FIR FILTERS, WINDOWS

WHAT IS A FINITE-IMPULSE-RESPONSE DIGITAL FILTER?

A digital filter is a signal-processing unit that takes a data sequence and manipulates it, to produce an output sequence that is, in a certain sense, an improved version of the input. A simple example is the averaging filter, which takes N most current input samples and computes the arithmetic mean of these samples as the output. Analytically, if one denotes the input and output sequences of the averaging filter by $\{u(n)\}$ and $\{y(n)\}$, respectively, then the filter can be described by

$$y(n) = \frac{u(n) + u(n-1) + \cdots + u(n-N+1)}{N} \tag{1}$$

If the number of samples used, N, is sufficiently large, one might think of Eq. (1) as an approximation of the integration

$$y(t) = \int_{t-T}^{t} u(\tau)\, d\tau \tag{2}$$

where T is the time duration for N samples. Since integration largely eliminates zero-mean high-frequency noise, Eq. (2) suggests that the averaging filter may be used for smoothing noise-contaminated signals. Figure 1 depicts the output of the averaging filter with $N = 10$ and 40, as applied to a noise-corrupted discrete-time signal (1024 samples).

The term *digital* is used to stress that a digital filter only admits discrete signals, which may be the result of a sampling process, as applied to continuous signals, and that the samples are usually rounded off to a finite number of digits. A digital filter is called a finite-impulse-response (FIR) filter, if its response to the unit impulse is of finite length. Alternatively, a digital filter is said to be an FIR filter if its output is determined by present and some past input samples. FIR filters are often referred to as *nonrecursive* filters as they determine output samples without using past output sample values. For the sake of comparison, recall that a digital filter is said to be an infinite-impulse-response (IIR) filter if its response to the unit impulse is of infinite length. Equivalently, output of an IIR filter depends on the present and some past input samples, as well as some past output samples. IIR digital filters are often called recursive digital filters. As an example, the averaging filter described in Eq. (1) is an FIR digital filter whose impulse response is an N-sample sequence $\{1/N, 1/N, \cdots, 1/N\}$.

The class of FIR digital filters that has been utilized most often in practice is the linear and time-invariant FIR filters. A filter is said to be linear if the linear superposition law holds. A filter is said to be time-invariant if its response to a shifted (by τ samples) input can also be produced by applying the filter to the original input and then shifting the output by τ samples. The linearity and time-invariance allow a powerful frequency-domain approach to the analysis, as well as design of these filters. As an example, one can easily verify that the averaging filter in Eq. (1) is a linear and time-invariant FIR filter, since the output is generated as a *linear* combination of N input samples with *constant* coefficients.

As was mentioned earlier, digital filters can be analyzed effectively by using a frequency-domain approach. In this approach, one examines a digital filter by applying it to a spectrum of sinusoidal signals with frequencies varying over a certain range. If the filter in question is linear and time-invariant, its response to a unit-amplitude sinusoidal function is also sinusoidal of same frequency with possibly a phase shift and a different amplitude. This leads to the frequency-domain characterization of a digital filter: a digital filter may be described by its amplitude response that collects the amplitudes of the outputs at various frequencies, and its phase response that collects the phase-shifts in the outputs at these frequencies. Combined the amplitude and phase responses of a linear time-invariant digital filter is called the *frequency response.*

A general FIR digital filter of length N is described by

$$y(n) = \sum_{k=0}^{N-1} h(k)u(n-k) \tag{3}$$

whose frequency response is given by

$$H(\omega) = \sum_{k=0}^{N-1} h(k)e^{-jk\omega T} \tag{4}$$

where T is the sampling interval, $j = \sqrt{-1}$, and ω is the frequency that varies from 0 to $\omega^* = 2\pi f$ in rad/s with $f = 1/T$ in Hertz (Hz). For example, the frequency response of the averaging filter is given by

$$H(\omega) = \frac{1}{N}\sum_{k=0}^{N-1} e^{-jk\omega T}$$

For an even N, we can write the above $H(\omega)$ as

$$H(\omega) = \frac{2e^{-j(N-1)\omega T/2}}{N} \sum_{k=0}^{\frac{N}{2}-1} \cos\left[\left(\frac{N-2k-1}{2}\right)\omega T\right]$$

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

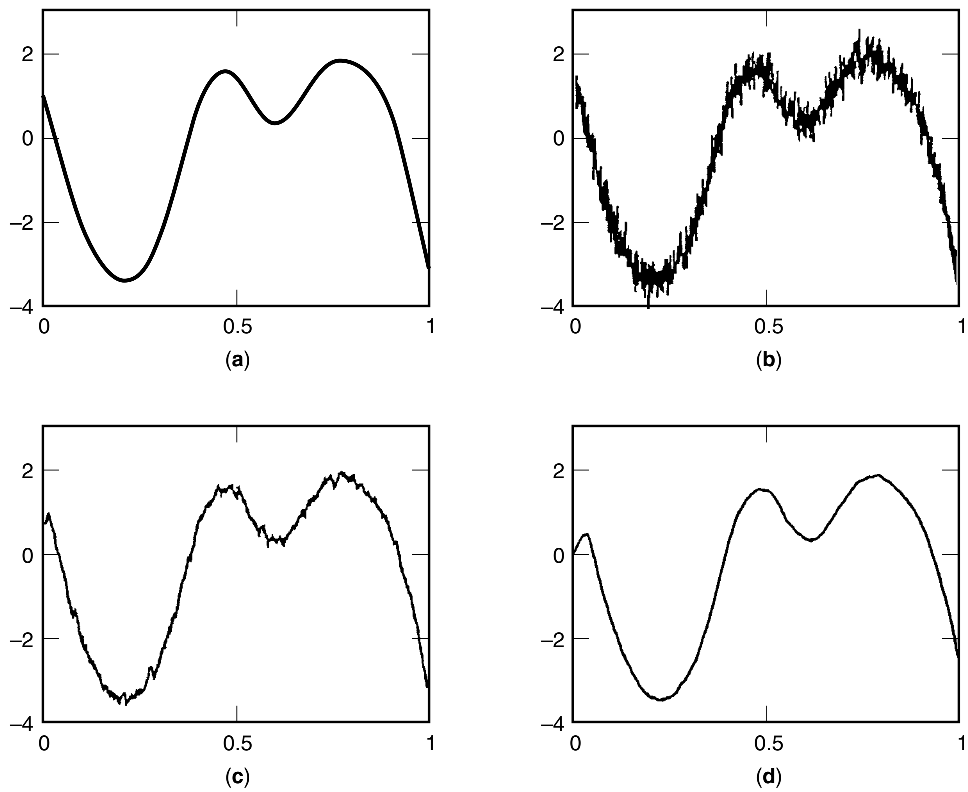


Figure 1. A pure and clean signal is shown in (a). The signal contains only low frequency components. When the signal is corrupted with high-frequency noise, it looks like the one shown in (b). The averaging filter acts like a low-pass filter, which can be utilized to reduce high-frequency noise. The outcomes of applying the averaging filter with $N = 10$ and $N = 40$ to the noise-corrupted signal are depicted in (c) and (d), respectively.

which implies a linear phase response $P(\omega) = -(N-1)\omega T/2$ and an amplitude response

$$A(\omega) = \frac{2}{N}\sum_{k=0}^{N/2-1}\cos\left[\left(\frac{N-2k-1}{2}\right)\omega T\right]$$

In particular, the 2-term averaging filter has the frequency response $H(\omega) = e^{-j\omega T/2}\cos\omega T/2$. Figure 2 depicts the magnitude response $|H(\omega)|$ of the averaging filter with $T = 1$ and $N = 2$, 10, and 40. A common feature of these filters is the near-unity gain at low frequencies and the large amplitude attenuation for input signals with high-frequency ingredients. This explains why the averaging filter acts like a smoothing operator, as it allows low-frequency components of an input to pass but rejects largely high-frequency components. Digital filters of this type are called lowpass filters. As the second example, the following is an FIR filter that finds the difference between two consecutive input samples:

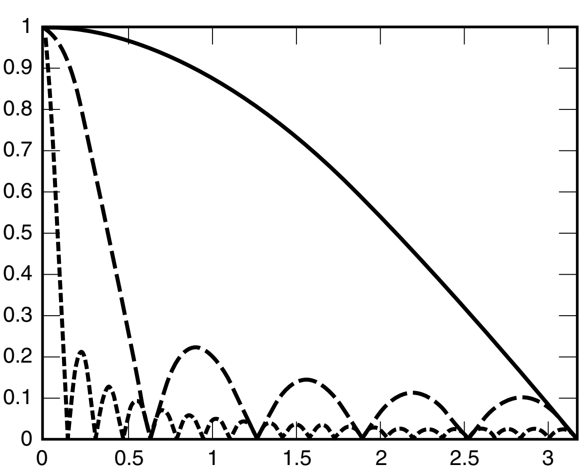


Figure 2. The magnitude response of a digital filter describes the filter's gain over the entire frequency band. Shown in the figure are the magnitude responses of the averaging filter with $N = 2$ (solid line), $N = 10$ (dashed line), and $N = 40$ (dotted line). Obviously, the averaging filter is a kind of low-pass filter whose passband and stopband widths largely depend on filter length N.

$$y(n) = \frac{u(n) - u(n-1)}{2} \tag{5}$$

which shall be referred to as the differentiating filter. Obviously, the input–output relation in Eq. (5) represents a linear, time-invariant FIR digital filter, whose frequency response is given by

$$H(\omega) = \frac{1 - e^{-j\omega T}}{2} = e^{-j\left(\frac{\omega T}{2}+\frac{\pi}{2}\right)}\sin\frac{\omega T}{2} \tag{6}$$

This implies that the differentiating filter has a linear phase response $P(\omega) = -(\omega T + \pi)/2$ and an amplitude response $\sin \omega T/2$. From Fig. 3, one can see that the differentiating filter has a gain close to 1 for high frequencies and large attenuation for low frequencies. Filters of this type are referred to as high-pass filters.

If z^{-1} is used to denote the delay operator, meaning that $z^{-1}u(i) = u(i-1)$, the averaging filter described by Eq. (1) can be written as

$$y(n) = H(z)u(n) \tag{7}$$

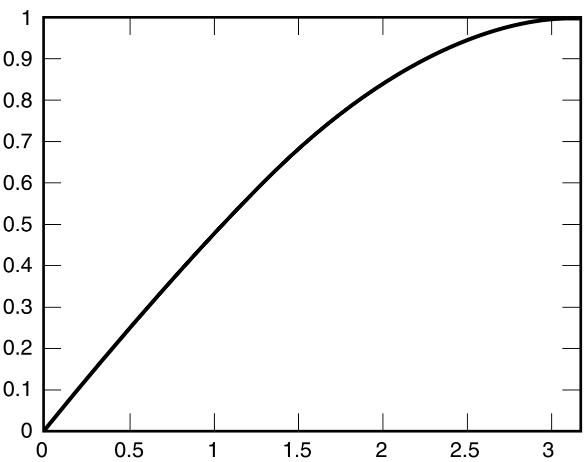


Figure 3. Magnitude response of the differentiating filter. Note the large gain of the filter in high-frequency range and small gain in low-frequency range. The differentiating filter is therefore often used as a high-pass filter.

with

$$H(z) = \frac{1 + z^{-1} + \cdots + z^{-(N-1)}}{N} \tag{8}$$

From Eq. (7) it is quite clear that the averaging filter is characterized by $H(z)$ as it *transfers* an input $\{u(n)\}$ to an output $\{y(n)\}$. Formally, one may apply the z transform to Eq. (1) to obtain

$$Y(z) = H(z)U(z) \tag{9}$$

where $U(z)$ and $Y(z)$ are the z transform of $\{u(n)\}$ and $\{y(n)\}$, respectively, and $H(z)$ is again given by Eq. (8) with z interpreted as a *complex variable.* The $H(z)$ in Eq. (9) is called the *transfer function* of the filter. From Eq. (9) note that the transfer function relates input $U(z)$ to output $Y(z)$ and is determined by their ratio: $H(z) = Y(z)/U(z)$. On the other hand, from Eq. (8) note that the same $H(z)$ is independent of both input and output. The independence means that the transfer function characterizes the digital filter itself, whose structure remains unchanged, regardless of the external signals, while its close connection to the input/output explains why the analysis, design, and implementation of a digital signal processing (DSP) system is primarily centered around the system's transfer function.

For the general FIR digital filter in Eq. (3), the transfer function is given by

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \tag{10}$$

where $\{h(n),\ n = 0, \cdots, N-1\}$ is the impulse response of the filter. The averaging filter and the differentiating filter fit easily into Eq. (10) with $h(n) = 1/N$, $0 \le n \le N-1$, and $N = 2$, $h(0) = -h(1) = 1/2$, respectively. A question that would naturally arise at this point is: For fixed filter length N, is the averaging filter the best choice for noise removal? If not, then how should one choose $\{h(n)\}$, in order for the filter to achieve improved or best performance in accordance with certain criteria? These questions will be addressed later, where several algorithms for the design of FIR digital filters are described in detail. The following presents a quantitative analysis of FIR digital filters, which shall lead to a better understanding of this class of filters and facilitate the development of several modern design methods.

ANALYSIS OF FIR DIGITAL FILTERS

First an implementation issue: Given the transfer function of an FIR filter and an input signal $\{u(n)\}$, how should the output $\{y(n)\}$ be computed?

Recall that the filter with transfer function in Eq. (10) can also be described by Eq. (3). The operation in Eq. (3) is known as the *convolution sum* of impulse response $\{h(n)\}$, with input $\{u(n)\}$. The substance of Eq. (3) is that the output is the inner product of two N-dimensional vectors $\mathbf{h} = [h(0)\ h(1)\ \cdots\ h(N-1)]^T$ and $\mathbf{u}_n = [u(n)\ u(n-1)\ \cdots\ u(n-N+1)]^T$:

$$y(n) = \mathbf{h}^T\mathbf{u}_n \tag{11}$$

With a given impulse response vector $\mathbf{h}$, one might think of Eq. (11) as sum of N products when sequence $\{\mathbf{u}_n\}$ is sliding forward on a finite-length supporting sequence $\{h(i)\}$, where the inner product is performed to generate output $y(n)$. This interpretation of FIR filtering is illustrated in Fig. 4.

Filter representation in Eq. (11) is also of use to conclude that an FIR digital filter is always *stable.* Essentially the stability of a filter means that small changes in the input do not, in any event, lead to large changes in the output. Let $\boldsymbol{\delta}_n$ be the disturbance vector to the input $\mathbf{u}_n$. The filter's response to the disturbed input is $y_d(n) = \mathbf{h}^T(\mathbf{u}_n + \boldsymbol{\delta}_n)$, hence the change in output due to the disturbances is

$$y_d(n) - y(n) = \mathbf{h}^T\boldsymbol{\delta}_n$$

By the Schwarz inequality (1), one obtains

$$|y_d(n) - y(n)| \le \|\mathbf{h}\|_2\|\boldsymbol{\delta}_n\|_2 \tag{12}$$

Since the Euclidean norm $\|\mathbf{h}\|_2$ is fixed and finite, Eq. (12) indicates that the changes in output will be small, as long as the magnitude of the input disturbances is sufficiently small.

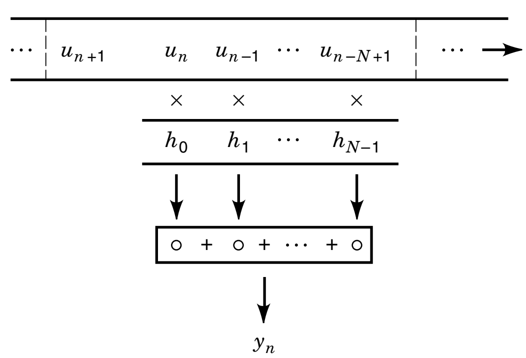


Figure 4. The convolution sum described in Eq. (3) can be interpreted as inner product of the impulse response vector with the vector composed of the block of N most current input samples.

Frequency Response of an FIR Filter

For the sake of simplicity, assume the frequency ω in Eq. (4) is normalized, such that ω varies from $-\pi$ to π. This implicitly means that the normalized frequency is equal to ωT in Eq. (4), hence the frequency response of the filter is given by

$$H(\omega) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} \tag{13}$$

with $-\pi \leq \omega < \pi$. The reason why the frequency response is usually considered on $[-\pi, \pi)$ is because $H(\omega)$ is 2π-periodic, i.e., $H(\omega + 2k\pi) = H(\omega)$ holds for any integer k. Recall that the discrete Fourier transform (DFT) of a length-N sequence $\{h(n), 0 \leq n \leq N - 1\}$ is defined by

$$D_N(k) = \sum_{n=0}^{N-1} h(n) e^{-j2\pi kn/N} \qquad 0 \leq k \leq N-1$$

It follows that

$$D_N(k) = H\left(\frac{2\pi k}{N}\right) \qquad 0 \leq k \leq N-1$$

Likewise, the DFT of a length-L sequence $\{h(0), \cdots, h(N - 1), 0, \cdots, 0\}$, in which $L - N$ zeros are appended, is given by

$$D_L(k) = \sum_{n=0}^{N-1} h(n) e^{-j2\pi kn/L} \qquad 0 \leq k \leq L-1 \tag{14}$$

which implies that

$$D_L(k) = H\left(\frac{2\pi k}{L}\right) \qquad 0 \leq k \leq L-1$$

Therefore, by appending appropriate number of zeros to the impulse response, the DFT in Eq. (14) can be used to evaluate the frequency response $H(\omega)$ at any number of equally spaced frequencies. The DFT of an L-sample sequence, particularly when L is large and is a power of 2, can be computed by means of efficient fast-Fourier-transform (FFT) algorithms. The reader is referred to Chapter 13 of Antoniou (2) for a detailed account of these algorithms and several applications of the DFT in the design and implementation of digital filters.

The frequency response $H(\omega)$ is a complex-valued function of real variable ω, whose physical meaning is frequency. With $z = e^{j\omega}$, Eq. (9) gives

$$Y(\omega) = H(\omega)U(\omega) \tag{15}$$

One might think of Eq. (15) as an input $U(\omega)$ passing through an "amplifier" whose gain, $H(\omega)$, is *frequency sensitive* and *complex-valued.* Hence a digital filter can be viewed as a device that modifies the frequency contents of an input signal and that modifies at a fixed frequency both amplitude and phase of the harmonic component of the input. Taking the 3-term averaging filter as an example, its frequency response is

$$H(\omega) = (1 + e^{-j\omega} + e^{-j2\omega})/3 = e^{-j\omega}\left(\frac{1 + 2\cos\omega}{3}\right)$$

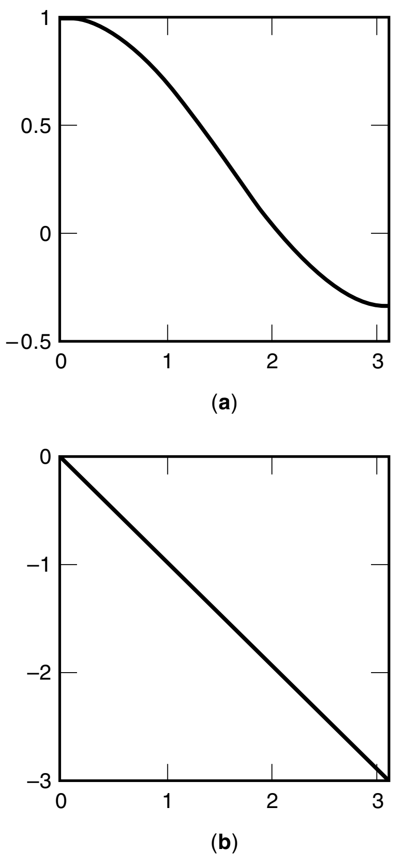


Figure 5. Frequency response of the 3-term averaging filter. Its low-pass amplitude response is shown in (a), while its linear phase response is shown in (b).

So if one writes

$$H(\omega) = e^{jP(\omega)} A(\omega) \tag{16}$$

where $A(\omega)$ and $P(\omega)$ are called the amplitude response and phase response, respectively, then, for the 3-term averaging filter, one obtains

$$A(\omega) = \frac{1 + 2\cos\omega}{3} \qquad \text{and} \qquad P(\omega) = -\omega$$

Note that $A(\omega)$ might take negative values for some ω, but the relation of $A(\omega)$ to the magnitude of $H(\omega)$ is simple: $A(\omega) = \pm|H(\omega)|$. The expression of $H(\omega)$ in Eq. (16) allows the phase response $P(\omega)$ to be a continuous function; see Parks and Burrus (3). The $A(\omega)$ and $P(\omega)$ of the 3-term averaging filter are depicted in Fig. 5.

Linear-Phase FIR Filters

A common interpretation of the phase response of a digital filter is related to the time delay or spatial shift occurred to the output as compared with the input. Indeed, if one defines the *group delay* of the filter as

$$D(\omega) = -\frac{dP(\omega)}{d\omega}$$

then $D(\omega)$ is the delay caused by the filtering process for the harmonic component of frequency ω in the input. Obviously, this delay is inevitable for any causal filter. When one examines Fig. 1 more carefully, for example, the delay in the filter's outputs can easily be observed. If the group delay is independent of frequency ω, that is, $D(\omega)$ remains constant for all frequencies, then there will be no phase distortion in the output, an obviously desirable property of a DSP system. Since a constant group delay is equivalent to a linear phase response, that is,

$$P(\omega) = k\omega + k_0 \qquad k, k_0 \text{ constant}$$

there is often interest in linear phase digital filters. There are four types of FIR digital filter with linear phase response:

Type 1. Symmetric $\{h(n), 0 \le n \le N-1\}$ with N odd. Namely,

$$h(n) = h(N-1-n) \qquad 0 \le n \le N-1 \tag{17}$$

the frequency response in this case can be written as

$$H(\omega) = e^{j\omega(N-1)/2} \sum_{k=0}^{(N-1)/2} a(k) \cos k\omega \tag{18}$$

where

$$a(k) = \begin{cases} h\left(\dfrac{N-1}{2}\right) & k = 0 \\ 2h\left(\dfrac{N-1}{2} - k\right) & 1 \le k \le (N-1)/2 \end{cases}$$

Type 2. Symmetric $\{h(n), 0 \le n \le N-1\}$ with N even. The frequency response is given by

$$H(\omega) = e^{-j\omega(N-1)/2} \sum_{k=1}^{N/2} a(k) \cos\left[\left(k - \frac{1}{2}\right)\omega\right] \tag{19}$$

where

$$a(k) = 2h\left(\frac{N}{2} - k\right) \qquad 1 \le k \le N/2$$

Type 3. Antisymmetric $\{h(n), 0 \le n \le N-1\}$ with N odd. Namely,

$$h(n) = -h(N-1-n) \qquad 0 \le n \le N-1 \tag{20}$$

Since N is odd, Eq. (20) implies that $h(N - 1/2) = 0$, and the frequency response in this case becomes

$$H(\omega) = e^{-j[\omega(N-1)+\pi]/2} \sum_{k=1}^{(N-1)/2} a(k) \sin k\omega \tag{21}$$

where

$$a(k) = 2h\left(\frac{N-1}{2} - k\right) \qquad 1 \le k \le (N-1)/2$$

Type 4. Antisymmetric $\{h(n), 0 \le n \le N-1\}$ with N even. The frequency response is given by

$$h(\omega) = e^{-j[\omega(N-1)+\pi]/2} \sum_{k=1}^{N/2} a(k) \sin\left[\left(k - \frac{1}{2}\right)\omega\right] \tag{22}$$

where

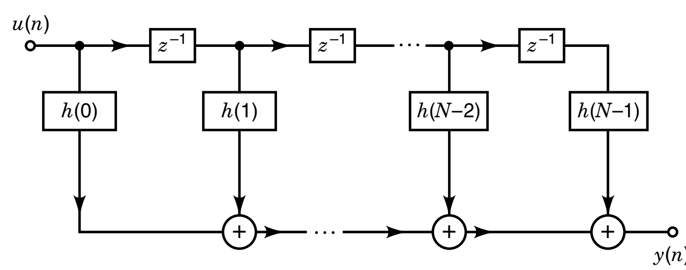
$$a(k) = 2h\left(\frac{N}{2} - k\right)$$

In summary, the symmetry constraints Eq. (17) or Eq. (20) ensure a linear phase response for the FIR filter in question. For a fixed filter length N, however, the symmetry constraints also reduce the number of independent filter coefficients by (or nearly by) half. The next section shall focus attention on how these independent coefficients should be chosen, so as for $H(\omega)$ to approximate a desired frequency response $H_d(\omega)$.

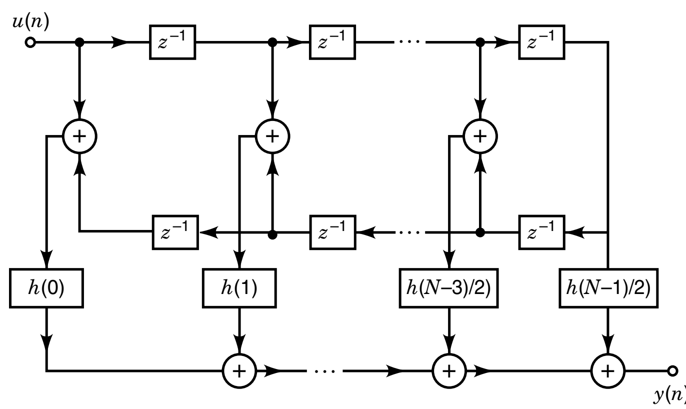
REALIZATION OF FIR FILTERS

Given the transfer function of an FIR filter, realization is the process of translating the transfer function into a digital-filter network, which can then be implemented by means of software, dedicated hardware, or a combination of both.

Consider the transfer function $H(z)$ of a general FIR filter in Eq. (10), a *direct form* for the realization of $H(z)$ is depicted in Fig. 6(a). In the literature, this structure is sometimes called a *transversal filter* or a *tapped delay line*. Obviously, the implementation of this direct form requires N multiplications per output sample. For linear-phase FIR filters, direct-form realizations with reduced number of multipliers are



(a)



(b)

Figure 6. (a) Direct form realization of an FIR filter. Among other things, implementing this realization requires N multipliers. The realization shown in (b) takes the advantage of a linear phase filter whose coefficients are symmetrical with respect to the midpoint. The number of multipliers in this case is reduced to $(N + 1)/2$.

available (4). Take a Type 1 filter of length N as an example, with Eq. (17) the transfer function can be written as

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

This leads to the direct-form realization of $H(z)$ in Fig. 6(b) which requires only $(N + 1)/2$ multiplications per output sample. Similar savings can be accomplished for other types of linear-phase FIR filter (4). If the order of $H(z)$ is higher than 2, then by treating $H(z)$ as a polynomial in z^{-1}, $H(z)$ can be expressed as a product of second-order or first-order polynomial factors. Namely,

$$H(z) = h(0)\prod_{k=1}^{K}(1 + h_k(1)z^{-1} + h_k(2)z^{-2})$$

where $h_k(2) = 0$ for a first-order factor. This leads to the *cascade form* for the realization of $H(z)$, see Fig. 7. For a Type 1 FIR filter, its cascade-form realization has $(N - 1)/2$ sections, and the total number of multiplications per output sample is N.

Another scheme for the realization of FIR filters is the so-called *polyphase realization* initiated in (5), which has found useful in multirate DSP (6). Take an FIR filter of length 7 as an example, one can write

$$H(z) = \sum_{n=0}^{6} h(n)z^{-n} = H_0(z^2) + z^{-1}H_1(z^2)$$

with

$$H_0(z) = h(0) + h(2)z^{-1} + h(4)z^{-2} + h(6)z^{-3}$$
$$H_1(z) = h(1) + h(3)z^{-1} + h(5)z^{-2}$$

which suggests a parallel realization of $H(z)$ as depicted in Fig. 8(a). Note that the same transfer function can also be expressed as

$$H(z) = H_0(z^3) + z^{-1}H_1(z^3) + z^{-2}H_2(z^3)$$

with

$$H_0(z) = h(0) + h(3)z^{-1} + h(6)z^{-2}$$
$$H_1(z) = h(1) + h(4)z^{-1}$$
$$H_1(z) = h(2) + h(5)z^{-1}$$

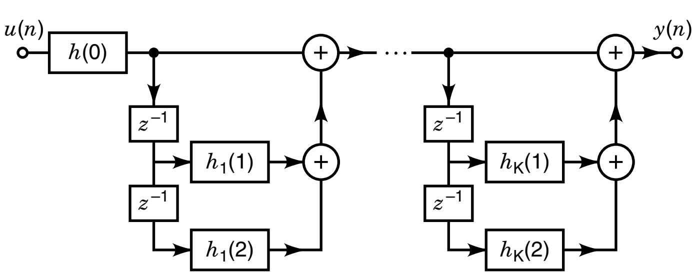


Figure 7. An FIR filter can be realized with a cascade form that consists of several second-order or first-order sections. The total number of multipliers used is N.

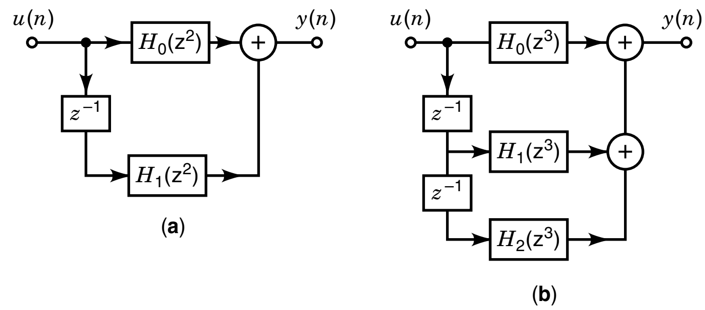


Figure 8. Two polyphase realizations for an FIR filter. Unlike the cascade form in Fig. 7, the polyphase decomposition components are connected in parallel through unit delay units.

which leads to the realization shown in Fig. 8(b). In general, an M-branch polyphase decomposition of $H(z)$ of length N assumes the form

$$H(z) = \sum_{k=0}^{M-1} z^{-k}H_k(z^M)$$

with

$$H_k(z) = \sum_{n=0}^{\lfloor N/M \rfloor} h(Mn + k)z^{-n} \qquad 0 \le k \le M - 1$$

and $h(n) \equiv 0$ for $n \ge N$, where $\lfloor N/M \rfloor$ denotes the largest integer that is no larger than N/M. The M-branch polyphase realization of $H(z)$ is illustrated in Fig. 9.

FIR FILTERS VERSUS IIR FILTERS

FIR digital filters have been the preferred filtering scheme in many DSP applications. This is due mainly to the advantages of the FIR filter designs compared to their IIR counterparts, which are summarized as follows:

(a) Exact linear phase response can be achieved easily by imposing certain symmetry condition on filter's coefficients.
(b) Stability and freedom of limit cycle difficulties in their finite wordlength implementations.
(c) Availability of effective methods for the design of a variety of FIR digital filters.

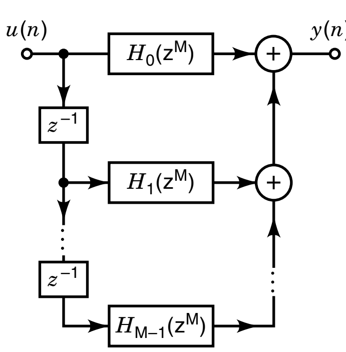


Figure 9. M-branch polyphase realization of an FIR filter.

(d) Low output noise due to coefficient quantization and multiplication roundoff errors.

On the contrary, for IIR filters exact linear phase response (in the passband) cannot be achieved in general, stability has always been a concern during the design, and output roundoff noise as well as coefficient sensitivity may become severe unless particular cares are taken to deal with these problems.

The main disadvantage of FIR designs is that the required order in an FIR design can be considerably higher than its IIR counterpart for the same design specification, especially when the transition bands are narrow. Consequently, the implementation of FIR filter with high selectivity (i.e., narrow transition bands) can be costly, although the implementation cost may be reduced by using fast convolution algorithms and multirate filtering (4,6). On the other hand, since the poles of the transfer function in a recursive filter can be placed anywhere inside the unit circle, high selectivity can readily be achieved with lower-order IIR filters. Therefore, for high-selectivity applications where computational efficiency is more important than the delay characteristic, IIR designs are more suitable.

In what follows we review several commonly used algorithms for the design of FIR filters. These include the design method based on Fourier series and window functions, design by weighted least-squares minimization, design based on weighted Chebyshev approximation, and methods for the design of Mth band and half-band filters.

DESIGN METHODS

Design Based on Fourier Series

Let $H_d(\omega)$ be the desired frequency response with normalized frequency $\omega \in [-\pi, \pi]$. The Fourier series of $H_d(\omega)$ is given by

$$H_d(\omega) = \sum_{n=-\infty}^{\infty} h_d(n) e^{-j\omega n}$$

where

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\omega) e^{j\omega n}\, d\omega \tag{23}$$

For an odd integer N, denote

$$H_1(z) = \sum_{n=-(N-1)/2}^{(N-1)/2} h_d(n) z^{-n}$$

where $h_d(n)$ is given by Eq. (23). Obviously, with a sufficiently large N, $H_1(\omega)$ represents a reasonable approximation of $H_d(\omega)$. A causal, linear phase, FIR filter of length N can now be obtained as

$$H(z) = z^{-(N-1)/2} H_1(z) \tag{24}$$

It can be shown that $h_d(n) = h_d(-n)$ for $1 \le n \le (N-1)/2$, hence $H(z)$ has a linear phase response. It also can be shown that the FIR filter represented by Eq. (24) is the optimal approximation of $H_d(\omega)$ for a given filter length N in the least-squares sense. This is to say that $H(z)$ in Eq. (24) minimizes the approximation error

$$e_2 = \int_{-\pi}^{\pi} |H(\omega) - H_d(\omega)|^2\, d\omega$$

A problem with this $H(z)$ is that it tends to introduce Gibbs' oscillations in the frequency response of the filter. These oscillations are particularly pronounced in the regions where the desired frequency response has abrupt discontinuities; see Antoniou (2). The magnitudes of Gibbs', oscillations can be reduced by using discrete *window functions.* A discrete window function is defined as $\{w(n),\ n = \text{integers}\}$, such that $w(n) = 0$ for $|n| > (N-1)/2$ and $w(n) = w(-n)$. Typically, the amplitude spectrum of a window function useful in filter design consists of a main lobe and several side lobes, with the area of the side lobes considerably smaller than that of the main lobe. The most frequently used window functions are as follows (parameter N in the window functions below is assumed to be an odd integer):

1. *Rectangular Window.*

$$w_r(n) = \begin{cases} 1 & \text{for } |n| \le (N-1)/2 \\ 0 & \text{elsewhere} \end{cases}$$

2. *von Hann and Hamming Windows.*

$$w_h(n) = \begin{cases} \alpha + (1-\alpha)\cos\dfrac{2\pi n}{N-1} & \text{for } |n| \le (N-1)/2 \\ 0 & \text{elsewhere} \end{cases}$$

where $\alpha = 0.5$ in the von Hann window and $\alpha = 0.54$ in the Hamming window.

3. *Blackman Window.*

$$w_b(n) = \begin{cases} 0.42 + 0.5\cos\dfrac{2\pi n}{N-1} + 0.08\cos\dfrac{4\pi n}{N-1} & \text{for } |n| \le (N-1)/2 \\ 0 & \text{elsewhere} \end{cases}$$

4. *Kaiser Window.*

$$w_k(n) = \begin{cases} I_0(\alpha)/I_0(\beta) & \text{for } |n| \le (N-1)/2 \\ 0 & \text{elsewhere} \end{cases}$$

where β is an independent parameter and

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

Function $I_0(x)$ is the zeroth-order modified Bessel function of the first kind and can be evaluated using

$$I_0(x) = 1 + \sum_{k=1}^{\infty}\left[\frac{1}{k!}\left(\frac{x}{2}\right)^k\right]^2$$

An attractive property of the Kaiser window is that its ripple ratio, which is defined as the ratio of the maximum side-lobe magnitude to the main-lobe magnitude, can be adjusted continuously from the low value of the Blackman window to the high value of the rectangular window by changing parameter β. Typical values for β are in the range $4 \leq \beta \leq 9$.

5. *Dolph-Chebyshev Window.*

$$w_{dc}(n) = \begin{cases} \dfrac{1}{N}\left[\dfrac{1}{r} + 2\displaystyle\sum_{k=1}^{(N-1)/2} C_{N-1}\left(x_0 \cos\frac{k\pi}{N}\right)\cos\frac{2k\pi n}{N}\right] & \text{for } |n| \leq (N-1)/2 \\ 0 & \text{elsewhere} \end{cases}$$

where

$$r = \frac{\text{amplitude of side lobes}}{\text{amplitude of main lobe}}$$

$$x_0 = \cosh\left(\frac{1}{N-1}\cosh^{-1}\frac{1}{r}\right)$$

and $C_k(x)$ is the Chebyshev polynomial given by

$$C_k(x) = \begin{cases} \cos(k\cos^{-1}x) & \text{for } |x| \leq 1 \\ \cosh(k\cosh^{-1}x) & \text{for } |x| > 1 \end{cases}$$

The most significant properties of the Dolph-Chebyshev window are that the magnitude of the side lobes are all equal; the main-lobe width is minimum for a given ripple ratio; and the ripple ratio can be assigned independently; see Antoniou (2).

Once a window function $\{w(n), |n| \leq (N-1)/2\}$ is chosen, the impulse response obtained by Eq. (23) is modified to

$$h_w(n) = w(n)h_d(n) \tag{25}$$

and a causal, linear phase FIR filter is obtained as

$$H(z) = z^{-(N-1)/2}\sum_{n=-(N-1)/2}^{(N-1)/2} h_w(n)z^{-n} \tag{26}$$

When an appropriate window function is employed, the modification described by Eq. (25) has proven effective in reducing Gibbs' oscillations. As an example, consider designing a linear phase FIR filter of length $N = 31$ to approximate the desired low-pass frequency response

$$A(\omega) = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & \omega_c < |\omega| \leq \pi \end{cases} \tag{27}$$

This implies that

$$h_d(n) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{jn\omega}d\omega = \frac{\sin n\omega_c}{n\pi} \tag{28}$$

with

$$h_d(0) = \frac{\omega_c}{\pi} \tag{29}$$

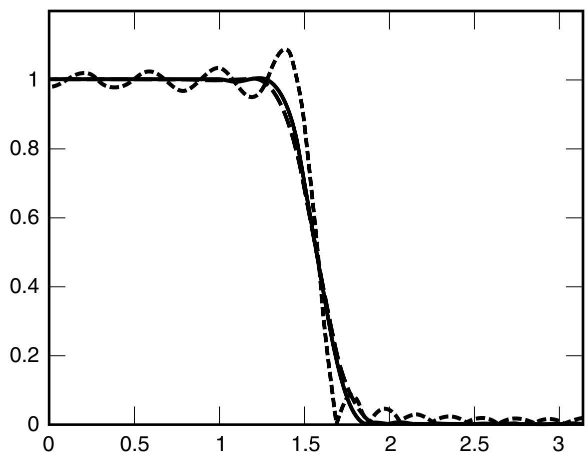


Figure 10. Magnitude responses of three FIR filters with $N = 41$. The response of the filter obtained using truncated Fourier series (dotted line) exhibits significant Gibbs' oscillations in the vicinity of cutoff frequency. The Gibb' oscillations are eliminated when a Kaiser window with $\beta = 4$ (solid line) or a Hamming window (dashed line) is employed.

With $\omega_c = \pi/2$, Fig. 10 depicts the magnitude responses of the filters designed using Hamming, and Kaiser (with $\beta = 4.0$) windows as compared with the truncated-Fourier-series based FIR filter described by Eq. (24). It can be observed that the pronounced Gibbs' oscillations in the vicinity of $\omega = \omega_c$ in the truncated-Fourier-series filter is considerably reduced in the window-function-based designs.

FIR digital filters satisfying prescribed specifications can readily be designed using popular window functions. As an example, below is a step-by-step description of such an algorithm for the design of a linear-phase lowpass FIR filter of odd length N, such that its largest passband ripple is no larger than r_p (dB), and its smallest stopband attenuation is no less than a_s (dB). The algorithm utilizes a Kaiser window with parameter β determined in Step 4 of the algorithm. The inputs of the algorithm are filter length N, normalized passband edge ω_p (so that the Nyquist frequency $\omega^*/2$ is π, where ω^* is the sampling frequency $\omega^* = 2\pi/T$), normalized stopband edge ω_s, upper bound of passband ripple r_p (dB), and lower bound of stopband attenuation a_s (dB).

Algorithm

1. Compute $h_d(n)$, using Eq. (28) and Eq. (29) with $\omega_c = (\omega_p + \omega_s)/2$.
2. Find $\delta = \min(\delta_1, \delta_2)$, where

$$\delta_1 = 10^{-0.05a_s} \qquad \text{and} \qquad \delta_2 = \frac{10^{0.05r_p} - 1}{10^{0.05r_p} + 1}$$

3. Calculate $a_s^* = -20 \log \delta$.
4. Choose parameter β as follows:

$$\beta = \begin{cases} 0 & \text{if } a_s^* \leq 21 \\ 0.5842(a_s^* - 21)^{0.4} + 0.07886(a_s^* - 21) & \text{if } 21 < a_s^* \leq 50 \\ 0.1102(a_s^* - 8.7) & \text{if } a_s^* > 50 \end{cases}$$

5. Choose parameter d as follows:

$$d = \begin{cases} 0.9222 & \text{if } a_s^* \le 21 \\ \dfrac{a_s^* - 7.95}{14.36} & \text{if } a_s^* > 21 \end{cases}$$

and select the smallest odd integer that is greater than or equal to $1 + 2\pi d/(\omega_a - \omega_p)$ as filter length N.

6. Compute the Kaiser window $\{\omega_k(n),\ |n| \le (N-1)/2\}$, using parameter β obtained from Step 4.
7. Obtain FIR transfer function $H(z)$, using Eq. (25) and Eq. (26), with $w(n) = w_k(n)$ obtained from Step 6.

Algorithms for the design of high-pass, bandpass, and bandstop FIR filters, which satisfy prescribed specifications, as well as illustrative design examples, can be found in Antoniou (2).

Design by Weighted Least-Squares Minimization

To consider designing a linear phase FIR filter of odd length N, let $A_d(\omega)$ be the desired amplitude response. It follows, from Eq. (18) that the filter's amplitude response is given by

$$A(\omega) = \sum_{k=0}^{(N-1)/2} a(k) \cos k\omega$$

A weighted least-squares design seeks to find coefficients $\{a(k),\ 0 \le k \le (N-1)/2\}$, such that the error

$$e_{2w} = \int_0^{\pi} W(\omega)[A(\omega) - A_d(\omega)]^2\, d\omega \tag{30}$$

is minimized, where $W(\omega) \ge 0$ is a known weighting function. The weighting function can be used for different purposes. For the design of typical digital filters such as low-pass, high-pass, and bandstop filters, with specified transition bands, piecewise constant $W(\omega)$ may be used to emphasize or deemphasize certain frequency regions. For instance, minimizing the weighted least-squares error

$$\tilde{e}_{2w} = \int_0^{\omega_p} [A(\omega) - 1]^2\, d\omega + w \int_{\omega_s}^{\pi} A^2(\omega)\, d\omega \tag{31}$$

subject to

$$|A(\omega) - 1| \le \delta_p \qquad 0 \le \omega \le \omega_p \tag{32}$$

$$|A(\omega)| \le \delta_s \qquad \omega_s \le \omega \le \pi \tag{33}$$

with sufficiently large weight w yields a low-pass FIR filter with nearly equiripple passband and peak-constrained least-squares stopband (ERPPCLSS), where ω_p and ω_s are passband and stopband edges, respectively, δ_p is the maximum ripple allowed in passband, and δ_s is the largest peak allowed in stopband; see Adams (7). Note that $\tilde{e}_{2w}$ in Eq. (31) is a special case of e_{2w} in Eq. (30) with

$$A_d(\omega) = \begin{cases} 1 & 0 \le \omega \le \omega_p \\ 0 & \omega_s \le \omega \le \pi \end{cases}$$

and

$$W(\omega) = \begin{cases} 1 & 0 \le \omega \le \omega_p \\ 0 & \omega_p < \omega < \omega_s \\ w & \omega_s \le \omega \le \pi \end{cases}$$

Also note that $\tilde{e}_{2w}$ in Eq. (31) can be expressed explicitly as a quadratic function of $\mathbf{x} = [a(0)\ a(1)\ \cdots\ a(N-1/2)]^T$:

$$\tilde{e}_{2w} = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{x}^T\mathbf{b} + \omega_p \tag{34}$$

where

$$\mathbf{Q} = 2\int_0^{\omega_p} \mathbf{c}(\omega)\mathbf{c}^T(\omega)\, d\omega + 2w \int_{\omega_s}^{\pi} \mathbf{c}(\omega)\mathbf{c}^T(\omega)\, d\omega$$

$$\mathbf{b} = -2\int_0^{\omega_p} \mathbf{c}(\omega)\, d\omega$$

with

$$\mathbf{c}(\omega) = \left[1\ \cos\omega \cdots \cos\left(\frac{N-1}{2}\right)\omega\right]^T$$

Obviously, matrix $\mathbf{Q}$ and vector $\mathbf{b}$ in Eq. (34) can be computed in closed form. The constraints in Eqs. (32) and (33) are linear inequalities with respect to $\mathbf{x}$:

$$1 - \delta_p \le \mathbf{c}^T(\omega)\mathbf{x} \le 1 + \delta_p \qquad 0 \le \omega \le \omega_p$$

$$-\delta_s \le \mathbf{c}^T(\omega)\mathbf{x} \le \delta_s \qquad \omega_s \le \omega \le \pi$$

Evaluating these inequalities at a set of grid points $\{\omega_{pi},\ 1 \le i \le n_p\}$ on $[0, \omega_p]$ and at a set of grid points $\{\omega_{si},\ 1 \le i \le n_s\}$ on $[\omega_s, \pi]$, all the constraints can be put together as

$$\mathbf{C}\mathbf{x} \le \mathbf{d} \tag{35}$$

with

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}^T(\omega_{p1}) \\ \vdots \\ \mathbf{c}^T(\omega_{p,n_p}) \\ -\mathbf{c}^T(\omega_{p1}) \\ \vdots \\ -\mathbf{c}^T(\omega_{p,n_p}) \\ \mathbf{c}^T(\omega_{s1}) \\ \vdots \\ \mathbf{c}^T(\omega_{s,n_s}) \\ -\mathbf{c}^T(\omega_{s1}) \\ \vdots \\ -\mathbf{c}^T(\omega_{s,n_s}) \end{bmatrix}, \qquad \mathbf{d} = \begin{bmatrix} \delta_p + 1 \\ \vdots \\ \delta_p + 1 \\ \delta_p - 1 \\ \vdots \\ \delta_p - 1 \\ \delta_s \\ \vdots \\ \delta_s \\ \delta_s \\ \vdots \\ \delta_s \end{bmatrix}$$

Minimizing the quadratic function $\tilde{e}_{2w}$ in Eq. (34) subject to linear constraints in Eq. (35) is known as a *quadratic programming* (QP) problem. This QP problem can be solved using efficient numerical optimization algorithms; see Fletcher (8).

Figure 11 shows the magnitude response of a low-pass FIR filter of length 41 designed by this weighted least-squares minimization method with parameters $\omega_p = 0.45\pi$, $\omega_s = 0.55\pi$, $w = 1000$, $\delta_p = 0.003$, $\delta_s = 0.03$, $n_p = 30$, and $n_s = 5$. Notice that the large weight w, in conjunction with the constraints in Eqs. (32) and (33) leads to a peak-constrained least-squares stopband and a nearly equiripple $A(\omega)$ in passband.

Design Based on Weighted Chebyshev Approximation

FIR digital filters with equiripple passbands and stopbands can be designed using efficient optimization methods. One of such methods is the well-known Parks–McClellan algorithm. The algorithm was developed during the seventies and has since found widespread applications. Details of the algorithm are now available in several texts; see, for example, Antoniou (2), Parks and Burrus (3), Jackson (9), and Oppenheim and Schafer (10). In what follows several key elements of the algorithm are illustrated, by considering an equiripple design of a linear-phase low-pass filter of odd length N.

A typical equiripple low-pass filter's amplitude response is shown in Fig. 12. From Eq. (18), the amplitude response is given by

$$A(\omega) = \sum_{k=0}^{(N-1)/2} a(k)\cos k\omega \tag{36}$$

and the weighted Chebyshev design can be formulated as to find parameters $\{a(k),\ 0 \leq k \leq (N-1)/2\}$, such that the weighted Chebyshev error

$$\max_{\omega\in\Omega} |E(\omega)|$$

is minimized, where $\Omega = [0, \omega_p] \cup [\omega_s, \pi]$,

$$\begin{aligned} E(\omega) &= W(\omega)[D(\omega) - A(\omega)] \\ D(\omega) &= \begin{cases} 1 & \text{for } 0 \leq \omega \leq \omega_p \\ 0 & \text{for } \omega_s \leq \omega \leq \pi \end{cases} \\ W(\omega) &= \begin{cases} 1 & \text{for } 0 \leq \omega \leq \omega_p \\ \delta/\delta_s & \text{for } \omega_s \leq \omega \leq \pi \end{cases} \end{aligned} \tag{37}$$

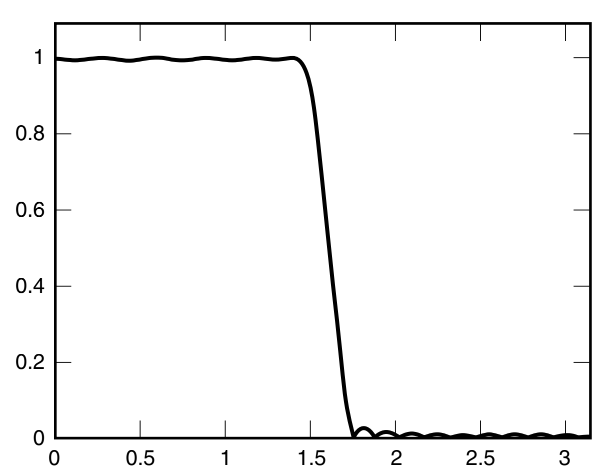


Figure 11. Magnitude response of an FIR filter with $N = 41$ designed by weighted least-squares optimization. Equiripple passband and peak-constrained least-squares stopband are achieved by using a large weight w and imposing adequate number of constraints on the magnitude response in the passband and stopband.

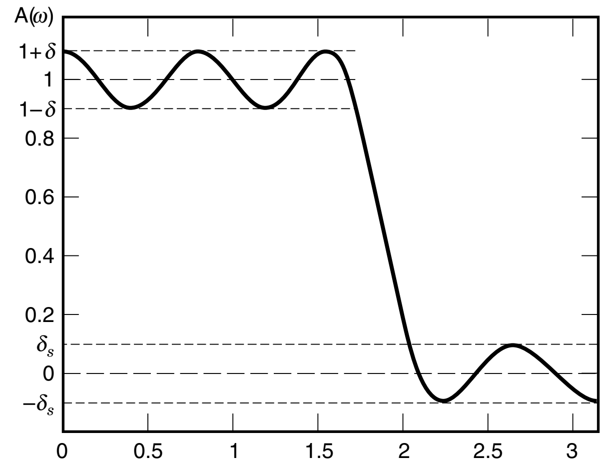


Figure 12. Amplitude response of a typical equiripple low-pass filter. Equiripple response is achieved in both passband and stopband.

Since the design is formulated as to minimize the maximum of $|E(\omega)|$, the present problem is often referred to as a *minimax* design. Evidently, if a linear-phase FIR filter with equiripple amplitude response, as shown in Fig. 12, is designed, then the weighting function defined above implies that the weighted error satisfies

$$|E(\omega)| \leq \delta \qquad \text{for } \omega \in \Omega \tag{38}$$

In other words, the ripples of the weighted error $E(\omega)$ in the passband and stopband are equal.

There are two key elements in the development of the Parks–McClellan algorithm. The first element is an application of the Alternation Theorem from approximation theory which reduces the minimax optimization problem to the problem of finding $(N + 3)/2$ extremal frequencies on Ω, and the second element is an application of an iterative algorithm known as the *Remez exchange algorithm* to efficiently find these extremal frequencies.

The Alternation Theorem states that $A(\omega)$ in Eq. (36) is the unique, best weighted Chebyshev approximation of a continuous function $D(\omega)$ on a compact set Ω on $[0, \pi]$, if and only if the weighted error function $E(\omega)$ in Eq. (37) exhibits at least $(N + 3)/2$ extremal frequencies in Ω, say $\omega_1 < \omega_2 < \cdots < \omega_K$, with $K = (N + 3)/2$, such that

$$E(\omega_i) = -E(\omega_{i+1}) \qquad \text{for } i = 1, \ldots, K-1 \tag{39}$$

and

$$|E(\omega_i)| = \max_{\omega\in\Omega} |E(\omega)| \qquad \text{for } i = 1, \ldots, K \tag{40}$$

In other words, the alternation theorem implies that the amplitude response of a low-pass FIR filter obtained from the above minimax design must possess an equiripple passband and an equiripple stopband, like the one depicted in Fig. 12. In addition, the filter length N determines a lower bound for the number of extremal frequencies of the filter. To explain why it so happens, recall that function cos $k\omega$ can be expressed as a polynomial in cos ω:

$$\cos 2\omega = 2\cos^2\omega - 1$$
$$\cos 3\omega = 4\cos^3\omega - 3\cos\omega$$
$$\cos 4\omega = 8\cos^4\omega - 8\cos^2\omega + 1$$
$$\vdots$$

These polynomials are known as the Chebyshev polynomials. It follows that the amplitude response $A(\omega)$ in Eq. (36) can be expressed as $A(\omega) = P_{(N-1)/2}(x)$, where $P_{(N-1)/2}(x)$ is a polynomial of order $(N - 1)/2$ in x with $x = \cos\omega$. Note that

$$\frac{dA(\omega)}{d\omega} = \frac{dA(\omega)}{dx}\frac{dx}{d\omega} = P'_{(N-1)/2}(x)\cdot\sin\omega$$

where $P'_{(N-1)/2}(x)$ is a polynomial of order $(N - 3)/2$, who has $(N - 3)/2$ zeros. Since $x = \cos\omega$ is periodic, there might be more than $(N - 3)/2$ frequencies in Ω at which $P'_{(N-1)/2} = 0$. In addition, $\sin\omega = 0$ at $\omega = 0$, hence $A'(\omega)$ has at least $(N - 1)/2$ zeros, corresponding to $(N - 1)/2$ extremal frequencies for $A(\omega)$. This, plus two additional extremal frequencies at $\omega = \omega_p$ and $\omega = \omega_s$ (see Fig. 12), implies that there are at least $(N + 3)/2$ extremal frequencies in Ω. If one denotes the $K = (N + 3)/2$ extremal frequencies of an equiripple FIR filter by $\omega_1 < \omega_2 < \cdots < \omega_K$, then, by Eqs. (38)–(40), one has a linear system of K equations

$$\begin{bmatrix} 1 & \cos\omega_1 & \cos 2\omega_1 & \cdots & \cos\frac{N-1}{2}\omega_1 & 1/W(\omega_1) \\ 1 & \cos\omega_2 & \cos 2\omega_2 & \cdots & \cos\frac{N-1}{2}\omega_2 & -1/W(\omega_2) \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos\omega_K & \cos 2\omega_K & \cdots & \cos\frac{N-1}{2}\omega_K & (-1)^{K-1}/W(\omega_K) \end{bmatrix} \cdot \begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ a\left(\frac{N-1}{2}\right) \\ \delta \end{bmatrix} = \begin{bmatrix} D(\omega_1) \\ D(\omega_2) \\ \vdots \\ D(\omega_K) \end{bmatrix} \tag{41}$$

It is interesting to note that the $K \times K$ coefficient matrix in Eq. (41) is always nonsingular as long as the K extremal frequencies are distinct; see Cheney (11). Therefore, if these frequencies were known, the filter parameters $\{a(k), 0 \le k \le N - 1\}$, as well as the smallest ripple it can achieve, could be determined by solving (41). Thus the design problem, at this point, has been reduced to the problem of finding these K extremal frequencies.

The second key element in the Parks–McClellan algorithm involves an application of the Remez exchange algorithm. In an iterative manner, the Remez algorithm performs a search on Ω, for K extremal frequencies at which the weighted Chebyshev error reach the maximum. It then follows from the alternation theorem that the coefficients of the equiripple filter can be found by solving Eq. (41). In the Parks–McClellan algorithm, Ω is a set of equally spaced grid points with a density approximately equal to $10\cdot N$. The Remez algorithm starts with a trial set of frequencies $\hat{\omega}_1 < \hat{\omega}_2 < \cdots < \hat{\omega}_K$, with each $\hat{\omega}_i \in \Omega$. Since the system in Eq. (41) is always nonsingular for distinct $\hat{\omega}_i$'s, (41) can be solved to obtain a $\delta = \delta_0$. At this point one obtains a FIR filter with error $E(\omega)$ oscillates with amplitude δ_0 on the trial set of frequencies. Next, evaluate $A(\omega)$ of the filter obtained on the entire grid set Ω. If $\max_{\omega\in\Omega}|E(\omega)| = \delta_0$, stop and claim the filter as the desired equiripple filter. Otherwise, identify K frequencies from Ω, at which the error $E(\omega)$ attains its maximum magnitude as the new trial set $\{\hat{\omega}_i, 1 \le i \le K\}$. Eq. (41), associated with the new trial set of frequencies, can then be solved to obtain a $\delta = \delta_1$. The iteration continues until a set of frequencies $\{\omega_i, 1 \le i \le K\} \subset \Omega$ is found, such that the corresponding δ obtained from Eq. (41) is equal to $\max_{\omega\in\Omega}|E(\omega)|$. With this set of extremal frequencies, the parameters $\{a(k), 0 \le k \le N - 1/2\}$, determined by Eq. (41), give the minimax design. Several empirical formulas are available in the literature to predict the length N of a low-pass FIR filter that would satisfy prescribed design specifications: ω_p—the normalized passband edge (so that the sampling frequency ω^* corresponds to 2π), ω_s—the normalized stopband edge, δ_p—the passband ripple, and δ_s—the stopband ripple. Kaiser proposed a prediction of N as the smallest odd integer, satisfying

$$N \ge \frac{-20\log_{10}\sqrt{\delta_p\delta_s} - 13}{2.3237(\omega_s - \omega_p)} + 1 \tag{42}$$

If the bandwidth of the filter is neither extremely narrow nor extremely wide, Eq. (42) often gives a good initial value for N. One can then design filters for decreasing or increasing values of N until the lowest value of N meeting the design specifications is obtained. Detailed discussion on this matter can be found in Antoniou (2) and Parks and Burrus (3).

As an example, consider designing an equiripple filter with $\omega_p = 0.45\pi$, $\omega_s = 0.55\pi$, and $\delta_p = \delta_s = 0.008$. Eq. (42) predicts $N = 41$. As the ripples of the filter designed with $N = 41$ slightly exceed 0.008, $N = 43$ was tried and the resulting design was found satisfactory. The amplitude response of the equiripple filter with $N = 43$ is shown in Fig. 13.

DESIGN OF HALF-BAND AND *M*TH BAND FILTERS

*M*th band filters, also known as *Nyquist filters,* are a class of FIR filters with one of the M-branch polyphase decomposition components, say $H_k(z)$, being a constant. For example, the impulse response of the Mth band filter for $k = 0$ is characterized by

$$h(Mn) = \begin{cases} 1/M & n = 0 \\ 0 & n \ne 0 \end{cases} \tag{43}$$

The Mth band filters have proved useful in decimator and interpolator design, as well as in the design of quadrature mirror-image filter banks (6).

An important subclass of Mth band filters is the case $M = 2$, whose transfer functions are characterized by

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

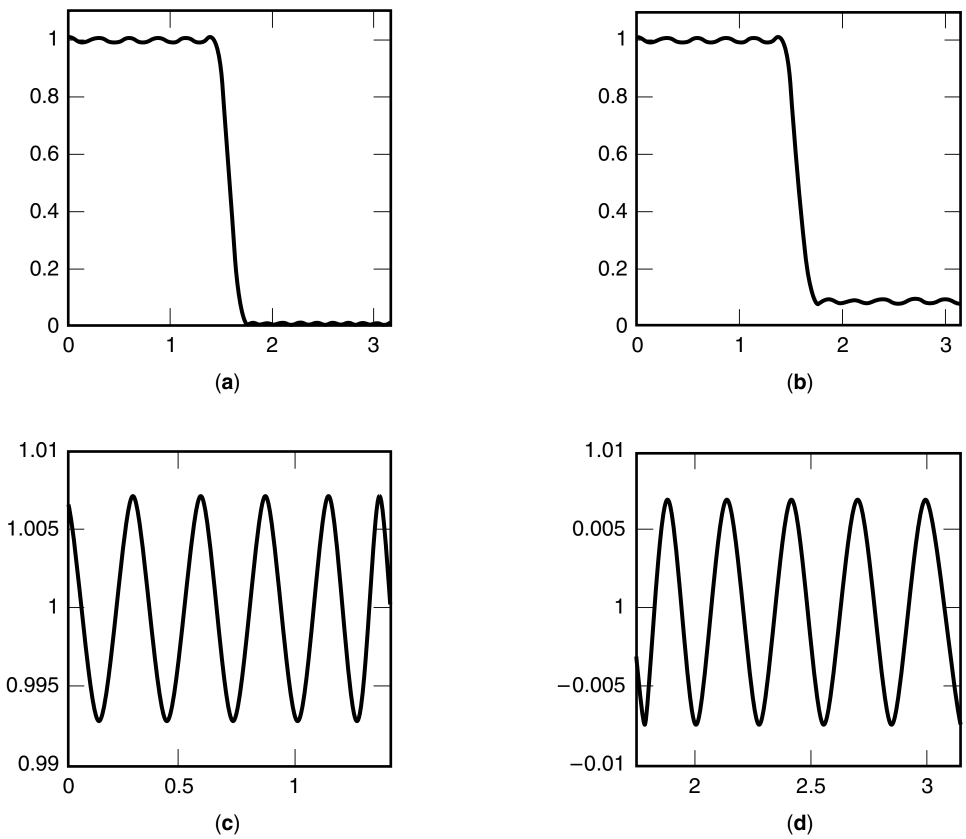


Figure 13. Low-pass FIR filter with N = 41 designed using the Parks–McClellan algorithm. Smallest possible passband and stopband ripples are achieved for given filter length and passband and stopband edges. (a) magnitude response; (b) amplitude response; (c) amplitude response in the passband; (d) amplitude response in the stopband.

which implies that

$$H(z) + H(-z) = 1$$

It can readily be verified that the frequency response satisfying the above equation exhibits odd symmetry with respect to the half-band frequency $\omega = \pi/2$. For this reason they are usually called *half-band filters.*

The design method based on Fourier series and window functions addressed earlier turns out to be a natural choice for the design of lowpass Mth band filters (4). As a matter of fact, for a lowpass, linear-phase, Mth-Band FIR filter with a cutoff frequency at $\omega_c = \pi/M$, the "desired" impulse response $h_d(n)$ in Eq. (28) becomes

$$h_d(n) = \frac{\sin(n\pi/M)}{n\pi} \tag{44}$$

which obviously satisfies Eq. (43). Therefore, an Mth band filter can be obtained as $H(z)$ in Eq. (26) with $h_w(n)$ given by Eqs. (25) and (44). Figure 14 depicts the amplitude response of a lowpass, linear phase, 8th-band FIR filter with $N = 71$. The design utilizes a Kaiser window with $\beta = 6$. The reader is referred to Chapter 4 of (12) and (4) for a complete treatment of Mth band and half-band filters.

Computer Programs and Software for the Design of FIR Digital Filters

Computer programs that implement various algorithms for the design of FIR digital filters have been available from research literature, textbooks, as well as commercial software products. FORTRAN codes of many design algorithms developed during the seventies and eighties can be found in Parks and Burrus (9) and (13). The second edition of the text by Antoniou (2) is now accompanied by a software package called D-FILTER developed by the same author that can be used in Windows95 (Microsoft) environment for the analysis and design of digital filters using the methods studied in the text. These include the window methods and the Remez methods for the FIR filters.

Commercial software for filter designs has been around since early eighties, in which design programs are usually embedded into software packages of fairly large size, which can perform a broad spectrum of DSP tasks. As new versions of the software emerge, design codes are often improved and enhanced, and additional programs that imlement new design

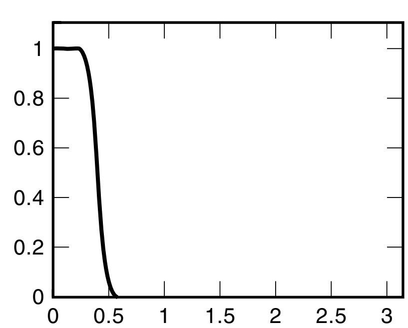


Figure 14. Amplitude response of a low-pass, linear phase, 8th-band FIR filter of length 71. The filter is designed using the window method with a Kaiser window (β = 6).

Table 1. MATLAB Commands for Generating Window Functions

Command	Functionality
bartlett	Bartlett window
blackman	Blackman window
boxcar	Rectangular window
chebwin	Chebyshev window
hamming	Hamming window
hanning	Hanning window
kaiser	Kaiser window
triang	Triangular window

algorithms are included. It is for this reason that commercial software of good quality has been utilized, by an increasing number of professionals, to perform day-to-day analysis/design work, as well as to conduct scientific and engineering research. One of the most noticeable software products for technical computing including filter designs is MATLAB from The MathWorks, Inc. As an easy-to-use high-level programming language for scientific computing, MATLAB provides a set of core commands for performing general numerical operations and connects itself to a number of toolboxes. Each toolbox offers a fairly large set of additional MATLAB commands and functions, which are especially suited for performing computations often needed in a specific technical field. The current version of the Signal Processing Toolbox (The MathWorks, Inc.) contains more than 135 commands that can be used with MATLAB to perform DSP-related computing. The toolbox provides the user with a variety of functions that are of immediate use for designing FIR filters. Tables 1 and 2 list sample MATLAB commands for generating window functions and designing window-based, least-squares, and equiripple FIR filters, respectively; see (14) for detailed instructions of these and other related MATLAB commands. It is worthwhile to note that commands useful in filter design can also be found in other MATLAB toolboxes. The quadratic programming problem formulated by Eqs. (34) and (35), for example, can readily be solved using the MATLAB function named qp, which is available from the Optimization Toolbox (The MathWorks, Inc.). The function requires four inputs, namely **Q**, **b**, **C**, and **d**, defined in Eqs. (34) and (35), and its output is a vector **x** that minimizes $\tilde{e}_{2w}$ in Eq. (34) subject to the constraints in Eq. (35).

Table 2. MATLAB Commands for FIR Filter Design

Command	Functionality
cremez	Complex and nonlinear phase equiripple FIR filter design
fir1	Window-based FIR filter design—low, high, band, stop, multipass
fir2	Window-based FIR filter design—arbitrary response
fircls	Constrained least-squares filter design—arbitrary response
fircls1	Constrained least-squares FIR filter design—low- and high-pass
firls	FIR filter design—arbitrary response with transition bands
firrcos	Raise cosine FIR filter design
intfilt	Interpolation FIR filter design
kaiserord	Window-based filter order selection using Kaiser window
remez	Parks–McClellan optimal FIR filter design
remezord	Parks–McClellan filter order selection

QUANTIZATION EFFECTS

Performing digital filtering on a digital computer might result in outcomes that deviate from the desired ones. In most circumstances, the deviation is primarily due to the very fact that numbers in a digital computer are represented, and arithmetic maniulations of these numbers are carried out in finite precision. Specifically, sampled continuous-time input signals, filter coefficients, and outputs of multipliers all need to be rounded off or truncated to finite number of digits. In other words, the effects of finite word length in FIR digital filters are mainly shown in terms of coefficient-quantization errors, product-quantization errors, and input-quantization errors.

Coefficient Quantization

Obviously, coefficient quantization has a direct impact on the quantized transfer function. For FIR filters with transfer function given by Eq. (10), the frequency response with quantized coefficients $\{\hat{h}(n),\ 0 \le n \le N-1\}$ is

$$\hat{H}(\omega) = \sum_{n=0}^{N-1} \hat{h}(n) e^{-jn\omega}$$

which deviates from the unquantized frequency response, $H(\omega)$, by

$$\begin{aligned} |\hat{H}(\omega) - H(\omega)| &= \left| \sum_{n=0}^{N-1} [\hat{h}(n) - h(n)] e^{-jn\omega} \right| \\ &\le N \cdot \max_n |\hat{h}(n) - h(n)| \end{aligned} \tag{45}$$

It is known that, for signed-magnitude, as well as one's or two's-complement numbers,

$$|\hat{h}(n) - h(n)| \le 2^{-L} \tag{46}$$

where L is the word length used in binary bits; see Antoniou (2). If $H(\omega)$ was designed to have ripples δ_p and δ_s in the passband and stopband, respectively, then Eqs. (45) and (46) imply that for ω in the passband

$$\begin{aligned} |\hat{H}(\omega) - 1| &\le |\hat{H}(\omega) - H(\omega) + H(\omega) - 1| \\ &\le |\hat{H}(\omega) - H(\omega)| + |H(\omega) - 1| \\ &\le N \cdot 2^{-L} + \delta_p \end{aligned}$$

and for ω in the stopband

$$\begin{aligned} |\hat{H}(\omega)| &= |\hat{H}(\omega) - H(\omega) + H(\omega)| \\ &\le |\hat{H}(\omega) - H(\omega)| + \delta_s \\ &\le N \cdot 2^{-L} + \delta_s \end{aligned}$$

We see that the coefficient quantization may cause both the passband and stopband ripples to increase by $N \cdot 2^{-L}$, at the

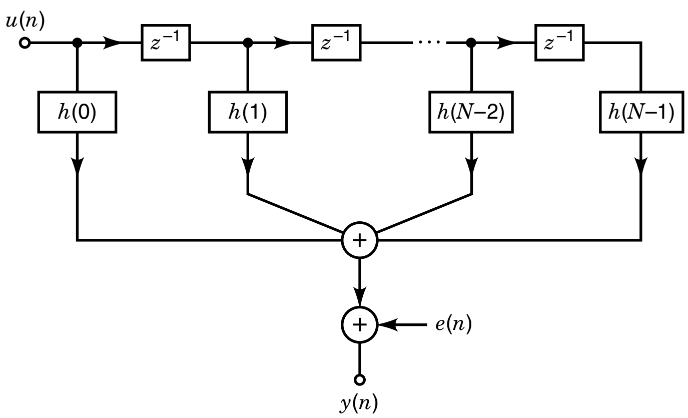


Figure 15. Direct implementation of an FIR filter with roundoff noise. Note that the roundoff noise, which can be treated as zero-mean white noise, occurs only at the output.

worst. This performance degradation can be kept as insignificant as desired by using a sufficiently large value of L. For example, if numbers are represented with 16 binary bits, then with $N = 100$ the increase of the ripples due to coefficient quantization would be no larger than 0.001526.

Roundoff Noise

The term *roundoff noise* is referred to as the accumulated errors due to rounding or truncation of multiplication products in the filter. For FIR filters, the most frequently employed implementation scheme is the direct implementation illustrated in Fig. 15. If the multipliers that generate products $h(k)u(n-k)$ for $0 \le k \le N-1$ are operated in double precision, then no product-quantization errors will be introduced. If this is not the case, roundoff noise will occur. As shown in Fig. 15, however, it occurs only at the output. Let $\hat{y}(n)$ and $y(n)$ be the quantized output and the output without signal quantization, respectively, then we have

$$e(n) = \hat{y}(n) - y(n)$$

where $\{e(n)\}$ can be treated as a zero-mean white noise with variance $\sigma_n^2 = 2^{-2L}/3$, L = the word length used in binary bits; see Parks and Burrus (3) for further details.

Network Scaling

One of the important factors that need to be considered when implementing a digital filter is to select appropriate scaling strategies for improved signal-to-noise (SNR) ratio without signal overflow. As the roundoff noise level is independent of signal levels within the filter, apparently a simple way to increase the SNR would be to increase the signal levels. However, this may result in overflow at some nodes in the filter if the available dynamic range of the fixed-point arithmetic is not sufficiently wide. For FIR filters, simple yet effective scaling techniques are available.

Let the magnitude of the input signal $\{u(n)\}$ be bounded by M and assume M is the dynamic range for both input $u(n)$ and output $y(n)$. It follows from Eq. (3) that

$$|y(n)| \le \sum_{k=0}^{N-1} |h(k)u(n-k)| \le M \sum_{k=0}^{N-1} |h(k)| \tag{47}$$

Eq. (47) implies that $y(n)$ would be kept within the available dynamic range M, if

$$\sum_{k=0}^{N-1} |h(k)| \le 1$$

Given an FIR filter

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

we define the 1-norm of its impulse response as

$$\|\mathbf{h}\|_1 = \sum_{k=0}^{N-1} |h(k)|$$

This $\|\mathbf{h}\|_1$ can be used as a scaling factor to modify the filter $H(z)$ to $\tilde{H}(z)$:

$$\tilde{H}(z) = \frac{H(z)}{\|\mathbf{h}\|_1} = \sum_{k=0}^{N-1} \tilde{h}(k) z^{-k}$$

with $\tilde{h}(k) = h(k)/\|\mathbf{h}\|_1$. Evidently, the impulse response of $\tilde{H}(z)$ satisfies

$$\sum_{k=0}^{N-1} |\tilde{h}(k)| = \frac{1}{\|\mathbf{h}\|_1} \sum_{k=0}^{N-1} |h(k)| = 1$$

Therefore, no overflow would occur if $\tilde{H}(z)$ (instead of $H(z)$) is utilized. The reader is referred to Jackson (9) and Parks and Burrus (3) for further discussion on the scaling issue.

BIBLIOGRAPHY

1. G. Strang, *Introduction to Linear Algebra,* Wellesley, MA: Wellesley-Cambridge Press, 1993.
2. A. Antoniou, *Digital Filters: Analysis, Design, and Applications,* 2nd ed., New York: McGraw-Hill, 1993.
3. T. W. Parks and C. S. Burrus, *Digital Filter Design,* New York: Wiley, 1987.
4. S. K. Mitra, *Digital Signal Processing—A Computer Based Approach,* New York: McGraw-Hill, 1998.
5. M. Bellanger, Digital filtering by polyphase network: Application to sample rate alteration and filter banks, *IEEE Trans. Acoust., Speech, Signal Process.,* **24**: 109–114, April 1976.
6. P. P. Vaidyanathan, *Multirate Systems and Filter Banks,* Englewood Cliffs, NJ: Prentice-Hall, 1993.
7. J. W. Adams, FIR digital filters with least-squares stopbands subject to peak-gain constraints, *IEEE Trans. Circuits Syst.,* **38**: 376–388, 1991.
8. R. Fletcher, *Practical Optimization Methods,* 2nd ed., New York: Wiley, 1987.
9. L. B. Jackson, *Digital Filters and Signal Processing,* 3rd ed., Boston: Kluwer Academic, 1996.
10. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1989.
11. E. W. Cheney, *Introduction to Approximation Theory,* New York: McGraw-Hill, 1966.
12. S. K. Mitra and J. F. Kaiser (ed.), *Handbook for Digital Signal Processing,* New York: Wiley, 1993.

13. *Programs for Digital Signal Processing,* New York: IEEE Press, 1979.
14. *User's Guide for MATLAB Signal Processing Toolbox,* Version 4, Natick, MA: The MathWorks, Inc., 1997.

W.-S. Lu
University of Victoria

FIRST-ORDER CIRCUITS. See Transient analysis.