J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*

IIR FILTERS

IIR or infinite-length impulse response digital filters have impulse responses of infinite length, which extend over all time. *FIR* or finite-length impulse response filters do not. IIR filters are also called recursive filters, or less frequently, ladder, lattice, wave digital, pole-zero, autoregression moving average (*ARMA*), and autoregression integrated moving average (*ARIMA*) filters. FIR filters are also called nonrecursive filters, as well as moving average, delay line and tapped delay line, feedforward, all-zero, and transversal filters (1). From all these names, we choose to classify digital filters as IIR or FIR.

Terms such as IIR are usually applied to digital filters rather than analog filters. Digital filters operate in discrete time, and analog filters operate in continuous time (2). Digital filters are usually analyzed using the z-transform (denoted $Z[\,]$), whereas analog filters are analyzed using the Laplace transform (denoted $L[\,]$). There are very close analogies between the two filter types, and these are often exploited in design.

From a design standpoint, IIR and FIR filters compete with each other. IIR filters have many desirable properties, which are listed in Table 1. Their design is easy if frequency transformations and other methods are used. Well-known classical analog filters like Butterworth can be used. Low-order IIR filters can have gains with sharp cut-off and high selectivity. IIR filters can be implemented using little storage, short delays, and a small number of arithmetic computations.

IIR have several undesirable properties. They can be unstable. They are stable only if their poles lie inside the unit circle (assuming causality). They can never have linear phase unlike FIR filters. They are generally more difficult to design unless frequency transforms are used. Nevertheless, their desirable characteristics generally far outweigh these undesirable properties, so IIR filters are the most widely used in industry.

IIR filters can be implemented using a variety of techniques. These are summarized in Table 2. The most popular methods include numerical integration or digital transforms, several invariant time domain response designs, and matched z-transform design. IIR filters can also be designed using FIR filter design techniques (1).

Basic Description

A linear digital filter is characterized by its impulse response $h(n)$. The impulse response relates the filter input $x(n)$ and output $y(n)$ as

$$y(n) = h(n) * x(n) \tag{1}$$

when the filter is time-invariant and $*$ represents the convolution operation. Taking the z-transform of both sides gives

$$Y(z) = H(z)X(z) \tag{2}$$

Table 1. Advantages of IIR and FIR Digital Filters

FIR Filters	IIR Filters
1. Always stable.	Can be unstable; stable only if poles inside unit circle.
2. Can be exactly linear phase.	Linear phase only with phase equalization.
3. FFT implementation possible permitting fast convolution.	Unnecessary.
4. Design methods easy and well-defined.	Design methods more difficult but well-defined. They are easy if frequency transforms are used.
5. No exact closed-form equations so most designs require iteration.	Some designs are exact with closed-form equations.
6. Cannot usually use these tabulated filters of Chap. 4.	Easy to incorporate classical, optimum, and adaptive filter tabulated results of Chap. 4 into design.
7. Low selectivity unless high order (orders 4–10 times as much).	Low-order filters give sharp cutoff or high selectivity.
8. More storage and greater number of arithmetic computations. Long delays may be necessary.	Small storage, short delays, and small number of arithmetic computations.

C. S. Lindquist, *Adaptive & Digital Signal Processing with Digital Filtering Applications,* International Series in Signal Processing and Filtering, vol. 2, p. 507. Miami, FL: Steward & Sons, 1989.

If the filter is time-varying, then $y(n) = h(n,m) * x(m)$ and $Y(z_2) = H(z_1,z_2)X(z_2)$. In this case, $h(n,m)$ and $H(z_1,z_2)$ are two-dimensional functions. Such digital filters will not be considered, but their analysis and design is a direct extension of what will be presented here (3). The general time domain form [Eq. (1)] and frequency domain form [Eq. (2)] of the digital filter are shown in Fig. 1.

IIR filters have transfer functions that can be expressed in the product form

$$H(z) = \frac{\prod_{k=-\infty}^{\infty}(z - z_k)}{\prod_{k=-\infty}^{\infty}(z - p_k)} = \sum_{k=-\infty}^{\infty} h_k z^{-k} \qquad (3)$$

The values of z where $H(z) = 0$ are called the zeros z_k of the filter. The values of z where $1/H(z) = 0$ are called the poles p_k of the filter. Any common terms in numerator and denominator are usually canceled out. The IIR filter gain can also be expressed in the polynomial form

$$H(z) = \frac{\sum_{k=-\infty}^{\infty} a_k z^{-k}}{\sum_{k=-\infty}^{\infty} b_k z^{-k}} = \sum_{k=-\infty}^{\infty} h_k z^{-k}, \qquad b_0 = 1 \qquad (4)$$

Table 2. Summary of IIR and FIR Digital Filter Design Techniques

Design Method	FIR Filters	IIR Filters
	Frequency Domain	
1. Fourier series	Optimum ISE gain response; ringing oscillation due to Gibb's effect.	Same but suboptimal.
2. Weighted Fourier series	Suboptimal ISE but improved gain response; reduced Gibb's effect.	Same but suboptimal.
3. Frequency sampling	Exact gain response at $2N$ 1 1 frequencies; ringing and oscillation present.	Same as nonrecursive.
4. Frequency transforms		
a. General s-to-z	Same as recursive.	Most popular approach; easy to apply, warps frequency response.
b. Bilinear	Not applicable.	Most widely used; alias free, warps frequency response.
c. Matched-z	Same as recursive.	Converts analog roots to digital roots directly.
d. Numerical interpolation/extrapolation	Easy to apply; alternative s-to-z transform.	Same as nonrecursive.
e. Numerical differentiation	Easy to apply; alternative s-to-z transform.	Same as nonrecursive.
f. Numerical integration	Not applicable.	Easy to apply; alternative s-to-z transform. Almost preserves time-domain response.
	Time Domain	
1. Invariant response	Same as recursive.	Preserves time-domain response.
a. Impulse invariant (z-transform)	Same as recursive.	Preserves impulse response, aliased gain response.
b. Step-invariant	Same as recursive.	Preserves step response, aliased and distorted gain response.
c. Modified-impulse invariant	Same as recursive.	Approximately preserves impulse and gain responses.
2. Matched z-transform	Same as recursive.	Does not preserve impulse and gain responses.

C. S. Lindquist, *Adaptive & Digital Signal Processing with Digital Filtering Applications,* International Series in Signal Processing and Filtering, vol. 2, p. 508. Miami, FL: Steward & Sons, 1989.

As mentioned before, based on the form of Eqs. (3) and (4), such digital filters are classified in two broad groups as IIR when the denominator $D(z) \neq 1$ and FIR when $D(z) = 1$.

Rearranging Eq. (4) as $Y(z) = H(z)X(z)$, taking the inverse z-transforms of both sides, and solving for $x(n)$ gives

$$y(n) = \sum_{k=-\infty}^{\infty} a_k x(n-k) - \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} b_k y(n-k) \qquad (5)$$

This is the time-domain equation used to generate the filter output $y(n)$ from the input $x(n)$. Because feedback of the output (or so-called recursion) depends upon b_k, Eq. (5) makes it clear that the digital filter is recursive when $b_k \neq 0$ and nonrecursive when $b_k = 0$. The characteristics of the digital filter are controlled by its gain $H(z)$ and impulse response $h(n)$.

We first describe some of the overall filter properties based on $H(z)$ and $h(n)$. A digital filter is causal or nonanticipatory if its impulse response is zero for negative time,

$$h(n) = 0, \qquad n < 0 \qquad (6)$$

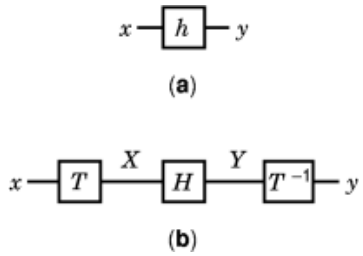

Fig. 1. Block diagram of digital filter in (a) time domain form and (b) frequency domain form.

Most real-time filters are designed to be causal. Causality can also be tested using $H(z)$ and the Paley-Wiener criterion (1). Noncausal filters can often be made causal by delaying its impulse response and truncating its negative time portion. This changes its gain $H(z)$ very little if only a small portion is truncated.

Another important property is stability. A causal digital filter is stable if its impulse response decays to zero as time approaches infinity,

$$h(n) \to 0, \qquad n \to \infty \tag{7}$$

Causality is easily tested by inspecting the poles of $H(z)$. A causal digital filter is stable when all the poles of $H(z)$ lie inside the unit circle of the z-plane and any poles on the unit circle are simple or first-order. Most digital filters are designed to be both causal and stable (4).

Digital filters are designed using frequency domain and time domain techniques. In the frequency domain approach, an analog filter transfer function $H(s)$ is directly converted into a digital filter transfer function $H(z)$ using some form of

$$H(z) = H(s)\Big|_{s=g(z)} = \frac{\mathcal{L}[y(t)]}{\mathcal{L}[x(t)]}\Big|_{s=g(z)} \tag{8}$$

$g(z)$ is the desired digital transform. The resulting $H(z)$ generally has frequency-domain and time-domain responses that differ considerably from the analog filter responses from which they were derived.

A time-domain approach preserves a particular temporal response of the analog filter used to form the digital filter as

$$H(z) = \frac{\mathcal{Z}[y(n)]}{\mathcal{Z}[x(n)]} \tag{9}$$

where $x(n)$ and $y(n)$ are the sampled input and output, respectively, of the analog filter. Equivalently, $H(z)$ is the z-transform of $h(n)$. When the sampling frequency $f_s = 1/T$ Hz is sufficiently high compared with the stopband frequency of the analog filter, the frequency response characteristics are preserved. Otherwise, there is noticeable aliasing distortion.

We should note a convention that is standard. The samples are taken at time $t = nT$. The input $x(t)$ is sampled to produce $x(nT)$. To simplify notation, this is written in shorthand as $x(n)$. This is true for all analog filter variables when they are converted into digital filter variables.

Frequency-Domain Algorithms

Digital transforms directly convert an analog transfer function $H(s)$ into a digital transfer function $H(z)$ using

$$H(z) = H(s)\Big|_{s=g(z)} \tag{10}$$

$g(z)$ is the desired digital or discrete transform. The resulting $H(z)$ generally has frequency-domain and time-domain responses that differ considerably from those of the analog filter from which they were derived. Only s-to-z mappings that map the imaginary axis of the s-plane onto the unit circle of the z-plane preserve certain frequency-domain characteristics like the passband and stopband magnitude response ripples and phase variations. Delay and time-domain responses are distorted.

Numerical Integration Transform Design. One especially simple way to obtain digital transforms is to approximate continuous integrators by discrete integrators (5). Equating the continuous time transfer function $H(s) = 1/s$ to the discrete time equivalent $H(z) = 1/g(z)$ results in $s = g(z)$, which is the digital transform. The digital transfer function $H_{dmn}(z)$ of the discrete integrator is chosen to equal

$$H_{dmn}(z) = \frac{z^{-d}\sum_{k=0}^{m-d} a_k z^{-k}}{\sum_{k=0}^{n} b_k z^{-k}}, \quad b_0 = 1 \tag{11}$$

d is the number of zeros at the origin, m is the number of finite zeros including those at the origin, and n is the number of finite poles in the z^{-1} plane. The a_k and b_k are selected so that H_{dmn} well approximates the transfer function of an analog integrator $H(s) = 1/s$ when $z = e^{-sT} = \Sigma^{\infty}{}_{k=0}(sT)^k/k!$. This method generates an infinite number of digital transforms (1).

Some of them and others are listed in Table 3. We have selected the backward and forward Euler, bilinear, modified bilinear, lossless, and optimum transforms. These transforms involve a constant C, which is adjusted so that the analog frequency $\omega = 1$ rad/s maps into any desired digital filter frequency B rad/s, and normalized frequency BT radians or $360^\circ f/f_s$ degrees. The bilinear transform, denoted as H_{011}, is perhaps the most popular and widely used where

$$H(z) = H(s)\Big|_{s=\frac{1}{C}\frac{1-z^{-1}}{1+z^{-1}}} \tag{12}$$

and $C = \tan(BT/2)$. This transformation maps the imaginary axis of the s-plane describing the analog filter unto the unit circle of the z-plane describing the digital filter. Therefore, except for frequency warping, the magnitude and phase characteristics of the analog filter are preserved exactly. The delay, which is the derivative of the phase, changes as does the step response. Of the many transforms available, the bilinear transform is the most widely used because it has the desirable mapping property just mentioned.

An important comment should be made. Causal and stable analog filters are not always mapped into causal and stable digital filters using these transforms. The backward Euler, bilinear, and optimum transforms do produce causal and stable filters, but the forward Euler and lossless transforms do not (1). When using general transforms, the poles of $H(z)$ should always be verified to be inside or on the unit circle (assuming causality).

Other Filter Types. The transforms of Table 3 are usually used to map analog filter transfer functions into digital filter transfer functions. If the analog filter is lowpass, these transforms produce a digital highpass filter. If the analog filter is bandpass, the transforms produce a digital bandpass filter.

Table 3. Some Analog Low-pass-to-Digital Low-pass Discrete Transforms. Reciprocal s_n Produce Analog Low-pass-to-Digital High-pass Discrete Transforms

H_{dmn}	Discrete Transform	Transformation $s_n = g(z)$
H_{001}	Backward Euler	$s_n = \frac{1}{C}\left(1 - z^{-1}\right)$ $C = \sin(BT)$
H_{111}	Forward Euler	$s_n = \frac{1}{C}\left(\frac{1 - z^{-1}}{z^{-1}}\right)$ $C = \sin(BT)$
H_{011}	Bilinear	$s_n = \frac{1}{C}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)$ $C = \tan(BT/2)$
H_{MBDI}	Modified bilinear	$s_n = \frac{1}{C}\left(\frac{1 - z^{-1}}{z^{-1}(1 + z^{-1})}\right)$ $C = \tan(BT/2)\cos(BT)$
H_{LDI}	Lossless	$s_n = \frac{1}{C}\left(\frac{1 - z^{-1}}{z^{-0.5}}\right)$ $C = 2\sin(BT/2)$
H_{ODI}	Optimum	$s_n = \frac{1}{C}\left(\frac{1 - z^{-0.5}}{z^{-0.25}}\right)$ $C = 2\sin(BT/4)$

C. S. Lindquist, *Adaptive & Digital Signal Processing with Digital Filtering Applications,* International Series in Signal Processing and Filtering, vol. 2, p. 401. Miami, FL: Steward & Sons, 1989.

However, they can also be used to convert analog lowpass filters to the other types directly. If the *reciprocals of s* are used in Table 3, digital highpass filters are produced. If the transforms in Table 4 are used, digital bandpass filters are produced directly. If the *reciprocals of s* are used in Table 4, then digital bandstop filters are produced directly. Thus analog lowpass filters can be converted directly into any of these other filter types by using Tables 3 and 4.

These other filters can be obtained by a simpler and more direct scheme. After a lowpass digital filter $H(z)$ is obtained, it can be converted into a highpass filter using $H(-z) = H(ze^{j\pi})$. It can be converted to a single-sideband bandpass filter using $H(ze^{j\omega_0 T})$. Therefore, by simply rotating the pole-zero pattern of the lowpass $H(z)$ by θ degrees in the z-plane, other filter types can be obtained. These filters have the same arithmetic shape but have different geometric shapes, the proper bandwidths, etc. The proper shape is always maintained by using the transforms in Tables 3 and 4.

Table 4. Some Analog Low-pass-to-Digital Bandpass Discrete Transforms. Reciprocal s_n Produce Analog Low-pass-to-Digital Bandstop Discrete Transforms

H_{dmn}	Discrete Transform	Transformation $s_{LP} = g(z)$
H_{001}	Backward Euler	$s_{LP} = \frac{1}{C}\left[\frac{D - 2z^{-1} + z^{-2}}{1 - z^{-1}}\right]$ $C = \sin(BT)$ $D = 1 + \sin^2(\omega_0 T)$
H_{111}	Forward Euler	$s_{LP} = \frac{1}{C}\left[\frac{1 - 2z^{-1} + Dz^{-2}}{z^{-1}(1 - z^{-1})}\right]$ $C = \sin(BT)$ $D = 1 + \sin^2(\omega_0 T)$
H_{011}	Bilinear	$s_{LP} = \frac{1}{C}\left[\frac{1 - Dz^{-1} + z^{-2}}{1 - z^{-2}}\right]$ $C = \tan(BT/2)$ $D = 2\cos(\omega_0 T)$
H_{MBDI}	Modified bilinear	$s_{LP} = \frac{1}{C}\left[\frac{1 - 2z^{-1} + Dz^{-2} + Ez^{-3} + Fz^{-4}}{z^{-1}(1 + z^{-2})}\right]$ $C = \tan(BT/2)\cos(BT)$ $D = 1 + F = 1 + \tan^2(\omega_0 T/2)\cos^2(\omega_0 T)$ $E = 2F = 2\tan^2(\omega_0 T/2)\cos^2(\omega_0 T)$ $F = \tan^2(\omega_0 T/2)\cos^2(\omega_0 T)$
H_{LDI}	Lossless	$s_{LP} = \frac{1}{C}\left[\frac{1 - Dz^{-1} + Z^{-2}}{z^{-0.5}(1 - z^{-1})}\right]$ $C = 2\sin(BT/2)$ $D = 2\cos(\omega_0 T)$
H_{ODI}	Optimum	$s_{LP} = \frac{1}{C}\left[\frac{1 - Dz^{-0.5} + z^{-1}}{z^{-0.25}(1 - z^{-0.5})}\right]$ $C = 2\sin(BT/4)$ $D = 2\cos(\omega_0 T/2)$

C. S. Lindquist, *Adaptive & Digital Signal Processing with Digital Filtering Applications,* International Series in Signal Processing and Filtering, vol. 2, p. 461. Miami, FL: Steward & Sons, 1989.

Time-Domain Algorithms

Time-domain algorithms preserve a particular temporal response of the analog filter used to form the digital filter. Mathematically, the invariant time-domain transform uses

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\mathcal{Z}[y(n)]}{\mathcal{Z}[x(n)]} = \frac{\mathcal{Z}\left\{\mathcal{L}^{-1}[H(s)X(s)]_s\right\}}{\mathcal{Z}\left\{\mathcal{L}^{-1}[X(s)]_s\right\}} \tag{13}$$

where $h(n)$ is the sampled impulse response of the analog filter. Equivalently, $H(z)$ is the z-transform of $h(n)$ (6). When the sampling frequency $f_s = 1/T$ Hz is sufficiently high compared with the stopband frequency of the analog filter, the frequency response characteristics are preserved, and there is noticeable aliasing distortion.

Impulse-Invariant Transform Design. The impulse-invariant transform preserves the impulse response $h(t)$ of the analog filter (7). Setting the input to be an impulse as $x(t) = U_0(t)$ in Eq. (13) gives $X(z) = 1/T$ and

$$H(z) = T\mathcal{Z}[h(n)] = T\mathcal{Z}\left\{\mathcal{L}^{-1}[H(s)]_s\right\} \tag{14}$$

If the analog transfer function $H(s)$ is band-limited to $|\omega| < \omega_s/2$, then the digital filter has exactly the same magnitude, phase, and delay responses of the analog filter for $|\omega| < \omega_s/2$. Otherwise, the frequency responses are not identical because aliasing occurs. Nevertheless, the impulse responses are identical at the sample times.

Suppose that $H(s)$ is causal. Its transfer function can be expanded as a partial fraction as

$$H(s) = d\frac{\prod_{k=1}^{M}(s+z_k)}{\prod_{k=1}^{N}(s+p_k)} = \sum_{k=1}^{N}\frac{K_k}{(s+p_k)}, \quad \mathrm{Re}(s) > s_0 \tag{15}$$

assuming $N > M$. The impulse response of the analog filter equals

$$h(t) = \sum_{k=1}^{N} K_k e^{-p_k t} U_{-1}(t) \tag{16}$$

Taking the z-transform of $h(t)$ and multiplying by T gives the impulse-invariant digital filter gain as

$$\begin{aligned} H_0(z) &= T\mathcal{Z}[h(n)] = T\sum_{k=1}^{N}\frac{K_k z}{z - e^{-p_k T}} \\ &= T'\frac{\prod_{k=1}^{N}(z-c_k)}{\prod_{k=1}^{N}(z-e^{-p_k T})}, \quad |z| > z_0 \end{aligned} \tag{17}$$

Modified Impulse-Invariant Transform Design. An interesting alternative the impulse-invariant transform is the modified impulse-invariant transform (8). For an analog transfer function $H(s) = N(s)/D(s)$, its poles were preserved but its zeros were not in Eq. (17). However, the zeros can also be preserved using the

following method. Express two new transfer functions as

$$H_1(s) = \frac{N(0)}{N(s)}, \quad H_2(s) = \frac{D(0)}{D(s)} \tag{18}$$

which are all-pole low-pass filters. Their impulse-invariant response versions equal

$$\begin{aligned} H_{01}(z) &= \frac{N_1(z)}{\prod_{k=1}^{M}(z - e^{-z_k T})}, \\ H_{02}(z) &= \frac{N_2(z)}{\prod_{k=1}^{N}(z - e^{-p_k T})} \end{aligned} \tag{19}$$

Forming the ratio of these two transfer functions produces the modified impulse-invariant transform as

$$H(z) = \frac{H_{01}(z)}{H_{02}(z)} = \frac{N_2(z)}{N_1(z)} \frac{\prod_{k=1}^{M}(z - e^{-z_k T})}{\prod_{k=1}^{N}(z - e^{-p_k T})} \tag{20}$$

This is the product of the matched-z transform of the next section and a $N_2(z)/N_1(z)$ compensator.

Because this method relies on closely approximating the frequency responses of both the numerator $N(s)$ and denominator $D(s)$ of $H(z)$ regardless of their orders, $H(z)$ well approximates $H(s)$ in ac steady state.

Matched-*z* Transform Design. One of the simplest design methods is the matched-z transform. It uses the z-transform $z = e^{sT}$ to map every analog pole $-p_k$ and every analog zero $-z_k$ into their equivalent digital pole $z_p = e^{-p_k T}$ and zero $z_z = e^{-z_k T}$, respectively. Using this approach, Eq. (15) maps as

$$H(s) = \frac{\prod_{k=1}^{M}(s + z_k)}{\prod_{k=1}^{N}(s + p_k)}, \quad H(z) = \frac{\prod_{k=1}^{M}(z - e^{-z_k T})}{\prod_{k=1}^{N}(z - e^{-p_k T})} \tag{21}$$

We see that its transfer function has a related form to Eqs. (17) and (20). The impulse-invariant transform Eq. (17) has the same denominator but a different numerator. The modified impulse-invariant transform Eq. (20) has the same denominator and numerator but is multiplied by a compensator. The matched-z transform does not preserve frequency-domain characteristics such as magnitude ripple and delay, nor does it preserve time-domain responses. Its major advantage is ease of application.

Complementary Design. On a related topic, a complementary digital filter $H_c(z)$ can be obtained from a digital filter $H(z)$ using

$$H_c(z) = 1 - H(z), \quad h_c(n) = U_{-1}(n) - h(n) \tag{22}$$

The impulse responses of these two filters add to a step function $U_{-1}(n)$ so the responses are said to be complementary. Therefore, these two filters maintain their time domain impulse response characteristics such as delay and rise (fall) times, overshoot (undershoot), etc. If one filter is low-pass, its complementary filter is high-pass. If one filter is bandpass, its complementary filter is bandstop. This is an especially convenient approach because it generates an additional filter with little additional computational cost (1).

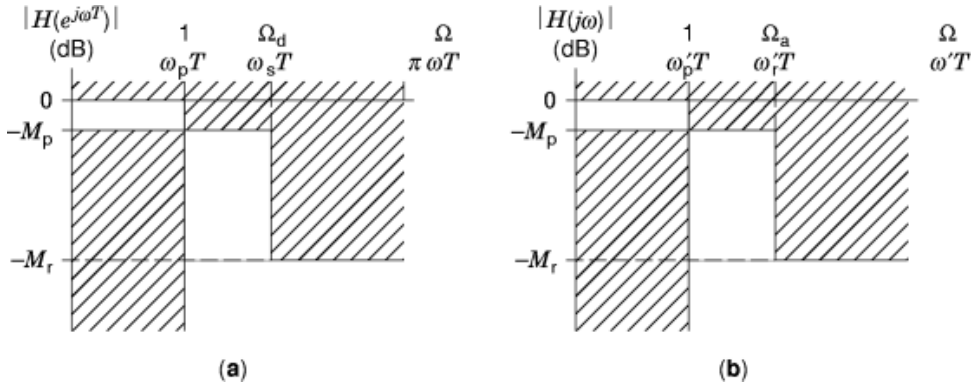

Fig. 2. Magnitude gain response specification of (a) digital filters and (b) analog filters.

Filter Orders

Digital filters generally have frequency-domain specifications like that drawn in Fig. 2 where

f_p = maximum passband frequency (Hz) for M_p (dB) maximum ripple
f_r = minimum stopband frequency (Hz) for M_r (dB) minimum rejection
f_s = sampling frequency (Hz) = $1/T$ seconds

These are usually converted to the normalized frequency form as where
$\omega_p T = 360^\circ f_p/f_s$ = maximum passband frequency in degrees
$\omega_r T = 360^\circ f_r/f_s$ = minimum stopband frequency in degrees
$\Omega_d = \omega_r T/\omega_p T$ = stopband/passband frequency ratio.

From such (M_p, M_r, Ω_r) frequency specifications, the digital filter order n is determined from nomographs using the following procedure. This can be easily done for classical analog filter transfer functions (like Butterworth and Chebyshev) combined with the digital transforms (like bilinear) discussed earlier (9).

Frequency Warping. Consider the bilinear transform of Table 3 and its associated frequency relation

$$s_n = \frac{1}{C}\frac{1-z^{-1}}{1+z^{-1}}, \quad \frac{vT}{2} = \tan\left(\frac{\omega T}{2}\right) \tag{23}$$

where $z = j\omega$ and $s = jv$ (after some manipulation). The analog filter frequencies $v = (0,1,\infty)$ map into the digital filter frequencies $\omega T = (0, BT, \pi)$. Therefore the bilinear transform compresses the high-frequency response of the analog filter into frequencies approaching π radians in the digital filter. Thus the digital filter order will be less than the analog filter order, or at most equal, using this transform.

Other transforms have other constants. For example, the Euler transforms have $vT = \sin(\omega T)$ or $\omega T = \sin^{-1}(vT)$. These transforms expand rather than compress the higher-frequency filter response. The digital filter orders will be no less, and often greater, than the analog filter orders using the Euler transforms.

As the sampling frequency f_s approaches infinity or the sampling interval T approaches zero, all properly formulated discrete transforms produce digital filters whose transfer functions approach that of the analog filter from which they were all derived. For example, the bilinear transform in Eq. (23) has frequencies that

Table 5. Maximum Normalized Frequency to Reduce Frequency Warping Percentage Error for Some Digital Transforms

Transform	Frequency Mapping vT	#1%	#5%	#10%	#20%
$H_{001}, H_{111}, H_{112}$	$\sin(gT)$	148	328	458	658
H_{011}	$2\tan(gT/2)$	208	438	598	808
H_{MBDI}	$2\tan(gT/2)\cos(gT)$	98	208	288	408
H_{LDI}	$2\sin(gT/2)$	288	638	908	1298
H_{ODI}	$4\sin(gT/4)$	568	1268	1808	2598

C. S. Lindquist, *Adaptive & Digital Signal Processing with Digital Filtering Applications*, International Series in Signal Processing and Filtering, vol. 2, p. 397. Miami, FL: Steward & Sons, 1989.

are related as

$$\frac{\omega T}{2} = \tan^{-1}\left(\frac{vT}{2}\right) = \frac{vT}{2} - \frac{1}{3}\left(\frac{vT}{2}\right)^3 + \frac{1}{5}\left(\frac{vT}{2}\right)^5 - \cdots \quad (24)$$

so that $\omega T \cong vT$ at low frequencies. Therefore, the low-frequency responses of all the digital filters and the analog filter must be identical. However, as T increases from zero, this is no longer true, and the frequencies begin to *warp* or diverge. In addition, for most transforms the locus of z no longer remains on the unit circle as s moves up the imaginary axis. The frequency responses cannot remain similar, and considerable distortion begins to appear. This does not occur with the bilinear transform whose locus remains on the unit circle.

To make this important point, we refer to Table 5. The various transforms are listed with their frequency mappings for small ωT. Across the top of the table are listed percentage errors in the $\omega T/vT$ ratios ranging from 1 to 20%. Also listed are the digital frequencies ωT required for the error to be below this limit. For the bilinear transform, normalized frequencies less than 43° will be warped less than 5%. For the Euler transforms, this normalized frequency is reduced to 32°.

Analog Filter Nomographs. Analog filter nomographs are well-known and readily available (9).They can be used directly to determine digital filter orders easily. The digital filter frequencies must be converted into their equivalent or warped analog filter frequencies. The necessary frequency ratios listed in Table 6 must be computed. These ratios are then entered onto the nomograph as shown in Fig. 3. The digital filter frequencies Ω_{d} are converted into their corresponding analog filter frequencies Ω_{a}. These analog filter frequencies are then transferred onto the nomograph as usual, and the analog filter order is determined.

Digital Filter Design Procedure. The design procedure for digital filters using discrete transform is straightforward. It consists of the following steps (1):

- 1a.Select a suitable analog filter type (e.g., Butterworth, Chebyshev, elliptic).
- 1b.Choose a particular s_n-to-z transform from Tables 3 and 4.
- 2.Determine the required analog filter order from the nomograph. Use Table 6.
- 3.Write the analog transfer function $H(s)$ having unit bandwidth and the desired magnitude.
- 4.Compute the normalization constant and the discrete transform from Tables 3 and 4.
- 5.Compute the digital transfer function $H(z)$ by substituting the s_n-to-z transform of step 4 into the analog transfer function $H(s)$ of step 3.
- 6.Implement the transfer function using one of the realization techniques discussed later.

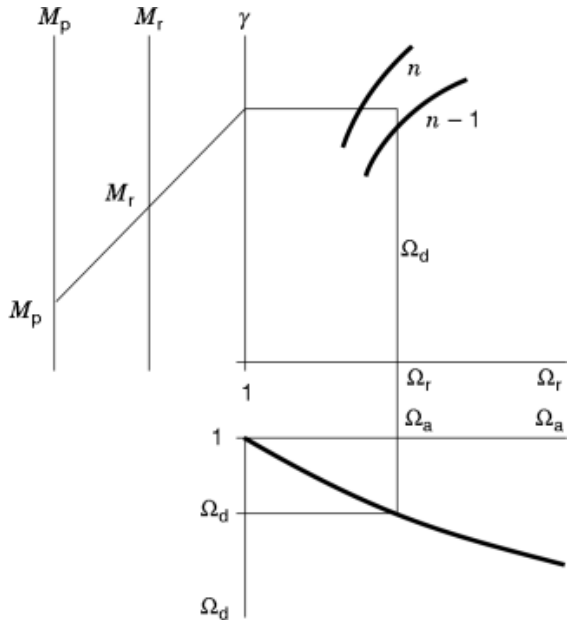

Fig. 3. Use of analog filter nomograph for computing digital filter order.

Table 6. Normalized Analog Stopband Frequency Ratio for Some Digital Transforms

Transform	Analog Frequency Ratio ν_a
$H_{001}, H_{111}, H_{112}$	$\dfrac{\sin(\omega_r T)}{\sin(\omega_p T)}$
H_{011}	$\dfrac{\tan(\omega_r T/2)}{\tan(\omega_p T/2)}$
H_{MBDI}	$\dfrac{\tan(\omega_r T/2)\cos(\omega_r T)}{\tan(\omega_p T/2)\cos(\omega_p T)}$
H_{LDI}	$\dfrac{\sin(\omega_r T/2)}{\sin(\omega_p T/2)}$
H_{ODI}	$\dfrac{\sin(\omega_r T/4)}{\sin(\omega_p T/4)}$

C. S. Lindquist, *Adaptive & Digital Signal Processing with Digital Filtering Applications,* International Series in Signal Processing and Filtering, vol. 2, p. 396. Miami, FL: Steward & Sons, 1989.

Design Examples

Now that we have introduced and described IIR filters, presented their frequency- and time-domain algorithms, and described their design procedure, we now will unify this using design examples. We will design a fourth-order elliptic lowpass filter with 0.28 dB in-band ripple, a minimum of 40 dB stopband rejection, and a 1 kHz

Table 7. Implementation Complexity of Digital Filter Structures (*Depends Upon Analog Filter Structure)

Chapter	Structure	Delays	Multipliers	Summer	Total
12.2	Multiple feedback 1F	N	$2N + 1$	2	$3N + 3$
	2F	N	$2N + 1$	$N + 1$	$4N + 2$
	3F	$2N$	$2N + 1$	1	$4N + 2$
	4F	$2N$	$2N + 1$	$2N$	$6N + 1$
12.3	Biquad section 1F	2	5	2	9
	2F	2	5	3	10
	3F	4	5	1	10
	4F	4	5	4	13
	5F	2	7	3	12
12.4	Cascade 1F	N	$2.5N$	$0.5N + 1$	$4N + 1$
	2F	N	$2.5N$	$1.5N$	$5N$
	3F	$N + 2$	$2.5N$	$0.5N$	$4N + 2$
	4F	$N + 2$	$2.5N$	$1.5N + 1$	$5N + 3$
	5F	N	$3.5N$	$1.5N$	$6N$
12.5	Parallel 1F	N	$2N + 1$	$0.5N + 1$	$3.5N + 1$
	2F	N	$2N + 1$	$N + 1$	$4N + 1$
	3F	$N + 1$	$2N + 1$	$0.5N + 1$	$3.5N + 3$
	4F	$N + 1$	$2N + 1$	$N + 2$	$4N + 4$
	5F	N	$3N + 1$	1	$4N + 2$
12.6	Lattice	N	$3N + 1$	$2N + 1$	$6N + 2$
12.7	Ladder	N	$2N + 1$	$2N$	$5N + 1$
12.8	Analog simulation*	N	$N + 2$	$N + 1$	$3N + 3$
12.9	Wave*	N	$2N$	$5N$	$8N$

C. S. Lindquist, *Adaptive & Digital Signal Processing with Digital Filtering Applications,* International Series in Signal Processing and Filtering, vol. 2, p. 817. Miami, FL: Steward & Sons, 1989.

bandwidth using a 10 kHz sampling frequency. Using tables, the analog transfer function equals

$$H(s) = \frac{0.1363[s_n^2 + 2.0173^2]}{[(s_n + 0.5213)^2 + 0.4828^2][(s_n + 0.1746)^2 + 1.0509^2]} \tag{25}$$

The scaling constant 0.1363 is selected so that the dc gain $H(0)$ is -0.28 dB. The maximum in-band gain is then 0 dB. Most digital filters are designed following the procedure [or some modified version (10)] described in the previous section. We have chosen both the analog filter type (i.e., elliptic) and order (i.e., 4). Now we choose the design method and will now demonstrate each of them as discussed previously (1).

Bilinear Transfer Design Example. We choose the bilinear transform of Table 3. Because the desired 0.28 dB bandwidth is 1 kHz and the sampling frequency is 10 kHz, the normalized digital filter bandwidth is $360^\circ(1 \text{ kHz}/10 \text{ kHz}) = 36^\circ$. The necessary constant then equals $C = \tan(36^\circ/2) = 0.3249 = 1/3.078$. The bilinear transform then equals

$$s = 3.078\frac{z-1}{z+1} \tag{26}$$

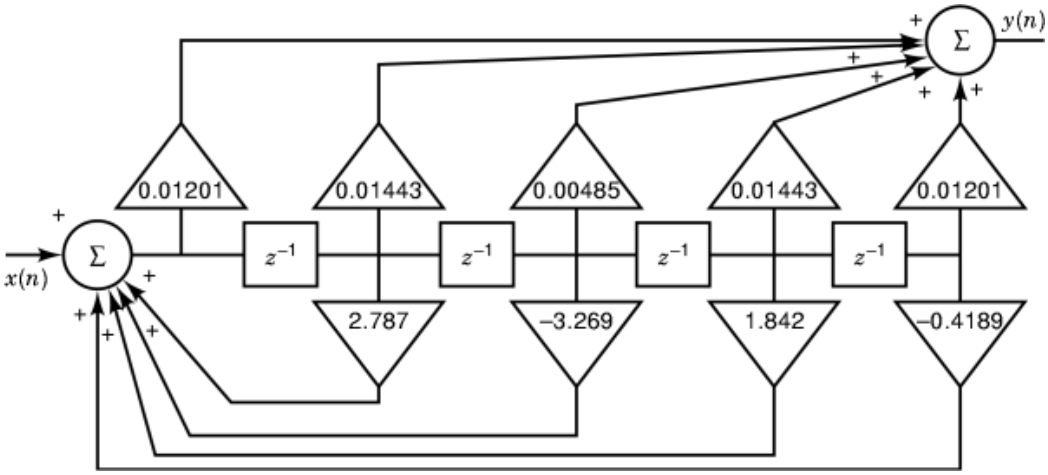

Fig. 4. Multiple feedback (1F) realization of fourth-order elliptic digital filter.

Substituting this into the analog transfer function Eq. (25) produces the digital transfer function

$$H(z) = H(s)\Big|_{s=3.078\frac{z-1}{z+1}} = \frac{0.01201(z+1)^2(z^2-0.7981+1)}{(z^2-1.360z+0.5133)(z^2-1.427z+0.8160)} \tag{27}$$

The bilinear transform preserves the magnitude and phase response behavior of $H(s)$ but with frequency warping.

Impulse-Invariant Transform Design. The impulse-invariant transform preserves the impulse response $h(t)$ of the analog filter. Expressing the gain Eq. (25) as a partial fraction expansion gives

$$H(s) = \frac{0.1129+j0.5467}{s_n+0.5213+j0.4828} + \frac{0.1129-j0.5467}{s_n+0.5213-j0.4828} + \frac{-0.1129-j0.1491}{s_n+0.1746+j1.0509} + \frac{-0.1129+j0.1491}{s_n+0.1746-j1.0509} \tag{28}$$

where $s_n = s/2\pi(1\text{ kHz})$. Taking the inverse Laplace transform, sampling the resulting impulse response $h(t)$ at $T = 1$ ms, and z-transforming the result gives

$$H_0(z) = \frac{0.04203z^2(z^2-0.6527+0.9311)}{(z^2-1.376z+0.5194)(z^2-1.416z+0.8030)} \tag{29}$$

This transform tends to maintain the frequency domain shaping of $H(s)$ but with some aliasing.

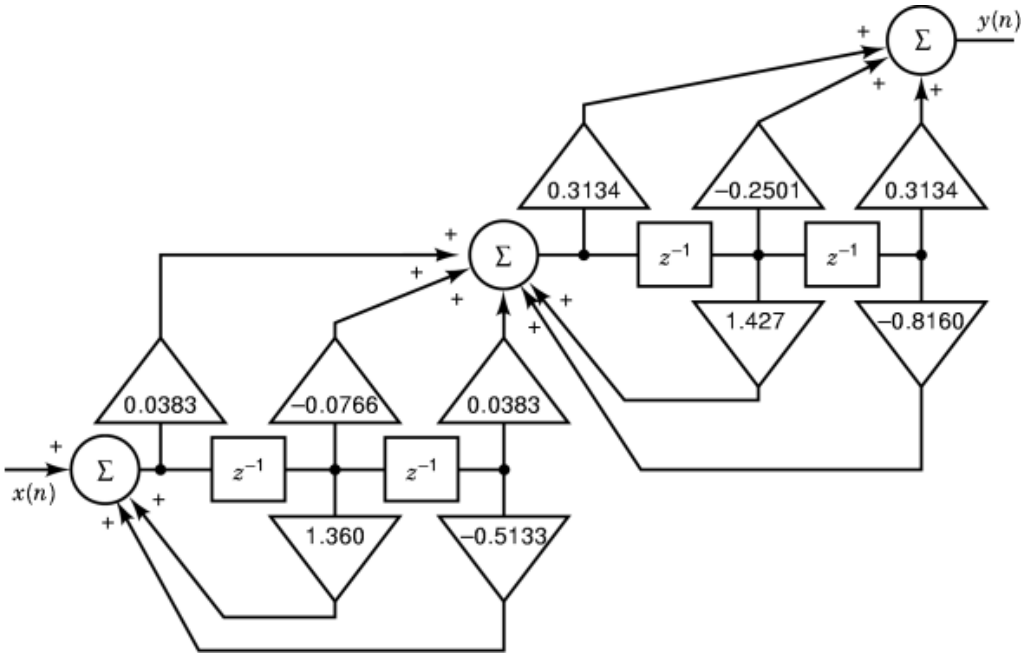

Fig. 5. Cascade (1F) realization of fourth-order elliptic digital filter.

Modified Impulse-Invariant Transform Design. We express the numerator and denominator of the analog filter Eq. (25) as separate transfer functions where

$$H_1(s) = \frac{0.9863 \times 2.0173^2}{s_n^2 + 2.0173^2}$$
$$H_2(s) = \frac{0.5729}{[(s_n + 0.5213)^2 + 0.4828^2][(s_n + 0.1746)^2 + 1.0509^2]} \tag{30}$$

We convert these using the impulse-invariant transform to

$$H_{01}(z) = \frac{1.899z}{z^2 - 0.5973z + 1}$$
$$H_{02}(z) = \frac{0.275z(z^2 + 0.1055z + 0.0218)}{(z^2 - 1.376z + 0.5194)(z^2 - 1.416z + 0.8030)} \tag{31}$$

Forming the ratio H_{02}/H_{01} gives the digital filter as

$$H(z) = \frac{H_{02}}{H_{01}}$$
$$= \frac{0.03398(z^2 + 0.1055z + 0.0218)(z^2 - 0.5973z + 1)}{(z^2 - 1.376z + 0.5194)(z^2 - 1.416z + 0.8030)} \tag{32}$$

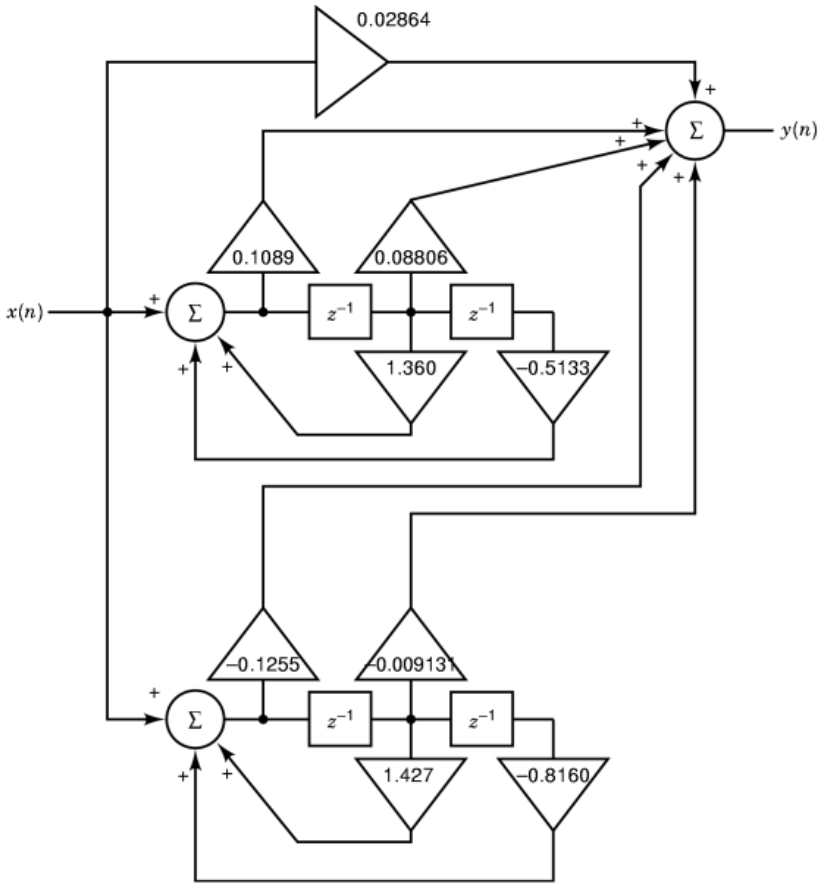

Fig. 6. Parallel (1F) realization of fourth-order elliptic digital filter.

The modified impulse-invariant transform produces a digital filter that more closely approximates the magnitude response of the analog filter.

Matched-*z* Transform Design. The analog filter poles and zeros of Eq. (25) are converted to digital filter poles and zeros using the z-transform $z = e^{sT}$. Converting the poles and zeros gives

$$\begin{aligned}
p_1, p_1^* &= \exp[(-0.5213 \pm j0.4828)2\pi(1\,\text{kHz}/10\,\text{kHz})] \\
&= 0.6878 \pm j0.2153 \\
p_2, p_2^* &= \exp[(-0.1746 \pm j1.0509)2\pi(1\,\text{kHz}/10\,\text{kHz})] \\
&= 0.7077 \pm j0.5496 \\
z_1, z_1^* &= \exp[(\pm j2.0173)2\pi(1\,\text{kHz}/10\,\text{kHz})] \\
&= 0.2987 \pm j0.9544 \\
z_2, z_2^* &= \exp[(-\infty \pm j\pi/2)2\pi(1\,\text{kHz}/10\,\text{kHz})] = 0
\end{aligned} \tag{33}$$

Grouping these terms together, the digital filter transfer function equals

$$H(z) = \frac{kz^2[(z-0.2987)^2 + 0.9544^2]}{[(z-0.6878)^2 + 0.2153^2][(z-0.7077)^2 + 0.5496^2]} = \frac{0.03831z^2(z^2 - 0.5973z + 1)}{(z^2 - 1.376z + 0.5194)(z^2 - 1.416z + 0.8030)} \quad (34)$$

This method is simple and gives fairly good responses.

IIR Filter Realizations

To implement an IIR digital filter $H(z)$ in either hardware or software, usually one of the structures listed in Table 7 must be chosen. These structures are multiple feedback, cascade, parallel, lattice, ladder, analog simulation, and wave. Because each form has many variations, a wide variety of implementations exist. To conserve space, some of these implementation forms are now shown by examples but with no discussion. These filter structures implement the fourth-order 0.28 dB elliptic digital filter whose $H(z)$ is given by Eq. (25) as

$$H(z) = \frac{0.01201(z+1)^2(z^2 - 0.7981 + 1)}{(z^2 - 1.360z + 0.5133)(z^2 - 1.427z + 0.8160)} \quad (35)$$

The different realizations or structures result by expressing $H(z)$ in different forms as will now be shown (1).

Multiple Feedback Structure. The multiple feedback structure uses $H(z)$ in the summation form Eq. (4) as

$$H(z) = \frac{0.01201 + 0.01443z^{-1} + 0.004850z^{-2} + 0.01443z^{-3} + 0.01201z^{-4}}{1 - 2.787z^{-1} + 3.269z^{-2} - 1.842z^{-3} + 0.4189z^{-4}} \quad (36)$$

The constant 0.01201 can be included in the numerator as was done here or treated as an external multiplier. (See Fig. 4.)

Cascade Structure. The cascade structure uses $H(z)$ in the product form of Eq. (3) with biquads as

$$H(z) = \frac{0.03803(1 - 2z^{-1} + z^{-2})}{1 - 1.360z^{-1} + 0.5133z^{-2}} \times \frac{0.3134(1 - 0.7981z^{-1} + z^{-2})}{1 - 1.427z^{-1} + 0.8160z^{-2}} \quad (37)$$

The poles and zeros are paired. Better performance is usually obtained by selecting pole-zero pairs separated by a relatively constant distance. (See Fig. 5.)

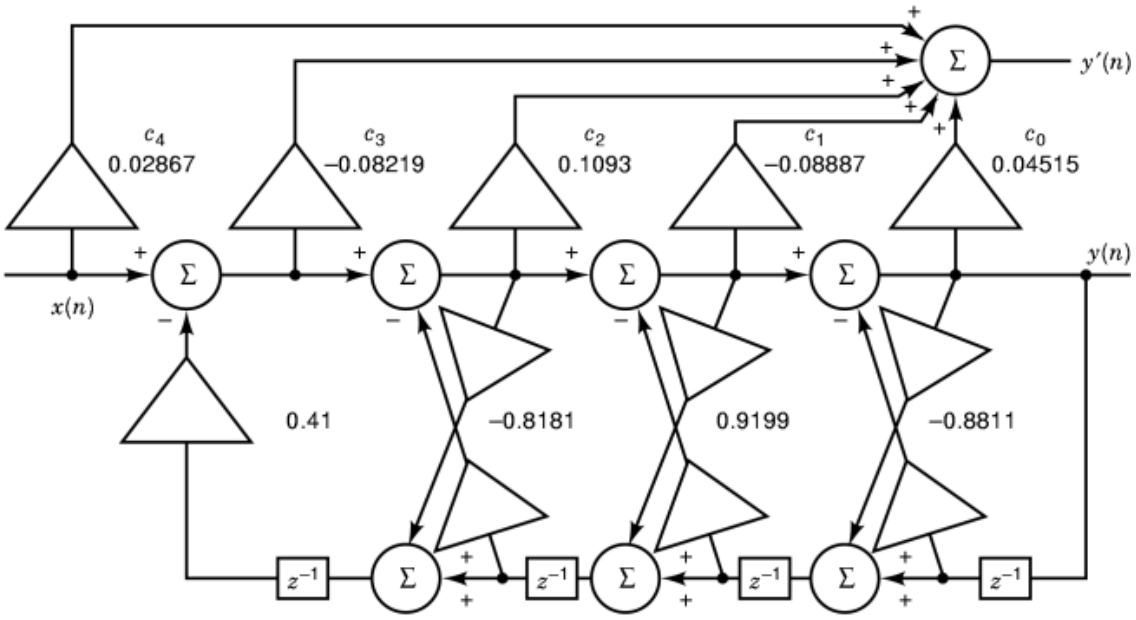

Fig. 7. Lattice realization of fourth-order elliptic digital filter.

Parallel Structure. The parallel structure uses $H(z)$ in the partial fraction expanson form of Eq. (17) with biquads as

$$H(z) = 0.02864 + \frac{0.1089 + 0.08806z^{-1}}{1 - 1.360z^{-1} + 0.5133z^{-2}} + \frac{-0.1255 - 0.009131z^{-1}}{1 - 1.427z^{-1} + 0.8160z^{-2}} \tag{38}$$

Notice here the biquad does not use a z^{-2} term but instead a 0.02864 constant. This reduces complexity. (See Fig. 6.)

Lattice Structure. The lattice structure uses $H(z)$ in a chain matrix product form. The process is standard but lengthy and involves the simultaneous solution of equations. (See Fig. 7.)

Ladder Structure. The ladder structure uses $H(z)$ in the continued fraction expansion or repeated long division form as

$$\frac{H(z)}{0.01201} = 1 + //0.2507z^{-1} - //1.928 - //1.052z^{-1} + //1.173 - //9.807z^{-1} - //0.02066 + //9.132z^{-1} + 2.163 \tag{39}$$

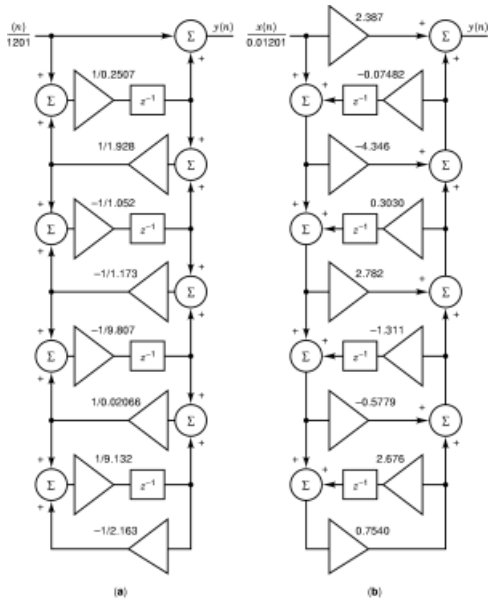

Fig. 8. (a) Cauer 1 and (b) Cauer 2 ladder realizations of fourth-order elliptic digital filter.

for Cauer 1 (see Fig. 8a) and

$$\begin{aligned}\frac{H(z)}{0.01201} = {} & 2.387 + //0.07482z^{-1} - //4.346 \\ & - //0.3030z^{-1} + //2.782 - //1.311z^{-1} \\ & - //0.5779 - //2.676z^{-1} + 0.7540\end{aligned} \tag{40}$$

for Cauer 2 (see Fig. 8b). The // denotes the repeated long division operation.

Analog Simulation Structure. The analog simulation structure uses tabulated *RLC* analog filters which implement standard $H(s)$. This ladder is put into flow graph form in which the L and C terms involve $1/s$. These analog integrator terms are replaced by digital transforms as found in Table 3. This produces $H(z)$ structures. (See Fig. 9.)

Wave Structure. The wave structure of $H(z)$ is a more complicated form of tabulated $H(s)$ *RLC* ladders (1).

IIR Filter Properties

Some of the most important digital filter properties are (1):

(1) Complexity Related to the total number of delays, multipliers, and summers.
(2) Cost Proportional to complexity.
(3) Speed/sampling rate Related to complexity.
(4) Memory Determined by the total number of delay elements and filter coefficients.

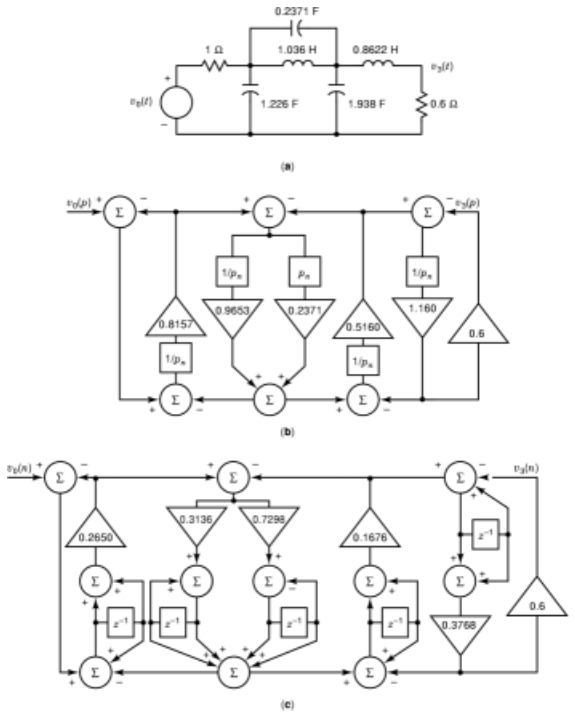

Fig. 9. Analog filter simulation realization of fourth-order elliptic filter showing (a) RLC analog filter, (b) block diagram equivalent, and (c) digital filter.

(5) Sensitivity of pole/zero locations Controlled by word length and arithmetic used in computations (fixed- or floating-point).

(6) Data quantization, coefficient truncation, and product roundoff noise Determined by word length.

(7) Limit cycles Low-level oscillation that continues indefinitely as a result of quantization effects.

(8) Dynamic range Determined by word length, arithmetic used, and filter structure.

A digital filter requires addition, multiplication, and delay z^{-1} elements. The complexity depends directly upon the number of elements required. They are listed in Table 7. Complexity depends indirectly upon filter type (low-pass, high-pass, etc.), the filter gain characteristic (Butterworth, etc.), and the arithmetic used for computations. Table 7 shows that multiple feedback structures are the simplest and wave structures are the most complex. Cost in the general sense is proportional to complexity.

Speed is determined by the speed of the adders, multipliers, and delay (write/read) operations. If these three digital filter elements have about the same speed, then speed is proportional to total complexity. Parallel processing techniques can be used to increase speed.

The memory requirements are dictated by the number of delay elements required (data storage) and the number of filter coefficients used (the a_k and b_k). Memory is minimized by using canonical forms having the minimum number of z^{-1} terms. Almost all the filter forms are canonical. The 3F and 4F multiple feedback forms should be avoided because they require twice the number of delay elements.

The sensitivity of the pole–zero locations of the filter and its response depends upon the word length (i.e., finite word size and coefficient truncation) and the type of arithmetic used in the computations. Generally floating-point arithmetic and long coefficient lengths produce the lowest sensitivity.

BIBLIOGRAPHY

1. C. S. Lindquist *Adaptive & Digital Signal Processing with Digital Filtering Applications.* Miami, FL: Steward & Sons, 1989.
2. A. V. Oppenheim A. S. Willsky *Signals & Systems*, 2nd ed., Chaps. 9–10. Englewood Cliffs, NJ: Prentice-Hall, 1997.
3. R. C. Gonzalez P. Wintz *Digital Image Processing*, 2nd ed. Reading, MA: Addison-Wesley, 1987.
4. S. J. Orfanides Introduction to Signal Processing, Chap. 3.5. Englewood Cliffs, NJ: Prentice-Hall, 1996.
5. J. G. Proakis D. M. Manolakis *Digital Signal Processing*, 3rd ed., Chap. 8.3. Englewood Cliffs, NJ: Prentice-Hall, 1996.
6. S. D. Stearns D. R. Rush *Digital Signal Analysis*, Chap. 11. Englewood Cliffs, NJ: Prentice-Hall, 1990.
7. A. V. Oppenheim R. W. Schafer *Discrete-Time Signal Processing*, Chap. 7.1. Englewood Cliffs, NJ: Prentice-Hall, 1989.
8. A. Antoniou *Digital Filters: Analysis and Design*, Chap. 7.4. New York: McGraw-Hill, 1979.
9. C. S. Lindquist *Active Network Design with Active Filtering Applications.* Miami, FL: Steward & Sons, 1977.
10. S. D. Stearns R. A. David *Signal Processing Algorithms*, Chap. 7. Englewood Cliffs, NJ: Prentice-Hall, 1988.

CLAUDE S. LINDQUIST
University of Miami