22

Filters

22.1 Introduction ... **22**-1
22.2 Filter Classification ... **22**-2
22.3 The Filter Approximation Problem ... **22**-3
Butterworth Filters • Chebyshev Filters or Chebyshev I Filters
Elliptic or Cauer Filters • Bessel Filters
22.4 Design Examples for Passive and Active Filters ... **22**-8
Passive *R, L, C* Filter Design • Active Filter Design
22.5 Discrete-Time Filters ... **22**-15
22.6 Digital Filter Design Process ... **22**-16
22.7 FIR Filter Design ... **22**-17
Windowed FIR Filters • Optimum FIR Filters • Design of Narrowband FIR Filters
22.8 IIR Filter Design ... **22**-21
Design of Arbitrary IIR Filters • Cascade-Form IIR Filter Structures
22.9 Wave Digital Filters ... **22**-24
22.10 Antialiasing and Smoothing Filters ... **22**-25
22.11 Switched-Capacitor Filters ... **22**-27
22.12 Adaptive Filters ... **22**-27

Rahman Jamal
National Instruments Germany

Robert Steer
Frequency Devices

22.1 Introduction

In its broadest sense, a filter can be defined as a signal processing system whose output signal, usually called the *response,* differs from the input signal, called the *excitation,* such that the output signal has some prescribed properties. In more practical terms an electric filter is a device designed to suppress, pass, or separate a group of signals from a mixture of signals according to the specifications in a particular application. The application areas of filtering are manifold, for example to band-limit signals before sampling to reduce aliasing, to eliminate unwanted noise in communication systems, to resolve signals into their frequency components, to convert discrete-time signals into continuous-time signals, to demodulate signals, etc. Filters are generally classified into three broad classes: *continuous-time, sampled-data,* and *discrete-time* filters depending on the type of signal being processed by the filter. Therefore, the concept of signals are fundamental in the design of filters.

A *signal* is a function of one or more independent variables such as time, space, temperature, etc. that carries information. The independent variables of a signal can either be continuous or discrete. Assuming that the signal is a function of time, in the first case the signal is called continuous-time and in the second, discrete-time. A continuous-time signal is defined at every instant of time over a given interval, whereas a discrete-time signal is defined only at discrete-time instances. Similarly, the *values* of a signal can also be classified in either continuous or discrete.

In real-world signals, often referred to as analog signals, both amplitude and time are continuous. These types of signals cannot be processed by digital machines unless they have been converted into discrete-time signals. By contrast, a digital signal is characterized by discrete signal values, that are defined only at discrete points in time. Digital signal values are represented by a finite number of digits, which are usually binary coded. The relationship between a continuous-time signal and the corresponding discrete-time signal can be expressed in the following form:

$$x(kT) = x(t)_{t=kT}, \quad k = 0, 1, 2, \ldots, \tag{22.1}$$

where T is called the sampling period.

Filters can be classified on the basis of the input, output, and internal operating signals. A continuous data filter is used to process continuous-time or analog signals, whereas a digital filter processes digital signals. Continuous data filters are further divided into *passive* or *active* filters, depending on the type of elements used in their implementation. Perhaps the earliest type of filters known in the engineering community are *LC* filters, which can be designed by using discrete components like inductors and capacitors, or crystal and mechanical filters that can be implemented using *LC* equivalent circuits. Since no external power is required to operate these filters, they are often referred to as *passive* filters. In contrast, *active* filters are based on active devices, primarily *RC* elements, and amplifiers. In a sampled data filter, on the other hand, the signal is sampled and processed at discrete instants of time. Depending on the type of signal processed by such a filter, one may distinguish between an *analog sampled data* filter and a *digital* filter. In an analog sampled data filter the sampled signal can principally take any value, whereas in a digital filter the sampled signal is a digital signal, the definition of which was given earlier. Examples of analog sampled data filters are switched capacitor (SC) filters and charge-transfer device (CTD) filters made of capacitors, switches, and operational amplifiers.

22.2 Filter Classification

Filters are commonly classified according to the filter function they perform. The basic functions are: low-pass, high-pass, bandpass, and bandstop. If a filter passes frequencies from zero to its cutoff frequency Ω_c and stops all frequencies higher than the cutoff frequencies, then this filter type is called an ideal **lowpass filter.** In contrast, an ideal **high-pass filter** stops all frequencies below its cutoff frequency and passes all frequencies above it. Frequencies extending from Ω_1 to Ω_2 are passed by an ideal **bandpass filter,** while all other frequencies are stopped. An ideal bandstop filter stops frequencies from Ω_1 to Ω_2 and passes all other frequencies. Figure 22.1 depicts the magnitude functions of the four basic **ideal filter** types.

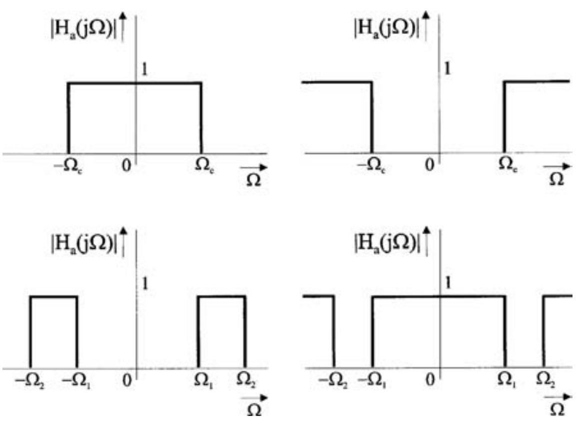

FIGURE 22.1 The magnitude function of an ideal filter is 1 in the passband and 0 in the stopband as shown for (a) low-pass, (b) high-pass, (c) bandpass, and (d) stopband filters.

So far we have discussed ideal filter characteristics having rectangular magnitude responses. These characteristics, however, are physically not realizable. As a consequence, the ideal response can only be approximated by some nonideal realizable system. Several classical approximation schemes have been developed, each of which satisfies a different criterion of optimization. This should be taken into account when comparing the performance of these filter characteristics.

22.3 The Filter Approximation Problem

Generally the input and output variables of a linear, time-invariant, causal filter can be characterized either in the time-domain through the convolution integral given by

$$y(t) = \int_0^t h_a(t-\tau)x(t)d\tau \tag{22.2}$$

or, equivalently, in the frequency-domain through the transfer function

$$H_a(s) = \frac{Y(s)}{X(s)} = \frac{\sum_{i=0}^{N} b_i s^i}{\sum_{i=0}^{N} a_i s^i} \Leftrightarrow H_a(s) = \frac{b_N}{a_N}\prod_{i=1}^{N}\left(\frac{s-s_{0i}}{s-s_{\infty i}}\right) \tag{22.3}$$

where $H_a(s)$ is the Laplace transform of the impulse response $h_a(t)$ and $X(s)$, $Y(s)$ are the Laplace transforms of the input signal $x(t)$ and the output or the filtered signal $y(t)$. $X(s)$ and $Y(s)$ are polynomials in $s = \sigma + j\Omega$ and the overall transfer function $H_a(s)$ is a real rational function of s with real coefficients. The zeroes of the polynomial $X(s)$ given by $s = s_{\infty i}$ are called the poles of $H_a(s)$ and are commonly referred to as the *natural frequencies* of the filter. The zeros of $Y(s)$ given by $s = s_{0i}$ which are equivalent to the zeroes of $H_a(s)$ are called the *transmission zeros* of the filter. Clearly, at these frequencies the filter output is zero for any finite input. Stability restricts the poles of $H_a(s)$ to lie in the left half of the s-plane excluding the $j\Omega$–axis, that is $\mathrm{Re}\{s_{\infty i}\} < 0$. For a stable transfer function $H_a(s)$ reduces to $H_a(j\Omega)$ on the $j\Omega$–axis, which is the continuous-time Fourier transform of the impulse response $h_a(t)$ and can be expressed in the following form:

$$H_a(j\Omega) = \left|H_a(j\Omega)\right| d^{j\theta(\Omega)} \tag{22.4}$$

where $\left|H_a(j\Omega)\right|$ is called the magnitude function and $\theta(\Omega) = \arg H_a(j\Omega)$ is the phase function. The gain magnitude of the filter expressed in decibels (dB) is defined by

$$\alpha(\Omega) = 20\ \log\left|H_a(j\Omega)\right| = 10\ \log\left|H_a(j\Omega)\right|^2 \tag{22.5}$$

Note that a filter specification is often given in terms of its attenuation, which is the negative of the gain function also given in decibels. While the specifications for a desired filter behavior are commonly given in terms of the loss response $\alpha(\Omega)$, the solution of the filter approximation problem is always carried out with the help of the characteristic function $C(j\Omega)$ giving

$$\alpha(\Omega) = 10\ \log\left[1 + \left|C(j\Omega)\right|^2\right] \tag{22.6}$$

Note that $\alpha(\Omega)$ is not a rational function, but $C(j\Omega)$ can be a polynomial or a rational function and approximation with polynomial or rational functions is relatively convenient. It can also be shown that

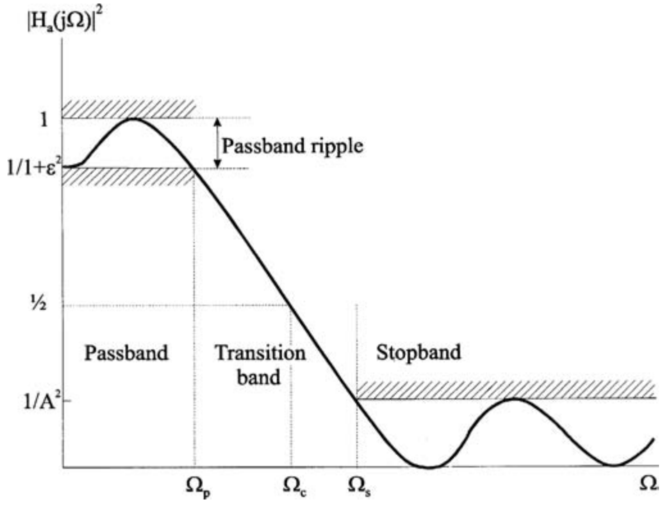

FIGURE 22.2 The squared magnitude function of an analog filter can have ripple in the passband and in the stopband.

frequency-dependent properties of $|C(j\Omega)|$ are in many ways identical to those of $\alpha(\Omega)$. The approximation problem consists of determining a desired response $|H_a(j\Omega)|$ such that the typical specifications depicted in Figure 22.2 are met. This so-called tolerance scheme is characterized by the following parameters:

- Ω_p Passband cutoff frequency (rad/s)
- Ω_s Stopband cutoff frequency (rad/s)
- Ω_c –3 dB cutoff frequency (rad/s)
- ε Permissible error in passband given by $\varepsilon = (10^{r/10} - 1)^{1/2}$, where r is the maximum acceptable attenuation in dB; note that $10 \log 1/(1 + \varepsilon^2)^{1/2} = -r$
- $1/A$ Permissible maximum magnitude in the stopband, i.e., $A = 10^{\alpha/20}$, where α is the minimum acceptable attenuation in dB; note that $20 \log (1/A) = -\alpha$.

The **passband** of a low-pass filter is the region in the interval $[0,\Omega_p]$ where the desired characteristics of a given signal are preserved. In contrast, the **stopband** of a low-pass filter (the region $[\Omega_s,\infty]$) rejects signal components. The **transition** band is the **region** between $(\Omega_x - \Omega_p)$, which would be 0 for an ideal filter. Usually, the amplitudes of the permissible ripples for the magnitude response are given in decibels.

The following sections review four different classical approximations: Butterworth, Chebyshev Type I, elliptic, and Bessel.

Butterworth Filters

The frequency response of an *N*th-order Butterworth low-pass filter is defined by the squared magnitude function

$$|H_a(j\Omega)|^2 = \frac{1}{1+(\Omega/\Omega_c)^{2N}} \tag{22.7}$$

It is evident from the Equation 22.7 that the Butterworth approximation has only poles, i.e., no finite zeros and yields a maximally flat response around zero and infinity. Therefore, this approximation is also called maximally flat magnitude (MFM). In addition, it exhibits a smooth response at all frequencies and a monotonic decrease from the specified cutoff frequencies.

Equation 22.7 can be extended to the complex *s*-domain, resulting in

$$H_a(s)H_a(-s) = \frac{1}{1+\left(s/j\Omega_c\right)^{2N}} \tag{22.8}$$

The poles of this function are given by the roots of the denominator

$$s_k = \Omega_c e^{j\pi\left[1/2+(2k+1)/2N\right]}, \quad k = 0, 1, \ldots, 2N-1 \tag{22.9}$$

Note that for any N, these poles lie on the unit circle of radius Ω_c in the *s*-plane. To guarantee stability, the poles that lie in the left half-plane are identified with $H_a(s)$. As an example, we will determine the transfer function corresponding to a third-order Butterworth filter, i.e., $N = 3$.

$$H_a(s)H_a(-s) = \frac{1}{1+\left(-s^2\right)^3} = \frac{1}{1-s^6} \tag{22.10}$$

The roots of denominator of Equation 22.10 are given by

$$s_k = \Omega_c e^{j\pi\left[1/2+(2k+1)/6\right]}, \quad k = 0, 1, 2, 3, 4, 5 \tag{22.11}$$

Therefore, we obtain

$$\begin{aligned}
s_0 &= \Omega_c e^{j\pi 2/3} = -1/2 + j\sqrt{2}/2 \\
s_1 &= \Omega_c e^{j\pi} = -1 \\
s_2 &= \Omega_c e^{j\pi 4/3} = -1/2 - j\sqrt{3}/2 \\
s_3 &= \Omega_c e^{j\pi 5/3} = 1/2 - j\sqrt{3}/2 \\
s_4 &= \Omega_c e^{j2\pi} = 1 \\
s_5 &= \Omega_c e^{j\pi/3} = 1/2 + j\sqrt{3}/2
\end{aligned} \tag{22.12}$$

The corresponding transfer function is obtained by identifying the left half-plane poles with $H_a(s)$. Note that for the sake of simplicity we have chosen $\Omega_c = 1$.

$$H_a(s) = \frac{1}{(s+1)\left(s+1/2-j\sqrt{3}/2\right)\left(s+1/2+j\sqrt{3}/2\right)} = \frac{1}{1+2s+2s^2+s^3} \tag{22.13}$$

Table 22.1 gives the Butterworth denominator polynomials up $N = 5$.

TABLE 22.1 Butterworth Denominator Polynomials

Order(N)	Butterworth Denominator Polynomials of $H(s)$
1	$s + 1$
2	$s^2 + \sqrt{2}s + 1$
3	$s^3 + 2s^2 + 2s + 1$
4	$s^4 + 2.6131s^3 + 3.4142s^2 + 2.6131s + 1$
5	$s^5 + 3.2361s^4 + 5.2361s^3 + 5.2361s^2 + 3.2361s + 1$

TABLE 22.2 Butterworth and Bessel Poles

	Butterworth Poles				Bessel Poles (–3 dB)			
	Re	Im(±*j*)			Re	Im(±*j*)		
N	*a*	*b*	Ω	*Q*	*a*	*b*	Ω	*Q*
1	–1.000	0.000	1.000	—	–1.000	0.000	1.000	—
2	–0.707	0.707	1.000	0.707	–1.102	0.636	1.272	0.577
3	–1.000	0.000	1.000	—	–1.323	0.000	1.323	—
	–0.500	0.866	1.000	1.000	–1.047	0.999	1.448	0.691
4	–0.924	0.383	1.000	0.541	–1.370	0.410	1.430	0.522
	–0.383	0.924	1.000	1.307	–0.995	1.257	1.603	0.805
5	–1.000	0.000	1.000	—	–1.502	0.000	1.502	—
	–0.809	0.588	1.000	0.618	–1.381	0.718	1.556	0.564
	–0.309	0.951	1.000	1.618	–0.958	1.471	1.755	0.916
6	–0.966	0.259	1.000	0.518	–1.571	0.321	1.604	0.510
	–0.707	0.707	1.000	0.707	–1.382	0.971	1.689	0.611
	–0.259	0.966	1.000	1.932	–0.931	1.662	1.905	1.023
7	–1.000	0.000	1.000	—	–1.684	0.000	1.684	—
	–0.901	0.434	1.000	0.555	–1.612	0.589	1.716	0.532
	–0.623	0.782	1.000	0.802	–1.379	1.192	1.822	0.661
	–0.223	0.975	1.000	2.247	–0.910	1.836	2.049	1.126
8	–0.981	0.195	1.000	0.510	–1.757	0.273	1.778	0.506
	–0.831	0.556	1.000	0.601	–1.637	0.823	1.832	0.560
	–0.556	0.831	1.000	0.900	–1.374	1.388	1.953	0.711
	–0.195	0.981	1.000	2.563	–0.893	1.998	2.189	1.226

Table 22.2 gives the Butterworth poles in real and imaginary components and in frequency and Q.

In the next example, the order N of a low-pass Butterworth filter is to be determined whose cutoff frequency (–3 dB) is Ω_c = 2 kHz and stopband attenuation is greater than 40 dB at Ω_s = 6 kHz. Thus the desired filter specification is

$$20 \log\left|H_a(j\Omega)\right| \le -40, \quad \Omega \ge \Omega_s \tag{22.14}$$

or equivalently,

$$\left|H_a(j\Omega)\right| \le 0.01, \quad \Omega \ge \Omega_s \tag{22.15}$$

It follows from Equation 22.7

$$\frac{1}{1+\left(\Omega_s/\Omega_c\right)^{2N}} = (0.01)^2 \tag{22.16}$$

Solving the above equation for N gives $N = 4.19$. Since N must be an integer, a fifth-order filter is required for this specification.

Chebyshev Filters or Chebyshev I Filters

The frequency response of an Nth-order Chebyshev low-pass filter is specified by the squared-magnitude frequency response function

$$\left|H_a(j\Omega)\right|^2 = \frac{1}{1+\varepsilon^2 T_N^2\left(\Omega/\Omega_p\right)} \tag{22.17}$$

where $T_N(x)$ is the Nth-order Chebyshev polynomial and ε is a real constant less than 1 which determines the ripple of the filter. Specifically, for nonnegative integers N, the Nth-order Chebyshev polynomial is given by

$$T_N(x) = \begin{cases} \cos\left(N \cos^{-1} x\right), & |x| \le 1 \\ \cosh\left(N \cosh^{-1} x\right), & |x| \ge 1 \end{cases} \tag{22.18}$$

High-order Chebyshev polynomials can be derived from the recursion relation

$$T_{N+1}(x) = 2xT_N(x) - T_{N-1}(x) \tag{22.19}$$

where $T_0(x) = 1$ and $T_1(x) = x$.

The Chebyshev approximation gives an **equiripple** characteristic in the passband and is maximally flat near infinity in the stopband. Each of the Chebyshev polynomials has real zeros that lie within the interval (–1,1) and the function values for $x \in [-1,1]$ do not exceed +1 and –1.

The pole locations for Chebyshev filter can be determined by generating the appropriate Chebyshev polynomials, inserting them into Equation 22.17, factoring, and then selecting only the left half plane roots. Alternatively, the pole locations P_k of an Nth-order Chebyshev filter can be computed from the relation, for $k = 1 \rightarrow N$

$$P_k = -\sin \Theta_k \sinh \beta + j \cos \Theta_k \cosh \beta \tag{22.20}$$

where $\Theta_k = (2k - 1)\pi/2N$ and $\beta = \sinh^{-1}(1/\varepsilon)$.

Note: P_{N-k+1} and P_k are complex conjugates and when N is odd there is one real pole at

$$P_{N+1} = -2 \sinh \beta$$

For the Chebyshev polynomials, Ω_p is the last frequency where the amplitude response passes through the value of ripple at the edge of the passband. For odd N polynomials, where the ripple of the Chebyshev polynomial is negative going, it is the $[-1/(1 + \varepsilon_2)]_{(1/2)}$ frequency and for even N, where the ripple is positive going, is the 0 dB frequency.

The Chebyshev filter is completely specified by the three parameters ε, Ω_p, and N. In a practical design application, ε is given by the permissible passband ripple and Ω_p is specified by the desired passband cutoff frequency. The order of the filter, i.e., N, is then chosen such that the stopband specifications are satisfied.

Elliptic or Cauer Filters

The frequency response of an Nth-order elliptic low-pass filter can be expressed by

$$\left|H_a(j\Omega)\right|^2 = \frac{1}{1 + \varepsilon^2 F_N^2\left(\Omega/\Omega_p\right)} \tag{22.21}$$

where $F_N(\cdot)$ is called the Jacobian elliptic function. The elliptic approximation yields an equiripple passband and an equiripple stopband. Compared with the same-order Butterworth or Chebyshev filters, the elliptic design provides the sharpest transition between the passband and the stopband. The theory of elliptic filters, initially developed by Cauer, is involved, therefore for an extensive treatment refer to Reference 1.

Elliptic filters are completely specified by the parameters ε, α, Ω_p, Ω_s, and N

where ε = passband ripple
a = stopband floor
Ω_p = the frequency at the edge of the passband (for a designated passband ripple)
Ω_s = the frequency at the edge of the stopband (for a designated stopband floor)
N = the order of the polynomial

In a practical design exercise, the desired passband ripple, stopband floor, and Ω_s are selected and N is determined and rounded up to the nearest integer value. The appropriate Jacobian elliptic function must be selected and $H_a(j\Omega)$ must be calculated and factored to extract only the left plane poles. For some synthesis techniques, the roots must expanded into polynomial form.

This process is a formidable task. While some filter manufacturers have written their own computer programs to carry out these calculations, they are not readily available. However, the majority of applications can be accommodated by use of published tables of the pole/zero configurations of low-pass elliptic transfer functions. An extensive set of such tables for a common selection of passband ripples, stopband floors, and shape factors is available in Reference 2.

Bessel Filters

The primary objectives of the preceding three approximations were to achieve specific loss characteristics. The phase characteristics of these filters, however, are nonlinear. The Bessel filter is optimized to reduce nonlinear phase distortion, i.e., a maximally flat delay. The transfer function of a Bessel filter is given by

$$H_a(s)=\frac{B_0}{B_N(s)}=\frac{B_0}{\sum_{k=0}^{N} B_k s^k},\quad B_k=\frac{(2N-k)!}{2^{N-k}k!(N-k)!}\quad k=0,1,\ldots,N \tag{22.22}$$

where $B_N(s)$ is the Nth-order Bessel polynomial. The overall squared-magnitude frequency response function is given by

$$\left|H_a(j\Omega)\right|^2=1-\frac{\Omega^2}{2N-1}+\frac{2(N-1)\Omega^4}{(2N-1)^2(2N-3)}+\cdots \tag{22.23}$$

To illustrate Equation 22.22 the Bessel transfer function for $N = 4$ is given below:

$$H_a(s)=\frac{105}{105+105s+45s^2+10s^3+s^4} \tag{22.24}$$

Table 22.2 lists the factored pole frequencies as real and imaginary parts and as frequency and Q for Bessel transfer functions that have been normalized to Ω_c = –3 dB.

22.4 Design Examples for Passive and Active Filters

Passive *R*, *L*, *C* Filter Design

The simplest and most commonly used passive filter is the simple, first-order ($N = 1$) R–C filter shown in Figure 22.3. Its transfer function is that of a first-order Butterworth low-pass filter. The transfer function and –3 dB Ω_c are

$$H_a(s)=\frac{1}{RCs+1}\quad\text{where}\quad\Omega_c=\frac{1}{RC} \tag{22.25}$$

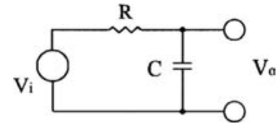

FIGURE 22.3 A passive first-order *RC* filter can serve as an antialiasing filter or to minimize high-frequency noise.

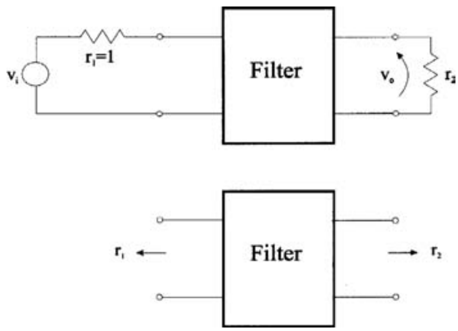

FIGURE 22.4 A passive filter can have the symbolic representation of a doubly terminated filter.

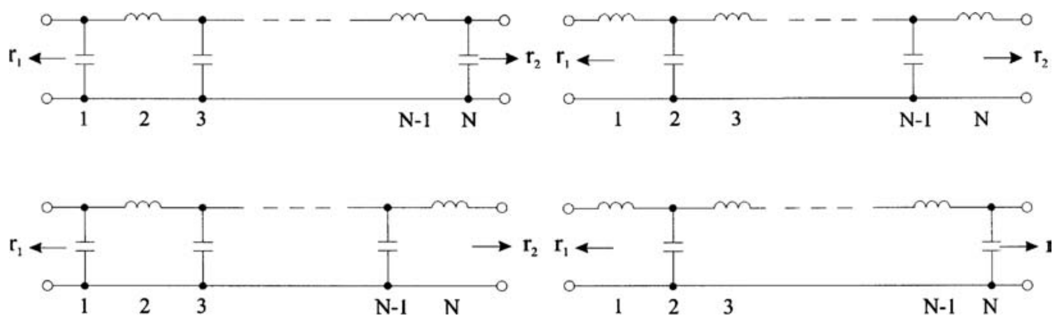

FIGURE 22.5 Even and odd N passive all-pole filter networks can be realized by several circuit configurations (N odd, above; N even, below).

While this is the simplest possible filter implementation, both source and load impedance change the dc gain and/or corner frequency and its rolloff rate is only first order, or –6 dB/octave.

To realize higher-order transfer functions, passive filters use R, L, C elements usually configured in a ladder network. The design process is generally carried out in terms of a doubly terminated two-port network with source and load resistors R_1 and R_2 as shown in Figure 22.4. Its symbolic representation is given below.

The source and load resistors are normalized in regard to a reference resistance $R_B = R_1$, i.e.,

$$r_i = \frac{R_1}{R_B} = 1, \quad r_2 = \frac{R_2}{R_B} = \frac{R_2}{R_1} \tag{22.26}$$

The values of L and C are also normalized in respect to a reference frequency to simplify calculations. Their values can be easily scaled to any desired set of actual elements.

$$l_v = \frac{\Omega_B L_v}{R_B}, \quad c_v = \Omega_B C_v R_B \tag{22.27}$$

Low-pass filters, whose magnitude-squared functions have no finite zero, i.e., whose characteristic functions $C(j\Omega)$ are polynomials, can be realized by lossless ladder networks consisting of inductors as the series elements and capacitors as the shunt elements. These types of approximations, also referred to as *all-pole approximations*, include the previously discussed Butterworth, Chebyshev Type I, and Bessel filters. Figure 22.5 shows four possible ladder structures for even and odd N, where N is the filter order.

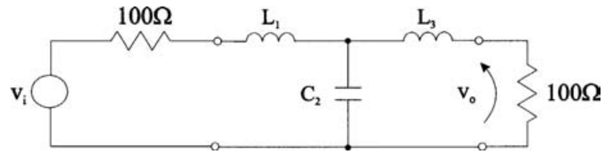

FIGURE 22.6 A third-order passive all-pole filter can be realized by a doubly terminated third-order circuit.

TABLE 22.3 Element Values for Low-Pass Filter Circuits

		N = 2, Element Number		N = 3, Element Number		
Filter Type	r_2	1	2	1	2	3
Butterworth	∞	1.4142	0.7071	1.5000	1.3333	0.5000
	1	1.4142	1.4142	1.0000	2.0000	1.0000
Chebyshev type I	∞	0.7159	0.4215	1.0895	1.0864	0.5158
0.1-dB ripple	1	—	—	1.0316	1.1474	1.0316
Chebyshev type I	∞	0.9403	0.7014	1.3465	1.3001	1.5963
0.5 dB ripple	1	—	—	1.5963	1.0967	1.5963
Bessel	∞	1.0000	0.3333	0.8333	0.4800	0.1667
	1	1.5774	0.4227	1.2550	0.5528	0.1922

In the case of doubly terminated Butterworth filters, the normalized values are precisely given by

$$a_v = 2\sin\left(\frac{(2v-1)\pi}{2N}\right), \quad v = 1, \ldots, N \tag{22.28}$$

where a_v is the normalized L or C element value. As an example we will derive two possible circuits for a doubly terminated Butterworth low-pass of order 3 with $R_B = 100\ \Omega$ and a cutoff frequency $\Omega_c = \Omega_B =$ 10 kHz. The element values from Equation 22.28 are

$$\begin{aligned}
l_1 &= 2\sin\left(\frac{(2-1)\pi}{6}\right) = 1 \Rightarrow L_1 = \frac{R_B}{\Omega_c} = 1.59\ \text{mH} \\
c_2 &= 2\sin\left(\frac{(4-1)\pi}{6}\right) = 2 \Rightarrow C_2 = \frac{2}{\Omega_c R_B} = 3.183\ \text{nF} \\
l_3 &= 2\sin\left(\frac{(6-1)\pi}{6}\right) = 1 \Rightarrow L_3 = \frac{R_B}{\Omega_c} = 1.59\ \text{mH}
\end{aligned} \tag{22.29}$$

A possible realization is shown in Figure 22.6.

Table 22.3 gives normalized element values for the various all-pole filter approximations discussed in the previous section up to order 3 and is based on the following normalization:

1. $r_1 = 1$;
2. All the cutoff frequencies (end of the ripple band for the Chebyshev approximation) are Ω_c = 1 rad/s;
3. r_2 is either 1 or ∞, so that both singly and doubly terminated filters are included.

The element values in Table 22.3 are numbered from the source end in the same manner as in Figure 22.4. In addition, empty spaces indicate unrealizable networks. In the case of the Chebyshev filter, the amount of ripple can be specified as desired, so that in the table only a selective sample can be given. Extensive tables of prototype element values for many types of filters can be found in Reference 4.

The example given above, of a Butterworth filter of order 3, can also be verified using Table 22.3. The steps necessary to convert the normalized element values in the table into actual filter values are the same as previously illustrated.

In contrast to all-pole approximations, the characteristic function of an elliptic filter function is a rational function. The resulting filter will again be a ladder network but the series elements may be parallel combinations of capacitance and inductance and the shunt elements may be series combinations of capacitance and inductance.

Figure 22.5 illustrates the general circuits for even and odd N, respectively. As in the case of all-pole approximations, tabulations of element values for normalized low-pass filters based on elliptic approximations are also possible. Since these tables are quite involved the reader is referred to Reference 4.

Active Filter Design

Active filters are widely used and commercially available with cutoff frequencies from millihertz to megahertz. The characteristics that make them the implementation of choice for several applications are small size for low-frequency filters because they do not use inductors; precision realization of theoretical transfer functions by use of precision resistors and capacitors; high input impedance that is easy to drive and for many circuit configurations the source impedance does not effect the transfer function; low-output impedance that can drive loads without effecting the transfer function and can drive the transient, switched-capacitive, loads of the input stages of A/D converters and low (N+THD) performance for pre-A/D antialiasing applications (as low as –100 dBc).

Active filters use R, C, A (operational amplifier) circuits to implement polynomial transfer functions. They are most often configured by cascading an appropriate number of first- and second-order sections.

The simplest first-order ($N = 1$) active filter is the first-order passive filter of Figure 22.3 with the addition of a unity gain follower amplifier. Its cutoff frequency (Ω_c) is the same as that given in Equation 22.25. Its advantage over its passive counterpart is that its operational amplifier can drive whatever load that it can tolerate without interfering with the transfer function of the filter.

The vast majority of higher-order filters have poles that are not located on the negative real axis in the s-plane and therefore are in complex conjugate pairs that combine to create second-order pole pairs of the form:

$$H(s) = s^2 + \frac{\omega_p}{Q} s + \omega_p^2 \Leftrightarrow s^2 + 2as + a^2 + b^2 \tag{22.30}$$

where $p_1, p_2 = a \pm jb$

$\omega_p^2 = a^2 + b^2$

$Q = \dfrac{\omega_p}{2a} = \dfrac{\sqrt{\left(a^2 + b^2\right)}}{2a}$

The most commonly used two-pole active filter circuits are the *Sallen and Key* low-pass resonator, the *multiple feedback* bandpass, and the *state variable* implementation as shown in Figure 22.7a, b, and c. In the analyses that follow, the more commonly used circuits are used in their simplest form. A more comprehensive treatment of these and numerous other circuits can be found in Reference 20.

The Sallen and Key circuit of Figure 22.7a is used primarily for its simplicity. Its component count is the minimum possible for a two-pole active filter. It cannot generate stopband zeros and therefore is limited in its use to monotonic roll-off transfer functions such as Butterworth and Bessel filters. Other limitations are that the phase shift of the amplifier reduces the Q of the section and the capacitor ratio becomes large for high-Q circuits. The amplifier is used in a follower configuration and therefore is subjected to a large common mode input signal swing which is not the best condition for low distortion performance. It is recommended to use this circuit for a section $Q < 10$ and to use an amplifier whose gain bandwidth product is greater than 100 f_p.

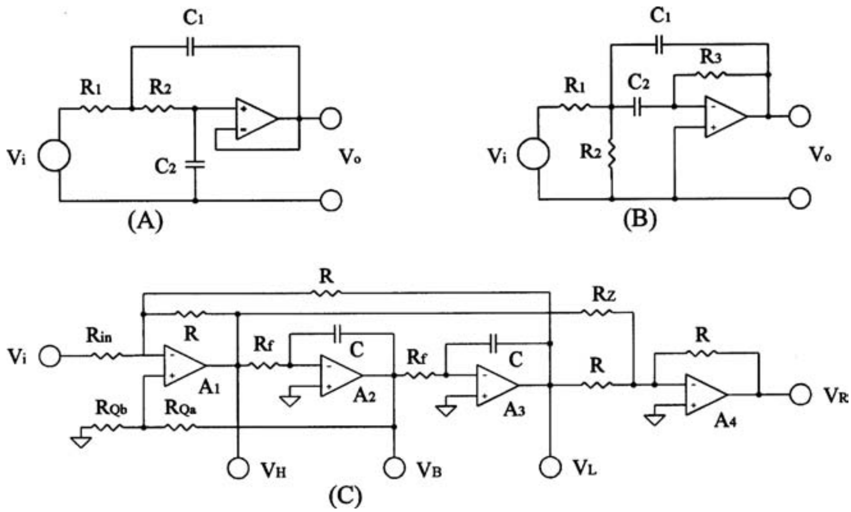

FIGURE 22.7 Second-order active filters can be realized by common filter circuits: (A) Sallen and Key low-pass, (B) multiple feedback bandpass, (C) state variable.

The transfer function and design equations for the Sallen and Key circuit of Figure 22.7a are

$$H(s) = \frac{\dfrac{1}{R_1R_2C_1C_2}}{s^2 + \dfrac{1}{R_1C_2}s + \dfrac{1}{R_1R_2C_1C_2}} = \frac{\omega_p^2}{s^2 + \dfrac{\omega_p}{Q}s + \omega_p^2} \tag{22.31}$$

from which obtains

$$\omega^2 = \frac{1}{R_1R_2C_1C_2}, \quad Q = \omega_p R_1C_2 = \sqrt{\frac{R_1C_2}{R_2C_1}} \tag{22.32}$$

$$R_1, R_2 = \frac{1}{4\pi f_p QC_2}\left[1 \pm \frac{4Q^2C_2}{C_1}\right] \tag{22.33}$$

which has valid solutions for

$$\frac{C_1}{C_2} \geq 4Q^2 \tag{22.34}$$

In the special case where

$$\begin{gathered} R_1 = R_2 = R, \text{ then} \\ C = 1/2\pi R f_p, \; C_1 = 2QC, \text{ and } C_2 = C/2Q \end{gathered} \tag{22.35}$$

The design sequence for Sallen and Key low-pass of Figure 22.7a is as follows:

For a required f_p and Q, select C_1, C_2 to satisfy Equation 22.34. Compute R_1, R_2 from Equation 22.33 (or Equation 22.35 if R_1 is chosen to equal R_2) and scale the values of C_1 and C_2 and R_1 and R_2 to desired impedance levels.

As an example, a three-pole low-pass active filter is shown in Figure 22.8. It is realized with a buffered single-pole *RC* low-pass filter section in cascade with a two-pole Sallen and Key section.

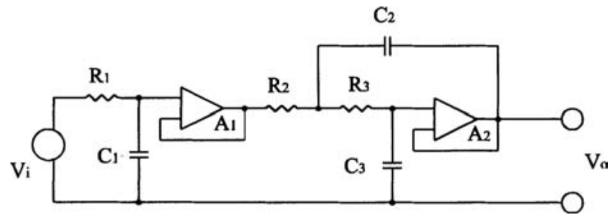

FIGURE 22.8 A three-pole Butterworth active can be configured with a buffered first-order *RC* in cascade with a two-pole Sallen and Key resonator.

To construct a three-pole Butterworth filter, the pole locations are found in Table 22.2 and the element values in the sections are calculated from Equation 22.25 for the single real pole and in accordance with the Sallen and Key design sequence listed above for the complex pole pair.

From Table 22.2, the normalized pole locations are

$$f_{p1} = 1.000, \qquad f_{p2} = 1.000, \qquad \text{and} \qquad Q_{p2} = 1.000$$

For a cutoff frequency of 10 kHz and if it is desired to have an impedance level of 10 kΩ, then the capacitor values are computed as follows:

For $R_1 = 10$ kΩ:

from Equation 22.25, $C_1 = \dfrac{1}{2\pi R_1 r_{p1}} = \dfrac{1}{2\pi(10{,}000)(10{,}000)} = \dfrac{10^{-6}}{200\pi} = 0.00159\ \mu\text{F}$

For $R_2 = R_3 = R = 10$ kΩ:

from Equation 22.35, $C = \dfrac{1}{2\pi R f_{p2}} = \dfrac{1}{2\pi(10{,}000)(10{,}000)} = \dfrac{10^{-6}}{200\pi} = 0.00159\ \mu\text{F}$

from which

$$C_2 = 2QC = 2(0.00159)\ \mu\text{F} = 0.00318\ \mu\text{F}$$

$$C_3 = C/2Q = 0.5(0.00159)\ \mu\text{F} = 0.000795\ \mu\text{F}$$

The *multiple feedback* circuit of Figure 22.7b is a minimum component count, two-pole (or one-pole pair), bandpass filter circuit with user definable gain. It cannot generate stopband zeros and therefore is limited in its use to monotonic roll-off transfer functions. Phase shift of its amplifier reduces the Q of the section and shifts the f_p. It is recommended to use an amplifier whose open loop gain at f_p is $> 100Q^2H_p$.

The design equations for the *multiple feedback* circuit of Figure 22.4b are

$$H(s) = \frac{\dfrac{s}{R_1C_1}}{s^2 + \dfrac{s}{R_3}\left(\dfrac{1}{C_1} + \dfrac{1}{C_2}\right) + \dfrac{(R_1 + R_2)}{R_1R_2R_3C_1C_2}} = -\frac{\dfrac{s\omega_p H_p}{Q}}{s^2 + \dfrac{s\omega_p}{Q} + \omega_p^2} \tag{22.36}$$

when $s = j\omega_p$, the gain H_p is

$$H_p = \frac{R_3C_2}{R_1(C_1 + C_2)} \tag{22.37}$$

From Equation 22.36 and 22.37 for a required set of ω_p, Q, and H_p:

$$R_1 = \frac{Q}{C_1 H_p \omega_p}, \quad R_2 = \frac{Q}{\omega_p}\left(\frac{1}{Q^2(C_1 + C_2) - H_p C_1}\right), \quad R_3 = \frac{R_1 H_p (C_1 + C_2)}{C_2} \tag{22.38}$$

For R_2 to be realizable,

$$Q^2(C_1 + C_2) \geq H_p C_1 \tag{22.39}$$

The design sequence for a *multiple feedback* bandpass filter is as follows:

Select C_1 and C_2 to satisfy Equation 22.39 for the H_p and Q required. Compute R_1, R_2, and R_3. Scale R_1, R_2, R_3, C_1, and C_2 as required to meet desired impedance levels.

Note that it is common to use $C_1 = C_2 = C$ for applications where $H_p = 1$ and $Q > 0.707$.

The *state variable* circuit of Figure 22.7c is the most widely used active filter circuit. It is the basic building block of programmable active filters and of switched capacitor designs. While it uses three or four amplifiers and numerous other circuit elements to realize a two-pole filter section, it has many desirable features. From a single input it provides low-pass (V_L), high-pass (V_H), and bandpass (V_B) outputs and by summation into an additional amplifier (A_4) (or the input stage of the next section) a band reject (V_R) or stopband zero can be created. Its two integrator resistors connect to the virtual ground of their amplifiers (A_2, A_3) and therefore have no signal swing on them. Therefore, programming resistors can be switched to these summing junctions using electronic switches. The sensitivity of the circuit to the gain and phase performance of its amplifiers is more than an order of magnitude less than single amplifier designs. The open-loop gain at f_p does not have to be multiplied by either the desired Q or the gain at dc or f_p. Second-order sections with Q up to 100 and f_p up to 1 MHz can be built with this circuit.

There are several possible variations of this circuit that improve its performance at particular outputs. The input can be brought into several places to create or eliminate phase inversions; the damping feedback can be implemented in several ways other than the R_{Qa} and R_{Qb} that are shown in Figure 22.7c and the f_p and Q of the section can be adjusted independently from one another. dc offset adjustment components can be added to allow the offset at any one output to be trimmed to zero.

For simplicity of presentation, Figure 22.7c makes several of the resistors equal and identifies others with subscripts that relate to their function in the circuit. Specifically, the feedback amplifier A_1, that generates the V_H output has equal feedback and input resistor from the V_L feedback signal to create unity gains from that input. Similarly, the "zero summing" amplifier, A_4 has equal resistors for its feedback and input from V_L to make the dc gain at the V_R output the same as that at V_L. More general configurations with all elements included in the equation of the transfer function are available in numerous reference texts including Reference 20.

The *state variable* circuit, as configured in Figure 22.7c, has four outputs. Their transfer functions are

$$V_L(s) = -\frac{R}{R_i (R_f C)^2}\left(\frac{1}{D(s)}\right) \tag{22.40a}$$

$$V_B(s) = \frac{R}{R_i}\left(\frac{\dfrac{s}{(R_f C)}}{D(s)}\right) \tag{22.40b}$$

$$V_{\mathrm{H}}(s) = -\frac{R}{R_{\mathrm{i}}}\left(\frac{s^2}{D(s)}\right) \tag{22.40c}$$

$$V_{\mathrm{R}}(s) = \frac{R}{R_{\mathrm{i}}(R_{\mathrm{f}}C)^2}\left(\frac{\left(\frac{R_{\mathrm{z}}}{R}\right)s^2 + 1}{D(s)}\right) \tag{22.40d}$$

where

$$D(s) = s^2 + \frac{a}{R_{\mathrm{f}}C}s + \frac{1}{(R_{\mathrm{f}}C)^2} = s^2 + \frac{\omega_{\mathrm{p}}}{Q}s + \omega_{\mathrm{p}}^2 \quad a = \frac{R_{\mathrm{Qb}}}{(R_{\mathrm{Qa}} + R_{\mathrm{Qb}})}\left(2 + \frac{R}{R_{\mathrm{i}}}\right) \tag{22.41}$$

Note that the dc gain at the low-pass output is

$$V_{\mathrm{L}}(0) = -\frac{R}{R_{\mathrm{i}}} \tag{22.42a}$$

from which obtains

$$\omega_{\mathrm{p}} = \frac{1}{R_{\mathrm{f}}C} \quad \text{and} \quad \frac{1}{Q} = \frac{R_{\mathrm{Qb}}}{(R_{\mathrm{Qa}} + R_{\mathrm{Qb}})}\left(2 + \frac{R}{R_{\mathrm{i}}}\right) \tag{22.42b}$$

The design sequence for the state variable filter of Figure 22.7c is

> Select the values of R_{f} and C to set the frequency ω_{p}, the values of R_{i} for the desired dc gain and R_{Qa} and R_{Qb} for the desired Q and dc gain.

22.5 Discrete-Time Filters

A digital filter is a circuit or a computer program that computes a discrete output sequence from a discrete input sequence. Digital filters belong to the class of discrete-time LTI (linear time invariant) systems, which are characterized by the properties of causality, recursibility, and stability, and may be characterized in the time domain by their impulse response and in the transform domain by their transfer function. The most general case of a discrete-time LTI system with the input sequence denoted by $x(kT)$ and the resulting output sequence $y(kT)$ can be described by a set *of linear difference equations with constant coefficients*.

$$y(kT) = \sum_{\mu=0}^{N} b_\mu x(kT - \mu T) - \sum_{\mu=1}^{N} a_\mu y(kT - \mu T) \tag{22.43}$$

where $a_0 = 1$. An equivalent relation between the input and output variables can be given through the convolution sum in terms of the impulse response sequence $h(kT)$:

$$y(kT) = \sum_{\mu=0}^{N} h(kT)x(kT - \mu T) \tag{22.44}$$

The corresponding transfer function is given by

$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{\mu=0}^{N} b_\mu z^{-\mu}}{1+\sum_{\mu=1}^{N} a_\mu z^{-\nu}} \Leftrightarrow H(z)=b_0\prod_{\mu=1}^{N}\left(\frac{z-z_{0\mu}}{z-z_{\infty\mu}}\right) \tag{22.45}$$

where $H(z)$ is the z-transform of the impulse response $h(kT)$ and $X(z)$, $Y(z)$ are the z-transform of the input signal $x(kT)$ and the output or the filtered signal $y(kT)$. As can be seen from Equation 22.44, if for at least one μ, $a_\mu \neq 0$, the corresponding system is recursive; its impulse response is of infinite duration — **infinite impulse response (IIR) filter**. If $a_\mu = 0$, the corresponding system is nonrecursive — **finite impulse response (FIR) filter**; its impulse response is of finite duration and the transfer function $H(z)$ is a polynomial in z^{-1}. The zeros of the polynomial $X(z)$ given by $z = z_{\infty i}$ are called the poles of $H(z)$ and are commonly referred to as the *natural frequencies* of the filter. The condition for the stability of the filter is expressed by the constraint that all the poles of $H(z)$ should lie inside the unit circle, that is $|z_{\infty i}| < 1$. The zeros of $Y(z)$ given by $z = z_{0t}$ which are equivalent to the zeros of $H(z)$ are called the *transmission zeros* of the filter. Clearly, at these frequencies the output of the filter is zero for any finite input.

On the unit circle, the transfer function frequency $H(z)$ reduces to the frequency response function $H(e^{j\omega T})$, the discrete-time Fourier transform of $h(kT)$, which in general is complex and can be expressed in terms of magnitude and phase

$$H(e^{j\omega T})=\left|H(e^{j\omega T})\right|e^{j\theta(\omega)} \tag{22.46}$$

The gain function of the filter is given as

$$\alpha(\Omega)=20\log_{10}\left|H(e^{j\omega T})\right| \tag{22.47}$$

It is also common practice to call the negative of the gain function the attenuation. Note that the attenuation is a positive number when the magnitude response is less than 1.

Figure 22.9 gives a block diagram realizing the difference equation of the filter, which is commonly referred to as the *direct-form I* realization. Notice that the element values for the multipliers are obtained directly from the numerator and denominator coefficients of the transfer function. By rearranging the structure in regard to the number of delays, one can obtain the canonic structure called *direct-form II* shown in Figure 22.10, which requires the minimum number of delays.

Physically, the input numbers are samples of a continuous signal and real-time digital filtering involves the computation of the iteration of Equation 22.43 for each incoming new input sample. Design of a filter consists of determining the constants a_μ and b_μ that satisfies a given filtering requirement. If the filtering is performed in real time, then the right side of Equation 22.46 must be computed in less than the sampling interval T.

22.6 Digital Filter Design Process

The digital filter design procedure consists of the following basic steps:

1. Determine the desired response. The desired response is normally specified in the frequency domain in terms of the desired magnitude response and/or the desired phase response.
2. Select a class of filters (e.g., linear-phase FIR filters or IIR filters) to approximate the desired response.
3. Select the best member in the filter class.

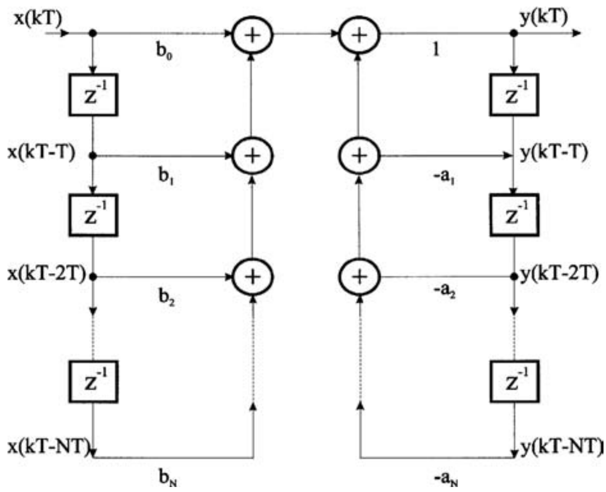

FIGURE 22.9 The difference equation of a digital filter can be realized by a direct-form I implementation that uses separate delay paths for the *X* and *Y* summations.

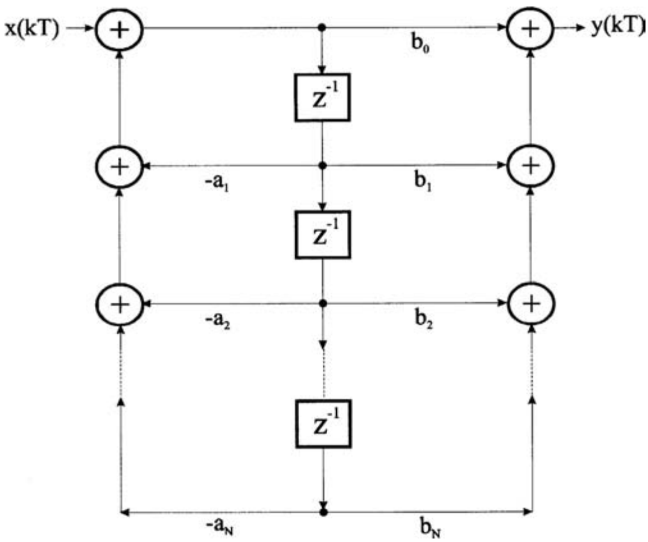

FIGURE 22.10 A direct-form II implementation of the difference equations minimizes the number of delay elements.

4. Implement the best filter using a general-purpose computer, a DSP, or a custom hardware chip.
5. Analyze the filter performance to determine whether the filter satisfies all the given criteria.

22.7 FIR Filter Design

In many digital signal-processing applications, FIR filters are generally preferred over their IIR counterparts, because they offer a number of advantages compared with their IIR equivalents. Some of the good properties of FIR filters are a direct consequence of their nonrecursive structure. First, FIR filters are inherently stable and free of limit cycle oscillations under finite-word length conditions. In addition, they exhibit a very low sensitivity to variations in the filter coefficients. Second, the design of FIR filters with exactly *linear phase* (constant group delay) vs. frequency behavior can be accomplished easily. This property is useful in many application areas, such as speech processing, phase delay equalization, image processing, etc.

Finally, there exists a number of efficient algorithms for designing optimum FIR filters with arbitrary specifications. The main disadvantage of FIR filters over IIR filters is that FIR filter designs generally require, particularly in applications requiring narrow transition bands, considerably more computation to implement.

An FIR filter of order N is described by a difference equation of the form

$$y(kT) = \sum_{\mu=0}^{N} b_\mu x(kT - \mu T) \tag{22.48}$$

and the corresponding transfer function is

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{\mu=0}^{N} b_\mu z^{-\mu} \tag{22.49}$$

The objective of FIR filter design is to determine $N \pm 1$ coefficients given by

$$h(0), h(1), \ldots, h(N) \tag{22.50}$$

so that the transfer function $H(e^{j\omega T})$ approximates a desired frequency characteristic. Note that because Equation 22.47 is also in the form of a convolution summation, the impulse response of an FIR filter is given by

$$h(kT) = \begin{cases} b_\mu, & k = 0, 1, \ldots, N \\ 0 & \text{otherwise} \end{cases} \tag{22.51}$$

Two equivalent structures for FIR filters are given in Figure 22.11.

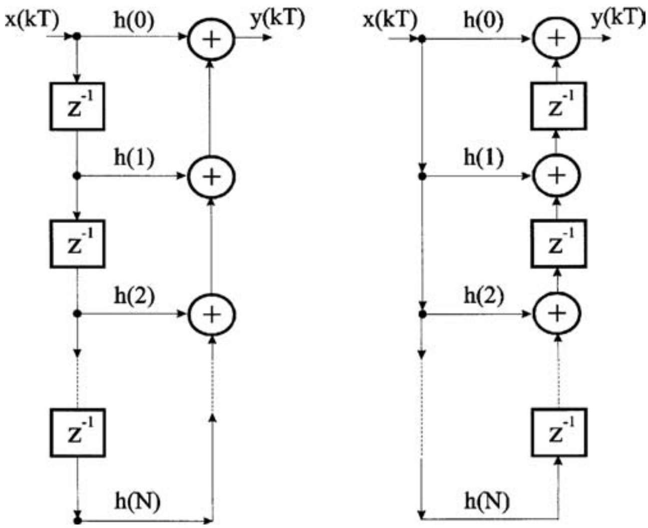

FIGURE 22.11 The sequence of the delays and summations can be varied to produce alternative direct-form implementations.

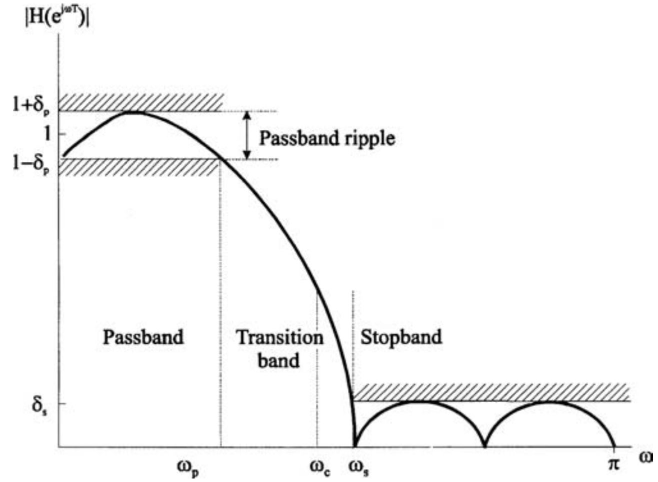

FIGURE 22.12 Tolerance limits must be defined for an FIR low-pass filter magnitude response.

The accuracy of an FIR approximation is described by the following parameters:

δ_P passband ripple
δ_s stopband attenuation
$\Delta\omega$ transition bandwidth

These quantities are depicted in Figure 22.12 for a prototype low-pass filter. δ_p and δ_s characterize the permissible errors in the passband and in the stopband, respectively. Usually, the passband ripple and stopband attenuation are given in decibels, in which case their values are related to the parameters δ_p and δ_s by

$$\text{Passband ripple (dB): } A_p = -20\log_{10}\left(1-\delta_p\right) \tag{22.52}$$

$$\text{Stopband ripple (dB): } A_s = 020\log_{10}\left(\delta_s\right) \tag{22.53}$$

Note that due to the symmetry and periodicity of the magnitude response of $|H(e^{j\omega T})|$, it is sufficient to give the filter specifications in the interval $0 \le \omega \le \pi$.

Windowed FIR Filters

Several design techniques can be employed to synthesize linear-phase FIR filters. The simplest implementation is based on *windowing*, which commonly begins by specifying the ideal frequency response and expanding it in a Fourier series and then truncating and smoothing the ideal impulse response by means of a window function. The truncation results in large ripples before and after the discontinuity of the ideal frequency response known as the Gibbs phenomena, which can be reduced by using a window function that tapers smoothly at both ends. Filters designed in this way possess equal passband ripple and stopband attenuation, i.e.,

$$\delta_p = \delta_s = \delta \tag{22.54}$$

To illustrate this method, let us define an ideal desired frequency response that can be expanded in a Fourier series

$$H_d\left(e^{j\omega T}\right) = \sum_{k=-\infty}^{\infty} h_d\left(kT\right)e^{-jk\omega T} \tag{22.55}$$

where $h_d(kT)$ is the corresponding impulse response sequence, which can be expressed in terms of $H_d(e^{j\omega T})$ as

$$h_d(kT) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d\left(e^{j\omega T}\right)e^{jk\omega T}d\omega \tag{22.56}$$

The impulse response of the desired filter is then found by weighting this ideal impulse response with a window $w(kT)$ such that

$$h(kT) = \begin{cases} w(kT)h_d(kT), & 0 \le k \le N \\ 0, & \text{otherwise} \end{cases} \tag{22.57}$$

Note that for $w(kT)$ in the above-given interval we obtain the rectangular window. Some commonly used windows are Bartlett (triangular), Hanning, Hamming, Blackmann, etc., the definitions of which can be found in Reference 15.

As an example of this design method, consider a low-pass filter with a cutoff frequency of ω_c and a desired frequency of the form

$$H_d\left(e^{j\omega T}\right) = \begin{cases} e^{-j\omega NT/2}, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi, \end{cases} \tag{22.58}$$

Using Equation 22.56 we obtain the corresponding ideal impulse response

$$h_d(kT) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{-j\omega TN/2}e^{jk\omega T}d\omega = \frac{\sin\left[\omega_c(kT - TN/2)\right]}{\pi(kT - TN/2)} \tag{22.59}$$

Choosing $N = 4$, $\omega_c = 0.6\pi$, and a Hamming window defined by

$$w(kT) = \begin{cases} 0.54 - 0.46\cos(2\pi kT/N), & 0 \le k \le N \\ 0, & \text{otherwise} \end{cases} \tag{22.60}$$

we obtain the following impulse response coefficients:

$$\begin{aligned} h(0) &= -0.00748 \\ h(1) &= 0.12044 \\ h(2) &= -0.54729 \\ h(3) &= 0.27614 \\ h(4) &= -0.03722 \end{aligned} \tag{22.61}$$

Optimum FIR Filters

As mentioned earlier, one of the principal advantages of FIR filters over their IIR counterparts is the availability of excellent design methods for optimizing arbitrary filter specifications. Generally, the design criterion for the optimum solution of an FIR filter design problem can be characterized as follows. The maximum error between the approximating response and the given desired response has to be minimized, i.e.,

$$E\left(e^{j\omega T}\right) = W_d\left(e^{j\omega T}\right)\left\|H_d\left(e^{j\omega T}\right)\right| - \left|H\left(e^{j\omega T}\right)\right\| \tag{22.62}$$

where $E(e^{j\omega T})$ is the weighted error function on a close range X of $[0,\pi]$ and $W_d(e^{j\omega T})$ a weighting function, which emphasizes the approximation error parameters in the design process. If the maximum absolute value of this function is less than or equal to ε on X, i.e.,

$$\varepsilon = \max_{\omega \in X}\left|E\left(e^{j\omega T}\right)\right| \tag{22.63}$$

the desired response is guaranteed to meet the given criteria. Thus, this optimization condition implies that the best approximation must have an equiripple error function. The most frequently used method for designing optimum magnitude FIR filters is the Parks–McClellan algorithm. This method essentially reduces the filter design problem into a problem in polynomial approximation in the Chebyshev approximation sense as discussed above. The maximum error between the approximation and the desired magnitude response is minimized. It offers more control over the approximation errors in different frequency bands than is possible with the window method. Using the Parks–McClellan algorithm to design FIR filters is computationally expensive. This method, however, produces optimum FIR filters by applying time-consuming iterative techniques. A FORTRAN program for the Parks–McClellan algorithm can be found in the IEEE publication Programs for DSP in Reference 12. As an example of an equiripple filter design using the Parks–McClellan algorithm, a sixth-order low-pass filter with a passband $0 \le \omega \le 0.6\pi$, a stopband $0.8\pi \le \omega \le \pi$, and equal weighting for each band was designed by means of this program.

The resulting impulse response coefficients are

$$\begin{aligned} h(0) &= h(6) = -0.00596 \\ h(1) &= h(5) = -0.18459 \\ h(2) &= h(4) = 0.25596 \\ h(3) &= 0.70055 \end{aligned} \tag{22.64}$$

Design of Narrowband FIR Filters

When using conventional techniques to design FIR filters with especially narrow bandwidths, the resulting filter lengths may be very high. FIR filters with long filter lengths often require lengthy design and implementation times, and are more susceptible to numerical inaccuracy. In some cases, conventional filter design techniques, such as the Parks–McClellan algorithm, may fail the design altogether. A very efficient algorithm called the interpolated finite impulse response (IFIR) filter design technique can be employed to design narrowband FIR filters. Using this technique produces narrowband filters that require far fewer coefficients than those filters designed by the direct application of the Parks–McClellan algorithm. For more information on IFIR filter design, see Reference 7.

22.8 IIR Filter Design

The main advantage of IIR filters over FIR filters is that IIR filters can generally approximate a filter design specification using a lower-order filter than that required by an FIR design to perform similar filtering operations. As a consequence, IIR filters execute much faster and do not require extra memory, because they execute in place. A disadvantage of IIR filters, however, is that they have a nonlinear phase response. The two most common techniques used for designing IIR filters will be discussed in this section. The first approach involves the transformation of an analog prototype filter. The second method is an optimization-based approach allowing the approximation of an arbitrary frequency response.

The transformation approach is quite popular because the approximation problem can be reduced to the design of classical analog filters, the theory of which is well established, and many closed-form design methods exist. Note that this is not true for FIR filters, for which the approximation problems are of an entirely different nature. The derivation of a transfer function for a desired filter specification requires the following three basic steps:

1. Given a set of specifications for a digital filter, the first step is to map the specifications into those for an equivalent analog filter.
2. The next step involves the derivation of a corresponding analog transfer function for the analog prototype.
3. The final step is to translate the transfer function of the analog prototype into a corresponding digital filter transfer function.

Once the corresponding analog transfer function for the analog prototype is derived, it must be transformed using a transformation that maps $H_a(s)$ into $H(z)$. The simplest and most appropriate choice for s is the well-known bilinear transform of the z-variable

$$s = \frac{2\left(1-z^{-1}\right)}{T_d\left(1+z^{-1}\right)} \Leftrightarrow z = \frac{1+\left(T_d/2\right)s}{1-\left(T_d/2\right)s} \tag{22.65}$$

which maps a stable analog filter in the s-plane into a stable digital filter in the z-plane. Substituting s with the right-hand side of Equation 22.63 in $H_a(s)$ results in

$$H(z) = H_a\left(\frac{2\left(1-z^{-1}\right)}{T_d\left(1+z^{-1}\right)}\right) \Rightarrow H\left(e^{j\omega T}\right)\Big|_{z=e^{j\omega T}} = H_a\left(\frac{2j}{T_d}\tan\left(\frac{\omega T}{2}\right)\right) \tag{22.66}$$

As it can be seen from Equation 22.66, the analog frequency domain (imaginary axis) maps onto the digital frequency domain (unit circle) nonlinearly. This phenomena is called frequency warping and must be compensated in a practical implementation. For low frequencies Ω and ω are approximately equal. We obtain the following relation between the analog frequency Ω and the digital frequency ω

$$\Omega = \frac{2}{T_d}\tan\left(\omega T/2\right) \tag{22.67}$$

$$\omega = \frac{2}{T}\arctan\left(\Omega T_d/2\right) \tag{22.68}$$

The overall bilinear transformation procedure is as follows:

1. Convert the critical digital frequencies (e.g., ω_p and ω_s for low-pass filters) to the corresponding analog frequencies in the s-domain using the relationship given by Equation 22.67.
2. Derive the appropriate continuous prototype transfer function $H_a(s)$ that has the properties of the digital filter at the critical frequencies.
3. Apply the bilinear transform to $H_a(s)$ to obtain $H(z)$ which is the required digital filter transfer function.

To illustrate the three-step IIR design procedure using the bilinear transform, consider the design of a second-order Butterworth low-pass filter with a cutoff frequency of $\omega_c = 0.3\pi$. The sampling rate of the digital filter is to be $f_s = 10$ Hz, giving $T = 0.1$ s. First, we map the cutoff frequency to the analog frequency

$$\Omega_c = \frac{2}{0.1}\tan\left(0.15\pi\right) = 10.19 \text{ rad/s} \tag{22.69}$$

The poles of the analog Butterworth filter transfer function $H_a(s)$ are found using Equation 22.11. As explained earlier, these poles lie equally spaced in the s-plane on a circle of radius Ω_c.

$$H_a(s) = \frac{1}{s^2 + \sqrt{2}\Omega_c s + \Omega_c^2} \tag{22.70}$$

Application of the bilinear transformation

$$s = \frac{2\left(1 - z^{-1}\right)}{0.1\left(1 + z^{-1}\right)} \tag{22.71}$$

gives the digital transfer function

$$H(z) = \frac{0.00002 + 0.00004z^{-1} + 0.00002z^{-2}}{1 - 1.98754z^{-1} + 0.98762z^{-2}} \tag{22.72}$$

The above computations were carried out using Reference 9, which greatly automates the design procedure.

Design of Arbitrary IIR Filters

The IIR filter design approach discussed in the previous section is primarily suitable for frequencyselective filters based on closed-form formulas. In general, however, if a design other than standard lowpass, highpass, bandpass, and stopband is required, or if the frequency responses of arbitrary specifications are to be matched, in such cases it is often necessary to employ algorithmic methods implemented on computers. In fact, for nonstandard response characteristics, algorithmic procedures may be the only possible design approach. Depending on the error criterion used, the algorithmic approach attempts to minimize the approximation error between the desired frequency response $H_d(e^{j\omega T})$ and $H(e^{j\omega T})$ or between the time-domain response $h_d(kT)$ and $h(kT)$. Computer software is available for conveniently implementing IIR filters approximating arbitrary frequency response functions [8,9].

Cascade-Form IIR Filter Structures

Recall that theoretically there exist an infinite number of structures to implement a digital filter. Filters realized using the structure defined by Equation 22.44 directly are referred to as direct-form IIR filters. The direct-form structure, however, is not employed in practice except when the filter order $N \leq 2$, because they are known to be sensitive to errors introduced by coefficient quantization and by finite-arithmetic conditions. Additionally, they produce large round-off noise, particularly for poles closed to the unit circle.

Two less-sensitive structures can be obtained by partial fraction expansion or by factoring the right-hand side of Equation 22.46 in terms of real rational functions of order 1 and 2. The first method leads to *parallel connections* and the second one to *cascade connections* of corresponding lower-order sections, which are used as building blocks to realize higher-order transfer functions. In practice, the cascade form is by far the preferred structure, since it gives the freedom to choose the pairing of numerators and denominators and the ordering of the resulting structure. Figure 22.13 shows a cascade-form implementation, whose overall transfer function is given by

$$H(z) = \prod_{k=1}^{M} H_k(z) \tag{22.73}$$

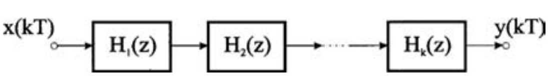

FIGURE 22.13 An IIR filter can be implemented by a cascade of individual transfer functions.

where the transfer function of the kth building block is

$$H_k(z) = \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-2} + a_{2k}z^{-2}} \tag{22.74}$$

Note this form is achieved by factoring Equation 22.45 into second-order sections.

There are, of course, many other realization possibilities for IIR filters, such as state-space structures [9], lattice structures [10], and wave structures. The last is introduced in the next section.

22.9 Wave Digital Filters

It was shown earlier that for recursive digital filters the approximation problem can be reduced to classical design problems by making use of the bilinear transform. For wave digital filters (WDFs) this is carried one step farther in that the structures are obtained directly from classical circuits. Thus, to every WDF there corresponds an *LCR* reference filter from which it is derived. This relationship accounts for their excellent properties concerning coefficient sensitivity, dynamic range, and all aspects of stability under finite-arithmetic conditions. The synthesis of WDFs is based on the wave network characterization; therefore, the resulting structures are referred to as wave digital filters. To illustrate the basic idea behind the theory of WDFs, consider an inductor L, which is electrically described by $V(s) = sLI(s)$. In the next step we define wave variables $A_1(s)$ and $B_1(s)$ as

$$\begin{aligned} A_1(s) &= V(s) + RI(s) \\ B_1(s) &= V(s) - RI(s) \end{aligned} \tag{22.75}$$

where R is called the port resistance. Substituting $V(s) = sLI(s)$ in the above relation and replacing s in $A_1(s)$ and $B_1(s)$ with the bilinear transform given by Equation 22.65, we obtain

$$B(z) = \frac{\left(1 - z^{-1}\right)L - \left(1 + z^{-1}\right)R}{\left(1 - z^{-1}\right)L + \left(1 + z^{-1}\right)R} A(z) \tag{22.76}$$

Letting $R = L$, the above relation reduces to

$$B(z) = -z^{-1} A(z) \tag{22.77}$$

Thus an inductor translates into a unit delay in cascade with an inverter in the digital domain. Similarly, it is easily verified that a capacitance can be simulated by a unit delay and a resistor by a digital sink. Figure 22.14 shows the digital realizations of impedances and other useful one-port circuit elements.

To establish an equivalence with classical circuits fully, the interconnections are also simulated by so-called wave adaptors. The most important of these interconnections are series and parallel connections, which are simulated by series and parallel adaptors, respectively. For most filters of interest, only two- and three-port adaptors are employed. For a complete design example consider Figure 22.15.

For a given *LC* filter, one can readily derive a corresponding WDF by using the following procedure. First, the various interconnections in the *LC* filter are identified as shown in Figure 22.15. In the next step the electrical elements in the *LC* filter are replaced by its digital realization using Figure 22.15. Finally, the interconnections are substituted using adaptors. Further discussions and numerical examples dealing with WDFs can be found in References 3, 13, and 14.

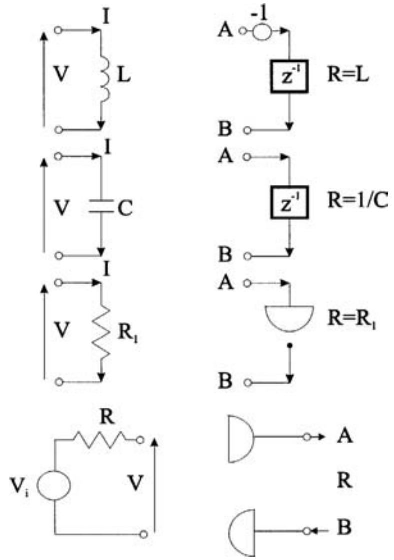

FIGURE 22.14 Digital filter implementations use functional equivalents to one-port linear filter elements.

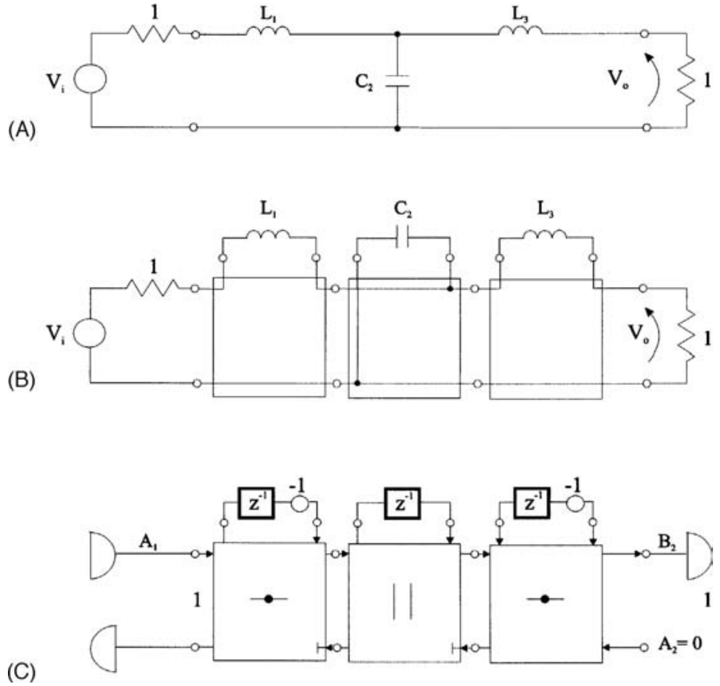

FIGURE 22.15 Digital wave filters establish equivalence with classical filter circuits by use of wave adapter substitutions: (A) *LC* reference low-pass; (B) identification of wire interconnections; (C) corresponding wave digital filter.

22.10 Antialiasing and Smoothing Filters

In this section two practical application areas of filters in the analog conditioning stage of a data acquisition system are discussed. A block diagram of a typical data acquisition system is shown in Figure 22.16, consisting of an **antialiasing filter** before the analog-to-digital converter (ADC) and a **smoothing filter** after the digital-to-analog converter (DAC).

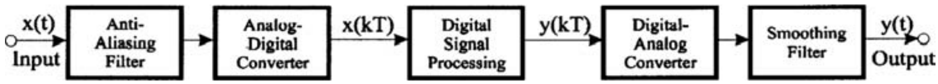

FIGURE 22.16 A data acquisition system with continuous time inputs and outputs uses antialias prefiltering, an A/D converter, digital signal processing, a D/A converter, and an output smoothing filter.

For a complete discrete reconstruction of a time-continuous, band-limited input signal having the spectrum $0 \le f \le f_{max}$, the sampling frequency must be, according to the well-known Shannon's sampling theorem, at least twice the highest frequency in the time signal. In our case, in order to be able to represent frequencies up to f_{max}, the sampling frequency $f_s = 1/T > 2f_{max}$. The necessary band limiting to $f \le f_{max}$ of the input time-continuous signal is performed by a low-pass filter, which suppresses higher spectral components greater than f_{max}. Violation of this theorem results in alias frequencies. As a result, frequency components above $f_x/2$, the so-called Nyquist frequency, appear as frequency components below $f_x/2$. Aliasing is commonly addressed by using antialiasing filters to attenuate the frequency components at and above the Nyquist frequency to a level below the dynamic range of an ADC before the signal is digitized. Ideally, a low-pass filter with a response defined by

$$H(j\Omega) = \begin{cases} 1, & |\Omega| \le \pi/T \\ 0, & |\Omega| \le \pi/T \end{cases} \tag{22.78}$$

is desired to accomplish this task. In practice, a variety of techniques based on the principles of continuous-time analog low-pass filter design can be employed to approximate this "brick-wall" type of characteristic. Antialiasing filters typically exhibit attenuation slopes in the range from 45 to 120 dB/octave and stopband rejection from 75 to 100 dB. Among the types of filters more commonly used for antialias purposes are the Cauer elliptic, Bessel, and Butterworth. The optimum type of filter depends on which kinds of imperfections, e.g., gain error, phase nonlinearity, passband and stopband ripple, etc., are most likely to be tolerated in a particular application. For example, Butterworth filters exhibit very flat frequency response in the passband, while Chebyshev filters provide steeper attenuation at the expense of some passband ripple. The Bessel filter provides a linear phase response over the entire passband but less attenuation in the stopband. The Cauer elliptic filter, with its extremely sharp roll-off, is especially useful as an antialiasing filter for multichannel digitizing data acquisition systems. However, the large-phase nonlinearity makes it more appropriate for applications involving analysis of the frequency content of signals as opposed to phase content or waveform shape.

Many considerations discussed above also apply to smoothing filters. Due to the sampling process, the frequency response after the digital-to-analog conversion becomes periodic with a period equal to the sampling frequency. The quantitization steps that are created in the DAC reconstruction of the output waveform and are harmonically related to the sampling frequency must be suppressed through a low-pass filter having the frequency response of Equation 22.78, also referred to as a smoothing or reconstruction filter. While an antialiasing filter on the input avoids unwanted errors that would result from undersampling the input, a smoothing filter at the output reconstructs a continuous-time output from the discrete-time signal applied to its input.

Consideration must be given to how much antialiasing protection is needed for a given application. It is generally desirable to reduce all aliasable frequency components (at frequencies greater than half of the sampling frequency) to less than the LSB of the ADC being used. If it is possible that the aliasable input can have an amplitude as large as the full input signal range of the ADC, then it is necessary to attenuate it by the full 2^N range of the converter. Since each bit of an ADC represents a factor of 2 from the ones adjacent to it, and 20 log(2) = 6 dB, the minimum attenuation required to reduce a full-scale input to less than a LSB is

$$\alpha < -20N(6\ \text{dB}) \tag{22.79}$$

where N is the number of bits of the ADC.

The amount of attenuation required can be reduced considerably if there is knowledge of the input frequency spectrum. For example, some sensors, for reasons of their electrical or mechanical frequency response, might not be able to produce a full-scale signal at or above the Nyquist frequency of the system and therefore "full-scale" protection is not required. In many applications, even for 16-bit converters that, in the worst case, would require 96 dB of antialias protection, 50 to 60 dB is adequate.

Additional considerations in antialias protection of the system are the noise and distortion that are introduced by the filter that is supposed to be eliminating aliasable inputs. It is possible to have a perfectly clean input signal which, when it is passed through a prefilter, gains noise and harmonic distortion components in the frequency range and of sufficient amplitude to be within a few LSBs of the ADC. The ADC cannot distinguish between an actual signal that is present in the input data and a noise or distortion component that is generated by the prefilter. It is necessary that both noise and distortion components in the output of the antialias filter must also be kept within an LBS of the ADC to ensure system accuracy.

22.11 Switched-Capacitor Filters

Switched-capacitor (SC) filters, also generally referred to as analog sampled data filters, provide an alternative to conventional active-*RC* filters and are commonly used in the implementation of adjustable antialiasing filters. SC filters comprise switches, capacitors, and op amps. Essentially, an SC replaces the resistor in the more traditional analog filter designs. Because the impedance of the SC is a function of the switching frequency, one can vary the cutoff frequency of the SC filter by varying the frequency of the clock signal controlling the switching. The main advantage of SC filters is that they can be implemented in digital circuit process technology, since the equivalent of large resistors can be simulated by capacitors having small capacitance values.

When using SC filters, one must also be aware that they are in themselves a sampling device that requires antialias protection on the input and filtering on their outputs to remove clock feedthrough. However, since clock frequencies are typically 50 to 100 times f_c of the filter, a simple first or second *RC* filter on their inputs and outputs will reduce aliases and noise sufficiently to permit their use with 12- to 14-bit ADCs. One need also to consider that they typically have dc offset errors that are large, vary with time, temperature, and programming or clock frequency. Interested readers may refer to References 5 and 14.

22.12 Adaptive Filters

Adaptive filtering is employed when it is necessary to realize or simulate a system whose properties vary with time. As the input characteristics of the filter change with time, the filter coefficients are varied with time as a function of the filter input. Some typical applications of adaptive filtering include spectral estimation of speech, adaptive equalization, echo cancellation, and adaptive control, to name just a few. Depending on the application, the variations in the coefficients are carried out according to an optimization criterion and the adaptation is performed at a rate up to the sampling rate of the system. The self-adjustment capability of adaptive filter algorithms is very valuable when the application environment cannot be precisely described. Some of the most widely used adaptive algorithms are LMS (least-mean square), RLS (recursive least-squares), and frequency domain, also known as block algorithm. The fundamental concept of an adaptive filter is depicted in Figure 22.17.

An adaptive filter is characterized by the filter input $x(kT)$ and the desired response $d(kT)$. The error sequence $\varepsilon(kT)$ formed by

$$\varepsilon(kT) = \sum_{\mu=0}^{N-1} w_\mu(kT)\, x(kT - \mu T) \tag{22.80}$$

and $x(kT),\ldots,x(kT - T(N-1))$ serve as inputs to an adaptive algorithm that recursively determines the coefficients $w_0(kT + T),\ldots,w_{N-1}(kT + T)$. A number of adaptive algorithms and structures can be found

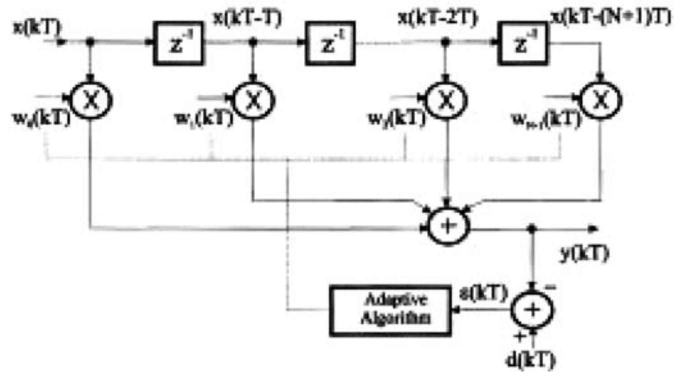

FIGURE 22.17 An adaptive filter uses an adaptive algorithm to change the performance of a digital filter in response to defined conditions.

in the literature that satisfy different optimization criteria in different application areas. For more detailed developments refer to References 1, 15, and 16.

Defining Terms

Antialiasing filter: Antialiasing filters remove any frequency elements above the Nyquist frequency. They are employed before the sampling operation is conducted to prevent aliasing in the sampled version of the continuous-time signal.

Bandpass filter: A filter whose passband extends from a lower cutoff frequency to an upper cutoff frequency. All frequencies outside this range are stopped.

Equiripple: Characteristic of a frequency response function whose magnitude exhibits equal maxima and minima in the passband.

Finite impulse response (FIR) filter: A filter whose response to a unit impulse function is of finite length, i.e., identically zero outside a finite interval.

High-pass filter: A filter that passes all frequencies above its cutoff frequency and stops all frequencies below it.

Ideal filter: An ideal filter passes all frequencies within its passband with no attenuation and rejects all frequencies in its stopband with infinite attenuation. There are five basic types of ideal filters: low-pass, high-pass, bandpass, stopband, and all-pass.

Infinite impulse response (IIR) filter: A filter whose response to a unit impulse function is of infinite length, i.e., nonzero for infinite number of samples.

Low-pass filter: A filter that attenuates the power of any signals with frequencies above its defined cutoff frequency.

Passband: The range of frequencies of a filter up to the cutoff frequency.

Stopband: The range of frequencies of a filter above the cutoff frequency.

Transition region: The range of frequencies of a filter between a passband and a stopband.

References

1. S. Mitra and J. Kaiser, *Handbook for Digital Signal Processing*, New York: John Wiley & Sons, 1993.
2. E. Christian and E. Eisenmann, *Filter Design Tables and Graphs*, New York: John Wiley & Sons, 1966.
3. A. Antoniou, *Digital Filter: Analysis and Design*, New York: McGraw-Hill, 1979.
4. R. Saal, *Handbuch zum Filterentwurf [Handbook of filter design]*, Frankfurt: Allgemeine Elektricitäts-Gesellschaft AEG-Telefunken, 1979.
5. G. Temes and S. Mitra, *Modern Filter Theory and Design*, New York: John Wiley & Sons, 1973.
6. E. Cunningham, *Digital Filtering*, New York: John Wiley & Sons, 1995.

7. P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ: Prentice-Hall, 1993.
8. M. Cerna and R. Jamal, The design of digital filters using graphical programming techniques, *Proc. MessComp*, 232–238, 1995.
9. *LabVIEW Digital Filter Design Toolkit*, Austin, TX: National Instruments, 1996.
10. R.A. Roberts and C.T. Multis, *Digital Signal Processing*, Reading, MA: Addison-Wesley, 1987.
11. A.H. Gray, Jr. and J.D. Markel, Digital lattice and ladder filter synthesis, *IEEE Trans. Acoust. Speech Signal Process.*, ASSP-23: 268–277, 1975.
12. DSP Committee, IEEE ASSP, Eds., *Programs for Digital Signal Processing*, New York: IEEE Press, 1979.
13. F. Taylor, *Digital Filter Design Handbook*, New York: Marcel Dekker, 1983.
14. A. Fettweis, *Wave digital filters: theory and practice*, *Proc. IEEE*, 74, 270–327, 1986.
15. W.K. Chen, *The Circuits and Filters Handbook*, Boca Raton, FL: CRC Press, 1995.
16. M.L. Honig and D. Messerschmitt, *Adaptive Filters — Structures, Algorithms, and Applications*, Boston: Kluwer Academic Publishers, 1984.
17. M. Bellanger, *Digital Processing of Signals — Theory and Practice*, New York: John Wiley & Sons, 1988.
18. R.C. Dorf, Ed., *The Electrical Engineering Handbook*, Boca Raton, FL: CRC Press, 1993.
19. A. Zverev, *Handbook of Filter Synthesis*, New York: John Wiley & Sons, 1967.
20. C. Lindquist, *Active Network Design*, Long Beach, CA: Steward & Sons, 1977.