

6

Infinite Impulse Response Filters

Ren Zhou

In this chapter we introduce the analysis and design of infinite impulse response (IIR) digital filters that have the potential of sharp rolloffs (Tompkins and Webster, 1981). We show a simple one-pole example illustrating the relationship between pole position and filter stability. Then we show how to design two-pole filters with low-pass, bandpass, high-pass, and band-reject characteristics. We also present algorithms to implement integrators that are all IIR filters. Finally, we provide a laboratory exercise that uses IIR filters for ECG analysis.

6.1 GENERIC EQUATIONS OF IIR FILTERS

The generic format of transfer function of IIR filters is expressed as the ratio of two polynomials:

$$H(z) = \frac{\sum_{i=0}^n a_i z^{-i}}{1 - \sum_{i=1}^n b_i z^{-i}} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}{1 - b_1 z^{-1} - b_2 z^{-2} - \dots - b_n z^{-n}} = \frac{Y(z)}{X(z)} \quad (6.1)$$

Rearranging the terms gives

$$Y(z) = b_1 Y(z)z^{-1} + \dots + b_n Y(z)z^{-n} + a_0 X(z) + a_1 X(z)z^{-1} + \dots + a_n X(z)z^{-n} \quad (6.2)$$

The $Y(z)$ terms on the right side of this equation are delayed feedback terms. Figure 6.1 shows these feedback terms as recursive loops; hence, these types of filters are also called recursive filters.

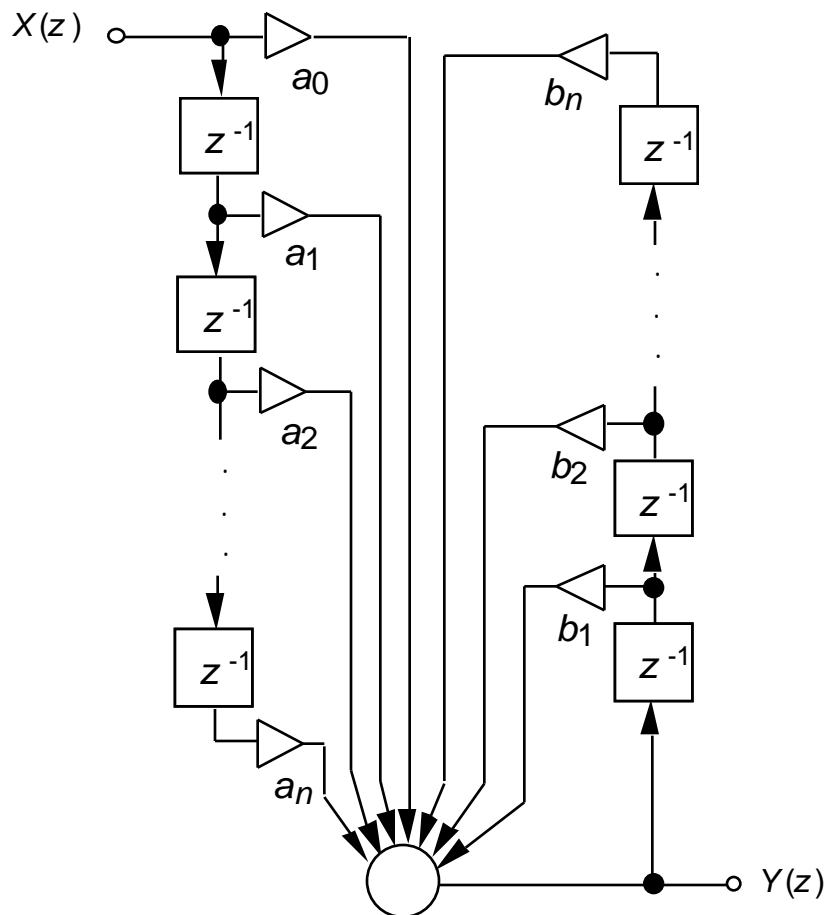


Figure 6.1 The output of an IIR filter is delayed and fed back.

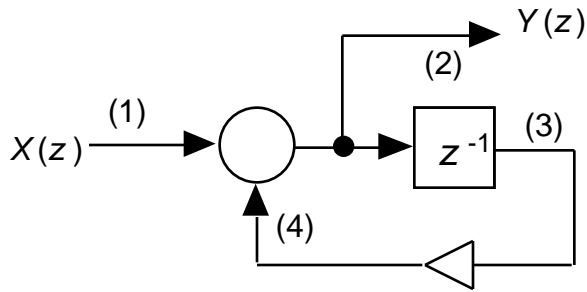
6.2 SIMPLE ONE-POLE EXAMPLE

Let us consider the simple filter of Figure 6.2(a). We find the transfer function by applying a unit impulse sequence to the input $X(z)$. Figure 6.2(b) shows the sequences at various points in the filter for a feedback coefficient β equal to $1/2$. Sequence (1) defines the unit impulse. From output sequence (2), we write the transfer function

$$H(z) = 1 + \frac{1}{2} z^{-1} + \frac{1}{4} z^{-2} + \frac{1}{8} z^{-3} + \dots \quad (6.3)$$

Using the binomial theorem, we write the infinite sum as a ratio of polynomials

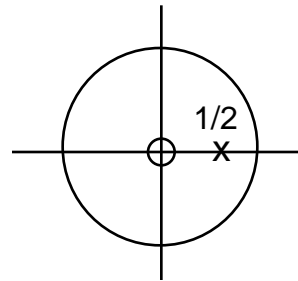
$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \frac{1}{2} z^{-1}} \quad (6.4)$$



(a)

- (1) (1, 0, 0, 0, 0, ...)
- (2) (1, 1/2, 1/4, 1/8, 1/16 ...)
- (3) (0, 1, 1/2, 1/4, 1/8, ...)
- (4) (0, 1/2, 1/4, 1/8, 1/16, ...)

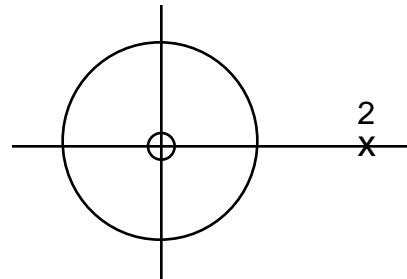
(b)



(c)

- (1) (1, 0, 0, 0, 0, ...)
- (2) (1, 2, 4, 8, 16 ...)
- (3) (0, 1, 2, 4, 8, ...)
- (4) (0, 2, 4, 8, 16, ...)

(d)



(e)

Figure 6.2 Simple one-pole recursive filters. (a) Block diagram. (b) Response to a unit pulse for $\beta = 1/2$. (c) Pole-zero plot for $\beta = 1/2$. (d) Response to a unit pulse for $\beta = 2$. (e) Pole-zero plot for $\beta = 2$.

We then rearrange this equation and write the output as a function of the feedback term and the input

$$Y(z) = \frac{1}{2} Y(z) z^{-1} + X(z) \tag{6.5}$$

Recognizing that $x(nT)$ and $y(nT)$ are points in the input and output sequences associated with the current sample time, they are analogous to the undelayed z -domain variables, $X(z)$ and $Y(z)$ respectively. Similarly $y(nT - T)$, the output value

one sample point in the past, is analogous to the output z -domain output variable delayed by one sample point, or $Y(z)z^{-1}$. We can then write the difference equation by inspection

$$y(nT) = \frac{1}{2} y(nT - T) + x(nT) \quad (6.6)$$

Unlike the FIR case where the current output $y(nT)$ is dependent only on current and past values of x , this IIR filter requires not only the current value of the input $x(nT)$ but also the previous value of the output itself $y(nT - T)$. Since past history of the output influences the next output value, which in turn influences the next successive output value, a transient requires a large number of sample points before it disappears from an output signal. As we have mentioned, this does not occur in an FIR filter because it has no feedback.

In order to find the poles and zeros, we first multiply the transfer function of Eq. (6.4) by z/z in order to make all the exponents of z positive.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \frac{1}{2}z^{-1}} \times \frac{z}{z} = \frac{z}{z - \frac{1}{2}} \quad (6.7)$$

By equating the numerator to zero, we find that this filter has a single zero at $z = 0$. Setting the denominator equal to zero in order to locate the poles gives

$$z - \frac{1}{2} = 0$$

Thus, there is a single pole at $z = 1/2$.

The pole-zero plot for this single-pole filter is shown in Figure 6.2(c). In order to find the amplitude and phase responses, we substitute $z = e^{j\omega T}$ into the transfer function

$$H(\omega T) = \frac{1}{1 - \frac{1}{2}e^{-j\omega T}} = \frac{1}{1 - \frac{1}{2}\cos(\omega T) + j\frac{1}{2}\sin(\omega T)} \quad (6.8)$$

Evaluating this function for the amplitude and phase responses, we find that this is a low-pass filter as illustrated in Figure 6.3.

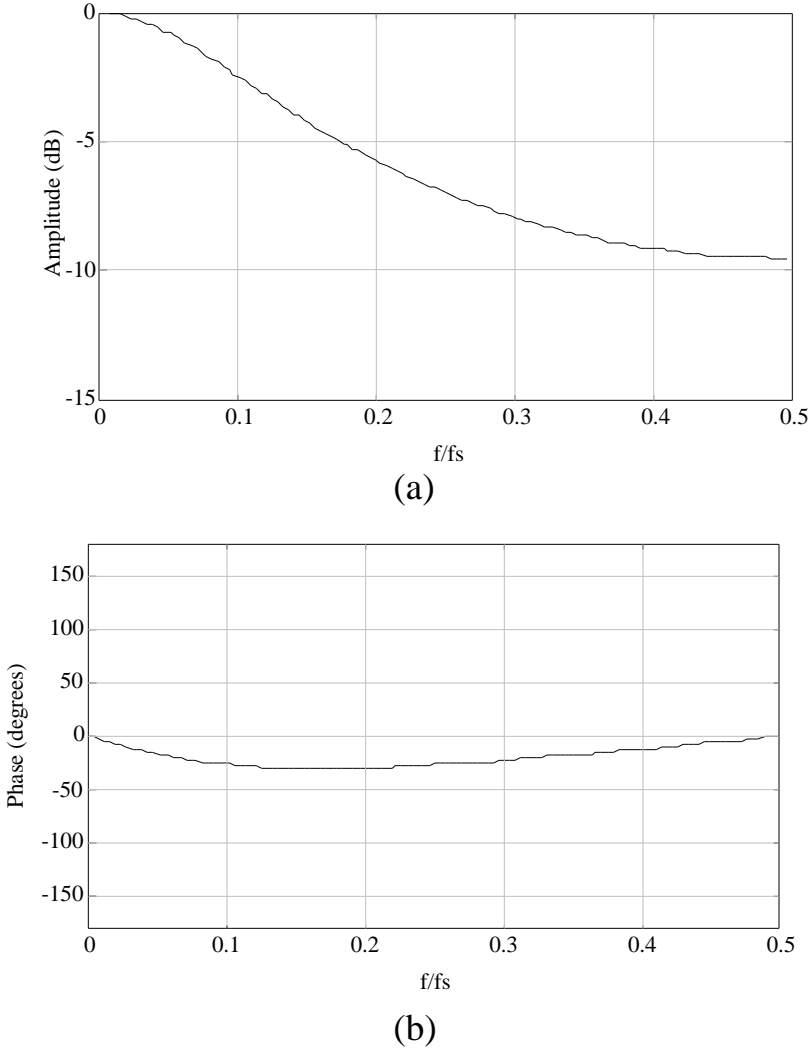


Figure 6.3 Simple one-pole recursive filter of Figure 6.2 with pole located at $z = 1/2$. (a) Amplitude response. (b) Phase response.

If we replace the multiplier constant β of $1/2$ in this filter by 2 , the output sequence in response to a unit impulse is $(1, 2, 4, 8, 16, \dots)$ as shown in Figure 6.2(d). The filter is unstable—the output increases with each successive sample. The response to a unit pulse input not only does not disappear with time, it increases by a factor of 2 each T s. We desire a unit pulse response that decays toward zero. Calculating the location of the pole with the multiplier of 2 , we find that the pole is at $z = 2$, as shown in Figure 6.2(e). It is outside the unit circle, and the filter is unstable, as expected.

If we replace the multiplier constant of $1/2$ in this filter by 1 , the pole is directly on the unit circle at $z = 1$, and the output response to a unit impulse applied to the input is $(1, 1, 1, 1, 1, \dots)$. This filter is a special IIR filter, a rectangular integrator (see section 6.3).

6.3 INTEGRATORS

The general form of the integral is

$$A = \int_{t_1}^{t_2} f(t) dt \quad (6.9)$$

where A is the area under the function between the limits t_1 to t_2 . The digital implementation for the determination of integral solutions is done by approximating the function using curve fitting techniques at a finite number of points, where

$$A = \sum_{n=t_1}^{t_2} f(n) \Delta t \quad (6.10)$$

The Laplace transform of an integrator is

$$H(s) = \frac{1}{s} \quad (6.11)$$

The amplitude and phase responses are found by substituting into the transfer function the relation, $s = j\omega$, giving

$$H(j\omega) = \frac{1}{j\omega} \quad (6.12)$$

Thus, the ideal amplitude response is inversely proportional to frequency

$$|H(j\omega)| = \frac{1}{\omega} \quad (6.13)$$

and the phase response is

$$H(j\omega) = \tan^{-1} \frac{-1/j\omega}{0} = -\frac{\pi}{2} \quad (6.14)$$

A number of numerical techniques exist for digital integration. Unlike the analog integrator, a digital integrator has no drift problems because the process is a computer program, which is not influenced in performance by residual charge on capacitors. We discuss here three popular digital integration techniques—rectangular summation, trapezoidal summation, and Simpson's rule.

6.3.1 Rectangular integration

This algorithm performs the simplest integration. It approximates the integral as a sum of rectangular areas. Figure 6.4(a) shows that each rectangle has a base equal in length to one sample period T and in height to the value of the most recently

sampled input $x(nT - T)$. The area of each rectangle is $Tx(nT - T)$. The difference equation is

$$y(nT) = y(nT - T) + T x(nT) \quad (6.15)$$

where $y(nT - T)$ represents the sum of all the rectangular areas prior to adding the most recent one. The error in this approximation is the difference between the area of the rectangle and actual signal represented by the sampled data. The z transform of Eq. (6.15) is

$$Y(z) = Y(z) z^{-1} + T X(z) \quad (6.16)$$

and

$$H(z) = \frac{Y(z)}{X(z)} = T \frac{1}{1 - z^{-1}} \quad (6.17)$$

Figure 6.4(a) shows that this transfer function has a pole at $z = 1$ and a zero at $z = 0$. The amplitude and phase responses are

$$|H(\omega T)| = \frac{T}{2 \sin(\omega T/2)} \quad (6.18)$$

and

$$H(\omega T) = \frac{\omega T}{2} - \bar{2} \quad (6.19)$$

Figure 6.5 shows the amplitude response. We can reduce the error of this filter by increasing the sampling rate significantly higher than the highest frequency present in the signal that we are integrating. This has the effect of making the width of each rectangle small compared to the rate of change of the signal, so that the area of each rectangle better approximates the input data. Increasing the sampling rate also corresponds to using only the portion of amplitude response at the lower frequencies, where the rectangular response better approximates the ideal response. Unfortunately, higher than necessary sampling rates increase computation time and waste memory space by giving us more sampled data than are necessary to characterize a signal. Therefore, we select other digital integrators for problems where higher performance is desirable.

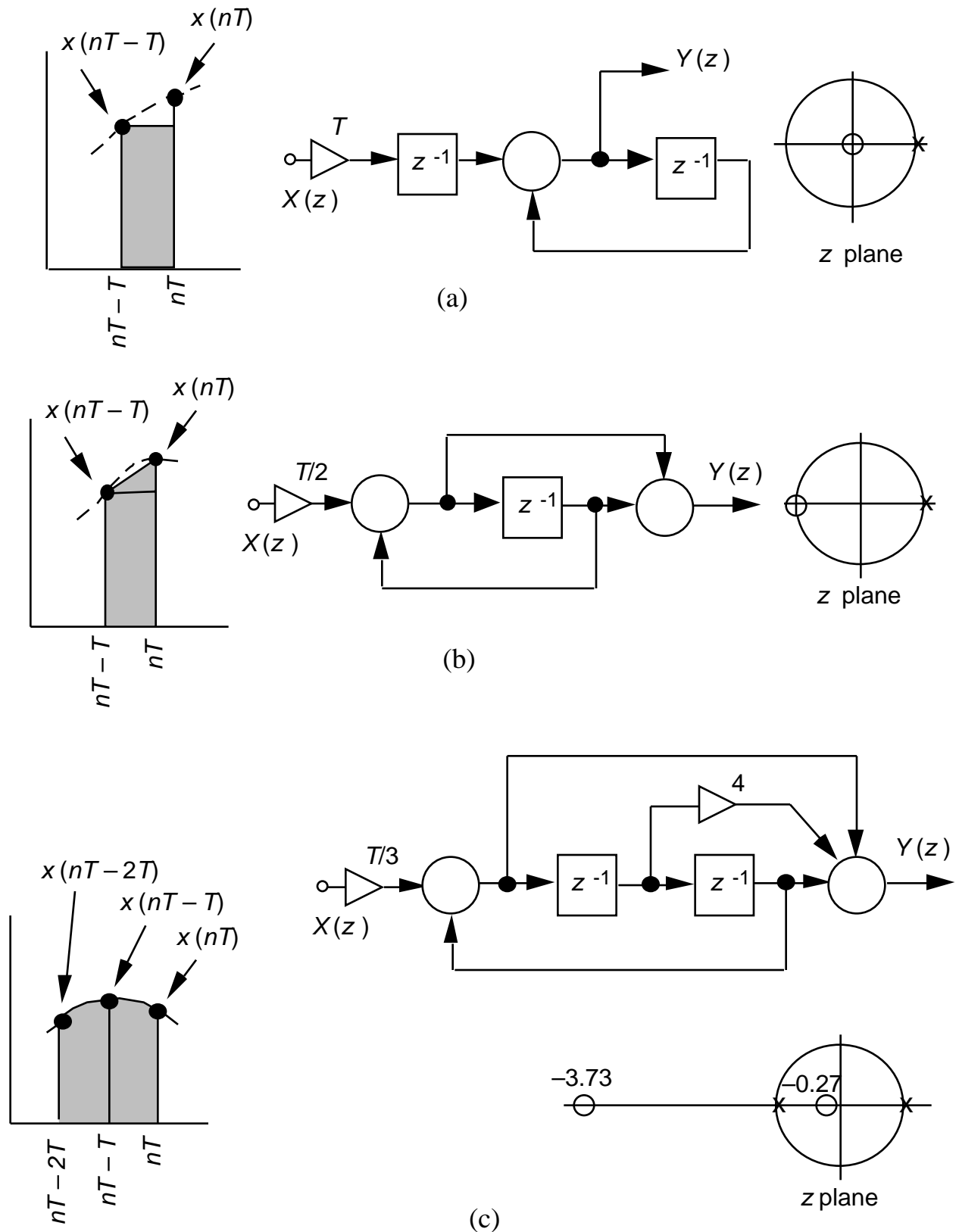
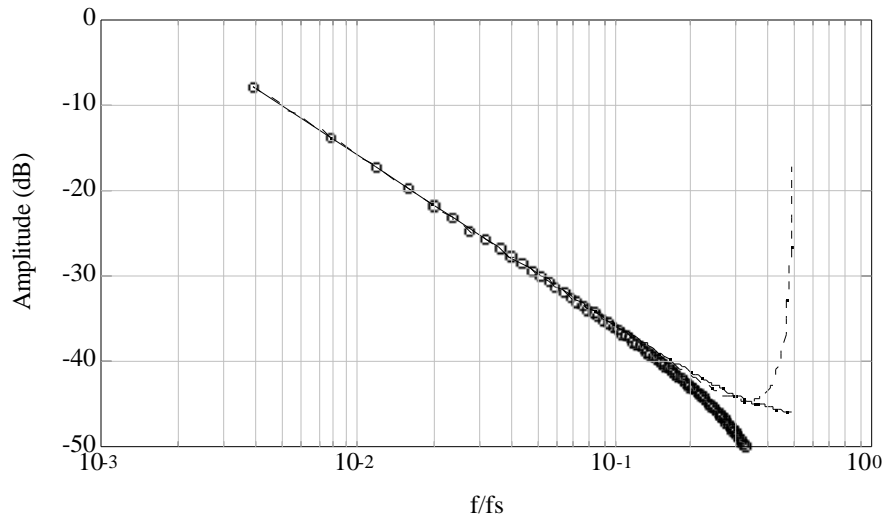
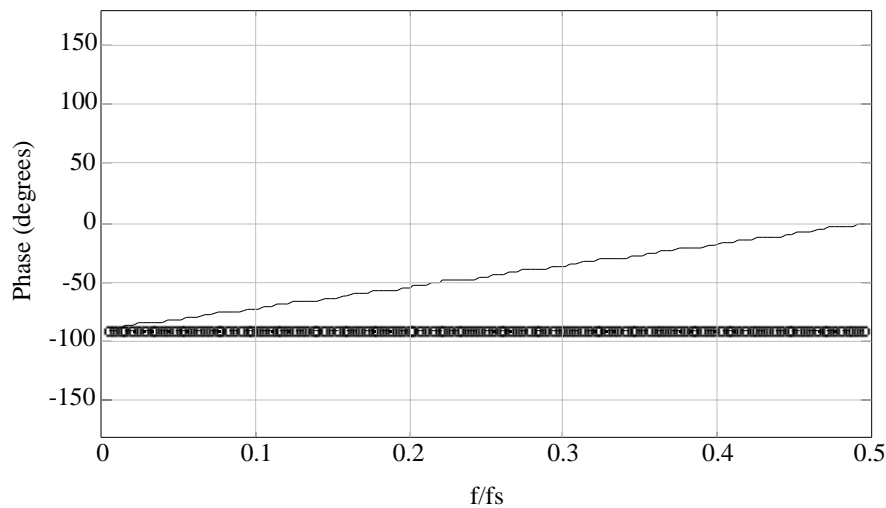


Figure 6.4 Integration. (a) Rectangular. (b) Trapezoidal. (c) Simpson's rule.



(a)



(b)

Figure 6.5 Digital integrators. The sampling period T is set equal to 1. (a) Amplitude responses. (b) Phase responses. Solid line: rectangular. Circles: trapezoidal. Dashed line: Simpson's rule.

6.3.2 Trapezoidal integration

This filter improves on rectangular integration by adding a triangular element to the rectangular approximation, as shown in Figure 6.4(b). The difference equation is the same as for rectangular integration except that we add the triangular element.

$$\begin{aligned}
 y(nT) &= y(nT - T) + Tx(nT - T) + \frac{T}{2} [x(nT) - x(nT - T)] \\
 &= y(nT - T) + \frac{T}{2} [x(nT) + x(nT - T)]
 \end{aligned} \tag{6.20}$$

This corresponds to the transfer function

$$H(z) = \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} \quad (6.21)$$

For the exact amplitude and phase responses, we use the analytical approach. Since we desire the frequency response, we evaluate the transfer function on the unit circle by simply substituting $e^{j\omega T}$ for every occurrence of z . For the recursive filter

$$H(z) = \frac{1 + z^{-1}}{1 - z^{-1}} \quad (6.22)$$

we obtain

$$H(\omega T) = \frac{1 + e^{-j\omega T}}{1 - e^{-j\omega T}} = \frac{e^{-j\frac{\omega T}{2}} e^{j\frac{\omega T}{2}} + e^{-j\frac{\omega T}{2}}}{e^{-j\frac{\omega T}{2}} e^{j\frac{\omega T}{2}} - e^{-j\frac{\omega T}{2}}} = \frac{2\cos \frac{\omega T}{2}}{2j\sin \frac{\omega T}{2}} = -j\cot \frac{\omega T}{2}$$

The amplitude response is the magnitude of $H(\omega T)$

$$|H(\omega T)| = \cot \frac{\omega T}{2} \quad (6.23)$$

and the phase response is

$$H(\omega T) = -\bar{2} \quad (6.24)$$

The block diagram and pole-zero plot are shown in Figure 6.4(b). Like the rectangular approach, this function still has a pole at $z = 1$, but the zero is moved to $z = -1$, the location of the folding frequency, giving us zero amplitude response at f_0 . The amplitude and phase response are

$$|H(\omega T)| = \frac{T}{2} \cot \frac{\omega T}{2} \quad (6.25)$$

and

$$H(\omega T) = -\bar{2} \quad (6.26)$$

The amplitude response approximates that of the ideal integrator much better than the rectangular filter and the phase response is exactly equal to that of the ideal response. The trapezoidal technique provides a very simple, effective recursive integrator.

Figure 6.6 shows a program that performs trapezoidal integration by direct implementation of the difference equation, Eq. (6.20).

```

/*****
 * Turbo C source code implementing a trapezoidal integrator
 *
 * Assume sampling period is 2 ms, the difference equation is:
 *
 *      y(nT) = y(nT - T) + 0.001x(nT) + 0.001x(nT - T)
 *
 *****/
float Trapezoid(signal)
float signal;          /* x[nT] */
{
    static float  x[2], y[2];
    int count;

    x[1]=x[0];          /* x[1]=x(nT-T) */
    x[0]=signal;        /* x[0]=x(nT) */

    y[0]=y[1]+0.001*x[0]+0.001*x[1]; /* difference equation */
    y[1]=y[0];          /* y[1]=y(nT - T) */

    return y[0];
}

```

Figure 6.6 A trapezoidal integrator written in the C language.

6.3.3 Simpson's rule integration

Simpson's rule is the most widely used numerical integration algorithm. Figure 6.4(c) shows that this approach approximates the signal corresponding to three input sequence points by a polynomial fit. The incremental area added for each new input point is the area under this polynomial. The difference equation is

$$y(nT) = y(nT - 2T) + \frac{T}{3} [x(nT) + 4x(nT - T) + x(nT - 2T)] \quad (6.27)$$

The transfer function is

$$H(z) = \frac{T}{3} \frac{1 + 4z^{-1} + z^{-2}}{1 - z^{-2}} \quad (6.28)$$

Figure 6.4(c) shows the block diagram and pole-zero plot for this filter. The amplitude and phase responses are

$$|H(\omega T)| = \frac{T}{2} \frac{2 + \cos \omega T}{\sin \omega T} \quad (6.29)$$

and

$$H(\omega T) = -\frac{1}{2} \quad (6.30)$$

Figure 6.5 shows the amplitude response of the Simpson's rule integrator. Like the trapezoidal technique, the phase response is the ideal $-\pi/2$. Simpson's rule approximates the ideal integrator for frequencies less than about $f_s/4$ better than the other techniques. However, it amplifies high-frequency noise near the foldover frequency. Therefore, it is a good approximation to the integral but is dangerous to use in the presence of noise. Integration of noisy signals can be accomplished better with trapezoidal integration.

6.4 DESIGN METHODS FOR TWO-POLE FILTERS

IIR filters have the potential for sharp rolloffs. Unlike the design of an FIR filter, which frequently is based on approximating an input sequence numerically, IIR filter design frequently starts with an analog filter that we would like to approximate. Their transfer functions can be represented by an infinite sum of terms or ratio of polynomials. Since they use feedback, they may be unstable if improperly designed. Also, they typically do not have linear phase response.

6.4.1 Selection method for r and θ

The general design equation has a standard recursive form for the four types of filters: low-pass, bandpass, high-pass, and band-reject:

$$H(z) = \frac{1 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} + b_2 z^{-2}} \quad (6.31)$$

where the zero locations are

$$z = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2}}{2} \quad (6.32)$$

and the pole locations are

$$z = \frac{b_1 \pm \sqrt{b_1^2 - 4b_2}}{2} \quad (6.33)$$

where

$$b_1 = 2r \cos \theta \quad \text{and} \quad b_2 = r^2 \quad (6.34)$$

Also

$$\theta = 2 \pi \frac{f_c}{f_s} \quad (6.35)$$

Figure 6.7 shows the structure of the two-pole filter. Assigning values to the numerator coefficients, a_1 and a_2 , as shown in Figure 6.8 establishes the placement of

the two zeros of the filter and defines the filter type. Values of b_1 and b_2 are determined by placing the poles at specific locations within the unit circle.

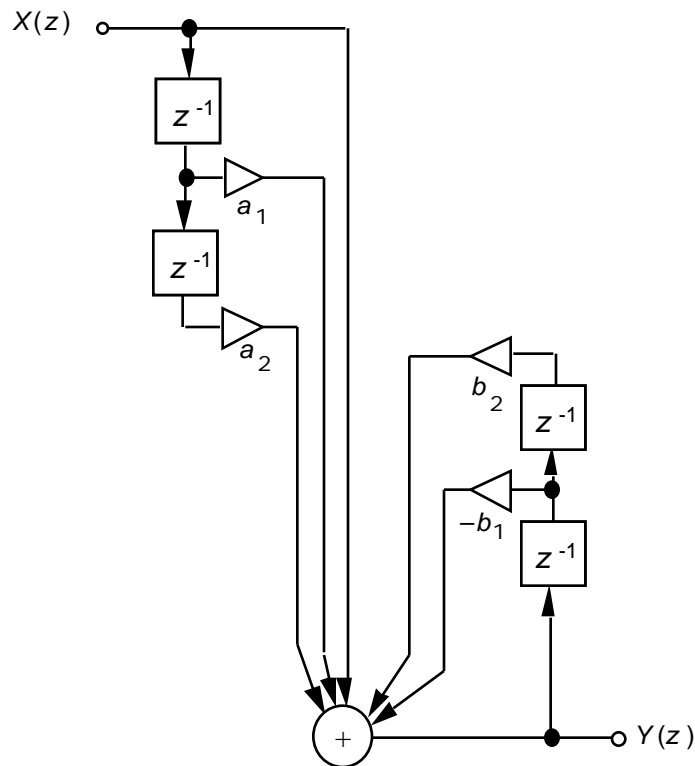


Figure 6.7 Block diagram showing two-pole filter structure.

We start the design by selecting the sampling frequency f_s , which must be at least twice the highest frequency contained in the input signal. Next we choose the critical frequency f_c . This is the cutoff frequency for low- and high-pass filters, the resonant frequency for a bandpass filter, and the notch frequency for a band-reject filter. These two choices establish θ , the angular location of the poles.

We then select r , the distance of the poles from the origin. This is the damping factor, given by

$$r = e^{-\alpha T} \tag{6.36}$$

We know from the amplitude response that underdamping occurs as the poles approach the unit circle (i.e., $r \rightarrow 1$ or $\alpha \rightarrow 0$) and overdamping results for poles near the origin (i.e., $r \rightarrow 0$ or $\alpha \rightarrow \infty$). Moving the poles by increasing r or θ causes underdamping, and decreasing either variable causes overdamping (Soderstrand, 1972). As we have done in other designs, we find the frequency response by substituting $e^{j\omega T}$ for z in the final transfer function.

	a_1	a_2
Low-pass	2	1
Bandpass	0	-1
High-pass	-2	1
Band-reject	$2\cos\theta$	1

Figure 6.8 Table of numerator coefficients for the two-pole filter implementations of Eq. (6.31).

We can write the difference equation for the two-pole filter by first rewriting Eq. (6.31):

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} + b_2 z^{-2}} \quad (6.37)$$

Then rearranging terms to find $Y(z)$, we get

$$Y(z) = b_1 Y(z)z^{-1} - b_2 Y(z)z^{-2} + X(z) + a_1 X(z)z^{-1} + a_2 X(z)z^{-2} \quad (6.38)$$

We can now directly write the difference equation using analogous discrete variable terms

$$y(nT) = b_1 y(nT - T) - b_2 y(nT - 2T) + x(nT) + a_1 x(nT - T) + a_2 x(nT - 2T) \quad (6.39)$$

Figure 6.9 shows the pole-zero plots and corresponding rubber membrane representations for each of the four filter types.

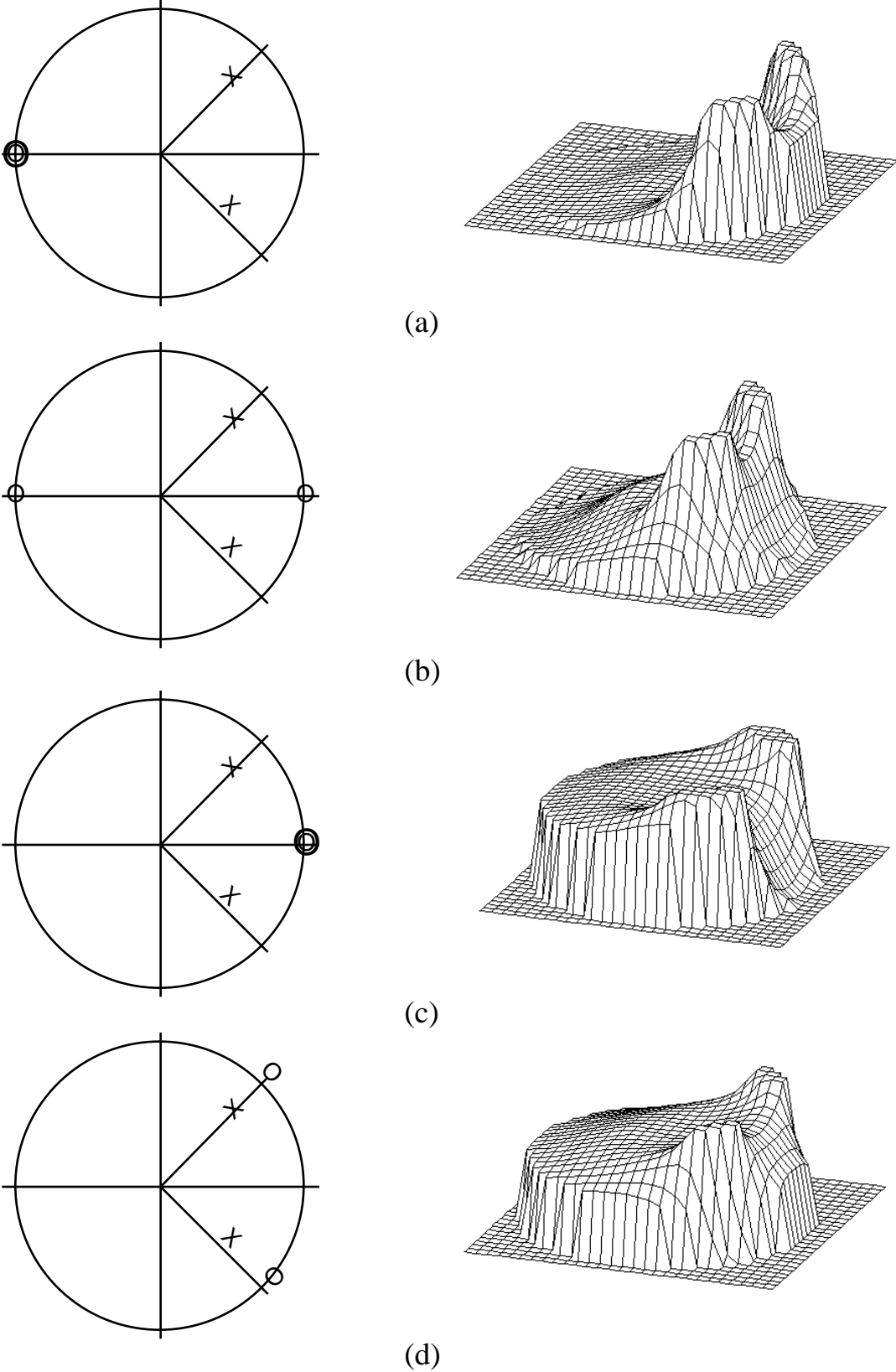


Figure 6.9 Pole-zero plots and rubber membrane views of elementary two-pole recursive digital filters. (a) Low-pass. (b) Bandpass. (c) High-pass. (d) Band-reject (notch).

Let us now design a filter and write a C-language program to implement it. First we choose a low-pass filter. This establishes the locations of the two zeros and establish the values for $a_1 = 2$ and $a_2 = 1$ (see Figure 6.8). Then we decide on the locations of the poles. For this design, we choose locations at $r = 0.5$ and $\theta = \pm 45^\circ$. This establishes the values for $b_1 = 0.707$ and $b_2 = 0.25$ [see Eq. (6.34)].

Software implementation of this digital filter is now fairly straightforward. We write a filter subroutine directly from the difference equation. Figure 6.10 shows how this low-pass filter is executed in five steps:

1. Pass in the current input signal.
2. Shift array elements to get delays of past input signal and store them.
3. Accumulate all past and/or current input and past output terms with multiplying coefficients.
4. Shift array elements to get delays of past output signal and store them.
5. Return with output signal.

```

/*****
 * Turbo C source code for implementing an IIR low-pass filter
 *
 * Assume we use the r and q method where r = 0.5, q = 45°.
 * The difference equation is:
 *
 *  $y(nT) = 0.707y(nT-T) - 0.25y(nT-2T) + x(nT) + 2x(nT-T) + x(nT-2T)$ 
 *  $y(nT) = 0.707*y[1] - 0.25*y[2] + x[0] + 2*x[1] + x[2]$ 
 *
 *****/
float LowPass(signal)
float signal;          /* x[nT] */
{
    static float  x[3], y[3];
    int count;

    for ( count=2; count>0; count-- )
        x[count]=x[count-1];          /* shift for x term delays */

    x[0]=signal;                    /* x[0] = x[nT] */
                                    /* difference equation */

    y[0]=0.707*y[1]-0.25*y[2]+x[0]+2.0*x[1]+x[2];

    for( count = 2; count > 0; count-- )
        y[count]=y[count-1];          /* shift for y term delays */

    return y[0];
}

```

Figure 6.10 An IIR low-pass filter written in the C language.

The same algorithm can be used for high-pass, bandpass, and band-reject filters as well as for integrators. The only disadvantage of the software implementation is its limited speed. However, this program will execute in real time on a modern PC with a math coprocessor for biomedical signals like the ECG.

These two-pole filters have operational characteristics similar to second-order analog filters. The rolloff is slow, but we can cascade several identical sections to improve it. Of course, we may have time constraints in a real-time system which limit the number of sections that can be cascaded in software implementations. In these cases, we design higher-order filters.

6.4.2 Bilinear transformation method

We can design a recursive filter that functions approximately the same way as a model analog filter. We start with the s -plane transfer function of the filter that we desire to mimic. We accomplish the basic digital design by replacing all the occurrences of the variable s by the approximation

$$s \approx \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (6.40)$$

This substitution does a nonlinear translation of the points in the s plane to the z plane. This warping of the frequency axis is defined by

$$\omega' = \frac{2}{T} \tan \frac{\omega T}{2} \quad (6.41)$$

where ω' is the analog domain frequency corresponding to the digital domain frequency ω . To do a bilinear transform, we first prewarp the frequency axis by substituting the relation for ω' for all the critical frequencies in the Laplace transform of the filter. We then replace s in the transform by its z -plane equivalent. Suppose that we have the transfer function

$$H(s) = \frac{\omega_c'}{s^2 + \omega_c'^2} \quad (6.42)$$

We substitute for ω_c' using Eq. (6.41) and for s with Eq. (6.40), to obtain

$$H(z) = \frac{\frac{2}{T} \tan \frac{\omega T}{2}}{\frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}^2 + \frac{2}{T} \tan \frac{\omega T}{2}} \quad (6.43)$$

This z transform is the description of a digital filter that performs approximately the same as the model analog filter. We can design higher-order filters with this technique.

6.4.3 Transform tables method

We can design digital filters to approximate analog filters of any order with filter tables such as those in Stearns (1975). These tables give the Laplace and z -transform equivalents for corresponding continuous and discrete-time functions. To illustrate the design procedure, let us consider the second-order filter of Figure 6.11(a). The analog transfer function of this filter is

$$H(s) = \frac{A}{LC} \frac{1}{s^2 + (R/L)s + 1/LC} \quad (6.44)$$

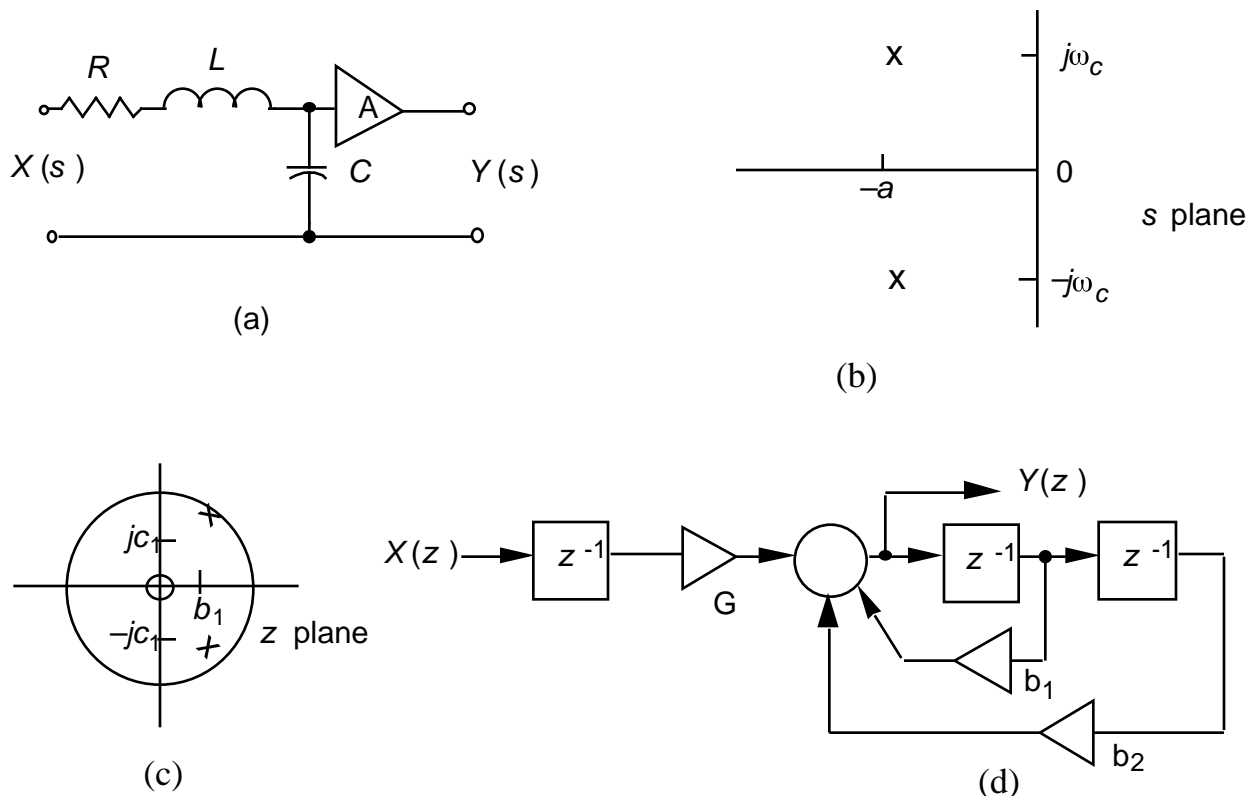


Figure 6.11 Second-order filter. (a) Analog filter circuit. (b) Transfer function pole-zero plot for the analog filter. (c) Pole-zero plot for digital version of the second-order filter. (d) Block diagram of the digital filter.

Solving for the poles, we obtain

$$s = -a \pm j\omega_c \quad (6.45)$$

where

$$a = \frac{R}{2L} \quad (6.46)$$

$$\omega_c = \frac{1}{LC} - \frac{R^2}{4L^2} \quad 1/2 \quad (6.47)$$

We can rewrite the transfer function as

$$H(s) = \frac{A}{LC} \frac{1}{(s+a)^2 + \omega_c^2} \quad (6.48)$$

Figure 6.11(b) shows the s -plane pole-zero plot. This s transform represents the continuous time function $e^{-at} \sin \omega_c t$. The z transform for the corresponding discrete-time function $e^{-naT} \sin n \omega_c T$ is in the form

$$H(z) = \frac{Gz^{-1}}{1 - b_1z^{-1} - b_2z^{-2}} \quad (6.49)$$

where

$$b_1 = 2e^{-aT} \cos \omega_c T \quad (6.50)$$

and

$$b_2 = -e^{-2aT} \quad (6.51)$$

Also

$$G = \frac{A}{\omega_c LC} e^{-aT} \sin \omega_c T \quad (6.52)$$

Variables a , ω_c , A , L , and C come from the analog filter design. This transfer function has one zero at $z = 0$ and two poles at

$$z = b_1 \pm j(b_1^2 + 4b_2)^{1/2} = b_1 \pm jc_1 \quad (6.53)$$

Figure 6.11(c) shows the z -plane pole-zero plot. We can find the block diagram by substituting the ratio $Y(z)/X(z)$ for $H(z)$ and collecting terms.

$$Y(z) = GX(z)z^{-1} + b_1Y(z)z^{-1} + b_2Y(z)z^{-2} \quad (6.54)$$

The difference equation is

$$y(nT) = Gx(nT - T) + b_1 y(nT - T) + b_2 y(nT - 2T) \quad (6.55)$$

From this difference equation we can directly write a program to implement the filter. We can also construct the block diagram as shown in Figure 6.11(d).

This transform-table design procedure provides a technique for quickly designing digital filters that are close approximations to analog filter models. If we have the transfer function of an analog filter, we still usually must make a substantial effort to implement the filter with operational amplifiers and other components.

However, once we have the z transform, we have the complete filter design specified and only need to write a straightforward program to implement the filter.

6.5 LAB: IIR DIGITAL FILTERS FOR ECG ANALYSIS

This lab provides experience with the use of IIR filters for processing ECGs. In order to design integrators and two-pole filters using UW DigiScope, select **(F)ilters** from the main menu, choose **(D)esign**, then **(I)IR**.

6.5.1 Integrators

This chapter reviewed three different integrators (rectangular, trapezoidal, and Simpson's rule). Which of these filters requires the most computation time? What do the unit-circle diagrams of these three filters have in common?

Run the rectangular integrator on an ECG signal (e.g., `ecg105.dat`). Explain the result. Design a preprocessing filter to solve any problem observed. Compare the output of the three integrators on appropriately processed ECG data with and without random noise. Which integrator is best to use for noisy signals?

6.5.2 Second-order recursive filters

Design three two-pole bandpass filters using $r = 0.7, 0.9,$ and 0.95 for a critical frequency of 17 Hz (sampling rate of 200 sps). Measure the Q for the three filters. Q is defined as the ratio of critical frequency to the 3-dB bandwidth (difference between the 3-dB frequencies). Run the three filters on an ECG file and contrast the outputs. What trade-offs must be considered when selecting the value of r ?

6.5.3 Transfer function (Generic)

The **(G)eneric** tool allows you to enter coefficients of a transfer function of the form

$$H(z) = \frac{a_0 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}}{1 + b_1z^{-1} + b_2z^{-2} + \dots + b_mz^{-m}} \quad \text{where } m \leq n$$

Using the example from the transform tables method discussed in section 6.4.3 with $R = 10 \text{ } \Omega$, $L = 10 \text{ mH}$, and $C = 1 \text{ } \mu\text{F}$, calculate and enter the coefficients of the resulting $H(z)$. Use a sampling rate of 200 sps. What is the Q of this filter?

6.5.4 Pole-zero placement

Create a 60-Hz notch FIR filter for a sampling rate of 180 sps by placing two zeros on the unit circle (i.e., $r = 1.0$) at angles of $\pm 120^\circ$ as shown in Figure 6.12(a). Create an IIR filter by adding poles inside the circle at angles of $\pm 110^\circ$ and $\pm 130^\circ$

at $r = 0.9$, as shown in Figure 6.12(b). Compare the amplitude response of this filter with that of the FIR filter. Measure the Q of both filters. Can you further increase the Q of the IIR filter? What happens to the phase response?

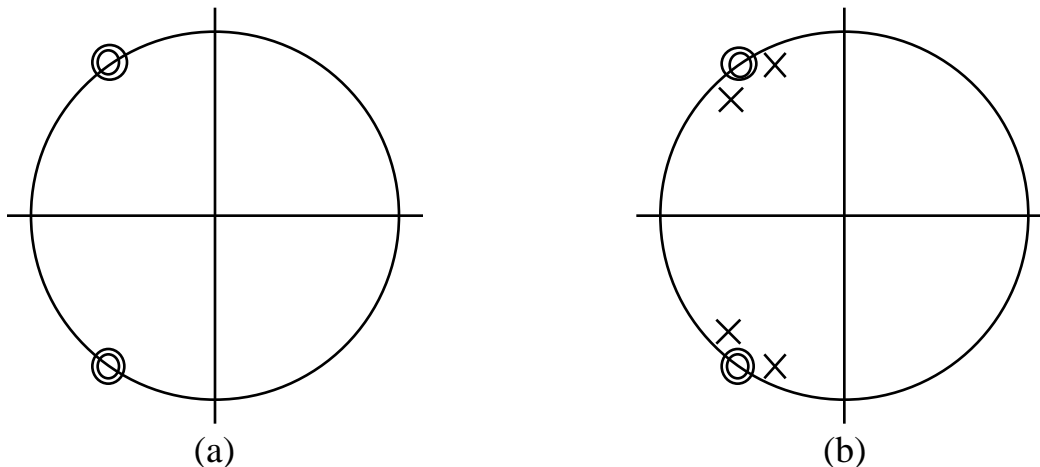


Figure 6.12 Notch filters. (a) FIR filter. (b) IIR filter.

6.6 REFERENCES

- Soderstrand, M. A. 1972. On-line digital filtering using the PDP-8 or PDP-12. *Computers in the Neurophysiology Laboratory*, **1**: 31-49. Maynard, MA: Digital Equipment Corporation.
- Stearns, S. D. 1975. *Digital Signal Analysis*. Rochelle Park, NJ: Hayden.
- Tompkins, W. J. and Webster, J. G. (eds.) 1981. *Design of Microcomputer-based Medical Instrumentation*. Englewood Cliffs, NJ: Prentice Hall.

6.7 STUDY QUESTIONS

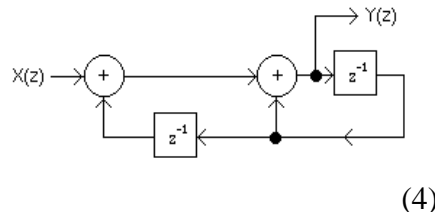
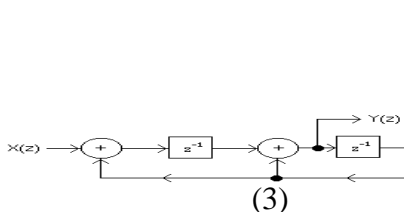
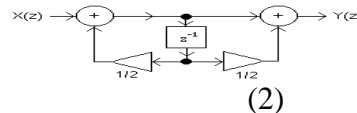
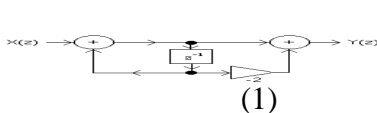
- 6.1 Describe the characteristics of the generic transfer function of recursive filters.
- 6.2 What is the difference between IIR filters and recursive filters?
- 6.3 How do you derive the frequency response of a recursive filter?
- 6.4 How do you write the difference equation from the transfer function of a recursive filter?
- 6.5 Design a low-pass recursive filter using the r and θ method.
- 6.6 Design a high-pass filter using bilinear transformation method.
- 6.7 Design a filter using transform tables method.
- 6.8 A digital filter has a *unit impulse* output sequence {i.e., 1, 0, 0, ...} when a *unit step* {i.e., 1, 1, 1, ...} is applied at its input. What is the transfer function of the filter?
- 6.9 Write a rectangular integrator in C .
- 6.10 Using the difference equation from the result of question 6.5, run the filtering program provided in section 6.4.
- 6.11 Draw the z -plane pole-zero plot for a filter with the z transform:

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

- 6.12 A filter has the following output sequence in response to a unit impulse: $\{-2, 4, -8, 16, \dots\}$. Write its z transform in closed form (i.e., as a ratio of polynomials). From the following list, indicate all the terms that describe this filter: recursive, nonrecursive, stable, unstable, FIR, IIR.
- 6.13 A digital filter has the following transfer function:

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

- (a) What traditional type of filter is this if $b =$ (1) 0.8; (2) -0.8 ; (3) 1; (4) -2 ; (5) $-1/2$?
 (b) If $b = -1/2$, what is the filter gain? (c) If $b = 1/2$, what is the difference equation for $y(nT)$?
- 6.14 The block diagrams for four digital filters are shown below. Write their (a) transfer functions, (b) difference equations.



- 6.15 Write the transfer function and comment on the stability of a filter with the following output sequence in response to a unit impulse: (a) $\{3, -6, 12, -24, \dots\}$. (b) $\{2, -1, 1, -1/2, 1/4, -1/8, \dots\}$.
- 6.16 A digital filter has a difference equation: $y(nT) = y(nT - 2T) + x(nT - T)$. What is its output sequence in response to a unit impulse applied to its input?
- 6.17 A filter's difference equation is: $y(nT) = x(nT) + 3y(nT - T)$. What is its output sequence in response to a unit impulse?
- 6.18 The difference equation of a filter is: $y(nT) = x(nT) + x(nT - 2T) + y(nT - 2T)$. Where are its poles and zeros located?
- 6.19 The general equation for a two-pole digital filter is:

$$H(z) = \frac{1 + a_1z^{-1} + a_2z^{-2}}{1 - b_1z^{-1} + b_2z^{-2}}$$

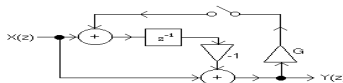
where $b_1 = 2r\cos(\theta)$ and $b_2 = r^2$

- (a) What traditional type of filter is this if (1) $a_1 = 2, a_2 = 1, -2 < b_1 < 2, 0 < b_2 < 1$, (2) $a_1 = -2\cos(\theta), a_2 = 1, r = 1, \theta = 60^\circ$? (b) What is the difference equation for $y(nT)$ if $a_1 = 0, a_2 = -1, r = 1/2, \theta = 60^\circ$?

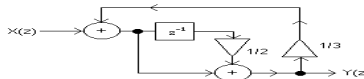
6.20 A digital filter has two zeros located at $z = 0.5$ and $z = 1$, and two poles located at $z = 0.5$ and $z = 0$. Write an expression for (a) its amplitude response as a function of a single trigonometric term, and (b) its phase response.

6.21 The block diagram of a digital filter is shown below.

- (a) Consider the case when the switch is **open**. (1) What is the filter's transfer function? (2) Write its difference equation. (3) What is the magnitude of its amplitude response at one-half the sampling frequency?
- (b) Consider the case when the switch is **closed**. (1) What is the filter's transfer function? (2) Write its difference equation. (3) If G is a negative fraction, what traditional filter type best describes this filter? (4) If $G = 1$, where are its poles and zeros located? (5) If a unit-amplitude step function is applied to the input with $G = 1$, what is the output sequence? (6) If a unit-amplitude step function is applied to the input with $G = -1$, what is the output sequence?



6.22 A digital filter has the block diagram shown below. (a) Write its transfer function. (b) Where are its poles and zeros located?



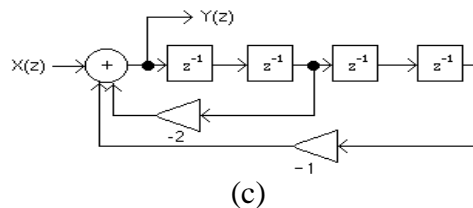
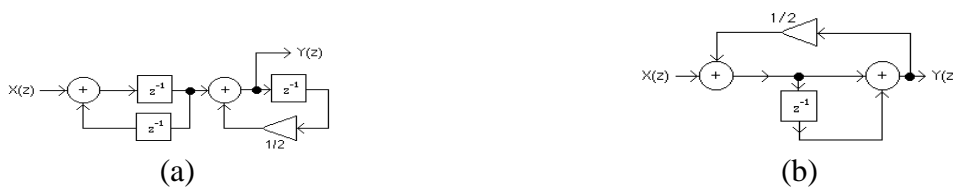
6.23 The difference equation for a filter is: $y(nT) = 2y(nT - T) + 2x(nT) + x(nT - T)$. Draw its z -plane pole-zero plot, indicating the locations of the poles and zeros.

6.24 Application of a unit **impulse** to the input of a filter produces the output sequence $\{1, 0, 1, 0, 1, 0, \dots\}$. What is the difference equation for this filter?

6.25 Application of a unit **step** to the input of a filter produces the output sequence $\{1, 1, 2, 2, 3, 3, \dots\}$. What is the difference equation for this filter? HINT: The z transform of a unit step is

$$\frac{1}{1 - z^{-1}}$$

6.26 Write the transfer functions of the following digital filters:



6.27 What is the phase response for a digital filter with the transfer function

$$H(z) = \frac{1 - z^{-6}}{1 + z^{-6}}$$

6.28 Write the amplitude response of a filter with the transfer function:

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

- 6.29 A filter has two poles at $z = 0.5 \pm j0.5$ and two zeros at $z = 0.707 \pm j0.707$. What traditional filter type best describes this filter?
- 6.30 A filter operating at a sampling frequency of 1000 samples/s has a pole at $z = 1/2$ and a zero at $z = 3$. What is the magnitude of its amplitude response at **dc**?
- 6.31 A filter is described by the difference equation: $y(nT) = x(nT) + x(nT - T) - 0.9y(nT - T)$. What is its transfer function?
- 6.32 A filter has the difference equation: $y(nT) = y(nT - 2T) + x(nT) + x(nT - T)$. What traditional filter type **best** describes this filter?
- 6.33 In response to a unit impulse applied to its input, a filter has the output sequence: $\{1, 1/2, 1/4, 1/8, \dots\}$. What is its transfer function?
- 6.34 The difference equation for a digital filter is: $y(nT) = x(nT) - ax(nT - T) - by(nT - T)$. Variables a and b are positive integers. What traditional type of filter is this if $a = 1$ and (a) $b = 0.8$, (b) $b > 1$?
- 6.35 Write the (a) amplitude response, (b) phase response, and (c) difference equation for a filter with the transfer function:

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

- 6.36 Write the (a) amplitude response, (b) phase response, and (c) difference equation for a filter with the transfer function:

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

- 6.37 A filter operating at a sampling frequency of 1000 samples/s has a pole at $z = 1$ and a zero at $z = 2$. What is the magnitude of its amplitude response at 500 Hz?
- 6.38 A filter operating at a sampling frequency of 200 samples/s has poles at $z = \pm j/2$ and zeros at $z = \pm 1$. What is the magnitude of its amplitude response at 50 Hz?
- 6.39 A filter is described by the difference equation: $2y(nT) + y(nT - T) = 2x(nT)$. What is its transfer function $H(z)$?
- 6.40 A filter has the difference equation: $y(nT) = y(nT - T) - y(nT - 2T) + x(nT) + x(nT - T)$. What is its transfer function?
- 6.41 In response to a unit impulse applied to its input, a filter has the output sequence: $\{1, 1, 1/2, 1/4, 1/8, \dots\}$. What is its transfer function?
- 6.42 A filter has a transfer function that is identical to the z transform of a unit step. A unit step is applied at its input. What is its output sequence?
- 6.43 A filter has a transfer function that is equal to the z transform of a ramp. A unit impulse is applied at its input. What is its output sequence? HINT: The equation for a ramp is $x(nT) = nT$, and its z transform is

$$X(z) = \frac{Tz^{-1}}{(1 - z^{-1})^2}$$

- 6.44 A ramp applied to the input of digital filter produces the output sequence: $\{0, T, T, T, T, \dots\}$. What is the transfer function of the filter?
- 6.45 A digital filter has a unit step {i.e., 1, 1, 1, 1, ...} output sequence when a unit impulse {i.e., 1, 0, 0, ...} is applied at its input. How is this filter **best** described?
- 6.46 A discrete impulse function is applied to the inputs of four different filters. For each of the output sequences that follow, state whether the filter is recursive or nonrecursive. (a) $\{1, 2, 3, 4, 5, 6, 0, 0, 0, \dots\}$, (b) $\{1, -1, 1, -1, 1, -1, \dots\}$, (c) $\{1, 2, 4, 8, 16, \dots\}$, (d) $\{1, 0.5, 0.25, 0.125, \dots\}$.

- 6.47 What similarities are common to all three integrator algorithms discussed in the text (i.e., rectangular, trapezoidal, and Simpson's rule)?
- 6.48 A differentiator is cascaded with an integrator. The differentiator uses the two-point difference algorithm:

$$H_1(z) = \frac{1 - z^{-1}}{T}$$

The integrator uses trapezoidal integration:

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

A unit impulse is applied to the input. What is the output sequence?

- 6.49 A differentiator is cascaded with an integrator. The differentiator uses the three-point central difference algorithm:

$$H_1(z) = \frac{1 - z^{-2}}{2T}$$

The integrator uses rectangular integration:

$$H_2(z) = T \frac{1}{1 - z^{-1}}$$

- (a) A unit impulse is applied to the input. What is the output sequence? (b) What traditional filter type best describes this filter?
- 6.50 A digital filter has two zeros located at $z = 0.5$ and $z = 1$, and a single pole located at $z = 0.5$. Write an expression for (a) its amplitude response as a function of a single trigonometric term, and (b) its phase response.
- 6.51 In response to a unit impulse applied to its input, a filter has the output sequence: $\{2, -1, 1, -1/2, 1/4, -1/8, \dots\}$. What is its transfer function?
- 6.52 The difference equation for a filter is: $y(nT - T) = x(nT - T) + 2x(nT - 4T) + 4x(nT - 10T)$. What is its transfer function, $H(z)$?
- 6.53 What is the transfer function $H(z)$ of a digital filter with the difference equation

$$y(nT - 2T) = y(nT - T) + x(nT - T) + x(nT - 4T) + x(nT - 10T)$$

- 6.54 A digital filter has the following output sequence in response to a *unit impulse*: $\{1, -2, 4, -8, \dots\}$. Where are its poles located?
- 6.55 A digital filter has a single zero located at $z = 0.5$ and a single pole located at $z = 0.5$. What are its amplitude and phase responses?
- 6.56 The difference equation for a filter is: $y(nT) = 2y(nT - T) + 2x(nT) + x(nT - T)$. What are the locations of its poles and zeros?
- 6.57 What traditional filter type best describes the filter with the z transform:

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

- 6.58 A discrete impulse function is applied to the inputs of four different filters. The output sequences of these filters are listed below. Which one of these filters has a pole outside the unit circle? (a) $\{1, 2, 3, 4, 5, 6, 0, 0, 0, \dots\}$ (b) $\{1, -1, 1, -1, 1, -1, \dots\}$ (c) $\{1, 2, 4, 8, 16, \dots\}$ (d) $\{1, 0.5, 0.25, 0.125, \dots\}$
- 6.59 Draw the block diagram of a filter that has the difference equation:

$$y(nT) = y(nT - T) + y(nT - 2T) + x(nT) + x(nT - T)$$

6.60 What is the transfer function $H(z)$ of a filter described by the difference equation:

$$y(nT) + 0.5y(nT - T) = x(nT)$$

6.61 A filter has an output sequence of $\{1, 5, 3, -9, 0, 0, \dots\}$ in response to the input sequence of $\{1, 3, 0, 0, \dots\}$. What is its transfer function?