FIR FILTERS, DESIGN

Filtering is a method of signal processing by which an input signal is passed through a system in order to be modified, reshaped, estimated, or generically manipulated as a way to make it conform to a prescribed specification. In a typical application, filters are a class of signal-processing systems that let some portion of the input signal's frequency spectrum pass through with little distortion and almost entirely cutting off the undesirable frequency band. Digital filtering is a method by which discrete time sequences are filtered by a discrete system.

Finite-duration impulse response (FIR) filters are a class of digital filters having a finite-length sequence as output when an impulse is applied to its input. The details of the FIR filtering method, its properties, and its applications will be considered in the following pages. Analytical tools that enable us to perform the time and frequency analysis of FIR filters will be examined and, in addition, systematic procedures for the design of these filters will be presented.

There are advantages and disadvantages in using FIR filters as opposed to infinite-duration impulse response (IIR) filters. FIR filters are always stable when realized nonrecursively, and they can be designed with exact linear phase, which is not possible with IIR filters. However, the approximation of sharp cutoff filters may require a lengthy FIR filter, which may cause problems in the realization or implementation of the filter.

There are many common applications of FIR filters, including

1. Differentiators: These filters have many uses in digital and analog systems, such as the demodulation of frequency modulated (FM) signals (1–6).
2. Decimators and interpolators: FIR filters appear in multirate systems as interpolator or decimator filters forming filter banks. Interpolators and decimators can be implemented, for instance, by comb filters, which are a special class of multiband filters, commonly used in the demodulation of video signals (1–6).
3. Power spectrum estimation: This process is used in digital speech, sonar, and radar systems. The moving average (MA) estimator is an example of an FIR filter (7).
4. Wiener and Kalman filters: These filters have been used for the estimation of signals of interest, and they both can be interpreted as extensions of FIR filters (8).
5. Adaptive filters: These systems have been widely used in communication, control, and so on. Examples of this type of system include adaptive antennas, digital equalization receivers, adaptive noise-canceling systems, and system modelers. Adaptive FIR filters are widely employed because the adaptation of the filter's coefficients searches for a unique optimal solution and does not cause instability in the transfer function, which is not always true for adaptive IIR filters (8–10).
6. Wavelets: The wavelet transform has been used as an orthogonal basis for decomposing signals in multiresolution layers. In image processing, it has been used for compressing the image data. In this setup, FIR filters have been normally employed as the basis function of the wavelet transform (11).

PROPERTIES OF FIR FILTERS

An FIR filter, with an input $x(n)$ and output $y(n)$, can be characterized by the following difference equation

$$\begin{aligned} y(n) &= a(0)x(n) + a(1)x(n-1) + \cdots + a(N-1)x(n-N+1) \\ &= \sum_{i=0}^{N-1} a(i)x(n-i) \end{aligned} \tag{1}$$

where N is the filter length, $a(n)$ (for $0 \le n \le N-1$) are the coefficients of the causal linear time-invariant system, and the input and output samples are taken at a discrete time $t = nT$, $n = 0, 1, \ldots$, with T as the sampling period. From Eq. (1), we notice that the present output sample $y(n)$ is formed by the combination of the present $x(n)$ and the past $N-1$ input samples weighted by the filter's coefficients in order attain some desired response.

Alternatively, Eq. (1) can be written as the convolution between the input signal samples $x(n)$ and the impulse response sequence $h(n)$ of the filter (1–6), that is,

$$\begin{aligned} y(n) &= \sum_{i=0}^{N-1} h(n-i)x(n) \\ &= \sum_{i=0}^{N-1} h(i)x(n-i) \end{aligned} \tag{2}$$

Clearly, then, $h(n)$ equals $a(n)$, for $0 \le n \le N-1$.

Stability

A digital system is said to be bounded input–bounded output (BIBO) stable if and only if any bounded discrete time sequence applied to the system's input yields a bounded output sequence. Therefore, the BIBO stability criterion can be written as (1–6)

$$\text{If } |x(n)| < \infty, \forall n, \text{then } |y(n)| < \infty, \forall n$$

For an FIR filter as expressed in Eq. (2), we obtain

$$|y(n)| = \left| \sum_{i=0}^{N-1} h(i)x(n-i) \right|$$

By using the well known Schwartz's inequality, we get

$$\begin{aligned} |y(n)| &\leq \sum_{i=0}^{N-1} |h(i)x(n-i)| \\ &\leq \sum_{i=0}^{N-1} |h(i)||x(n-i)| \end{aligned}$$

As the BIBO criterion imposes,

$$|x(n)| \leq M < \infty, \forall n$$

we thus have

$$|y(n)| \leq M \sum_{i=0}^{N-1} |h(i)| \tag{3}$$

Since the filter's coefficients are assumed finite, the right-hand side of Eq. (3) is always finite and so is $|y(n)|$. This implies that an FIR filter is always stable since the linear system coefficients are finite.

Alternatively, we can test filter stability by identifying its poles. The stability of any digital filter can also be verified by checking the poles of its transfer function. A necessary and sufficient condition for BIBO filter stability is that all system poles are inside the unit circle (1–6). In the case of an FIR filter, its transfer function is obtained as

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \tag{4}$$

which can be written as

$$\begin{aligned} H(z) &= h(0) + h(1)z^{-1} + \cdots + h(N-1)z^{-(N-1)} \\ &= \frac{h(0)z^{N-1} + h(1)z^{N-2} + \cdots + h(N-1)}{z^{N-1}} \end{aligned} \tag{5}$$

From Eq. (5), we see that an FIR filter of length N has $N - 1$ poles, all located at the origin of the z-plane, and therefore, it is always stable. As will be seen later, an FIR filter can be realized nonrecursively and recursively. The stability of an FIR filter is guaranteed only when the realization is nonrecursive since the quantization of filter coefficients in a recursive realization with finite precision arithmetic may cause instability.

In some applications, like speech, image coding, and signal transmission, it is possible to use FIR filters with complex coefficients to process complex signals. Despite these cases, we concentrate our efforts here on the analysis and design of FIR filters with real coefficients, without much loss of generality. The interested reader, however, may refer to (12,13).

Linear Phase

In many applications, the design of a signal processing system with linear phase is desirable. Nonlinear phase causes distortion in the processed signal, which is very perceptible in applications like data transmission, image processing, and so on. One of the major advantages of using an FIR filter is that it can be designed with an exact linear phase, a task that cannot be done with IIR filters. A linear-phase system does not cause any distortion, only delay.

The frequency response of an FIR filter, as described in Eq. (4), is given by

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n)e^{-j\omega n} \tag{6}$$

The magnitude $M(\omega)$ and phase $\theta(\omega)$ responses of the filter are respectively defined as

$$M(\omega) = |H(e^{j\omega})|$$

and

$$\begin{aligned} \theta(\omega) &= \arg[H(e^{j\omega})] \\ &= \tan^{-1}\left\{\frac{\text{Im}[H(e^{j\omega})]}{\text{Re}[H(e^{j\omega})]}\right\} \end{aligned}$$

where Im(.) and Re(.) represent the imaginary and real parts of a complex number, respectively. The expression in Eq. (6) can also be written as

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n)[\cos(\omega n) - j\sin(\omega n)]$$

Now, by assuming that $h(n)$ is a sequence of real numbers, the phase response is given by

$$\theta(\omega) = \tan^{-1}\left[\frac{-\sum_{n=0}^{N-1} h(n)\sin(\omega n)}{\sum_{n=0}^{N-1} h(n)\cos(\omega n)}\right] \tag{7}$$

To obtain a linear-phase response, $\theta(\omega)$ is constrained to be of the form

$$\theta(\omega) = -\omega\tau_0 \tag{8}$$

for $-\pi \leq \omega \leq \pi$, where τ_0 is the constant delay of the filter. When using the results of Eq. (7) and Eq. (8), we get

$$\tan^{-1}\left[\frac{-\sum_{n=0}^{N-1} h(n)\sin(\omega n)}{\sum_{n=0}^{N-1} h(n)\cos(\omega n)}\right] = -\omega\tau_0$$

which can be written as

$$\tan(\omega\tau_0) = \frac{\sum_{n=0}^{N-1} h(n)\sin(\omega n)}{\sum_{n=0}^{N-1} h(n)\cos(\omega n)} \tag{9}$$

Eq. (9) admits two solutions. The first is a trivial one when $\tau_0 = 0$, which implies $h(n) = \delta(n)$; that is, the filter's impulse response is an impulse at $n = 0$. This solution has very little utility. The second solution is when $\tau_0 \neq 0$, and thus, Eq. (9) can be expressed as

$$\frac{\sin(\omega\tau_0)}{\cos(\omega\tau_0)} = \frac{\sum_{n=0}^{N-1} h(n)\sin(\omega n)}{\sum_{n=0}^{N-1} h(n)\cos(\omega n)}$$

Consequently,

$$\sum_{n=0}^{N-1} h(n)[\sin(\omega n)\cos(\omega\tau_0) - \cos(\omega n)\sin(\omega\tau_0) = 0$$

and accordingly,

$$\sum_{n=0}^{N-1} h(n)\sin(\omega n - \omega\tau_0) = 0 \qquad (10)$$

The solution of Eq. (10) can be shown to be the following set of conditions

$$\tau_0 = \frac{N-1}{2} \qquad (11a)$$

$$h(n) = \pm h(N-n-1) \qquad (11b)$$

It turns out that different solutions are obtained depending on the value of N being either even or odd, and depending on the two possibilities as expressed by Eq. (11b), that is, symmetrical (even symmetry) or antisymmetrical (odd symmetry) filters. Therefore, from the set of conditions defined by Eq. (11), it is practical to define four types of linear-phase FIR filters, namely:

Type I. Length N even and symmetrical impulse response
Type II. Length N even and antisymmetrical impulse response
Type III. Length N odd and symmetrical impulse response
Type IV. Length N odd and antisymmetrical impulse response

Examples of these four types of linear-phase FIR filters are depicted in Fig. 1. When N is even, see Fig. 1(a) and Fig. 1(b), we notice that the axis of symmetry is located between two samples; that is, the constant delay τ_0 is not an integer value. Meanwhile, if N is odd, τ_0 is integer, and thus, the location of the axis of symmetry is over a sample, as observed in Fig. 1(c) and Fig. 1(d). For N odd and an antisymmetrical impulse response filter, the middle sample must be zero to satisfy this symmetry, as seen in Fig. 1(d).

In some applications, the long delay associated with linear-phase FIR filters is not allowed and then a nonlinear-phase filter is required. An example of such filter is the filter with minimum-phase distortion, the zeros of which are located strictly inside the Z-domain unit circle. This class of filters can be designed by starting from Type I and III (even symmetric) linear-phase filters as mentioned, for instance, in (12,14,15).

Frequency Response

The four types of FIR linear phase filters defined before have distinct frequency responses, as we shall see below.

Type I. From Eq. (6) and having that $h(n) = h(N-n-1)$, we can write the frequency response as

$$H(e^{j\omega}) = \sum_{n=0}^{N/2-1} h(n)[e^{-j\omega n} + e^{-j\omega(N-n-1)}]$$

which can be written in the form

$$\begin{aligned} H(e^{j\omega}) &= \sum_{n=0}^{N/2-1} h(n)e^{-j\omega\left(\frac{N-1}{2}\right)}\left[e^{j\omega\left(\frac{N-1-2n}{2}\right)} + e^{-j\omega\left(\frac{N-1-2n}{2}\right)}\right] \\ &= e^{-j\omega\left(\frac{N-1}{2}\right)} \sum_{n=0}^{N/2-1} 2h(n)\cos\left[\omega\left(\frac{N-1-2n}{2}\right)\right] \end{aligned}$$

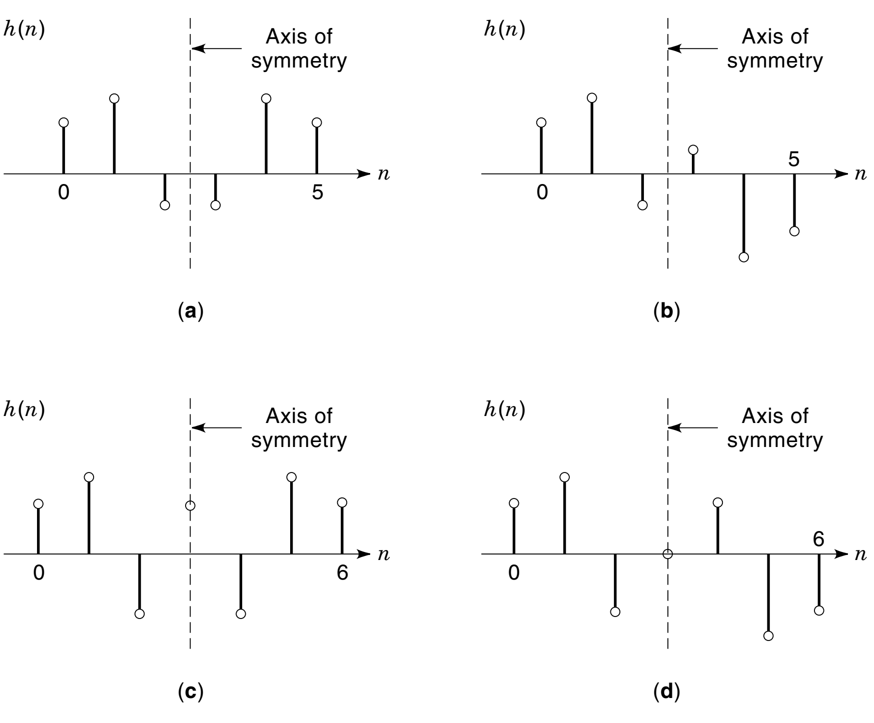

Figure 1. Typical impulse responses for linear-phase FIR filters: (a) Type I: N even, symmetric filter; (b) Type II: N even, antisymmetric filter; (c) Type III: N odd, symmetric filter; (d) Type IV: N odd, antisymmetric filter.

Finally, letting

$$a(n) = 2h \approx \left(\frac{N}{2} - n\right)$$

for $n = 1, \ldots, N/2$, the last summation in the above equation can be written as

$$H(e^{j\omega}) = e^{-j\omega\left(\frac{N-1}{2}\right)} \sum_{n=1}^{N-2} a(n) \cos\left[\omega\left(n - \frac{1}{2}\right)\right] \qquad (12)$$

which is the desired result, having a pure delay term and an even-symmetric amplitude term.

Type II. For this case, the frequency response is similar to Type I above, except that in Eq. (12) instead of cosine summations we have sine summations multiplied by j or, equivalently, multiplied by $e^{j(\pi/2)}$. Hence, Eq. (12) should be replaced by

$$H(e^{j\omega}) = e^{-j\omega\left(\frac{N-1}{2}\right)} e^{j(\pi/2)} \sum_{n=1}^{N/2} a(n) \sin\left[\omega\left(n - \frac{1}{2}\right)\right] \qquad (13)$$

Type III. By applying the even symmetry condition to Eq. (6) with N odd, we obtain

$$\begin{aligned} H(e^{j\omega}) &= h \approx \left(\frac{N-1}{2}\right) e^{-j\omega\left(\frac{N-1}{2}\right)} + \sum_{n=0}^{(N-3)/2} h(n) e^{-j\omega n} \\ &= h \approx \left(\frac{N-1}{2}\right) e^{-j\omega\left(\frac{N-1}{2}\right)} \\ &\quad + e^{-j\omega\left(\frac{N-1}{2}\right)} \sum_{n=0}^{(N-3)/2} h(n) e^{-j\omega n} e^{j\omega\left(\frac{N-1}{2}\right)} \end{aligned}$$

The above equation can be written as

$$\begin{aligned} H(e^{j\omega}) = e^{-j\omega\left(\frac{N-1}{2}\right)} \Bigg\{ & h \approx \left(\frac{N-1}{2}\right) \\ & + \sum_{n=0}^{(N-3)/2} h(n)\left[e^{j\omega\left(\frac{N-1}{2}-n\right)} + e^{-j\omega\left(\frac{N-1}{2}-n\right)}\right]\Bigg\} \end{aligned}$$

where $h(N - 1/2)$ is the middle sample of the filter's impulse response. The above equation can also be expressed as

$$\begin{aligned} H(e^{j\omega}) = e^{-j\omega\left(\frac{N-1}{2}\right)} \Bigg\{ & h \approx \left(\frac{N-1}{2}\right) \\ & + \sum_{n=0}^{(N-3)/2} 2h(n) \cos\left[\omega\left(\frac{N-1}{2} - n\right)\right]\Bigg\} \end{aligned}$$

Now, when replacing the variable n by $(N - 1)/2 - n$, we obtain

$$\begin{aligned} H(e^{j\omega}) = e^{-j\omega\left(\frac{N-1}{2}\right)} \Bigg[& h \approx \left(\frac{N-1}{2}\right) \\ & + \sum_{n=1}^{(N-1)/2} 2h \approx \left(\frac{N-1}{2} - n\right) \cos(\omega n)\Bigg] \end{aligned}$$

Letting

$$b(n) = 2h \approx \left(\frac{N-1}{2} - n\right)$$

for $n = 1, \ldots, (N - 1)/2$, and $b(0) = h(N - 1/2)$, we get

$$H(e^{j\omega}) = e^{-j\omega\left(\frac{N-1}{2}\right)} \sum_{n=0}^{(N-1)/2} b(n) \cos(\omega n) \qquad (14)$$

which is the desired result.

Type IV. For this case, the frequency response is similar to Type III, except that in Eq. (14) instead of cosine summations, we have, as before, sine summations multiplied by $e^{j\pi/2}$; that is,

$$H(e^{j\omega}) = e^{-j\omega\left(\frac{N-1}{2}\right)} e^{j\frac{\pi}{2}} \sum_{n=0}^{(N-1)/2} b(n) \sin(\omega n) \qquad (15)$$

In this specific case, $b(0) = h(N - 1/2) = 0$.

In Table 1, we summarize the properties of the four types of linear-phase FIR filters, as given in Eq. (12–15).

Locations of Zeros

The locations of zeros in the Z-plane for a linear-phase FIR filter is highly restricted by the set of conditions defined by Eq. (11). When N is even, by applying these conditions to the transfer function in Eq. (4), we obtain

$$\begin{aligned} H(z) &= \sum_{n=0}^{N/2-1} h(n)\left[z^{-n} \pm z^{-(N-n-1)}\right] \\ &= \sum_{n=0}^{N/2-1} h(n) z^{-\left(\frac{N-1}{2}\right)}\left[z^{\left(\frac{N-1-2n}{2}\right)} \pm z^{-\left(\frac{N-1-2n}{2}\right)}\right] \\ &= \frac{\sum_{n=0}^{N/2-1} h(n)\left[z^{\left(\frac{N-1-2n}{2}\right)} \pm z^{-\left(\frac{N-1-2n}{2}\right)}\right]}{z^{\left(\frac{N-1}{2}\right)}} \end{aligned} \qquad (16)$$

where the positive sign applies for symmetrical filters and the negative sign for antisymmetrical filters. Now, by examining the numerator of Eq. (16), that is, the zeros of the transfer function, we see that if we replace z by z^{-1}, we obtain the same or the negative numerator, respectively, for symmetrical and antisymmetrical filters. In both cases, the positions of the zeros are similar. Therefore, if the transfer function has a zero at point $z = \rho\, e^{j\theta}$, it will also have a zero at point $z = (1/\rho)e^{-j\theta}$, where ρ is the magnitude, and θ is the phase of the mentioned zero. In such a case, the numerator of $H(z)$ is said to have the mirror-image zero property with respect to the unit circle.

Similarly, for N odd we obtain the following transfer function

$$H(z) = \frac{h\left(\frac{N-1}{2}\right) + \sum_{n=0}^{(N-3)/2} h(n)\left[z^{\left(\frac{N-1-2n}{2}\right)} \pm z^{-\left(\frac{N-1-2n}{2}\right)}\right]}{z^{\left(\frac{N-1}{2}\right)}} \qquad (17)$$

where, as before, the positive sign applies for symmetrical filters and the negative sign for antisymmetrical filters. Also, in this case, we see that the numerator of $H(z)$ has the mirror-image zero property.

Hence, the locations of the zeros of all four types of linear-phase FIR filters have the mirror-image common property. This implies that the locations of zeros have the following possibilities:

Table 1. Characteristics of Linear-Phase FIR Filters

Type	N	$h(n)$	$H(e^{j\omega})$	Coefficients
I	even	symmetrical	$e^{-j\omega(N-1/2)}\sum_{n=1}^{N/2} a(n)\cos\left[\omega\left(n-\frac{1}{2}\right)\right]$	$\begin{cases} a(n) = 2h \approx \left(\frac{N}{2} - n\right) \\ n = 1, \ldots, N/2 \end{cases}$
II	even	antisymmetrical	$e^{-j\omega(N-1/2)}e^{J(\pi/2)}\sum_{n=1}^{N/2} a(n)\sin\left[\omega\left(n-\frac{1}{2}\right)\right]$	$\begin{cases} a(n) = 2h \approx \left(\frac{N}{2} - n\right) \\ n = 1, \ldots, N/2 \end{cases}$
III	odd	symmetrical	$e^{-j\omega(N-1/2)}\sum_{n=0}^{(N-1)/2} b(n)\cos(\omega n)$	$\begin{cases} b(n) = 2h \approx \left(\frac{N-1}{2} - n\right) \\ n = 1, \ldots, (N-1)/2 \\ b(0) = h \approx \left(\frac{\mathrm{N}-1}{2}\right) \end{cases}$
IV	odd	antisymmetrical	$e^{-j\omega(N-1/2)}e^{j(\pi/2)}\sum_{n=0}^{(N-1)/2} b(n)\sin(\omega n)$	$\begin{cases} b(n) = 2h \approx \left(\frac{N-1}{2} - n\right) \\ n = 1, \ldots, (N-1)/2 \\ b(0) = h \approx \left(\frac{N-1}{2}\right) = 0 \end{cases}$

1. Complex zeros located off the unit circle appear as a set of four conjugate reciprocal zeros of the form

$$z_{11} = \rho e^{j\theta},\ z_{12} = \rho e^{-j\theta},\ z_{13} = \frac{1}{\rho}e^{j\theta},\ z_{14} = \frac{1}{\rho}e^{-j\theta}$$

2. Complex zeros located on the unit circle appear as a set of conjugate pairs of the form

$$z_{21} = e^{j\theta},\ z_{22} = e^{-j\theta}$$

3. Real zeros off the unit circle appear as a set real pairs of the form

$$z_{31} = \rho,\ z_{32} = \frac{1}{\rho}$$

or

$$z_{31} = -\rho,\ z_{32} = -\frac{1}{\rho}$$

4. Real zeros on the unit circle appear in an arbitrary number

$$z_{41} = 1$$

or

$$z_{42} = -1$$

The locations of zeros at points $z = \pm 1$ have additional importance. By examining the transfer function at these points, it turns out that

5. A type I linear-phase FIR filter must have a zero at $z = -1$. Hence, high-pass filters cannot be designed with this type of filter.
6. A type II linear-phase FIR filter must have a zero at $z = 1$. Thus, low-pass filters cannot be designed with this type of filter.
7. A type IV linear-phase FIR filter must have zeros at both $z = 1$ and $z = -1$. Therefore, either low-pass or high-pass filters cannot be designed with this type of filter.

Figure 2 depicts a typical plot of the zeros of a linear-phase FIR filter.

FIR FILTER DESIGN

The complete design of FIR filters involves three distinct stages, namely approximation, realization, and implementa-

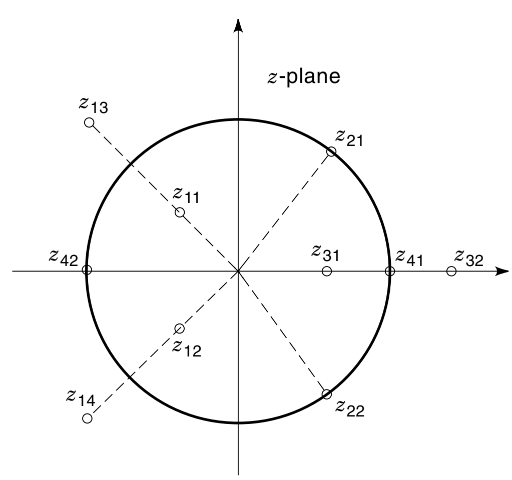

Figure 2. Example of the locations of zeros of a linear-phase FIR filter: In this case, z_{11}, z_{12}, z_{13}, and z_{14} are four complex-valued zeros that satisfy rule 1; the pair of complex zeros on the unit circle z_{21} and z_{22} obey rule 2; the pair of real zeros z_{31} and z_{32} satisfy rule 3; real zeros on the unit circle, like z_{41} and z_{42}, may appear in any number, obeying rule 4.

tion. Approximation is the process by which a required set of filter specifications yields a suitable transfer function with the desired filter characteristics. The realization process translates the transfer function obtained in the approximation stage into a digital network. Implementation is the process that transforms this digital network into a specific piece of hardware or software code. Because they are easier to understand this way, we now analyze these tasks in their reverse order.

Implementation

An FIR filter is a digital system, the implementation of which can be accomplished by means of dedicated hardware, general purpose hardware (e.g., digital signal processors), or computer programs. Both hardware and software forms are suitable for processing real-time and nonreal-time signals. Dedicated-hardware filters, however, tend to be much faster, thus being able to process signals with higher frequency components and/or using high-order filters. The manufacturing stage of these filters, however, can be a very cost- and time-expensive process. In the other extreme, computer programs are mainly used, although not restricted, to filter signals in a nonreal-time fashion or to simulate the performance of practical systems. Generally speaking, filters implemented with digital signal processors (DSPs) represent a good compromise of cost and processing speed compared to the other two alternatives.

Currently, the two most well-known families of DSPs are the TMS320 from Texas Instruments and the DSP56000 from Motorola. Both families have DSPs that use fixed- or floating-point arithmetic, parallel processing, clock-rates from tens to hundreds of MHz, and cost in the range of a few dollars to several hundred dollars. Today, DSPs are becoming increasingly cheaper and faster, and this process shows no indication of slowing down. For this reason, one can only predict this type of implementation for FIR digital filters becoming more and more popular in the future.

Realization

For any given transfer function, there is a wide variety of network structures that are able to translate the mathematical aspect of the transfer function into an equivalent digital circuit. In this section, we show some of the most commonly used forms for realizing an FIR transfer function (1–6).

Direct Form. As given in Eq. (2), the transfer function of FIR filters with length N assumes the form

$$H(z) = \sum_{n=0}^{N-1} h_n z^{-n} \tag{18}$$

where h_n is the filter coefficient, corresponding to the filter's impulse response $h(n)$, for $0 \leq n \leq N-1$. The change in notation, in this section, from $h(n)$ to h_n, is important to avoid confusion with adaptive filters and to yield a more natural representation of other realizations. The most natural way to perform the operations in Eq. (18) is probably the direct form shown in Fig. 3. This figure clearly depicts how, in this realization, each delayed value of the input signal is appropriately weighted by the corresponding coefficient h_n and how the resulting products are added to compose the output signal $y(n)$. Due to the chain of delay elements on the top of that diagram, this structure is also referred to as the tapped delay line or transversal filter.

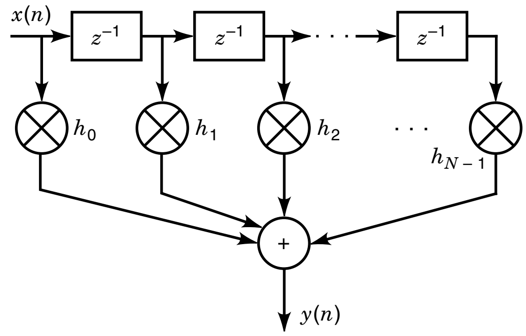

Figure 3. Direct-form realization of an FIR filter. Its derivation is straightforward following Eq. (18).

Linear-Phase Direct Form. As seen before, FIR filters with linear phase present a symmetrical or antisymmetrical transfer function. This fact can be used to halve the number of multiplications needed to realize the filter. In fact, the transfer function of linear-phase FIR filters with length N even can be written as

$$H(z) = \sum_{n=0}^{N/2-1} h_n\left[z^{-n} \pm z^{-(N-n-1)}\right]$$

leading to the structure shown in Fig. 4.

Meanwhile, the transfer function of linear-phase FIR filters with length N odd can be written as

$$H(z) = \sum_{n=0}^{(N-3)/2} h_n\left[z^{-n} \pm z^{-(N-n-1)}\right] + h \approx \left(\frac{N-1}{2}\right) z^{\left(\frac{N-1}{2}\right)}$$

and is suitable to be realized by the structure shown in Fig. 5. In both of these figures, the plus sign is associated to the

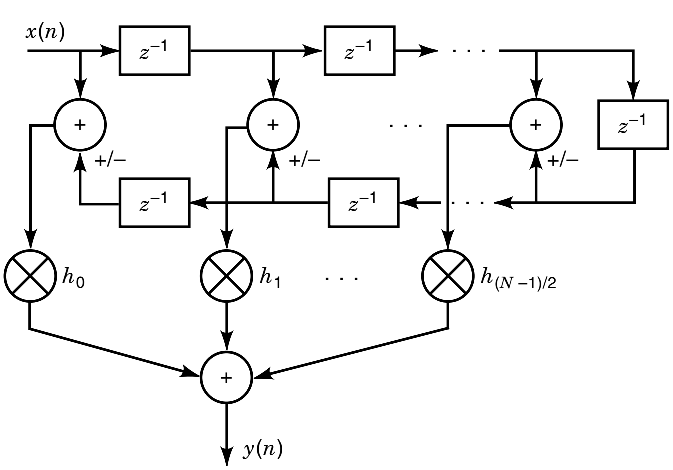

Figure 4. Direct-form realization of a linear-phase FIR filter with N even: Type I and Type II filters. The reader should verify reduction on the number of multiplications in the order of 50% when compared to the general direct-form realization seen in Fig. 3.

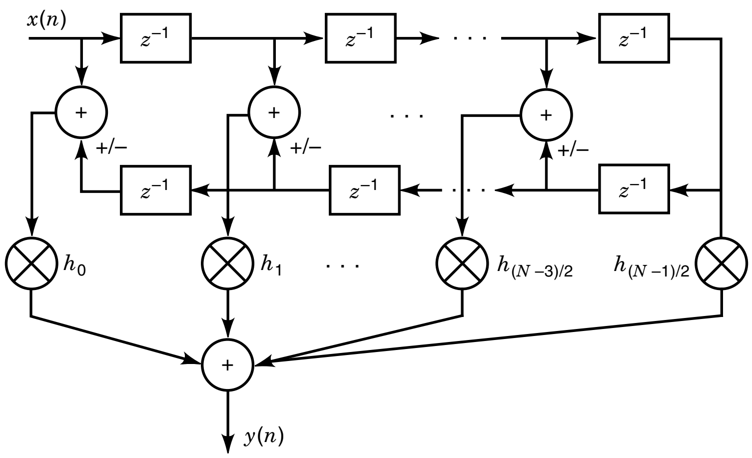

Figure 5. Direct-form realization of a linear-phase FIR filter with N odd: Type III and Type IV filters. The reader should verify reduction on the number of multiplications in the order of 50% when compared to the general direct-form realization seen in Fig. 3.

symmetrical impulse response, and the minus sign is associated to the antisymmetrical case, as included in Table 1. It is important to notice that the linear-phase direct forms shown here preserve this important characteristic even when the filter coefficients are quantized, that is, are represented with a finite number of bits.

Cascade Form. Any FIR-filter transfer function can be factored into a product of second-order polynomials with real coefficients; that is,

$$H(z) = b_0 \prod_{j=1}^{M} (1 + b_{1j} z^{-1} + b_{2j} z^{-2}) \tag{19}$$

where M is the smallest integer greater or equal to $(N - 1)/2$. If N is even, then the coefficient b_{2M} is equal to zero. The block diagram representing Eq. (19) is shown in Fig. 6. Notice how the second-order blocks appear in sequence, thus originating the name of the cascade-form realization.

Lattice Form. Figure 7 depicts the block diagram of an FIR lattice filter of length N, where $e(m)$ and $\tilde{e}(m)$ are auxiliary signals that appear in lattice-type structures. This realization is called lattice due to its highly regular structure formed by concatenating basic blocks of the form shown in Fig. 8.

To obtain a useful relation between the lattice parameters and the filter's impulse response, we must analyze the recurrent relationships that appear in Fig. 8. These equations are

$$e_i(n) = e_{i-1}(n) + k_i \tilde{e}_{i-1}(n-1) \tag{20a}$$

$$\tilde{e}_i(n) = \tilde{e}_{i-1}(n-1) + k_i e_{i-1}(n) \tag{20b}$$

for $i = 1, \ldots, N - 1$, with $e_0(n) = \tilde{e}_0(n) = k_0 x(n)$, and $e_{N-1}(n) = y(n)$. In the z domain, Eq. (20) becomes

$$E_i(n) = E_{i-1}(z) + k_i z^{-1} \tilde{E}_{i-1}(z)$$

$$\tilde{E}_i(z) = z^{-1} \tilde{E}_{i-1}(z) + k_i E_{i-1}(z)$$

with $E_0(z) = \tilde{E}_0(z) = k_0 X(z)$ and $E_{N-1}(n) = Y(z)$.

By defining the auxiliary polynomials, $H_i(z)$ and $\tilde{H}_i(z)$, and the auxiliary coefficients $h_{m,i}$ as:

$$H_i(z) = k_0 \frac{E_i(z)}{E_0(z)} = \sum_{m=0}^{i} h_{m,i} z^{-m}$$

$$\tilde{H}_i(z) = k_0 \frac{\tilde{E}_i(z)}{\tilde{E}_0(z)}$$

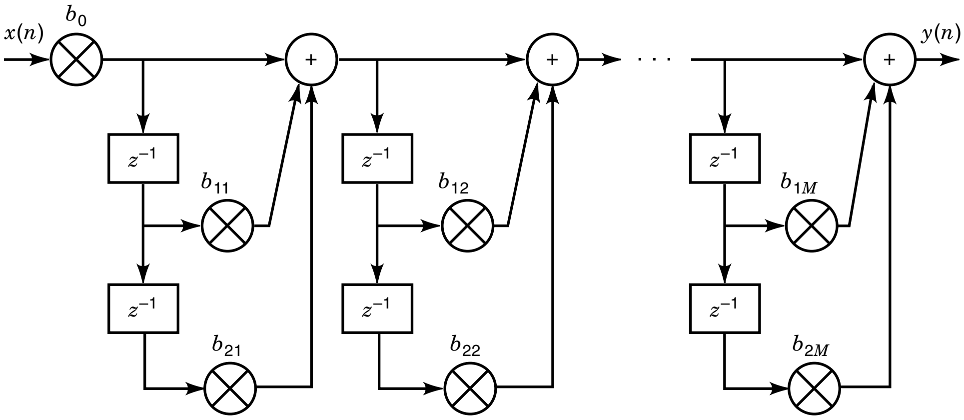

Figure 6. Cascade-form realization of an FIR filter. Its derivation is straightforward following Eq. (19).

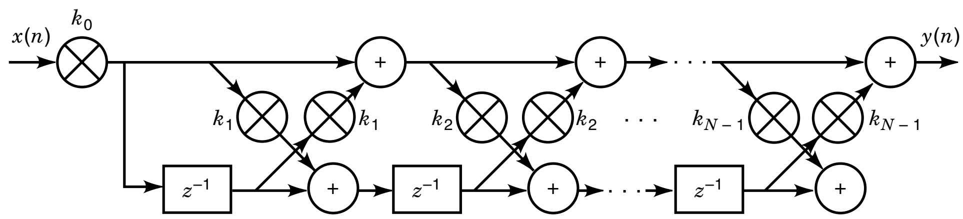

Figure 7. Lattice-form realization of a FIR filter. Its name results from the intricate structure of each building block implementing Eq. (20).

we can demonstrate, by induction, that these polynomials obey the recurrence formulas (3)

$$H_i(z) = H_{i-1}(z) + k_i z^{-i} H_{i-1}(z^{-1}) \tag{21a}$$

$$\tilde{H}_i(z) = z^{-i} H_i(z^{-i}) \tag{21b}$$

with $H_0(z) = \tilde{H}_0(z) = k_0$ and $H_{N-1}(z) = H(z)$. Therefore,

$$H_{i-1}(z) = \frac{1}{1-k_i^2}[H_i(z) - k_i z^{-i} H_i(z^{-1})] \tag{22}$$

and then, the reflection coefficients k_i can be obtained from the direct-form coefficients h_i by successively determining the polynomials $H_{i-1}(z)$ from $H_i(z)$, using Eq. (22), and making

$$k_i = h_{i,i}$$

for $i = N - 1, \ldots, 0$.

To determine the filter's impulse response from the set of lattice coefficients k_i, we use Eq. (21a) to determine the auxiliary polynomials $H_i(z)$ and make

$$h_i = h_{i,N}$$

for $i = 0, \ldots, N - 1$.

The direct, cascade, and lattice structures differ from each other with respect to a few implementation aspects. In general, the direct form is used when perfect linear-phase is essential. Meanwhile, the cascade and lattice forms present better transfer-function sensitivities with respect to coefficient quantization, but their dynamic range may be an issue, as their states can reach very high levels, forcing signal scaling in fixed-point implementations (3,4).

Recursive Form. FIR filters are often associated to nonrecursive structures, as the ones previously seen here. However, there are some recursive structures that do possess an FIR characteristic. Consider, for instance, the moving average-filter of length N

$$H(z) = \frac{1}{N}\sum_{n=0}^{N-1} z^{-n}$$

As its name indicates, this filter determines, for each n, the average value of N consecutive samples of a given signal. Adding all N samples of the input signal at each time instant n, however, can be a very time-consuming procedure. Fortunately, the same computation can be performed recursively using the previous sum if we subtract the past input sample delayed of N cycles and add the present input sample. This procedure is equivalent to rewriting the moving-average transfer function as

$$H(z) = \frac{1}{N}\frac{z^{-N}-1}{z^{-1}-1} \tag{23}$$

The realization associated to this filter is seen in Fig. 9, where the recursive nature of its transfer function is easily observed. Other FIR filters can also be realized with recursive structures that make use of some form of zero/pole cancellation, as exemplified here. This procedure, however, is somewhat problematic in practice, as the quantization of the filter coefficients or of the filter internal signals can lead to a nonexact cancellation, which can cause filter instability.

Frequency-Domain Form. The computation of the output signal of an FIR filter in the direct form is performed as

$$y(n) = \sum_{i=0}^{N-1} h_i x(n-i)$$

If the input signal $x(n)$ is known for all n, and null for $n < 0$ and $n > L$, a different approach to compute $y(n)$ can be derived based on the discrete Fourier transform, which is usually implemented through an efficient algorithm commonly referred to as the fast Fourier transform (FFT) (16). Complet-

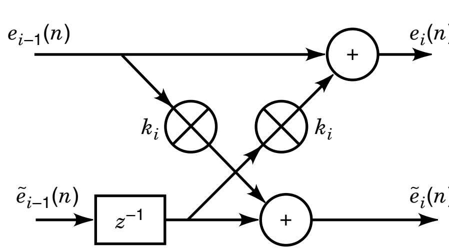

Figure 8. Basic block for the lattice realization in Fig. 7.

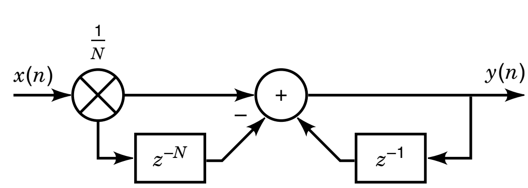

Figure 9. Recursive-form realization of a moving-average filter. The reader should be able to identify the feedback loop with the z^{-1} block that originates the transfer-function denominator term in Eq. (23).

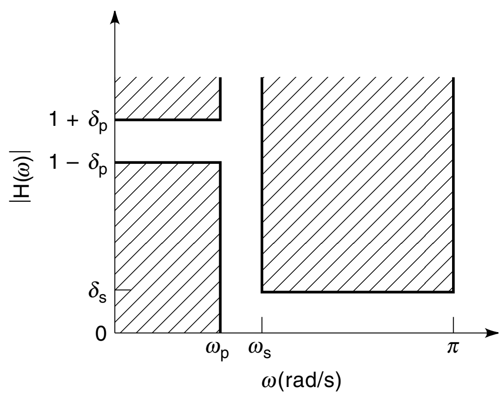

Figure 10. A typical set of specifications for a low-pass filter includes definition of the maximum passband ripple δ_p, the minimum stopband attenuation δ_s, as well as the passband and stopband edges ω_p and ω_s, respectively.

ing these sequences with the necessary number of zeros (zero-padding procedure) and determining the resulting $(N + L)$-element FFTs of h_n, $x(n)$, and $y(n)$, designated here as $H(k)$, $X(k)$, and $Y(k)$, respectively, we then have

$$Y(k) = H(k)X(k)$$

and then,

$$y(n) = \text{FFT}^{-1}\{\text{FFT}h_n]\text{FFT}[x(n)]\}$$

Using this approach, we are able to compute the entire sequence $y(n)$ with a number of arithmetic operations proportional to $\log_2(L + N)$, per output sample, as opposed to NL, as in the case of direct evaluation. Clearly, for large values of N and L, the FFT method is the more efficient one.

In the above approach, the entire input sequence must be available to allow one to compute the output signal. In this case, if the input is extremely long, the complete computation of $y(n)$ can result in a long input-output delay, which is objectionable in several applications. For such cases, the input signal can be sectioned, and each data block processed separately using the so-called overlap-and-save and overlap-and-add methods, as described in (3,4,16).

Approximation

As mentioned before, the approximation process searches for the transfer function that best fits a complete set of specifications determined by the application in hand. There are two major forms of solving the approximation problem: using closed-form methods or using numerical methods. Closed-form approaches are very efficient and lead to very straightforward designs. Their main disadvantage, however, is that they are useful only for the design of filters with piecewise-constant amplitude responses. Numerical methods are based on iterative optimization methods, and, therefore, can be very computationally cumbersome. Nevertheless, numerical methods often yield superior results when compared to closed-form methods, besides being useful also for designing FIR filters with arbitrary amplitude and phase responses.

A description of a low-pass filter is represented in Fig. 10, where δ_p is the passband maximum ripple, δ_s is the stopband minimum attenuation, and ω_p and ω_s are the passband and stopband edges, respectively. Based on these values, we define

$$DB_p = 20\log_{10}\left(\frac{1+\delta_p}{1-\delta_p}\right)\text{dB} \tag{23a}$$

$$DB_s = 20\log_{10}(\delta_s)\,\text{dB} \tag{23b}$$

$$B_t = (\omega_s - \omega_p)\,\text{rad/s} \tag{23c}$$

Basically, DB_p and DB_s are the passband maximum ripple, δ_p, and the stopband minimum attenuation S_p, expressed in decibel (dB), respectively. Also, B_t is the width of the transition band, where no specification is provided.

We now analyze the methods most used to convert this typical set of specifications into a realizable transfer function.

Closed-Form Methods: The Kaiser Window. The most important class of closed-form methods to approximate a given frequency response using FIR filters is the one based on window functions.

A discrete-time frequency response is a periodic function of ω, and, thus, can be expressed as a Fourier series given by

$$\tilde{H}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n)e^{-j\omega n}$$

where

$$h(n) = \frac{1}{2\pi}\int_{-\pi/2}^{\pi/2}\tilde{H}(e^{j\omega})e^{j\omega n}\,d\omega$$

By making the variable transformation $z = e^{j\omega}$, we have

$$\tilde{H}(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$$

Unfortunately, however, this function is noncausal and of infinite length. These problems can be solved, for instance, by truncating the series symmetrically for $|n| \le (N-1)/2$, with N odd, and by multiplying the resulting function by $z^{-(N-1)/2}$, yielding

$$\tilde{H}(z) \approx H(z) = \sum_{n=-(N-1)/2}^{(N-1)/2} h(n)z^{-n-\left(\frac{N-1}{2}\right)} \tag{24}$$

This approach, however, results in ripples known as Gibbs' oscillations appearing near transition bands of the desired frequency response. An easy-to-use technique to reduce these oscillations is to precondition the resulting impulse response $h(n)$ with a class of functions collectively known as window functions. There are several members of this family of functions, including the rectangular window, the Hamming window, the Hanning (von Hann) window, the Blackman window, the Adams window, the Dolph–Chebyshev window, and the Kaiser window. The rectangular window is essentially the approximation introduced in Eq. (24). Due to its importance, we concentrate our exposition here solely on the Kaiser window. Explanation of the other window functions can be found in (1–6,17).

The most important feature of a given window function is to control the transition bandwidth and the ratio between the ripples in the passband and stopband in an independent manner. The Kaiser window allows that control and is defined as (4):

$$w(n) = \begin{cases} \dfrac{I_0(\beta)}{I_0(\alpha)} & \text{for } |n| \le (N-1)/2 \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

where α is an independent parameter, β is given by

$$\beta = \alpha\sqrt{1 - \left(\frac{2n}{N-1}\right)^2}$$

and $I_0(.)$ is the zeroth-order modified Bessel function of the first kind.

The ripple ratio resulting from the Kaiser window can be adjusted continuously from the low value in the Blackman window to the high value of the rectangular window by simply varying the parameter α. In addition, the transition bandwidth can be varied with the filter length N. The most important property of the Kaiser window is that empirical formulas are available relating the parameters α and N to any specific values of ripple ratio and transition bandwidth. In that manner, given the definitions in Eq. (24), a filter satisfying these specifications can be readily designed based on the Kaiser window as (4,6):

1. Determine $h(n)$ using the Fourier series, assuming an ideal frequency response

$$\tilde{H}(e^{j\omega}) = \begin{cases} 1 & \text{for } |\omega| \le \omega_c \\ 0 & \text{for } \omega_c \le |\omega| \le \pi \end{cases}$$

with $\omega_c = (\omega_p + \omega_s)/2$

2. Choose δ as the minimum of δ_p and δ_s.
3. Compute DB_p and DB_s with that value of δ in Eq. (24a) and Eq. (24b), respectively.
4. Choose the parameter α as follows:

$$\alpha = \begin{cases} 0 & \text{for } DB_s \le 21 \\ 0.5842(DB_s - 21)^{0.4} + 0.07886(DB_s - 21) & \\ & \text{for } 21 < DB_s \le 50 \\ 0.1102(DB_s - 8.7) & \text{for } DB_s > 50 \end{cases}$$

5. Choose the value of D as follows:

$$D = \begin{cases} 0.9222 & \text{for } DB_s \le 21 \\ \dfrac{DB_s - 7.95}{14.36} & \text{for } DB_s > 21 \end{cases}$$

and then select the lowest odd value of the filter length N such that

$$N \ge \frac{\omega_s D}{B_t} + 1$$

6. Determine $w(n)$ using Eq. (25).
7. Finally, compute

$$H(z) = z^{-\left(\frac{N-1}{2}\right)} \sum_{n=-(N-1)/2}^{(N-1)/2} [w(n)h(n)]z^{-n}$$

High-pass, bandpass, or bandstop filters are designed in a very similar manner. A few variables, however, must be redefined, as it is summarized in Table 2. For bandpass and bandstop filters, ω_{p1} and ω_{p2} are the passband edges with $\omega_{p1} < \omega_{p2}$, and ω_{s1} and ω_{s2} are the stopband edges with $\omega_{s1} < \omega_{s2}$.

Numerical Methods. Numerical methods are often able to approximate the required frequency response using lower order filter than their closed-form counterparts. The design of FIR filters using numerical methods is dominated in the literature by the Chebyshev and the weighted-least-squares (WLS) approaches. The Chebyshev scheme minimizes the maximum absolute value of a weighted error function between the prototype's transfer function and a given ideal solution. For that reason, Chebyshev filters are also said to satisfy a minimax criterion. The universal availability of minimax computer routines, has motivated their widespread use. The WLS approach, which minimizes the sum of the squares of the weighted error function, is characterized by a very simple implementation. Its basic problem, however, results from the well-known Gibbs phenomenon which corresponds to large error near discontinuities of the desired response.

To understand the basic problem formulation of the numerical methods for approximating FIR filters, consider the transfer function associated to a linear-phase filter of length N

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n}$$

and assume that N is odd, and $h(n)$ is symmetrical. Other cases of N even or $h(n)$ antisymmetrical can be dealt with in a very similar way and are not further discussed here. The frequency response of such filter is then given by

$$H(e^{j\omega}) = e^{-j\omega\frac{(N-1)}{2}}\hat{H}(\omega)$$

where

$$\hat{H}(\omega) = \sum_{n=0}^{\tau_0} a(n)\cos(\omega n) \tag{26}$$

with $\tau_0 = (N-1)/2$, $a(0) = h(\tau_0)$, and $a(n) = 2h(\tau_0 - n)$, for $n = 1, \ldots, \tau_0$.

If $e^{-j\omega\tau_0}\tilde{H}(\omega)$ is the desired frequency response, and $W(\omega)$ is a strictly-positive weighting function, consider the weighted error function $E(\omega)$ defined in the frequency domain as

$$E(\omega) = W(\omega)[\tilde{H}(\omega) - \hat{H}(\omega)] \tag{27}$$

The approximation problem for linear-phase nonrecursive digital filters resumes to the minimization of some objective function of $E(\omega)$ in such a way that

$$|E(\omega)| \le \delta$$

Table 2. Filter Definitions to Use With the Kaiser Window

Type	B_t	ω_c	$H(e^{j\omega})$
low-pass	$\omega_p - \omega_s$	$\omega_c = \dfrac{\omega_p + \omega_s}{2}$	$\begin{cases} 1, & \text{for } \lvert\omega\rvert \le \omega_c \\ 0, & \text{for } \omega_c \le \lvert\omega\rvert \le \pi \end{cases}$
high-pass	$\omega_s - \omega_p$	$\omega_c = \dfrac{\omega_p + \omega_s}{2}$	$\begin{cases} 0, & \text{for } \lvert\omega\rvert \le \omega_c \\ 1, & \text{for } \omega_c \le \lvert\omega\rvert \le \pi \end{cases}$
bandpass	$\min[(\omega_{p1} - \omega_{s1}), (\omega_{s2} - \omega_{p2})]$	$\omega_{c1} = \omega_{p1} - \dfrac{B_t}{2}$ $\omega_{c2} = \omega_{p2} - \dfrac{B_t}{2}$	$\begin{cases} 0, & \text{for } \lvert\omega\rvert \le \omega_{c1} \\ 1, & \text{for } \omega_{c1} \le \lvert\omega\rvert \le \omega_{c2} \\ 0, & \text{for } \lvert\omega\rvert \le \pi \end{cases}$
bandstop	$\min[(\omega_{s1} - \omega_{p1}), (\omega_{p2} - \omega_{s2})]$	$\omega_{c1} = \omega_{p1} - \dfrac{B_t}{2}$ $\omega_{c2} = \omega_{p2} - \dfrac{B_t}{2}$	$\begin{cases} 1, & \text{for } \lvert\omega\rvert \le \omega_{c1} \\ 0, & \text{for } \omega_{c1} \le \lvert\omega\rvert \le \omega_{c2} \\ 1, & \text{for } \lvert\omega\rvert \le \pi \end{cases}$

and then,

$$|H(\omega) - \hat{H}(\omega)| \le \frac{\delta}{W(\omega)}$$

By evaluating the error function defined in Eq. (27), with $\hat{H}(\omega)$ as in Eq. (26), on a dense frequency grid with $0 \le \omega_i \le \pi$, for $i = 1, \ldots, MN$, a good discrete approximation of $E(\omega)$ can be obtained. For practical purposes, for a filter of length N, using $8 \le M \le 16$ is suggested. Points associated to the transition band are disregarded, and the remaining frequencies should be linearly redistributed in the passband and stopband to include their corresponding edges. Thus, the following vector equation results

$$\boldsymbol{E} = W(\boldsymbol{H} - U\boldsymbol{A})$$

where

$$\boldsymbol{E} = [E(\omega_1)\, E(\omega_2) \ldots E(\omega_{\overline{M}N})]^T \tag{28a}$$

$$W = \mathrm{diag}[W(\omega_1)\, W(\omega_2) \ldots W(\omega_{\overline{M}N})] \tag{28b}$$

$$\boldsymbol{H} = [\tilde{\boldsymbol{H}}(\omega_1)\, \tilde{H}(\omega_2) \ldots \tilde{H}(\omega_{\overline{M}N})]^T \tag{28c}$$

$$U = \begin{bmatrix} 1 & \cos(\omega_1) & \cos(2\omega_1) & \ldots & \cos(\tau_0\omega_1) \\ 1 & \cos(\omega_2) & \cos(2\omega_2) & \ldots & \cos(\tau_0\omega_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \cos(\omega_{\overline{M}N}) & \cos(2\omega_{\overline{M}N}) & \ldots & \cos(\tau_0\omega_{\overline{M}N}) \end{bmatrix} \tag{28d}$$

$$\boldsymbol{A} = [a(0)\, a(1) \ldots a(\tau_0)]^T \tag{28e}$$

with $\overline{M} < M$, as the original frequencies in the transition band were discarded.

The design of a lowpass digital filter as specified in Fig. 10, using either the minimax method or the WLS approach, is achieved making the ideal response and weight functions respectively equal to

$$\tilde{H}(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_p \\ 0 & \text{for } \omega_s \le \omega \le \pi \end{cases}$$

and

$$W(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_p \\ \delta_p/\delta_s & \text{for } \omega_s \le \omega \le \pi \end{cases}$$

Numerical Methods: The Chebyshev Approach. Chebyshev filter design consists of the minimization over the set of filter coefficients $\boldsymbol{A}$ of the maximum absolute value of $E(\omega)$; that is,

$$\|E(\omega)\|_\infty = \min_{\boldsymbol{A}} \max_{0 \le \omega \le \pi} [W(\omega)|\tilde{H}(\omega) - \hat{H}(\omega)|]$$

With the discrete set of frequencies, in Eq. (28), this minimax function becomes

$$\|E(\omega)\|_\infty \approx \min_{\boldsymbol{A}} \max_{0 \le \omega_i \le \pi} [W|\boldsymbol{H} - U\boldsymbol{A}|]$$

If we refer to Fig. 10, the minimax method effectively optimizes

$$DB_\delta = 20 \log_{10} \min(\delta_p, \delta_s) \text{ dB}$$

This problem is commonly solved with the Parks-McClellan algorithm (18–20) or some variation of it (21,22). These methods are based on the Reméz exchange routine, the solution of which can be tested for optimality using the alternation theorem as described in (18). An important feature of minimax filters is the fact that they present equiripple errors within the bands, as can be observed in Fig. 11.

Numerical Methods: The Weighted-Least-Squares Approach. The weighted least-squares (WLS) approach minimizes the function

$$\|E(\omega)\|_2^2 = \int_0^\pi |E(\omega)|^2 \, d\omega = \int_0^\pi W^2(\omega)|\tilde{H}(\omega) - \hat{H}(\omega)|^2 \, d\omega$$

For the discrete set of frequencies, in Eq. (28), this objective function is estimated by

$$\|E(\omega)\|_2^2 \approx \boldsymbol{E}^T\boldsymbol{E}$$

the minimization of which is achieved with

$$\boldsymbol{A}^* = (U^T W^2 U)^{-1} U^T W^2 \boldsymbol{H}$$

If we refer to Fig. 10, the WLS objective is to maximize the passband-to-stopband ratio (PSR) of energies

$$\text{PSR} = 10 \log_{10} \left(\frac{E_p}{E_s}\right) \text{dB}$$

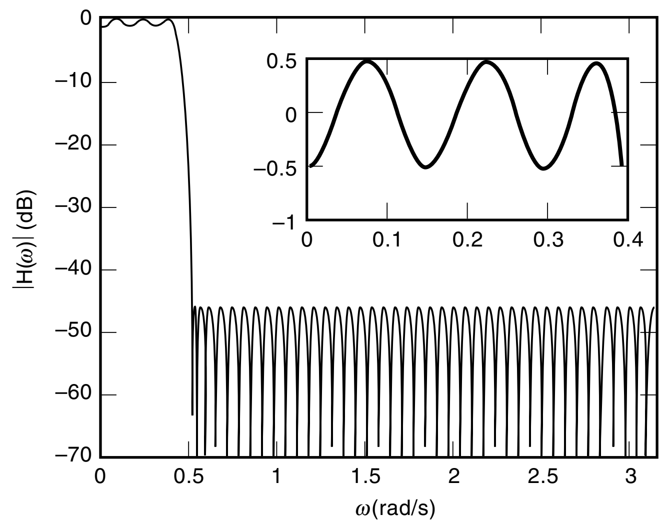

Figure 11. Equiripple bands (passband in detail) are typical for FIR filters approximated with the Chebyshev approach.

where E_p and E_s are the passband and stopband energies, respectively; that is,

$$E_\mathrm{p} = 2\int_0^{\omega_\mathrm{p}} |\hat{H}(\omega)|^2\,\mathrm{d}\omega$$
$$E_\mathrm{s} = 2\int_{\omega\mathrm{s}}^{\pi} |\hat{H}(\omega)|^2\,\mathrm{d}\omega$$

A typical lowpass nonrecursive digital filter designed with the WLS method is depicted in Fig. 12, where the large ripples near the band edges are easily identified.

In 1961, Lawson derived a scheme that performs a Chebyshev approximation as a limit of a special sequence of weighted least-p (L_p) approximations with p fixed. The particular case with $p = 2$ thus relates the Chebyshev approximation to the WLS method, taking advantage of the substantially simpler implementation of the latter. As applied to the nonrecursive digital-filter design problem, the L_2 Lawson algorithm is implemented by a series of WLS approximations using a time-varying weight matrix W_k, the elements of which are calculated by (23)

$$W_{k+1}^2(\omega) = W_k^2(\omega)|E_k(\omega)|$$

Convergence of the Lawson algorithm is slow, as usually 10 to 15 WLS iterations are required in practice to approximate the minimax solution. Accelerated versions of the Lawson algorithm can be found in (23–25).

FIR filters can be designed based solely on power-of-two coefficients. This is a very attractive feature for VLSI implementations, as time-consuming multiplications are avoided. This approach is a very modern research topic, and interested readers are referred to (12,26,27).

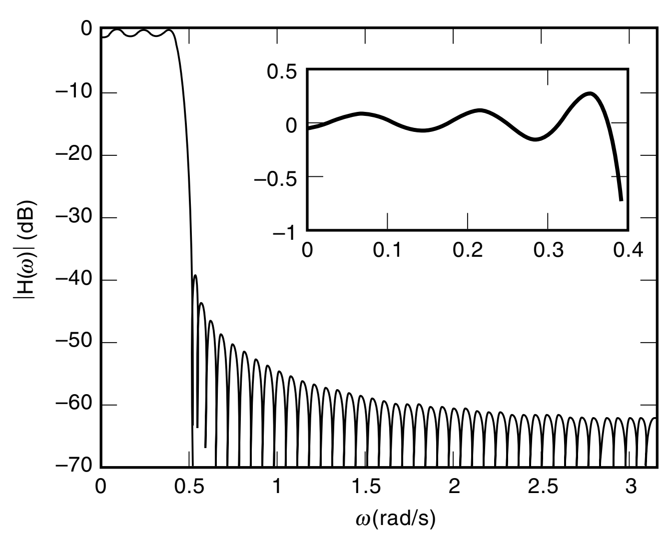

Figure 12. Large errors near band discontinuities (passband in detail) are typical for FIR filters designed with the weighted-least-squares approach.

BIBLIOGRAPHY

1. L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1975.
2. J. G. Proakis and D. G. Manolakis, *Introduction to Digital Signal Processing,* New York: Macmillan, 1988.
3. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1989.
4. A. Antoniou, *Digital Filters: Analysis, Design, and Applications,* 2nd ed., New York: McGraw-Hill, 1993.
5. L. B. Jackson, *Digital Filters and Signal Processing,* 3rd ed., Norwell, MA: Kluwer, 1996.
6. D. J. DeFatta, J. G. Lucas, and W. S. Hordgkiss, *Digital Signal Processing: A System Design Approach,* New York: Wiley, 1988.
7. S. M. Kay, *Modern Spectral Estimation,* Englewood Cliffs, NJ: Prentice-Hall, 1988.
8. S. S. Haykin, *Adaptive Filter Theory,* 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1991.
9. B. Widrow and S. Stearns, *Adaptive Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1985.
10. P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation,* Norwell, MA: Kluwer, 1997.
11. G. Strang and T. Nguyen, *Wavelets and Filter Banks,* Wellesley, MA: Wellesley-Cambridge Press, 1996.
12. S. K. Mitra and J. E. Kaiser (eds.), *Handbook for Digital Signal Processing,* New York: Wiley, 1993.
13. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1983.
14. R. Boite and H. Leich, A new procedure for the design of high order minimum phase FIR digital or CCD filters, *Signal Process,* **3**: 101–108, 1981.
15. Y. Kamp and C. J. Wellekens, Optimal design of minimum-phase FIR filters, *IEEE Trans. Acoust. Speech Signal Process.,* **ASSP-31**: 922–926, 1983.
16. E. O. Brigham, *The Fast Fourier Transform and Its Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1988.
17. J. W. Adams, A new optimal window, *IEEE Trans. Signal Process.,* **39**: 1753–1769, 1991.
18. T. W. Parks and J. H. McClellan, Chebyshev approximation for nonrecursive digital filters with linear phase, *IEEE Trans. Circuit Theory,* **CT-19**: 189–195, 1972.
19. J. H. McClellan, T. W. Parks, and L. R. Rabiner, A computer program for designing optimum FIR linear phase digital filters, *IEEE Trans. Audio Electroacoust.,* **AU-21**: 506–526, 1973.

20. L. R. Rabiner, J. H. McClellan, and T. W. Parks, FIR digital filter design techniques using weighted Chebyshev approximation, *Proc. IEEE,* **63**: 595–610, 1975.

21. J. W. Adams and A. N. Wilson, Jr., On the fast design of high-order FIR digital filters, *IEEE Trans. Circuits Syst.,* **CAS-32**: 958–960, 1985.

22. D. J. Shpak and A. Antoniou, A generalized Reméz method for the design of FIR digital filters, *IEEE Trans. Circuits Syst.,* **37**: 161–173, 1990.

23. J. R. Rice and K. H. Usow, The Lawson algorithm and extensions *Mathematics of Computation,* in press.

24. Y. C. Lim et al., A weight least squares algorithm for quasi-equiripple FIR and IIR digital filter design, *IEEE Trans. Signal Process.,* **40**: 551–558, 1992.

25. R. H. Yang and Y. C. Lim, Efficient computational procedure for the design of FIR digital filters using WLS technique, *IEE Proc.-G,* **140**: 355–359, 1993.

26. Y. C. Lim and S. R. Parker, Discrete coefficients FIR digital filter design based upon an LMS criteria, *IEEE Trans. Circuits Syst.,* **CAS-30**: 723–739, 1983.

27. T. Saramaki, A systematic technique for designing highly selective multiplier-free FIR filters, *Proc. IEEE Int. Symp. Circuits Syst.,* 1991, pp. 484–487.

SERGIO L. NETTO
GELSON V. MENDONÇA
Federal University of Rio de Janeiro