DIGITAL FILTERS

FILTERS, DIGITAL

DISCRETE-TIME FILTERS

Most phenomena in nature occur in continuous time, such as temperature change, lifetime of a human being, wind speed at a given location, and so on. As a result, if we intend to design a system to interfere with or to measure a natural phenomenon, the system should be analog. A widely used procedure to design systems with interaction with a natural phenomenon is to convert some quantities from the nature into electric signals. Electric signals, which are represented by voltage or current, have a continuous-time form. However, continuous-time signals are not suitable to be processed using computer-type processors (digital machines), which are meant to deal with sequential computation involving numbers. Fortunately, many signals taken from nature can be fully represented by their sampled versions, where the sampled signals coincide with the original analog signals at predefined time instants. Let's take a real live example by supposing we are watching a movie at home. If the movie is monotonous, we can pay attention to what is happening in the movie only from time to time and still understand the story. On the other hand, if the movie gives important information at short periods of time, we can not miss it for a long time. In the latter case, the director already made a tough sample of the story for the viewer. In conclusion, if we know how fast the important information changes, we can always sample and convert the information in numbers for a fast enough digital machine. Fortunately, the electronic technology is at our side, by allowing very fast digital processors to be built at a reasonable cost. This is one of the reasons the so called digital filters, which are filters suitable to process sampled signals implemented in digital machines, are replacing the analog filters in a number of applications. Also, there are a number of signals that are originally discrete-time, take for example the stock-market daily financial indicators.

The rapid development of high-speed digital integrated circuit technology in the last three decades has made digital signal processing not only a tool for the simulation of analog systems but also a technique for the implementation of very complex systems. Digital signal processing has found applications in many areas such as image processing, multimedia systems, speech analysis and synthesis, mobile radio, sonar, radar, biomedical engineering, seismology, and modern communication systems.

The main advantages of digital systems relative to analog systems are high reliability, ease of modifying the characteristics of the filter, and low cost. These advantages motivated the digital implementation of many signal processing systems, which were usually implemented with analog circuit technology. In addition, a number of new applications became viable after the availability of the very-large-scale integration (*VLSI*) technology. Usually in the VLSI implementation of a digital signal processing system the concern is in reducing power consumption or area, or in increasing the circuits speed in order to meet the demands of high-throughput applications.

The digital filter is in general the most important tool in most digital signal processing systems. The digital filter processes signals that are discrete in time and in amplitude, that is, signals occurring at distinct and usually equidistant times that can assume a discrete set of amplitude values. In this article, we are primarily concerned with linear, shift-invariant digital filters implemented using finite-precision arithmetic.

In practice, a digital filter is implemented using software on a general-purpose digital computer or a digital signal processor (*DSP*), or by using application-specific hardware usually in the form of an integrated circuit. In any type of implementation, quantization errors are inherent due to finite-precision arithmetic. In implementations for specific applications there are techniques such as algorithms and topologies for digital filters that allow us to meet low-power, low-area, and/or high-speed specifications.

The quantization errors can be classified as follows:

- Roundoff errors resulting when the internal signals like the output of multipliers are quantized before or after additions
- Errors in the magnitude and phase response of the filter caused by the use of finite wordlength for the representation of the multiplier coefficients
- Errors due to the representation of the input signal with a set of discrete levels

The quantization errors described depend on the type of arithmetic used in the actual implementation. If the digital filter is implemented on a general-purpose processor or a DSP, floating-point arithmetic is usually available; therefore this type of arithmetic is the choice. On the other hand, if the digital filter is implemented by means of application-specific hardware or lower cost DSPs, fixed-point arithmetic is usually the best choice because of its low complexity in terms of silicon area for the hardware. In this article, only fixed-point arithmetic is addressed.

DIGITAL FILTERS

In a digital filter represented in the block diagram of Fig. 1, the input signal $x(n)$ is a sequence of numbers, indexed by the integer n, which can assume only a finite number of amplitude values. Such input sequence comes, most of the time, from an analog (or continuous-time) signal $x(t)$ by periodically sampling it at the time instants $t = nT$, where T is called the sampling interval. The output sequence $y(n)$ is the response of the digital filter when excited by the input $x(n)$, with the relationship between $x(n)$ and $y(n)$ represented by the operator $\mathcal{H}$ as

$$y(n) \equiv \mathcal{H}[x(n)] \qquad (1)$$

The most important class of digital filters is composed by linear, time-invariant (*LTI*) and causal filters. A *linear* digital filter is one whose response to a weighted sum of input signals is equal to the same weighted sum of the

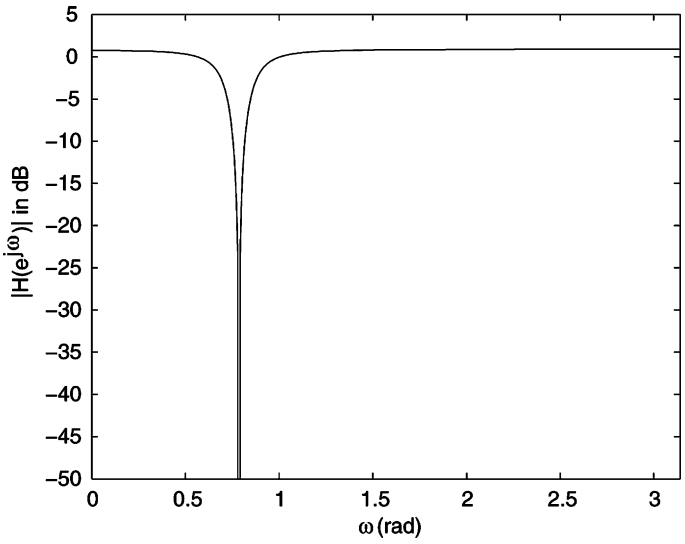

Figure 1. Frequency response of the notch filter of Eq. (7) with $r = 0.9$ and $\theta = \pi/4$.

corresponding individual responses, that is,

$$\mathcal{H}[\alpha x_1(n) + \beta x_2(n)] = \alpha \mathcal{H}[x_1(n)] + \beta \mathcal{H}[x_2(n)] \qquad (2)$$

for any sequences $x_1(n)$ and $x_2(n)$, and any arbitrary constants α and β. A digital filter is said to be *time invariant* when its response to an input sequence is always the same, independent of the time instant when the input is applied to the filter (assuming that the filter is always operating under the same initial conditions); that is, if $\mathcal{H}[x(n)] = y(n)$, then

$$\mathcal{H}[x(n - n_0)] = y(n - n_0) \qquad (3)$$

for all integers n and n_0. A *causal* digital filter is one whose response does not depend on the future values of the excitation signal. Therefore, for any two input sequences $x_1(n)$ and $x_2(n)$ such that $x_1(n) = x_2(n)$ for $n \le n_0$, the corresponding responses of the digital filter (with same initial conditions) are identical, that is,

$$\mathcal{H}[x_1(n)] = \mathcal{H}[x_2(n)], \quad \text{for } n \le n_0 \qquad (4)$$

An LTI digital filter is completely characterized by its response to the unit sample or impulse sequence $\delta(n)$ (assuming it is initially relaxed). The impulse sequence is defined as

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \qquad (5)$$

and the filter response when excited by such a sequence is denoted by $h(n)$ and it is referred to as *impulse response* of the digital filter. Observe that if the digital filter is causal, then $h(n) = 0$ for $n < 0$. An arbitrary input sequence can be expressed as a sum of delayed and weighted impulse sequences; that is,

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n - k) \qquad (6)$$

and the response of an LTI digital filter to $x(n)$ can then be expressed by

$$\begin{aligned} y(n) &= \mathcal{H}\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n - k)\right] \\ &= \sum_{k=-\infty}^{\infty} x(k)\mathcal{H}[\delta(n - k)] \\ &= \sum_{k=-\infty}^{\infty} x(k)h(n - k) \end{aligned} \qquad (7)$$

The summation in the last line of the above expression, called the *convolution sum,* relates the output sequence of a digital filter to its impulse response $h(n)$ and to the input sequence $x(n)$. The convolution operation is represented by

$$y(n) = x(n) * h(n) \qquad (8)$$

By applying a change of variables in the summation of Eq. (7), one can verify that the convolution operation is commutative; that is, the output of a digital filter with impulse response $h(n)$ and input $x(n)$ is also given by

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{\infty} h(k)x(n - k) \\ &= h(n) * x(n) \end{aligned} \qquad (9)$$

Defining the z transform of a sequence $x(n)$ as

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \qquad (10)$$

the *transfer function* of a digital filter is the ratio of the z transform of the output sequence to the z transform of the input signal; that is,

$$H(z) = \frac{Y(z)}{X(z)} \qquad (11)$$

Taking the z transform of both sides of the convolution expression of Eq. (9), that is,

$$\sum_{n=-\infty}^{\infty} y(n)z^{-n} = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h(k)x(n - k)z^{-n} \qquad (12)$$

and substituting variables ($m = n - k$),

$$\sum_{n=-\infty}^{\infty} y(n)z^{-n} = \sum_{k=-\infty}^{\infty} h(k)z^{-k} \sum_{m=-\infty}^{\infty} x(m)z^{-m} \qquad (13)$$

the following relation among the z transforms of the output $Y(z)$, of the input $X(z)$ and of the impulse response $H(z)$ of a digital filter is obtained:

$$Y(z) = H(z)X(z) \qquad (14)$$

Hence, the transfer function of an LTI digital filter is the z transform of its impulse response.

SAMPLING RATE

Most of the signals encountered in science, such as speech, biological signals, seismic signals, radar, and sonar, are analog. To process them by a digital filter, they need to be sampled and converted to digital by an analog-to-digital (A/D) converter.

The *sampling theorem* states that a bandlimited analog signal $x(t)$ whose highest frequency component is at the frequency f_{max} can be exactly recovered from its sample values $x(n)$ at the time instants $t = nT$, if the sampling frequency $f_s = 1/T$ is larger than twice f_{max}. The sampling rate $2f_{max}$ is called the *Nyquist rate.* The original continuous-time signal can be recovered from the sampled signal $x(n)$ by the interpolation formula

$$x(t) = \sum_{n=-\infty}^{\infty} x(n) \frac{\sin[\omega_s(t - nT)/2]}{\omega_s(t - nT)/2} \tag{15}$$

with $\Omega_s = 2\pi f_s$.

The recovery of an analog signal from its samples by the above interpolation formula is impractical, because it involves the summation of infinite duration functions and the knowledge of future samples of the signal $x(t)$ involved in the summation. Practical circuits which convert back the filtered signal to analog form are called digital-to-analog (D/A) converters.

In general, if an analog signal $x(t)$ is sampled with a sampling frequency f_s smaller than twice its maximum frequency f_{max}, then distinct frequency components of $x(t)$ will be mixed, causing an undesirable distortion in the recovered continuous-time signal referred to as *aliasing.*

FREQUENCY RESPONSE

The response of an LTI digital filter to a complex exponential (or complex sinusoid) of radian frequency Ω, that is, $x(n) = e^{j\Omega n}$ for $-\infty < n < \infty$, is a complex exponential of the same frequency Ω with a possible complex amplitude modification. Such property can be verified from the convolution expression of Eq. (9), where the output of a digital filter with impulse response $h(n)$ and excited by the complex exponential of frequency Ω is given by

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{\infty} h(k) e^{j\omega(n-k)} \\ &= e^{j\omega n} \sum_{k=-\infty}^{\infty} h(k) e^{-j\omega k} \\ &= x(n) H(z)|_{z=e^{j\omega}} \end{aligned} \tag{16}$$

In the above expression, $H(e^{j\Omega})$ describes the changes in amplitude introduced by the digital filter to the complex exponential input signal. Such function of Ω is called the *frequency response* of the digital filter, and it corresponds to the transfer function $H(z)$ evaluated on the unit circle in the z plane ($|z| = |e^{j\Omega}| = 1$).

Observe that if the complex sinusoidal sequence comes from sampling an analog sinusoidal signal, the relation between the frequency of the discrete-time sinusoid Ω and the frequency of the continuous-time analog sinusoid Ω_a is obtained by making $t = nT$ in the analog signal and equating both signals, resulting in $\Omega = \Omega_a T = \Omega_a / f_s$. Hence, the digital frequency is equivalent to the analog frequency normalized by the sampling frequency, and, therefore, is always between $-\pi$ and π if the sampling theorem is satisfied. The low frequencies are the frequencies Ω close to zero, whereas the high frequencies are the frequencies close to π, with $\Omega = \pi$ corresponding to the Nyquist frequency ($\Omega_a = \Omega_{s/2}$).

The frequency response $H(e^{j\Omega})$ is a periodic function of Ω with period 2π; that is,

$$H[e^{j(\omega + 2\pi k)}] = H(e^{j\omega}) \tag{17}$$

for any integer k. Thinking in terms of analog frequency (for sampled analog signals), the function $H(e^{j\Omega_a T})$ is periodic in Ω_a with period $2\pi/T = \Omega_s$. This periodicity is explained by observing that continuous-time sinusoidal signals of frequencies Ω_a and $\Omega_a + k\Omega_s$ result, when sampled, in identical sequences. Therefore, both signals must produce the same output when processed by the same digital filter.

In general, $H(e^{j\Omega})$ is a complex-valued function, which can be expressed in polar form as

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{j \angle H(e^{j\omega})} \tag{18}$$

where $|H(e^{j\Omega})|$ and $\angle H(e^{j\Omega})$ are called the *magnitude response* and *phase response,* respectively, of the digital filter.

A large class of sequences can be expressed in the form

$$x(n) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(e^{j\omega}) e^{j\omega n} \, d\omega \tag{19}$$

where

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} \tag{20}$$

$X(e^{j\Omega})$ is called the *Fourier transform* of the sequence $x(n)$. A sufficient but not necessary condition for the existence of the Fourier transform $X(e^{j\Omega})$ is that the sequence $x(n)$ is absolutely summable, that is,

$$\lim_{M \to \infty} \sum_{n=-M}^{M} |x(n)| < \infty$$

Using the above Fourier representation, the input sequence can be written as the sum of the complex exponentials $e^{j\Omega n}$ weighted by $X(e^{j\Omega}) d\Omega$. From the superposition property of linear systems, the output of the digital filter with frequency response $H(e^{j\Omega})$ and with $x(n)$ as input is given by the corresponding sum of the responses to each complex exponential, that is,

$$y(n) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(e^{j\omega}) X(e^{j\omega}) e^{j\omega n} \, d\omega \tag{21}$$

Hence, each frequency component of the input sequence $x(n)$ is modified by the frequency response of the digital filter at the corresponding frequency.

From Eq. (22) and from the definition of the Fourier transform in Eq. (20), the frequency response of a digital filter is the Fourier transform of its impulse response. From Eq. (22), the Fourier transform of the output of a digital filter is given by the product of the Fourier transforms of the input and of the impulse response, that is,

$$Y(e^{j\omega}) = H(e^{j\omega}) X(e^{j\omega}) \tag{22}$$

Filters that select the low-frequency components of the input signal are called *low-pass* filters; those that select only high-frequency components of the input are called *high-pass* filters; *bandpass* and *bandstop* filters keep and reject,

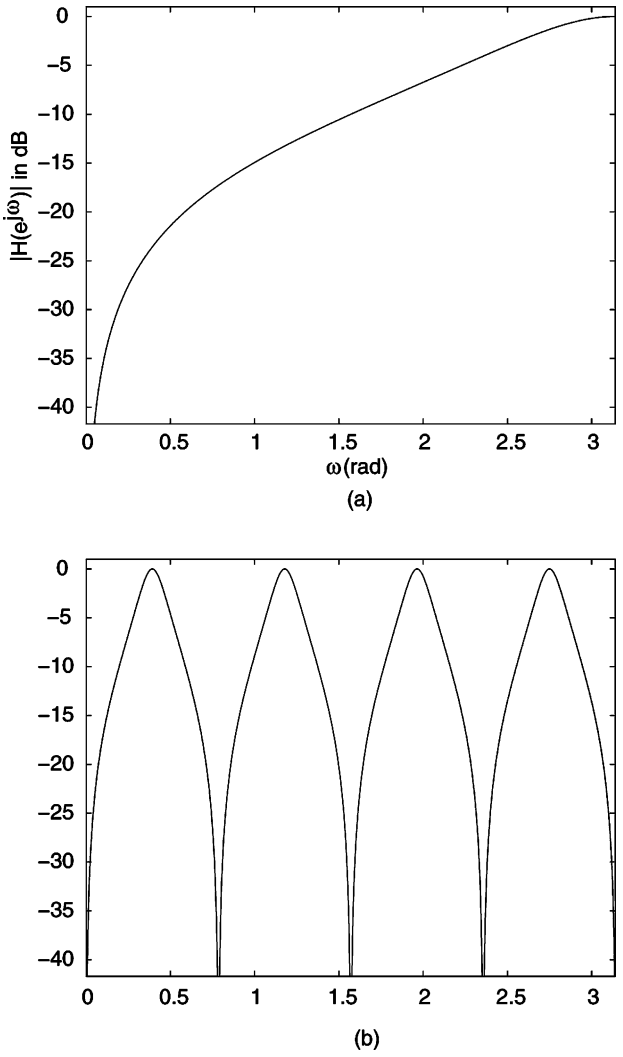

Figure 2. (a) Frequency response of highpass filter of Eq. (8); (b) Frequency response of comb filter of Eq. (9).

respectively, components in a frequency band in the interval $0 \leq \Omega < \pi$. The ideal frequency responses of such filters are illustrated in Fig. 2.

DIFFERENCE EQUATIONS

A large and important subclass of linear time-invariant digital filters consists of the filters whose input and output sequences satisfy an equation of the form

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{m=0}^{M} b_m x(n-m) \tag{23}$$

where a_k and b_m are constant coefficients. The above equation is referred to as an Nth order difference equation. For a causal filter with input–output related by the above difference equation with coefficients scaled such that $a_0 = 1$, the present value of the output $y(n)$ can be computed from the N past output values and from the present and M past input values by

$$y(n) = \sum_{m=0}^{M} b_m x(n-m) - \sum_{k=1}^{N} a_k y(n-k) \tag{24}$$

In the particular case when $a_k = 0$ for $k = 1, \ldots, N$, the digital filter implemented by the above equation is called a *nonrecursive* filter, because there is no feedback from the past output in the computation of its present value. When there is feedback, that is, $a_k \neq 0$ for at least one k for $k = 1, \ldots, N$, the filter implementation given by the above difference equation is called recursive.

The computation of each output value by Eq. (25) requires the storage of past samples of the input and output sequences, the multiplication of these samples by the corresponding coefficients of the difference equation, and the addition of the results of such multiplications. Therefore, the calculation of $y(n)$ can be represented in a block diagram through the interconnection of the three basic elements, with symbols shown in Fig. 3: the unit delay, the multiplier, and the adder. The unit delay is represented by z^{-1}, which is the transfer function associated with it. The block diagram corresponding to Eq. (25) when there is feedback (recursive implementation) is given in Fig. 4. Such filter implementation is called a *direct form I* structure. Another implementation of recursive digital filters which satisfies Eq. (24) is based on the following pair of equations

$$\begin{aligned} v(n) &= -\sum_{k=1}^{N} a_k v(n-k) + x(n) \\ y(n) &= \sum_{m=0}^{M} b_m v(n-m) \end{aligned} \tag{25}$$

The block diagram of the resulting implementation is shown in Fig. 5 and is called a *direct form II* structure. The direct form II realization requires a number of unit delays (or memory locations) equal to the maximum value of M and N. This value is the minimum number of delays needed to obtain the output of the filter satisfying Eq. (24), and, therefore, the direct form II structure is said to be *canonic.*

The block diagram corresponding to the nonrecursive case is shown in Fig. 6, where there is no feedback of the past output values.

The transfer function of a system with input and output related by a difference equation can be obtained by taking the z transform of both sides of Eq. (24), that is,

$$\sum_{k=0}^{N} a_k z^{-k} Y(z) = \sum_{m=0}^{M} b_m z^{-m} X(z) \tag{26}$$

where we have used the linearity property of the z transform and the fact that the z transform of a delayed sequence $x(n - n_d)$ is given by $z^{-n_d} X(z)$, where $X(z)$ is the z transform of $x(n)$. Thus, from the above equation,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{\sum_{k=0}^{N} a_k z^{-k}} \tag{27}$$

Therefore, the transfer function of a digital filter that satisfies Eq. (24) is a rational function of z; that is, it is given by a ratio of polynomials in z, with the coefficients of such poly-

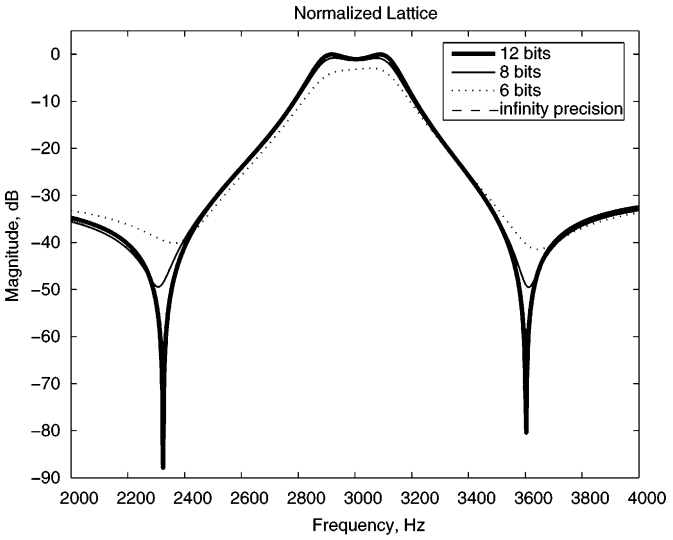

Figure 3. Frequency responses of a fourth-order elliptic filter realized with the normalized lattice structure, with coefficients quantized to 12, 8 and 6 bits.

nomials equal to the coefficients of the difference equation. The values of z for which $H(z) = 0$ are called the *zeros* of the transfer function of the digital filter, and are the roots of the numerator polynomial of $H(z)$. The roots of the denominator polynomial of $H(z)$ are called the *poles* of the digital filter's transfer function, and are the values of z for which $H(z)$ is infinite. The transfer function $H(z)$ can be written in terms of its poles p_k and zeros z_m as

$$H(z) = \frac{b_0 z^{N-M} \prod_{m=1}^{M}(z - z_m)}{a_0 \prod_{k=1}^{N}(z - p_k)} \tag{28}$$

The above factored form of $H(z)$ can be useful for estimating the frequency response of a digital filter from its zeros and poles. From Eq. (16), the frequency response of a digital filter is equal to its transfer function evaluated on the unit circle in the z plane, that is, at $z = e^{j\Omega}$. Representing the differences $(z - z_m)$ for $z = e^{j\Omega}$ in the z plane by the vectors C_m and the differences $(z - p_k)$ for $z = e^{j\Omega}$ by the vectors D_k, we can express the magnitude and phase of the frequency response by

$$|H(e^{j\omega})| = \frac{b_0 \prod_{m=1}^{M} |C_m|}{a_0 \prod_{k=1}^{N} |D_k|} \tag{29}$$

and

$$\angle H(e^{j\omega}) = (N - M)\omega + \sum_{m=1}^{M} \angle C_m - \sum_{k=1}^{N} \angle D_k \tag{30}$$

where $|C_m|$ and $|D_k|$ represent the magnitudes of the vectors C_m and D_k, and $\angle C_m$ and $\angle D_k$ represent the angles of the vectors C_m and D_k as related to the real axis measured counterclockwise, respectively.

Figure 7 illustrates the pole–zero diagram as well as the vectors defined above for the second-order digital filter

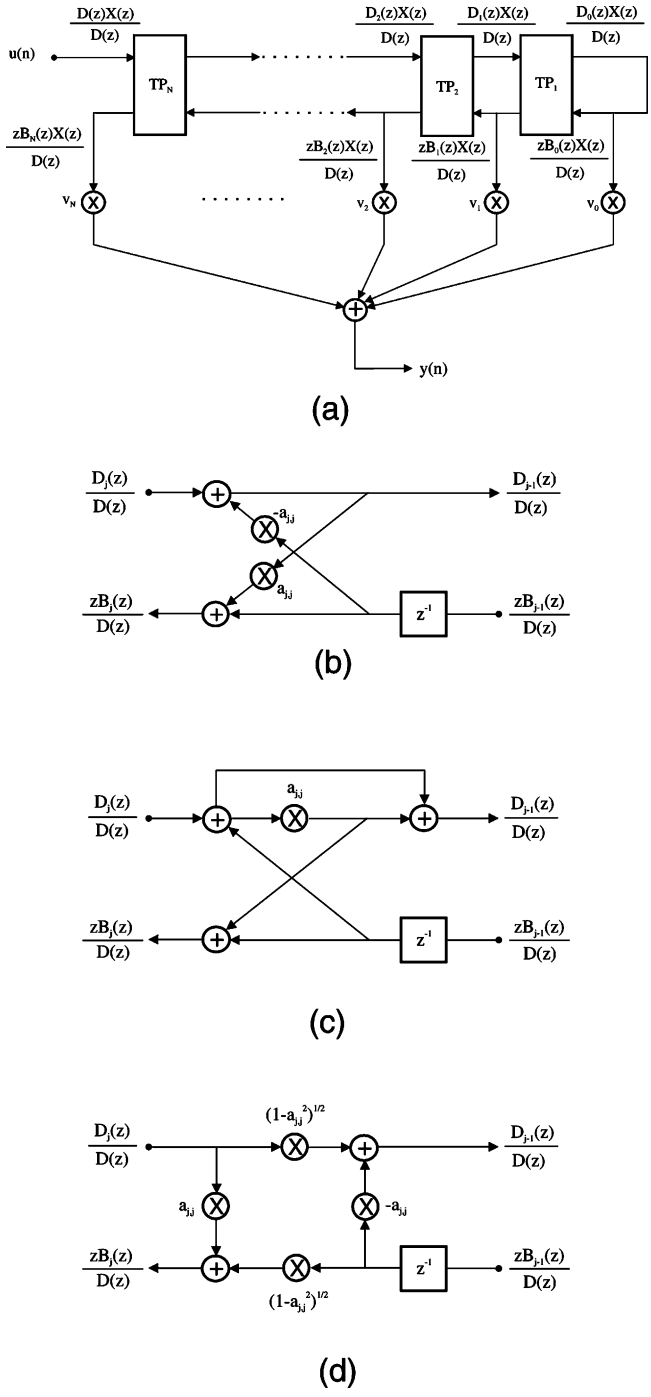

Figure 4. Recursive implementation of a digital filter obtained from the difference equation given in Eq. (25) (direct form I structure).

with transfer function

$$H(z) = \frac{z^2 - 0.9\sqrt{3}z + 0.81}{z^2 - 0.8\sqrt{2}z + 0.64} = \frac{[z - 0.9e^{j(5\pi/6)}][z - 0.9e^{-j(5\pi/6)}]}{[z - 0.8e^{j(\pi/4)}][z - 0.8e^{-j(\pi/4)}]} \tag{31}$$

One can observe that for frequencies Ω near the zeros, $|H(e^{j\Omega})|$ will be very small since the zeros are close to the unit circle and the vectors from the zeros to $e^{j\Omega}$ will have small magnitudes. The phase response, $\angle H(e^{j\Omega})$, will

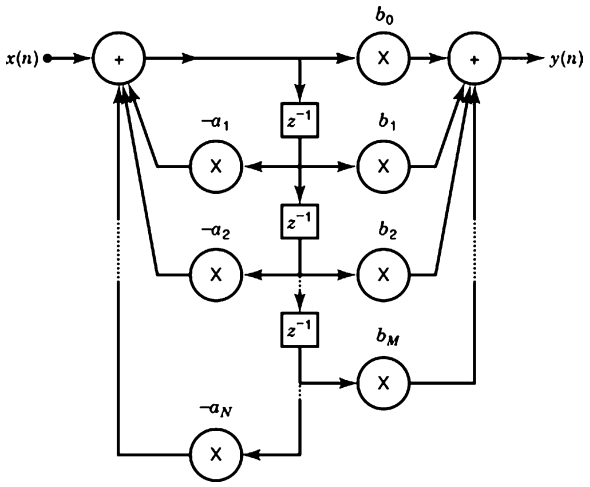

Figure 5. Canionic recursive implementation obtained from the pair of difference equations given in Eq. (26) (direct form II structure).

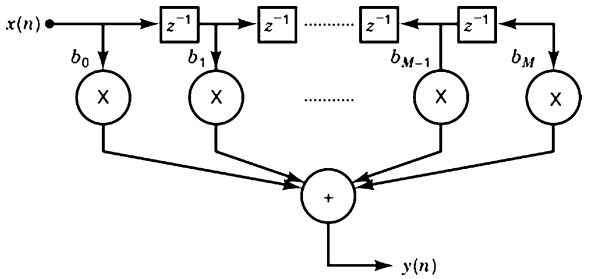

Figure 6. Nonrecursive implementation of a digital filter from the difference equation given in Eq. (25) with $a_k = 0$ for $k = 1, \ldots, N$.

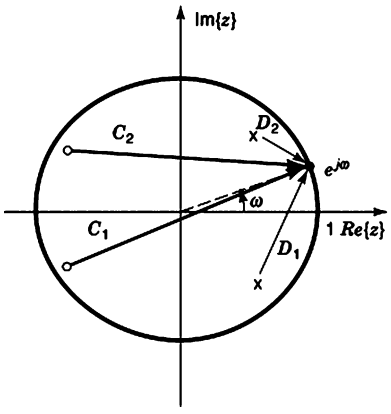

Figure 7. Geometric frequency response evaluation from the poles and zeros of the transfer function $H(z)$ of Eq. (32).

change by almost π rad near the zeros frequencies. For frequencies close to the poles, there will be a peak in $|H(e^{j\Omega})|$, and $\angle H(e^{j\Omega})$ will change by almost $-\pi$ rad. Such frequency response estimation can be verified in the plots of the magnitude and phase responses shown in Fig. 8.

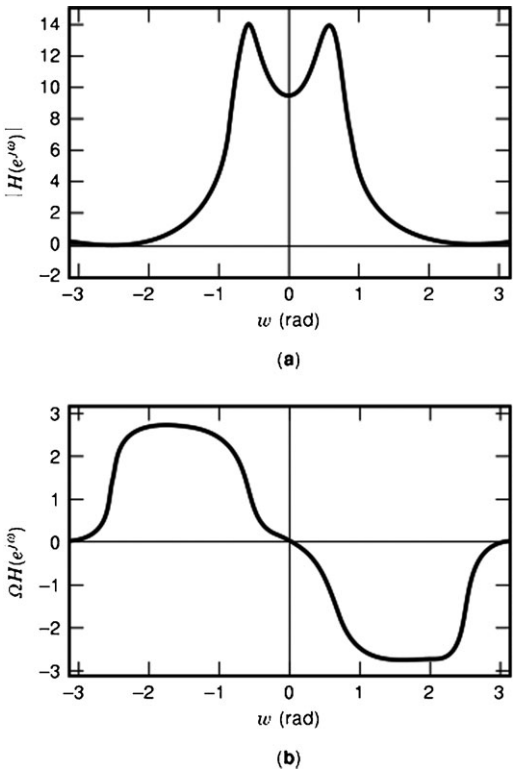

Figure 8. Frequency response of the digital filter with transfer function $H(z)$ given in Eq. (32).

FINITE IMPULSE RESPONSE FILTERS

A digital filter whose impulse response is of finite duration (i.e., it is zero outside a finite interval) is called a *finite impulse response* (*FIR*) filter. From the convolution expression of Eq. (9), the output of a causal FIR filter with $h(n) = 0$ for $n < 0$ and for $n > M$ is given by

$$y(n) = \sum_{m=0}^{M} h(m)x(n-m) \qquad (32)$$

Comparing the above expression with the output of the difference equation given in Eq. (25), we can observe that the output of an FIR filter can be obtained by the nonrecursive implementation of Fig. 4 with the multipliers b_m equal to the impulse response samples $h(m)$. Usually FIR filters are implemented nonrecursively, even though recursive implementations of FIR filters are also possible resulting in a reduction in the number of multipliers in some cases.

The transfer function of an FIR filter has the form

$$H(z) = \sum_{m=0}^{M} h(m)z^{-m} \qquad (33)$$

which has all of its M poles at $z = 0$. Hence, an FIR filter is characterized by the locations of the zeros of its transfer function.

A simple example of an FIR filter is the moving average filter, whose output is the average of the present and the

last M samples of the input sequence; that is,

$$y(n) = \frac{1}{M+1}\sum_{m=0}^{M} x(n-m) \tag{34}$$

The impulse response of this system is

$$h(n) = \begin{cases} \dfrac{1}{M+1}, & \text{for } 0 \le m \le M \\ 0, & \text{for } m < 0 \text{ and } m > M \end{cases} \tag{35}$$

and its transfer function is given by

$$\begin{aligned} H(z) &= \frac{1}{M+1}\sum_{m=0}^{M} z^{-m} \\ &= \frac{1}{M+1}\left[\frac{1-z^{-(M+1)}}{1-z^{-1}}\right] \end{aligned} \tag{36}$$

The zeros of $H(z)$ are at $z_m = e^{j[2\pi m/(M+1)]}$ for $m = 1, \ldots, M$. From the first line of the above equation, the output of this system can be obtained by the nonrecursive implementation of Fig. 6 with coefficients $b_m = 1/(M+1)$, for $m = 0, \ldots, M$. The second line of the above expression suggests a recursive implementation such as that of Fig. 4, with the nonzero coefficients given by $a_1 = -1$, $b_0 = 1$, and $b_{M+1} = -1$.

FIR filters are specially useful for implementing linear-phase transfer functions, that is, transfer functions $H(z)$ such that

$$\angle H(e^{j\omega}) = -\alpha\omega - \beta \tag{37}$$

The response of a linear-phase digital filter to a complex sinusoid of frequency Ω is given by

$$y(n) = |H(e^{j\omega})|e^{-j\beta}e^{j\omega(n-\alpha)} \tag{38}$$

which for α integer is

$$y(n) = |H(e^{j\omega})|e^{-j\beta}x(n-\alpha) \tag{39}$$

Hence, the phase modification introduced by the linear phase filter in the sinusoidal input signal corresponds to a constant delay that is independent of Ω. From the superposition property of LTI systems, an arbitrary signal $x(n)$ filtered by a linear phase filter will have all its frequency components delayed by the same amount α. Defining the group delay of a filter as

$$\tau(\omega) = -\frac{d\angle H(e^{j\omega})}{d\omega} \tag{40}$$

a linear phase filter with phase response as in Eq. (38) has a constant group delay α.

An FIR filter with linear phase response can be easily obtained by imposing one of the symmetry conditions below on the impulse response of the filter:

$$h(n) = h(M-n) \tag{41}$$

or

$$h(n) = -h(M-n) \tag{42}$$

The transfer function of an FIR filter satisfying one of the above conditions with M even is

$$\begin{aligned} H(z) &= \sum_{m=0}^{M} h(m)z^{-m} \\ &= z^{-M/2}\left[h(M/2) + \sum_{m=0}^{M/2-1} h(m)(z^{-m+M/2} \pm z^{m-M/2})\right] \end{aligned} \tag{43}$$

and with M odd is

$$H(z) = z^{-M/2}\sum_{m=0}^{(M-1)/2} h(m)(z^{-m+M/2} \pm z^{m-M/2}) \tag{44}$$

where the $\pm$ signs in the above equations represent the + sign for symmetric impulse responses satisfying Eq. (42) and $-$ for antisymmetric impulse responses as in Eq. (43). The frequency responses corresponding to the above transfer functions with $z = e^{j\Omega}$ can be written in the form

$$H(e^{j\omega}) = e^{-j\omega M/2}R(\omega) \tag{45}$$

for symmetric impulse responses, and

$$H(e^{j\omega}) - je^{-j\omega M/2}R(\omega) \tag{46}$$

for antisymmetric impulse responses, with $R(\Omega)$ being a real-valued function of Ω. For M even, the corresponding group delay $M/2$ is an integer, and $R(\Omega)$ is given by

$$R(\omega) = h(M/2) + 2\sum_{m=0}^{M/2-1} h(m)\cos[\omega(m-M/2)] \tag{47}$$

for a symmetric impulse response, and

$$R(\omega) = 2\sum_{m=0}^{M/2-1} h(m)\sin[\omega(m-M/2)] \tag{48}$$

for an antisymmetric impulse response, where $h(M/2) = 0$ in the latter case. For M odd, $M/2$ is not an integer, resulting in a group delay that does not correspond to an integer number of sampling periods. $R(\Omega)$ is given by

$$R(\omega) = 2\sum_{m=0}^{(M-1)/2} h(m)\cos[\omega(m-M/2)] \tag{49}$$

for a symmetric impulse response, and

$$R(\omega) = 2\sum_{m=0}^{(M-1)/2} h(m)\sin[\omega(m-M/2)] \tag{50}$$

for an antisymmetric impulse response.

An FIR filter with antisymmetric impulse response presents a zero at $z = 1$ for M even or odd, and a zero at $z = -1$ for M even, as can be seen from Eqs. (44) and (45). Therefore, FIR filters with antisymmetric impulse responses cannot implement lowpass filters. Also, such filters cannot implement highpass filters when M is even. An FIR filter with symmetric impulse responses and M odd presents a zero at $z = -1$ and, therefore, cannot implement highpass filters. The other zeros of a linear-phase FIR filter $H(z)$ are such that if z_m is a zero of $H(z)$, so is $1/z^*_m$.

An FIR filter can be designed truncating the infinite impulse response of an ideal filter $h_{\text{ideal}}(n)$ through the multiplication of $h_{\text{ideal}}(n)$ by a finite length sequence $w(n)$ called

window. Other FIR filter design methods are based on optimization techniques, such as the Remez exchange algorithm, which minimizes the maximum deviation of the frequency response of the filter from a prescribed specification. Such design methods can be found elsewhere (1).

INFINITE IMPULSE RESPONSE FILTERS

Filters with infinite-length impulse responses are called *infinite impulse response* (IIR) filters. The output of an IIR filter is obtained by a recursive implementation, such as the direct-form structures shown in Figs. 4 and 5. The direct-form structures have high sensitivity to coefficient variations, especially when implementing transfer functions with poles clustered close to the unit circle.

Other IIR filter structures, which present lower sensitivity than the direct form, are based on the implementations of the filter transfer function $H(z)$ in a factored form. In the cascade structure, $H(z)$ is factored as a product of first- and/or second-order transfer functions, which are implemented separately by either one of the direct-form structures of Figs. 4 and 5 and connected in cascade. The parallel structure is based on the implementation of $H(z)$ when expressed as a sum of first- and/or second-order transfer functions, obtained by partial fraction expansion of $H(z)$. The wave and lattice realizations, both presenting low sensitivities, will be introduced in this article.

As opposed to FIR filters, IIR filters have poles in locations other than the origin of the z plane. To guarantee stability of an IIR filter, that is, that an input of bounded amplitude results in a bounded output sequence when processed by the filter, the poles of $H(z)$ must lie inside the unit circle in the z plane (1). The design of an IIR filter consists of finding the coefficients or the poles and zeros of the transfer function, such that the frequency response of the filter satisfies a given specification. Some IIR filter design techniques use established analog filter approximation methods with the application of a transformation technique to the analog transfer function or impulse response to obtain the digital transfer function. One of such transformations is the bilinear transformation, where the variable s in the transfer function of the analog filter is replaced by

$$s = \frac{1}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \tag{51}$$

Other design techniques for IIR filters are based on optimization methods, such as the quasi-Newton method described elsewhere (1).

Examples of first-order IIR low-pass and high-pass filters are, respectively,

$$H_{LP}(z) = k\,\frac{1 + z^{-1}}{1 - \alpha z^{-1}} \tag{1}$$

and

$$H_{HP}(z) = k\,\frac{1 - z^{-1}}{1 - \alpha z^{-1}} \tag{2}$$

with $|\alpha| < 1$ for stability. The constant k determines the filter gain, while the parameter α controls the passband width. Examples of second-order IIR bandpass and bandstop filters are, respectively,

$$H_{BP}(z) = k\,\frac{1 - z^{-2}}{1 - 2r\cos\theta z^{-1} + r^2 z^{-2}} \tag{3}$$

and

$$H_{BS}(z) = k\,\frac{1 - 2\cos\theta z^{-1} + z^{-2}}{1 - r^2 z^{-2}} \tag{4}$$

with $|r| < 1$ for stability. The parameters r and θ determine, respectively, the passband width and the center frequency of the passband/stopband.

Some special IIR filters are allpass, notch, and comb filters. The allpass filters are useful for phase equalization, and are characterized by having unity magnitude response for all frequencies, i.e.,

$$|H_{AP}(e^{j\omega})| = 1, \quad \text{for all}\,\omega \tag{5}$$

The transfer function of an N-th order allpass filter is of the form:

$$H_{AP}(z) = \pm\frac{\sum_{k=0}^{N} a_k z^{-N+k}}{\sum_{k=0}^{N} a_k z^{-k}} = \pm z^{-N}\,\frac{A(z^{-1})}{A(z)} \tag{6}$$

Observe that if $z_0 = re^{j\phi}$ is a pole of $H_{AP}(z)$, $z_0^{-1} = \frac{1}{r}\,e^{-j\phi}$ will be a zero of $H_{AP}(z)$.

The notch filters are bandstop filters with very narrow rejection band. They are useful in eliminating narrowband noise, such as the 60-Hz power-line interference. A typical second-order notch transfer function has zeros over the unity circle and poles close to it, with same angles $\pm\,\theta$, i.e.,

$$H_N(z) = \frac{1 - 2\cos\theta z^{-1} + z^{-2}}{1 - 2r\cos\theta z^{-1} + r^2 z^{-2}} \tag{7}$$

with $|r| < 1$. Figure 1 shows the frequency response of the notch filter of Eq. (7) with $r = 0.9$ and $\theta = \frac{\pi}{4}$.

Comb filters find application in pitch detection of speech signals and cancellation of periodic interferences, among others. Their frequency responses are periodic with period $\frac{2\pi}{M}$, where M is a positive integer. The transfer function of a comb filter can be obtained from a single passband transfer function $H(z)$ by substituting z^M for z, that is, $H_C(z) = H(z^M)$. For example, the high-pass filter with transfer function given by

$$H_{HP}(z) = 0.25\,\frac{1 - z^{-1}}{1 + 0.5z^{-1}} \tag{8}$$

and frequency response illustrated in Fig. 2(a), generates the 8-band comb filter with transfer function

$$H_C(z) = H_{HP}(z^8) = 0.25\,\frac{1 - z^{-8}}{1 + 0.5z^{-8}} \tag{9}$$

Its frequency response is shown in Fig. 2(b).

WAVE DIGITAL FILTERS

The designer of filters, regardless of the implementation technology, is usually interested in finding structures with low sensitivity to coefficient variations. In digital filter design the low sensitivity implies small effect on the overall transfer function when the values of the coefficients

deviate from their ideal values. As a result, the coefficients of a low-sensitivity digital filter can be implemented with short wordlengths without violating the prescribed specifications. Also, coefficients with short wordlengths are cheaper, faster, and simpler to implement. It is possible to show that low-sensitivity realizations usually generate low roundoff noise.

It is well known from classical analog circuit theory that doubly terminated lossless filters have zero sensitivities of the transfer function with respect to the lossless components at frequencies at which the maximal power is transferred to the filter load. For filter approximations with equiripple characteristics in the passband, such as Chebyshev and elliptic filters, there are several frequencies in which maximal power transfers to the load. Because the ripple values are small in the passband, the sensitivities remain small over the frequency range consisting of the passband. As a result, several methods have been proposed attempting to imitate the behavior of the doubly terminated lossless filters.

The simplest and most widely used method of transformation of a transfer function from the Laplace (s-domain) to the z-transform domain is the bilinear transformation [see Eq. (52)]. This transformation is the one used to establish the correspondence between the analog prototype and the wave digital filter. The bilinear transformation keeps a frequency domain correspondence between the analog and digital filters. The direct simulation of the internal quantities, such as voltages and currents, of the analog prototype in the digital domain leads to delay-free loops. A delay-free loop does not contain any delay, and as such cannot be computed sequentially since all node values in the loop are initially unknown (1). The values of the previous nodes must be known before we start computing any value in the loop. Alternative transformations can be tried; however, practice has shown that it is desirable to use the bilinear transformation. As a solution a linear combination of voltage and current are used in the transformation from continuous to the discrete-time domain.

It is well known that any analog n-port network can be characterized by using the concepts of incident and reflected waves quantities known from scattering parameter theory (5). Through the application of wave characterization, and the use of the bilinear transformation, digital filter realizations can be obtained from passive and active filters as first proposed by Fettweis (6, 7). By this means, analog filters can be converted in digital filter structures that are free from delay-free loops. The name wave digital filter derives from the fact that wave quantities are used to represent the internal analog circuit signals in the simulation in the digital domain. The possible wave quantities are voltage, current or power quantities. The choice of power waves leads to more complicated digital realizations, whereas the choice between voltage and current waves is irrelevant. Traditionally, voltage wave quantities are the choice in the open literature.

Another advantage of the wave digital filters imitating doubly terminated lossless filters is their inherent stability under linear conditions (i.e., infinite precision arithmetic) as well as in the nonlinear case where the signals are subjected to quantization. In the real-life nonlinear case, if magnitude truncation is applied to quantize suitable signals inside the wave digital filter structure, no zero-input limit cycles can be sustained. Also, as will be discussed in the section on quantization effects, the wave digital filters are free of overflow limit cycles when simple overflow nonlinearities are employed such as saturation arithmetic.

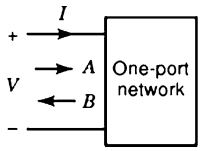

Figure 9. General representation of a generic one-port network.

The wave digital filters are also adequate to analog systems simulation, such as power systems, due to their topological equivalence with their analog counterparts.

An analog one-port network, see Fig. 9, can be described in terms of wave characterization as

$$\begin{aligned} a &= v + Ri \\ b &= v - Ri \end{aligned} \tag{52}$$

where a and b are the incident and reflected voltage wave quantities, respectively, and R is the port resistance assigned to the one-port network. The value of the port resistor is chosen appropriately to simplify the one-port realization. In the frequency domain the wave quantities are A and B which are given by

$$\begin{aligned} A &= V + RI \\ B &= V - RI \end{aligned} \tag{53}$$

In the equations above, we notice that the voltage waves consist of a linear combination of the voltage and current of the one-port network.

Consider now the case where the one-port impedance consists of a single element. That is, $Z(s) = cs^l$, where $l = 0$ for a resistor, $l = 1$ for an inductor, $l = -1$ for a capacitor, and c is a positive constant. Since

$$Z(s) = \frac{V(s)}{I(s)} \tag{54}$$

from Eq. (54), one can easily deduce the ratio between the reflected and incident voltage waves as follows:

$$\frac{B}{A} = \frac{Z(s) - R}{Z(s) + R} \tag{55}$$

By applying the bilinear transformation, that is, by substituting

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

the digital realization of the one-port network is obtained. However, the choice of the port resistance is crucial to obtain a simple realization, where in the present case the choice is

$$R = c\left(\frac{2}{T}\right)^l \tag{56}$$

For the capacitor, the port resistor is chosen as $R = T/2C$, where T is the sample period and C is the value of the capacitor, leading to a digital realization of the wave ratio

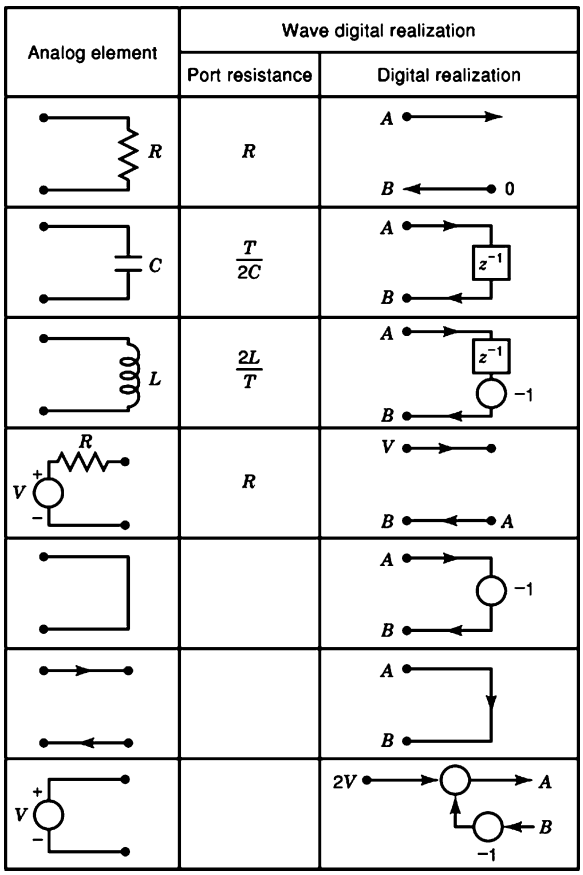

Figure 10. Wave digital realization of the main one port elements.

as $B/A = z^{-1}$. For the resistor the choice is $R = \mathscr{R}$, where $\mathscr{R}$ is the resistor value and the resulting digital realization is $B/A = 0$. Finally, the inductor is simulated by $B/A = -z^{-1}$ if $R = 2L/T$, where L is the inductor value. Figure 10 depicts the realization of some important one-port elements.

Similarly an analog N-port network can be described in terms of wave characterization as

$$\begin{aligned} A_i &= V_i + R_i I_i \\ B_i &= V_i - R_i I_i \end{aligned} \qquad (57)$$

for i = 1, . . . , N, where the parameters A_i and B_i are the incident and reflected voltage wave quantities, respectively, and R_i is the resistance of port i.

The main multiport elements required in the wave digital filter realization are the adaptors. The adaptors guarantee that the current and voltage Kirchoff laws are satisfied at the series and parallel interconnections of ports with different port resistances.

Consider the interconnection of two elements with port resistances given by R_1 and R_2, respectively, as shown in Fig. 11(a). The wave equations in this case are given by

$$\begin{aligned} A_1 &= V_1 + R_1 I_1 \\ A_2 &= V_2 + R_2 I_2 \\ B_1 &= V_1 - R_1 I_1 \\ B_2 &= V_2 - R_2 I_2 \end{aligned} \qquad (58)$$

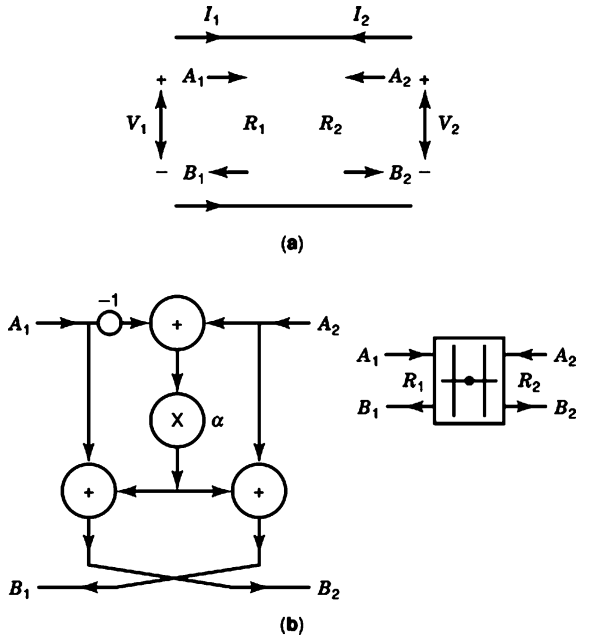

Figure 11. (a) Parallel connection of two ports. (b) Realization of two-port adaptor.

Because $V_1 = V_2$ and $I_1 = -I_2$, we have that

$$\begin{aligned} A_1 &= V_1 + R_1 I_1 \\ A_2 &= V_1 - R_2 I_1 \\ B_1 &= V_1 - R_1 I_1 \\ B_2 &= V_1 - R_2 I_1 \end{aligned} \qquad (59)$$

If we eliminate V_1 and I_1 in the above equations we have

$$\begin{aligned} B_1 &= A_2 + \alpha(A_2 - A_1) \\ B_2 &= A_1 + \alpha(A_2 - A_1) \end{aligned} \qquad (60)$$

where $\alpha = (R_1 - R_2)/(R_1 + R_2)$. A realization for the two-port adaptor is depicted in Fig. 11(b). It should be noted that there are other realizations for the two-port adaptor.

The same approach can be extended to derive the three-port series and parallel adaptors. The parallel interconnection of three ports is shown in Fig. 12(a), where we have $V_1 = V_2 = V_3$, $I_1 + I_2 + I_3 = 0$, and $G_i = 1/R_i$. The digital realization of the parallel adaptor has internal multipliers whose coefficients are given by

$$\alpha_i = \frac{2G_i}{G_1 + G_2 + G_3} \qquad (61)$$

Because $\alpha_1 + \alpha_2 + \alpha_3 = 2$, one of the required multiplier coefficients can be eliminated. With the definition above, after a few manipulations one can show that a possible set of relations between the incident and reflected waves is given by

$$\begin{aligned} B_3 &= A_3 + \alpha_1(A_1 - A_3) + \alpha_2(A_2 - A_3) \\ B_1 &= B_3 + A_3 - A_1 \\ B_2 &= B_3 + A_3 - A_2 \end{aligned} \qquad (62)$$

The realization corresponding to these equations is shown in Fig. 12(b).

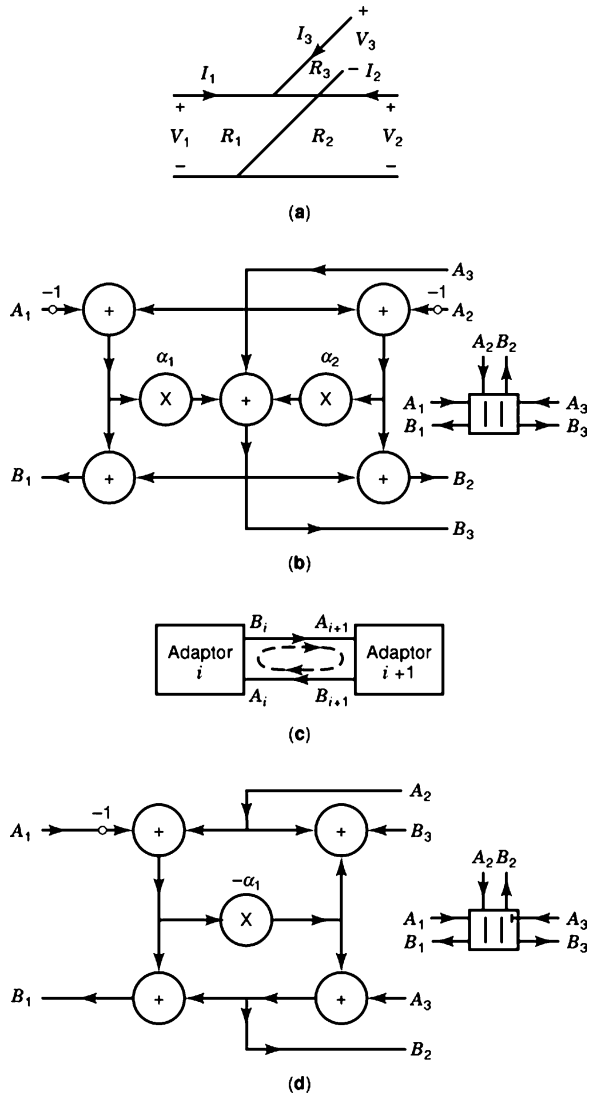

Figure 12. (a) Parallel connection of three ports. (b) A possible realization of a three-port parallel adaptor. (c) Interconnection between two adaptors. (d) Realization of a reflection-free three-port parallel adaptor $\alpha_2 = 1 - \alpha_1$.

If two adaptors are connected directly, a delay-free loop appears between the adaptors as shown in Fig. 12(c). A solution to this problem is to constrain one of the coefficients of the adaptor to be equal to one, for example $\alpha_3 = 1$. In this case, the adaptor equations are given by

$$\begin{aligned} G_3 &= G_1 + G_2 \\ B_3 &= \alpha_1 A_1 + \alpha_2 A_2 \\ B_2 &= \alpha_1 A_1 - \alpha_1 A_2 + A_3 \\ B_1 &= (1-\alpha_1)(A_1 - A_2) + A_3 \end{aligned} \tag{63}$$

where since $\alpha_1 + \alpha_2 = 1$, one of the required multiplier coefficients can also be eliminated. The realization of the reflection-free is depicted in Fig. 12(d), where it can be verified that there is no direct path between A_3 and B_3. As a consequence, the reflection-free property of port three is key to allow the connection between adaptors.

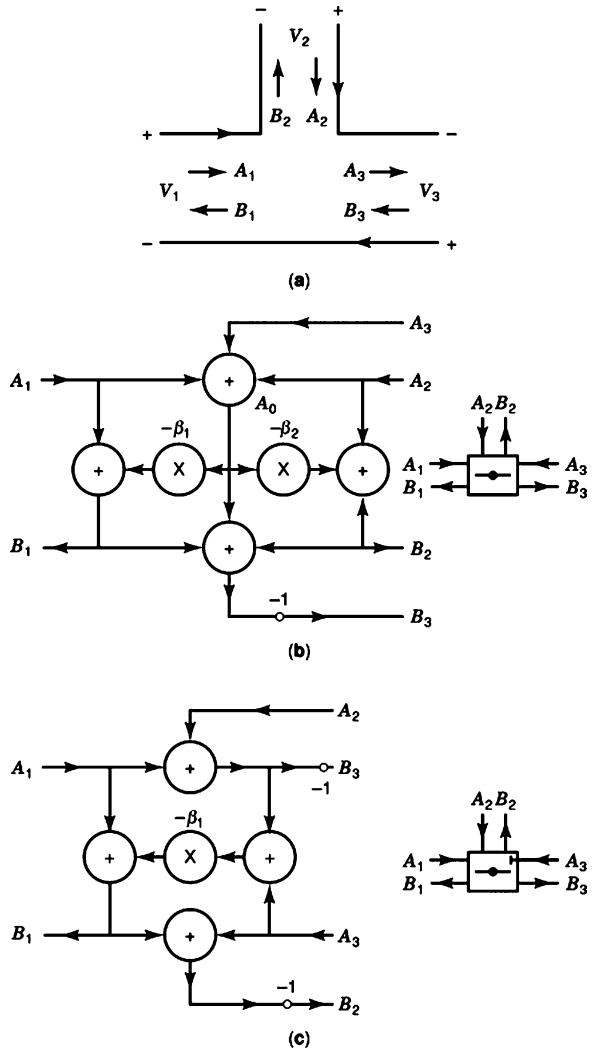

Figure 13. (a) Series connection of three ports. (b) A possible realization of a three-port parallel adaptor. (c) Realization of a reflection-free three-port series adaptor.

The series interconnection of three ports is shown in Fig. 13(a), where we have $V_1 + V_2 + V_3 = 0$ and $I_1 = I_2 = I_3$. The equations related to the series adaptor are derived by following the same procedure used for the parallel adaptor. The resulting equations are

$$\begin{aligned} B_1 &= A_1 - \beta_1(A_1 + A_2 + A_3) \\ B_2 &= A_2 - \beta_2(A_1 + A_2 + A_3) \\ B_3 &= -A_1 - A_2 - A_3 - B_1 - B_2 \end{aligned} \tag{64}$$

where

$$\beta_i = \frac{2R_k}{R_1 + R_2 + R_3}$$

for $i = 1, 2, 3$. The realization corresponding to these equations is shown in Fig. 13(b). The reflection-free series adaptor can be generated by considering $\beta_3 = 1$.

As an illustration, consider the third-order elliptic low-pass analog filter depicted in Fig. 14(a). The corresponding wave-digital realization is shown in Fig. 14(b), where the multiplier coefficients of the adaptors are calculated as fol-

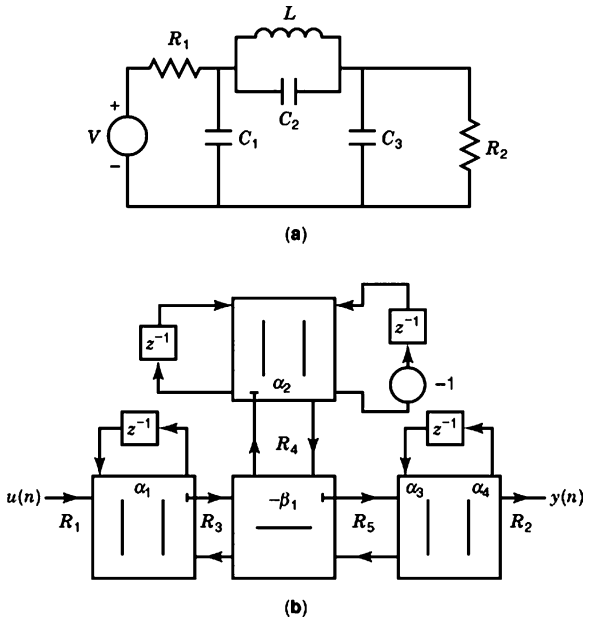

Figure 14. (a) LC doubly-terminated ladder; element values: $C_1 = C_3 = 0.968F, C_2 = 0.085_{\mathrm{F}}, L = 1.058H, R_1 = R_2 = 1\,\Omega$. (b) A possible wave digital realization. Notice that there are other choices for the position of the reflection-free ports. The sampling period is $T = \frac{1}{4}$ s.

lows:

$$\alpha_1 = \frac{2G_1}{G_1+G_2+G_3} = \frac{G_1}{G_1+G_2} = \frac{G_1}{G_1+\dfrac{2C_1}{T}} = 0.1144$$

because

$$G_3 = G_1 + G_2 = 7.7440$$

following similar procedure for the remaining elements we have

$$\begin{aligned}
\alpha_2 &= \frac{2C_2/T}{(2C_2/T)+(T/2L_1)} = 0.8514\\
R_4 &= 1.2520\\
\beta_1 &= \frac{R_3}{R_3+R_4} = 0.0935\\
R_5 &= R_3 + R_4 = 1.3811\\
\alpha_3 &= \frac{2G_5}{G_2+G_5+(2C_3/T)} = 0.1529\\
\alpha_4 &= \frac{2G_2}{(2C_3/T)+G_2+G_5} = 0.2112
\end{aligned}$$

LATTICE FILTERS

The general transfer function we aim to realize using the lattice structure is described by

$$H(z) = \frac{N_M(z)}{D_N(z)} = \frac{\sum_{i=0}^{M} b_{i,M} z^{-i}}{1+\sum_{i=1}^{N} a_{i,N} z^{-i}} \tag{65}$$

In the lattice construction, we first concentrate in the realization of the denominator polynomial through an order-reduction strategy. Define the polynomial

$$zB_N(z) = D_N(z^{-1})z^{-N} = z^{-N} + \sum_{i=1}^{N} a_{i,N} z^{i-N} \tag{66}$$

We calculate a reduced order polynomial as

$$\begin{aligned}
(1-a_{N,N}^2)D_{N-1}(z) &= D_N(z) - a_{N,N} zB_N(z)\\
&= (1-a_{N,N}^2) + \cdots + (a_{N,N} - a_{N,N})z^{-N}
\end{aligned} \tag{67}$$

where

$$D_{N-1}(z) = 1 + \sum_{i=1}^{N-1} a_{i,N-1} z^{-i}$$

Note that the first and last coefficients of $D_N(z)$ are 1 and $a_{N,N}$, whereas the first and last elements of the polynomial $zB_N(z)$ are $a_{N,N}$ and 1, respectively. This strategy to achieve the order reduction turns the polynomial $D_{N-1}(z)$ monic (i.e., with the leading coefficient equal to one).

By induction, this procedure can be repeated as described in the following equations:

$$\begin{aligned}
zB_j(z) &= D_j(z^{-1})z^{-N}\\
D_{j-1}(z) &= \frac{1}{1-a_{j,j}^2}[D_j(z) - a_{j,j} zB_j(z)]
\end{aligned} \tag{68}$$

for $j = N, N-1, \ldots, 0$, where $zB_0(z) = D_0(z) = 1$.

The pair of equations above leads to a convenient relation given by:

$$\begin{bmatrix} D_{j-1}(z) \\ B_j(z) \end{bmatrix} = \begin{bmatrix} 1 & -a_{j,j} \\ a_{j,j}z^{-1} & (1-a_{j,j}^2)z^{-1} \end{bmatrix} \begin{bmatrix} D_j(z) \\ B_{j-1}(z) \end{bmatrix} \tag{69}$$

Assuming that we can implement the desired denominator using the recursion above, it is required that we implement the numerator polynomial as well. The convenient way to form the desired numerator is to apply weights to the polynomials $zB_j(z)$ such that:

$$N_M(z) = \sum_{j=0}^{M} v_j zB_j(z) \tag{70}$$

where the tap-coefficients are calculated through the following order-reduction recursion:

$$\begin{aligned}
v_M &= b_{M,M}\\
N_{j-1}(z) &= N_j(z) - zv_j B_j(z)
\end{aligned} \tag{71}$$

for $j = M, M-1, \ldots, 1$, and $v_0 = b_{0,0}$.

The overall transfer function of the lattice structure is

$$H(z) = \frac{\sum_{j=0}^{M} v_j zB_j(z)}{D_N(z)} \tag{72}$$

The recursive lattice realization derives from Eqs. (75) and (78) in a simple way. Let us consider that the relations represented in Eq. (75) divided by $D_N(z)$ (due to the recursive structure) is implemented through a two-port network. Starting with $zB_0(z)/D_N(z) = D_0(z)/D_N(z) = 1/D_N(z)$, the structure of Fig. 15(a) results. Figure 15(b) depicts a realization for the two-port network.

There are some properties related to the lattice realization that are worth mentioning. If $D_N(z)$ has all the roots

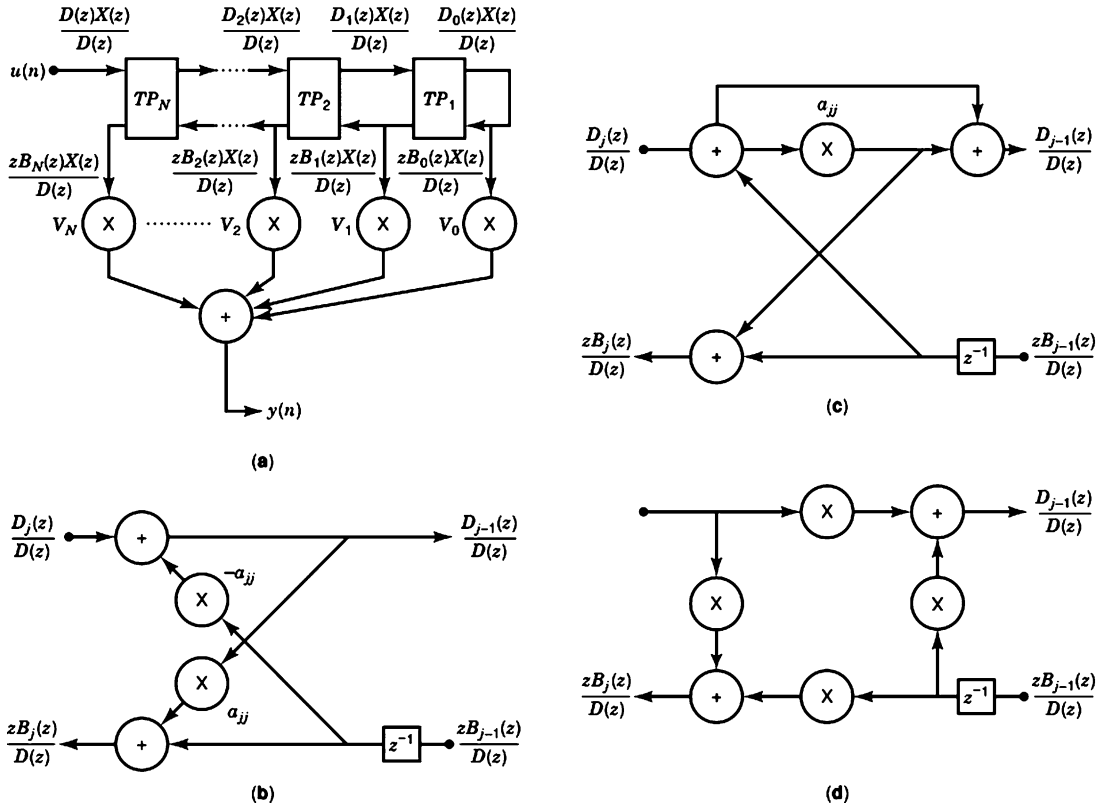

Figure 15. (a) General lattice digital filter structure. (b) The two-multiplier realization of the two-port network. (c) The single-multiplier realization of the two-port network. The plus and minus signs indicates that two different realizations are possible. The choice of these signs can vary from section to section aiming the reduction of the quantization noise at the filter output. (d) The normalized section for the lattice structure.

inside the unit circle the lattice structure will have all coefficients $a_{j,j}$ with magnitude less than one. Otherwise $H(z)$ represents an unstable system. The straightforward stability test turns the lattice realization useful to implement time-varying filters. Also in the lattice realization, the polynomials $zB_j(z)$ for $j = 0, \ldots, M$, form an orthogonal set; this feature justifies the choice of these polynomials to form the desired numerator polynomial $N_M(z)$.

The overall transfer function of the lattice realization will not change if any kind of internal scaling is applied to the internal signals in the following way:

$$\begin{aligned} \overline{D}_j(z) &= \overline{k}_j D_j(z) \\ \overline{B}_j(z) &= \overline{k}_j B_j(z) \end{aligned} \qquad (73)$$

where the numerator coefficients have to be scaled according to $\bar{v}_j = v_j/k_j$, with $\bar{k}_j = k_j k_{j-1} \cdots k_1$. Each coefficient k_i is the individual scaling factor applied to the lattice section i. With this possibility, we can derive a more economical two-port network using a single multiplier as shown in Fig. 15(c).

Another important realization for the two-port network results when the scaling parameters k_i are chosen such that the polynomials $z\bar{B}_j(z)$ become orthonormal. The appropriate scaling can be easily derived by induction if we recall that $zB_0(z) = A_0(z) = 1$. Since

$$zB_1(z) = z^{-1} + a_{1,1} \qquad (74)$$

a polynomial with unit norm results [i.e., $z\bar{B}_1(z)$] if we multiply $zB_1(z)$ by $k_1 = 1/\sqrt{1 - a_{1,1}^2}$. Identically, we can easily show that

$$zB_2(z) = z\overline{B}_1(z) + a_{2,2}\overline{D}_1(z) = z\overline{B}_1(z) + a_{2,2}z^{-2}\overline{B}_1(z^{-1}) \qquad (75)$$

Because $z\bar{B}_1(z)$ has unit norm, $zB_2(z)$ will have unit norm if we choose $k_2 = 1/\sqrt{1 - a_{2,2}^2}$. Following a similar procedure, we can show that the appropriate value for the scaling factor j is $k_j = 1/\sqrt{1 - a_{1,j}^j}$. After a few manipulations of Eqs. (76) and (80), we can show that the two-port section of the normalized lattice is as depicted in Fig. 15(d). The most important feature of the normalized lattice realization is that all its internal nodes have unit energy leading to an automatic scaling in the L_2 norm sense. This explains the low roundoff noise generated by the normalized lattice realization as compared with the other forms of the lattice realization.

QUANTIZATION EFFECTS IN DIGITAL FILTERS

The choice of a digital filter structure for a given application is based on evaluating the performance of known structures and choosing the most suitable one. The effects of quantization are important factors to be considered when assessing the performance of digital filter structures.

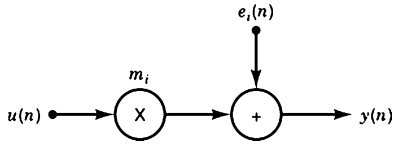

Figure 16. Model for the noise generated after a multiplication.

Quantization Noise

A number with modulus less than one can be represented in fixed-point arithmetic as follows:

$$x = b_0 b_1 b_2 b_3 \cdots b_b$$

where b_0 is the sign bit and $b_1 b_2 b_3 \cdots b_b$ represents the modulus of the number using a binary code. The most widely used binary code is the two's-complement representation where for positive numbers $b_0 = 0$ and for negative numbers $b_0 = 1$. The fractionary part of the number, called x_2 here, is represented as

$$x_2 = \begin{cases} x & \text{if } b_0 = 0 \\ 2 - |x| & \text{if } b_0 = 1 \end{cases}$$

In floating-point a number is represented as

$$x = x_m 2^c$$

where x_m is the mantissa and c is the number exponent, with $\frac{1}{2} \le |x_m| < 1$. In floating-point arithmetic, the mantissa must be quantized after every multiplication and addition, whereas in fixed-point arithmetic quantization is required only after multiplications. The main advantage of the floating-point representation is the large dynamic range, while fixed-point representations are easier to implement. Our discussion from now on concentrates in the fixed-point implementation.

A finite-wordlength multiplier can be modeled in terms of an ideal multiplier followed by a single noise source $e(n)$ as shown in Fig. 16. If the product quantization is performed by rounding and the signal levels throughout the filter are much larger than the quantization step $q = 2^{-b}$, it can be shown that the power spectral density of the noise source $e_i(n)$ is given by

$$P_{e_i}(z) = \frac{q^2}{12} = \frac{2^{-2b}}{12} \tag{76}$$

which means that $e_i(n)$ represents a zero mean white-noise process. Also, we can consider that in practice $e_i(n)$ and $e_j(n + l)$ are statistically independent for any value of n or l (for $i \neq j$). As a consequence the contributions of different noise sources can be accounted for separately using the principle of superposition.

In a fixed-point digital-filter implementation, the power spectral density of the output noise is given by

$$P_y(z) = \sigma_e^2 \sum_{i=1}^{K} G_i(z) G_i(z^{-1}) \tag{77}$$

where $P_e(e^{j\Omega}) = \sigma^2{}_e$, $G_i(z)$ are the transfer functions from each multiplier output $[g_i(n)]$ to the output of the filter as shown in Fig. 17. The wordlength, including sign, is $b + 1$ bits and K is the number of multipliers of the filter.

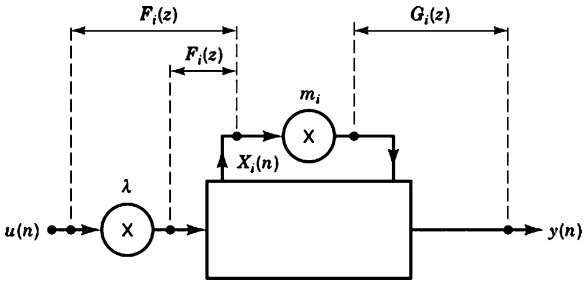

Figure 17. Digital filter including scaling and the relevant transfer functions for scaling, noise analysis and sensitivity calculation.

A figure of merit usually employed in evaluating the performance of digital filters is the relative power spectral density (RSPD) of the output noise in decibels given by

$$\text{RPSD} = 10 \log \frac{P_y(e^{j\omega})}{P_e(e^{j\omega})} \tag{78}$$

The RPSD eliminates the dependence of the output noise on the wordlength. Hence the RPSD is a measure of the extent to which the output noise depends upon the internal structure of the filter.

Another useful performance criterion to evaluate the roundoff-noise generated in digital filters is the noise gain or the relative noise variance (1, 2) given by

$$\begin{aligned} \frac{\sigma_y^2}{\sigma_e^2} &= \frac{1}{\pi} \int_0^{\pi} \sum_{i=1}^{K} |G_i(e^{j\omega})|^2 \, d\omega \\ &= \frac{1}{\pi} \sum_{i=1}^{K} \int_0^{\pi} |G_i(e^{j\omega})|^2 \, d\omega \\ &= \sum_{i=1}^{K} \|G_i(e^{j\omega})\|_2^2 \end{aligned} \tag{79}$$

where we used the relation

$$[G_i(z) G_i(z^{-1})]_{z=e^{j\omega}} = |G_i(e^{j\omega})|^2.$$

The input signal quantization is similar to product quantization and can be represented by including a noise source at the input of the digital filter structure.

Granularity Limit Cycles

On many occasions, signal levels in a digital filter can become constant or very low, at least for short periods of time. Under such circumstances, the noise signals become highly correlated from sample to sample and from source to source. This correlation can cause autonomous oscillations called granularity limit cycles.

Limit-cycles oscillations can occur in recursive digital filters implemented with rounding, magnitude truncation (where the magnitude of the number is reduced aiming the decrease of its energy), and other types of quantization. In many applications, the presence of limit cycles can be a serious problem. Thus, it is desirable to eliminate limit cycles or to keep their amplitude bounds low.

For wave and lattice digital filters and some second-order structures, the stability under finite precision arithmetic can be proved by means of the second method of Lyapunov. Magnitude truncation is applied to quantize suit-

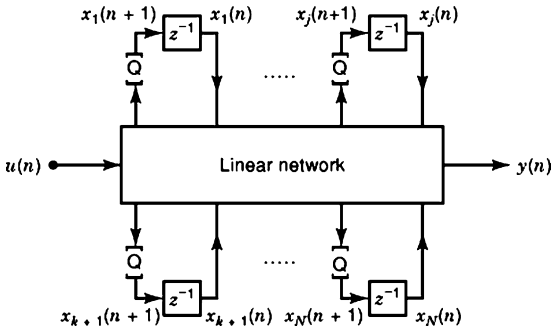

Figure 18. Digital filter including quantizers at the delay inputs.

able signals inside the structure such that a defined positive definite pseudoenergy function is proved to be a Lyapunov function.

The concept of pseudoenergy function can be applied to show how to eliminate zero-input limit cycles in some digital filter structures, for example, in wave, lattice, and state-space digital filter structures. The basic strategy is to apply magnitude truncation to the state variables of the digital filter, namely the delay inputs. An interesting result (11) establishes how constant-input limit cycles can also be eliminated in digital filters in which zero-input limit cycles can be eliminated.

In a recursive filter implemented with fixed-point arithmetic each internal loop contains a quantizer in order to keep the wordlength limited. Assuming that the quantizers are placed at the delay input (i.e., at the state variables as shown in Fig. 18), we can describe the digital filter, including the quantizers, using the state–state formulation as follows:

$$\begin{aligned} \mathbf{x}(n+1) &= [A\mathbf{x}(n) + \mathbf{b}u(n)]_Q \\ y(n) &= \mathbf{c}\mathbf{x}(n) + du(n) \end{aligned} \tag{80}$$

where $[\cdot]_Q$ indicates the quantized value of $[\cdot]$, A is the state matrix, $\mathbf{b}$ is the input vector, $\mathbf{c}$ is the output vector, and d represents the direct connection between the input and output of the filter.

Given that the digital filter has a state matrix with eigenvalues inside the unit circle such that

$$\hat{\mathbf{x}}^T(G - A^T G A)\hat{\mathbf{x}} \geq 0 \tag{81}$$

where G is an $N \times N$ diagonal positive definite matrix, and $\hat{u}$ is any $N \times 1$ vector. Then, the granular zero-input limit cycles can be eliminated if the quantization is performed through magnitude truncation. In this case, the quadratic energy function given by

$$p[\mathbf{x}(n)] = \mathbf{x}^T(n) G \mathbf{x}(n) \tag{82}$$

is used as Lyapunov function.

Overflow Limit Cycles

Overflow limit cycles can occur when the magnitude of the internal signals exceed the available register range. To avoid the increase of the signal wordlength in recursive digital filters, overflow nonlinearities must be applied to the signal. Overflow nonlinearities influence the most significant bits of the signal causing severe distortion. An overflow can give rise to self-sustained, high-amplitude oscillations known as overflow limit cycles.

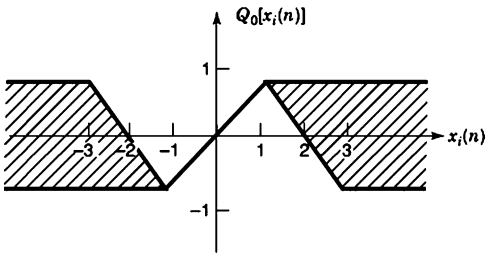

Figure 19. Regions allowed for the overflow nonlinearities in order to guarantee freedom from overflow limit cycles.

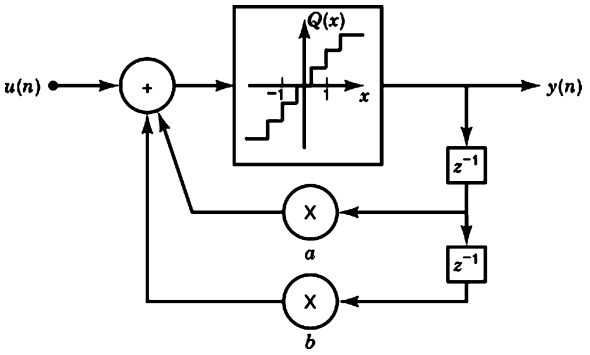

Figure 20. A second-order digital filter including a granularity and overflow quantizer.

A digital filter structure is considered as free of overflow limit cycles or to have a stable forced response if the error, which is introduced in the filter after an overflow, decreases with time in such a way that the output of the nonlinear filter (including the quantizers) converges to the output of the ideal linear filter (10).

A digital filter that is free of zero-input limit cycles, according to the condition of Eq. (92), is also forced-input stable if the overflow nonlinearities are in the shaded regions of Fig. 19. Figure 20 illustrates a digital filter realization incorporating a quantizer implementing rounding for the granular quantization and saturation arithmetic for the overflow nonlinearity.

In the presence of input signal, overflow can occur in any digital filter structure. As a consequence, input signal scaling is required to reduce the probability of overflow to an acceptable level. Ideally, signal scaling should be applied so as to ensure that the probability of overflow is the same at each internal node of the digital filter, to maximize the signal to noise ratio in fixed-point implementations.

The choice of two's complement arithmetic leads to a simplified scaling technique, where only the multiplier inputs require to be scaled. Specifically, in this type of number representation the addition of two or more numbers will be correct independently of the order in which they are added even if overflow occurs in a partial summation, as long as the overall sum is within the available range of representable numbers. In this scaling technique a scaling multiplier is used at the input of the filter section as illustrated in Fig. 17. We know that the signal at the multiplier

input is given by

$$x_i(n) = \frac{1}{2\pi j}\oint_c X_i(z)z^{(n-1)}\,dz = \frac{1}{2\pi}\int_0^{2\pi} F_i(e^{j\omega})U(e^{j\omega})e^{j\omega n}\,d\omega \tag{83}$$

where c is the convergence region common to $F_i(z)$ and $U(z)$.

The constant λ is usually chosen on the basis of the L_p norm of the transfer function from the filter input to the multiplier input $F_i(z)$, depending on the known properties of the input signal. The L_p norm of $F_i(z)$ is defined as

$$\|F_i(e^{j\omega})\|_p = \left[\frac{1}{2\pi}\int_0^{2\pi} |F_i(e^{j\omega})|^p\,d\omega\right]^{1/p} \tag{84}$$

for each $p \geq 1$, such that $\int^{2\pi}_0 |F_i(e^{j\Omega})|^p\,d\Omega \leq \infty$. In general the following expression is valid

$$|x_i(n)| \leq \|F_i\|_p \|U\|_q, \quad \left(\frac{1}{p}+\frac{1}{q}=1\right) \tag{85}$$

for $p, q = 1, 2$ and ∞.

The scaling ensures that the amplitudes of multiplier inputs are bounded by a number M when $|u(n)| \leq M$. Therefore, to ensure that all multiplier inputs are bounded by M, we must choose λ as follows

$$\lambda = \frac{1}{Max\{\|F_1\|_p, \ldots, \|F_i\|_p, \ldots, \|F_K\|_p\}} \tag{86}$$

which means that

$$\|F_i'(e^{j\omega})\|_p \leq 1, \text{ for } \|U(e^{j\omega})\|_q \leq U_{\max} \tag{87}$$

where K is the number of multipliers in the filter section.

The norm p is usually chosen to be infinity or 2. The L_∞ norm is used for input signals that have some dominating frequency component, whereas the L_2 norm is most commonly used for random input signal. Usually, the scaling coefficients are powers of two, provided they satisfy the overflow constraints. In this way, the scaling parameters can be implemented by simple shift operations.

In case of modular realizations such as cascade or parallel realizations of digital filters, optimum scaling is accomplished by applying one scaling multiplier per section.

Coefficient Quantization

During the approximation step the coefficients of a digital filter are calculated with high accuracy. If these coefficients are quantized, the frequency response of the realized digital filter will deviate from the ideal response. In fact, the quantized filter may even fail to meet the prescribed specifications. The sensitivity of the filter response to errors in the coefficients is highly dependent on the type of structure. This fact led to the development of low-sensitivity digital filter realizations such as the wave and lattice.

Several sensitivity criteria exist to evaluate the effect of the variation of a coefficient value on the digital filter transfer function. In this article, the sensitivity of the transfer function $H(z)$ with respect to variations in the multiplier constant m_i is defined as

$$S_{m_i}^{H(z)}(z) = \frac{\partial H(z)}{\partial m_i} \tag{88}$$

The variation in the magnitude response of the digital filter due to the variations in the multiplier coefficients are approximated by

$$\Delta|H(e^{j\omega})| \approx \sum_{i=1}^{K} \frac{\partial|H(e^{j\omega})|}{\partial m_i}\Delta m_i \tag{89}$$

where m_i, $i = 1, 2, \ldots, K$, are the multiplier coefficients of the digital filter. If we consider that the multiplier coefficients were rounded and that the errors introduced are statistically independent, the variance of the error in each coefficient is given by

$$\sigma^2_{\Delta m_i} = \frac{2^{-2b}}{12} \tag{90}$$

where b is the number of fractionary bits.

With the assumptions above the variance of $\Delta|H(e^{j\Omega})|$ is given by

$$\sigma^2_{\Delta|H(e^{j\omega})|} \approx \sigma^2_{\Delta m_i}\sum_{i=1}^{K}\left[\frac{\partial|H(e^{j\omega})|}{\partial m_i}\right]^2 - \sigma^2_{\Delta m_i}S^2(e^{j\omega}) \tag{91}$$

If we assume that $\Delta|H(e^{j\Omega})|$ has a Gaussian distribution, it is possible to estimate the probability of $\Delta|H(e^{j\Omega})|$ as less or equal to $x\sigma_{\Delta|H(e^{j\Omega})|}$. The equation below estimates the number of bits that are required in the fractionary part for a given digital filter to meet a given modulus specification.

$$2^{-b} = MAX_{0\leq\omega\leq\pi}\left|\frac{\sqrt{12}(\rho(\omega) - \|H(e^{j\omega})| - |H_d(e^{j\omega})\|)}{xS(e^{j\omega})}\right| \tag{92}$$

where $\rho(\Omega)$ is the tolerance on the magnitude response given in the specifications $H_d(e^{j\Omega})$.

Example:

An elliptic bandpass filter was designed satisfying the following prescribed specifications:

- Maximum ripple in the passband: 1 dB.
- Minimum attenuation in the stopband: 30 dB.
- Passband frequency range: 2880 to 3120 Hertz.
- Stopband frequency edges: 2450 and 3550 Hertz.
- Sampling frequency: 10000 Hertz.

The resulting filter has order four.

In order to access the coefficient quantization effects of a given realization, the fourth-order elliptic filter was implemented utilizing a normalized lattice structure. The coefficients of the lattice are displayed in Table 1. These coefficients are then quantized to 12, 8 and 6 bits, respectively. The resulting transfer functions are depicted in Figure 3. As can be observed the transfer functions for 8 and 6 bits deviate substantially from the desired one, whereas 12 bits leads to acceptable results.

A procedure widely used in practice to evaluate the design of digital filters with finite-coefficient wordlength is to design the filters with tighter specifications than required, quantize the coefficients, and check if the prescribed specifications are still met.

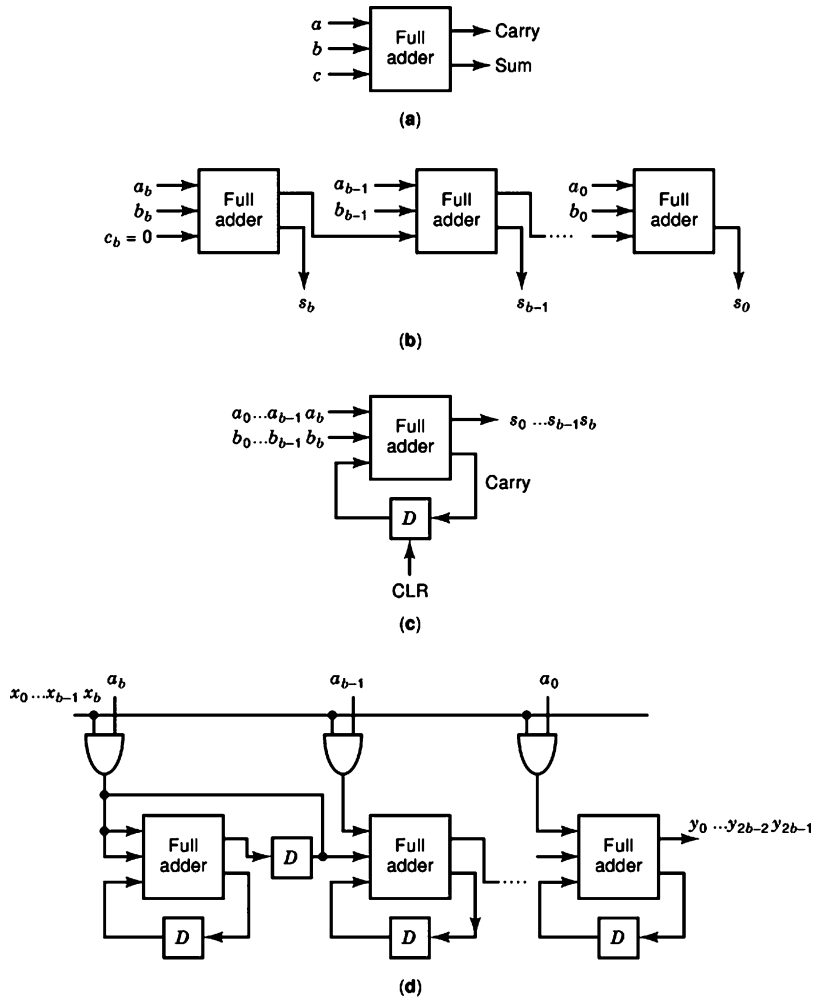

Figure 21. (a) Full adder; (b) bit-parallel; (c) bit-serial adder; (d) serial/parallel multiplier.

Table 1. Normalized Lattice Coefficients

	a_{jj}	v_j
0	−0.31382198601433	0.01451571512296
1	−0.98733578085783	0.01127246045032
2	−0.30596231306686	−0.01164209398292
3	−0.85033475836150	−0.00464776905230
4	–	0.03432034632233

DIGITAL FILTER IMPLEMENTATION

Digital filters can be implemented by software in general-purpose computers, in digital signal processor (*DSP*) chips, or by hardware in special-purpose logic circuits. Although software implementations allow rapid prototyping and flexibility in testing and modifying the filter characteristics, special-purpose hardware implementations allow for higher-speed and lower-consumption performances.

A software implementation consists of generating the program code corresponding to the digital filter structure being implemented, in a high-level language or directly in assembly language. A compiler then generates a set of instructions to the processor from the code. Because, in general-purpose computers the instructions are executed sequentially, the speed of the digital filter becomes limited by the execution time of each instruction. Digital signal processor chips are specially designed to execute very efficiently sum-of-product operations, which are the main computations required in the implementation of digital filters and of other digital processing algorithms, as can be seen in Figs. 4, 5, 6, 14, and 15. The efficient implementation of the multiply-and-accumulate operation, as well as the high-degree of parallelism with which the instructions are executed in a DSP, result in a relatively high input–output throughput rate.

A hardware implementation consists in the design and integration of a digital circuit, specified in terms of logical gates. Advances as well as reduction in costs of integrated circuit technologies have made special-purpose hardware implementations of digital filters even more attractive for high-speed real-time applications and/or for large production quantities. Besides providing higher-speed and lower-power consumption, hardware implementations using VLSI (very-large-scale integration) technologies permit to include in a single chip not only a digital filter but a whole signal processing system. Arithmetic operations required in the implementation of digital filters can be performed either in bit-serial or in bit-parallel form. The bit-parallel implementation of arithmetic operations uses

a basic element the full-adder shown in Fig. 21(a), which adds the two input-bits a and b, and the carry-in bit c, resulting in a sum bit and an output carry bit. The sum of two $(b + 1)$-bit numbers, A and B, in bit-parallel arithmetic can be implemented by connecting $(b + 1)$ full-adders, one for each bit, as shown in Fig. 21(b), where a_i and b_i represent the ith bit of A and B, respectively, with a_b and b_b corresponding to the least significant bits (*LSB*). A bit-parallel implementation for the product of two numbers A and B uses a full-adder for the partial sum of each bit product $a_i b_k$, requiring about $b(b + 1)$ full-adders. Reduction in chip area can be achieved by using bit-serial arithmetic. In such an implementation approach, the bits are processed one at each clock period, with the LSB treated first. A bit-serial adder is shown in Fig. 21(c), where D represents a one-bit delay (or a flip-flop) and CLR is set to zero during the processing of the LSB. A serial/parallel implementation of a multiplier (for $B > 0$) is shown in Fig. 21(d), where A is treated in bit-parallel form, whereas B is processed in bit-serial form. More details and other implementations of bit-serial, bit-parallel, and serial/parallel arithmetics can be found elsewhere (4). A different hardware implementation approach of digital filters, called distributed arithmetic, uses a look-up table to obtain partial results of the required sum-of-products. Such approach uses memory (corresponding to the look-up table) to replace most of the circuitry required to implement the computations.

The execution speed of the operations as well as the degree of parallelism associated to the digital filter implementation determine the maximum sampling rate for which the filter can operate. To increase this maximum rate, block processing algorithms have been proposed, where a block of output samples is calculated using a block of input samples. There is a delay in the production of the output samples in such algorithms, which might not be tolerable in some real-time applications. More details of block processing algorithms are given elsewhere (2, 3).

BIBLIOGRAPHY

1. P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto. *Digital Signal Processing, System Analysis and Design*. Cambridge, UK: Cambridge, 2002.
2. A. Antoniou. *Digital Signal Processing*. New York, NY: McGraw Hill, 2006.
3. S. K. Mitra. *Digital Signal Processing*. New York, NY: McGraw Hill, 3rd Edition, 2005.
4. L. Wanhammer. *DSP Integrated Circuits*. New York, NY: Academic Press, 1999.
5. V. Belevitch. *Classical Network Theory*. San Francisco, CA: Holden-Day, 1968.
6. A. Fettweis. Digital filters structures related to classical filters networks. *Archiv. fur Elektronik und Ubertragungstechnik*, **25**: 79–89, 1971.
7. A. Fettweis. Wave digital filters: theory and practice. *Proc. IEEE*, **74**: 270–327, 1986.
8. A. H. Gray, Jr. and J. D. Markel. Digital lattice and ladder filter synthesis. *IEEE Trans. Audio Electroacoust.*, **AU-21**: 491–500, 1973.
9. A. H. Gray, Jr. and J. D. Markel. A normalized digital filter structure. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-23**: 268–277, 1975.
10. T. A. C. M. Claasen, W. F. G. Mecklenbräuker, and J. B. H. Peek. On the stability of the forced response of digital filters with overflow nonlinearities. *IEEE Trans. Circuits Syst.*, **CAS-22**: 692–696, 1975.
11. P. S. R. Diniz, and A. Antoniou. More economical state-space digital filter structures which are free of constant-input limit cycles. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-34**: 807–815, 1986.

PAULO S. R. DINIZ
MARIANE R. PETRAGLIA
Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
Federal University of Rio de Janeiro, Rio de Janeiro, Brazil