block diagram in Figure 8.6.

8.6.2 Summary of the impulse invariant method

(1) Determine a normalized analog filter, $H(s)$, that satisfies the specifications for the desired digital filter.

(2) If necessary, expand $H(s)$ using partial fractions to simplify the next step.

(3) Obtain the z-transform of each partial fraction to obtain Equation 8.9.

(4) Obtain $H(z)$ by combining the z-transforms of the partial fractions into second-order terms and possibly one first-order term. If the actual sampling frequency is used then multiply $H(z)$ by T.

8.6.3 Remarks on the impulse invariant method

(1) The impulse response of the discrete filter, $h(nT)$, is identical to that of the analog filter, $h(t)$, at the discrete time instants $t = nT$, $n = 0, 1, \ldots$; see Figure 8.7 for example. It is for this reason that the method is called the impulse invariant method.

(2) The sampling frequency affects the frequency response of the impulse invariant discrete filter. A sufficiently high sampling frequency is necessary for the frequency response to be close to that of the equivalent analog filter.

(3) As is the case with sampled data systems, the spectrum of the impulse invariant filter corresponding to $H(z)$ would be the same as that of the original analog filter, $H(s)$, but repeats at multiples of the sampling frequency as shown in Figure 8.8, leading to aliasing. However, if the roll-off of the original analog filter is sufficiently steep or if the analog filter is bandlimited before the

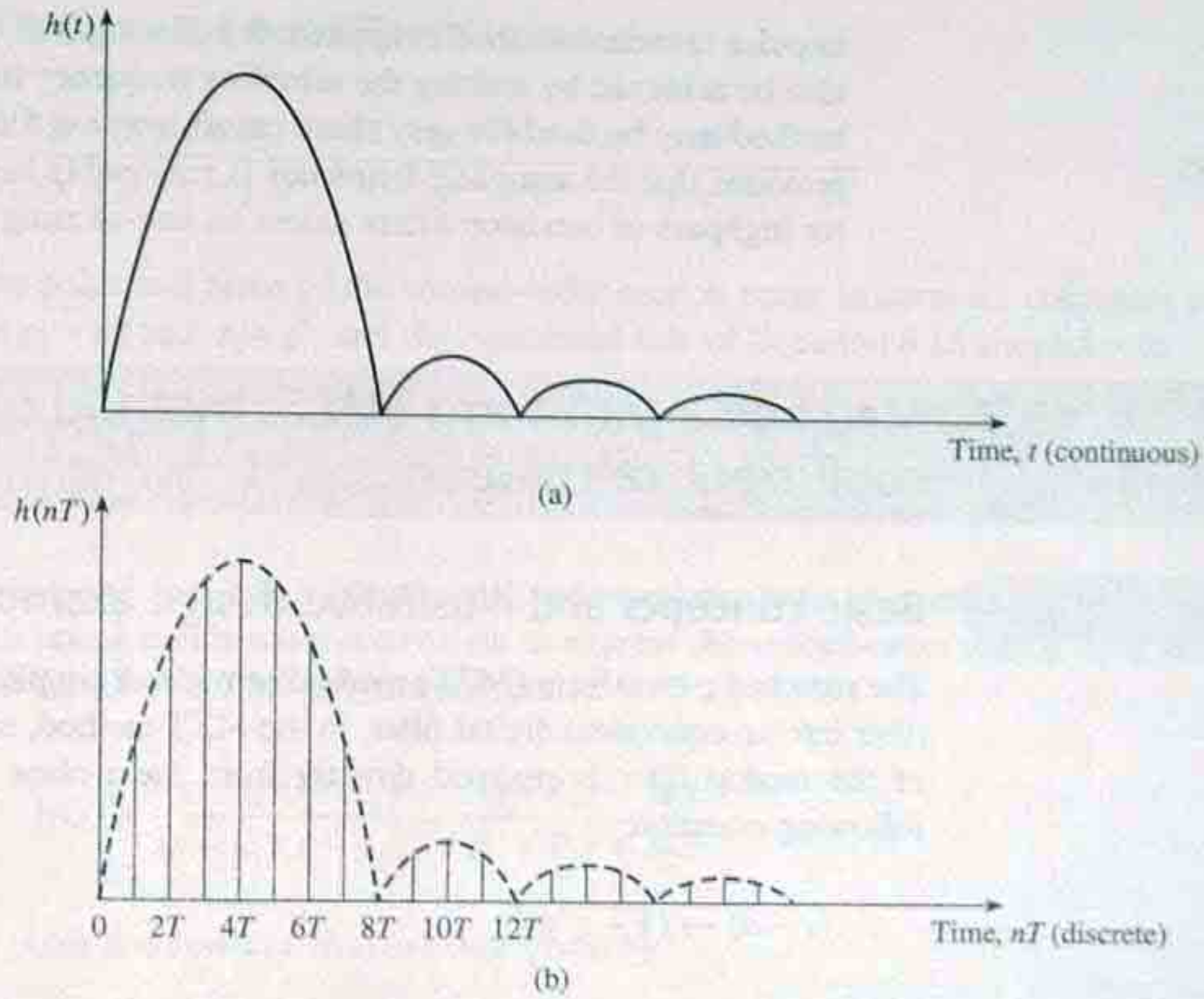


Figure 8.7 Comparison of the impulse response of (a) an analog filter, $h(t)$, and (b) its digital filter equivalent, $h(nT)$. In the impulse invariant method, the two impulse responses are identical at the sampling instants.

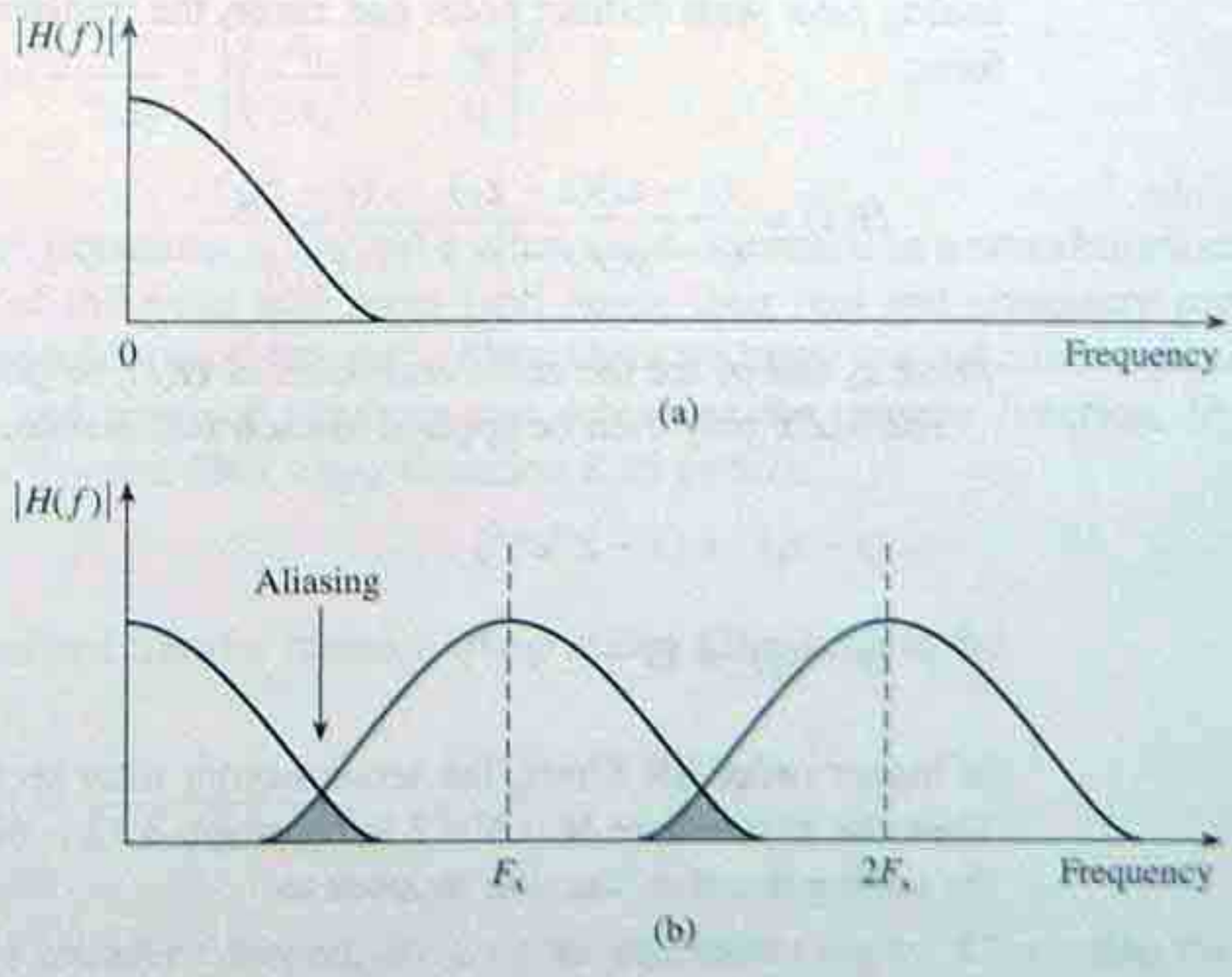


Figure 8.8 (a) Spectrum of an analog filter and (b) spectrum of an equivalent impulse invariant digital filter showing effects of aliasing.

impulse invariant method is applied, the aliasing will be low. Low aliasing can also be achieved by making the sampling frequency high. We conclude that the method may be used for very sharp cutoff lowpass filters with little aliasing, provided that the sampling frequency is reasonably high, but it is unsuitable for highpass or bandstop filters unless an anti-aliasing filter is used.

8.7 Matched z-transform (MZT) method of coefficient calculation

8.7.1 Basic concepts and illustrative design examples

The matched z-transform (MZT) method provides a simple way to convert an analog filter into an equivalent digital filter. In the MZT method, each of the poles and zeros of the analog filter is mapped directly from the s-plane to the z-plane using the following equation:

$$(s - a) \rightarrow (1 - z^{-1}e^{aT}) \tag{8.12}$$

where T is the sampling period. Equation 8.12 maps a pole (or zero) at the location $s = a$ in the s-plane onto a pole (or zero) in the z-plane at $z = e^{aT}$.

For higher order analog filters, the transfer function has several poles and/or zeros and these need to be mapped from the s-plane onto the z-plane. For a higher order analog filter with distinct poles and zeros, the transfer function may be written in the form:

$$H(s) = \frac{(s - z_1)(s - z_2)\ldots(s - z_M)}{(s - p_1)(s - p_2)\ldots(s - p_N)} \tag{8.13}$$

where z_k and p_k are the zeros and poles of $H(s)$, respectively.

The MZT may then be applied to each factor separately:

$$(s - z_k) \rightarrow (1 - z^{-1}e^{z_kT})$$

$$(s - p_k) \rightarrow (1 - z^{-1}e^{p_kT})$$

In higher order IIR filters, the second-order filter section is the basic building block. Thus, the case where $M = N = 2$ in Equation 8.13 is of particular interest. In this case, the analog transfer function reduces to

$$H(s) = \frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)} \tag{8.14}$$

The application of the MZT to this gives

$$\frac{(s-z_1)(s-z_2)}{(s-p_1)(s-p_2)} \rightarrow \frac{1-(e^{z_1T}+e^{z_2T})z^{-1}+e^{(z_1+z_2)T}z^{-2}}{1-(e^{p_1T}+e^{p_2T})z^{-1}+e^{(p_1+p_2)T}z^{-2}} \tag{8.15}$$

If the poles and zeros of the second-order section occur in complex conjugate pairs, then $p_2 = p_1^*$ and $z_2 = z_1^*$, and the right-hand side of Equation 8.15 simplifies to

$$\frac{1-2e^{z_rT}\cos(z_iT)z^{-1}+e^{z_rT}z^{-2}}{1-2e^{p_rT}\cos(p_iT)z^{-1}+e^{p_rT}z^{-2}} \tag{8.16}$$

where z_r and z_i, p_r and p_i, are the real and imaginary parts of z_1 and p_1, respectively.

In practice, it is more convenient to express the second-order analog filter section in the familiar rational polynomial format:

$$H(s) = \frac{(s-z_1)(s-z_2)}{(s-p_1)(s-p_2)} = \frac{A_0+A_1s+A_2s^2}{B_0+B_1s+B_2s^2}$$

The poles and zeros of $H(s)$ are then given by

$$p_{1,2} = -\frac{B_1}{2B_2} \pm \left[\left(\frac{B_1}{2B_2}\right)^2 - \frac{B_0}{B_2}\right]^{\frac{1}{2}} \tag{8.17a}$$

$$z_{1,2} = -\frac{A_1}{2A_2} \pm \left[\left(\frac{A_1}{2A_2}\right)^2 - \frac{A_0}{A_2}\right]^{\frac{1}{2}} \tag{8.17b}$$

In practice, Equations 8.17a and b allow us to determine in a straightforward way the locations of the poles and zeros (and hence their real and imaginary parts), given the transfer function of the analog filter. Once we know the real and imaginary parts of the zeros and poles of $H(s)$, we can determine the transfer function, $H(z)$, of the equivalent discrete filter using Equation 8.15 or 8.16.

Example 8.5 The normalized transfer function of an analog filter is given by

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

Obtain the transfer function, $H(z)$, of an equivalent digital filter using the matched z-transform method. Assume a 3 dB cutoff frequency of 150 Hz and a sampling frequency of 1.28 kHz.

Solution The cutoff frequency may be expressed as $\omega_c = 2\pi \times 150 = 942.4778$ rad/s. The transfer function of the denormalized analog filter is obtained by replacing s by s/ω_c:

$$H'(s) = H(s)\Big|_{s \leftarrow \frac{s}{\omega_c}}$$

$$= \frac{\omega_c^2}{s^2 + \sqrt{2}\omega_c s + \omega_c^2}$$

The poles of the filter are located at

$$p_{1,2} = -\frac{\sqrt{2}\omega_c}{2} \pm \left[\left(\frac{\sqrt{2}\omega_c}{2}\right)^2 - \omega_c^2\right]^{\frac{1}{2}}$$

$$= -\frac{\sqrt{2}\omega_c}{2} \pm \omega_c\left[\left(\frac{\sqrt{2}}{2}\right)^2 - 1\right]^{\frac{1}{2}}$$

$$= -\frac{\sqrt{2}\omega_c}{2}(1 \mp \mathrm{j})$$

For the problem, the real and imaginary parts of the poles are

$$p_r = -\frac{\sqrt{2}\omega_c}{2} = -666.4324, \quad p_i = \frac{\sqrt{2}\omega_c}{2}\mathrm{j} = 666.4324\mathrm{j}$$

Thus, $p_r T = -0.520\,650\,3$, $p_i T = 0.520\,650\,3$, $\cos(p_i T) = 0.867\,496$ and $e^{p_r T} = 0.594\,134$. The resulting transfer function becomes

$$H(z) = \frac{8.8876 \times 10^5}{1 - 1.030\,818z^{-1} + 0.594\,134z^{-2}}$$

8.7.2 Summary of the matched z-transform method

(1) Determine a suitable analog transfer function, $H(s)$, that meets the specifications of the desired digital filter.

(2) Find the locations of the poles and zeros of $H(s)$. This may require factorizing the analog transfer function, $H(s)$.

(3) Map the poles and zeros from the s-plane to the z-plane using Equation 8.12. Equations 8.15 and 8.16 may be used for a second-order section.

(4) Combine the z-plane equations as appropriate to obtain the transfer function, $H(z)$.

8.7.3 Remarks on the matched z-transform method

(1) The MZT method requires a knowledge of the locations of the poles and zeros of the analog filter. This can be obtained by factorizing the analog transfer function, $H(s)$. Thereafter, the MZT is relatively easy to apply.

(2) The MZT and the impulse invariant methods lead to discrete filters with identical denominators. Compare, for example, the denominators of the MZT of Equation 8.15 and the result of the impulse invariant transformation given in Equation 8.10. This is also evident if we compare the denominators of the transfer functions obtained in Examples 8.4 and 8.5.

(3) In digital filters, the useful frequency band extends from zero to the Nyquist frequency (half the sampling frequency) whereas in an analog filter it is from zero to infinity. Thus, the MZT mapping as with other mappings compresses the infinite analog frequency band into a finite frequency band. This causes a distortion in the frequency response of the equivalent digital filters compared to the analog filter. For the MZT, the resulting filters tend to provide less attenuation compared to the analog filters. In Section 8.12, we will discuss this further and show how this feature may be exploited.

(4) If the analog filter has poles at frequencies close to or zeros beyond the Nyquist frequency (that is half the sampling frequency), the frequency response of the resulting MZT-based digital filter will be distorted because of aliasing (see later). In such cases, the response of the analog filter beyond the Nyquist frequency will still be significant. This is folded back into the desired band by the implicit sampling process.

(5) The MZT is also unsuitable for digitizing an all-pole analog filter because of the absence of zeros above the Nyquist frequency. The problem in this case may be alleviated by adding zeros at $z = -1$ (i.e. at the Nyquist frequency).

The effects of the MZT on the frequency response will be discussed further in Section 8.12.

8.8 Bilinear z-transform (BZT) method of coefficient calculation

8.8.1 Basic concepts and illustrative design examples

This is by far the most important method of obtaining IIR filter coefficients. In the BZT method, the basic operation required to convert an analog filter $H(s)$ into an equivalent digital filter is to replace s as follows:

$$s = k\frac{z-1}{z+1}, \quad k = 1 \text{ or } \frac{2}{T} \tag{8.18a}$$

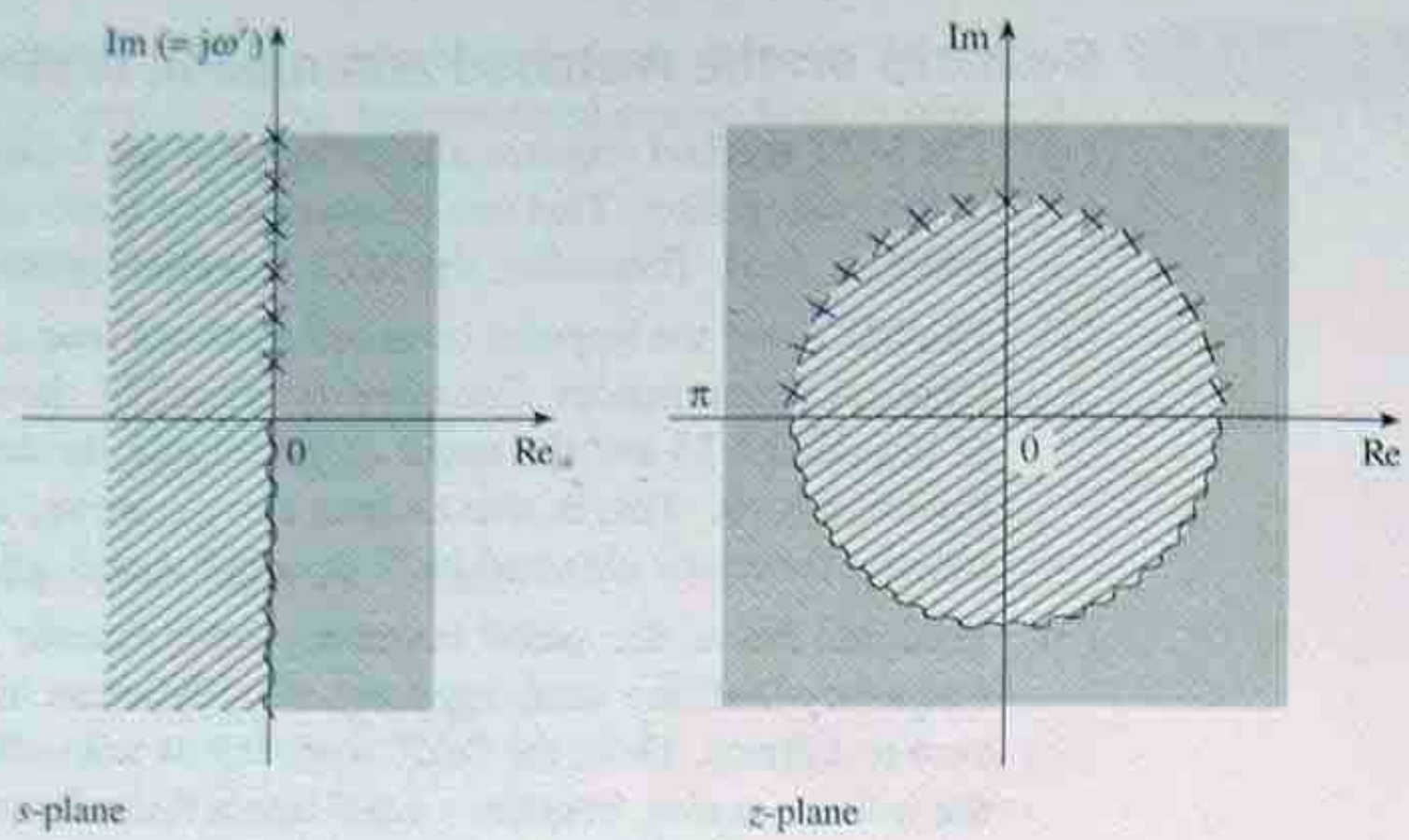


Figure 8.9 An illustration of the s-plane to z-plane mapping using the bilinear z-transformation. Note that the positive $j\omega'$ axis in the s-plane (that is, $s = 0$ to $s = j\infty$) maps to the upper half of the unit circle, and the negative $j\omega'$ axis maps to the lower half.

The above transformation maps the analog transfer function, $H(s)$, from the s-plane into the discrete transfer function, $H(z)$, in the z-plane as shown in Figure 8.9. Notice that in the figure the entire $j\omega$ axis in the s-plane is mapped onto the unit circle, the left-half s-plane is mapped inside the unit circle, and the right-half s-plane is mapped outside the z-plane unit circle. Thus, a stable analog filter, with poles on the left half of the s-plane, will lead to a digital filter with poles inside the unit circle.

Unfortunately, direct replacement of s in $H(s)$ as indicated in Equation 8.18a may lead to a digital filter with an undesirable response. This is readily shown by making the substitution $z = e^{j\omega T}$ and $s = j\omega'$ in Equation 8.18a. Simplifying, we find that the analog frequency ω' and the digital frequency ω are related as

$$\omega' = k \tan\left(\frac{\omega T}{2}\right), \quad k = 1 \text{ or } \frac{2}{T} \tag{8.18b}$$

Equation 8.18b is sketched in Figure 8.10. It is seen that the relationship between the analog frequency ω' and the digital frequency ω is almost linear for small values of ω, but becomes nonlinear for large values of ω, leading to a distortion (or warping) of the digital frequency response. Note, for example, that the passbands for the analog filter on the left-hand side are of constant width and centred at regular intervals, whereas the passbands for the digital equivalent are somewhat squashed up. This effect is normally compensated for by prewarping the analog filter before applying the bilinear transformation.

To compensate for the effect, we prewarp one or more critical frequencies before applying the BZT. For example, for a lowpass filter, we often prewarp the cutoff or bandedge frequency as follows:

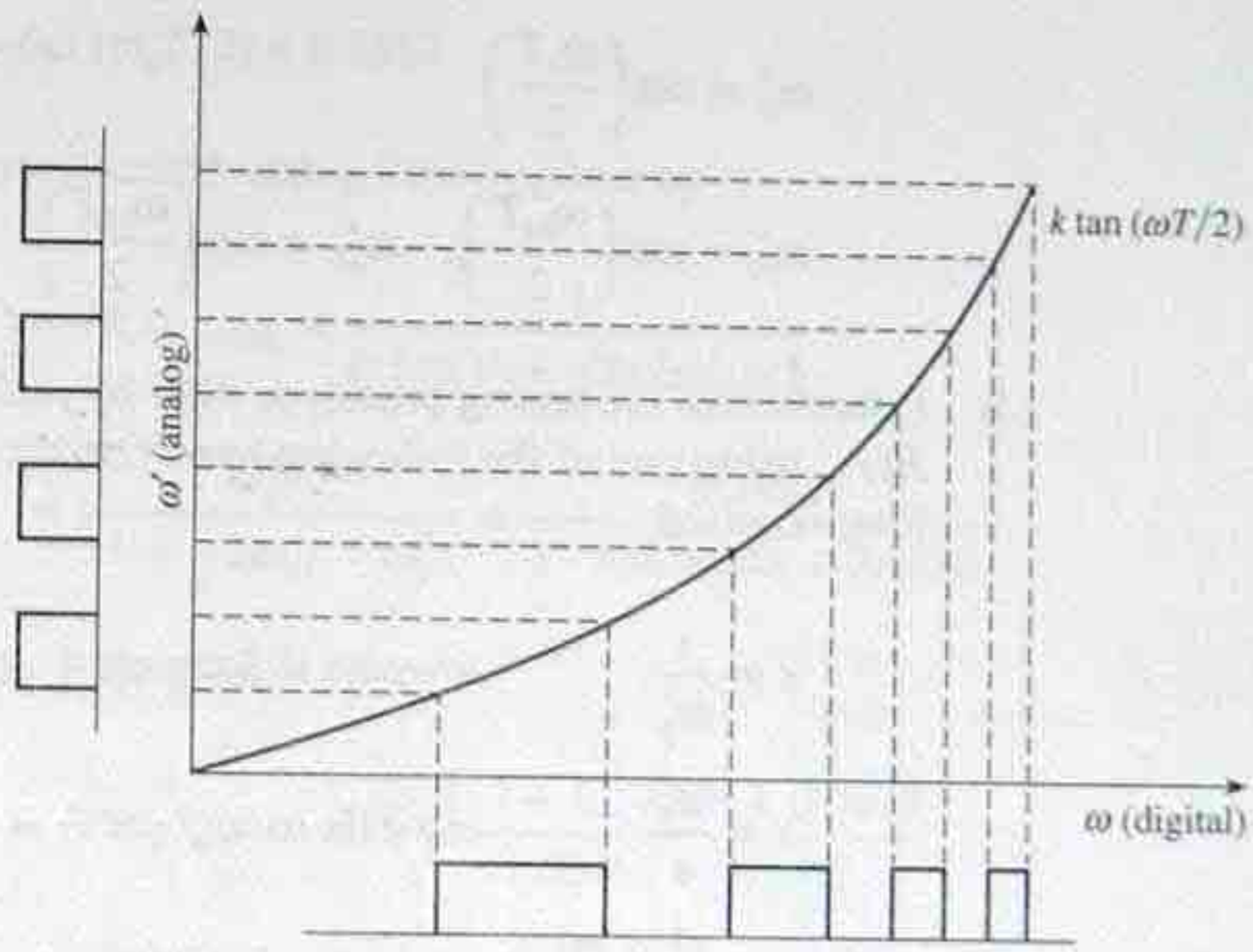


Figure 8.10 Relationship between analog and digital frequencies showing the warping effect. Notice that the equally spaced analog passbands are pushed together at the high frequency and, after transformation, in the digital domain.

$$\omega'_p = k \tan\left(\frac{\omega_p T}{2}\right) \tag{8.19}$$

where

ω_p = specified cutoff frequency
ω'_p = prewarped cutoff frequency
$k = 1$ or $\frac{T}{2}$
T = sampling period

8.8.2 Summary of the BZT method of coefficient calculation

For standard, frequency-selective IIR filters, the steps for using the BZT may be summarized as follows:

(1) Use the digital filter specifications to find a suitable normalized, prototype, analog lowpass filter, $H(s)$.

(2) Determine and prewarp the bandedge or critical frequencies of the desired filter. For lowpass or highpass filters there is just one bandedge or cutoff frequency (ω_p, say). For bandpass and bandstop filters, we have the lower and upper passband edge frequencies, ω_{p1} and ω_{p2}, each of which needs to be prewarped (the stopband edge frequencies may also be specified):

$$\omega'_p = \tan\left(\frac{\omega_p T}{2}\right) \tag{8.20a}$$

$$\omega'_{p1} = \tan\left(\frac{\omega_{p1} T}{2}\right); \quad \omega'_{p2} = \tan\left(\frac{\omega_{p2} T}{2}\right) \tag{8.20b}$$

(3) Denormalize the analog prototype filter by replacing s in the transfer function, $H(s)$, using one of the following transformations, depending on the type of filter required:

$$s = \frac{s}{\omega'_p} \quad \text{lowpass to lowpass} \tag{8.21a}$$

$$s = \frac{\omega'_p}{s} \quad \text{lowpass to highpass} \tag{8.21b}$$

$$s = \frac{s^2 + \omega_0^2}{Ws} \quad \text{lowpass to bandpass} \tag{8.21c}$$

$$s = \frac{Ws}{s^2 + \omega_0^2} \quad \text{lowpass to bandstop} \tag{8.21d}$$

where

$$\omega_0^2 = \omega'_{p2}\omega'_{p1}, \quad W = \omega'_{p2} - \omega'_{p1}$$

(4) Apply the BZT to obtain the desired digital filter transfer function, $H(z)$, by replacing s in the frequency-scaled (i.e. denormalized) transfer function, $H'(s)$, as follows:

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

Example 8.6 *Lowpass filter* It is required to design a digital lowpass filter to approximate the following transfer function:

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

Using the BZT method obtain the transfer function, $H(z)$, of the digital filter, assuming a 3 dB cutoff frequency of 150 Hz and a sampling frequency of 1.28 kHz.

Solution The critical frequency, $\omega_p = 2\pi \times 150$ rad/s, and $F_s = 1/T = 1.28$ kHz, giving a prewarped critical frequency of

$$\omega'_p = \tan(\omega_p T/2) = 0.3857$$

The frequency-scaled analog filter is given by

$$H'(s) = H(s)|_{s=s/\omega'_p} = \frac{1}{(s/\omega'_p)^2 + \sqrt{2}s/\omega'_p + 1}$$

$$= \frac{\omega'^2_p}{s^2 + \sqrt{2}\omega'_p s + \omega'_p s} = \frac{0.1488}{s^2 + 0.5455s + 0.1488}$$

Applying the BZT gives

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}} = \frac{0.0878z^2 + 0.1756z + 0.0878}{z^2 - 1.0048z + 0.3561}$$

$$= \frac{0.0878(1 - 2z^{-1} + z^{-2})}{1 - 1.0048z^{-1} + 0.3561z^{-2}}$$

Example 8.7 *Highpass filter* The normalized transfer function of a simple, analog lowpass, resistance–capacitance (RC) filter is given by

$$H(s) = \frac{1}{s+1}$$

Starting from the s-plane equation, determine, using the BZT method, the transfer function of an equivalent discrete-time highpass filter. Assume a sampling frequency of 150 Hz and a cutoff frequency of 30 Hz.

Solution The critical frequency for the digital filter is $\omega_p = 2\pi \times 30$ rad/s. The cutoff frequency, after prewarping, is $\omega'_p = \tan(\omega_p T/2)$. With $T = 1/150$ Hz, $\omega'_p = \tan(\pi/5) = 0.7265$.

Using the lowpass-to-highpass transformation of Equation 8.21a, the denormalized analog transfer function is obtained as

$$H'(s) = H(s)|_{s=\omega'_p/s} = \frac{1}{\omega'_p/s + 1} = \frac{s}{s + 0.7265}$$

The z-plane transfer function is obtained by applying the BZT:

$$H(z) = H'(s)|_{s=(z-1)/(z+1)} = \frac{(z-1)/(z+1)}{(z-1)/(z+1) + 0.7265}$$

Simplifying, we have

$$H(z) = 0.5792\frac{1 - z^{-1}}{1 + 0.1584z^{-1}}$$

The coefficients of the discrete-time filter are

$$b_0 = 0.5792, \quad a_1 = 0.1584$$
$$b_1 = -0.5792$$

Example 8.8 *Bandpass filter* A discrete-time bandpass filter with Butterworth characteristics meeting the specifications given below is required. Obtain the coefficients of the filter using the BZT method:

passband	200–300 Hz
sampling frequency	2 kHz
filter order, N	2

Solution A first order, normalized analog lowpass filter is required (since the frequency band transformation for bandpass filters – Equation 8.21c – will double the filter order). Thus

$$H(s) = \frac{1}{s + 1}$$

The prewarped critical frequencies are

$$\omega'_{p1} = \tan\left(\frac{\omega_{p1}T}{2}\right) = \tan\left(\frac{2\pi \times 200}{2 \times 2000}\right) = 0.3249$$

$$\omega'_{p2} = \tan\left(\frac{\omega_{p2}T}{2}\right) = \tan\left(\frac{2\pi \times 300}{2 \times 2000}\right) = 0.5095$$

$$\omega_0^2 = \omega'_{p1}\omega'_{p2} = 0.1655$$
$$W = \omega'_{p2} - \omega'_{p1} = 0.1846$$

Using the lowpass-to-bandpass transformation, Equation 8.21c, we have

$$H'(s) = H(s)\Big|_{s=\frac{s^2+\omega_0^2}{Ws}} = \frac{1}{\dfrac{s^2 + \omega_0^2}{Ws} + 1}$$

$$= \frac{Ws}{s^2 + Ws + \omega_0^2}$$

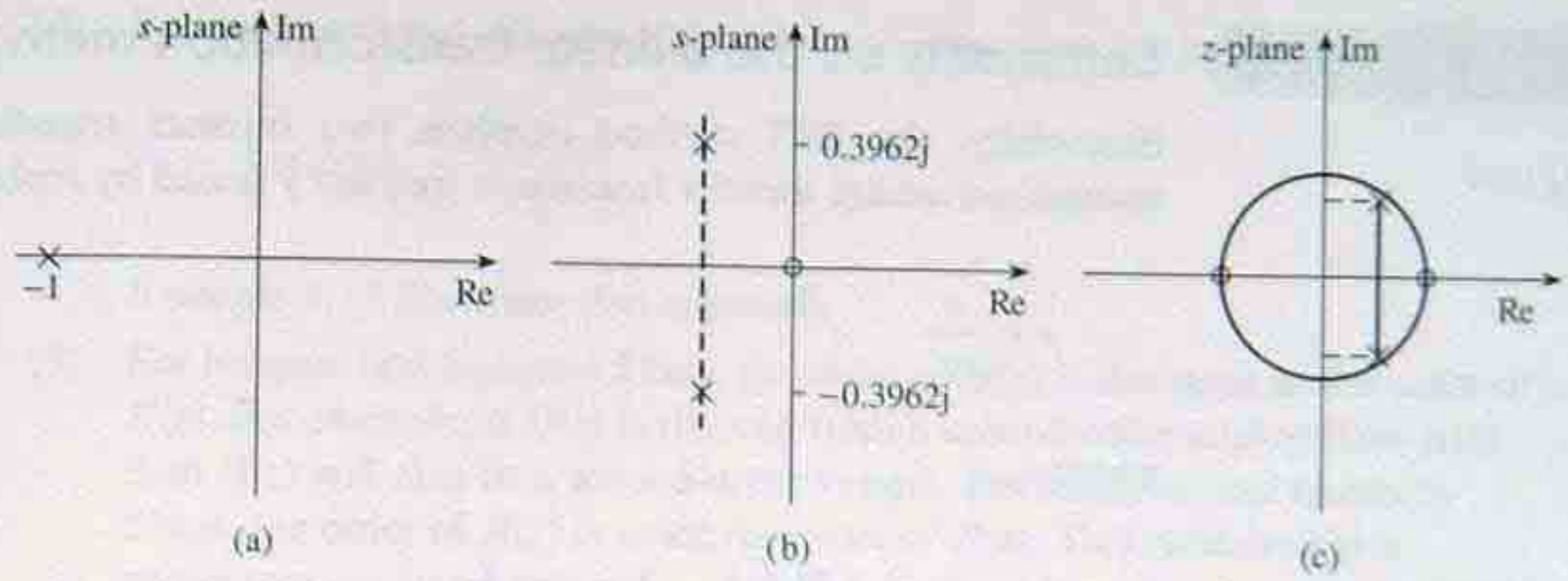


Figure 8.11 Pole–zero diagrams for (a) a prototype lowpass filter and those of (b) intermediate analog bandpass and (c) discrete bandpass filters obtained by band transformation.

Applying the BZT gives

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}} = \frac{W\left(\dfrac{z-1}{z+1}\right)}{\left(\dfrac{z-1}{z+1}\right)^2 + W\left(\dfrac{z-1}{z+1}\right) + \omega_0^2}$$

Substituting the values of ω_0^2 and W, and simplifying, we have

$$H(z) = 0.1367\frac{1 - z^{-2}}{1 - 1.2362z^{-1} + 0.7265z^{-2}}$$

The pole–zero diagrams of the normalized prototype lowpass filter (LPF), the analog bandpass filter and the discrete-time bandpass filter are depicted in Figure 8.11. Note that the lowpass-to-bandpass transformation has introduced a single zero at the origin of the s-plane and at infinity. The BZT then maps the zeros to $z = \pm 1$. Thus, the zeros of the discrete bandpass filter are at $z = 1$ and $z = -1$. Its poles are at $z = 0.6040 \pm 0.6015j$. The analog bandpass zeros are at $s = 0$ and infinity (not shown) and the poles at $s = -0.0923 \pm 0.3962j$.

In practice, higher order IIR filters (for example, $N > 3$) are normally realized as cascades or parallel combinations of second- and/or first-order filter sections to reduce the effects of finite wordlength on filter performance (see later). Thus, after converting an analog filter into discrete form the resulting z-transfer function $H(z)$, if it is of a high order, will need to be expressed in factored form (for cascade realization) or as the sum of second- and/or first-order terms (for parallel realization). To simplify this task we can expressed $H(s)$, at the outset, in factored form and then transform each factor separately. The resulting factors can then be combined or rearranged into a suitable format for the desired realization. This is basically the approach used in Example 8.15.

8.8.3 Comments on the bilinear transformation method

Essentially, the BZT method involves two separate transformations. First, the normalized analog transfer function is frequency scaled by replacing s as follows:

$$s = \frac{s}{\omega_p'} \tag{8.22a}$$

where

$$\omega_p' = k \tan\left(\frac{\omega_p T}{2}\right), \quad k = 1 \text{ or } \frac{2}{T}$$

Second, the BZT is applied by replacing s in the new transfer function as

$$s = k\frac{z-1}{z+1} \tag{8.22b}$$

(1) It is common practice in many texts (for example, Rabiner and Gold, 1975) to use the factor $k = 2/T$ in the two operations above. It should be mentioned that both $k = 1$ and $k = 2/T$ lead to the same results because k is cancelled out anyway. To illustrate, consider the following simple filter:

$$H(s) = \frac{1}{s+1}$$

Assuming that the digital filter is to have a cutoff frequency of ω_p, then we must frequency scale $H(s)$ with the following frequency:

$$\omega_p' = k \tan\left(\frac{\omega_p T}{2}\right)$$

Thus the transfer function is

$$H'(s) = H(s)\big|_{s=s/\omega_p'} = \frac{1}{s/k \tan(\omega_p T/2) + 1}$$

Next, we replace s by $k(z-1)/(z+1)$:

$$H(z) = H'(s)\big|_{s=k(z-1)/(z+1)} = \frac{1}{[\cancel{k}(z-1)/(z+1)]/\cancel{k} \tan(\omega_p T/2) + 1}$$

From the above we see that the factor k is cancelled out, and it would not have mattered whether k was 1 or $2/T$.

(2) For computational efficiency, the two transformations can be combined into one:

$$s = \cot\left(\frac{\omega_p T}{2}\right)\frac{z-1}{z+1} \tag{8.23}$$

Example 8.15 illustrates this approach.

(3) For lowpass and highpass filters, the order of $H(z)$ is the same as the order of $H(s)$. For example, if $H(z)$ is derived from a second-order analog filter $H(s)$ then $H(z)$ will also be a second-order system. For bandpass and bandstop filters, the order of $H(z)$ is twice the order of $H(s)$. This relationship is sometimes exploited to cut down on the algebraic manipulations in the BZT method (see for example Stanley *et al.*, 1984).

(4) In practice, we sometimes encounter a situation where it is required to convert the s-plane transfer function, $H(s)$, of an existing analog filter, into an equivalent bandpass discrete-time filter. Such is the case, for example, in digital audio where an analog filter which has been successfully used for equalization may need to be converted into an equivalent digital filter (Clark *et al.*, 2000).

In such cases, the analog transfer function of the actual filter is already available and so the BZT can be applied directly, after prewarping and frequency scaling using a direct lowpass-to-lowpass frequency scaling. The following examples illustrate the design issues involved.

Example 8.9 *An illustration of the BZT method when the analog transfer function of the actual filter is available* Determine, using the BZT technique, the coefficients of a discrete-time bell filter for audio signal processing in a digital mixer when the control settings correspond to a Q-factor of 2, a boost (peak) of 6.02 dB at 5 kHz. Assume a sampling frequency of 48 kHz, and that the equivalent analog filter has an s-plane transfer function of the form

$$H(s) = \frac{s^2 + (3+k)\dfrac{\omega_0}{Q}s + \omega_0^2}{s^2 + (3-k)\dfrac{\omega_0}{Q}s + \omega_0^2}$$

where

$$k = 3\left(\frac{G-1}{G+1}\right)$$

ω_0 = boost frequency

G = gain factor

Q = Q-factor

Solution The gain or boost of 6.02 dB corresponds to

$$G = 10^{\frac{6.02}{20}} = 1.9999; \quad k = 0.9999$$

The s-plane transfer function becomes

$$H(s) = \frac{s^2 + 4\dfrac{\omega_0}{Q}s + \omega_0^2}{s^2 + 2\dfrac{\omega_0}{Q}s + \omega_0^2}$$

The analog frequency response may be obtained by making the substitution $s = j\omega$ in this equation.

Now the prewarped boost frequency is

$$\omega_0' = \tan\left(\frac{\omega_0 T}{2}\right) = 0.339\,454$$

and the frequency scaled factor (Clark *et al.*, 2000) is given by

$$p = \frac{\omega_0}{\tan\left(\omega_0 T/2\right)} = \frac{\omega_0}{0.339\,454}$$

Thus, the prewarped analog transfer function is

$$H'(s) = H(s)\big|_{s=ps} = \frac{p^2 s^2 + 4\dfrac{\omega_0}{Q}ps + \omega_0^2}{p^2 s^2 + 2\dfrac{\omega_0}{Q}ps + \omega_0^2}$$

Applying the BZT,

$$\begin{aligned} H(z) &= H'(s)\Big|_{s=\frac{z-1}{z+1}} \\ &= \frac{\left(\dfrac{z-1}{z+1}\right)^2 + 0.678\,908\,5\left(\dfrac{z-1}{z+1}\right) + 0.115\,229}{\left(\dfrac{z-1}{z+1}\right)^2 + 0.339\,454\left(\dfrac{z-1}{z+1}\right) + 0.115\,229} \\ &= \frac{1.233\,352 - 1.216\,444z^{-1} + 0.299\,94z^{-2}}{1 - 1.216\,444z^{-1} + 0.533\,294\,6z^{-2}} \end{aligned}$$

Example 8.10 A simple LRC notch filter has the following normalized, s-plane transfer function:

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

Determine the transfer function of an equivalent discrete-time filter using the BZT method. Assume a notch frequency of 50 Hz and a sampling frequency of 500 Hz.

Solution Since we already have the s-plane transfer function, it is inappropriate to apply a lowpass to bandstop transformation as this would amount to a double transformation. The critical frequency is

$$\omega'_{\mathrm{p}} = \tan\left(\frac{\omega_{\mathrm{p}} T}{2}\right) = \tan\left(\frac{2\pi \times 50}{500 \times 2}\right) = 0.324\,919\,6$$

The frequency-scaled s-plane transfer function is

$$\begin{aligned} H'(s) &= H(s)\Big|_{s \to \frac{s}{\omega'_{\mathrm{p}}}} \\ &= \frac{\left(\dfrac{s}{\omega'_{\mathrm{p}}}\right)^2 + 1}{\left(\dfrac{s}{\omega'_{\mathrm{p}}}\right)^2 + \dfrac{s}{\omega'_{\mathrm{p}}} + 1} \\ &= \frac{s^2 + 0.105\,572}{s^2 + 0.324\,919\,6s + 0.105\,572} \end{aligned}$$

Apply the BZT:

$$\begin{aligned} H(z) &= H'(s)\Big|_{s \to \frac{z-1}{z+1}} \\ &= \frac{\left(\dfrac{z-1}{z+1}\right)^2 + 0.105\,572}{\left(\dfrac{z-1}{z+1}\right)^2 + 0.324\,919\,6\left(\dfrac{z-1}{z+1}\right) + 0.105\,572} \end{aligned}$$

8.9 Use of BZT and classical analog filters to design IIR filters

In many practical cases, the analog transfer function, $H(s)$, from which $H(z)$ is obtained, may not be available and will have to be determined from the specifications of the desired digital filters. For standard frequency selective digital filtering tasks (i.e. involving lowpass, highpass, bandpass and bandstop filters), $H(s)$ can be derived from the classical filters with Butterworth, Chebyshev or elliptic characteristics (see Figure 8.12). In this section, we will present methods of designing IIR filters from such classical analog filters. We will consider the basic concepts in detail and illustrate the design issues with worked examples. First, we will review briefly the relevant features of three classical analog filters which are needed to design IIR filters. Only lowpass prototype filters are considered, since other filter types are normally derived from normalized lowpass filters as will be shown later.

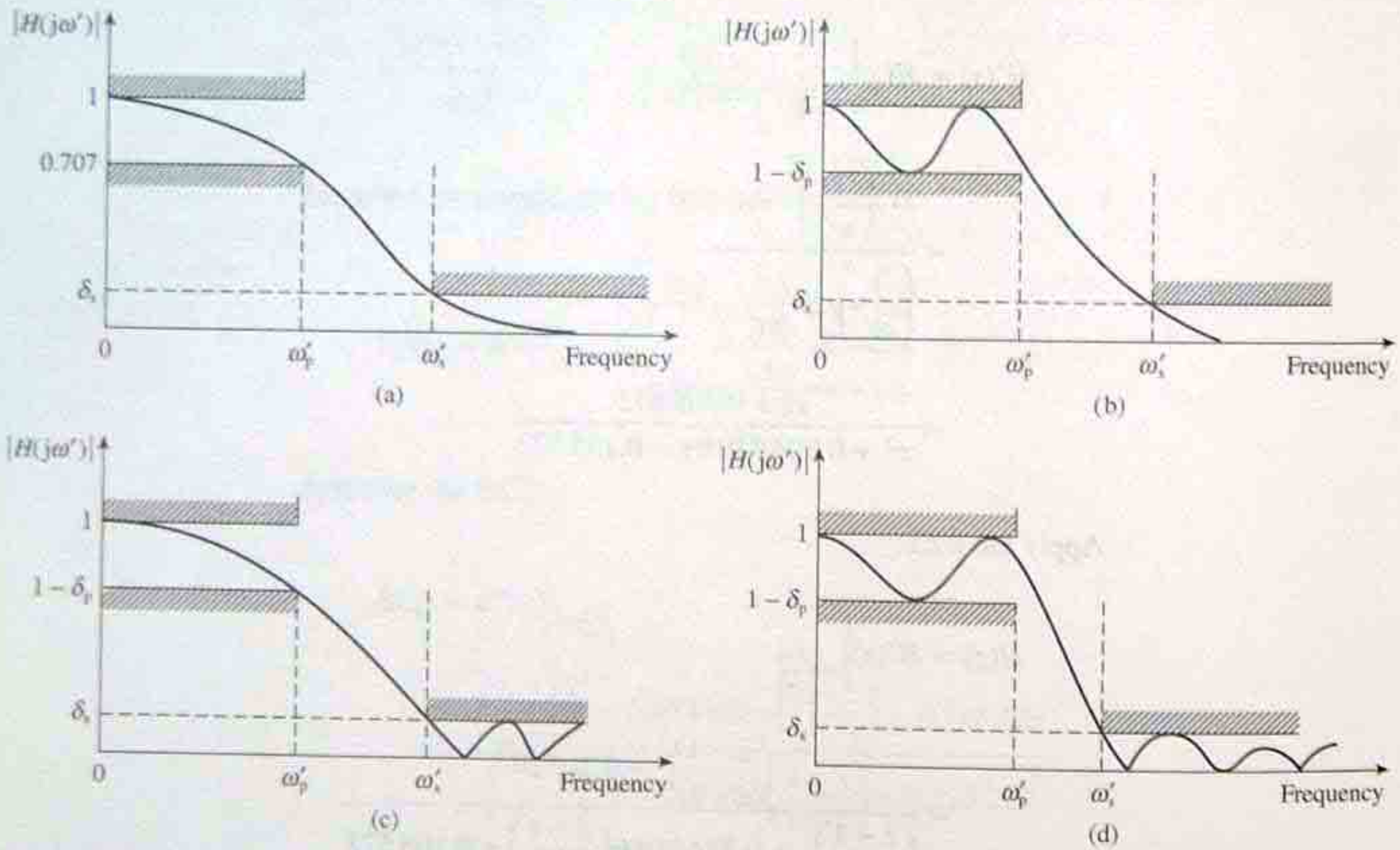


Figure 8.12 Sketches of frequency responses of some classical analog filters: (a) Butterworth response; (b) Chebyshev type I; (c) Chebyshev type II; (d) elliptic.

8.9.1 Characteristic features of classical analog filters

8.9.1.1 Butterworth filter

The lowpass Butterworth filter is characterized by the following magnitude-squared–frequency response:

$$|H(\omega)|^2 = \frac{1}{1 + \left(\dfrac{\omega}{\omega_p^p}\right)^{2N}} \tag{8.24}$$

where N is the filter order and ω_p^p is the 3 dB cutoff frequency of the lowpass filter (for the normalized prototype filter, $\omega_p^p = 1$). The magnitude–frequency response of a typical Butterworth lowpass filter is depicted in Figure 8.12(a), and is seen to be monotonic in both the passband and stopband. The response is said to be maximally flat because of its initial flatness (with a slope of zero at dc).

The filter order, N, is given by

$$N \geqslant \frac{\log\left(\dfrac{10^{\frac{A_s}{10}} - 1}{10^{\frac{A_p}{10}} - 1}\right)}{2\log\left(\dfrac{\omega_s^p}{\omega_p^p}\right)} \tag{8.25}$$

where A_p and A_s are, respectively, the passband ripple and stopband attenuation in dB, and ω_s^p is the stopband edge frequency above.

The transfer function of the normalized analog Butterworth filter, $H(s)$, contains zeros at infinity and poles which are uniformly spaced on a circle of radius 1 in the s-plane at the following locations (Stearns and Hush, 1990; Jong, 1982):

$$s_k = e^{j\pi(2k+N-1)/2N} = \cos\left[\frac{(2k+N-1)\pi}{2N}\right] + j\sin\left[\frac{(2k+N-1)\pi}{2N}\right], \quad k = 1, 2, \ldots, N \tag{8.26}$$

The poles occur in complex conjugate pairs and lie on the left-hand side of the s-plane.

8.9.1.2 Chebyshev filter

The Chebyshev characteristic provides an alternative way of obtaining a suitable analog transfer function, $H(s)$. There are two types of Chebyshev filters, types I and II, with the following features (Figures 8.12(b) and 8.12(c)):

- Type I, with equal ripple in the passband, monotonic in the stopband;
- Type II, with equal ripple in the stopband, monotonic in the passband.

Type I Chebyshev filters, for example, are characterized by the magnitude-squared response

$$|H(\omega')|^2 = \frac{K}{1 + \varepsilon^2 C_N^2(\omega'/\omega'_p)} \tag{8.27a}$$

where $C_N(\omega'/\omega'_p)$ is a Chebyshev polynomial which exhibits equal ripple in the passband, N is the order of the polynomial as well as that of the filter, and ε determines the passband ripple, which in decibels is given by

$$\text{passband ripple} \leqslant 10 \log_{10}(1 + \varepsilon^2) = -20 \log_{10}(1 - \delta_p) \tag{8.27b}$$

A typical amplitude response of a type I Chebyshev characteristic is shown in Figure 8.12(b). The transfer function, $H(s)$, for the Chebyshev response depends on the desired passband ripple and the filter order, N. The filter order N is given by

$$N \geqslant \frac{\cosh^{-1}\left(\dfrac{10^{\frac{A_s}{10}} - 1}{10^{\frac{A_p}{10}} - 1}\right)}{\cosh^{-1}\left(\dfrac{\omega_s^p}{\omega_p^p}\right)} \tag{8.28}$$

where A_p and A_s are, respectively, the passband ripple and stopband attenuation in dB, and ω_s^p is the stopband edge frequency.

The poles of the normalized Chebyshev LPF lie on an ellipse in the s-plane and have coordinates given by (Stearns and Hush, 1990)

$$s_k = \sinh(\alpha)\cos(\beta_k) + \mathrm{j}\cosh(\alpha)\sin(\beta_k) \tag{8.29}$$

where

$$\alpha = \frac{1}{N}\sinh^{-1}\left(\frac{1}{\varepsilon}\right); \beta_k = \frac{(2k + N - 1)\pi}{2N}, \quad k = 1, 2, \ldots, N$$

8.9.1.3 Elliptic filter

The elliptic filter exhibits equiripple behaviour in both the passband and the stopband; see Figure 8.12(d). It is characterized by the following magnitude-squared response:

$$|H(\omega')|^2 = \frac{K}{1 + \varepsilon^2 G_N^2(\omega')} \tag{8.30}$$

where $G_N(\omega')$ is a Chebyshev rational function. Unlike the Butterworth and Chebyshev filters, there is no simple expression for the poles of the elliptic filter. A

procedure is available for computing locations of the poles (for example, in Antoniou, 1979; Jong, 1982; DeFatta *et al.*, 1988). The zeros of the elliptic lowpass filter are entirely imaginary.

The elliptic characteristic provides the most efficient filters in terms of amplitude response. It yields the smallest filter order for a given set of specifications and should be the method of first choice in IIR filter design, except where the phase response is of concern when the Butterworth response may be preferred.

Tables of the polynomials of $H(s)$ for the Butterworth, Chebyshev and elliptic characteristics are available in most analog design books in normalized form and can be used in the bilinear transformation. In practice, however, the computation of $H(z)$ from $H(s)$ is via a software package as we shall see later.

8.9.2 The BZT methodology using classical analog filters

In cases where the prototype lowpass filter does not exist, the stages of the BZT method are:

(1) Prewarp the bandedge or critical frequencies of the digital filter as described previously.

(2) Find a suitable lowpass prototype analog filter, based on the digital filter specifications and one of the classical filter characteristics. This involves using one of the frequency transformation equations (depending on the type of digital filter – LP, HP, BP or BS) in reverse to determine the specifications of the prototype LP filter. From this the order of the prototype filter and hence its transfer function, $H(s)$, are found.

(3) Denormalize the prototype analog LP filter, $H(s)$, by frequency transformation and scaling to obtain a new transfer function, $H'(s)$, as described previously.

(4) Apply the BZT to obtain the desired digital filter transfer function, $H(z)$, by replacing s in the frequency scaled transfer function, $H'(s)$, as described previously.

We will now look at the basic concepts for each of the filter types (LP, HP, BP and BS).

8.9.2.1 Lowpass filter – basic concepts

The lowpass-to-lowpass transformation is given by (see Equation 8.21a):

$$s = \frac{s}{\omega_p'}$$

If we replace s by $j\omega$ in the equation and denote frequencies for the prototype filter by ω^p and those for the lowpass filter to be designed by ω_{lp}, to distinguish between them, then the above equation becomes

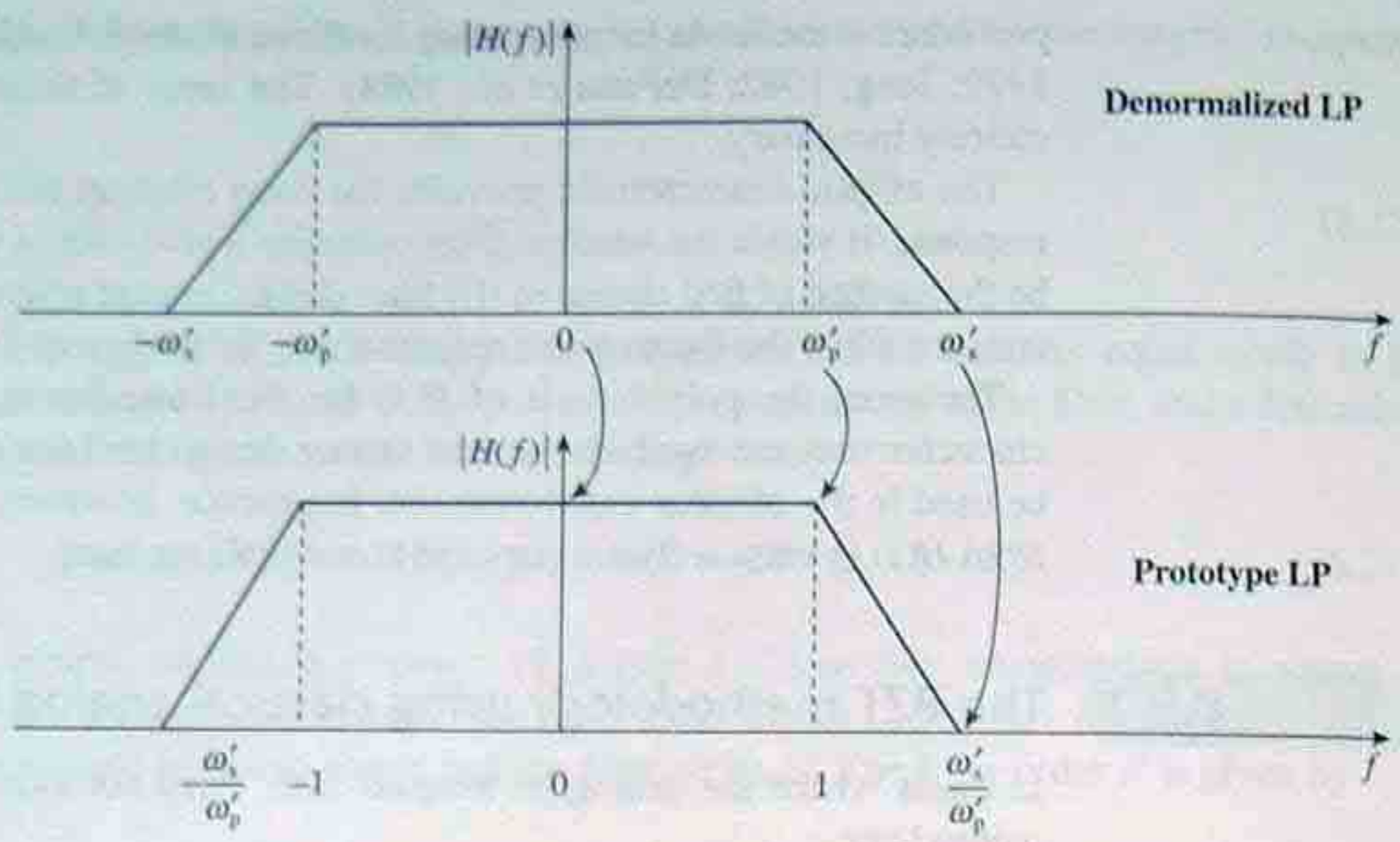


Figure 8.13 Relationships between frequencies in the denormalized LP and prototype LP filters.

$$j\omega^p = j\frac{\omega_{lp}}{\omega'_p}, \text{ i.e. } \omega^p = \frac{\omega_{lp}}{\omega'_p} \tag{8.31}$$

Equation 8.31 defines the relationship between the frequencies in the prototype filter response and those of the denormalized lowpass filter that we wish to design. Given the critical frequencies for a denormalized lowpass filter, we use Equation 8.31 to determine the critical frequencies for the prototype filter and hence its specifications.

The three key critical frequencies for the prototype filter are: 0, passband edge frequency; ω^p_p (this is, in fact, always 1); and the stopband edge frequency, ω^p_s:

(1) when $\omega_{lp} = 0$, $\omega^p = 0$ (from Equation 8.31)

(2) when $\omega_{lp} = \omega'_p$ (i.e. the passband edge frequency), $\omega^p = \omega'_p/\omega'_p = 1 = \omega^p_p$

(3) when $\omega_{lp} = \omega'_s$, $\omega^p = \omega'_s/\omega'_p = \omega^p_s$.

Thus, the critical frequencies for the prototype filter are: 0, 1, ω'_s/ω'_p.

The relationships between the frequencies of the denormalized lowpass filter and those of the prototype filter are shown in Figure 8.13.

8.9.2.2 Highpass filter – basic concepts

From the lowpass-to-highpass transformation, $s = \omega'_p/s$, and denoting the frequencies of the denormalized highpass filter by ω_{hp} and those of the prototype LP filter by ω^p (as before), the following relationship between the frequencies of the prototype LP filter and the desired highpass filter is obtained:

$$\omega^{p} = -\frac{\omega'_{p}}{\omega_{hp}} \tag{8.32}$$

Using Equation 8.32 we can now specify the critical frequencies of the prototype LP filter in terms of those of the desired highpass filter:

(1) when $\omega_{hp} = 0$, $\omega^{p} = \infty$ (using Equation 8.32)

(2) when $\omega_{hp} = \omega'_{p}$ (i.e. the passband edge frequency), $\omega^{p} = -1$

(3) when $\omega_{hp} = \omega'_{s}$, $\omega^{p} = -\dfrac{\omega'_{p}}{\omega'_{s}}$

(4) when $\omega_{hp} = -\omega'_{p}$, $\omega^{p} = 1$

(5) when $\omega_{hp} = -\omega'_{s}$, $\omega^{p} = \dfrac{\omega'_{p}}{\omega'_{s}}$.

Thus, the three key critical frequencies for the prototype lowpass filter for designing the highpass filter are 0, 1 and ω'_{p}/ω'_{s}.

The critical frequencies for the prototype LP filter and their relationships with the frequencies of the denormalized highpass filter are depicted in Figure 8.14. We note that the lowpass-to-highpass transformation maps frequencies in the denormalized highpass filter as follows – it maps zero frequency to infinity, ω_{p} to unity and infinity to zero.

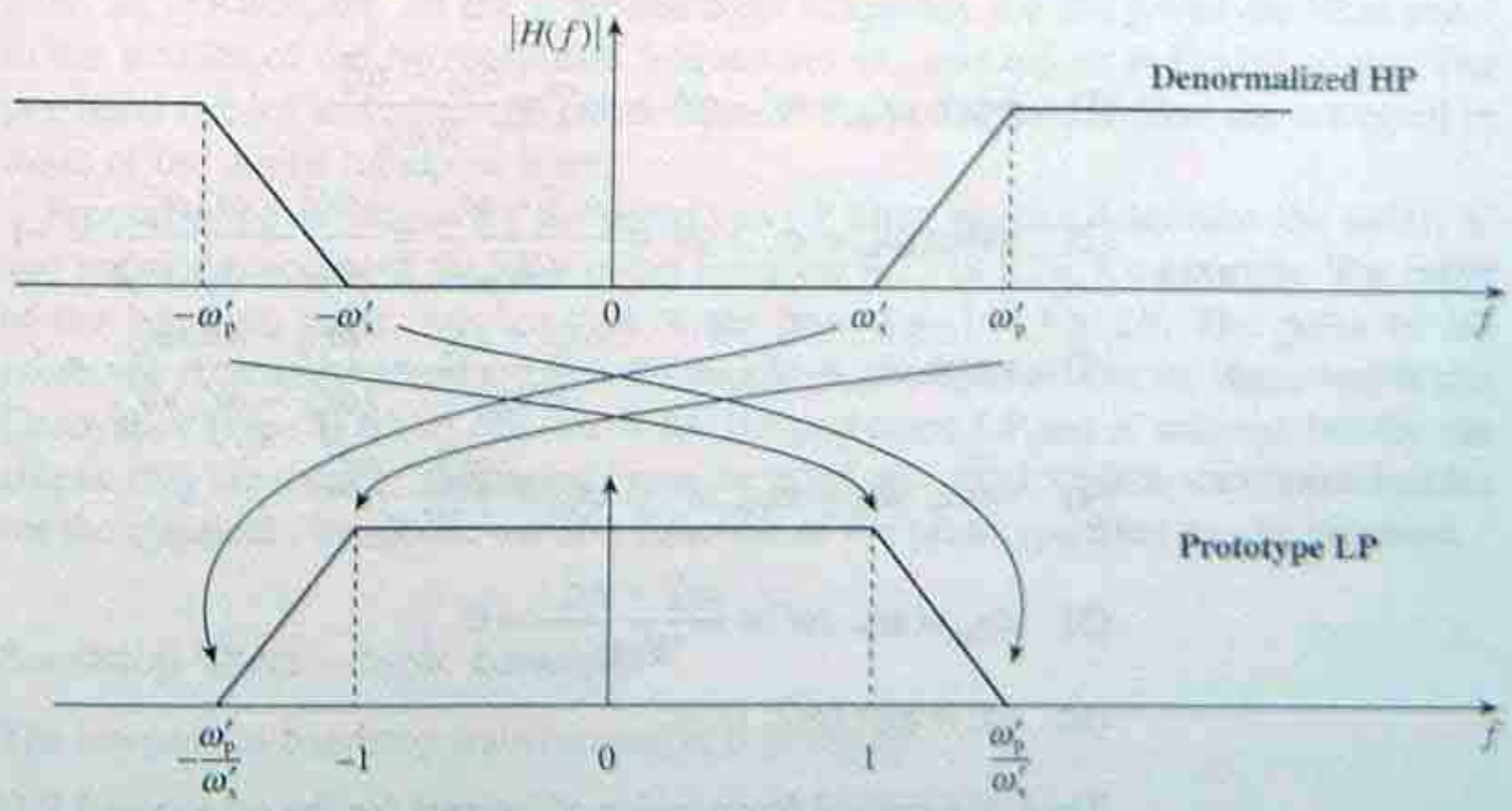


Figure 8.14 Relationships between frequencies in the denormalized HP and prototype LP filters.

8.9.2.3 Bandpass filters – basic concepts

The lowpass-to-bandpass transformation is given by

$$s = \frac{s^2 + \omega_0^2}{Ws}$$

From the lowpass-to-bandpass transformation, the frequencies of the bandpass filter, ω_{bp}, and those of the prototype LPF, ω^p, are related as

$$j\omega^p = \frac{(j\omega_{bp})^2 + \omega_0^2}{jW\omega_{bp}}$$

i.e.

$$\omega^p = \frac{\omega_{bp}^2 - \omega_0^2}{W\omega_{bp}} \tag{8.33}$$

Now, a bandpass filter has four critical or bandedge frequencies and a centre frequency:

$\omega'_{p1}, \omega'_{p2}$ = lower and upper passband edge frequencies

$\omega'_{s1}, \omega'_{s2}$ = lower and upper stopband edge frequencies

ω_0 = centre frequency ($\omega_0^2 = \omega'_{p1}\omega'_{p2}$)

Using the relationship given in Equation 8.33, the bandedge frequencies for the prototype LP filter can be found in terms of the bandedge frequencies for the bandpass filter:

(1) when $\omega_{bp} = \omega'_{s1}$, $\omega^p = \omega'^p_{s1} = \dfrac{\omega'^2_{s1} - \omega_0^2}{W\omega'_{s1}}$

(2) when $\omega_{bp} = \omega'_{p1}$, $\omega^p = \dfrac{\omega'^2_{p1} - \omega_0^2}{W\omega'_{p1}} = \dfrac{\omega'^2_{p1} - \omega'_{p1}\omega'_{p2}}{(\omega'_{p2} - \omega'_{p1})\omega'_{p1}} = -1$

(3) when $\omega_{bp} = \omega'_{p2}$, $\omega^p = \dfrac{\omega'^2_{p2} - \omega_0^2}{W\omega'_{p2}} = \dfrac{\omega'^2_{p2} - \omega'_{p1}\omega'_{p2}}{(\omega'_{p2} - \omega'_{p1})\omega'_{p2}} = 1$

(4) when $\omega_{bp} = \omega'_{s2}$, $\omega^p = \omega'^p_{s2} = \dfrac{\omega'^2_{s2} - \omega_0^2}{W\omega'_{s2}}$

(5) $\omega_{bp} = \omega_0$, $\omega^p = \dfrac{\omega'^2_0 - \omega_0^2}{W\omega_0^2} = 0$

(6) $\omega^p_s = \min(\omega'^p_{s1}, \omega'^p_{s2})$.

Thus, the critical frequencies of interest for the prototype LP filter are

$0, 1, \min(\omega^p_{s1}, |\omega^p_{s2}|)$

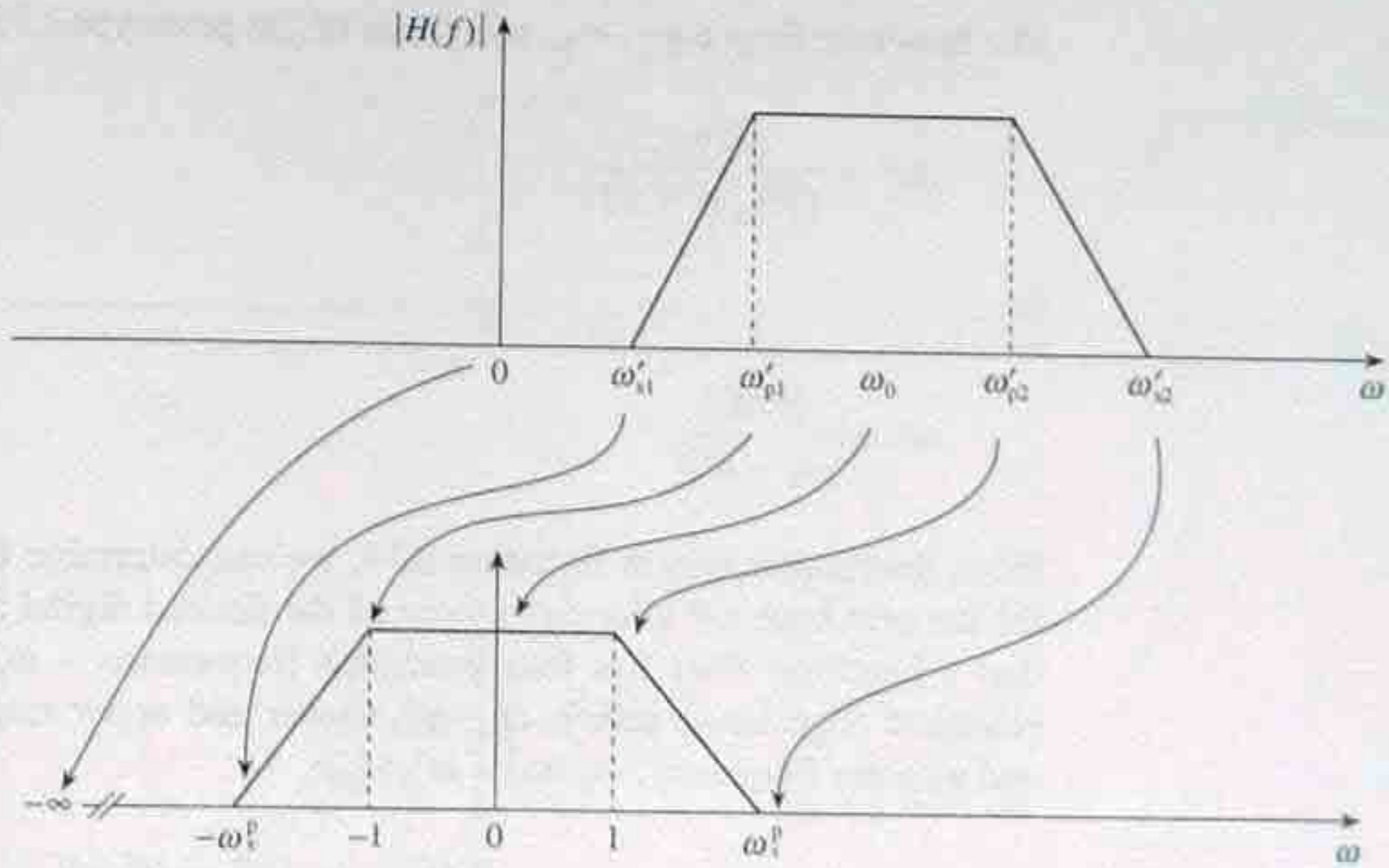


Figure 8.15 Mapping of the prototype LP to BPF.

The mappings of the frequencies between the bandpass filter and the prototype LP filter are depicted in Figure 8.15. We note, for example, that the centre frequency of the bandpass filter is mapped to zero in the prototype filter, and the upper passband and stopband edge frequencies, ω'_{p2} and ω'_{s2}, respectively, are mapped to the positive passband and stopband edge frequencies in the prototype filter. On the other hand, the lower passband and stopband edge frequencies, ω'_{p1} and ω'_{s1}, respectively, are mapped to the negative passband and stopband edge frequencies in the prototype filter. In practice, we set the stopband edge frequency for the prototype filter equal to the smaller of the two stopband frequencies ω^p_{s1} and ω^p_{s2} as indicated above. The passband ripples and stopband attenuation of the prototype LP filter are set equal to those of the digital bandpass filter.

From the specifications for the prototype LP filter, we can determine the order, N, and transfer function of the filter using Equation 8.25 or 8.28, for eaxmple. The order of the bandpass filter is twice that of the prototype LP, i.e. $2N$. The poles of the prototype filter are determined from Equations 8.26 and 8.29. For the Butterworth and Chebyshev (type I) filters, the zeros for the prototype LP are at infinity, but for the elliptic they are entirely imaginary. From the poles and zero locations (or standard tables for the classical filters), the transfer function of the prototype filter can be obtained.

8.9.2.4 Bandstop filters – basic concepts

The lowpass-to-bandstop transformation is given by

$$s = \frac{Ws}{s^2 + \omega_0^2}$$

The bandstop frequency, ω_{bs}, and those of the prototype LPF, ω^p, are related as

$$j\omega^p = \frac{jW\omega_{bs}}{(j\omega_{bs})^2 + \omega_0^2}$$

i.e.

$$\omega^p = \frac{W\omega_{bs}}{\omega_0^2 - \omega_{bs}^2} \tag{8.34}$$

From the relationship in Equation 8.34, we can determine the bandedge frequencies for the prototype LP filter from those of the desired digital bandstop filter. We recall that a bandstop filter has four bandedge frequencies – ω'_{p1}, ω'_{p2} (lower and upper passband edge frequencies), ω'_{s1}, ω'_{s2} (lower and upper stopband edge frequencies), and a centre frequency, ω_0 ($\omega_0^2 = \omega'_{p1}\omega'_{p2}$):

(1) when $\omega_{bs} = \omega'_{p1}$, $\omega^p = \dfrac{W\omega'_{p1}}{\omega_0'^2 - \omega_{p1}'^2} = \dfrac{(\omega'_{p2} - \omega'_{p1})\omega'_{p1}}{\omega'_{p1}\omega'_{p2} - \omega_{p1}'^2} = 1$

(2) when $\omega_{bs} = \omega'_{s1}$, $\omega^p = \omega_s^{p(1)} = \dfrac{W\omega'_{s1}}{\omega_0^2 - \omega_{s1}^2}$

(3) when $\omega_{bs} = \omega'_{s2}$, $\omega^p = \omega_s^{p(2)} = \dfrac{W\omega'_{s2}}{\omega_0^2 - \omega_{s2}^2}$

(4) when $\omega_{bs} = \omega_0$, $\omega^p = \dfrac{W\omega_0}{\omega_0^2 - \omega_0^2} = \infty$

(5) when $\omega_{bs} = \omega'_{p2}$, $\omega^p = \dfrac{W\omega'_{p2}}{\omega_0^2 - \omega_{p2}'^2} = \dfrac{(\omega'_{p2} - \omega'_{p1})\omega'_{p2}}{\omega'_{p1}\omega'_{p2} - \omega_{p2}'^2} = -1.$

Thus, the stopband edge frequency for the prototype LP filter, $\omega_s^p = \min(\omega_s^{p(1)}, \omega_s^{p(2)})$ and its passband edge frequency is 1. The passband ripple and stopband attenuation are, respectively, A_p and A_s. The mappings between the frequencies of the bandstop filter and those of the prototype lowpass filter are shown in Figure 8.16. We see that the upper bandstop and bandpass edge frequencies in the bandstop filter are mapped to the negative frequencies in the prototype filter whereas the lower passband and stopband edge frequencies are mapped to the positive frequencies in the prototype filter.

The critical frequencies of interest for the prototype LP filter are

$$0,\ 1,\ \omega_s^p \text{ (where } \omega_s^p = \min(\omega_s^{p(1)}, \omega_s^{p(2)})).$$

From the specifications for the prototype LP filter, we can determine the order, N, and transfer function of the filter using Equation 8.25 or 8.28, for example. The order of the bandpass filter is twice that of the prototype LP, i.e. $2N$.

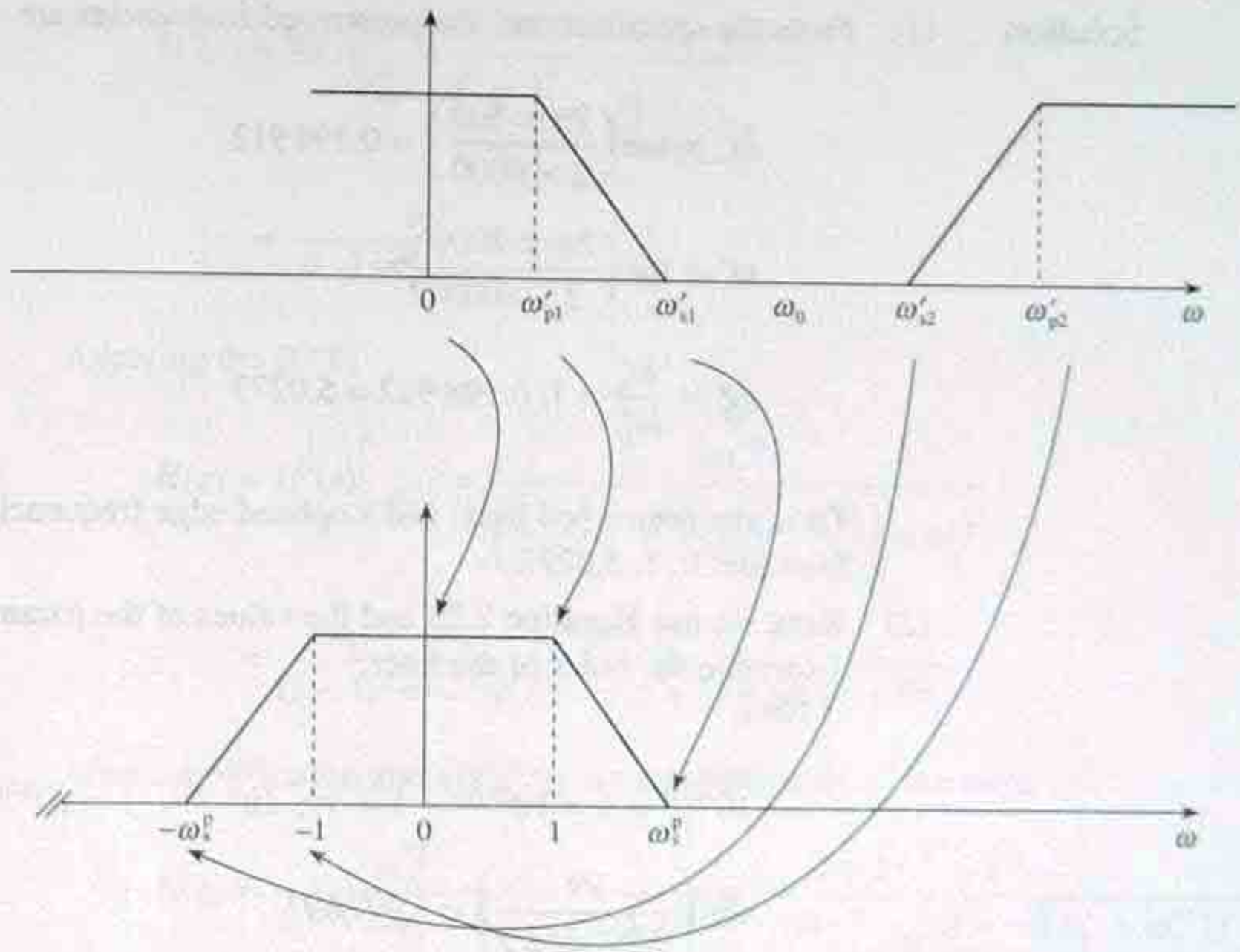


Figure 8.16 Relationships between the frequencies in the denormalized BS and prototype LP filters.

8.9.3 Illustrative design examples (lowpass, highpass, bandpass and bandstop filters)

Example 8.11 *Lowpass filter* A lowpass digital filter meeting the following specifications is required:

passband	0–500 Hz
stopband	2–4 kHz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	8 kHz

Determine the following:

(1) pass- and stopband edge frequencies for a suitable analog prototype lowpass filter;

(2) order, N, of the prototype lowpass filter;

(3) coefficients, and hence the transfer function, of the discrete-time filter using the bilinear z-transform.

Assume a Butterworth characteristic for the filter.

Solution (1) From the specifications, the prewarped frequencies are

$$\omega_p' = \tan\left(\frac{2\pi \times 500}{2 \times 8000}\right) = 0.198\,912$$

$$\omega_s' = \tan\left(\frac{2\pi \times 2000}{2 \times 8000}\right) = 1$$

$$\omega_s^p = \frac{\omega_s'}{\omega_p'} = 1/0.198\,912 = 5.0273$$

Thus, the prewarped pass- and stopband edge frequencies for the prototype LP filter are: 0, 1, 5.0273.

(2) Next, we use Equation 8.25 and the values of the parameters above to determine the order of the filter.

Now

$$10^{A_s/10} - 1 = 10^{20/10} - 1 = 99;\ 10^{A_p/10} - 1 = 10^{3/10} - 1 = 0.995\,262\,3;$$

$$\log\left(\frac{99}{0.995\,262\,3}\right) = 1.997\,697$$

For the prototype LPF

$$\omega_p^p = 1;\quad \omega_s^p = 5.0273;\quad \log\left(\frac{\omega_s^p}{\omega_p^p}\right) = 2\log(5.0273) = 1.402\,66.$$

$$N \geqslant \frac{1.997\,697}{1.402\,66} = 1.424.\ \text{Let } N = 2$$

(3) The poles of the prototype filter are (from Equation 8.26)

$$s_{p,1} = \cos\left[\frac{(2+2-1)\pi}{4}\right] + j\sin\left[\frac{(2+2-1)\pi}{4}\right] = -\frac{\sqrt{2}}{2} + j\frac{\sqrt{2}}{2}$$

$$s_{p,2} = -\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}$$

The s-plane transfer function, $H(s)$, is

$$H(s) = \frac{1}{(s - s_{p,1})(s - s_{p,2})} = \frac{1}{s^2 + \sqrt{2}s + 1}$$

The frequency scaled s-plane transfer function is

$$H'(s) = H(s)\Big|_{\frac{s}{\omega'_p}} = \frac{1}{\left(\frac{s}{\omega'_p}\right)^2 + \sqrt{2}\frac{s}{\omega'_p} + 1}$$

$$= \frac{\omega'^2_p}{s^2 + \sqrt{2}s\omega'_p + \omega'^2_p}$$

Applying the BZT:

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}} = \frac{\omega'^2_p}{\left(\frac{z-1}{z+1}\right)^2 + \sqrt{2}\omega'_p\left(\frac{z-1}{z+1}\right) + \omega'^2_p}$$

$$= \frac{\omega'^2_p(z+1)^2}{(z-1)^2 + \sqrt{2}\omega'_p(z-1)(z+1) + \omega'^2_p(z+1)^2}$$

After simplification and dividing top and bottom by z^2, we have

$$H(z) = \frac{\omega'^2_p}{1 + \sqrt{2}\omega'_p + \omega'^2_p} \times \frac{1 + 2z^{-1} + z^{-2}}{1 + \frac{2(\omega'^2_p - 1)z^{-1}}{1 + \sqrt{2}\omega'_p + \omega'^2_p} + \frac{(1 - \sqrt{2}\omega'_p + \omega'^2_p)z^{-2}}{1 + \sqrt{2}\omega'_p + \omega'^2_p}}$$

Using the values of the parameters

$$1 + \sqrt{2}\omega'_p + \omega'^2_p = 1.320\,87;\ \omega'^2_p - 1 = -0.960\,43$$

$$1 - \sqrt{2}\omega'_p + \omega'^2_p = 0.758\,285\,8;\ \omega'^2_p = 0.039\,565\,9$$

and substituting in the equation above and simplifying, we have

$$H(z) = \frac{0.029\,95(1 + 2z^{-1} + z^{-2})}{1 - 1.4542z^{-1} + 0.574\,08z^{-2}}$$

Example 8.12 *Highpass filter* A highpass digital filter meeting the following specifications is required:

passband	2–4 kHz
stopband	0–500 Hz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	8 kHz

Determine the following:

(1) pass- and stopband edge frequencies for a suitable analog prototype lowpass filter;

(2) order, N, of the prototype lowpass filter;

(3) coefficients, and hence the transfer function, of the discrete-time filter using the bilinear z-transform.

Assume a Butterworth characteristic for the filter.

Solution (1) From the specifications, the prewarped frequencies are

$$\omega_s' = \tan\left(\frac{2\pi \times 500}{2 \times 8000}\right) = 0.198\,912$$

$$\omega_p' = \tan\left(\frac{2\pi \times 2000}{2 \times 8000}\right) = 1$$

$$\omega_s^p = \frac{\omega_p'}{\omega_s'} = 1/0.198\,912 = 5.0273$$

Thus, the pass- and stopband edge frequencies for the prototype LP filter are: 0, 1, 5.0273.

(2) Next, we use Equation 8.25 and the values of the parameters above to determine the order of the filter.

Now

$$10^{A_s/10} - 1 = 10^{20/10} - 1 = 99;\ 10^{A_p/10} - 1 = 10^{3/10} - 1 = 0.995\,262\,3$$

$$\log\left(\frac{99}{0.995\,262\,3}\right) = 1.997\,697$$

For the prototype LPF

$$\omega_p^p = 1;\quad \omega_s^p = 5.0273;\quad \log\left(\frac{\omega_s^p}{\omega_p^p}\right) = 2\log(5.0273) = 1.402\,66$$

$$N \geqslant \frac{1.997\,697}{1.402\,66} = 1.424$$

Let $N = 2$.

The poles of the prototype filter are (from Equation 8.26)

$$s_{p,1} = \cos\left[\frac{(2+2-1)\pi}{4}\right] + j\sin\left[\frac{(2+2-1)\pi}{4}\right] = -\frac{\sqrt{2}}{2} + j\frac{\sqrt{2}}{2}$$

$$s_{p,2} = -\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}$$

Example

The s-plane transfer function, $H(s)$, is

$$H(s) = \frac{1}{(s - s_{p,1})(s - s_{p,2})} = \frac{1}{s^2 + \sqrt{2}s + 1}$$

The frequency scaled s-plane transfer function is

$$H'(s) = H(s)\Big|_{\frac{\omega'_p}{s}} = \frac{1}{\left(\frac{\omega'_p}{s}\right)^2 + \sqrt{2}\frac{\omega'_p}{s} + 1}$$

$$= \frac{s^2}{s^2 + \sqrt{2}s\omega'_p + \omega'^2_p}$$

Applying the BZT:

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}} = \frac{\left(\frac{z-1}{z+1}\right)^2}{\left(\frac{z-1}{z+1}\right)^2 + \sqrt{2}\omega'_p\left(\frac{z-1}{z+1}\right) + \omega'^2_p}$$

$$= \frac{(z-1)^2}{(z-1)^2 + \sqrt{2}\omega'_p(z-1)(z+1) + \omega'^2_p(z+1)^2}$$

After simplification and dividing top and bottom by z^2, we have

$$H(z) = \frac{1}{1 + \sqrt{2}\omega'_p + \omega'^2_p} \times \frac{1 - 2z^{-1} + z^{-2}}{1 + \frac{2(\omega'^2_p - 1)z^{-1}}{1 + \sqrt{2}\omega'_p + \omega'^2_p} + \frac{(1 - \sqrt{2}\omega'_p + \omega'^2_p)z^{-2}}{1 + \sqrt{2}\omega'_p + \omega'^2_p}}$$

Using the values of the parameters

$$1 + \sqrt{2}\omega'_p + \omega'^2_p = 3.414\,21;\ \omega'^2_p - 1 = 0$$

$$1 - \sqrt{2}\omega'_p + \omega'^2_p = 0.585\,786;\ \omega'^2_p = 1$$

and substituting in the equation above and simplifying, we have

$$H(z) = \frac{0.292\,89(1 - 2z^{-1} + z^{-2})}{1 + 0.171\,57z^{-2}}$$

Example 8.13 *Bandpass filter* A requirement exists for a bandpass digital filter, with a Butterworth magnitude–frequency response, that satisfies the following specifications:

lower passband edge frequency	200 Hz
upper passband edge frequency	300 Hz
lower stopband edge frequency	50 Hz
upper stopband edge frequency	450 Hz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	1 kHz

Determine the following:

(1) pass- and stopband edge frequencies of a suitable prototype lowpass filter;
(2) order, N, of the prototype lowpass filter;
(3) coefficients, and hence the transfer function, of the discrete-time filter using the BZT method.

Solution The prewarped critical frequencies for the bandpass filter are

$$\omega_0 = 1; \qquad W = 0.6498$$
$$\omega'_{p1} = 0.7265; \qquad \omega'_{p2} = 1.376\,38$$
$$\omega'_{s1} = 0.1584; \qquad \omega'_{s2} = 6.3138$$

Thus, the bandedge frequencies for the prototype LP filter are (using the relationships above)

$$\omega_p^p = 1; \qquad \omega_s^p = 9.4721$$

Thus, we require a prototype LPF with $\omega_p^p = 1$; $\omega_s^p = 9.4721$, $A_p = 3$ dB; $A_s = 20$ dB. From Equation 8.25, the order of the prototype LPF is obtained as

$$10^{A_s/10} - 1 = 10^{20/10} - 1 = 99; \; 10^{A_p/10} - 1 = 10^{3/10} - 1 = 0.9952623;$$

$$\log\left(\frac{99}{0.995\,262\,3}\right) = 1.997\,697\,6; \quad \frac{\omega_s^p}{\omega_p^p} = 9.4721; \quad 2\log(9.4721) = 1.952\,89$$

$$N = \frac{1.997\,697\,6}{1.952\,89} = 1.0229$$

N must be an integer, and for simplicity we will use $N = 1$. The s-plane transfer function for a first-order prototype LP filter is given by

$$H(s) = \frac{1}{s+1}$$

Using the lowpass-to-bandpass transformation from the table, we obtain

$$H'(s) = H(s)\Big|_{s=\frac{s^2+\omega_0^2}{Ws}} = \frac{1}{\left(\dfrac{s^2+\omega_0^2}{Ws}\right)+1}$$

$$= \frac{Ws}{s^2 + Ws + \omega_0^2}$$

Applying the BZT gives

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}}$$

$$= \frac{W\left(\dfrac{z-1}{z+1}\right)}{\left(\dfrac{z-1}{z+1}\right)^2 + W\left(\dfrac{z-1}{z+1}\right) + \omega_0^2}$$

Substituting the values for ω_0^2 and W, and simplifying, we have

$$H(z) = \frac{0.2452(1 - z^{-2})}{1 + 0.5095\,z^{-2}}$$

Example 8.14 *Bandstop filter* A requirement exists for a bandstop digital IIR filter, with a Butterworth magnitude–frequency response, that meets the following specifications:

lower passband	0–50 Hz
upper passband	450–500 Hz
stopband	200–300 Hz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	1 kHz

Determine the following:

(1) pass- and stopband edge frequencies of a suitable prototype lowpass filter;

(2) order, N, of the prototype lowpass filter;

(3) coefficients, and hence the transfer function, of the discrete-time filter using the BZT method.

Solution The prewarped critical frequencies for the bandpass filter are

$$\omega_0 = 1; \qquad W = 6.1554$$
$$\omega'_{p1} = 0.1584; \qquad \omega'_{p2} = 6.3138$$
$$\omega'_{s1} = 0.7265; \qquad \omega'_{s2} = 1.376\,38$$

Thus, the bandedge frequencies for the prototype filter are (using the relationships above)

$$\omega_p^p = 1; \qquad \omega_s^p = 9.4721$$

We require a prototype LPF with $\omega_p^p = 1$, $\omega_s^p = 9.4721$, $A_p = 3$ dB, $A_s = 20$ dB. From Equation 8.25, the order of the prototype LPF is obtained as

$$10^{A_s/10} - 1 = 10^{20/10} - 1 = 99; \; 10^{A_p/10} - 1 = 10^{3/10} - 1 = 0.995\,262\,3$$

$$\log\left(\frac{99}{0.995\,262\,3}\right) = 1.997\,697\,6; \quad \frac{\omega_s^p}{\omega_p^p} = 9.4721; \quad 2\log(9.4721) = 1.952\,89$$

$$N = \frac{1.997\,697\,6}{1.952\,89} = 1.0229$$

N must be an integer, and for simplicity we will use $N = 1$. The s-plane transfer function for a first-order prototype LP filter is given by

$$H(s) = \frac{1}{s+1}$$

Using the lowpass-to-bandstop transformation from the table, we obtain

$$H'(s) = H(s)\Big|_{s=\frac{Ws}{s^2+\omega_0^2}} = \frac{1}{\left(\dfrac{Ws}{s^2+\omega_0^2}\right)+1}$$

$$= \frac{s^2+\omega_0^2}{s^2+Ws+\omega_0^2}$$

Applying the BZT gives

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}}$$

$$= \frac{\left(\dfrac{z-1}{z+1}\right)^2+\omega_0^2}{\left(\dfrac{z-1}{z+1}\right)^2+W\left(\dfrac{z-1}{z+1}\right)+\omega_0^2}$$

Substituting the values for ω_0^2 and W, and simplifying, we have

$$H(z) = \frac{0.2452(1 + z^{-2})}{1 - 0.5095\, z^{-2}}$$

Example 8.15 Obtain the transfer function of a lowpass digital filter meeting the following specifications:

passband	0–60 Hz
stopband	>85 Hz
stopband attenuation	>15 dB

Assume a sampling frequency of 256 Hz and a Butterworth characteristic.

Solution This example illustrates how to combine steps 4 and 5 in the BZT process into one, as suggested by Equation 8.23.

(1) The critical frequencies for the digital filter are

$$\omega_1 T = \frac{2\pi f_1}{F_s} = \frac{2\pi 60}{256} = 2\pi \times 0.2344$$

$$\omega_2 T = \frac{2\pi f_2}{F_s} = \frac{2\pi 85}{256} = 2\pi \times 0.3320$$

(2) The prewarped equivalent analog frequencies are

$$\omega_1' = \tan\left(\frac{\omega_1 T}{2}\right) = 0.906\,347;\ \omega_2' = \tan\left(\frac{\omega_2 T}{2}\right) = 1.715\,80$$

(3) Next we need to obtain $H(s)$ with Butterworth characteristics, a 3 dB cutoff frequency of 0.906 347, and a response at 85 Hz that is down by 15 dB. For an attenuation of 15 dB and a passband ripple of 3 dB, from Equation 8.25 $N = 2.68$. We use $N = 3$, since it must be an integer. A normalized third-order filter is given by

$$H(s) = \frac{1}{(s+1)(s^2+s+1)} = \frac{1}{s+1}\,\frac{1}{s^2+s+1}$$
$$= H_1(s)\, H_2(s)$$

$$\cot\left(\frac{\omega_1 T}{2}\right) = \cot\left(\frac{2\pi \times 0.2344}{2}\right) = 1.103\,155$$

Performing the transform in two stages, one for each of the factors of $H(s)$ above, we obtain

$$H_2(z) = H_2(s)\big|_{s=\cot(\omega_c T/2)[(z-1)/(z+1)]}$$

$$= 0.3012\,\frac{1 + 2z^{-1} + z^{-2}}{1 - 0.1307z^{-1} + 0.3355z^{-2}}$$

which we have arrived at after considerable manipulation. Similarly, we obtain $H_1(z)$ as

$$H_1(z) = 0.4754\,\frac{1 + z^{-1}}{1 - 0.0490z^{-1}}$$

$H_1(z)$ and $H_2(z)$ may then be combined to give the desired transfer function, $H(z)$:

$$H(z) = H_1(z)H_2(z) = 0.1432\,\frac{1 + 3z^{-1} + 3z^{-2} + z^{-3}}{1 - 0.1801z^{-1} + 0.3419z^{-2} - 0.0165z^{-3}}$$

8.10 Calculating IIR filter coefficients by mapping *s*-plane poles and zeros

8.10.1 Basic concepts

An alternative, and perhaps more powerful and flexible, method of computing the coefficients of $H(z)$ for practical IIR filters is to map the individual poles and zeros of a suitable analog filter from the s-plane into the z-plane and then to derive the digital filter coefficients from the z-plane poles and zeros. This approach is exploited by a number of commercial software implementations and is attractive when the filter is of a high order.

The procedure for calculating IIR coefficients by mapping s-plane poles and zeros into the z-plane is summarized below.

8.10.1.1 Step 1

As before, the designer starts with a normalized, Nth-order analog lowpass filter of Butterworth, Chebyshev or elliptic type, depending on the design requirements. The poles of the normalized LPF are then obtained using Equation 8.26 for a Butterworth or Equation 8.29 for a Chebyshev filter. For an elliptic filter each pole is complex and, in general, has the form

$$s_{l,k} = \alpha_{p,k} + j\beta_{p,k} \tag{8.35}$$

For Butterworth and Chebyshev (type I) filters, the zeros for the prototype LPF are at infinity, but for elliptic filters they are entirely imaginary. In general, the locations of the zeros of the normalized LPF are easier to determine than those of the poles.

8.10.1.2 Step 2

Next, the poles and zeros of the normalized analog LPF are converted into those of an LP, HP, BP or BSF using an appropriate transformation in Equations 8.21a–8.21d.

Lowpass and highpass filters

For a lowpass or highpass digital filter, the N poles of the normalized LP are transformed as follows (see Equations 8.21a and 8.21b):

$$s_{l,k} = s_{l,k}/\omega'_p \quad k = 1, 2, \ldots, N \text{ lowpass-to-lowpass} \tag{8.36a}$$

$$s_{h,k} = \omega'_p/s_{l,k} \quad k = 1, 2, \ldots, N \text{ lowpass-to-highpass} \tag{8.36b}$$

where ω'_p is the desired passband edge frequency, $s_{l,k}$ are the poles of the analog lowpass filter and $s_{h,k}$ are the poles of the analog highpass filter.

The similarity between Equations 8.36a and 8.36b is evident and is due to the duality between the lowpass and highpass characteristics. For N even, there will be $N/2$ complex pole pairs. For N odd there will be $(N-1)/2$ complex pole pairs and a single real pole.

For the classic filters – Butterworth, Chebyshev and elliptic – the transformations of Equations 8.36a and 8.36b map the zeros of the prototype LPF onto the imaginary axis in the s-plane. In the case of the Butterworth or Chebyshev filters, the zeros of the prototype filter are at infinity. In either case, the transformations map the zeros from infinity to infinity (for lowpass filters) or from infinity to the origin (for highpass filters), as shown in Figures 8.17(a)(ii) and 8.17(b)(ii).

Bandpass and bandstop filters

For bandpass digital filters, the analog BPF poles are obtained from those of the normalized prototype LPF using the transformation

$$s_{l,k} = \frac{s_{b,k}^2 + \omega_0^2}{W s_{b,k}} \tag{8.37}$$

where $s_{l,k}$ are the poles of the prototype analog LPF, $s_{b,k}$ are the pole pairs of the intermediate analog BPF, $W = \omega'_2 - \omega'_1$ is the width of the filter passband and $\omega_0^2 = \omega'_1\omega'_2$ gives the centre frequency of the passband. Equation 8.37 yields the following quadratic equation in $s_{b,k}$:

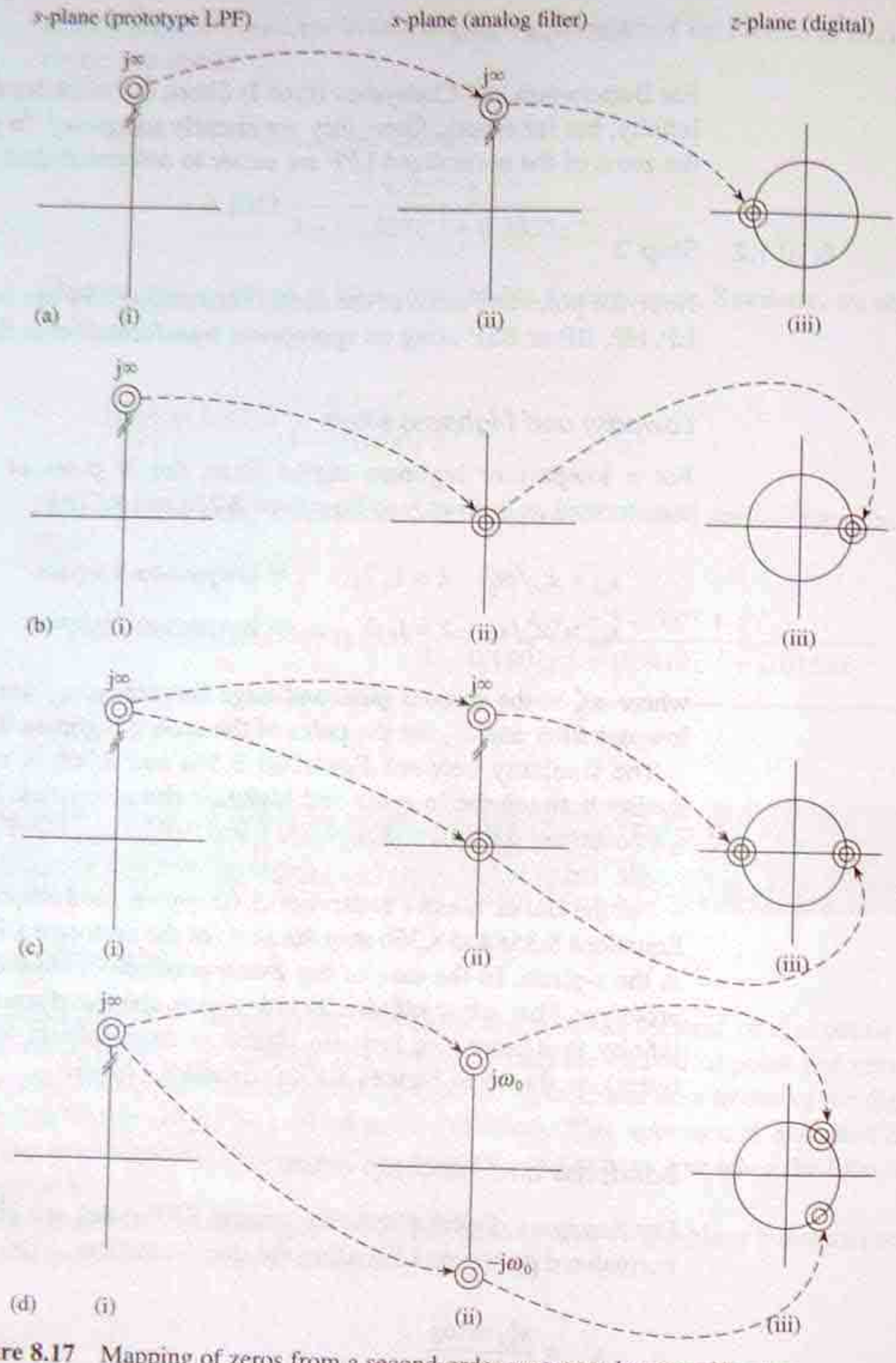


Figure 8.17 Mapping of zeros from a second-order prototype lowpass filter into (a) lowpass, (b) highpass, (c) bandpass and (d) bandstop.

$$s_{b,k}^2 - Ws_{l,k}s_{b,k} + \omega_0^2 = 0 \tag{8.38}$$

Solving for $s_{b,k}$ leads to the following expression for the bandpass analog poles:

$$s_{b,k} = \frac{W}{2}\left[s_{l,k} \pm \left(s_{l,k}^2 - \frac{4\omega_0^2}{W^2}\right)^{1/2}\right] \tag{8.39}$$

It is seen from Equation 8.39 that each analog LP pole, $s_{l,k}$ leads to a pair of analog BPF poles as a result of the s^2 term in the transformation. In general, the analog lowpass filter pole $s_{l,k}$ is complex and so the square-root term is complex. The evaluation of a complex square root using real arithmetic requires some care if the correct solution is to be obtained (see Appendix 8C).

For the bandstop digital filters, the following lowpass-to-bandstop transformation may be used:

$$s_{l,k} = \frac{Ws_{r,k}}{s_{r,k}^2 + \omega_0^2} \tag{8.40}$$

where $s_{l,k}$ are the poles of the analog prototype LPF, $s_{r,k}$ are the poles of the intermediate analog bandstop filter, W is the width of the stopband and ω_0^2 is the centre frequency in the stopband. Equation 8.40 leads to the following expression for obtaining analog bandstop poles from the prototype LPF:

$$s_{r,k} = \frac{W}{2}\left[s_{l,k}^{-1} \pm \left(s_{l,k}^{-2} - \frac{4\omega_0^2}{W^2}\right)^{1/2}\right] \tag{8.41}$$

Not surprisingly, Equation 8.41 has the same format as Equation 8.39 except for the inversion of the LPF poles due to the duality of the BPF and BSF.

As in the case of the lowpass and highpass filters, the transformations of Equations 8.39 and 8.41 map the zeros onto the imaginary axis. For Butterworth and Chebyshev filters, the lowpass-to-bandpass transformation maps the N zeros of the prototype lowpass filter from infinity to infinity and to the origin in the s-plane (see Figure 8.17(c)). The lowpass-to-bandstop transformation on the other hand maps the N zeros of the prototype lowpass filter from infinity to $\pm j\omega_0$ in the s-plane; see Figure 8.17(d). For elliptic bandpass and bandstop filters, the transformation maps the zeros of the prototype LPF which are entirely imaginary to other points on the $j\omega$ axis.

8.10.1.3 Step 3

The BZT is then used to map the poles and zeros from the s-plane into the digital z-plane. Each s-plane pole, $s_{p,k}$, is mapped as follows:

$$z_{p,k} = \frac{1 + s_{p,k}}{1 - s_{p,k}} \tag{8.42}$$

Similarly, each s-plane zero, $s_{z,k}$ of the transformed analog filters is mapped onto the z-plane as follows:

$$z_{z,k} = \frac{1 + s_{z,k}}{1 - s_{z,k}} \tag{8.43}$$

Figures 8.17(a)–8.17(d) illustrate the way the zeros are mapped from the s-plane to the z-plane via the BZT for the Butterworth and Chebyshev filters. Note, for example, that for lowpass filters, Figure 8.17(a)(ii), the zeros are at infinity in the s-plane and the BZT of Equation 8.43 maps these to the point $z = -1$ in the z-plane, Figure 8.17(a)(iii), whereas for the bandpass filter the s-plane zeros, Figure 8.17(c)(ii), are mapped from the origin to the point $z = 1$, and from infinity to $z = -1$.

For the elliptic lowpass, highpass, bandpass and bandstop filters, $s_{z,k}$ are imaginary and the BZT maps these onto the unit circle in the z-plane. In general, the z-plane zeros for all the classical filters (Butterworth, Chebyshev and elliptic) obtained via the BZT lie on the unit circle, regardless of the filter type. As a result, the numerator coefficients of $H(z)$ of classical filters are always integers (0, ±1, ±2).

8.10.1.4 Step 4

The final step is to determine the numerator and denominator coefficients of the second- and/or first-order filter sections. This is achieved by combining complex conjugate pole and zero pairs as follows:

$$H_i(z) = \frac{(z - z_{z,k})(z - z^*_{z,k})}{(z - z_{p,k})(z - z^*_{p,k})} \tag{8.44a}$$

$$= \frac{1 + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \tag{8.44b}$$

In general, a complex pole or zero pair located at $\alpha \pm j\beta$ leads to a quadratic in z of the form

$$[z - (\alpha + j\beta)][z - (\alpha - j\beta)] = z^2 - 2\alpha z + \alpha^2 + \beta^2$$
$$= 1 - 2\alpha z^{-1} + (\alpha^2 + \beta^2)z^{-2} \tag{8.45a}$$

A single pole or zero on the real axis (that is, at $z = \pm\alpha$) leads to a first-order factor of the form

$$z \pm \alpha = 1 \pm \alpha z^{-1} \tag{8.45b}$$

Often we find that an Nth-order filter has N real zeros on the real axis in the z-plane. In this case, we may pair the z-plane zeros so that the numerator of each filter section is a quadratic of the form

$$1 \pm 2z^{-1} + z^{-2} \tag{8.45c}$$

The overall transfer function, $H(z)$, is given by

$$H(z) = KH_1(z)H_2(z) \ldots H_M(z)$$

where K is a gain factor which is used to adjust the magnitude response in the passband to a desired level. In most cases, K is set to a value to make the maximum response in the passband equal to unity.

8.10.2 Illustrative examples

Example 8.16 A second-order digital bandpass filter (BPF), with Butterworth characteristics and a passband between 200 and 300 Hz at a sampling frequency of 2 kHz, is required for a certain DSP application. Determine the transfer function of the digital filter by mapping the s-plane poles and/or zeros of a suitable prototype analog lowpass filter into the z-plane.

Sketch and label the pole–zero diagrams of the prototype LPF, intermediate analog BPF, and the digital BPF.

Solution A first-order normalized LPF is required, since the order of the filter is doubled by the bandpass transformation. Thus

$$H(s) = \frac{1}{s+1}$$

This transfer function has a single pole at $s_{L1} = -1$. Now, $F_s = 2\text{ kHz} = 1/T$. Thus the prewarped bandedge frequencies are

$$\omega_1' = \tan\left(\frac{\omega_1 T}{2}\right) = \tan\left(\frac{2\pi \times 200}{2 \times 2000}\right) = 0.3249$$

$$\omega_2' = \tan\left(\frac{\omega_2 T}{2}\right) = \tan\left(\frac{2\pi \times 300}{2 \times 2000}\right) = 0.5095$$

Thus ω_0 and W are given by

$$\omega_0^2 = \omega_1'\omega_2' = 0.1655,\ W = \omega_2' - \omega_1' = 0.1846$$

The single pole for the LPF is transformed into two poles for the BPF, using Equation 8.39, as follows:

$$s_{b,1} = \frac{0.1846}{2}\left\{-1 + \left[(-1)^2 - \frac{4 \times 0.1655}{(0.1846)^2}\right]^{1/2}\right\}$$

$$= -0.0923 + 0.4172j$$

$$s_{b,2} = \frac{0.1846}{2}\left\{-1 - \left[(-1)^2 - \frac{4 \times 0.1655}{(0.1846)^2}\right]^{1/2}\right\}$$

$$= -0.0923 - 0.4172j = s_{b,1}^*$$

From the BZT transformation, Equation 8.42, we have

$$z_{p,1} = \frac{1 - 0.0923 + 0.4172j}{1 + 0.0923 - 0.4172j} = 0.5979 + 0.6103j$$

$$z_{p,2} = z_{p,1}^*$$

The prototype LPF has a single zero at infinity. This is mapped by the lowpass-to-bandpass transformation to the origin and to infinity in the bandpass s-plane. That is, $s_{z,1} = 0$, $s_{z,2} = \infty$. The BZT maps these zeros to the points $z = 1$ and $z = -1$ in the z-plane.

$$s_{z,1} \rightarrow z_{z,1} = 1; \quad s_{z,2} \rightarrow z_{z,2} = -1$$

We can now determine the discrete transfer function, $H(z)$, from the poles and zeros:

$$H(z) = \frac{(z-1)(z+1)}{(z - z_{p,1})(z - z_{p,2})}$$

$$= \frac{z^2 - 1}{z^2 - 1.1958z + 0.7995} = \frac{1 - z^{-2}}{1 - 1.1958z^{-1} + 0.7995z^{-2}}$$

The transfer function is identical to that of Example 8.9 to within a constant factor. The pole–zero diagrams are also identical to those given in Figure 8.11.

Example 8.17

Starting with a suitable analog LPF, find the transfer function of a Chebyshev digital HPF in factored form to meet the following specifications:

passband edge frequency	15 kHz
attenuation at 18 kHz	>30 dB
passband ripple	1 dB
sampling frequency	48 kHz

Solution The prewarped critical frequencies are

$$\omega_p' = \tan\left(\frac{15\,000\pi}{48\,000}\right) = 1.4966$$

$$\omega_s' = \tan\left(\frac{18\,000\pi}{48\,000}\right) = 2.4142$$

From the passband specifications, $\varepsilon = 0.3493$. The order of a suitable LP Chebyshev filter is 5 (the nearest integer).

With $\alpha = 1/N \sinh^{-1}(1/\varepsilon) = 0.3548$, $\sinh(\alpha) = 0.3623$, and $\cosh(\alpha) = 1.0636$, the left-hand poles of the normalized lowpass Chebyshev filter are located at ($\omega_p' = 1$)

$$s_{l,1} = 0.3623\cos\left[\frac{(2+5-1)\pi}{10}\right] + 1.0636\sin\left[\frac{(2+5-1)\pi}{10}\right]\mathrm{j}$$
$$= -0.111\,96 + 1.0115\mathrm{j}$$

$$s_{l,2} = 0.3623\cos\left[\frac{(4+5-1)\pi}{10}\right] + 1.0636\sin\left[\frac{(4+5-1)\pi}{10}\right]\mathrm{j}$$
$$= -0.2931 + 0.6252\mathrm{j}$$

$$s_{l,3} = 0.3623\cos\left[\frac{(6+5-1)\pi}{10}\right] + 1.0636\sin\left[\frac{(6+5-1)\pi}{10}\right]\mathrm{j} = -0.3623$$

$$s_{l,4} = 0.3623\cos\left[\frac{(8+5-1)\pi}{10}\right] + 1.0636\sin\left[\frac{(8+5-1)\pi}{10}\right]\mathrm{j}$$
$$= -0.2931 - 0.6252\mathrm{j}$$

$$s_{l,5} = 0.3623\cos\left[\frac{(10+5-1)\pi}{10}\right] + 1.0636\sin\left[\frac{(10+5-1)\pi}{10}\right]\mathrm{j}$$
$$= -0.111\,96 - 1.0115\mathrm{j}$$

The reader should note the symmetry in the pole distribution and that $s_{l,1}$ and $s_{l,5}$ form a complex conjugate pair as do $s_{l,2}$ and $s_{l,4}$. Note also that each pole lies on the left-hand side of the s-plane, a necessary condition for stability.

Each prototype lowpass pole is transformed into the desired HP pole using Equation 8.36b:

$$s_{h,1} = -0.1618 - 1.4616\mathrm{j}$$

$$s_{h,2} = -0.920\,13 - 1.9625\mathrm{j}$$

$$s_{h,3} = -4.1306$$

$$s_{h,4} = s_{h,2}^* = -0.920\,13 + 1.9625\mathrm{j}$$

$$s_{h,5} = s_{h,1}^* = -0.1618 + 1.4616\mathrm{j}$$

Next, the poles are mapped from the s-plane into the z-plane using the BZT. From Equation 8.42 the z-plane poles after the BZT are given by (only the poles above the real axis are considered):

$$z_{h,1} = -0.3335 + 0.8386j$$

$$z_{h,2} = -0.4906 + 0.5207j$$

$$z_{h,3} = -0.6102$$

All the zeros, $z_{z,k}$, are located at $z = 1$. The coefficients of the second- and first-order filter sections can then be obtained from the poles and zeros as (Equations 8.45a and 8.45b)

$$b_{11} = -2 \qquad b_{12} = -2 \qquad b_{13} = -1$$

$$b_{21} = 1 \qquad b_{22} = 1 \qquad b_{23} = 0$$

$$a_{11} = 0.6670 \qquad a_{12} = 0.9812 \qquad a_{13} = 0.6102$$

$$a_{21} = 0.8145 \qquad a_{22} = 0.5118 \qquad a_{23} = 0$$

Finally, the transfer function is given by

$$H(z) = KH_1(z)H_2(z)H_3(z)$$

where

$$H_1(z) = \frac{1 - 2z^{-1} + z^{-2}}{1 + 0.6670z^{-1} + 0.8145z^{-2}}$$

$$H_2(z) = \frac{1 - 2z^{-1} + z^{-2}}{1 + 0.9812z^{-1} + 0.5118z^{-2}}$$

$$H_3(z) = \frac{1 - z^{-1}}{1 + 0.6102z^{-1}}$$

8.11 Using IIR filter design programs

Whatever the method we adopt, it is evident that the bilinear transform method involves considerable algebraic manipulation, with plenty of opportunity to make mistakes. Efficient computer programs now exist in the literature or commercially for calculating filter coefficients using the bilinear method, by merely specifying filter parameters of interest (IEEE, 1979; Gray and Markel, 1976; Parks and Burrus, 1987;

Jong, 1982; DeFatta *et al.*, 1988). Most of the computer programs in the literature are written in FORTRAN. There is a move away from such languages to more modern languages such as C or BASIC. A C language program for computing the filter coefficients using the BZT is on the CD in the companion handbook (see the Preface for details). The use of the program is illustrated in the following example. In Appendix 8B we illustrate how to use MATLAB to design a variety of IIR digital filters.

Example 8.18 *Illustrating the use of the IIR design program* Obtain the coefficients of an audio digital filter with Chebyshev characteristics meeting the following specifications:

passband	0–2.5 kHz
stopband edge	2820 kHz
passband ripple	0.47 dB
sampling frequency	10 kHz
filter order	4

Solution Using the program, the listing given below was obtained.

k	A_k	B_k
0	$1.000\,000 \times 10^{0}$	$1.934\,410 \times 10^{-1}$
1	$-2.516\,884 \times 10^{-1}$	$3.783\,311 \times 10^{-1}$
2	$1.054\,118 \times 10^{0}$	$5.241\,429 \times 10^{-1}$
3	$-2.406\,030 \times 10^{-1}$	$3.783\,311 \times 10^{-1}$
4	$1.985\,861 \times 10^{-1}$	$1.934\,410 \times 10^{-1}$

8.12 Choice of coefficient calculation methods for IIR filters

With the impulse invariant method, after digitizing the analog filter, the impulse response of the original analog filter is preserved, but not its magnitude–frequency response. Because of inherent aliasing, the method is inappropriate for highpass or bandstop filters. The bilinear z-transform method, on the other hand, yields very efficient filters and is well suited to the calculation of coefficients of frequency selective filters. It allows the design of digital filters with known classical characteristics such as Butterworth, Chebyshev and elliptic filters. Digital filters resulting from the bilinear transform method will, in general, preserve specific features of the magnitude response characteristics of the analog filter (e.g. bandedge frequencies, passband ripple and stopband attenuation) but not necessarily the time-domain properties. The impulse invariant method is good for simulating analog systems with lowpass characteristics, but the bilinear method is best for frequency selective IIR filters. The matched z-transform shares most of the inherent problems of the impulse

invariant method. For simple filtering applications, the pole–zero placement method provides a simple, but effective, way of obtaining the filter coefficients.

8.12.1 Nyquist effect

The three methods for converting analog filters into equivalent discrete-time filters (namely, the matched z-transform, the impulse invariant and the bilinear z-transform methods) can have a significant effect on the filter characteristics (e.g. magnitude, phase and group delay responses), in certain cases. As stated earlier, the available frequency band for analog filters extends from zero to infinity, whereas for digital filters it is from zero to the Nyquist frequency (i.e. half the sampling frequency). Thus, the magnitude–frequency response of digital filters designed using either of the three methods may be significantly different from those of the analog filter because the entire analog frequency band (zero to infinity) is now compressed into a narrow band (zero to Nyquist frequency). This difference represents a distortion which is sometimes referred to as the Nyquist effect.

In many applications, the Nyquist effect is not harmful besides the provision of greater attenuation than specified. However, in applications where it is desirable to retain the analog filter response, such as in professional and semi-professional audio work (Clark *et al.*, 1996; 2000), the effect represents an undesirable distortion. In such applications, the extent of the distortion would be a factor in the choice of a transform for converting an analog filter into an equivalent discrete-time filter. The impact on other filter characteristics, such as group delay and impulse responses, may also be a factor in the choice of a method (Rabiner and Gold, 1975).

In this section, we will consider the consequences of the Nyquist effect briefly. A number of problems are provided at the end of the chapter to allow the reader to explore the relative advantages of the methods of converting analog filters into equivalent discrete filters.

Example 8.19

A lowpass, discrete-time filter, with Butterworth characteristics, is required to meet the following specifications:

cutoff frequency	300 Hz
filter order	5
sampling frequency	1000 Hz

(1) With the aid of MATLAB, obtain and plot
   (a) the magnitude–frequency and group delay responses of the filter using the impulse invariant method;
   (b) the magnitude–frequency and group delay responses of the filter using the bilinear z-transform method.

(2) Compare the two methods (bilinear z-transform and impulse invariant methods) in terms of the magnitude response distortions due to the Nyquist effect.

Solution (1) (a) We will start by designing an analog filter to serve as a benchmark. The MATLAB m-file for the analog filter is listed in Program 8.1. The magnitude–frequency response of the filter, obtained with the aid of the program, is shown in Figure 8.18.

The magnitude–frequency and group delay responses for the equivalent discrete filter, designed via the impulse invariant method, are shown in Figures 8.19(a) and (b). The MATLAB m-file for the filter is shown in Program 8.2. The group delay response was obtained using the **grpdelay** command in the MATLAB Signal Processing Toolbox.

Program 8.1 MATLAB m-file for the analog filter design.

```
%
%  Program name: EX8-1.m
%
FN=1000/2;
fc=300;                          %  Cutoff frequency
N=5;                             %  Filter order
[z, p, k]=buttap(N);             %  Create an analog filter
w=linspace(0, FN/fc, 1000);      %  Plot the response of filter
h=freqs(k*poly(z), poly(p), w);
f=fc*w;
plot(f, 20*log10(abs(h))), grid
ylabel('Magnitude (dB)')
xlabel('Frequency (Hz)')
```

Program 8.2 MATLAB m-file for the design of the impulse invariant filter.

```
%
%  Program name: EX8-2.m
%
Fs=1000;                         %  Sampling frequency
fc=300;                          %  Cutoff frequency
WC=2*pi*fc;                      %  Cutoff frequency in radian
N=5;
[b,a]=butter(N, WC, 's');        %  Create an analog filter
[z,p,k]=butter(N, WC, 's');
[bz, az]=impinvar(b, a, Fs);     %  Determine coeffs of IIR filter
[h, f]=freqz(bz, az, 512, Fs);
plot(f, 20*log10(abs(h))), grid
xlabel('Frequency (Hz)')
ylabel('Magnitude Response (dB)')
```

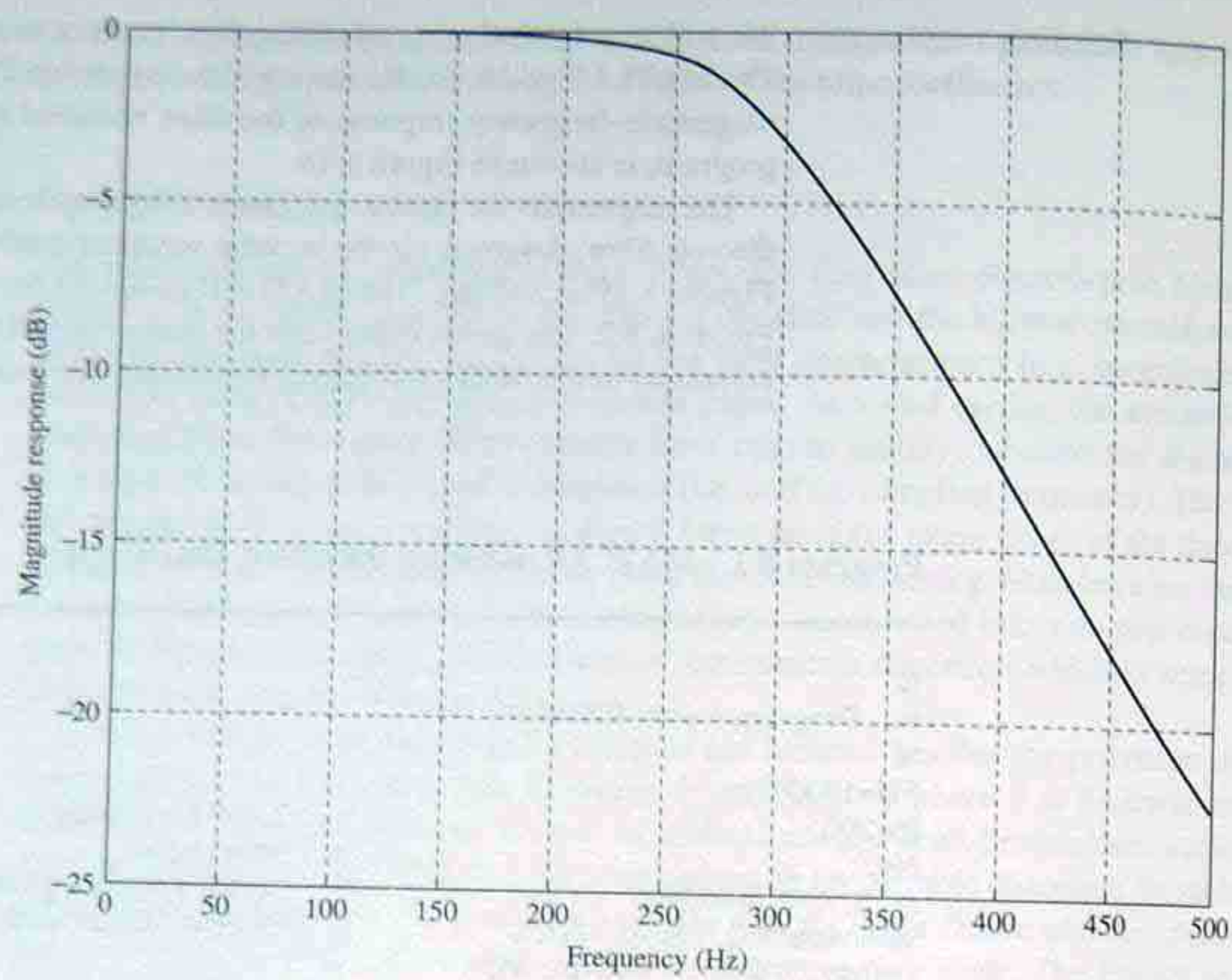


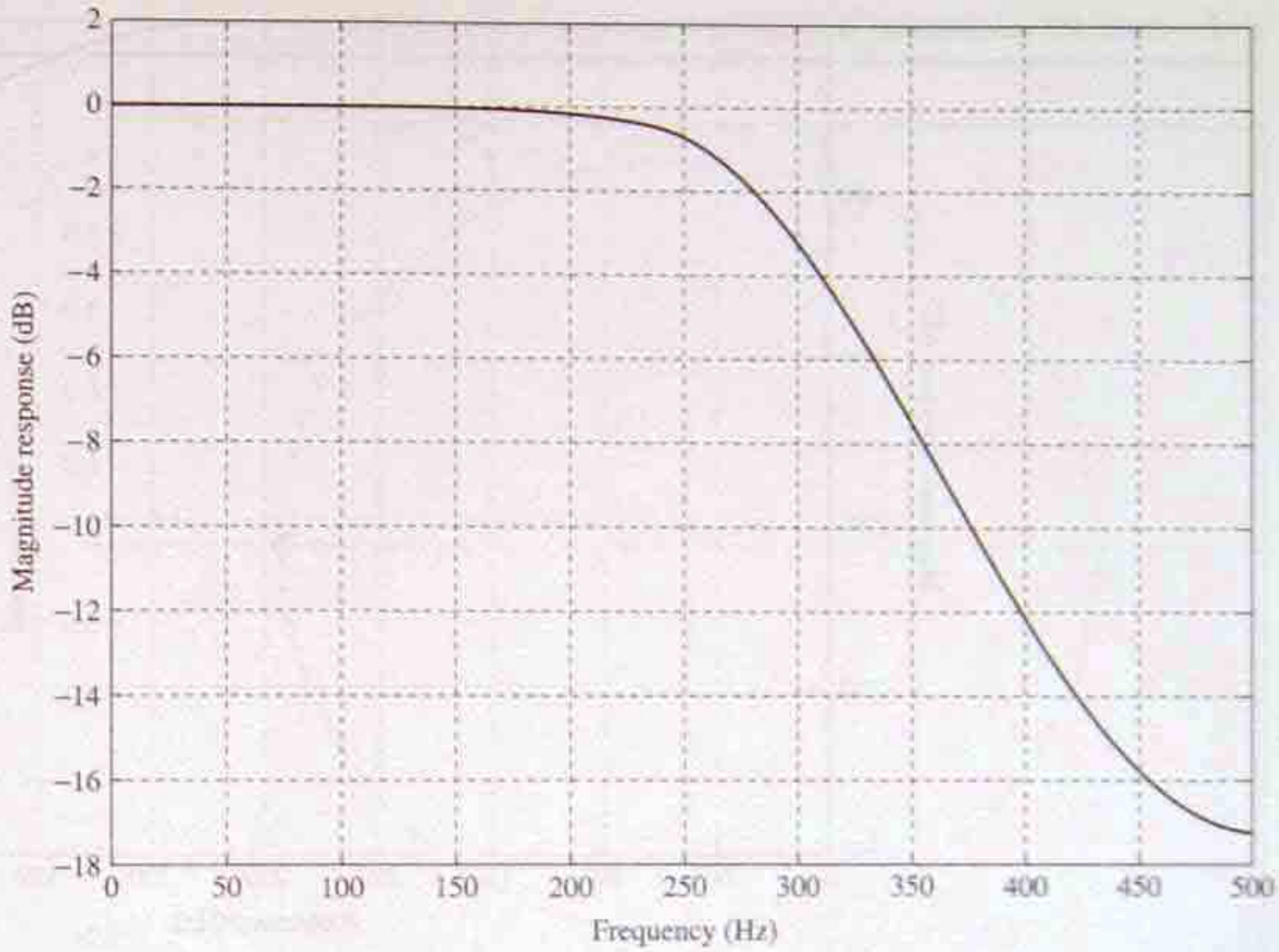
Figure 8.18 Analog filter magnitude response.

Program 8.3 MATLAB m-file for the design of the BZT filter.

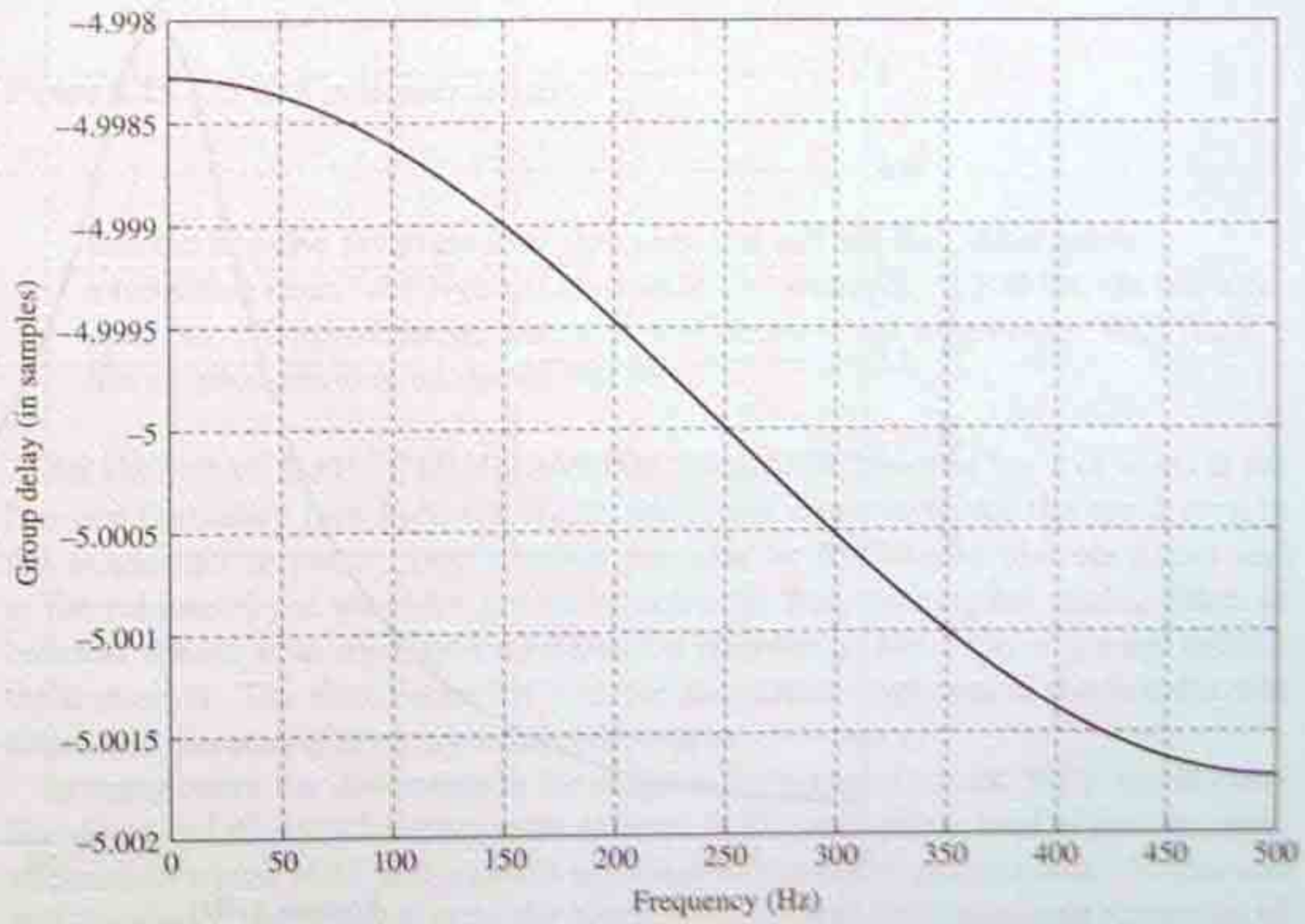
```
%
%  Program name: EX8-3.m
%
Fs=1000;                                  %  Sampling frequency
FN=Fs/2;
fc=300;                                   %  Cutoff frequency
N=5;
[z, p, k]=butter(N, fc/FN);
[h, f]=freqz(k*poly(z), poly(p), 512, Fs);
plot(f, 20*log10(abs(h))), grid
ylabel('Magnitude (dB)')
xlabel('Frequency (Hz)')
```

(b) The magnitude–frequency and group delay responses for the equivalent discrete filter, designed via the BZT method, are shown in Figures 8.20(a) and (b). The MATLAB m-file for the filter is shown in Program 8.3.

(2) If we compare the magnitude–frequency responses of the impulse invariant filter and the bilinear z-transform filter to that of the analog filter, it is evident



(a)



(b)

Figure 8.19 (a) Impulse invariant method. (b) Group delay response for the impulse invariant method.

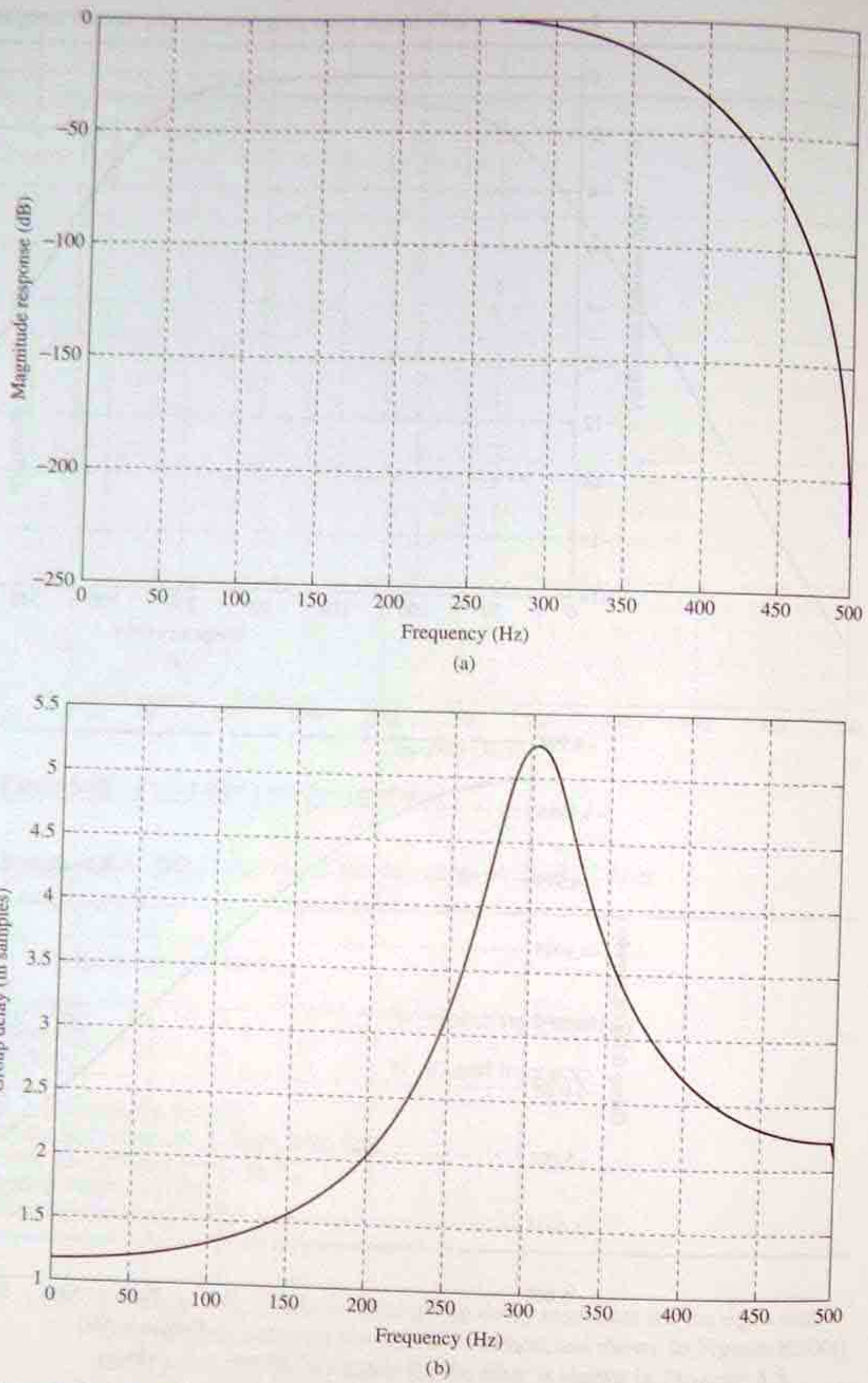


Figure 8.20 (a) Magnitude–frequency response for the BZT. (b) Group delay response for the BZT example.

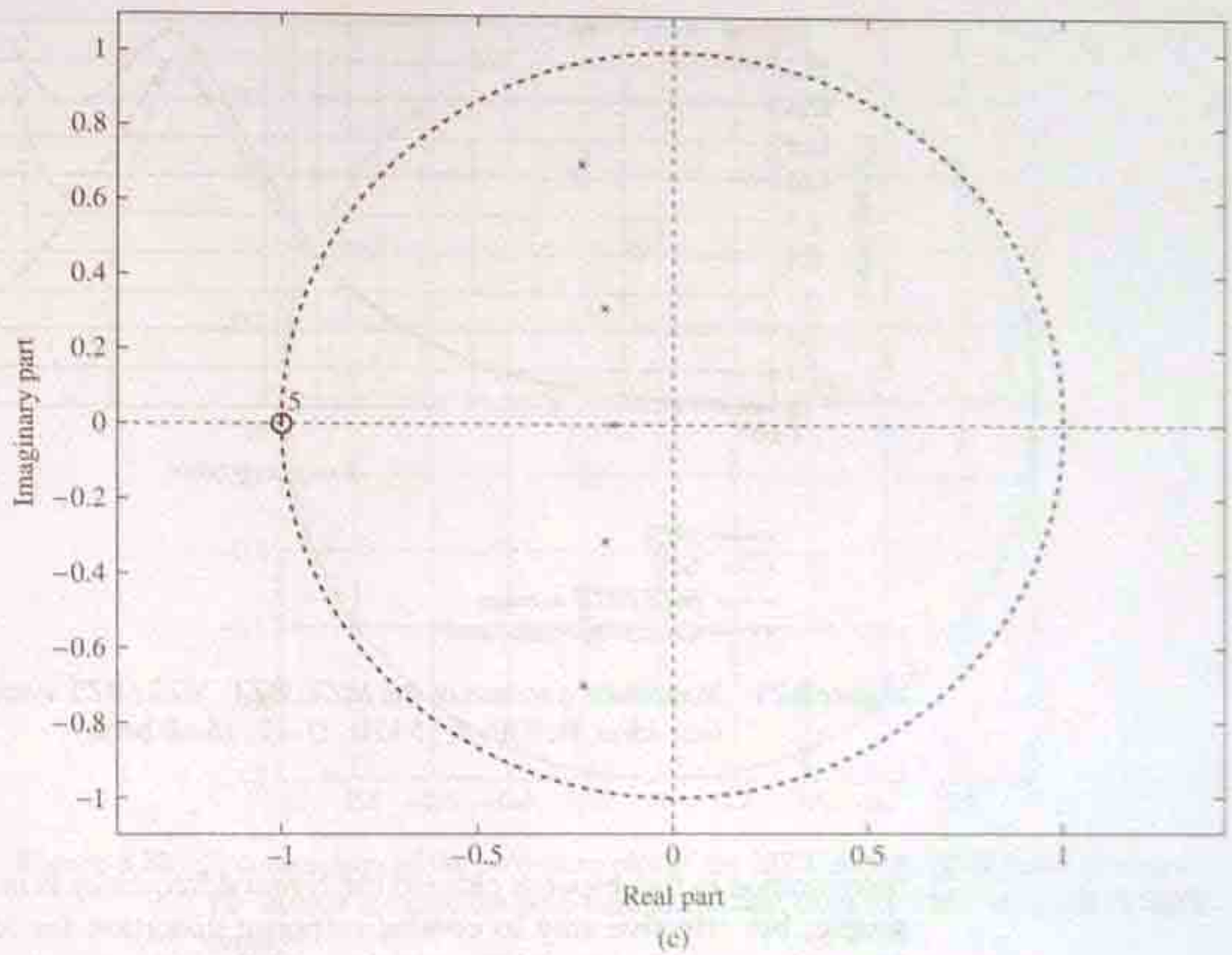


Figure 8.20 (c) BZT pole–zero diagram.

that the impulse invariant filter provides less and the BZT filter more attenuation nearer the Nyquist frequency. For example, at 500 Hz, the impulse invariant filter provides an attenuation of about 17 dB whereas the BZT filter has an attenuation in excess of 200 dB.

An analysis of the BZT filter shows that the transfer function has five zeros at the Nyquist frequency (see Figure 8.20(c)) which are responsible for the rapid drop in the magnitude response. This is often the case in BZT-based discrete filters and is the reason why it provides greater attenuation than the original analog filter. In contrast, the impulse invariant-based discrete filter has a zero at the origin and outside the unit circle. The differences between the magnitude responses of the two discrete filters and the analog filter represent a distortion.

In many cases, the distortions in the magnitude responses for the MZT (or impulse invariant) and the BZT have opposite effects. BZT-based filters tend to provide more attenuation whilst MZT and impulse invariant-based filters provide less. At low and mid-frequency bands (relative to the Nyquist frequency), the magnitude responses of lowpass and bandpass filters designed by the MZT and impulse invariant methods are reasonably close to those of the original analog filters, but deteriorate near the Nyquist frequency. The same is true of filters designed by the BZT method, except that the

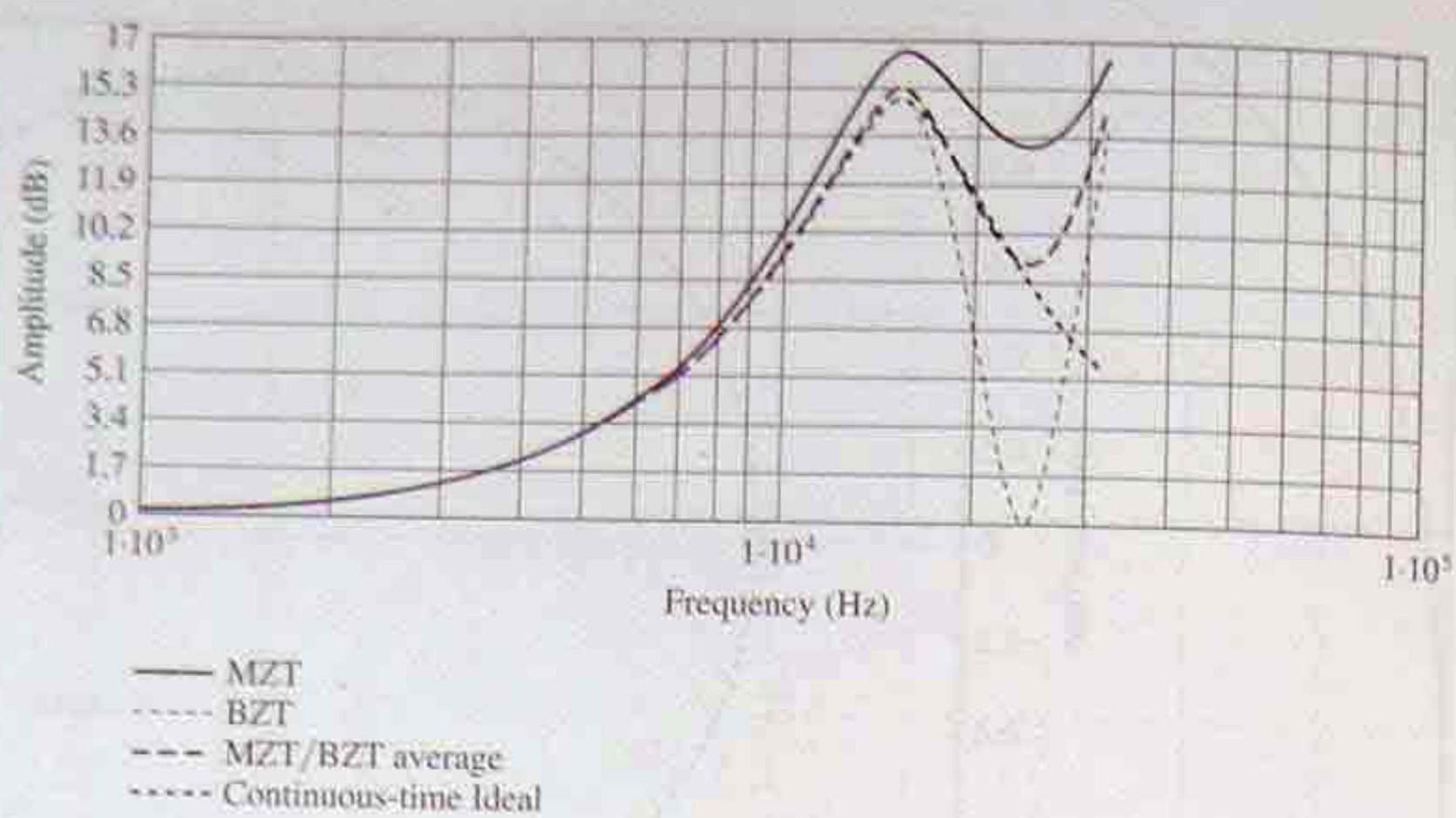


Figure 8.21 Magnitude response of the MZT, BZT, MZT/BZT average and the continuous-time ideal, Bell filter, 15 kHz, $Q = 2$, 15 dB boost.

deterioration in the response close to the Nyquist frequency is in the opposite sense. A simple, but effective way to combat response distortion for lowpass and bandpass filters is to combine the BZT with either MZT or the impulse invariant method by averaging their coefficients (Clark *et al.*, 1996; 2000). In this approach, the lowpass or bandpass filter is first designed separately by the BZT and MZT. The coefficients are then averaged to obtain a new set of coefficients as follows:

$$b_k' = [b_k(BZT) + b_k(MZT)]/2,\ k = 0, 1, 2$$

$$a_k' = [a_k(BZT) + a_k(MZT)]/2,\ k = 0, 1, 2$$

Figure 8.21 compares the responses of two discrete filters, one designed by the MZT method and the other by the BZT method. The filters were intended for an audio signal processing application where there is a need for efficient means of generating filter coefficients online at low response distortion. As before, the BZT-based filter gives a sharper magnitude–frequency response beyond 10 kHz (the Nyquist frequency is 24 kHz) whereas the MZT gives less attenuation. Both deviations in the magnitude response represent a distortion in this application. The averaged response (Figure 8.21) on the other hand has a reduced response distortion. The pole–zero diagram of Figure 8.22 compares the locations of the poles and zeros for the MZT and the averaged MZT/BZT filters. As would be expected, the averaging process has altered the positions of the poles and zeros of the MZT filter. In particular, it has introduced a 'soft' zero at the Nyquist frequency. The zero is said to be soft because it is located well inside the unit circle and so does not pull the magnitude response of the filter rapidly towards zero as in the BZT.

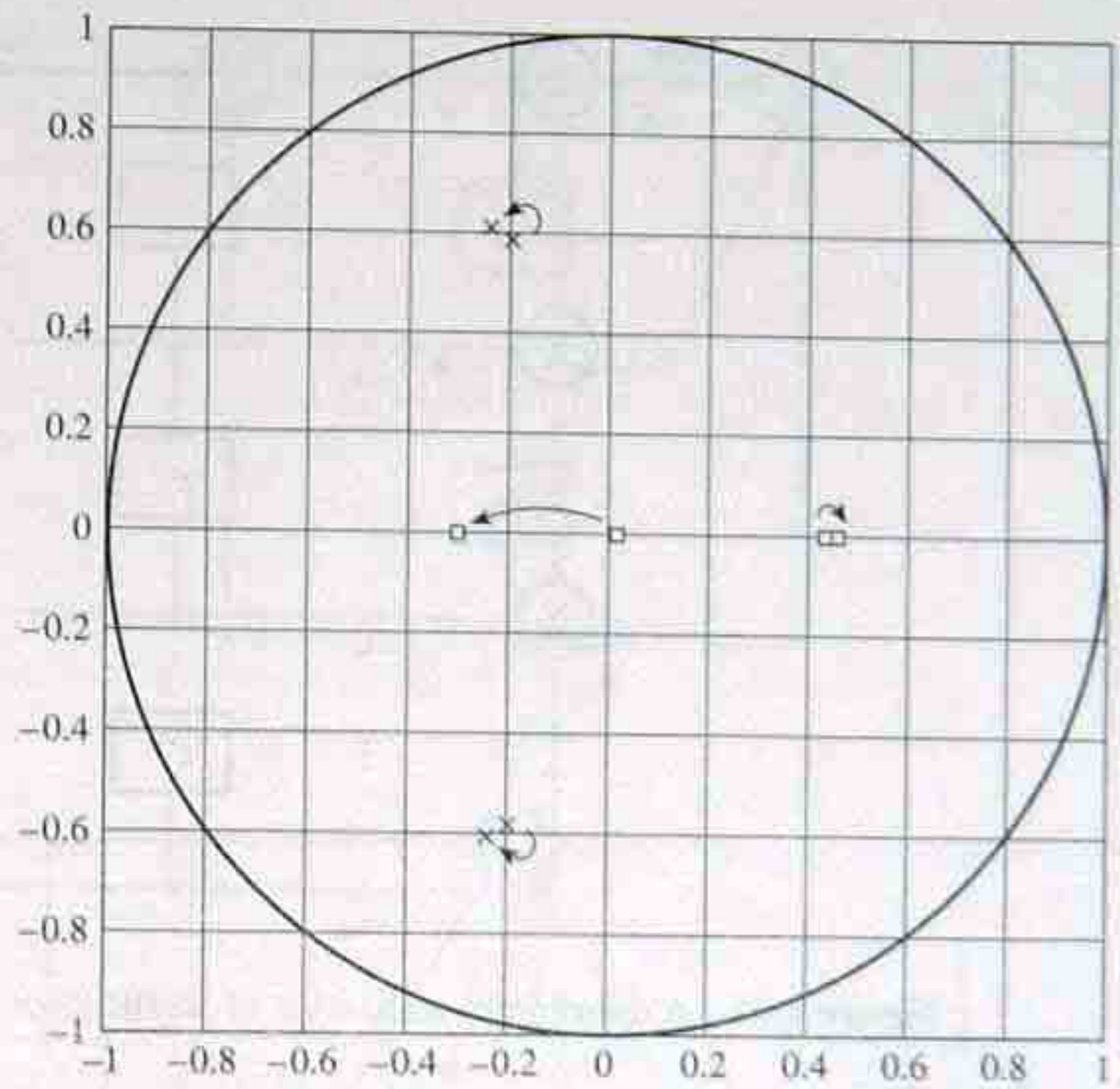


Figure 8.22 A comparison of the pole–zero plot of the MZT and the MZT/BZT average. The arrows indicate the pole–zero movement from the MZT to the MZT/BZT average.

8.13 Realization structures for IIR digital filters

Realization involves converting a given transfer function, $H(z)$, into a suitable filter structure. Flow or block diagrams are normally used to depict filter structures and they show the computational procedure for implementing the digital filter. The basic elements of realization structures are multipliers, adders, and delay elements; see Figure 8.23.

Recall that the IIR filter is characterized by the following equations:

$$H(z) = \sum_{k=0}^{N} b_k z^{-k} \Big/ \left(1 + \sum_{k=1}^{M} a_k z^{-k}\right) \quad M \geqslant N \tag{8.46a}$$

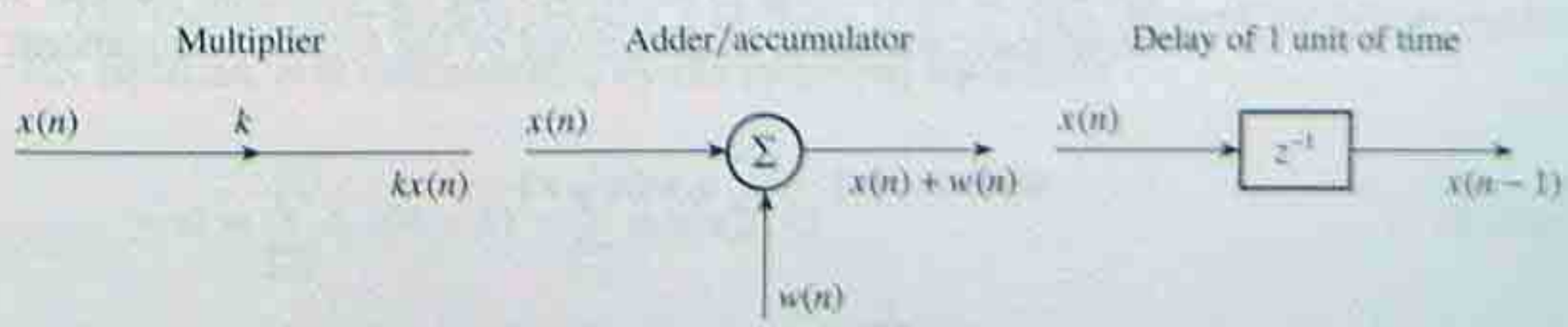


Figure 8.23 The basic elements of filter structures.

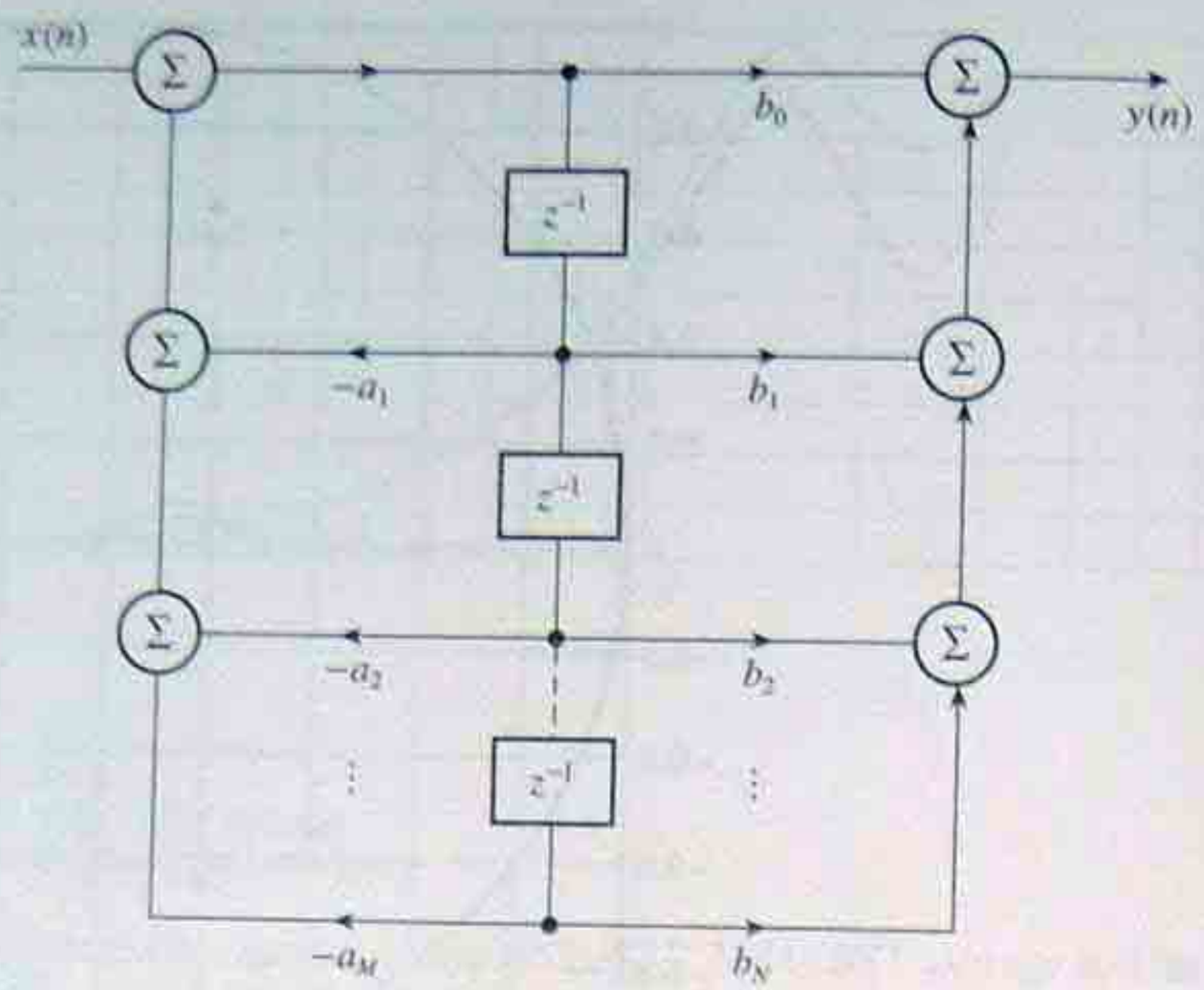


Figure 8.24 A direct form realization of an IIR filter.

$$y(n) = \sum_{k=0}^{N-1} b_k x(n-k) - \sum_{k=1}^{M} a_k y(n-k) \tag{8.46b}$$

A direct form realization of Equation 7.46 is shown in Figure 8.24, where $N = M$ for simplicity. Note that the coefficients used in the diagram are the same as in the transfer function, but with the signs reversed for the denominator coefficients. When the filter order is high, for example $M > 3$, direct realization of the filter as in Figure 8.24 is very sensitive to finite wordlength effects and should be avoided in these cases. In practice, $H(z)$ is normally broken down into smaller sections, typically second- and/or first-order blocks, which are then connected up in cascade or in parallel (see later).

8.13.1 Practical building blocks for IIR filters

Examples of practical second-order building blocks used in realizing higher-order IIR filters are depicted in Figure 8.25. The first (Figure 8.25(a)) is often called a canonic section (or direct form 2) because it has the minimum number of delay elements. This biquadratic section is characterized by the following equations:

$$w(n) = x(n) - \sum_{k=1}^{2} a_k w(n-k) \tag{8.47a}$$

$$y(n) = \sum_{k=0}^{2} b_k w(n-k) \tag{8.47b}$$

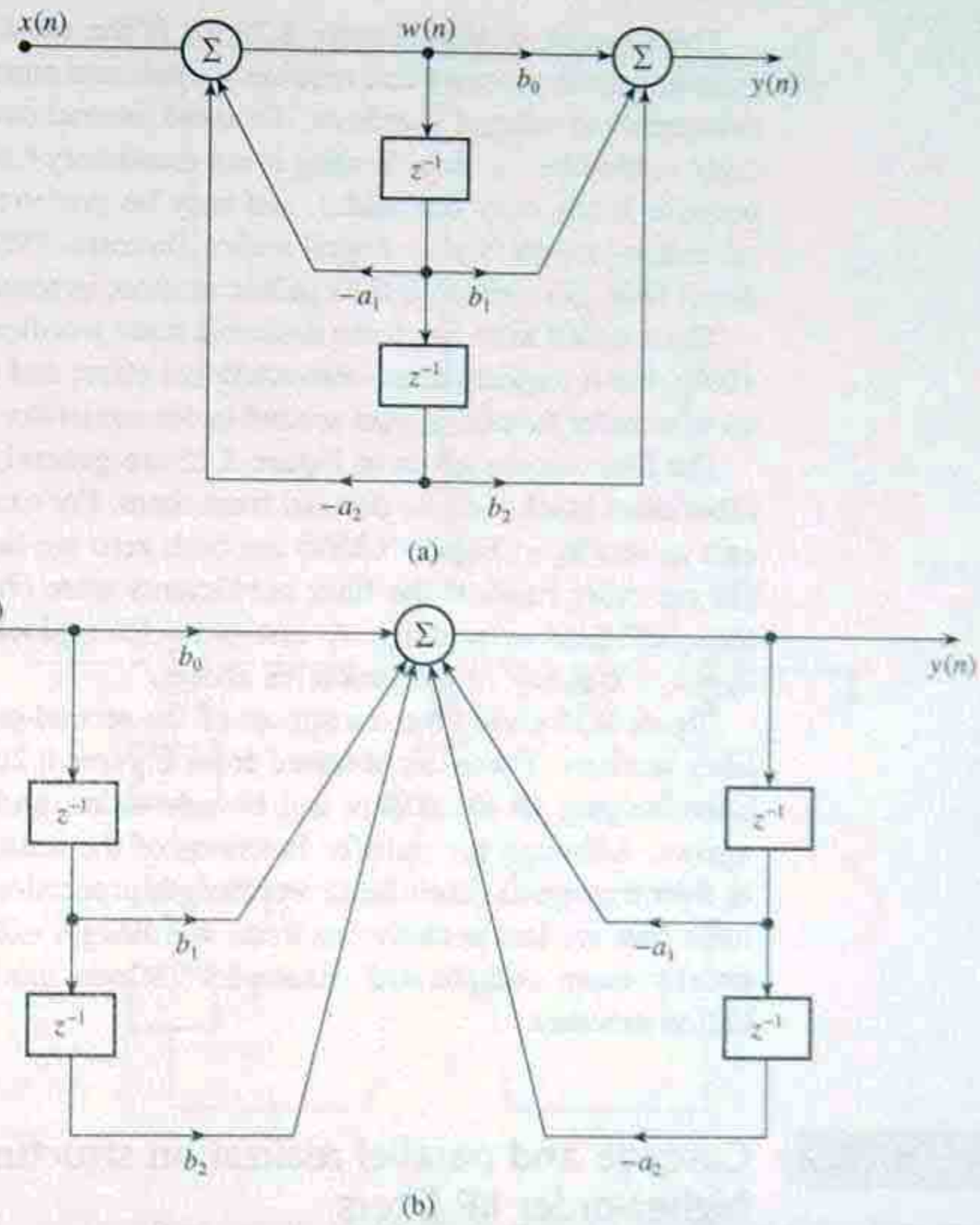


Figure 8.25 Practical building blocks for IIR realization structures: (a) canonic second-order section; (b) direct form second-order section.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{8.47c}$$

The second filter section (Figure 7.18(b)) is a direct realization of the second-order IIR equation. It is characterized by the following equations:

$$y(n) = \sum_{k=0}^{2} b_k x(n-k) - \sum_{k=1}^{2} a_k y(n-k) \tag{8.48a}$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{8.48b}$$

The canonic section (Figure 8.25(a)) is the most popular because it has a good roundoff noise property and requires a minimum number of storage elements, but it is susceptible to internal overflows. To avoid internal overflow it is necessary to scale the input to the filter section. Scaling is not mandatory for the direct form (Figure 8.25(b)) because it has only one adder, and may be preferred where scaling is not desired, for example high fidelity digital audio (Dattorro, 1988). Under certain conditions the direct form is superior to the canonic section, in terms of noise performance.

The coupled form has some desirable finite wordlength properties (Gold and Rader, 1969), but it requires more computational effort and cannot be readily used to implement transfer functions with second-order numerator coefficient.

The filter blocks given in Figure 8.25 are general, second-order, sections. Several other filter blocks can be derived from them. For example, if the numerator coefficients a_1 and a_2 in Figure 8.25(a) are both zero we have a purely recursive structure. On the other hand, if the filter coefficients were obtained using elliptical functions the coefficient a_2 is unity. A first-order filter block is readily obtained by setting $a_2 = b_2 = 0$ in any of the structures above.

Figure 8.26 shows the transposes of the second-order canonic and the direct form filter sections. These are obtained from Figures 8.26(a) and 8.26(b) respectively, by interchanging all the adders and branch nodes, and reversing the directions of the arrows. Although the transfer functions of the sections in Figure 8.26 are identical to their transposes, their finite wordlength properties are quite different. Other structures that are less sensitive to finite wordlength effects are available, but these are usually more complicated. Examples include minimum-noise, state-variable, and lattice structure.

8.13.2 Cascade and parallel realization structures for higher-order IIR filters

In practice higher-order transfer functions are realized as cascades or parallel combinations of second- and/or first-order building blocks described above. Typically, in cascade realization the transfer function is factored into $N/2$ second-order factors:

$$H(z) = \prod_{k=1}^{N/2} \left[\frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}} \right]$$

$$= \prod_{k=1}^{N/2} \frac{N_k(z)}{D_k(z)} \tag{8.49a}$$

where

$$N_k(z) = b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}$$
$$D_k(z) = 1 + a_{1k}z^{-1} + a_{2k}z^{-2} \tag{8.49b}$$

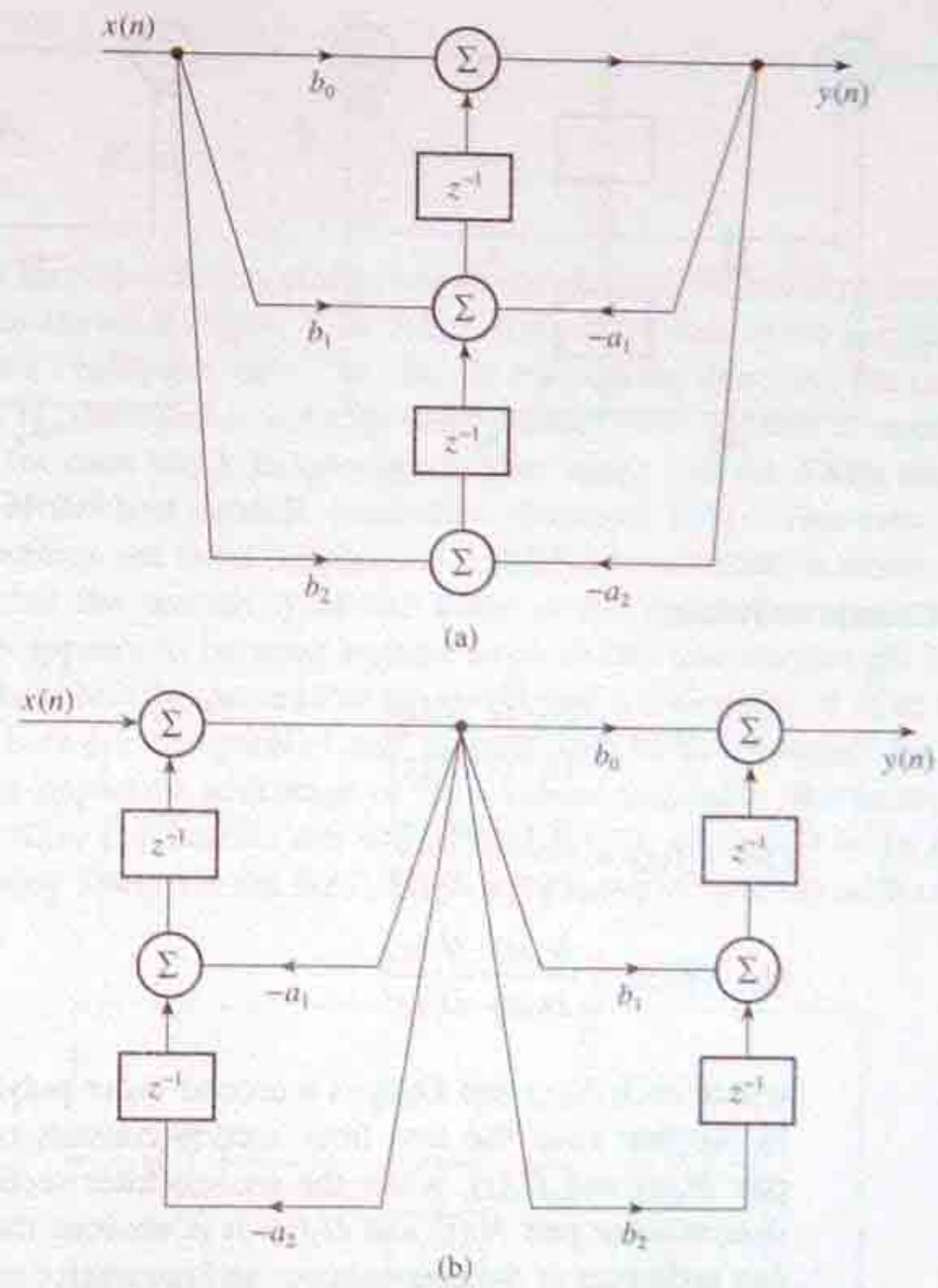


Figure 8.26 (a) Transpose canonic second-order section; (b) transpose direct form second-order section.

and N, the filter order, is assumed to be even. If N is odd, then one of the $H_k(z)$ will be a first-order section.

Each second-order factor, $H_k(z)$, can be realized using one of the building blocks and the resulting blocks connected in cascade; see Figure 8.27. Three difficulties arise with the cascade realization: (1) how to pair the numerator factors with the denominator factors; (2) the order in which the individual sections should be connected; (3) the need to scale the signal levels at various points within the filter to avoid the levels becoming too large or too small.

The numerator and denominator factors can be ordered in a variety of ways. For example, a fourth-order filter can be factored into two second-order sections, and then paired and ordered in one of four different ways:

$$(1) \quad H(z) = \frac{N_1(z)}{D_1(z)} \frac{N_2(z)}{D_2(z)}$$

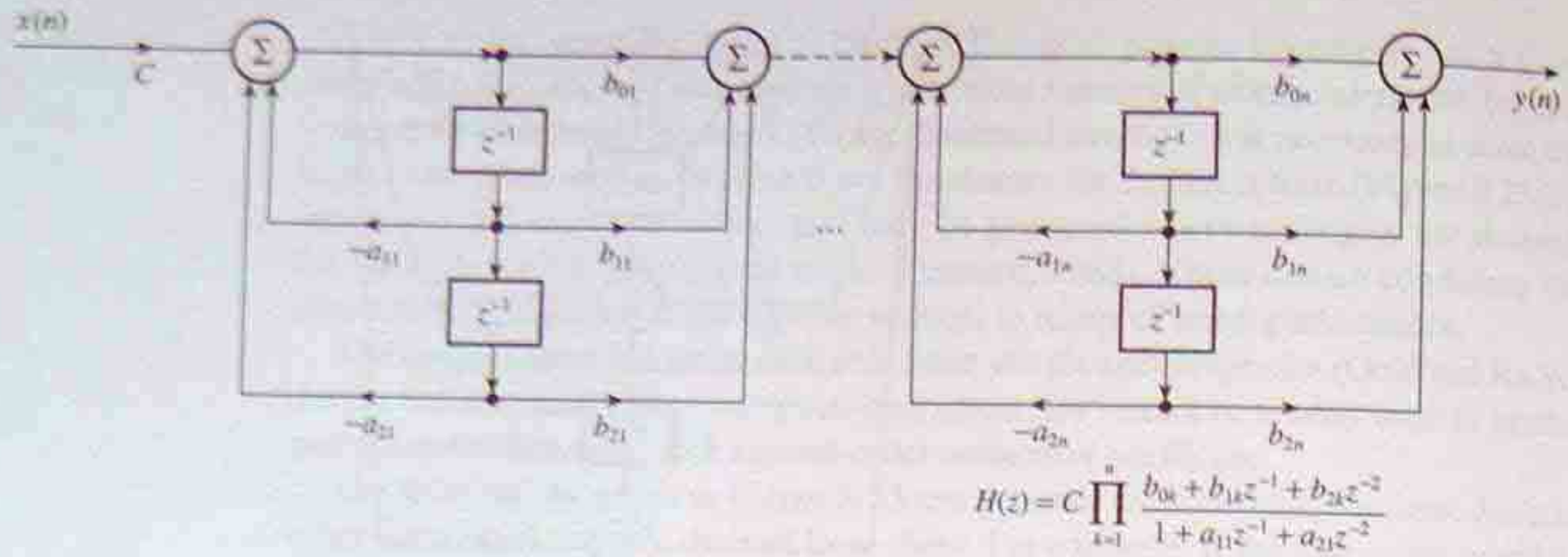


Figure 8.27 Cascade realization.

(2) $H(z) = \dfrac{N_2(z)}{D_2(z)}\dfrac{N_1(z)}{D_1(z)}$

(3) $H(z) = \dfrac{N_1(z)}{D_2(z)}\dfrac{N_2(z)}{D_1(z)}$

(4) $H(z) = \dfrac{N_2(z)}{D_1(z)}\dfrac{N_1(z)}{D_2(z)}$

where each $N_k(z)$ and $D_k(z)$ is a second-order polynomial defined in Equation 8.49b. In the first case, the first filter section consists of the numerator and denominator pair $N_1(z)$ and $D_1(z)$, while the second filter section consists of the numerator and denominator pair $N_2(z)$ and $D_2(z)$. It is obvious that the number of possible pairings and orderings of the denominator and numerator is quite large. Typically, for an Nth-order filter, the number of different pairings and orderings possible is

$$\left(\frac{N}{2}!\right)^2 \tag{8.50}$$

A rule of thumb is to pair $N_i(z)$ with $D_k(z)$ if the zeros of $N_i(z)$ are closest to the poles of $D_k(z)$ to avoid having a large amplitude response at the frequency corresponding to the pole, and to place the second-order section with poles closest to the unit circle last in the cascade (Jackson, 1986). A number of efficient schemes have been developed for pairing and ordering the filter sections, based on which ordering gives the best signal-to-noise ratio, a topic intimately linked with pairing and ordering (see Chapter 13).

In parallel realization, an Nth-order transfer function $H(z)$ is expanded using partial fractions as

$$H(z) = C + \sum_{k=1}^{N/2} H_k(z) \tag{8.51}$$

where

$$C = \frac{b_N}{a_N}, \quad H_k(z) = \frac{b_{0k} + b_{1k}z^{-1}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}$$

Again, each second-order section can be realized using the building blocks described previously as shown in Figure 8.28. It is worth noting that, in the parallel realization, the numerator coefficient for z^{-2} is zero. In the parallel structure, the order in which the sections is connected is not important. Furthermore, scaling is easier and can be carried out for each block independently (see later), and the SNRs are comparable with those of the best cascade realization (Jackson, 1986). However, the zeros of parallel structures are more sensitive to coefficient quantization errors. It should be mentioned that the sensitivity of the zeros of the parallel structure to coefficient quantization appears to be most serious when coefficient wordlength is down to as little as 5 bits or less. It appears that for coefficient wordlengths of 12 or more bits the differences between the parallel and cascade structures are small for most filters. However, an important advantage of the cascade method is that between 25% and 50% of the filter coefficients are simple integers (0, ±1 or ±2) when derived from classical analog filters via the BZT. This is attractive in systems with only primitive

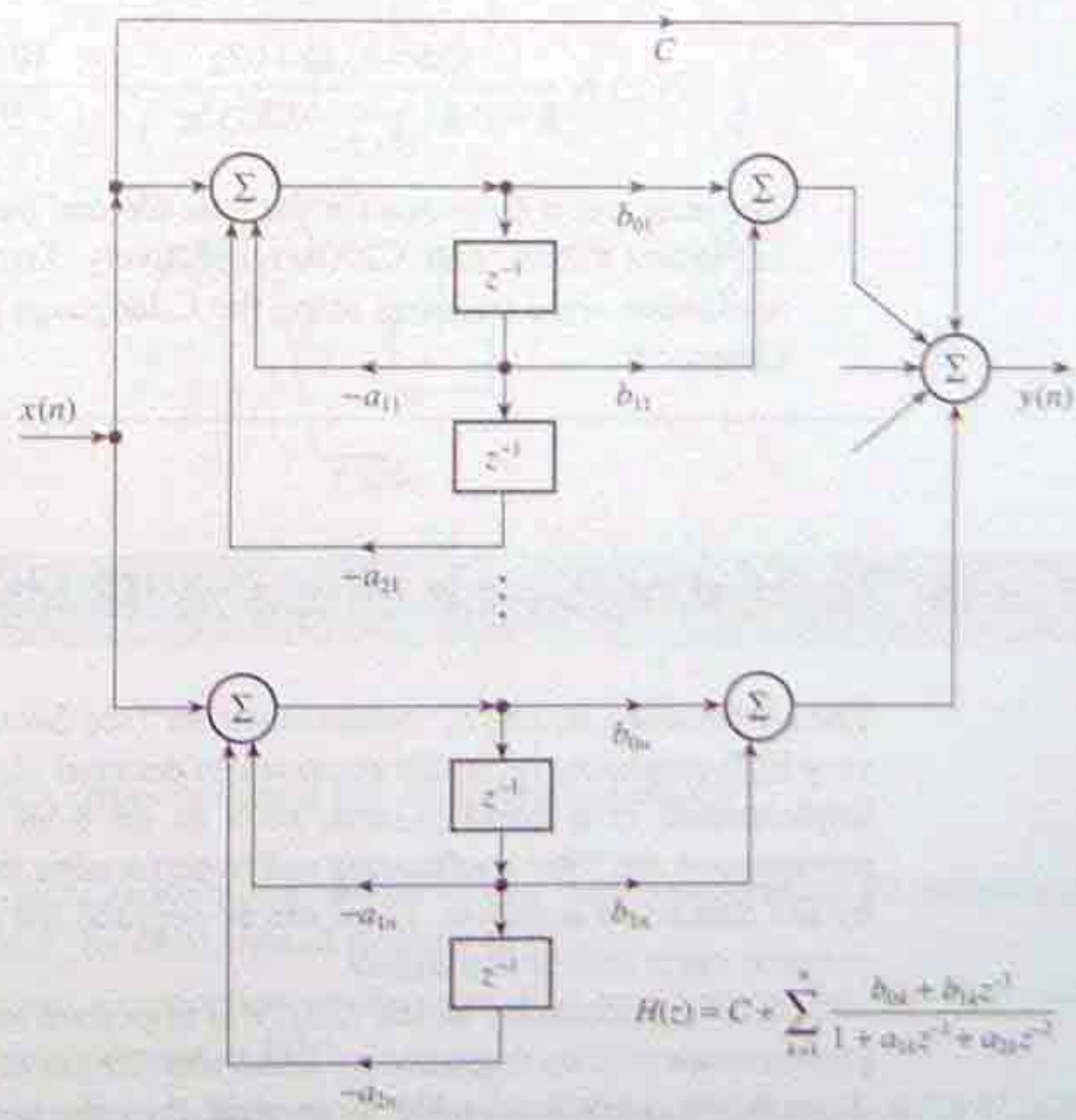


Figure 8.28 Parallel realization.

arithmetic capability where the number of multiplications must be kept low. Further, most available software packages produce coefficients for cascade realization but not for the parallel structure. It is for these reasons that the cascade method has become very popular.

Example 8.20 Develop the transfer functions for (1) the cascade and (2) the parallel realization structures for the filter characterized by the following transfer function using second- and first-order sections:

$$H(z) = \frac{0.1432(1 + 3z^{-1} + 3z^{-2} + z^{-3})}{1 - 0.1801z^{-1} + 0.3419z^{-2} - 0.0165z^{-3}}$$

Solution (1) For cascade realization, $H(z)$ is expressed in factored form:

$$H(z) = 0.1432\,\frac{1 + 2z^{-1} + z^{-2}}{1 - 0.1307z^{-1} + 0.3355z^{-2}}\,\frac{1 + z^{-1}}{1 - 0.0490z^{-1}}$$

(2) For parallel realization $H(z)$ is expressed as the sum of second- and first-order sections using partial fractions:

$$H(z) = \frac{1.2916 - 0.08407z^{-1}}{1 - 0.131z^{-1} + 0.3355z^{-2}} + \frac{10.1764}{1 - 0.049z^{-1}} - 8.7107$$

The realization diagrams for the cascade and parallel realizations are given in Figures 8.29(a) and 8.29(b) respectively. The coefficients for the parallel realization were obtained using the C language program presented in Chapter 4.

8.14 Finite wordlength effects in IIR filters

The coefficients, a_k and b_k, obtained earlier (see Sections 8.4–8.10) are of infinite or very high precision, typically six to seven decimal places. When an IIR digital filter is implemented in a small system, such as an 8-bit microcomputer, errors arise in representing the filter coefficients and in performing the arithmetic operations indicated by the difference equation. These errors degrade the performance of the filter and in extreme cases lead to instability.

Before implementing an IIR filter, it is important to ascertain the extent to which its performance will be degraded by finite wordlength effects and to find a remedy if the degradation is not acceptable. In general, the effects of these errors can be reduced to acceptable levels by using more bits but this may be at the expense of increased cost.

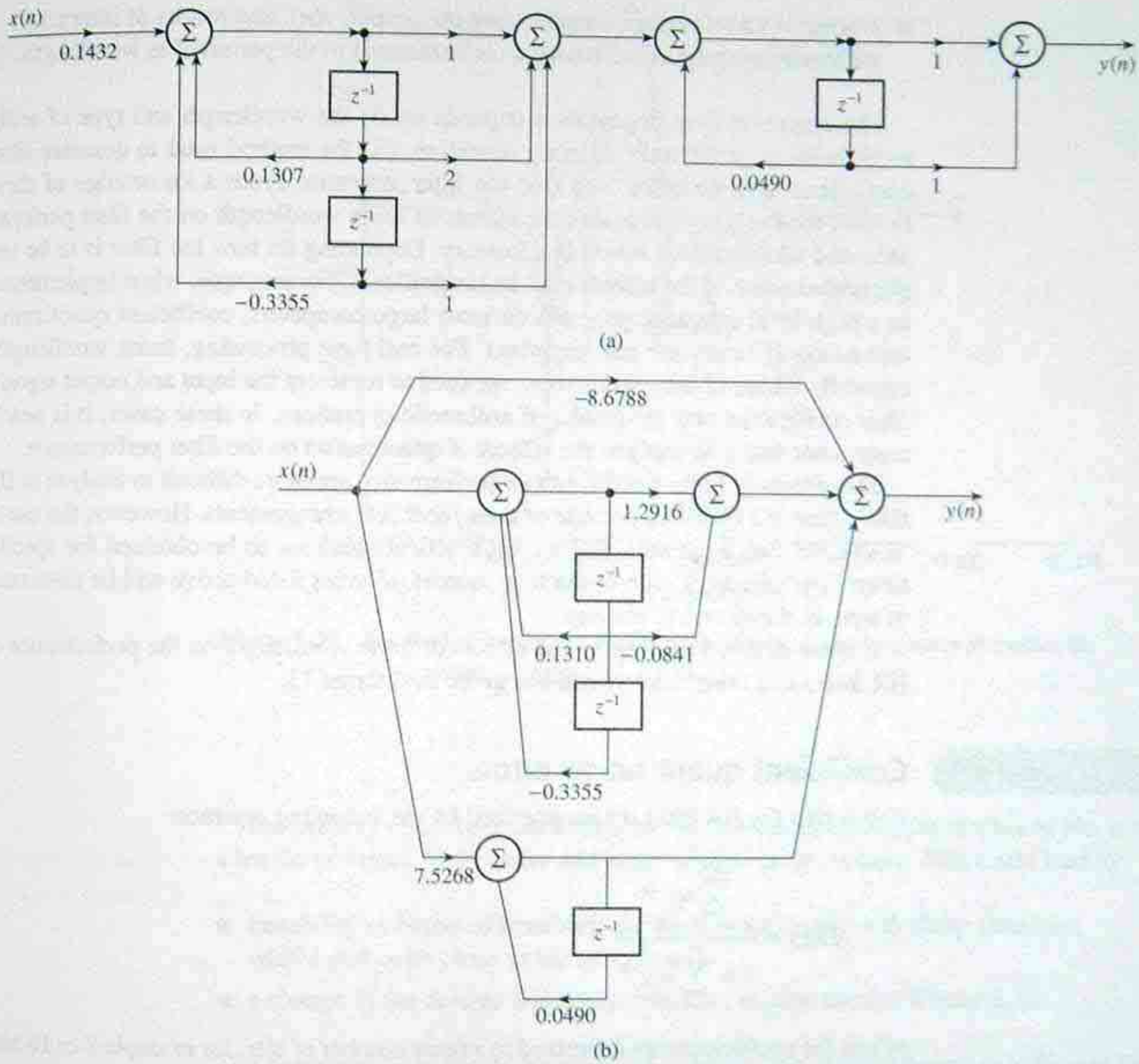


Figure 8.29 (a) Cascade and (b) parallel realizations for Example 8.20.

The main errors in digital IIR filters are as follows:

- ADC quantization noise, which results from representing the samples of the input data, $x(n)$, by only a small number of bits;
- coefficient quantization errors, caused by representing the IIR filter coefficients by a finite number of bits;
- overflow errors, which result from the additions or accumulation of partial results in a limited register length;

- product roundoff errors, caused when the output, $y(n)$, and results of internal arithmetic operations are rounded (or truncated) to the permissible wordlength.

The extent of filter degradation depends on (i) the wordlength and type of arithmetic used to perform the filtering operation, (ii) the method used to quantize filter coefficients and variables, and (iii) the filter structure. From a knowledge of these factors, the designer can assess the effects of finite wordlength on the filter performance and take remedial action if necessary. Depending on how the filter is to be implemented some of the effects may be insignificant. For example, when implemented as a high level language program on most large computers, coefficient quantization and roundoff errors are not important. For real-time processing, finite wordlengths (typically 8 bits, 12 bits, and 16 bits) are used to represent the input and output signals, filter coefficients and the results of arithmetic operations. In these cases, it is nearly always necessary to analyze the effects of quantization on the filter performance.

The effects of finite wordlength on performance are more difficult to analyze in IIR filters than in FIR filters because of their feedback arrangements. However, the use of MATLAB (see Appendix 8B) allows practical solutions to be obtained for specific filters. The effects of each of the four sources of errors listed above will be discussed, in turn, in the next few sections.

A more detailed analysis of the effects of finite wordlength on the performance of IIR filters and other DSP systems is given in Chapter 13.

8.14.1 Coefficient quantization errors

Recall that the IIR filter is characterized by the following equation:

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}}$$

When the coefficients are quantized to a finite number of bits, for example 8 or 16 bits, the quantized transfer function may be written as

$$[H(z)]_q = \frac{\sum_{k=0}^{N} [b_k]_q z^{-k}}{1 + \sum_{k=1}^{M} [a_k]_q z^{-k}} \tag{8.52}$$

where

$$[b_k]_q = b_k + \Delta b_k; \qquad [a_k]_q = a_k + \Delta a_k$$

Δb_k, Δa_k are changes in the coefficients b_k and a_k, respectively
q denotes a quantized coefficient

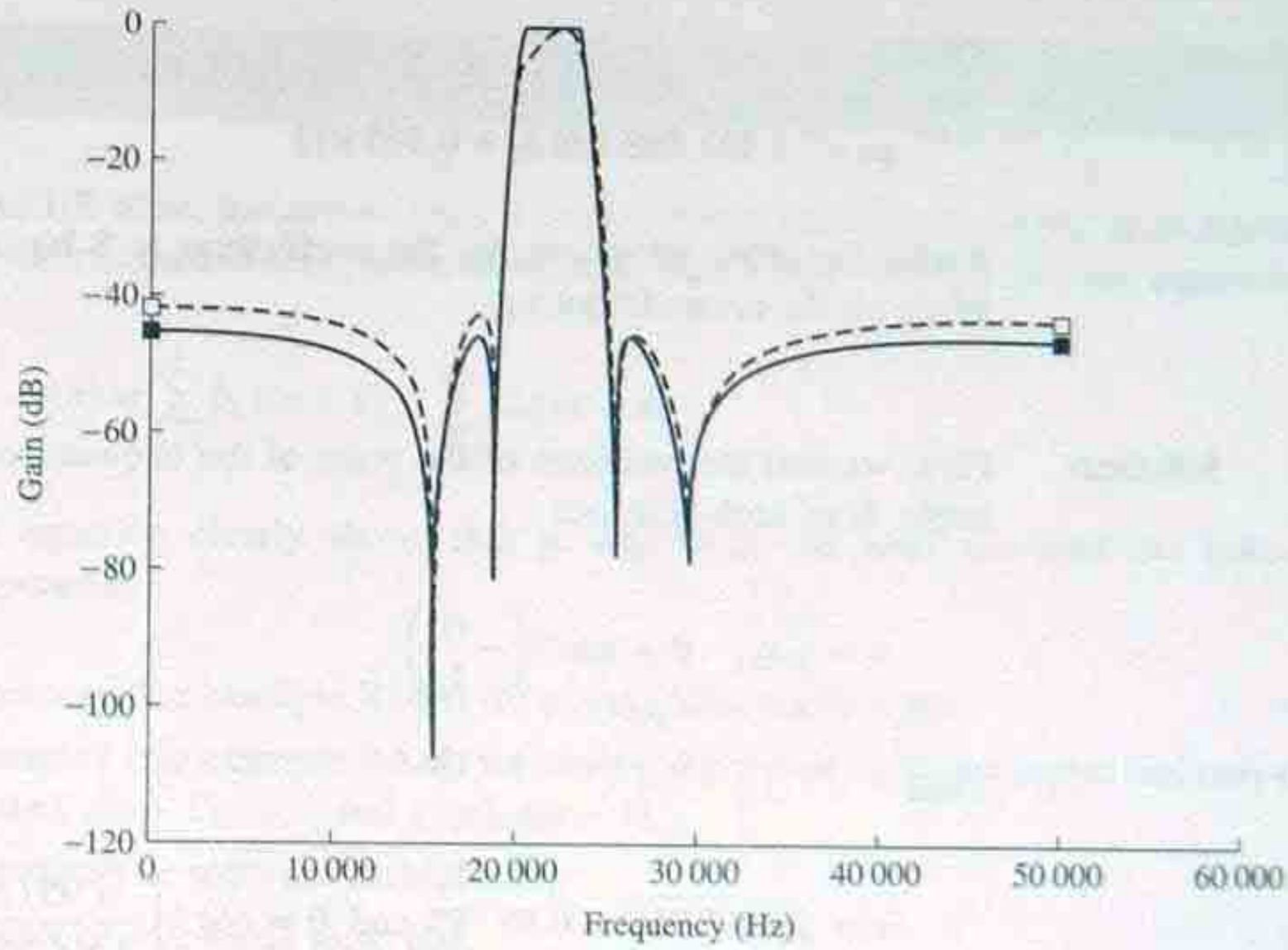


Figure 8.30 Practical effects of coefficient quantization on the frequency response: ■, unquantized; □, quantized, 5 bits.

The primary effect of quantizing filter coefficients using a finite number of bits is to alter the positions of the poles and zeros of $H(z)$ in the z-plane. This could lead to:

- instability or potential instability for high-order filters, with sharp transition widths and poles close to the unit circle;
- a change in the desired frequency response, as illustrated in Figure 8.30.

The quantized filter should be analyzed to ensure that its wordlength is sufficient for both stability and satisfactory frequency response. We will illustrate the effects of coefficient quantization on the frequency response of an IIR filter in the next example. A more detailed analysis of the effects of coefficient quantization on filter performance is given in Chapter 13.

Example 8.21 A bandpass digital filter is to be used for digital clock recovery at 4.8 kbaud and a sampling frequency of 153.6 kHz. The filter is characterized by the following transfer function:

$$H(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

where

$$a_1 = -1.957\,558 \text{ and } a_2 = 0.995\,913$$

Assess the effect of quantizing the coefficients to 8 bits on the pole positions and hence on the centre frequency.

Solution First, we find the positions of the poles of the unquantized filter. The radius, r, and angle, θ, of each pole are:

$$r = \sqrt{a_2}, \quad \theta = \cos^{-1}\left(-\frac{a_1}{2r}\right)$$

Thus

$$r = \sqrt{0.995\,913} = 0.997\,95 \text{ and } \theta = \cos^{-1}\left(\frac{1.957\,558}{2 \times 0.997\,95}\right) = 11.25°$$

This corresponds to a centre frequency of

$$\left(\frac{11.25}{360}\right) \times 153.6 \times 10^3 = 4.7999 \text{ kHz}$$

As one of the coefficients is greater than unity, we will assign 1 bit as the sign bit, 1 bit for the integer and 6 bits for the fractional part of the coefficients. Thus, after quantizing to 8 bits the coefficient values become:

$$a_1 = -1.957\,558 \times 2^6 = -125 \; (\equiv 10000100)$$

$$a_2 = 0.995\,913 \times 2^6 = 63 \text{ (maximum positive fraction) } (\equiv 00111111)$$

In fractional notation, the quantized coefficient values are:

$$a_1 = -\frac{125}{64} = -1.953\,125; \quad a_2 = \frac{63}{64} = 0.984\,375.$$

The new pole position becomes: $r = 0.992\,156$, $\theta = 10.17°$; and the centre frequency now becomes

$$f_0 = \left(\frac{10.17}{360}\right) \times 153.6 \times 10^3 = 4.34 \text{ kHz}$$

8.15 Implementation of IIR filters

In the IIR filter, the output, $y(n)$, is computed for each input sample, $x(n)$. Assuming cascade realization using the second-order direct form, the key filtering equation is

$$y(n) = \sum_{k=0}^{2} b_k x(n-k) - \sum_{k=1}^{2} a_k y(n-k)$$

This equation clearly shows that to implement the filter we need the following components:

- memory (for example ROM) for storing filter coefficients;
- memory (for example RAM) for storing the present and past inputs and outputs $\{x(n), x(n-1), \ldots\}$ and $\{y(n), y(n-1), \ldots\}$;
- hardware or software multiplier(s);
- adder or arithmetic logic unit.

In modern real-time DSP, the filtering operations are efficiently performed with a DSP processor such as the TMS320C50. These processors have all the basic blocks on-board, including in-built hardware multiplier(s). In some applications, standard 8-bit or 16-bit microprocessors such as the Motorola 6800 or 68000 families offer attractive alternative implementations. In addition to the signal processing hardware, the designer must also provide suitable input–output (such as analog–digital–analog conversion) interfaces to the digital hardware, depending on the type of data source and sink. This approach may be described as hardware implementation.

In batch or off-line processing a suitable high level language is used to implement the filter. In this case, the filter is often implemented in a high level language (for example C or FORTRAN) and runs in a general purpose computer, such as a personal computer or a mainframe computer, where all the basic blocks are already configured. Thus, batch processing may be described as a purely software implementation.

Computational requirements

The designer must analyze the impact of the computational requirements of a digital filter on the processor that will be used. The primary requirements for digital filters are multiplication, additions, accumulation and delays or shifts. For example, a filter consisting of a second-order section would require typically four multiplications, four additions, and some shifts and storage. If the filtering is performed in real time, for example at 44.1 kHz (for digital audio), the arithmetic operations must be performed once every 1/(44.1 kHz). Allowance must also be made for other overheads such as fetching the input data or saving or outputting the filtered data samples as well as other housekeeping operations.

8.16 A detailed design example of an IIR digital filter

This example will be used to illustrate some of the many concepts presented in this chapter. In particular, we shall see how the five-stage design procedure is applied.

Stage 1: filter specifications

Design and implement a lowpass IIR digital filter using a software package and the TMS320C50-based target board to meet the following specifications:

sampling frequency	15 kHz
passband	0–3 kHz
transition width	450 Hz
passband ripple	0.5 dB
stopband attenuation	45 dB

Note: The target board has a 12-bit ADC and 12-bit DAC.

Stage 2: coefficient calculation

Using the software design program (on the CD in the companion handbook) for IIR filters, it was found that a fourth-order elliptic filter, via the bilinear transform method, would be required to satisfy the specifications. The output listing of the design program is summarized below.

	Denominator A_k	*Numerator* B_k
1	1.000000E+00	5.846399E–02
2	–1.325263E+00	1.359507E–01
3	1.480202E+00	1.820297E–01
4	–7.841098E–01	1.359506E–01
5	2.339270E–01	5.846398E–02

Poles		*Coefficients*	
Real	*Imaginary*	z^{-1}	z^{-2}
0.247967	0.836885	–0.495935	0.761864
0.414664	0.367559	–0.829328	0.307046

Zeros		*Coefficients*	
Real	*Imaginary*	z^{-1}	z^{-2}
–0.337859	0.941197	0.675718	1.000000
–0.824828	0.565383	1.649656	1.000000

From the listing, the transfer function of the filter, in direct form, is given by

$$H(z) = \frac{0.058\,463\,99 + 0.135\,950\,7z^{-1} + 0.182\,097\,9z^{-2} + 0.135\,950\,6z^{-3} + 0.058\,463\,98z^{-4}}{1 - 1.325\,263z^{-1} + 1.480\,202z^{-2} - 0.784\,109\,8z^{-3} + 0.233\,927z^{-4}}$$

Stage 3: realization

As explained earlier, direct form realization of $H(z)$ is very sensitive to many adverse effects of finite wordlength such as coefficient quantization errors, so it is important to break $H(z)$ down into smaller sections and then connect these up, for example in cascade or parallel structure. Assuming cascade structure, $H(z)$ is broken down into two second-order sections $H_1(z)$ and $H_2(z)$:

$$H(z) = H_1(z)H_2(z)$$

where

$$H_1(z) = \frac{b_{01} + b_{11}z^{-1} + b_{12}z^{-2}}{1 + a_{11}z^{-1} + a_{21}z^{-2}}$$

$$H_2(z) = \frac{b_{02} + b_{12}z^{-1} + b_{22}z^{-2}}{1 + a_{12}z^{-1} + a_{22}z^{-2}}$$

The realization diagram is depicted in Figure 8.31, with each filter section realized using a standard biquad structure. The corresponding sets of difference equations, which define how the filtering operation will be carried out, are as follows.

Filter section 1

$$w_1(n) = (1/s_1)x(n) - a_{11}w_1(n-1) - a_{21}w_1(n-2)$$

$$y_1(n) = b_{01}w_1(n)s_1/s_2 + b_{11}w_1(n-1)s_1/s_2 + b_{21}w_1(n-2)s_1/s_2$$

Filter section 2

$$w_2(n) = y_1(n) - a_{12}w_2(n-1) - a_{22}w_2(n-2)$$

$$y_2(n) = b_{02}w_2(n)s_2 + b_{12}w_2(n-1)s_2 + b_{22}w_2(n-2)s_2$$

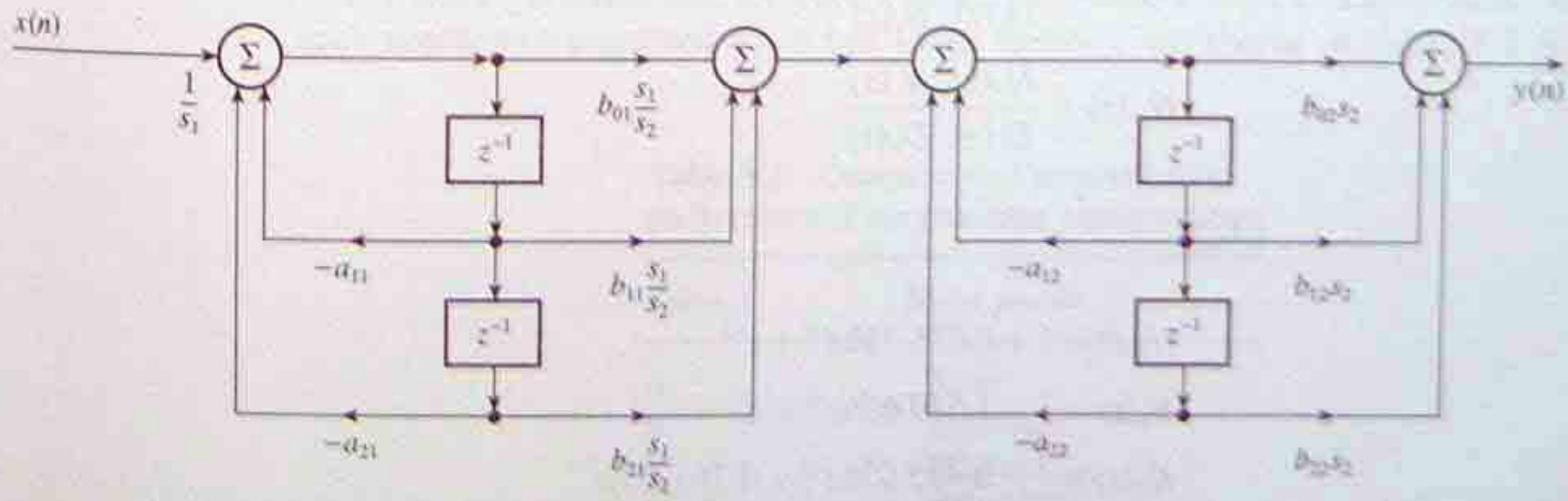


Figure 8.31 Realization diagram for the detailed design example.

The exact values of the coefficients, a_{ij} and b_{ij}, are dependent on how we pair the numerator and denominator polynomials of $H(z)$ and how the second-order filter sections used to realize the polynomials are ordered. The best pairing and ordering can only be determined from finite wordlength analysis.

Stage 4: analysis of finite wordlength effects

For the problem, based on the specifications given, we will assume that fixed point two's complement arithmetic will be used and that each coefficient will be quantized, by rounding, to a 16-bit wordlength.

The main objective here is to assess the effects of the various quantization errors on the filter performance and to determine the best filter configuration to implement, in terms of signal-to-noise ratios. The sources of errors of concern are (see Chapter 13 for details):

- overflow errors,
- roundoff errors, and
- coefficient quantization errors.

To avoid overflow at the output of the adders in Figure 8.31, suitable scale factors are introduced before the adders as shown in the figure.

Since $H(z)$ is fourth order to be realized as two second-order sections, its numerator and denominator factors can be paired and ordered in four possible ways:

$$H_A(z) = \frac{N_1(z)}{D_1(z)}\frac{N_2(z)}{D_2(z)}$$

$$H_B(z) = \frac{N_2(z)}{D_2(z)}\frac{N_1(z)}{D_1(z)}$$

$$H_C(z) = \frac{N_1(z)}{D_2(z)}\frac{N_2(z)}{D_1(z)}$$

$$H_D(z) = \frac{N_2(z)}{D_1(z)}\frac{N_1(z)}{D_2(z)}$$

where

$$N_1(z) = 1 + 0.675\,718z^{-1} + z^{-2}$$

$$N_2(z) = 1 + 1.649\,656z^{-1} + z^{-2}$$

$$D_1(z) = 1 - 0.495\,935z^{-1} + 0.761\,864z^{-2}$$

$$D_2(z) = 1 - 0.829\,328z^{-1} + 0.307\,046z^{-2}$$

Table 8.1 Scale factors for the four filter configurations.

Filter	*Scale factor*	L_1	L_2	L_∞
A	s_1	5.524 844	1.608 890	4.379 544
	s_2	11.821 571	3.677 381	7.262 393
B	s_1	2.479 158	1.359 467	2.175 539
	s_2	18.908 47	10.880 490	12.548 114
C	s_1	2.479 158	1.359 467	2.175 539
	s_2	11.821 571	10.880 490	7.262 393
D	s_1	5.524 844	1.608 890	4.379 544
	s_2	18.908 47	5.727 459	12.548 114

Each of the four possible filter configurations will have different scale factors as well as different signal to roundoff noise performance. An objective at this stage is to determine the best pairing and ordering, in terms of signal-to-noise performance, for the filter. Overflow and roundoff errors are intimately linked, and so scaling and roundoff analysis should be performed simultaneously.

Using the finite wordlength analysis program, scale factors based on the L_1 norm, L_2 norm and L_∞ norm for the four possible filters above were obtained and are summarized in Table 8.1. In this example, we have used the L_1 norm. For a fourth-order filter realized as a cascade of two second-order canonic sections, the roundoff noise at the output, after scaling, is given by

$$\sigma_o^2 = \frac{q^2}{12}[3s_1^2\|H_1(z)H_2(z)\|_2^2 + 5s_2^2\|H(z)\|_2^2 + 3]$$

where q is the quantization stepsize or rounding, $\|\cdot\|_2^2$ symbolizes the L_2 norm squared, $H_1(z)$ is the transfer function for the first filter stage, $H_2(z)$ is the transfer function for the second filter stage, s_1 is the scale factor for the first filter stage and s_2 is the scale factor for the second filter stage.

The noise performances for each of the four possible filter configurations, with each coefficient quantized to 16 bits (after scaling), are shown in Table 8.2. It is

Table 8.2 Comparison of roundoff noise performance of the four filter configurations.

Filter	*Noise power*
A	$703q^2$
B	$326.378q^2$
C	$382.32q^2$
D	$570.453q^2$

evident that filter B has the best roundoff noise performance. The scaled filter transfer function is given by

$$H(z) = H'_g(z) = \frac{s_1}{s_2}\frac{N_2(z)}{D_2(z)}\frac{N_1(z)}{D_1(z)}s_2$$

$$= 0.1311136\,\frac{1 + 1.649\,656z^{-1} + z^{-2}}{1 - 0.829\,328z^{-1} + 0.307\,046z^{-2}}$$

$$\times \frac{1 + 0.675\,718z^{-1} + z^{-2}}{1 - 0.495\,935z^{-1} + 0.761\,864z^{-2}}\,10.880\,490$$

Next, we analyze the effects of coefficient quantization errors. Specifically, we check that the specified coefficient wordlength is adequate for stability and to meet the frequency response specifications. As the poles are not very close to the unit circle a 16-bit coefficient wordlength is adequate for stability. For example, for the first filter section the FWA program shows that as few as three bits are adequate for stability, and quantizing the coefficients to 16 bits altered the poles radius by a mere 0.000 48%. The FWA program also showed that only 12 bits are required to maintain the frequency response within the tolerance limits. With 16-bit coefficient wordlength, the response of the filter is essentially the same as the unquantized filter response. The frequency response and pole–zero diagram for the unquantized filter are depicted in Figure 8.32.

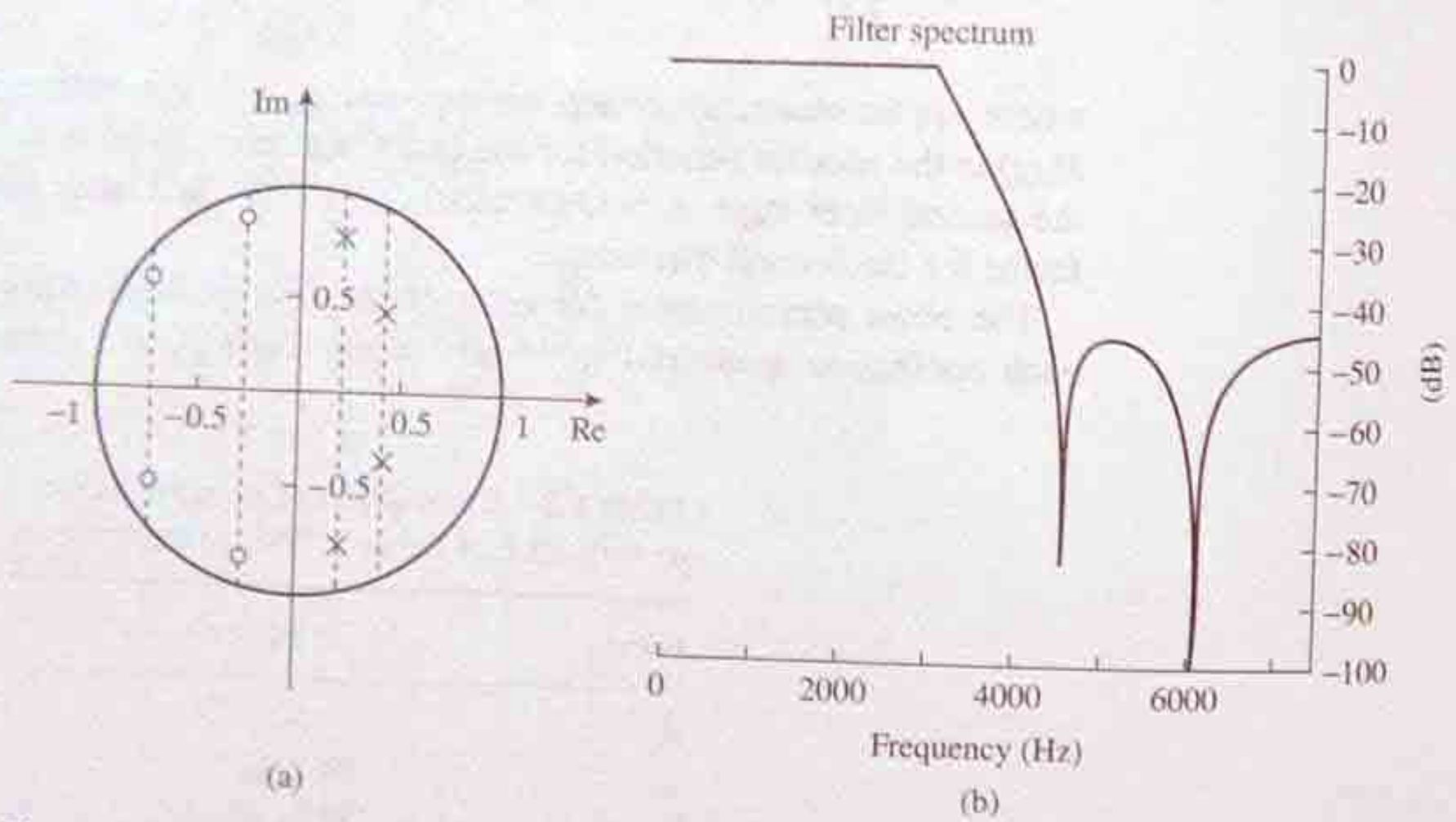


Figure 8.32 (a) The pole–zero diagram, and (b) the frequency response for the detailed design example.

Double-length accumulation of sums of products and post-accumulation quantization are used to keep the effects of roundoff noise to a minimum.

Stage 5

The quantized coefficients (obtained by multiplying the scaled coefficients by 2^{15}) are entered into the TMS320C25 IIR filter program listed in the appendix. Chapter 12 gives a more detailed discussion of the IIR filtering routines and their development.

8.17 Summary

The design of IIR filters can be divided into five interrelated stages (see Figure 8.1). Filter specifications are often application dependent, but should include details such as bandedges, tolerance limits for the amplitude response, sampling rate and I/O requirements. For filters with standard characteristics, the coefficients required to meet the amplitude response specifications can be efficiently obtained via the BZT. This approach as well as other useful coefficient calculation methods are described in the text and illustrated with many examples. Higher-order IIR filters are realized as cascades or parallel combinations of second- and first-order sections to keep changes in the positions of the poles and zeros due to the effects of finite wordlength small. Scaling the input to each section to prevent overflow in the internal nodes is necessary.

The performance of an IIR digital filter is limited by the number of bits used in its implementation. The four common sources of errors are those due to (1) input quantization, (2) coefficient quantization, (3) product roundoff and (4) addition overflow. Techniques for analyzing their effects on filter performance and, where possible, for eliminating or minimizing them have been presented. Coefficient wordlength must be adequate to minimize the effects of coefficient quantization on the frequency response and to prevent the possibility of instability. The stability of an IIR filter is always of concern. An IIR filter that is otherwise stable when implemented with infinite precision may become unstable if implemented with finite precision. In high fidelity audio work, for example, 24-bit coefficients are said to be necessary for processing low frequency audio signals. In most other cases, representing the coefficients with 16 or more bits and carrying out the arithmetic operations with double-length accumulators are sufficient to minimize the effects of finite wordlength.

Truncation or roundoff errors due to finite precision arithmetic operations create a nonlinear effect in the filter, such as limit cycles, whereby the filter output oscillates even in the absence of an input (or a constant input). The effects of roundoff error on filter performance can be quantified in terms of SNRs at the filter output. Reduction in the SNR due to roundoff error can be offset by the use of the error spectral shaping (ESS) scheme (see Chapter 13 for details). The primary effect of such schemes is to nullify the 'amplifying' effect of the poles of the filter on the roundoff errors. The

price paid for this is an increase in the number of multiplications and additions, although first-order ESS with integer coefficients is computationally efficient.

Design programs are provided on the CD in the companion handbook to enable designers to calculate the filter coefficients and to analyze some of the effects of finite wordlength on filter performance (see the Preface for details).

8.18 Application examples in digital audio and instrumentation

This section provides an overview of several applications where the IIR filter either has been used or is appropriate.

8.18.1 Digital audio

Digital filters have found use in many areas of digital audio, especially in systems with high quality digital sources such as the CD player and DAT, where it makes sense to carry out digitally as many of the signal processing operations as possible. DSP also makes it possible to generate acoustic properties of locations such as concert halls, jazz clubs and discos. Applications in digital audio where an IIR filter has been used include graphic equalization, tone control, channel equalization, noise shaping in ADC/DAC, and band splitting.

In digital graphic equalizers, for example, IIR filters are used to split the entire audio frequency range into bands enabling a comprehensive tone adjustment of the reproduced sound to personal taste instead of the bass and treble controls. A typical five-band graphic equalizer would split the audio frequency range into five bands with centre frequencies of 100 Hz, 330 Hz, 3.3 kHz, 10 kHz and 16 kHz and allow adjustable signal level in each band in the range ±10 dB.

A simple filtering arrangement for graphic equalization is shown in Figure 8.33.

8.18.2 Digital control

With the increased awareness of the benefits of DSP and the availability of low cost processors, controllers are now being implemented digitally to achieve better accuracy and flexibility. Figure 8.34 shows the principles of digital control of an analog plant, $H(s)$, which could be a car or motor, for example. In general, the digital controller has an IIR characteristic.

8.18.3 Digital frequency oscillators

IIR filters have been used to generate accurate waveforms instead of the traditional look-up table approach. The approach exploits the fact that an IIR filter with poles on

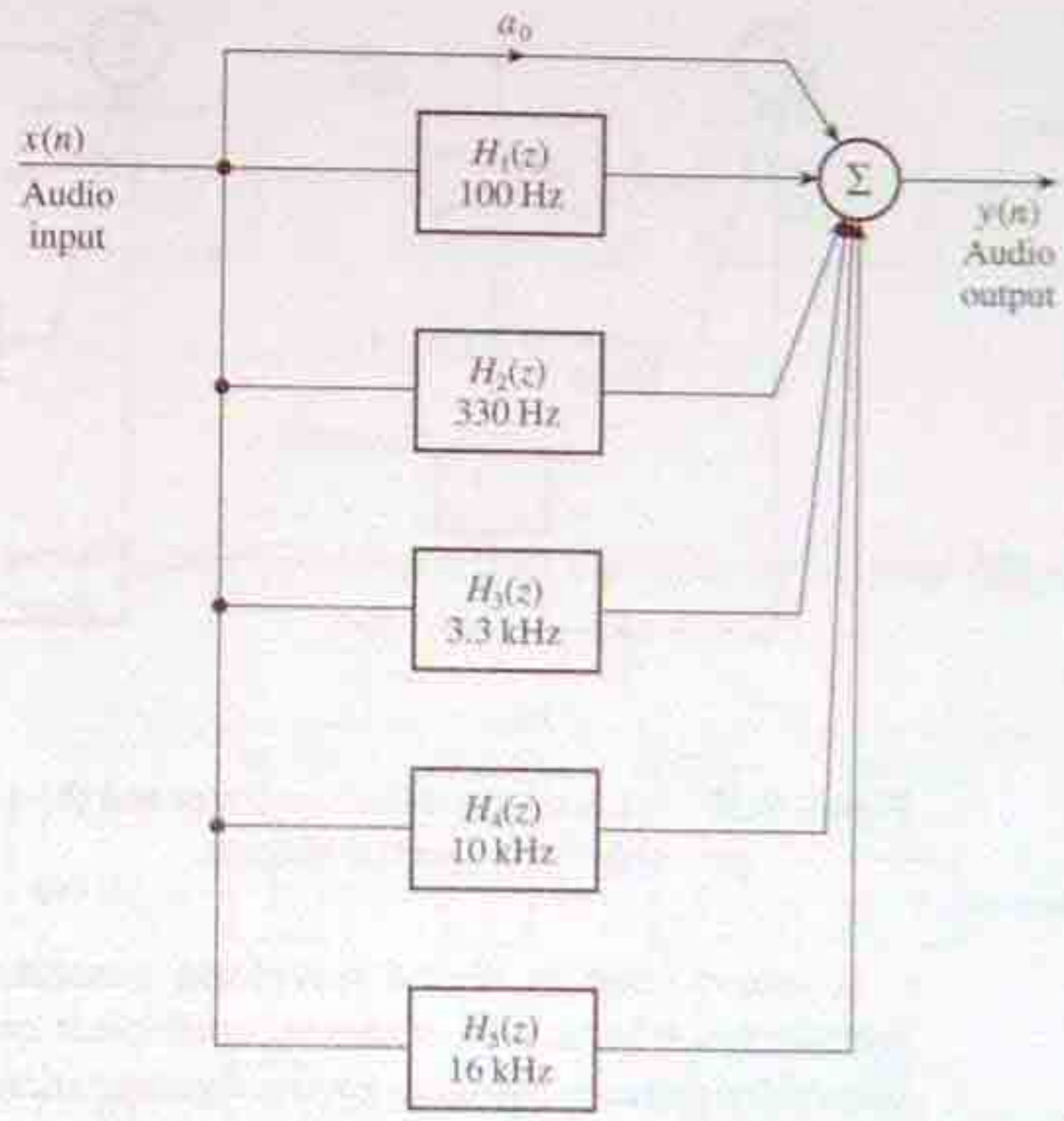


Figure 8.33 Simplified diagram of an all-digital graphic equalizer. The main component is a bank of parallel IIR filters with different centre frequencies. The gain of each filter is individually adjustable, for example with a sliding potentiometer, in the range ±10 dB, say.

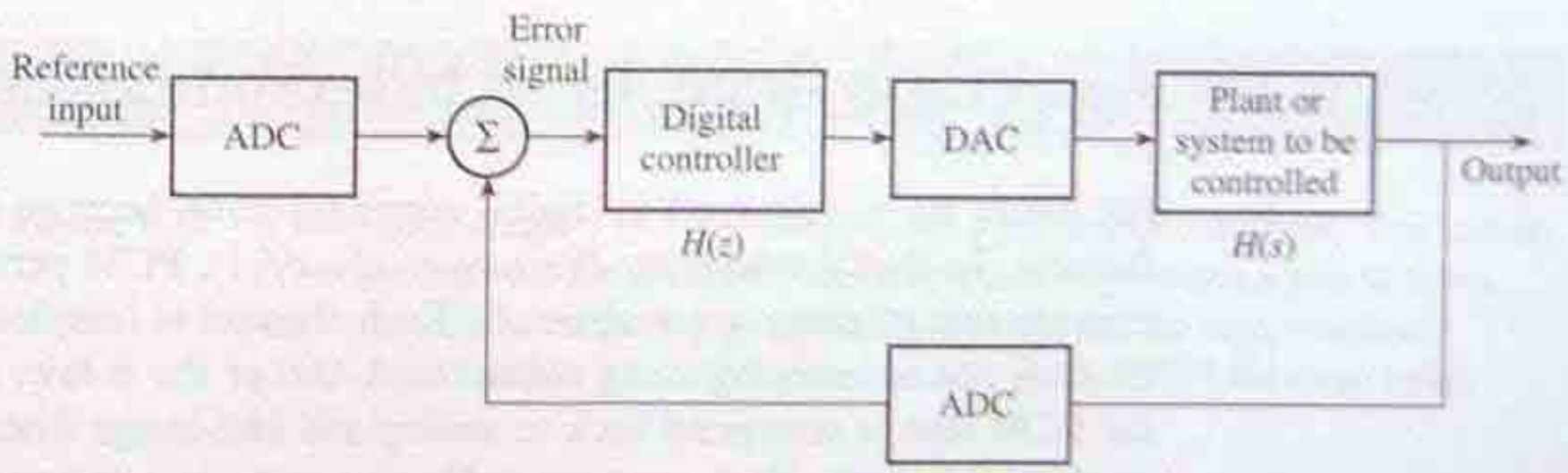


Figure 8.34 Principles of digital control of an analog plant.

the unit circle is essentially unstable. A simple sine wave oscillator is depicted in Figure 8.35(a). The poles of the IIR filter are located at $e^{\pm j\theta}$ and the frequency of oscillation is given by

$$\theta' = w_0 T_B$$

where T is the sampling period. The filter coefficient, $2\cos\theta'$, is constrained to be an integer by taking the integer part of $2^B \times 2\cos\theta'$ (B is the number of bits).

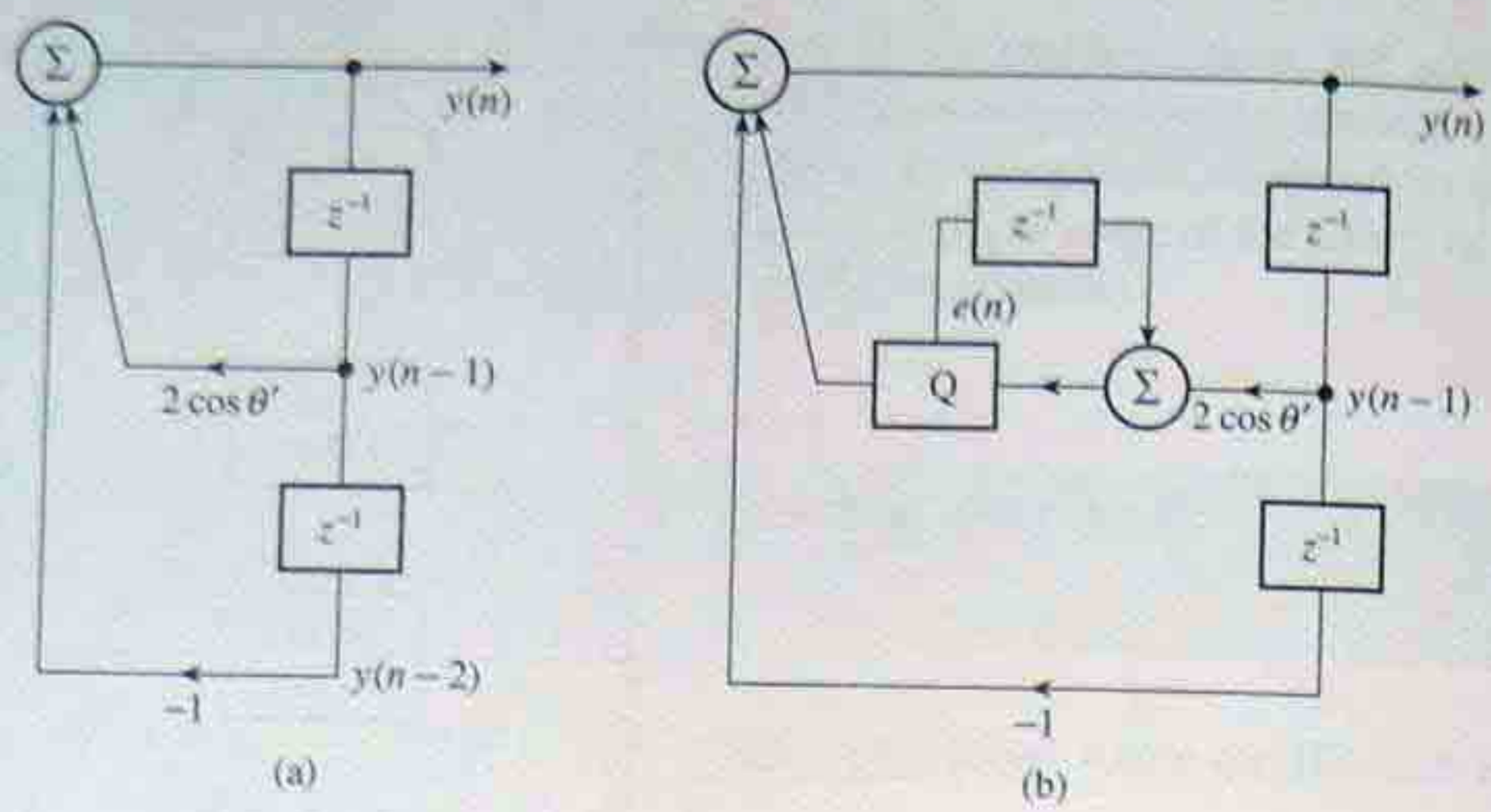


Figure 8.35 (a) A simple digital oscillator and (b) a simple digital oscillator with a first-order error spectral shaping.

A major issue in digital waveform generation using IIR filters concerns finite wordlength effects. For example, coefficient quantization would lead to unequally spaced frequencies, whereas product quantization leads to a buildup of roundoff errors which soon renders the waveform generator useless. However, using ESS techniques these errors can be reduced to a minimum. Figure 8.35(b) shows an oscillator utilizing the ESS technique (Abu-el-Haija and Al-Ibrahim, 1986), which offers a significant reduction in the roundoff noise effects.

8.19 Application examples in telecommunication

IIR filters are widely used in digital communication because of their characteristic features. In digital telephony (Feeney *et al.*, 1971), PCM permits the simultaneous transmission of many voice channels. Each channel is sampled at 8 kHz after band-limiting and is encoded using either the A-law or the μ-law. At the receiving end, the PCM data is converted back to analog and anti-image filtered. Digital IIR filters can be used to provide the necessary filtering at the transmit and receive ends (Figure 8.36). In this case, the filtering is performed at a higher sampling rate, for example 32 kHz, and then converted from linear to standard PCM codes.

In the next two sections, we will discuss in more detail two specific applications of IIR filters in digital communication – one for digital pushbutton telephones and the other for clock recovery in data communication.

8.19.1 Touch-tone generation and reception for digital telephones

An excellent application of IIR filter is the all digital dual-tone multifrequency touch-tone receiver (Jackson *et al.*, 1968; Mock, 1985).

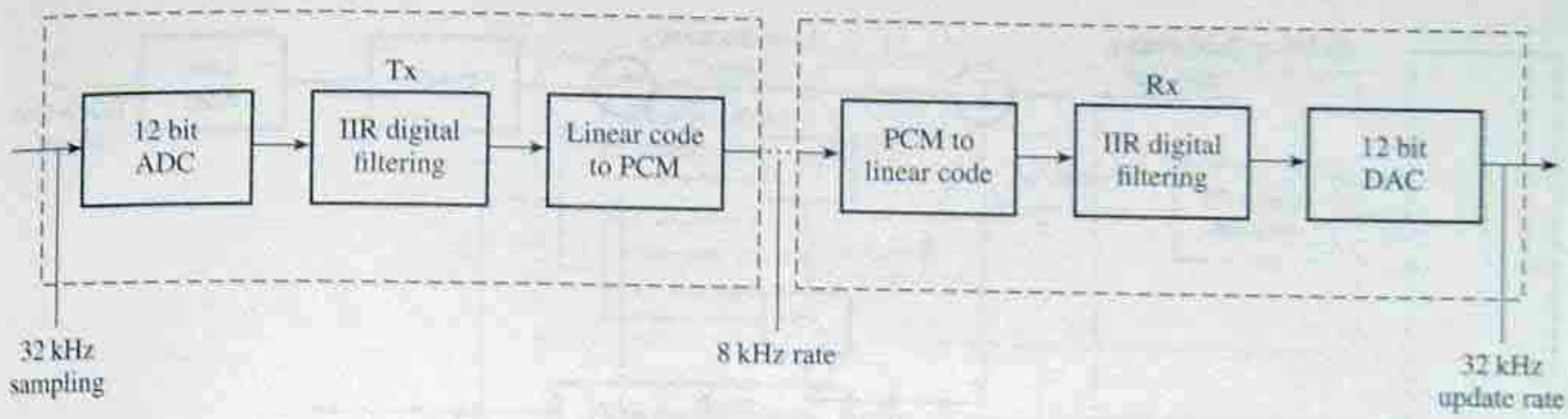


Figure 8.36 A PCM channel showing possible use of IIR filters for the main anti-aliasing filter (TX end) and anti-imaging filtering (RX end).

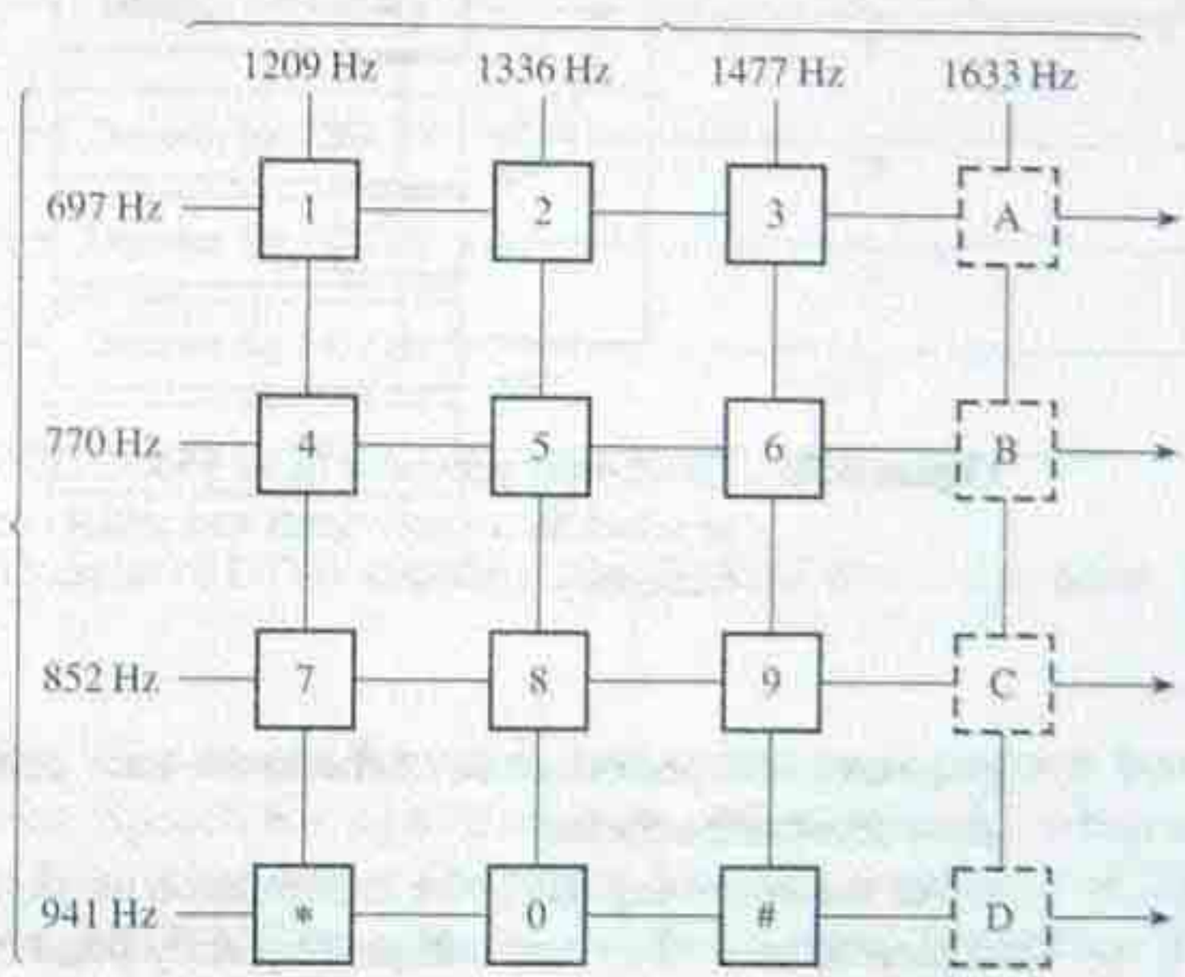


Figure 8.37 A simplified diagram of a 4 × 4 keypad for a touch-tone telephone. The buttons in broken lines are not available. Pressing a button generates a pair of tones, one from the low frequency group and the other from the high frequency group. For example, pressing 9 generates 852 Hz and 1477 Hz tones (after Mock, 1985).

In modern telephone systems, the information required to establish communication, and for maintenance and charging, is normally provided by a multifrequency code. Typically, the telephone set generates two tones, one low frequency tone and another high frequency tone (see Figure 8.37).

The tone generator can be implemented using a pair of programmable second-order IIR oscillators (Figure 8.38). When a button is pressed the code for the dialled digit is used to select the appropriate filter coefficients and initializing conditions from ROMs to produce a pair of tones (one high frequency tone and one low frequency tone). The tones are added to produce the touch-tone signal. As with the digital sine wave

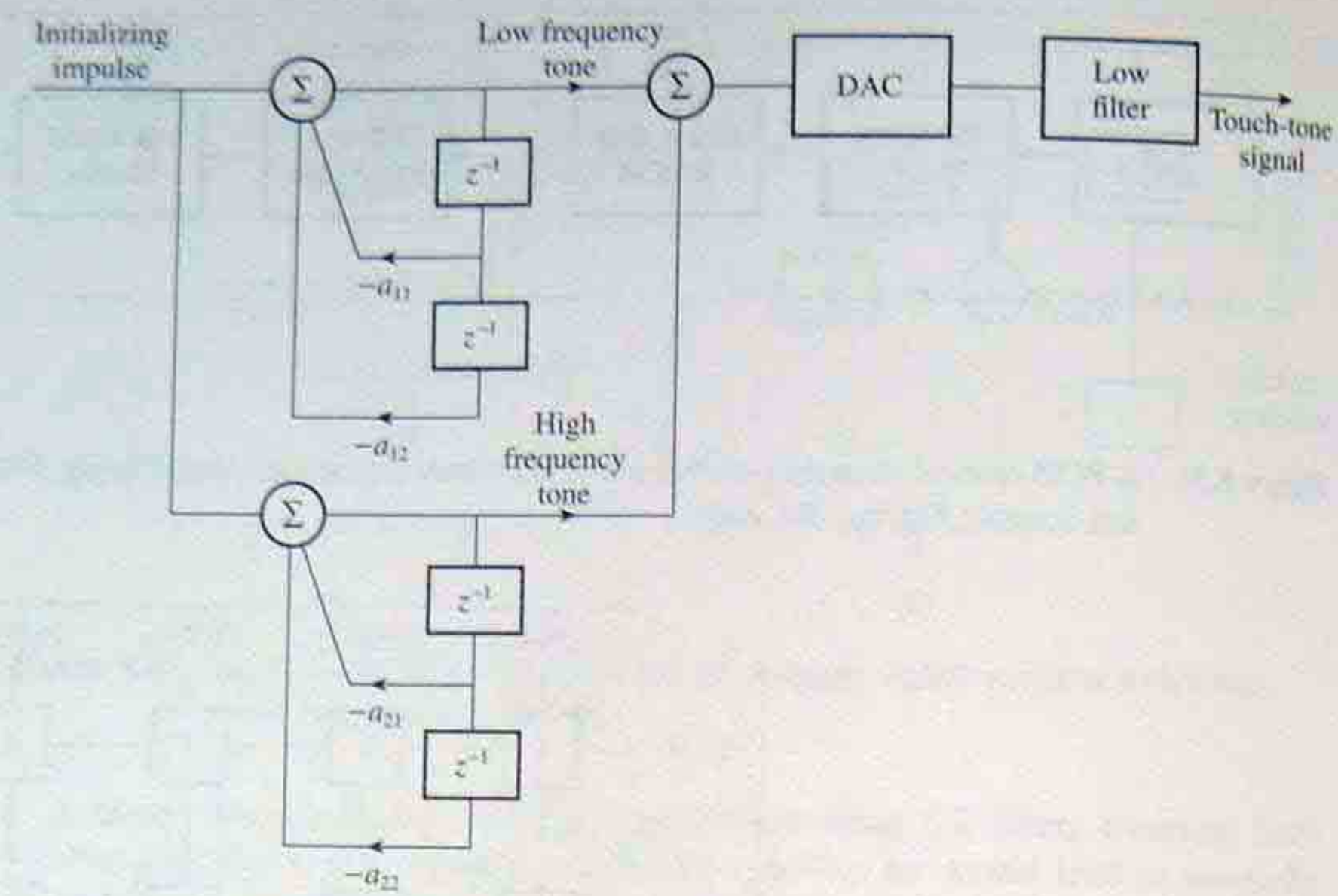


Figure 8.38 Touch-tone generator (after Mock, 1985). The code for the digit dialled is used to select filter coefficients and initial conditions and hence the oscillator frequency.

generator, the performance of the touch-tone generator can be improved by using an error feedback scheme.

At the receiving end, the information is digitized at a rate of 8 kHz and then separated into a low and a high frequency band by the front-end bandpass IIR filters. To detect the presence of a tone, level detection is carried out. This is performed by combined bandpass filtering and full wave rectification followed by lowpass filtering. To determine which of the low frequency tones is present the low frequency band is split into four bands by two sets of four BPFs. The same is true of the high frequency band. The resulting eight levels are passed to decision logic to determine the received code.

8.19.2 Digital telephony: dual tone multifrequency (DTMF) detection using the Goertzel algorithm

The Goertzel algorithm may be used as an alternative to standard IIR filters for the detection of DTMF tones (Mock, 1985; Marven, 1990; Chen, 1996; Texas Instruments, 1997). The Goertzel algorithm is a special IIR filter implementation of the discrete Fourier transform (DFT). The block diagram of a DTMF detection scheme based on the Goertzel algorithm is depicted in Figure 8.39. It consists of a parallel bank of eight pairs of Goertzel filters. Each filter pair detects a DTMF tone and its

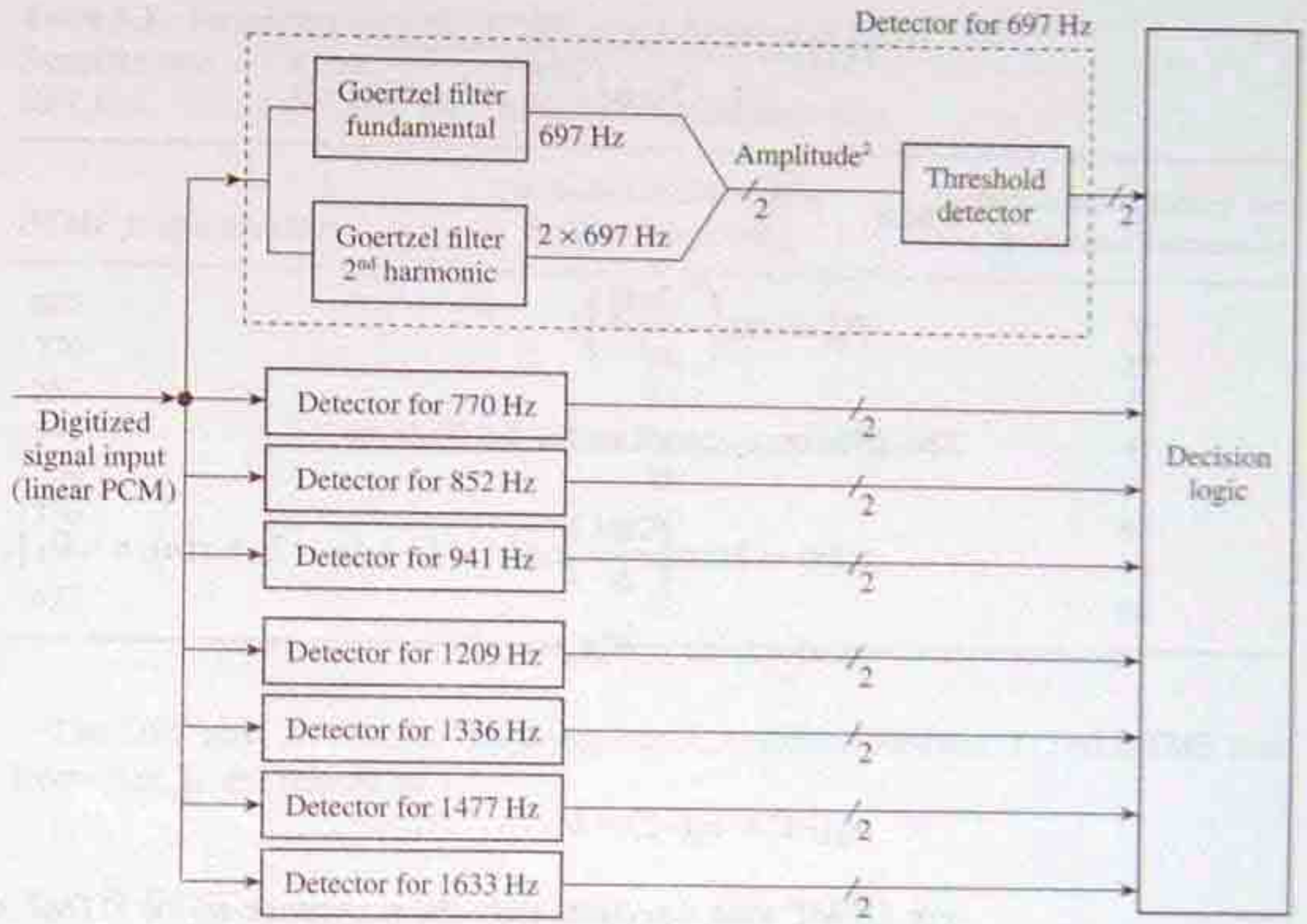


Figure 8.39 Principles of DTMF decoding using Goertzel filters (after Mock, 1985).

second harmonic. The second harmonic is required to distinguish between speech and DTMF tones. Speech has significant even-order harmonics whereas the DTMF signals do not. Each filter output is squared to obtain a measure of signal strength at each of the eight DTMF frequencies and their second harmonics. The strongest signal pair from the high and low frequency groups are used to determine the received digit.

Each Goertzel filter is a high Q, narrowband, second-order, bandpass IIR filter characterized by the following transfer function (see Figure 8.40):

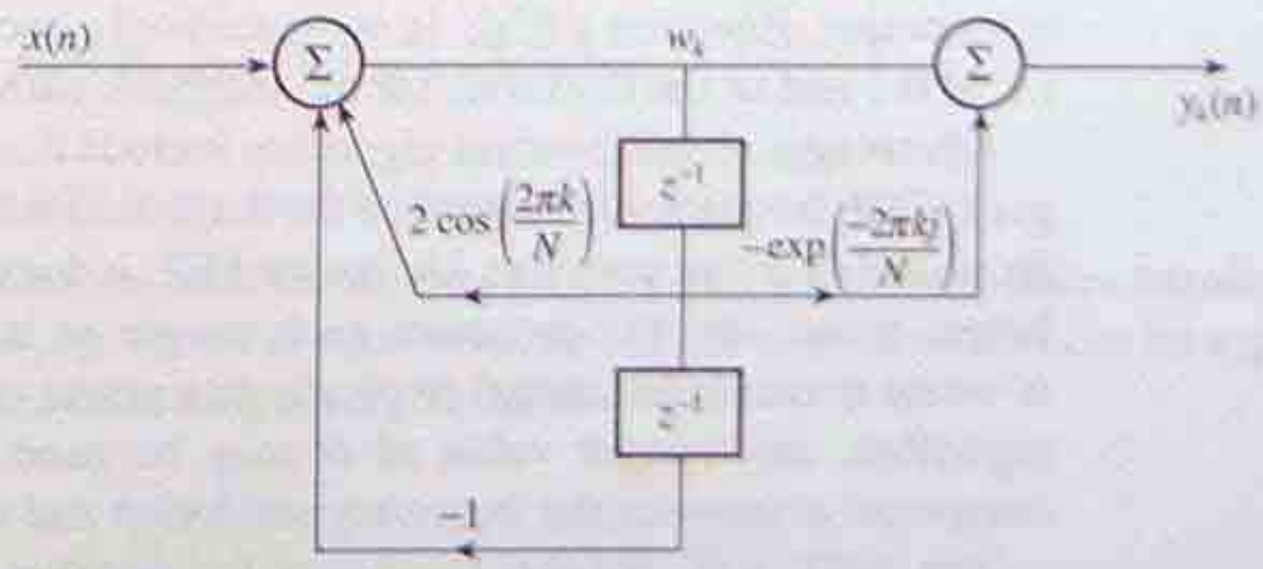


Figure 8.40 Structure of a second-order Goertzel filter.

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{1 - 2\cos\left(\frac{2\pi k}{N}\right)z^{-1} + z^{-2}} \tag{8.53a}$$

where

$$W_N^k = \exp\left(-\frac{2\pi kj}{N}\right)$$

The difference equations for the filter are

$$v_k(n) = 2\cos\left(\frac{2\pi k}{N}\right)v_k(n-1) - v_k(n-2) + x(n),\ n = 0, 1, \ldots, N \tag{8.53b}$$

$$y_k(n) = v_k(n) - W_N^k v_k(n-1) \tag{8.53c}$$

where

$$w_k(-1) = w_k(-2) = 0$$

For DTMF tone detection, only the magnitude of the DTMF signal is required (the phase information is ignored) and so the GA is modified to produce magnitude squared outputs only:

$$|y_k(N)|^2 = v_k^2(N) + v_k^2(N-1) - 2\cos\left(\frac{2\pi k}{N}\right)v_k(N)v_k(N-1) \tag{8.53d}$$

The modified DTMF algorithm (Equations 8.53b and 8.53d) requires only one real coefficient, i.e.

$$2\cos\left(\frac{2\pi k}{N}\right)$$

to calculate the magnitude of each DTMF signal, and avoids complex arithmetic. The filter output, Equation 8.53d, is computed only once for each DTMF tone for $n = N$, i.e. at the end of the iterations for the feedback path of the filter (Equation 8.53b).

Advantages of the Goertzel algorithm include that it needs only one real coefficient per DTMF frequency to determine the magnitude of the signal, takes up little memory space and executes very fast. Unlike the FFT, it does not wait for the complete data set before processing, but processes each sample as it arrives. With FFTs, the value of N would normally be limited (typically to a power of 2) for efficiency. In the Goertzel algorithm, any integer value of N may be used, although the choice of N is a compromise between the frequency resolution and computation time.

The DFT size, N, and frequency bin number, k, determine the value of the coefficient of the filter and hence the frequency detected.

Table 8.3 Parameters for a DTMF decoding scheme (after Mock, 1985).
Sampling rate: 8 kHz
DFT size: 205 (first harmonic), 210 (second harmonic).

DTMF frequency (Hz)	*Discrete frequency bin,* k *(first harmonic)*	*Discrete frequency bin,* k *(second harmonic)*
697	18	35
770	20	39
852	22	43
941	24	47
1209	31	61
1336	34	67
1477	38	74
1633	42	82

The DFT size, N, discrete frequency bin, k, sampling interval, T, and DTMF tone frequency, f_k, are related as

$$f_k = \frac{k}{NT}$$

The sampling frequency and tone frequencies are set in accordance with international standards. The DFT size, N, may be varied. Table 8.3 lists the parameters for a possible decoding scheme (Mock, 1985).

It should be pointed out that the Goertzel filter has its poles on the unit circle and is sensitive to finite wordlength effects and these should not be ignored. Further, if the number of frequency points to be determined is relatively large the FFT may be more suitable.

Example 8.22

(a) The discrete Fourier transform (DFT) of a data sequence, $x(n)$, $n = 0, 1, \ldots, N-1$, may be defined as

$$X(k) = \sum_{m=0}^{N-1} x(m) W_N^{km}, \quad k = 0, 1, \ldots, N-1$$

where W_N^{km} is the twiddle factor.

(i) Starting with the above equation, show that the z-plane transfer function, $H_k(z)$, for a Goertzel filter for DTMF tone detection can be expressed in the following recursive form:

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

(ii) Deduce an expression for the magnitude squared output of the Goertzel filter, $|y_k(n)|^2$, at the discrete time, $n = N$, and hence show that complex arithmetic is not required in the modified Goertzel algorithm.

(b) A DTMF tone detection scheme for a pushbutton telephone system is based on the specifications summarized in Table 8.3 and uses a second-order Goertzel filter.

Calculate the coefficients for the Goertzel filter to decode the digits at the receiver if the number dialled is '99'.

Solution (a) (i) Now

$$X(k) = \sum_{m=0}^{N-1} x(m) W_N^{km}$$

Exploiting the periodicity of the twiddle factor, we can write the DFT equation as follows (since $W_N^{-kN} = 1$):

$$\begin{aligned} X(k) &= W_N^{-kN} \sum_{m=0}^{N-1} x(m) W_N^{km} \\ &= \sum_{m=0}^{N-1} x(m) W_N^{-k(N-m)} \end{aligned} \tag{8.54}$$

Equation 8.54 has the same form as the convolution equation. Thus, if we define a data sequence, $y_k(n)$, as follows:

$$y_k(n) = \sum_{m=0}^{N-1} x(m) W_N^{-k(n-m)} \tag{8.55}$$

it is seen that $y_k(n)$ can be viewed as the output of an FIR filter which has as input the data sequence, $x(m)$, and N coefficients, $h_k(n)$, given by

$$h_k(n) = W_N^{-kn} \tag{8.56}$$

Comparing Equations 8.54 and 8.55, it is seen that the filter output when $n = N$ gives the DFT, $X(k)$, at the frequency bin, k:

$$X(k) = y_k(n)|_{n=N}$$

The z-plane transfer function of the filter is given by

$$\begin{aligned} H_k(z) &= \sum_{n=0}^{\infty} W_N^{-kn} z^{-n} \\ &= \frac{1}{1 - W_N^{-k} z^{-1}} \end{aligned} \tag{8.57}$$

This is a first-order filter with a single complex pole (as the twiddle factors are complex) on the unit circle at $z = W_N^{-k}$. To avoid complex arithmetic, the single poles are combined into complex conjugate pole pairs to give a second-order filter section. This may be achieved by multiplying Equation 8.57 by

$$\left(\frac{1 - W_N^k z^{-1}}{1 - W_N^k z^{-1}}\right)$$

to give

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

(ii) From Figure 8.40, the two-step difference equation for the Goertzel filter is

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

$$y_k(n) = v_k(n) - W_N^k v_k(n-1)$$

At $n = N$

$$\begin{aligned}
y_k(N) &= v_k(N) - W_N^k v_k(N-1) \\
&= v_k(N) - v_k(N-1)\left[\cos\left(\frac{2\pi k}{N}\right) - \mathrm{j}\sin\left(\frac{2\pi k}{N}\right)\right] \\
&= v_k(N) - v_k(N-1)\cos\left(\frac{2\pi k}{N}\right) + \mathrm{j}v_k(N-1)\sin\left(\frac{2\pi k}{N}\right)
\end{aligned}$$

$$\begin{aligned}
|y_k(N)|^2 &= (\textit{real part})^2 + (\textit{imaginary part})^2 \\
&= \left[v_k(N) - v_k(N-1)\cos\left(\frac{2\pi k}{N}\right)\right]^2 + \left[v_k(N-1)\sin\left(\frac{2\pi k}{N}\right)\right]^2 \\
&= v_k^2(N) - 2v_k(N)v_k(N-1)\cos\left(\frac{2\pi k}{N}\right) \\
&\quad + v_k^2(N-1)\cos^2\left(\frac{2\pi k}{N}\right) + v_k^2(N-1)\sin^2\left(\frac{2\pi k}{N}\right) \\
&= v_k^2(N) - 2v_k(N)v_k(N-1)\cos\left(\frac{2\pi k}{N}\right) \\
&\quad + v_k^2(N-1)\left[\cos^2\left(\frac{2\pi k}{N}\right) + \sin^2\left(\frac{2\pi k}{N}\right)\right] \\
&= v_k^2(N) + v_k^2(N-1) - 2\cos\left(\frac{2\pi k}{N}\right)v_k(N)v_k(N-1) \qquad (8.58)
\end{aligned}$$

(b) The DTMF tones for the digit '9' are 1477 Hz and 852 Hz. The detection of each tone requires a pair of Goertzel IIR filters. For the 1477 tone, the frequency bins are 38 and 74:

$$a_1 = 2\cos\left(\frac{2\pi \times 38}{205}\right) = 0.79; \quad a_2 = -1$$

$$a_1' = 2\cos\left(\frac{2\pi \times 74}{210}\right) = -1.1996; \quad a_2' = -1$$

For the 852 Hz tone, the coefficients are:

$$a_1 = 1.5623,\ a_2 = -1; \qquad a_1' = 0.5,\ a_2' = -1$$

8.19.3 Clock recovery for data communication

A fundamental problem in most digital data communication over long distances is that of generating a clock at the receiving end at the correct frequency and phase so that the data may be correctly decoded. The clock is normally derived from the received data.

Traditionally, analog circuits, for example using phase lock loops, are used to recover the clock, but these are susceptible to drift with age and temperature. Further, such circuits are unsuitable in applications that involve burst transmissions because of their slow response or where more than one data rate is involved (Smithson, 1992).

The input data stream is normally scrambled at the transmitting end (to provide clock information during idle periods) and then encoded, with each code representing a symbol. The codes are then transmitted at the so-called symbol rate. The problem at the receiving end is to recover the symbol clock.

The principles of symbol clock recovery using DSP are shown in Figure 8.41. The data stream is added modulo-2, that is exclusive ORed, with a version of itself delayed by one-half of a clock period to produce an output which contains level changes at the symbol rate (point C). The data is then applied to a marginally stable bandpass IIR filter. The impulse response of such a filter decays very slowly with time, producing a 'damped oscillation' at a frequency w_0, the centre frequency of the filter. The use of a marginally stable filter ensures that there is an output even when there are no transitions in the input data stream for reasonably long periods. The sampling frequency of the filter is chosen to be a multiple of the symbol rate. The desired symbol clock is derived from the output of the filter by zero-crossing detection (point E in Figure 8.41). For a 2's complement representation, this is trivially implemented by examining the signs of the data samples at the output of the digital filter.

A simple all-pole, IIR filter of the form shown in Figure 8.42 may be used for the symbol clock recovery. The filter is characterized by the following transfer function:

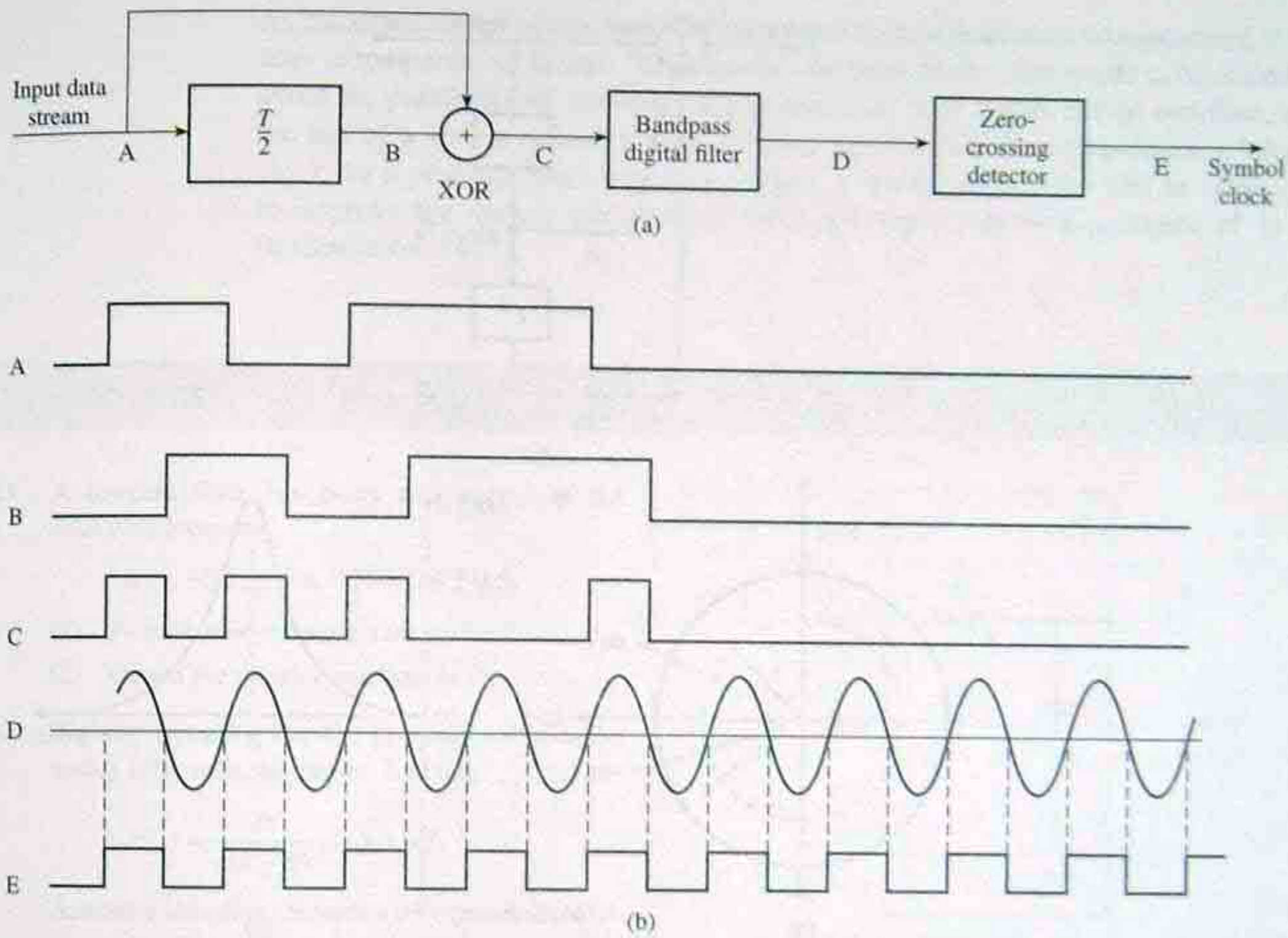


Figure 8.41 An illustration of the principles of symbol clock recovery for data communication.

$$H(z) = \frac{1}{[z - r\exp(-jw_0T)][z - r\exp(jw_0T)]}$$
$$= \frac{1}{z^2 - 2r\cos(w_0T)z - r^2}$$

where w_0 is the centre frequency of the bandpass filter, r is the radius of the pole and T is the reciprocal of the sampling frequency. w_0 is normally chosen to be identical or very close to the symbol clock frequency to be recovered, and the sampling frequency is a multiple of the centre frequency. The bandwidth of the filter is determined by the radius of the pole (see Equation 8.4). To ensure that the impulse response decays slowly with time the pole is normally located very close to the unit circle, typically in the range $0.99 < r < 1$. As discussed in Section 8.5.1 (Equation 8.5), the pole radius, r, and the bandwidth of the filter, bw, are related as

$$r \approx 1 - (\text{bw}/F_s)\pi$$

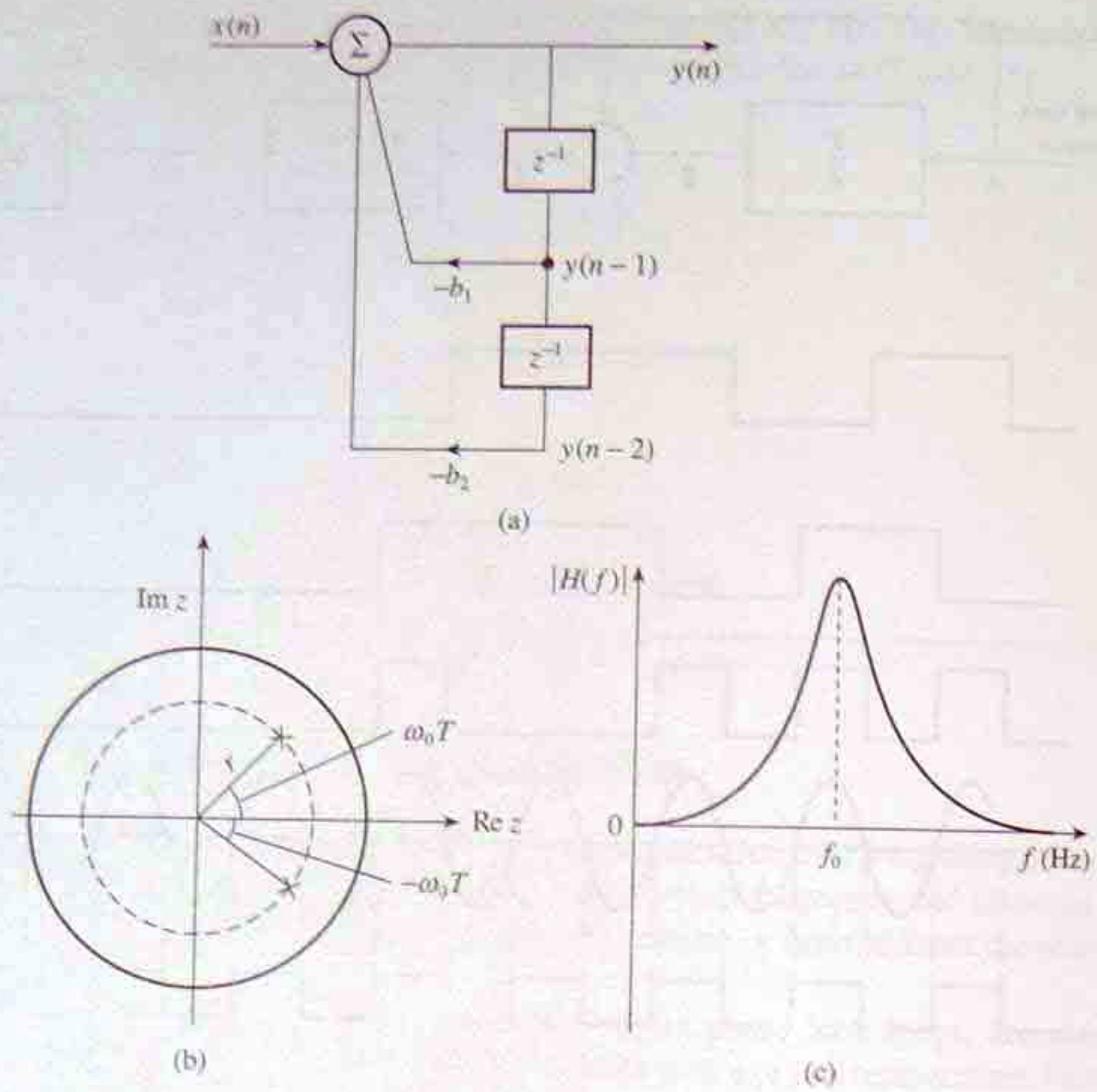


Figure 8.42 (a) Structure of the symbol clock recovery IIR filter, (b) pole diagram and (c) spectrum of filter.

where $F_s = 1/T$ is the sampling frequency of the filter.

For example, to recover the symbol clock for a hypothetical 4800 baud modem, suitable filter parameters are

data rate	4.8 kbaud
filter centre frequency, f_0	4.8 kHz
sampling frequency	153.6 kHz
bandwidth, bw	100 Hz

In this case, the pole radius (from the above equation), $r = 0.997\,954\,69$, and the pole angle $w_0T = 2\pi f_0T = (2\pi \times 4.8 \times 10^3/153.6 \times 10^3) = 0.196\,35$ rad $\approx 11.25°$. The resulting transfer function becomes

$$H(z) = \frac{1}{z^2 - 1.957\,558z + 0.995\,913}$$

As discussed earlier in the chapter, finite wordlength effects must be considered if the filter is to operate as desired. In particular, the input to the filter needs to be scaled to avoid the possibility of self-sustaining oscillations at its output due to overflow, and the use of a simple roundoff noise shaping scheme may help to produce a 'clean' clock. In a practical clock recovery system, a second filter stage will be necessary to improve the system performance when the input data is a sequence of 1s or 0s (Smithson, 1992).

Problems

8.1 A lowpass filter has poles and a zero at the following locations:

zero, −0.5; poles, 0.370, 0.6 ± 0.5j

(1) Plot the pole–zero diagram.

(2) Obtain the transfer function, $H(z)$.

8.2 Digitize, using the impulse invariant method, the analog filter with the transfer function

$$H(s) = \frac{\alpha}{s(s+\alpha)}, \quad \alpha = 0.5$$

Assume a sampling frequency of 1 (normalized).

8.3 A requirement exists to simulate in a digital computer an analog system with the following normalized characteristic:

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

Obtain a suitable transfer function using

(1) the impulse invariant method, and

(2) the bilinear transform method.

Assume a sampling frequency of 5 kHz and a 3 dB cutoff frequency of 1 kHz.

8.4 Determine, using the BZT method, the transfer function and difference equation for the digital equivalent of the resistance–capacitance (RC) filter shown in Figure 8.43. Assume a sampling frequency of 150 Hz and a cutoff frequency of 30 Hz.

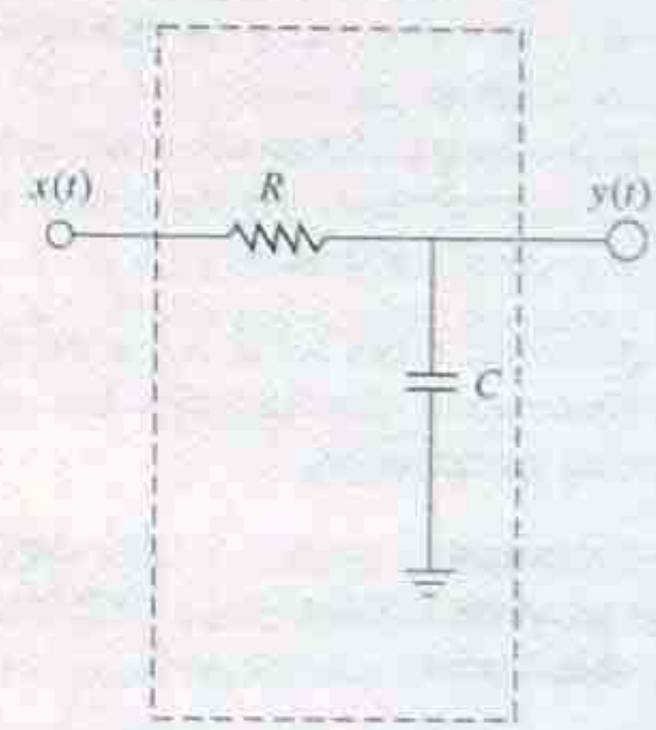


Figure 8.43

8.5 An IIR digital bandpass filter meeting the specifications given below is required. Starting with a suitable normalized analog LPF, (i) obtain the coefficients of $H(z)$, in factored form, using an appropriate band transformation and the BZT, and (ii) sketch the pole–zero diagrams for the analog filters and the resulting digital bandpass filter.

passband	8–10 kHz
sampling frequency	32 kHz
bandpass filter order	4
filter type	Butterworth

8.6 (a) Comment on the practical implications of the 'warping effect' associated with the bilinear z-transform (BZT) design method in applications where the response of the analog filter to be digitized is close to the Nyquist frequency. Give an example of a specific application.

(b) A requirement exists to design an IIR highpass filter, with a Butterworth characteristic, meeting the following specifications:

passband	2–4 kHz
stopband	0–500 Hz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	8 kHz

Determine:

(i) the pass and stopband edge frequencies for a suitable analog prototype lowpass filter. Your answer must include details of the basic concepts of how the prototype lowpass filter is obtained from the specifications;

(ii) the order, N, of the prototype lowpass filter;

(iii) the transfer function, and hence the coefficients of the IIR filter using the BZT method.

8.7 A requirement exists for a digital bandpass filter with Butterworth characteristics and meeting the following specifications:

passband	200–300 Hz
sampling frequency	2000 Hz
filter order	2

(a) (i) Starting with a suitable prototype analog lowpass filter, determine the transfer function of the digital filter using the bilinear z-transform method.

(ii) Explain, with the aid of sketches of pole–zero diagrams, how the poles and zeros of the prototype filter have been successively mapped from the s-plane to the z-plane.

(b) Assume the filter in (a) is to be implemented in fixed-point arithmetic. Assess the effects of quantizing the coefficients to 8 bits on the pole positions and hence the centre frequency.

Note. The lowpass-to-bandpass transformation is given by

$$s \rightarrow \frac{s^2 + \omega_0^2}{Bs}$$

where

$$\omega_0^2 = \omega_1' \omega_2'; \qquad B = \omega_2' - \omega_1'$$

ω_1', ω_2' and ω_0 are the lower and upper band-edge frequencies, and the centre frequency respectively.

8.8 Obtain suitable coefficients for an IIR digital filter using the bilinear transform method and an elliptic characteristic to meet the following specifications:

passband	4–12 kHz
stopbands	0–3.4 kHz, 12.6–16 kHz
passband ripples	<0.1 dB
stopband attenuation	>30 dB
sampling frequency	32 kHz

Determine a suitable coefficient wordlength to ensure that the filter is stable and the frequency response lies within the specified limits.

8.9 Design and implement in software a digital lowpass filter to meet the following specifications:

passband edge	2.5 kHz
stopband edge	3 kHz
passband deviation	<0.1 dB
stopband attenuation	>60 dB
sampling frequency	15 kHz

8.10 Obtain, via the bilinear transform, the coefficients of a digital filter that is maximally flat in the passband, 0 to 4 kHz, and has an attenuation of at least 25 dB at frequencies over 10 kHz. Assume a sampling frequency of 32 kHz.

8.11 The requirements for a certain lowpass filter are given below:

passband	0–30 Hz
stopband edge	50 Hz
stopband attenuation	>40 dB, at $f > 50$ Hz
sampling frequency	256 Hz

Assuming that the filter has a Butterworth response, determine via the bilinear transformation the transfer function, $H(z)$, of the filter. Obtain the realization diagram for the filter in cascade form using second- and/or first-order sections.

8.12 A bandpass digital filter with a Butterworth response is required for a certain real-time digital signal processing system. The filter is to satisfy the following requirements:

passband	0.3–3.4 kHz
stopband	0–0.2 kHz and 4–8 kHz
stopband attenuation	25 dB
sampling frequency	32 kHz

Obtain a suitable transfer function for the filter using the bilinear transform method.

8.13 A digital filter is required to remove baseline wander and artefacts due to body movement in a certain biomedical application. The filter is required to meet the following requirements:

passband	1–30 Hz
stopband	0–0.5 Hz and 40–128 Hz
passband ripple	<0.1 dB
stopband attenuation	>30 dB
sampling frequency	256 Hz

Determine the order of a suitable IIR filter and its transfer function $H(z)$.

8.14 A narrowband reject digital filter is required to remove an interfering signal. The filter should satisfy the following specifications:

passband edges	45 Hz and 55 Hz
passband ripple	< 0.1 dB
stopband attenuation	> 50 dB
sampling frequency	500 Hz

Obtain the coefficients of the filter.

8.15 Determine, using the impulse invariant method, the transfer function and difference equation for the digital equivalent of a single-pole RC lowpass filter. Assume a sampling frequency of 150 Hz and a 3 dB cutoff frequency of 30 Hz.

8.16 A standard second-order analog filter section, with simple poles, can be expressed as

$$\frac{A_0 + A_1 s}{B_0 + B_1 s + B_2 s^2} = \frac{C_1}{s + p_1} + \frac{C_2}{s + p_2}$$

where C_1 and C_2 are partial fraction coefficients and p_1 and p_2 are s-plane poles. Assume the impulse invariant transformation for the second-order section is given by

$$\frac{A_0 + A_1 s}{B_0 + B_1 s + B_2 s^2}$$

$$\rightarrow \frac{c_1 + c_2 - (c_1 e^{-p_2 T} + c_2 p^{-p_1 T})z^{-1}}{1 - (e^{-p_1 T} + e^{-p_2 T})z^{-1} + e^{-(p_1 + p_2)T} z^{-2}}$$

$$= \frac{a_0 - a_1 z^{-1}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

where T is the sampling interval.

(1) Find expressions for p_1, p_2, C_1 and C_2 in terms of A_0, A_1, B_0, B_1, and B_2.

(2) Obtain expressions for the coefficients a_0, a_1, b_1 and b_2 for the case when the poles are complex conjugates.

(3) Repeat part (2) when the poles are real and unequal.

(4) Given the following normalized analog transfer function, use your results to obtain the coefficients of an equivalent discrete-time filter. Assume a sampling frequency of 10 kHz and a cutoff frequency of 2 kHz.

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

8.17 Starting with a suitable analog Chebyshev LPF, obtain the transfer function of a digital bandstop filter to meet the following specifications:

stopband	10–15 kHz
sampling frequency	50 kHz
passband ripple	0.5 dB
filter order	6

8.18 Starting with a suitable analog Chebyshev LPF and the biquadratic transformation given below, determine, using the s-plane to z-plane pole–zero mapping approach, the transfer function of a digital bandpass filter to meet the following specifications:

passband	10–15 kHz
sampling frequency	50 kHz
passband ripple	0.5 dB
bandpass filter order	6

Alternatives to the BZT for the bandpass and bandstop filters are the following biquadratic transformations (Gold and Rader, 1969; Gray and Markel, 1976):

$$s = \cot\left[\frac{(\omega_2 - \omega_1)T}{2}\right]\left[\frac{z^2 - 2z\cos\gamma + 1}{z^2 - 1}\right]$$

lowpass to bandpass

$$s = \tan\left[\frac{(\omega_2 - \omega_1)T}{2}\right]\left[\frac{z^2 - 1}{z^2 - 2z\cos\gamma + 1}\right]$$

lowpass to bandstop

where

$$\cos\gamma = \cos\left[\frac{(\omega_2 + \omega_1)T}{2}\right] \Big/ \cos\left[\frac{(\omega_2 - \omega_1)T}{2}\right]$$

ω_1 and ω_2 are the lower and upper bandedge frequencies (passband edge frequencies for BPFs, stopband edge frequencies for BSFs) and γ is the centre frequency.

8.19 An analog lowpass filter is characterized by a pair of s-plane poles at

$$p_{1,2} = -1.4 \pm 1.2936j$$

It is desired to convert the filter into a digital bandpass filter with passband edges of 3 kHz and 5 kHz at a sampling frequency of 15 kHz. Given the digital lowpass to bandpass transformation and the BZT in Equation 8.40, determine

(1) the poles and zeros of the digital bandpass filter, and

(2) its transfer function in factored form.

8.20 A dual-tone multifrequency (DTMF) detection scheme for a digital telephone uses a series of second-order Goertzel filters to extract the DTMF tones and their second harmonics. Assume that the DTMF tones for digit '0' are 941 Hz and 1336 Hz. Determine the values of the coefficients in the feedback path of the Goertzel filters for the low frequency tone, 941 Hz, if the sequence lengths $N = 205$ and 210 are used for the fundamental and second harmonic, respectively, with corresponding discrete frequency bins of 24 and 47.

8.21 (a) Explain, with the aid of a diagram, the principles of a dual-tone multifrequency (DTMF) detection scheme for push-button telephones using the Goertzel algorithm.

(i) The discrete Fourier transform (DFT) of a data sequence, $x(n)$, $n = 0, 1, \ldots, N-1$, may be defined as

$$X(k) = \sum_{m=0}^{N-1} x(m)W_N^{km},$$

$$k = 0, 1, \ldots, N-1$$

where W_N^{km} is the twiddle factor.

(ii) Starting with the above equation, show that the z-plane transfer function, $H_k(z)$, for a Goertzel filter for DTMF tone detection can be expressed in the following recursive form:

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

(iii) Deduce an expression for the magnitude squared output of the Goertzel filter, $|y_k(n)|^2$, at the discrete time, $n = N$, and hence show that complex arithmetic is not required in the modified Goertzel algorithm.

(b) A DTMF tone detection scheme for a push-button telephone system is based on the specifications summarized in Table 8.3 and uses a second-order Goertzel filter.

Calculate the coefficients for the Goertzel filter to decode the digits at the receiver if the number dialled is '00'.

Compare the computational complexities of the Goertzel algorithm with the radix-2 FFT algorithm for DTMF tone detection, and hence justify the use of the Goertzel algorithm, instead of an FFT algorithm, for this application. State any reasonable assumptions made.

MATLAB problems

8.22 A requirement exists for a bandpass digital filter with a Chebyshev characteristic to meet the following specifications:

passband	1200–1800 Hz
stopband attenuation	>30 dB
passband ripple	<0.5 dB
transition width	400 Hz
sampling frequency	7.5 kHz

Using a suitable software program, obtain the filter coefficients.

8.23 An analog filter is to be converted into an equivalent digital filter that will operate at a sampling frequency of 256 Hz. Assume that the analog filter has the following transfer function:

$$H(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}$$

(1) Obtain suitable coefficients for the digital filter.

(2) Assuming that the digital filter is to be realized using the cascade structure, draw a suitable realization block diagram and develop the difference equations.

(3) Repeat part (2) for the parallel structure.

8.24 An IIR filter has the following transfer function:

$$H(z) = \frac{0.1436 + 0.2872z^{-1} + 0.1436z^{-2}}{1 - 1.8353z^{-1} + 0.9748z^{-2}}$$

(1) Determine the positions of the poles and zeros and sketch the pole–zero diagram.

(2) Determine the radial distance of the pole from the origin.

(3) Estimate the number of bits required to represent each coefficient

(a) to maintain stability, and

(b) so that the amplitude response in the passband does not change by more than 1%.

8.25 Compare the matched z-transform, the impulse invariant and the bilinear z-transform methods in terms of

(a) the Nyquist effect on the magnitude frequency, phase and group delay responses;

(b) the distribution of the pole–zero diagrams.

Use MATLAB and the following filters as vehicles in your study.

(1) Lowpass filter
An elliptic, lowpass digital filter with the following specifications:

passband	0–1 kHz
stopband	3–5 kHz
passband ripple	1 dB
stopband attenuation	60 dB
sampling frequency	10 kHz

(2) Highpass filter
An elliptic, highpass digital filter with the following specifications:

stopband	0–1 kHz
passband	3–5 kHz
passband ripple	1 dB
stopband attenuation	60 dB
sampling frequency	10 kHz

(3) Bandpass filter

(a) A Butterworth, bandpass digital filter with the following specifications:

passband	200–300 Hz
sampling frequency	2000 Hz
filter order	8

(b) A Butterworth, bandpass digital filter with the following specifications:

passband	800–900 Hz
sampling frequency	2000 Hz
filter order	8

(4) Bandstop filter
An elliptic, bandstop digital filter with the following specifications:

passbands	0–15 kHz, 30–50 kHz
stopband	20–25 kHz
passband ripple	0.2 dB
stopband attenuation	40 dB
sampling frequency	100 kHz

8.26 Compare the features of digital Butterworth, Chebyshev type I, Chebyshev type II and elliptic filters in terms of:

(a) the distribution of their poles and zeros;

(b) filter order;

(c) transition widths, pass and stopband ripples of the magnitude–frequency responses;

(d) phase and group delay responses.

Use the following filters in your study.

(1) Lowpass filter

passband	0–500 Hz
stopband	2–4 kHz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	8 kHz

(2) Highpass filter

stopband	0–500 Hz
passband	2–4 kHz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	8 kHz

(3) Bandpass filters

(a)

lower passband edge frequency	250 Hz
upper passband edge frequency	300 Hz
lower stopband edge frequency	50 Hz
upper stopband edge frequency	450 Hz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	1 kHz

(b)

passbands	0–15 kHz, 30–50 kHz
stopband	20–25 kHz
passband ripple	0.2 dB
stopband attenuation	40 dB
sampling frequency	100 kHz

(4) Bandstop filters

(a)

passbands	0–50 Hz, 450–500 Hz
stopband	250–300 Hz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	1 kHz

(b)

passbands	0–15 kHz, 30–50 kHz
stopband	20–25 kHz
passband ripple	0.2 dB
stopband attenuation	40 dB
sampling frequency	100 kHz

8.27 (a) (i) Determine, using the BZT method and MATLAB, the transfer function of a discrete bell filter for audio signal processing in a digital mixer when the control settings correspond to a Q-factor of 2, a boots (peak) of 6.02 dB at 10 kHz. Assume a sampling frequency of 48 kHz, and that the equivalent analog filter has an s-plane transfer function given below.

Plot the magnitude, phase and group delay responses and the pole–zero diagram of the discrete filter.

(ii) Repeat the above problem at a sampling frequency of 96 kHz.

(b) Repeat (a) (i) and (ii) using the MZT method.

(c) Repeat (a) (i) and (ii) using the impulse invariant method.

(d) Compare the results in (a) to (c).

$$H(s) = \frac{s^2 + (3+k)\dfrac{\omega_0}{Q}s + \omega_0^2}{s^2 + (3-k)\dfrac{\omega_0}{Q}s + \omega_0^2}$$

where

$$k = 3\left(\frac{G-1}{G+1}\right); \quad \omega_0 = \text{boost frequency}$$

$$G = \text{gain factor}; \quad Q = \text{Q-factor}.$$

8.28 (a) Develop a C-language pseudo-code for the detection of DTMF tones and their second-order harmonics. State any assumptions made.

(b) Repeat part (a) using MATLAB.

(c) Generate suitable frequency tones, using MATLAB, and use these to test your MATLAB program for detecting DTMF tones.

References

Abu-el-Haija A. and Al-Ibrahim M.M. (1986) Improving performance of digital sinusoidal oscillators by means of error feedback circuits. *IEEE Trans. Circuits and Systems*, **33**(4), 373–80.

Antoniou A. (1979) *Digital Filters Analysis and Design*. New York: McGraw-Hill.

Chen C.J. (1996) *Modified Goertzel Algorithm in DTMF Detection using the TMS320C80*, Texas Instruments, Application Report SPRA066, June.

Clark R.J., Ifeachor E.C. and Rogers G.M. (1996) Real-time equaliser coefficient realisation with minimised computational load and distortion. Preprint 4360, 101st Audio Engineering Society Convention.

Clark R.J., Ifeachor E.C., Rogers G.M. and Van Eetvelt P.W.J. (2000) Techniques for generating digital equaliser coefficients. *J. Audio Engineering Society*, **48**(4), 281–98.

Dattorro J. (1988) The implementation of recursive digital filters for high-fidelity audio. *J. Audio Engineering Society*, **36**(11), 851–78.

DeFatta D.J., Lucas J.G. and Hodgkiss W.S. (1988) *Digital Signal Processing*. New York: Wiley.

Feeney S.L., Kieburtz R.B., Mina K.V. and Tewksbury S.K. (1971) Design of digital filters for an all digital frequency division multiplex–time division multiplex translator. *IEEE Trans. Circuit Theory*, **18**, 702–11.

Gold B. and Rader C.M. (1969) *Digital Processing of Signals*. New York: McGraw-Hill.

Gray A.H. and Markel J.D. (1976) A computer program for designing elliptic filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**(6), 529–38.

IEEE (1979) *Programs for Digital Signal Processing*. New York: IEEE Press.

Jackson L.B. (1986) *Digital Filters and Signal Processing*. Boston MA: Kluwer.

Jackson L.B., Kaiser J.F. and McDonald H.S. (1968) An approach to the implementation of digital filters. *IEEE Trans. Audio and Electroacoustics*, **16**(3), 413–21.

Jong M.T. (1982) *Methods of Discrete Signal System Analysis*. New York: McGraw-Hill.

Lynn P.A. and Fuerst W. (1989) *Introductory Digital Signal Processing with Computer Applications*. Chichester: Wiley.

Marven C. (1990) General-purpose tone decoding and DTMF detection, in *Theory, Algorithms, and Implementations, Digital Signal Processing Applications with the TMS320 Family*, Vol. 2, literature number SPRA016, Texas Instruments.

Mock P. (1985) Add DTMF generation and decoding to DSP-μp designs. *EDN*, **30**.

Parks T.W. and Burrus C.S. (1987) *Digital Filter Design*. New York: Wiley.

Rabiner L.R. and Gold B. (1975) *Theory and Applications of Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Rader C.M. and Gold B. (1967) Effects of parameter quantization on the poles of a digital filter. *Proc. IEEE*, **55**, 688–9.

Smithson P. (1992) Clock recovery for a satellite data modem, University of Plymouth (personal communication).

Stanley W.D., Dougherty G.R. and Dougherty R. (1984) *Digital Signal Processing*, 2nd edn. Reston VA: Reston Publishing, Inc.

Stearns S.D. and Hush D.R. (1990) *Digital Signal Analysis*, 2nd edn. Englewood Cliffs NJ: Prentice-Hall.

Bibliography

Abu-el-Haija A.I. and Peterson A.M. (1979) An approach to eliminate roundoff errors in digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **27**, 195–8.

Ahmed N. and Natarajan T. (1983) *Discrete-time Signals and Systems*. Reston VA: Reston Publishing, Inc.

Allen J. (1975) Computer architecture for signal processing. *Proc. IEEE*, **63**(4), 624–48.

Arjmand M. and Roberts R.A. (1981) On comparing hardware implementations of fixed point digital filters. *IEEE Circuits Systems Mag.*, **3**(2), 2–8.

Avenhaus E. (1972) Filters with coefficients of limited wordlength. *IEEE Trans. Audio Electroacoustics*, **20**, 206–12.

Barnes C.W., Tran B.N. and Leung S.H. (1985) On the statistics of fixed-point roundoff error. *IEEE Trans. Acoustics, Speech and Signal Processing*, **33**, 595–606.

Bellanger M. (1984) *Digital Processing of Signals. Theory and Practice*. New York: Wiley.

Chang T.L. (1978) A low roundoff noise digital filter structure. In *Proc. IEEE Int. Symp. on Circuits and Systems*, May 1978, pp. 1004–8.

Chang T.L. (1979) Error-feedback digital filters. *Electronics Lett.*, **15**, 348–9.

Chang T.L. (1980) Comments on 'An approach to eliminate roundoff errors in digital filters'. *IEEE Trans. Acoustics, Speech and Signal Processing*, **28**(2), 244–5.

Chang T.L. (1981) Suppression of limit cycles in digital filters designed with one magnitude-truncation quantizer. *IEEE Trans. Circuits and Systems*, **28**(2), 107–11.

Chang T.L. (1981) On low-roundoff noise and low-sensitivity digital filter structures. *IEEE Trans. Acoustics, Speech and Signal Processing*, **29**(5), 1077–80.

Chang T.L. and White S.A. (1981) An error cancellation digital-filter structure and its distributed-arithmetic implementation. *IEEE Trans. Circuits and Systems*, **28**(4), 339–42.

Charalambous C. and Best M.J. (1974) Optimization of recursive digital filters with finite wordlengths. *IEEE Trans. Acoustics, Speech and Signal Processing*, **22**(6), 424–31.

Chassaing R. and Horning D.W. (1990) *Digital Signal Processing with the TMS320C25*. New York: Wiley.

Claasen T.A.C.M. and Kristiansson L.O.G. (1975) Necessary and sufficient conditions for the absence of overflow phenomena in a second order recursive digital filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(6), 509–15.

Claasen T.A.C.M., Mecklenbrauker W.F.G. and Peek J.B.H. (1973) Second-order digital filter with only one magnitude-truncation quantiser and having practically no limit cycles. *Electronics Lett.*, **9**, 531–2.

Claasen T.A.C.M., Mecklenbrauker W.F.G. and Peek J.B.H. (1973) Some remarks on the classification of limit cycles in digital filters. *Philips Research Rep.*, **28**, 297–305.

Claasen T., Mecklenbrauker W.F.G. and Peek J.B.H. (1975) Frequency domain criteria for the absence of zero-input limit cycles in nonlinear discrete-time systems, with applications to digital filters. *IEEE Trans. Circuits and Systems*, **22**, 232–9.

Claasen T.A.C.M., Mecklenbrauker W.F.G. and Peek J.B.H. (1976) Effects of quantization and overflow in recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**(6), 517–28.

Clark R.J., Ifeachor E.C., Rogers G.M. and Van Eetvelt P.W.J. (2000) Techniques for generating digital equaliser coefficients. *Journal of Audio Engineering Society*, **48**(4), 281–98.

Crochiere R.E. (1975) A new statistical approach to the coefficient wordlength problem for digital filters. *IEEE Trans. Circuits and Systems*, **22**, 190–6.

Crochiere R.E. and Oppenheim A.V. (1975) Analysis of linear digital networks. *Proc. IEEE*, **63**(4), 581–94.

Diniz P.S.R. and Antoniou A. (1985) Low-sensitivity digital filter structures which are amenable to error-spectrum shaping. *IEEE Trans. Circuits and Systems*, **32**(10), 1000–7.

Elliot D.F. (ed.) (1987) *Handbook of Digital Signal Processing*. London: Academic Press.

IEEE (1978) Digital Signal Processing II. Institute of Electrical and Electronics Engineers.

Jackson L.B. (1970) On the interaction of roundoff noise and dynamic range in digital filters. *BSTJ*, **49**(2), 159–84.

Jackson L.B. (1976) Roundoff noise bounds derived from coefficient sensitivities for digital filters. *IEEE Trans. Circuits and Systems*, **23**(8), 481–5.

Knowles J.B. and Olcayto E.M. (1968) Coefficient accuracy and digital filter response. *IEEE Trans. Circuit Theory*, **15**, 31–41.

Liu B. (1971) Effect of finite wordlength on the accuracy of digital filters – a review. *IEEE Trans. Circuit Theory*, **18**, 670–7.

Liu B. and Kaneko T. (1969) Error analysis of digital filters realized with floating-point arithmetic. *Proc. IEEE*, **57**(10), 1735–47.

Markel J.D. and Gray A.H. (1975) Fixed-point implementation algorithms for a class of orthogonal polynomial filter structures. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(5), 486–94.

Markel J.D. and Gray A.H. (1975) Roundoff noise characteristics of a class of orthogonal polynomial structures. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(5), 473–86.

Motorola (1988) *Digital Stereo 10-band Graphic Equalizer Using the DSP56001*. Motorola Application Note.

Mullis C.T. and Roberts R.A. (1976) Round-off noise in digital filters: frequency transformations and invariants. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**(6), 538–50.

Munson D.C. and Liu B. (1980) Low-noise realization for narrow-band recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **28**, 41–54.

Nagle H.T. and Nelson V.P. (1981) Digital filter implementation on 16 bit microcomputers. *IEEE Micro*, **1**, 23–41.

Oppenheim A.V. and Schafer R.W. (1975) *Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Oppenheim A.V. and Weinstein, C.J. (1972) Effects of finite register length in digital filtering and the fast Fourier transform. *Proc. IEEE*, **60**, 957–76.

Peled A., Liu B. and Steiglitz K. (1974) A new hardware realization of digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **22**, 456–62.

Peled A., Liu B. and Steiglitz K. (1975) A note on implementation of digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**, 387–9.

Rabiner L.R., Cooley J.W., Helms H.D., Jackson L.B., Kaiser J.F., Rader C.M., Schafer R.W., Steiglitz K. and Weinstein C.J. (1972) Terminology in digital signal processing. *IEEE Trans. Audio and Electroacoustics*, **20**, 322–37.

Sandberg I.W. and Kaiser J.F. (1972) A bound on limit cycles in fixed-point implementations of digital filters. *IEEE Trans. Audio and Electroacoustics*, **20**, 110–12.

Schmalzel J.L., Heine D.N. and Ahmed N. (1980) Some pedagogical considerations of digital filter hardware implementation. *IEEE Circuits and Systems Mag.*, **2**(1), 4–13.

Sim P.K. and Pang K.K. (1985) Effects of input-scaling on the asymptotic overflow-stability properties of second recursive digital filters. *IEEE Trans. Circuits and Systems*, **32**(10), 1008–15.

Steiglitz K. (1971) Designing short-word recursive digital. *Proc. 9th Ann. Allerton Conf. on Circuit and System Theory*, 6–8 October, 1971, pp. 778–88.

Steiglitz K., Bede L. and Liu B. (1976) An improved algorithm for ordering poles and zeros of fixed-point recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**, 341–3.

Taylor F.J. (1983) *Digital Filter Design Handbook*. New York: Marcel Dekker.

Thong T. (1976) Finite wordlength effects in the ROM digital filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**, 436–7.

Thong T. and Liu B. (1977) Error spectrum shaping in narrowband recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **25**, 200–3.

Williamson D. and Sridharan S. (1985) An approach to coefficient wordlength reduction in digital filters. *IEEE Trans. Circuits and Systems*, **32**(9), 893–903.

Appendices

8A C programs for IIR digital filter design

Several C language programs for designing IIR digital filters have been developed for the book. Because of lack of space, only the program for the impulse invariant method of coefficient calculation is listed here. The following programs are not listed in the book, but are available on the CD in the companion handbook. MATLAB m-files that may be used to perform similar tasks to some of the C programs are available on the web (see the Preface for details).

- bilinear transformation;
- coefficient calculation for classical IIR filters (Butterworth, Chebyshev and elliptic) via the bilinear transform method;
- finite wordlength analysis for IIR filters.

8A.1 C program for the impulse invariant method

Please note that in the C programs the coefficients A_i and B_i for the analog transfer functions and a_i and b_i for the discrete-time filters are the numerator and denominator coefficient vectors, respectively. Unlike the programs, we have modified the use of these coefficients in Equations 8A.1 to 8A.8 to be consistent with those of MATLAB.

The program for the impulse invariant method is listed in Program 8A.1. First, we will summarize the concepts on which the program is based and then illustrate its use by an example. The discussion will be based on the second-order filter section, since this is the basic building block for digital IIR filters.

Consider a general second-order s-plane transfer function which is to be converted into a discrete transfer function:

$$H(s) = \frac{B_0 + B_1 s}{A_0 + A_1 s + A_2 s^2} \tag{8A.1}$$

The equivalent z-transfer function is also a second-order section of the form

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{8A.2}$$

Program 8A.1 Impulse invariant method (*Note*: In the program the coefficients A's and B's, and a's and b's, are reversed compared to the equations above.).

```
/*-----------------------------------------------------------------------*
 *      Impulse invariant method                                         *
 *                                                                       *
 *      The analog transfer function must be frequency-scaled            *
 *      (normalized frequency) before using program                      *
 *      30.10.92                                                         *
 *-----------------------------------------------------------------------*
 */
#include    <stdio.h>
#include    <math.h>
#include    <dos.h>

void        dfilter();
double      T;
double      a0, a1, a2, b0, b1, b2;
double      p1, p2, pr, pi;
double      c1, c2, cr, ci;
float       A0, A1, B0, B1, B2, temp;

main()
{

                                                    /* initialize coeffs */
            A0=0; A1=0; B0=1; B1=0; B2=0;
            a0=0; a1=0; b0=1; b1=0; b2=0;
            c1=0; c2=0; p1=0; p2=0; a2=0;

                                                    /* read s-plane coefficients */
```

```
        printf("impulse invariant discrete filters \n");
        printf("\n");
        printf("enter s-plane coefficients \n");
        printf("enter denominator coeffs: B0, B1, B2 \n");
        scanf("%f %f %f", &B0, &B1, &B2);
        printf("enter numerator coeffs: A0, A1 \n");
        scanf("%f %f", &A0, &A1);
        T=1;
        dfilter();
        printf("\n");
        printf("press enter to continue\n");
        getch();
        exit(0);
}
/*-----------------------------------------------------------------------------------------------*/
void        dfilter()
{
/* Find the s-plane pole positions */
        temp = B1*B1 - 4*B0*B2;

        if(B2==0){                                          /* a single pole */
                p1=-B0/B1;
                a0=A0/B1;
                b1=-exp(p1*T);
        }
        if(temp>0){                                         /* real and unequal poles */
                pr=-B1/(2*B2);
                pi=(pr*pr)-B0/B2;
                pi=sqrt(pi);
                p1=pr+pi;
                p2=pr-pi;
                c1=(A0+A1*p1)/((p1-p2)*B2);
                c2=A1/B2-c1;
                a0=c1+c2;
                a1=-(c1*exp(p2*T) + c2*exp(p1*T));
                b1=-exp(p1*T)-exp(p2*T);
                b2=exp((p1+p2)*T);
        }
        if(temp<0){                                         /* complex conjugate poles */
                pr=-B1/(2*B2);
                pi=(pr*pr)-B0/B2;
                pi=sqrt(-pi);
                cr=A1/(B2*2);
                ci=-(A0+A1*pr)/(2*pi*B2);
                a0=2*cr;
                a1=-(cr*cos(pi*T)+ci*sin(pi*T))*2*exp(pr*T);
                b1=-2*exp(pr*T)*cos(pi*T);
                b2=exp(2*pr*T);
        }
        printf("discrete filter coeffs: \n");
        printf("a0 a1 a2: \t%f %f %f \n", a0, a1, a2);
        printf("b0 b1 b2: \t%f %f %f \n", b0, b1, b2);
}
```

Given the values of the coefficients of the analog transfer function, $H(s)$, the C program listed in Program 8A.1 computes the coefficients of the equivalent z-transfer function, $H(z)$. To see how the program works, we will establish the relationship between the coefficients of $H(s)$ and those of $H(z)$.

Using a partial fraction expansion, the s-plane transfer function of Equation 8A.1 can be expressed as

$$\frac{B_0/A_2 + (B_1/A_2)s}{A_0/A_2 + (A_1/A_2)s + s^2} = \frac{c_1}{s - p_1} + \frac{c_2}{s - p_2} \tag{8A.3}$$

where p_1 and p_2 are the s-plane poles of $H(s)$ given by

$$p_{1,2} = \frac{-A_1}{2A_2} \pm \left[\left(\frac{A_1}{2A_2}\right)^2 - \frac{A_0}{A_2}\right]^{1/2} \tag{8A.4}$$

Multiplying both sides of Equation 8A.3 by $(s - p_1)(s - p_2)$ and equating coefficients of s and the constant terms we have

$$\frac{B_0}{A_2} = -(c_1 p_2 + c_2 p_1) \tag{8A.5a}$$

$$\frac{B_1}{A_2} = c_1 + c_2 \tag{8A.5b}$$

Solving for c_1 and c_2 we have

$$c_1 = \frac{B_0 + B_1 p_1}{(p_1 - p_2)A_2} \tag{8A.6a}$$

$$c_2 = \frac{B_1}{A_2} - c_1 \tag{8A.6b}$$

Applying the impulse invariant transformation to Equation 8A.3 gives the discrete transfer function, $H(z)$:

$$\begin{aligned} H(z) &= \frac{c_1 + c_2 - (c_1 e^{p_2 T} + c_2 e^{p_1 T})z^{-1}}{1 - (e^{p_1 T} + e^{p_2 T})z^{-1} + e^{(p_1 + p_2)T} z^{-2}} \\ &= \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \end{aligned} \tag{8A.7}$$

where

$$b_0 = c_1 + c_2, \qquad b_1 = -(c_1 e^{p_2 T} + c_2 e^{p_1 T})$$
$$a_1 = -(e^{p_1 T} + e^{p_2 T}), \qquad a_2 = e^{(p_1 + p_2)T}$$

p_1 and p_2 are defined in Equation 8A.4, and c_1 and c_2 are defined in Equation 8A.6.

Thus, given the s-plane coefficients for a second-order filter (that is A_0, A_1, A_2, B_0 and B_1), the coefficients of the equivalent discrete filter can be obtained directly using the relationships above. Evaluation of the coefficients of $H(z)$ in Equation 8A.7 depends on the type of the s-plane poles, p_1 and p_2. In practice, three cases arise: these are when the two poles are (i) real

and unequal, (ii) a complex conjugate pair, or (iii) real and equal (that is, coincident poles). Only the first two cases are considered as the third does not occur often and is more involved.

In the first case, Equation 8A.7 can be used directly to obtain the coefficients of $H(z)$. In the second case, a simpler form of Equation 8A.7 is used to avoid complex arithmetic. Exploiting the properties of the poles, Equation 8A.7 (for the second case) becomes

$$H(z) = \frac{(c_1 + c_1^*) - (c_1 e^{p_1^* T} + c_1^* e^{p_1 T}) z^{-1}}{1 - (e^{p_1 T} + e^{p_1^* T}) z^{-1} + e^{(p_1 + p_1^* T)} z^{-2}}$$

$$= \frac{2c_r - [c_r \cos(p_i T) + c_i \sin(p_i T)] 2e^{p_r T} z^{-1}}{1 - 2e^{p_r T} \cos(p_i T) z^{-1} + e^{2p_r T} z^{-2}} \tag{8A.8}$$

where p_r is the real part of p_1, p_i the imaginary part of p_1, c_r the real part of c_1 and c_i the imaginary part of c_1. From Equation 8A.4, the real and imaginary parts of p_1 are given by

$$p_r = -\frac{A_1}{2A_2}, \quad p_i = \left\{ -\left[\left(\frac{A_1}{2A_2} \right)^2 - \frac{A_0}{A_2} \right] \right\}^{1/2}$$

and, from Equation 8A.6, the partial fraction coefficient, c_1, is given by

$$c_1 = \frac{B_1}{2A_2} - \frac{B_0 + B_1 p_r}{2p_i A_2} \mathrm{j} = c_r + c_i \mathrm{j}$$

Thus, in standard form, the coefficients of the second-order z-transfer function for the case where the poles of $H(s)$ are complex conjugates are

$$b_0 = 2c_r, \qquad b_1 = -[c_r \cos(p_i T) + c_i \sin(p_i T)] 2e^{p_r T}$$

$$a_1 = -2e^{p_r T} \cos(p_i T), \qquad a_2 = e^{2p_r T}$$

Example 8A.1

We will use the C program in Program 8A.1 to compute the coefficients for the discrete filter of Example 8.4.

The program expects the frequencies to be normalized. With a sampling frequency of 1280 Hz and a cutoff frequency of 150 Hz, the normalized cutoff frequency is 150/1280. The transfer function is first frequency scaled by replacing s by s/α where $\alpha = 2\pi \times 150/1280 = 0.736\,31$:

$$H'(s) = \frac{\alpha^2}{\alpha^2 + \sqrt{2}\alpha s + s^2} = \frac{1}{1 + (\sqrt{2}/\alpha)s + (1/\alpha^2)s^2} = \frac{1}{1 + 1.920\,675s + 1.844\,96s^2}$$

The prompts and outputs of the program are given below. The discrete coefficients are identical to the values obtained in Example 8.4.

```
impulse invariant discrete filters
enter s-plane coefficients
enter denominator coeffs: A0, A1, A2
1   1.920675   1.84496
enter numerator coeffs: B0, B1
1   0
discrete filter coeffs:
b0   b1   b2:   0.000000   0.307718   0.000000
a0   a1   a2:   1.000000   -1.030953   0.353088
```

From the above listings, the z-transfer function can be written down directly:

$$H(z) = \frac{0.307718z^{-1}}{1 - 1.030953z^{-1} + 0.353088z^{-2}}$$

The coefficients are identical to the values obtained in Example 8.4.

8B IIR filter design with MATLAB

The MATLAB Signal Processing Toolbox provides many useful functions for the design and analysis of classical digital IIR filters (e.g. Butterworth, Chebyshev types I and II, and elliptic filters) for a given set of specifications (e.g. pass- and stopband edge frequencies, passband ripples and stopband attenuation).

In particular, the Toolbox provides functions for converting classical analog filters into equivalent discrete-time filters as discussed in the main text.

As the reader may recall, a crucial stage in the design of digital IIR filters is coefficient calculation. For classical digital IIR filters, the steps involved at this stage may be summarized as follows:

(1) Specify the desired filter.

(2) Determine a suitable analog prototype lowpass filter with, e.g. Butterworth, Chebyshev type I, Chebyshev type II or elliptic characteristics.

(3) Transform the prototype analog filter to create a lowpass, highpass, bandpass or bandstop filter.

(4) Convert the transformed filter into an equivalent discrete-time filter (for example, using the impulse invariant or the bilinear z-transform method).

The MATLAB Signal Processing Toolbox provides a number of high-level functions for carrying out steps 2 to 4 simultaneously or separately. For example, the syntax for the MATLAB command to create a lowpass, highpass, bandpass or bandstop filter with a Butterworth characteristic is (steps 1 and 2):

```
[b, a] = butter (N, Wc, options)
[z, p, k] = butter (N, Wc, options)
```

The first command calculates the numerator and denominator coefficients of an Nth order discrete-time Butterworth filter with a 3 dB cutoff frequency (or bandedge frequency), W_c, normalized to the Nyquist frequency. The numerator and denominator coefficients of the filter are returned in the vectors b and a, respectively, in ascending negative powers of z.

If the word 'options' is left out, the command defaults to a lowpass filter (unless W_c is a vector of frequencies, in which case it defaults to a bandpass filter). For highpass and bandstop filters the words 'high' and 'stop' are used as the options. For bandpass and bandstop filters, W_c is a two-element vector that specifies the cutoff (or bandedge) frequencies:

$$W_c = [\omega_{c1}, \omega_{c2}]$$

where $\omega_{c1} < \omega < \omega_{c2}$ is the passband (bandpass filters) or stopband (bandstop filters).

The second command returns the positions of the zeros and poles, in rectangular form, in z and p, and the filter gain, k.

Similar commands exist for other classical filter types. For example, for Chebyshev types I and II, and elliptic filters, the syntax for the MATLAB commands is

```
[b, a] = cheby1 (N, Ap, Wc, options)
[z, p, k] = cheby1 (N, Ap, Wc, options)

[b, a] = cheby2 (N, As, Wc, options)
[z, p, k] = cheby2 (N, As, Wc, options)

[b, a] = ellip (N, Ap, As, Wc, options)
[z, p, k] = ellip (N, Ap, As, Wc, options)
```

where Ap and As are, respectively, the passband ripple and stopband attenuation, in dB.

There are a number of other useful commands that may be used to carry out intermediate tasks in the coefficient calculation process. For example, the **buttord**, **cheby1ord**, and **ellipord** commands may be used to determine the order of a suitable filter. The parameters of suitable prototype analog lowpass filters may be determined using the **butterp**, **cheby1p**, **cheby2p** and **ellipp** commands.

Example 8B.1 *Design of a simple lowpass filter using impulse invariant method and MATLAB* A lowpass IIR filter with a Butterworth characteristic is required to meet the following specifications:

cutoff frequency	150 Hz
sampling frequency	1.28 kHz
filter order, N	2

(a) Determine, using the impulse invariant method and MATLAB,
  (i) the coefficients, poles and zeros of the frequency-scaled analog filter;
  (ii) the coefficients, poles and zeros of the IIR discrete filter. Write down its transfer function.

(b) Plot the magnitude–frequency response and the pole–zero diagram of the discrete filter.

Solution The MATLAB m-file for the problem is given in Program 8B.1.

(a) (i) Using the m-file, the coefficients, poles and zeros of the analog filter are

Coefficients:	b = 1.0e+005*[0, 0, 8.8826] a = 1.0e+005*[0.00001, 0.0133, 8.8826]
Poles:	1.0e+002*(−6.6643 ± 6.6643j)
Zeros:	None
Gain	8.8826e+005

Program 8B.1 MATLAB m-file for Example 8B.1.

```
%
% Program name: EX8B1.m
% A simple lowpass filter
%
N=2;                                  % Filter order
Fs=1280;                              % Sampling frequency
fc=150;                               % Cutoff frequency
WC=2*pi*fc;                           % Cutoff frequency in radians
[b, a]=butter (N,WC,'s');             % Create the analog filter
[z, p, k]=butter (N, WC, 's');
[bz, az]=impinvar (b, a, Fs);         % Convert into discrete filter
subplot (2,1,1)                       % Plot magnitude freq. response
[H, f]=freqz(bz, az, 512, Fs);
plot(f, 20*log10(abs(H)))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response (dB)')
subplot(2,1,2)                        % Plot pole-zero diagram
zplane(bz, az)
zz=roots(bz);                         % Determine poles and zeros
pz=roots(az);
```

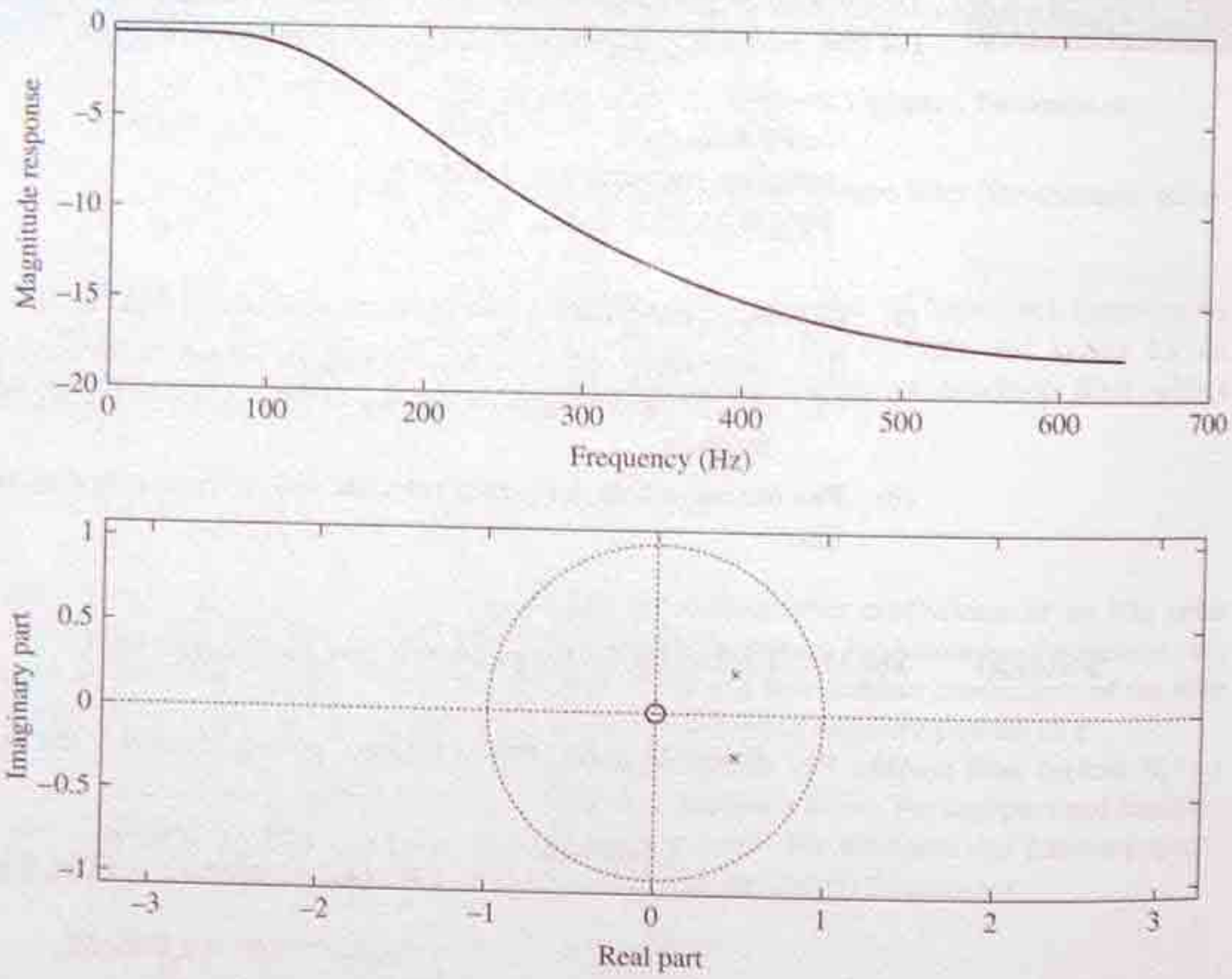


Figure 8B.1

(ii) Using the m-file, the coefficients, poles and zeros of the discrete-time filter are

Coefficients:	b = [0 0.3078 0] a = [1.0000 −1.0308 0.3530]
Poles:	0.5154 ± 0.2955j
Zero:	0

Using the coefficients, the transfer function of the filter is given by

$$H(z) = \frac{0.3078z^{-1}}{1 - 1.0308z^{-1} + 0.3553z^{-2}}$$

The results are the same as in Example 8.4.

(b) The magnitude–frequency response and the pole–zero diagram are depicted in Figure 8B.1.

Example 8B.2 *Design of a simple lowpass filter using the bilinear z-transform method and MATLAB* A digital IIR filter with Butterworth characteristics is required to meet the following specifications:

cutoff frequency	150 Hz
sampling frequency	1.28 kHz
filter order, N	2

(a) Determine, using the bilinear z-transform method and MATLAB, the coefficients, poles and zeros of the discrete filter.

(b) Plot the magnitude–frequency response and the pole–zero diagram of the discrete filter.

Solution The MATLAB m-file is listed in Program 8B.2.

(a) With the aid of the m-file, the coefficient vectors (b and a), zeros and poles (z and p) of the IIR filter are

```
b = [0.0878, 0.1756, 0.0878]
a = [1.0000, −1.0048, 0.3561]
z = [−1, −1]
p = [0.5024 ± 0.3220j]
k = 0.0878
```

Using the coefficients, the transfer function of the filter is given by

$$H(z) = \frac{0.0878(1 + 2z^{-1} + z^{-2})}{1 - 1.0308z^{-1} + 0.3553z^{-2}}$$

(b) The magnitude–frequency response and the pole–zero diagram are depicted in Figure 8B.2.

Program 8B.2 MATLAB m-file for Example 8B.2.

```
%
% Program name: EX8B2.m
% A simple lowpass filter
%
N=2;                                  % Filter order
Fs=1280;                              % Sampling frequency
FN=Fs/2;
fc=150;                               % Cutoff frequency
Fc=fc/FN;                             % Normalized Cutoff frequency
[b,a]=butter (N,Fc);                  % Create and digitize analog filter.
[z,p,k]=butter (N, Fc);
subplot(2,1,1)                        % Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot(f, abs(H))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response ')
subplot(2,1,2)                        % Plot pole-zero diagram
zplane(b, a)
```

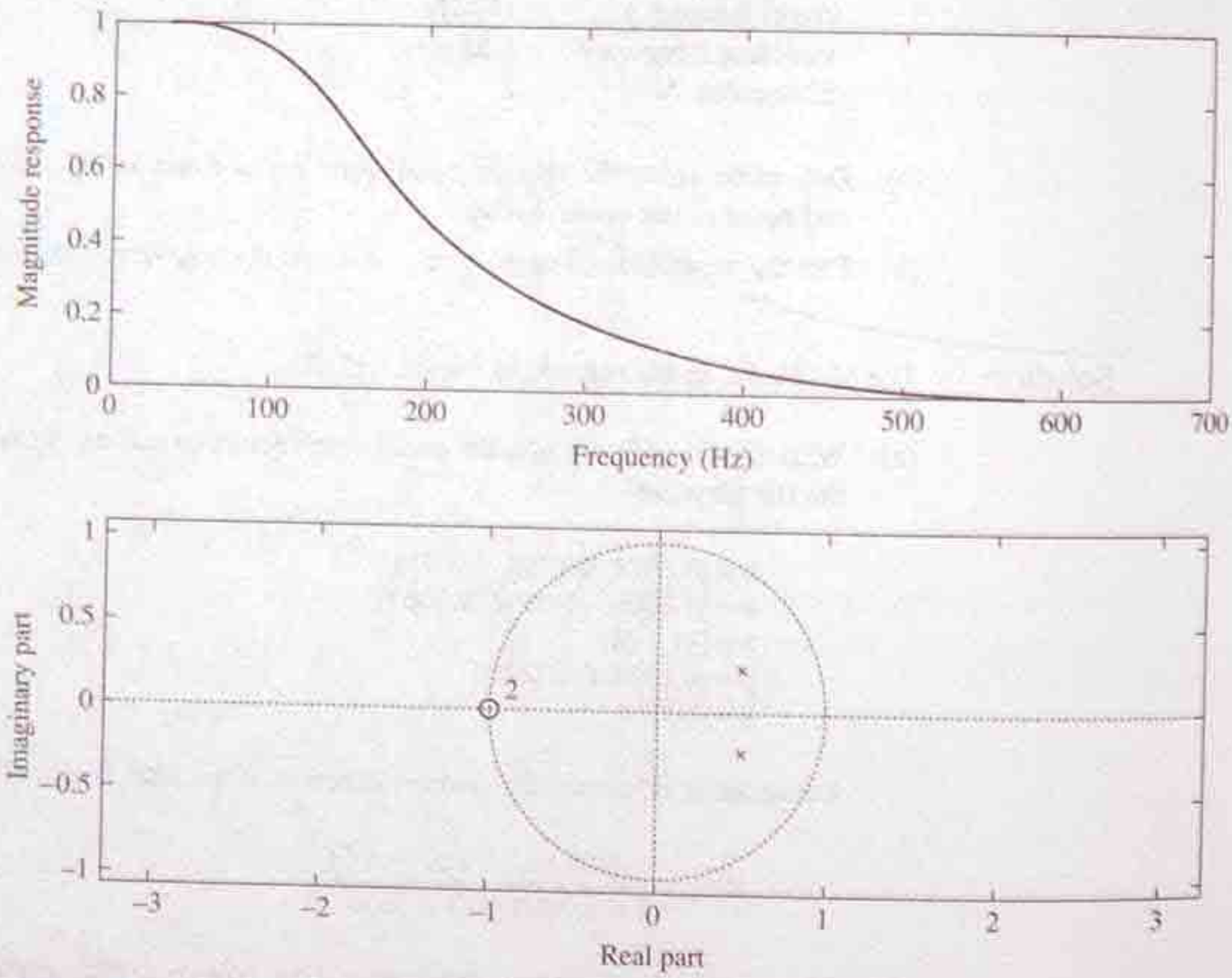


Figure 8B.2

Program 8B.3 MATLAB m-file for Example 8B.3.

```
%
%  Program name: EX8B3.m
%  A simple bandpass filter
%
Fs=2000;                                  %  Sampling frequency
FN=Fs/2;
fc1=200/FN;
fc2=300/FN;
[b,a]=butter (4,[fc1, fc2]);              %  Create/digitize analog filter.
[z,p,k]=butter (4, [fc1, fc2]);
subplot(2,1,1)                            %  Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot(f, abs(H))
xlabel ('Frequency (Hz)')
ylabel ('Magnitude Response ')
subplot (2,1,2)                           %  Plot pole-zero diagram
zplane (b, a)
```

Example 8B.3 *Design of a simple bandpass filter using the bilinear z-transform method and MATLAB*

(a) Calculate the coefficients of a discrete bandpass filter with Butterworth characteristics that meets the following specifications:

passband	200–300 Hz
sampling frequency	2000 Hz
filter order	8

(b) Plot its magnitude–frequency response and pole–zero diagram.

Solution The MATLAB m-file is listed in Program 8B.3.

(a) The coefficients, b and a, and the zeros, poles and gain, z, p and k respectively, are given below (note that the filter order used in the m-file is half that specified in the problem; for bandpass and bandstop filters the order is $2N$):

b = [0.0004, 0, –0.0017, 0, 0.0025, 0, –0.0017, 0, 0.0004]
a = [1.0000, –5.1408, 13.1256, –20.9376, 22.6982, 17.0342, 8.6867, 2.7672, 0.4383]
z = [1, 1, 1, 1, –1, –1, –1, –1]
p = [0.5601 ± 0.7475j, 0.5800 ± 0.6286j, 0.6656 ± 0.5628j, 0.7647 ± 0.5648j]
k = 4.1660e–004

(b) The magnitude–frequency response and the pole–zero diagram are depicted in Figure 8B.3.

Example 8B.4 *Design of a lowpass filter with specified pass- and stopband edge frequencies and ripples/attenuation* A lowpass digital filter with a Butterworth characteristic is required to meet the following specifications:

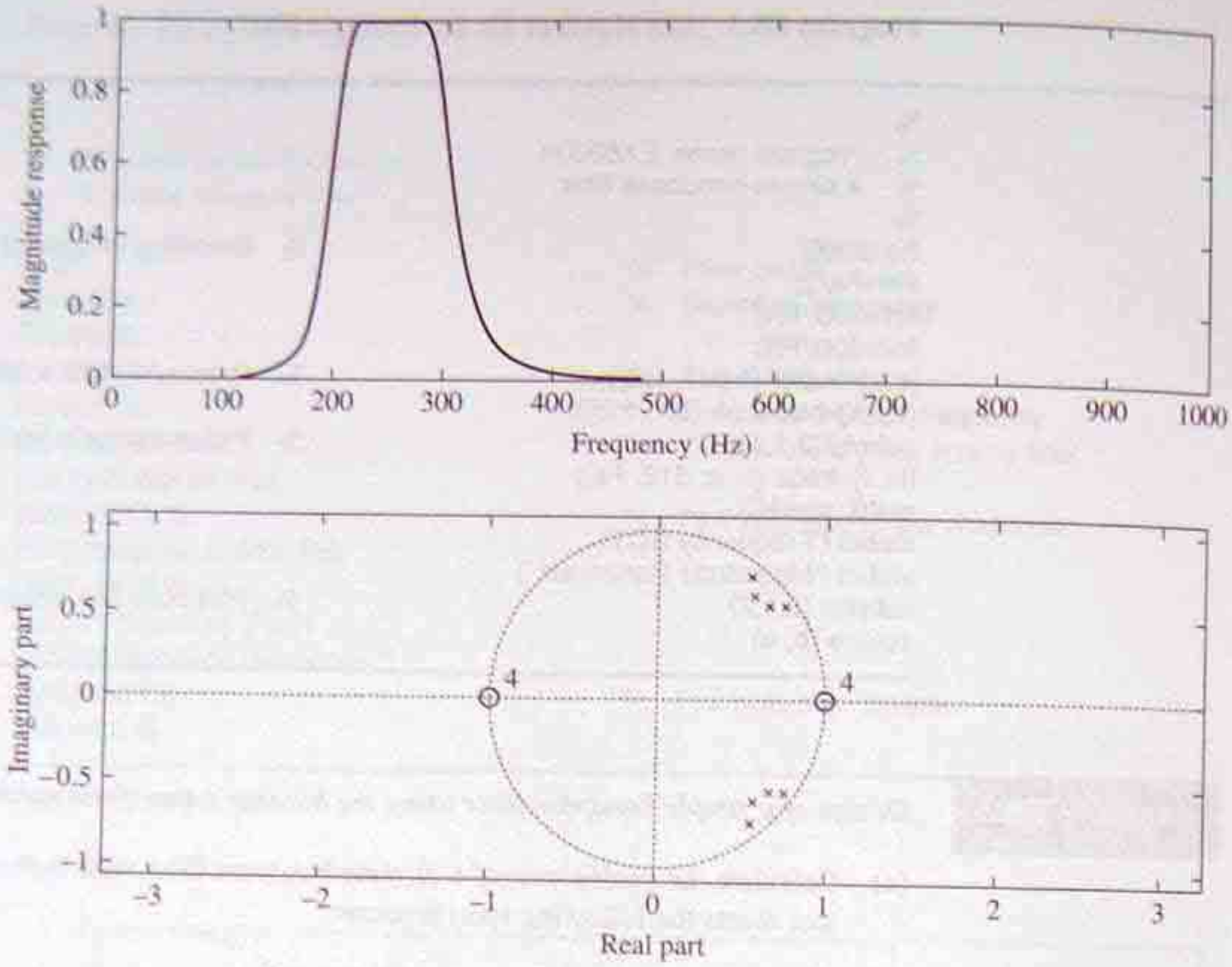


Figure 8B.3

passband	0–500 Hz
stopband	2–4 kHz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	8 kHz

(a) Determine the order, N, and the coefficients of the discrete-time filter using the bilinear z-transform and MATLAB.

(b) Plot the magnitude–frequency response and the pole–zero diagram of the filter.

Solution The first part of the example is similar to Example 8.11 which was solved by hand. Here we will use MATLAB. It is possible to use separate MATLAB commands to determine the order of an analog filter (**buttord** command), the poles and zeros of a prototype analog lowpass filter (**buttap** command), the parameters of the transformed analog filter (**butter** command), and finally the coefficients of the discrete-time filter (**bilinear** command). However, here we will make use of the digital domain version of the butter command as this performs the intermediate steps automatically to obtain the filter coefficients.

(a) The MATLAB implementation for the problem in the form of an m-file is given in Program 8B.4.

The order, N, zeros and poles (zz and pz), gain (kz) and coefficients of the filter (b and a) are

Program 8B.4 MATLAB m-file for Example 8B.4.

```
%
%   Program name: EX8B4.m
%   Lowpass filter
%
Fs=8000;                                    %   Sampling frequency
Ap=3;
As=20;
wp=500/4000;
ws=2000/4000;
[N, wc]=buttord(wp, ws, Ap, As);            %   Determine filter order
[zz, pz, kz]=butter (N,500/4000);           %   Digitize filter
[b, a]=butter(N, 500/4000);
subplot (2,1,1)                             %   Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot(f, abs(H))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response ')
subplot(2,1,2)                              %   Plot pole-zero diagram
zplane(b, a)
```

```
N  = 2
zz = [−1, −1]
pz = [0.7271 ± 0.2130j]
kz = 0.0300
b  = [0.0300   0.0599   0.0300]
a  = [1.0000  −1.4542   0.5741]
```

The coefficients are the same as those obtained by hand in Example 8.11, and the transfer function is

$$H(z) = \frac{0.030(1 + 2z^{-1} + z^{-2})}{1 - 1.4542z^{-1} + 0.5741z^{-2}}$$

(b) The magnitude–frequency response and the pole–zero diagram are depicted in Figure 8B.4.

Example 8B.5 *Design of a highpass filter with specified pass- and stopband edge frequencies and ripples/attenuation* A highpass digital filter meeting the following specifications is required:

passband	2–4 kHz
stopband	0–500 Hz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	8 kHz

Determine the following:

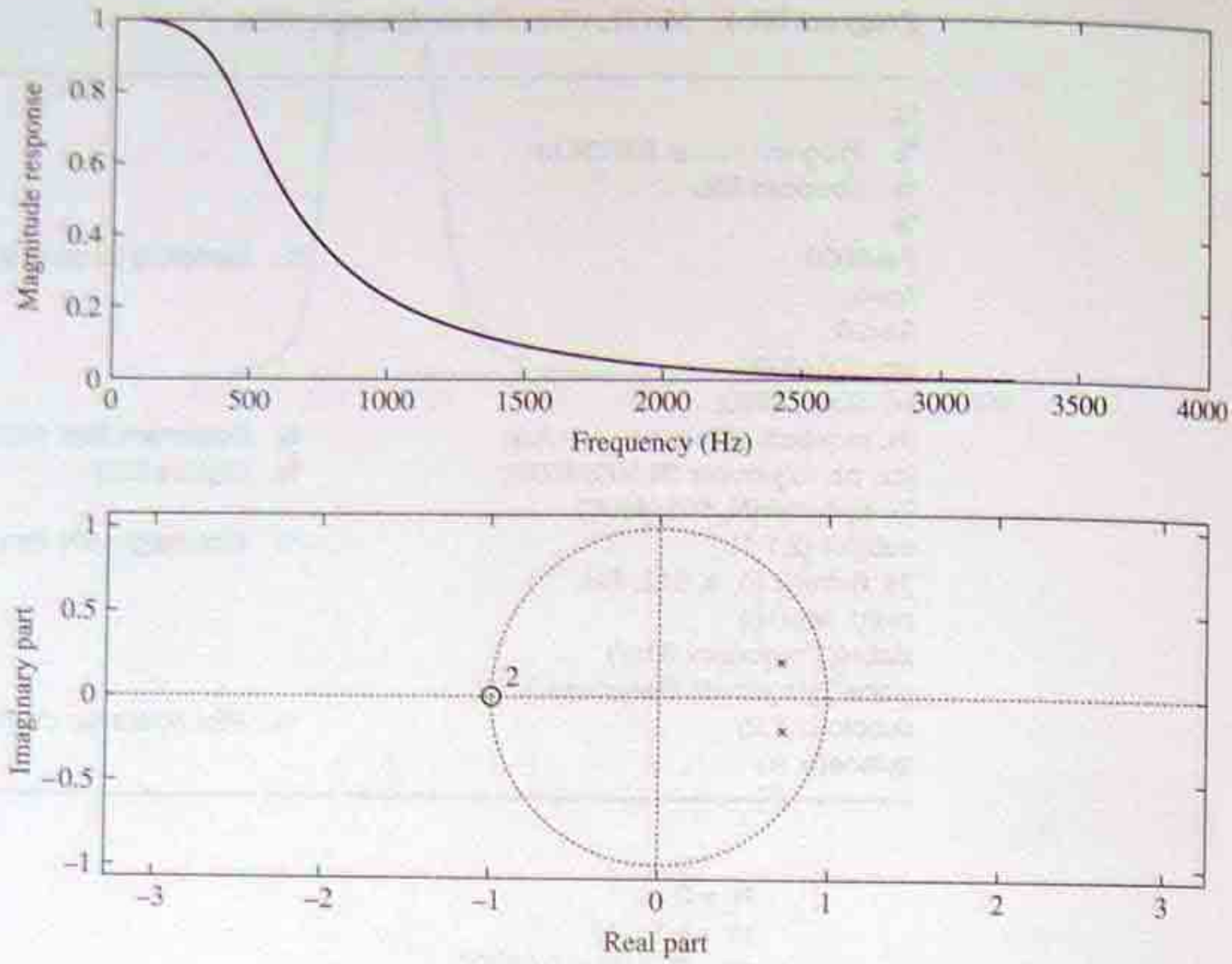


Figure 8B.4

(a) Pass- and stopband edge frequencies for a suitable analog prototype lowpass filter.
(b) Order, N, of the prototype lowpass filter.
(c) Coefficients, and hence the transfer function, of the discrete-time filter using the bilinear z-transform.

Solution

(a) The MATLAB implementation for the problem in the form of an m-file is given in Program 8B.5.
(b) The order, N, zeros and poles (zz and pz), gain (kz) and coefficients of the filter (b and a) are

```
N = 2
zz = [1,1]
pz = ±0.4142j
kz = 0.2929
b = [0.2929, –0.5858, 0.2929]
a = [1.0000, 0.0000, 0.1716]
```

(c) Using the coefficients, the transfer function of the filter is given by

$$H(z) = \frac{0.2929(1 - 2z^{-1} + z^{-2})}{1 + 0.1716z^{-2}}$$

The magnitude–frequency response and the pole–zero diagram are shown in Figure 8B.5.

Program 8B.5 MATLAB m-file for Example 8B.5.

```
%
%  Program name: EX8B5.m
%
Fs=8000;                                  %  Sampling frequency
Ap=3;
As=20;
wp=2000/4000;
ws=500/4000;
[N, wc]=buttord (wp, ws, Ap, As);         %  Determine filter order
[zz, pz, kz]=butter (N, 2000/4000, 'high');  %  Digitize filter
[b, a]=butter (N, 2000/4000, 'high');
subplot(2,1,1)                            %  Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot(f, abs(H))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response ')
subplot(2,1,2)                            %  Plot pole-zero diagram
zplane(b, a)
```

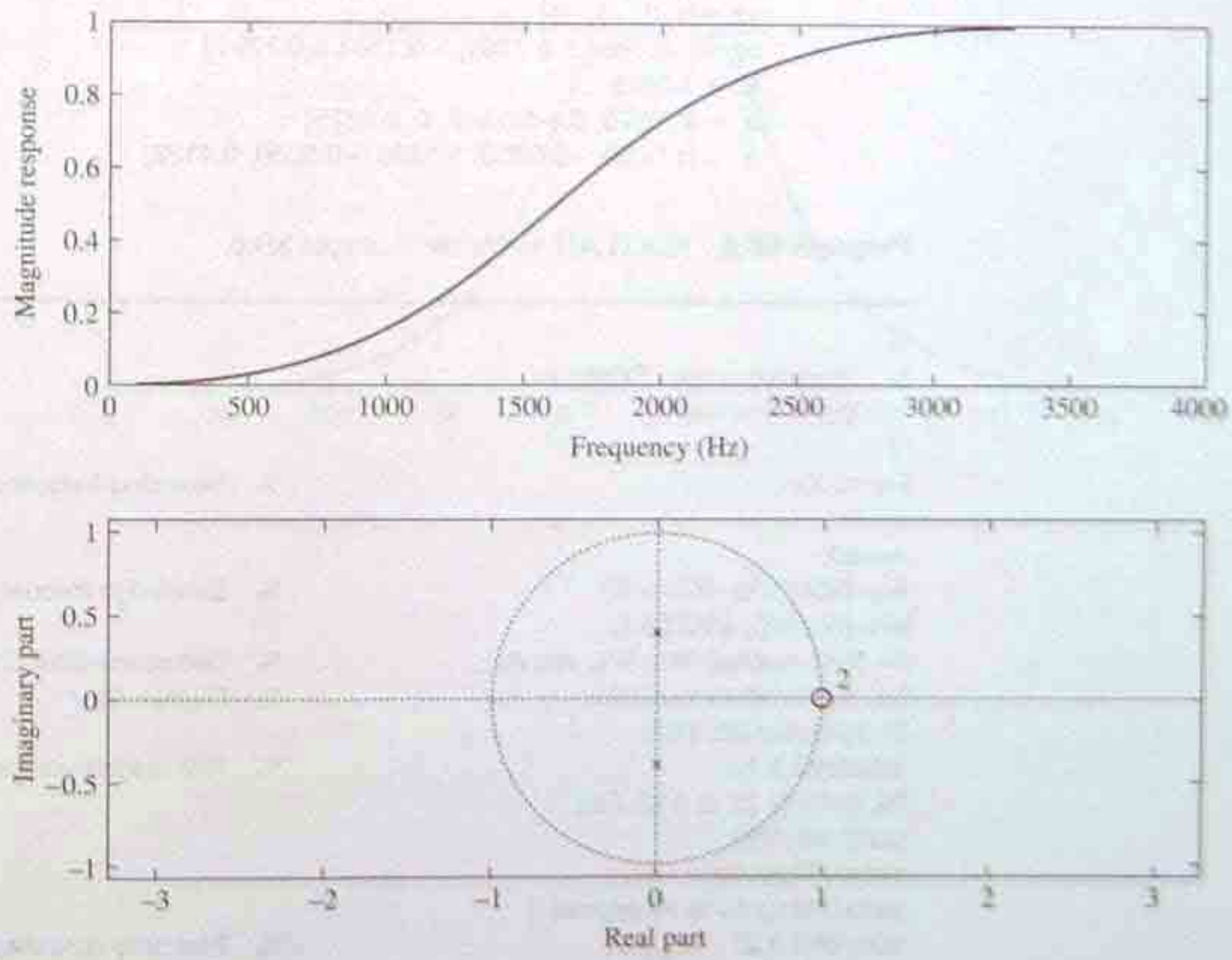


Figure 8B.5

Example 8B.6 *Design of a bandpass filter with specified bandedge frequencies and pass- and stopband ripples* A requirement exists for a bandpass digital filter, with a Butterworth magnitude–frequency response, that satisfies the following specifications:

lower passband edge frequency	200 Hz
upper passband edge frequency	300 Hz
lower stopband edge frequency	50 Hz
upper stopband edge frequency	450 Hz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	1 kHz

(a) Determine, using the BZT method and MATLAB:

(i) the order, N, of the filter;

(ii) the poles, zeros, gain, coefficients and transfer function of the discrete-time filter.

(b) Plot the magnitude–frequency response and the pole–zero diagram of the filter.

Solution (a) The MATLAB m-file is given in Program 8B.6. Using the m-file, the order, N, of the filter, the zeros and poles (zz and kz), the gain (kz) and the coefficients (b and a) of the discrete filter are

```
N  = 2 (the order of the bandpass filter is 2 * N, i.e. 4)
zz = [1, 1, −1, −1]
pz = [−0.1884 ± 0.7791j, −0.1884 ± 0.7791j]
kz = 0.0675
b  = [0.0675, 0, −0.1349, 0, 0.0675]
a  = [1.0000, −0.0000, 1.1430, −0.0000, 0.4128]
```

Program 8B.6 MATLAB m-file for Example 8B.6.

```
%
% Program name: EX8B6.m
% Bandpass filter
%
Fs=1000;                                  % Sampling frequency
Ap=3;
As=20;
Wp=[200/500, 300/500];                    % Bandedge frequencies
Ws=[50/500, 450/500];
[N, Wc]=buttord (Wp, Ws, Ap, As);         % Determine filter order
[zz, pz, kz]=butter (N,Wp);               % Digitize filter
[b, a]=butter (N, Wp);
subplot(2,1,1)                            % Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot(f, abs(H))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response ')
subplot(2,1,2)                            % Plot pole-zero diagram
zplane(b, a)
```

The corresponding transfer function, $H(z)$, is

$$H(z) = \frac{0.0675(1 - 2z^{-2} + z^{-4})}{1 + 1.143z^{-2} + 0.4128z^{-4}}$$

(b) If we set $N = 1$ (as in Example 8.11), the zeros (zz), poles (pz), gain (kz) and coefficients (b and a) become

```
zz = [-1, 1]
pz = [0.0000 ± 0.7138j]
kz = 0.2452

b  = [0.2452, 0, -0.2452]
a  = [1.0000, -0.0000, 0.5095]
```

The corresponding transfer function, $H(z)$, is

$$H(z) = \frac{0.2452(1 - z^{-2})}{1 + 0.5095z^{-2}}$$

The magnitude–frequency response and the pole–zero diagram are shown in Figure 8B.6.

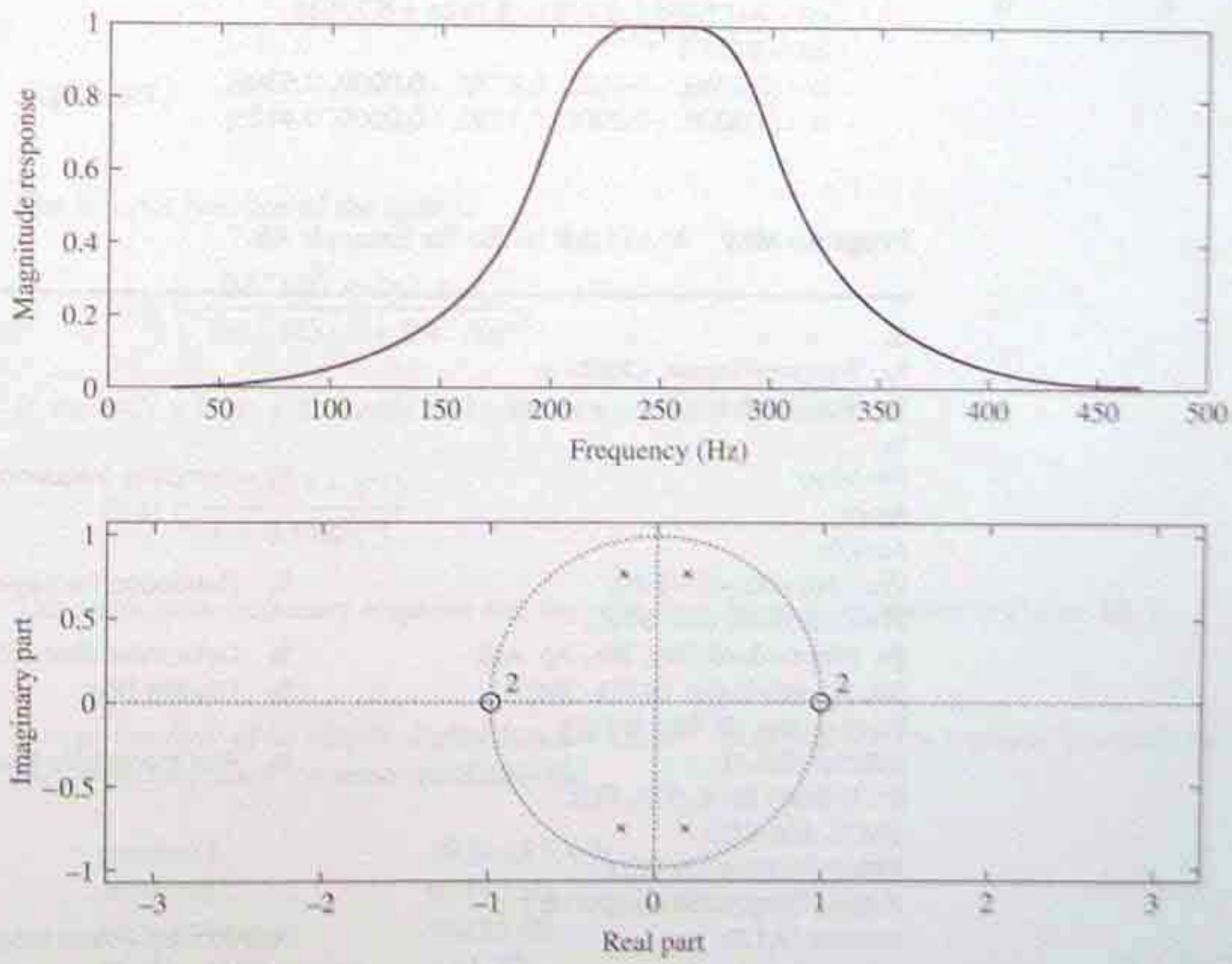


Figure 8B.6

Example 8B.7 *Design of a bandstop filter with specified bandedge frequencies and pass- and stopband ripples*

A requirement exists for a bandstop digital IIR filter, with a Butterworth magnitude–frequency response, that meets the following specifications:

lower passband	0–50 Hz
upper passband	450–500 Hz
stopband	200–300 Hz
passband ripple	3 dB
stopband attenuation	20 dB
sampling frequency	1 kHz

Determine:

(a) The pass- and stopband edge frequencies of a suitable prototype lowpass filter.

(b) The order, N, of the prototype lowpass filter.

(c) The coefficients, and hence the transfer function, of the discrete-time filter using the BZT method.

Solution The MATLAB m-file for the example is listed in Program 8B.7. Using the m-file, the filter order, N, zeros and poles (zz and pz), gain (kz) and coefficients are

```
N  = 2 (order of the bandstop filter = 2 * N, i.e. 4)
zz = two pairs of zeros at j and at –j
pz = [–0.1884 ± 0.7791j, 0.1884 ± 0.7791j]
kz = 0.6389
b = [0.6389, –0.0000, 1.2779, –0.0000, 0.6389]
a = [1.0000, –0.0000, 1.1430, –0.0000, 0.4128]
```

Program 8B.7 MATLAB m-file for Example 8B.7.

```
%
% Program name: EX8B7.m
% Bandstop filter
%
Fs=1000;                                  % Sampling frequency
Ap=3;
As=20;
Wp=[50/500, 450/500];                     % Bandedge frequencies
Ws=[200/500, 300/500];
[N, Wc]=buttord (Wp, Ws, Ap, As);         % Determine filter order
[zz, pz, kz]=butter (N,Ws, 'stop');       % Digitize filter
[b, a]=butter (N, Ws, 'stop');
subplot (2,1,1)                           % Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot (f, abs (H))
xlabel ('Frequency (Hz)')
ylabel ('Magnitude Response')
subplot (2,1,2)                           % Plot pole-zero diagram
zplane (b, a)
```

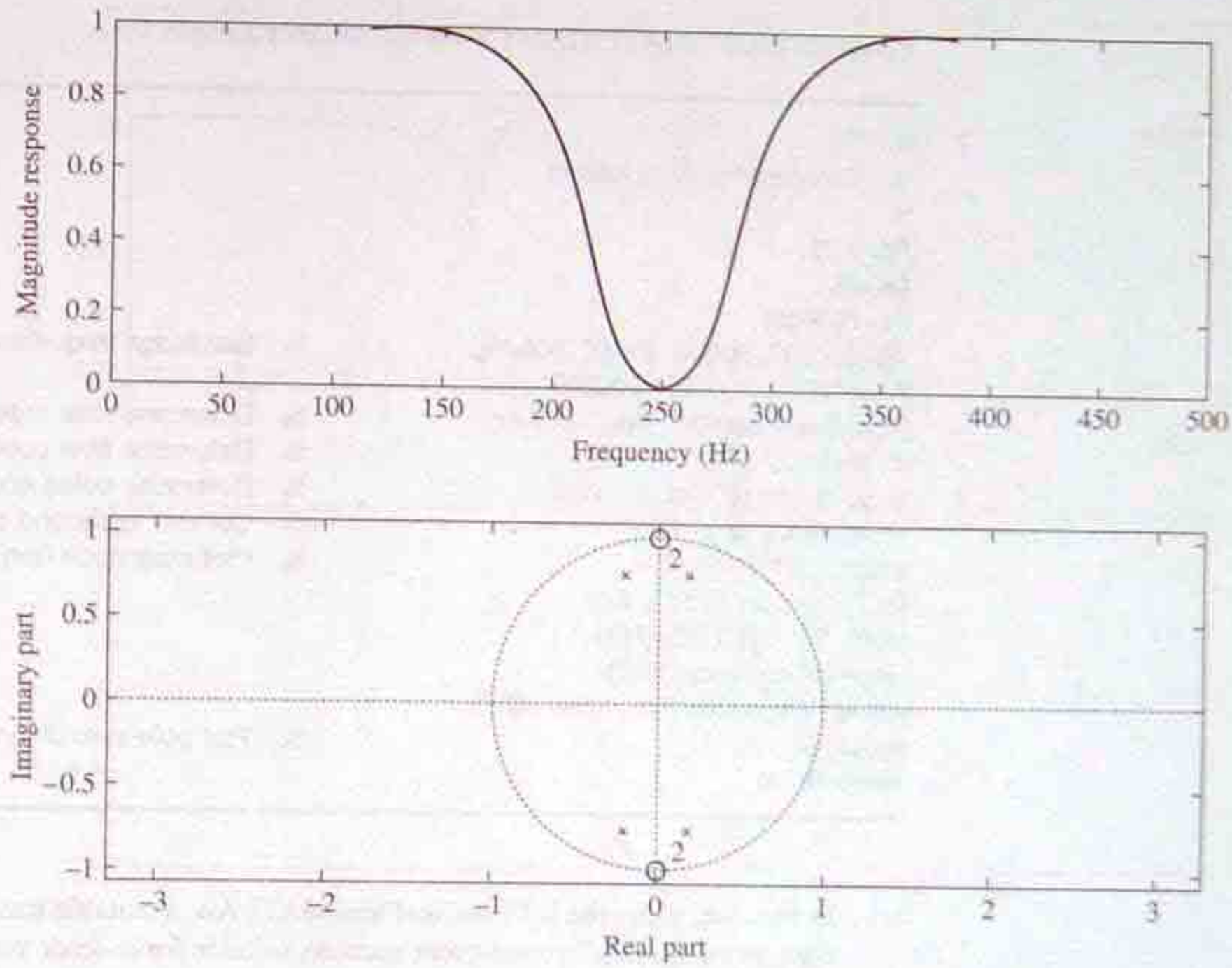


Figure 8B.7

The transfer function of the filter is

$$H(z) = \frac{0.6389(1 + 2z^{-2} + z^{-4})}{1 + 1.143z^{-2} + 0.4128z^{-4}}$$

If we set $N = 1$ (as in Example 8.12), the transfer function becomes

$$H(z) = \frac{0.7548(1 + z^{-2})}{1 + 0.5095z^{-2}}$$

The magnitude–frequency response and the pole–zero diagram are shown in Figure 8B.7.

Example 8B.8 *Design example of an elliptic digital bandpass filter* A digital filter is required to satisfy the following frequency response specifications:

passband	20.5–23.5 kHz
stopband	0–19 kHz, 25–50 kHz
passband ripple	≤0.25 dB
stopband attenuation	>45 dB
sampling frequency	100 kHz

Program 8B.8 MATLAB m-file for the design Example 8B.8.

```
%
%   Program Name: EX8B.m
%
Ap=0.25;
As=45;
Fs=100000;
Wp=[20500/50000, 23500/50000];          %  Bandedge frequencies
Ws=[19000/50000, 25000/50000];
[N,Wc]=ellipord(Wp, Ws, Ap, As);        %  Determine filter order
[b, a]=ellip(N, Ap, As, Wc);            %  Determine filter coeffs
[z, p, k]=ellip(N, Ap, As, Wc);         %  Determine poles and zeros
sos=zp2sos (z, p, k);                   %  Convert to second order sections
subplot(2,1,1)                          %  Plot magnitude freq. response
[H, f]=freqz(b, a, 512, Fs);
plot(f, 20*log10 (abs(H)))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response (dB)')
subplot(2,1,2)                          %  Plot pole-zero diagram
zplane(b, a)
```

(a) Determine, using the BZT method and MATLAB, a suitable transfer function for the filter in the form of second-order sections suitable for cascade realization.

(b) Obtain and plot the magnitude–frequency response and pole–zero diagram of the filter.

Assume an elliptic characteristic.

Solution

(a) The MATLAB m-file for the design problem is listed in Program 8B.8. The **ellipord** command is used to determine the order of a suitable elliptic prototype filter. The **ellip** command is then used to determine the coefficients, poles and zeros of the transfer function. The **zp2sos** command converts the transfer function into second-order sections. The coefficients for the second-order sections are returned in the matrix, sos.

The filter coefficients (b and a), poles and zeros (p and z), gain (k), and the matrix of second-order sections, sos, are listed below:

```
Filter order, N = 4
b = [0.0061, −0.0083, 0.0236, −0.0221, 0.0351, −0.0221, 0.0236, −0.0083, 0.0061]
a = [1.0000, −1.4483, 4.4832 −4.2207, 6.6475, −3.9458, 3.9187, −1.1828, 0.7634]
z = [−0.0118 ± 0.9999j, 0.3737 ± 0.9275j, −0.2553 ± 0.9669j, 0.5663 ± 0.8242j]
p = [0.0872 ± 0.9793j, 0.1352 ± 0.9404j, 0.2795 ± 0.9432j, 0.2223 ± 0.9246j]
k = 0.0061
sos =
    0.1203    0.0614    0.1203    1.0000   −0.2705    0.9026
    0.2051   −0.2323    0.2051    1.0000   −0.4446    0.9042
    0.3740    0.0088    0.3740    1.0000   −0.1744    0.9665
    0.6642   −0.4965    0.6642    1.0000   −0.5589    0.9678
```

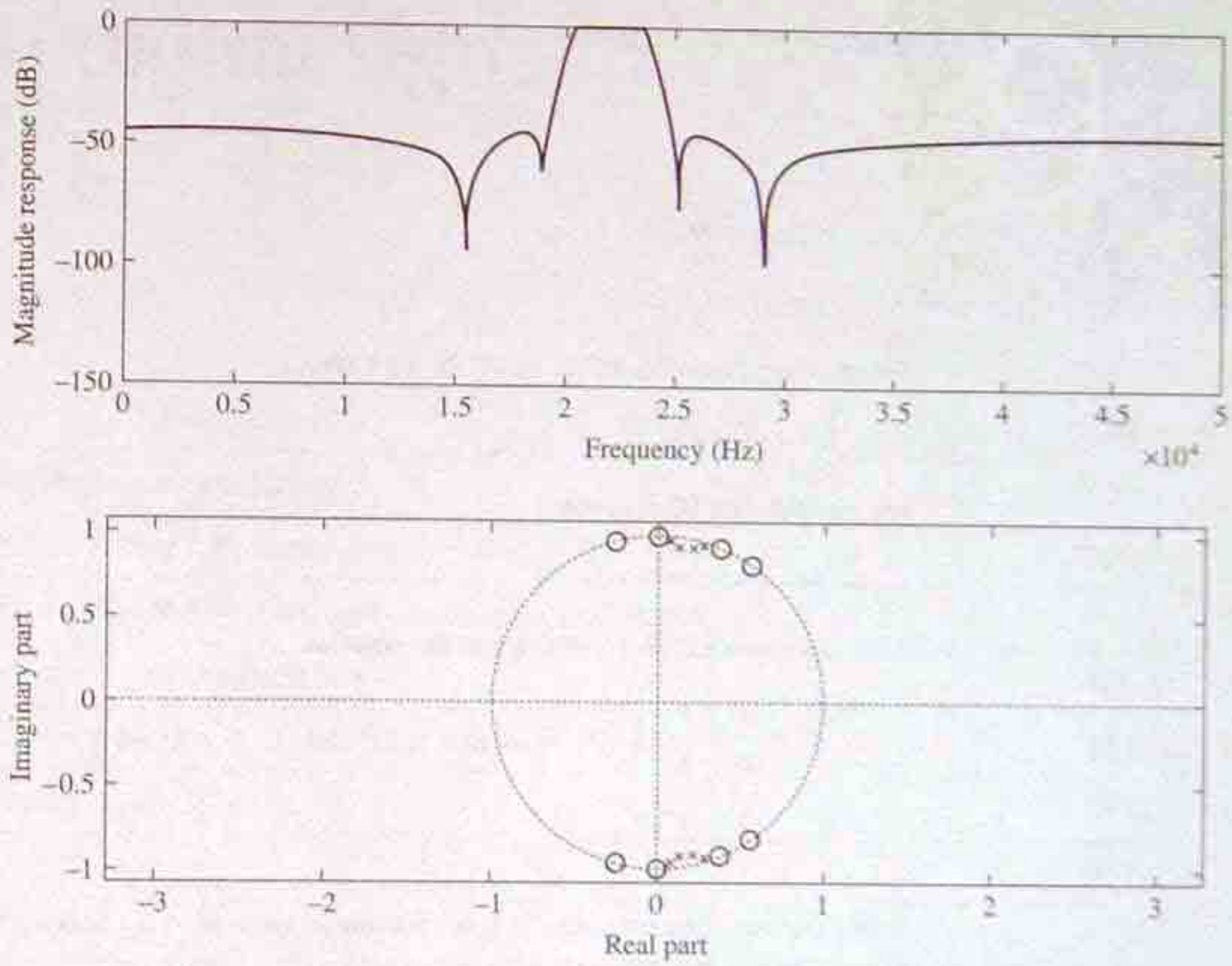


Figure 8B.8

The sos is a 4×6 matrix of second-order sections which has the following format:

$$\begin{matrix} b_{01} & b_{11} & b_{21} & a_{01} & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & a_{02} & a_{12} & a_{22} \\ b_{03} & b_{13} & b_{23} & a_{03} & a_{13} & a_{23} \\ b_{04} & b_{14} & b_{24} & a_{04} & a_{14} & a_{24} \end{matrix}$$

It is evident that the coefficients of the second-order sections are held in the rows, one row per second-order section.

The magnitude–frequency response and the pole–zero diagram are shown in Figure 8B.8.

8C Evaluation of complex square roots using real arithmetic

Given a complex number with rectangular coordinates $x + jy$, let its square root be $u + jv$:

$$u + jv = (x + jy)^{1/2} \tag{8C.1}$$

Squaring Equation 8.C1 we have

$$u^2 - v^2 + j2uv = x + jy$$

Equating real and imaginary parts gives

$$u^2 - v^2 = x \tag{8C.2a}$$

$$2uv = y \tag{8C.2b}$$

From Equation 8C.2b

$$v = y/2u \tag{8C.2c}$$

Substituting Equation 8C.2c in 8C.2a we have

$$u^2 - y^2/4u^2 = x \tag{8C.3}$$

and simplifying 8C.3 gives

$$4u^4 - 4xu^2 - y^2 = 0$$

This is a quadratic in u^2, which has the solution

$$u^2 = \frac{x}{2} + \frac{1}{8}(16x^2 + 16y^2)^{1/2}$$

$$= \frac{x + (x^2 + y^2)^{1/2}}{2}$$

(Note that the negative solution is not permitted since $u^2 \geqslant 0$.) Solving for u and using Equation 8C.2c, the solution for u is

$$u = \left[\frac{x + (x^2 + y^2)^{1/2}}{2}\right]^{1/2} \tag{8C.4a}$$

$$v = y/2u \tag{8C.4b}$$

or

$$u = -\left[\frac{x + (x^2 + y^2)^{1/2}}{2}\right]^{1/2} \tag{8C.5a}$$

$$v = y/2u \tag{8C.5b}$$

The algorithm for computing the square root of $x + jy$ is then

- Step 1: let $u + jv = (x + jy)^{1/2}$
- Step 2: $u = \pm\left[\dfrac{x + (x^2 + y^2)^{1/2}}{2}\right]^{1/2} = \pm\left(\dfrac{x + |x + jy|}{2}\right)^{1/2}$
- Step 3: $v = \pm y/2u$

For example

$$(-1 + j)^{1/2} = 0.455\,09 + 1.098\,68j \text{ and } -0.455\,09 - 1.098\,68j$$

$$(-1 - j)^{1/2} = 0.455\,09 - 1.098\,68j \text{ and } -0.455\,09 + 1.098\,68j$$