CHAPTER 12

Introduction to the Design of Discrete Filters

CONTENTS

When in doubt,
don't.

Benjamin Franklin (1706–1790),
printer, inventor, scientist and diplomat.

12.1 INTRODUCTION

Filtering is an important application of linear time-invariant (LTI) systems. According to the eigenfunction property of discrete-time LTI systems, the steady-state response of a discrete-time LTI system to a sinusoidal input is also a sinusoid of the same frequency as that of the input, but with magnitude and phase affected by the response of the system at the frequency of the input. Since periodic as well as aperiodic signals have Fourier representations consisting of sinusoids of different frequencies, these signal components can be modified by appropriately choosing the frequency response of a LTI system, or filter. Filtering can thus be seen as a way to change the frequency content of an input signal.

The appropriate filter is specified using the spectral characterization of the input and the desired spectral characteristics of the output of the filter. Once the specifications of the filter are set, the problem becomes one of approximation, either by a ratio of polynomials or by a polynomial (if possible). After establishing that the filter resulting from the approximation satisfies the given specifications, it is then necessary to check its stability (if not guaranteed by the design method)—in the case of the filter being a rational approximation—and if stable, we need to figure out what would be the best possible way to implement the filter in hardware or in software. If not stable, we need either to repeat the approximation or to stabilize the filter before its implementation.

In the continuous-time domain, filters are obtained by means of rational approximation. In the discrete-time domain, there are two possible types of filters: one that is the result of rational approximation—these filters are called recursive or infinite impulse response (IIR) filter. The other is the non-recursive

Signals and Systems Using MATLAB©, Second Edition. http://dx.doi.org/10.1016/B978-0-12-394812-0.00012-7

or finite impulse response (FIR) filter that results from polynomial approximation. As we will see, the discrete filter specifications can be in the frequency or in the time domain. For recursive or IIR filters, the specifications are typically given in the form of magnitude and phase specifications, while the specifications for non-recursive or FIR filters can be in the time domain as a desired impulse response. The discrete filter design problem then consists in: Given the specifications of a filter we look for a polynomial or rational (ratio of polynomials) approximation to the specifications. The resulting filter should be realizable, which besides causality and stability requires that the filter coefficients be real-valued.

There are different ways to attain a rational approximation for discrete IIR filters: by transformation of analog filters, or by optimization methods that include stability as a constraint. We will see that the classical analog design methods (Butterworth, Chebyshev, Elliptic, etc.) can be used to design discrete filters by means of the bilinear transformation that maps the analog s-plane into the Z-plane. Given that the FIR filters are unique to the discrete domain, the approximation procedures for FIR filters are unique to that domain.

The difference between discrete and digital filters is in quantization and coding. For a discrete filter we assume that the input and the coefficients of the filter are represented with infinite precision, i.e., using an infinite number of quantization levels and thus no coding is performed. The coefficients of a digital filter are binary, and the input is quantized and coded. Quantization thus affects the performance of a digital filter, while it has no effect in discrete filters.

Considering continuous to discrete (C/D) and discrete to continuous (D/C) ideal converters simply as samplers and reconstruction filters, theoretically it is possible to implement the filtering of band-limited analog signals using discrete filters (Figure 12.1). In such an application, an additional specification for the filter design is the sampling period. In this process it is crucial that the sampling period in the C/D and D/C converters be synchronized. In practice, filtering of analog signals is done using analog to digital (A/D) and digital to analog (D/A) converters together with digital filters.

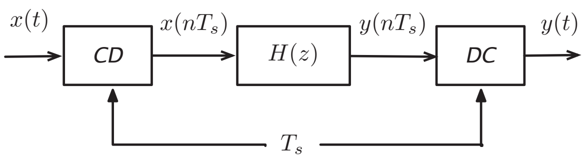

FIGURE 12.1
Discrete filtering of analog signals using ideal continuous to discrete (C/D), or a sampler, and discrete to continuous (D/C) converter, or a reconstruction filter.

12.2 FREQUENCY SELECTIVE DISCRETE FILTERS

The principle behind discrete filtering is easily understood by considering the response of a linear time-invariant (LTI) system to sinusoids. If $H(z)$ is the transfer function of a discrete-time LTI system, and the input is

$$x[n] = \sum_k A_k \cos(\omega_k n + \phi_k)$$

(if the input is periodic the frequencies $\{\omega_k\}$ are harmonically related, otherwise they are not). According to the eigenfunction property of LTI systems the steady-state response of the system is

$$y_{ss}[n] = \sum_k A_k |H(e^{j\omega_k})| \cos(\omega_k n + \phi_k + \theta(\omega_k))$$

where $|H(e^{j\omega_k})|$ and $\theta(\omega_k)$ are the magnitude and the phase of $H(e^{j\omega})$, the frequency response of the system, at the discrete frequency ω_k. The frequency response is the transfer function computed on the unit circle, i.e., $H(e^{j\omega}) = H(z)|_{z=e^{j\omega}}$. It becomes clear from the above equation that by judiciously choosing the frequency response of the LTI system we can select the frequency components of the input we wish to have at the output, and as such attenuate or amplify their amplitudes or change their phases. In general, for an input $x[n]$ with Z-transform $X(z)$, the Z-transform of the output of the filter is

$$\begin{aligned} Y(z) &= H(z)X(z) \quad \text{or on the unit circle when } z = e^{j\omega} \\ Y(e^{j\omega}) &= H(e^{j\omega})X(e^{j\omega}) \end{aligned}$$

By selecting the frequency response $H(e^{j\omega})$ we allow some frequency components of $x[n]$ to appear in the output, and others to be filtered out. Although ideal frequency selective filters such as low-pass, high-pass, band-pass, and stop-band—selecting low, high, and middle frequency components—cannot be realized, they serve as prototypes for the actual filters.

12.2.1 Phase Distortion

A filter changes the spectrum of its input in magnitude as well as in phase. Distortion in magnitude can be avoided by using an all-pass filter with unit magnitude response for all frequencies. Phase distortion can be avoided by requiring the phase response of the filter to be linear.

For instance, when transmitting a voice signal in a communication system it is important that the signals at the transmitter and at the receiver be equal within a time delay and a constant attenuation factor. To achieve this, the transfer function of an ideal communication channel should be an all-pass filter with a linear phase. Indeed, if the output of an ideal discrete transmitter is a signal $x[n]$ and the recovered signal at an ideal discrete receiver is $\alpha x[n - N_0]$, for an attenuation factor α and a time delay N_0, the channel is represented by the transfer function of an all-pass filter:

$$H(z) = \frac{\mathcal{Z}(\alpha x[n - N_0])}{\mathcal{Z}(x[n])} = \alpha z^{-N_0}$$

The constant gain of the all-pass filter permits all frequency components of the input to appear in the output. As we will see, the linear phase simply delays the signal, which is a very tolerable distortion that can be reversed.

To appreciate the effect of linear phase, consider the filtering of a signal

$$x[n] = 1 + \cos(\omega_0 n) + \cos(\omega_1 n) \quad \omega_1 = 2\omega_0 \quad n \geq 0$$

using an all-pass filter with transfer function $H(z) = \alpha z^{-N_0}$. The magnitude response of this filter is α, for all frequencies, and its phase is linear as shown in Figure 12.2(a). The steady-state output of the all-pass filter is

$$\begin{aligned} y_{ss}[n] &= 1H(e^{j0}) + |H(e^{j\omega_0})|\cos(\omega_0 n + \angle H(e^{j\omega_0})) + |H(e^{j\omega_1})|\cos(\omega_1 n + \angle H(e^{j\omega_1})) \\ &= \alpha\,[1 + \cos(\omega_o(n - N_0)) + \cos(\omega_1(n - N_0))] = \alpha x[n - N_0] \end{aligned}$$

which is the input signal attenuated by α and delayed N_0 samples.

Suppose then that the all-pass filter has a phase function which is non-linear, for instance the one in Figure 12.2(b). The steady-state output would then be

$$\begin{aligned} y_{ss}[n] &= 1H(e^{j0}) + |H(e^{j\omega_0})|\cos(\omega_0 n + \angle H(e^{j\omega_0}) + |H(e^{j\omega_1})|\cos(\omega_1 n + \angle H(e^{j\omega_1})) \\ &= \alpha[1 + \cos(\omega_0(n - N_0)) + \cos(\omega_1(n - 0.5N_0))] \neq \alpha x[n - N_0] \end{aligned}$$

In the case of the linear phase each of the frequency components of $x[n]$ is delayed N_0 samples, and thus the output is just a delayed version of the input. On the other hand, in the case of the non-linear phase the frequency component of frequency ω_1 is delayed less than the other two frequency components creating distortion in the signal so that the output is not a delayed version of the input and the phase effect cannot be reversed.

12.2.1.1 Group Delay

A measure of linearity of the phase $\theta(\omega)$ of a LTI system is obtained from the **group delay function** which is defined as:

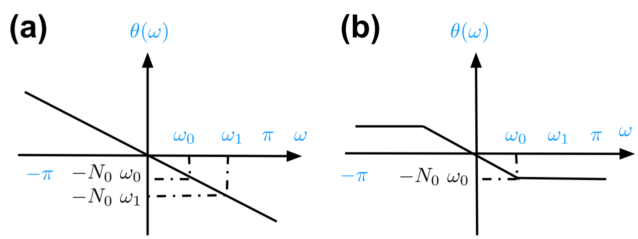

FIGURE 12.2
(a) Linear, (b) non-linear phase.

$$\tau(\omega) = -\frac{d\theta(\omega)}{d\omega} \tag{12.1}$$

The group delay is constant when the phase is linear. Deviation of the group delay from a constant indicates the non-linearity of the phase.

In the above cases, when the phase is linear, i.e., for $0 \leq \omega \leq \pi$

$$\theta(\omega) = -N_0\omega \Rightarrow \tau(\omega) = N_0$$

and when the phase is non-linear

$$\theta(\omega) = \begin{cases} -N_0\omega & 0 < \omega \leq \omega_0 \\ -N_0\omega_0 & \omega_0 < \omega \leq \pi \end{cases}$$

then we have that the group delay is

$$\tau(\omega) = \begin{cases} N_0 & 0 < \omega \leq \omega_0 \\ 0 & \omega_0 < \omega \leq \pi \end{cases}$$

which is not constant.

Remarks

1. *Integrating (12.1) when $\tau(\omega)$ is a constant τ gives a general expression for the linear phase: $\theta(\omega) = -\tau\omega + \theta_0$. If $\theta_0 = 0$ the phase, as a function of ω, is a line through the origin with slope $-\tau$. Otherwise, as an odd function of ω the phase is*

$$\theta(\omega) = \begin{cases} -\tau\omega - \theta_0 & -\pi \leq \omega < 0 \\ 0 & \omega = 0 \\ -\tau\omega + \theta_0 & 0 < \omega \leq \pi \end{cases}$$

i.e., it has a discontinuity at $\omega = 0$ but it is still considered linear.

2. *The group delay τ of a linear-phase system is not necessarily an integer. Suppose we have an ideal analog low-pass filter with frequency response*

$$H(j\Omega) = [u(\Omega + \Omega_0) - u(\Omega - \Omega_0)]e^{-j\zeta\Omega}, \quad \zeta > 0$$

Sampling the impulse response $h(t)$ of this filter using a sampling frequency $\Omega_s = 2\Omega_0$ gives an all-pass discrete filter with frequency response

$$H(e^{j\omega}) = 1e^{-j\zeta\omega/T_s} \quad -\pi < \omega \leq \pi$$

where the group delay ζ/T_s can be an integer or a real positive value.

3. *Notice in the above example that when the phase is linear the group delay corresponds to the time delay in the output signal. This is due to the phase delay being proportional to the frequency of the signal component. When this proportionality is not present, i.e., the phase is not linear, the delay is different for different signal components causing the phase distortion.*

12.2.2 IIR and FIR Discrete Filters

- A discrete filter with transfer function

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{m=0}^{M-1} b_m z^{-m}}{1 + \sum_{k=1}^{N-1} a_k z^{-k}} = \sum_{n=0}^{\infty} h[n] z^{-n} \quad (12.2)$$

is called **infinite impulse response** *or* **IIR** since its impulse response $h[n]$ typically has infinite length. It is also called **recursive** because if the input of the filter $H(z)$ *is* $x[n]$ and $y[n]$ its output, the input/output relationship is given by the following difference equation:

$$y[n] = -\sum_{k=1}^{N-1} a_k y[n-k] + \sum_{m=0}^{M-1} b_m x[n-m] \quad (12.3)$$

where the output recurs on previous outputs (i.e., the output is fed back).

- The transfer function of a **finite impulse response** or **FIR** filter is

$$H(z) = B(z) = \sum_{m=0}^{M-1} b_m z^{-m} \quad (12.4)$$

Its impulse response is $h[n] = b_n, n = 0, \ldots, M-1$, and zero otherwise, thus of finite length. The input/output relationship

$$y[n] = \sum_{m=0}^{M} b_m x[n-m] = (b * x)[n] \quad (12.5)$$

coincides with the convolution sum of the filter coefficients (or impulse response) and the input. This filter is also called **non-recursive** as the output only depends on the input.

For practical reasons, these filters must be causal (i.e., $h[n] = 0$ for $n < 0$) and bounded-input bounded-output (BIBO) stable (i.e., all the poles of $H(z)$ must be *inside the unit circle*). This guarantees that the filter can be implemented and used in real-time processing, and that the output remains bounded when the input is bounded.

Remarks

1. *Calling the IIR filters recursive is more appropriate. Although it is traditional to refer to these filters as IIR, it is possible to have a filter with a rational transfer function that does not have an infinite length impulse response. For instance, consider a filter with transfer function*

$$H(z) = \frac{1}{M} \frac{z^M - 1}{z^{M-1}(z-1)}$$

for some integer $M \geq 1$. This filter appears to be IIR, but if we express its transfer function in negative powers of z we obtain

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

which is an M-order FIR filter with impulse response
$h[n] = 1/M, n = 0, \ldots, M-1.$

2. *When comparing the IIR and the FIR filters, neither has a definite advantage:*
   - *IIR filters are implemented more efficiently than FIR filters in terms of number of operations and required storage (having similar frequency responses, an IIR filter has fewer coefficients than an FIR filter).*
   - *The implementation of an IIR filter using the difference equation, resulting from its transfer function, is simple and computationally efficient, but FIR filters can be implemented using the computationally efficient fast Fourier transform (FFT) algorithm.*
   - *Since the transfer function of any FIR filter only has poles at the origin of the Z-plane, FIR filters are always BIBO stable, but for an IIR filter we need to check that the poles of its transfer function (i.e., zeros of its denominator $A(z)$) be inside the unit circle if the design procedure does not guarantee stability.*
   - *FIR filters can be designed to have linear phase, while IIR filters usually have non-linear phase, but approximately linear phase in the passband region.*

Example 12.1

The phase of IIR filters is always non-linear. Although it is possible to design FIR filters with linear phase, not all FIR filters have linear phase. Consider the following two filters with input/output equations

$$(i)\; y[n] = 0.5y[n-1] + x[n], \quad (ii)\; y[n] = \frac{1}{3}(x[n-1] + x[n] + x[n+1])$$

where $x[n]$ is the input and $y[n]$ the output. Use MATLAB to compute and plot the magnitude and phase response of each of these filters.

Solution

The transfer functions of the given filters are

$$(i)\; H_1(z) = \frac{1}{1 - 0.5z^{-1}}$$

$$(ii)\; H_2(z) = \frac{1}{3}[z^{-1} + 1 + z] = \frac{1 + z + z^2}{3z} = \frac{(z - 1e^{j2.09})(z - 1e^{-j2.09})}{3z}$$

thus the first is an IIR filter and the second an FIR filter (notice that this filter is non-causal as it requires future values of the input to compute the present output). The phase response of the IIR filter is:

$$H_1(e^{j\omega}) = \frac{1}{[1 - 0.5\cos(\omega)] + j0.5\sin(\omega)} \quad \text{then}$$

$$\angle H_1(e^{j\omega}) = -\tan^{-1}\left(\frac{0.5\sin(\omega)}{1 - 0.5\cos(\omega)}\right)$$

which is clearly non-linear. For the FIR filter we have that

$$H_2(e^{j\omega}) = \frac{1}{3}(e^{-j\omega} + 1 + e^{j\omega}) = \frac{1+2\cos(\omega)}{3} \quad \text{then}$$

$$\angle H_2(e^{j\omega}) = \begin{cases} 0 & \text{when } 1+2\cos(\omega) \geq 0 \\ -\pi & \text{when } 1+2\cos(\omega) < 0 \end{cases}$$

Since $H_2(z)$ has zeros at $z = 1e^{\pm j2.09}$, its magnitude response becomes zero at $\omega = \pm 2.09$ rads. The frequency response $H_2(e^{j\omega})$ is real-valued, and for $\omega \in [0, 2.09]$ the response $H_2(e^{j\omega}) \geq 0$ so the phase is zero, while for $\omega \in (2.09, \pi]$ the response $H_2(e^{j\omega}) < 0$ so the phase is π. The phase of the two filters is computed using the MATLAB function *freqz* and together with the corresponding magnitudes are shown in Figure 12.3. ■

■ Example 12.2

A simple model for the multi-path effect in the channel of a wireless system is

$$y[n] = x[n] - \alpha x[n - N_0], \quad \alpha = 0.8, \; N_0 = 11$$

i.e., the output $y[n]$ is a combination of the input $x[n]$, and of a delayed and attenuated version of the input, $\alpha x[n - N_0]$. Determine the transfer function of the filter representing the channel, i.e., that gives the above input/output equation. Use MATLAB to plot its magnitude and phase. If the phase is non-linear,

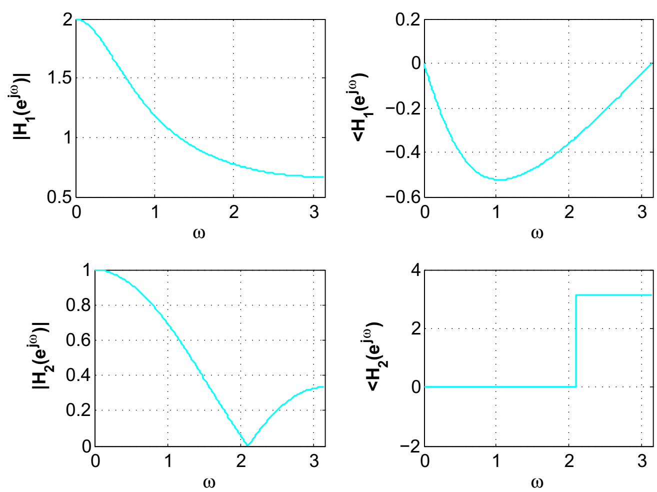

FIGURE 12.3

Magnitude and phase responses of an IIR filter with transfer function $H_1(z) = 1/(1 - 0.5z^{-1})$ (top), and of an FIR filter with transfer function $H_2(z) = (z - 1e^{j2.09})(z - 1e^{-j2.09})/3z$ (bottom). Notice the phase responses are non-linear.

how would you recover the input $x[n]$ (which is the message)? Let the input be $x[n] = 2 + \cos(\pi n/4) + \cos(\pi n)$. In practice, the delay N_0 and the attenuation α are not known at the receiver and need to be estimated. What would happen if the delay is estimated to be 12 and the attenuation 0.79?

Solution

The transfer function of the filter with input $x[n]$ and output $y[n]$ is

$$H(z) = \frac{Y(z)}{X(z)} = 1 - 0.8z^{-11} = \frac{z^{11} - 0.8}{z^{11}}$$

with a pole $z = 0$ of multiplicity 11, and zeros the roots of $z^{11} - 0.8 = 0$ or

$$z_k = (0.8)^{1/11} e^{j2\pi k/11} = 0.9799 e^{j2\pi k/11} \quad k = 0, \ldots, 10$$

Using the *freqz* function to plot its magnitude and phase responses (see Figure 12.4), we find that the phase is non-linear and as such the output of

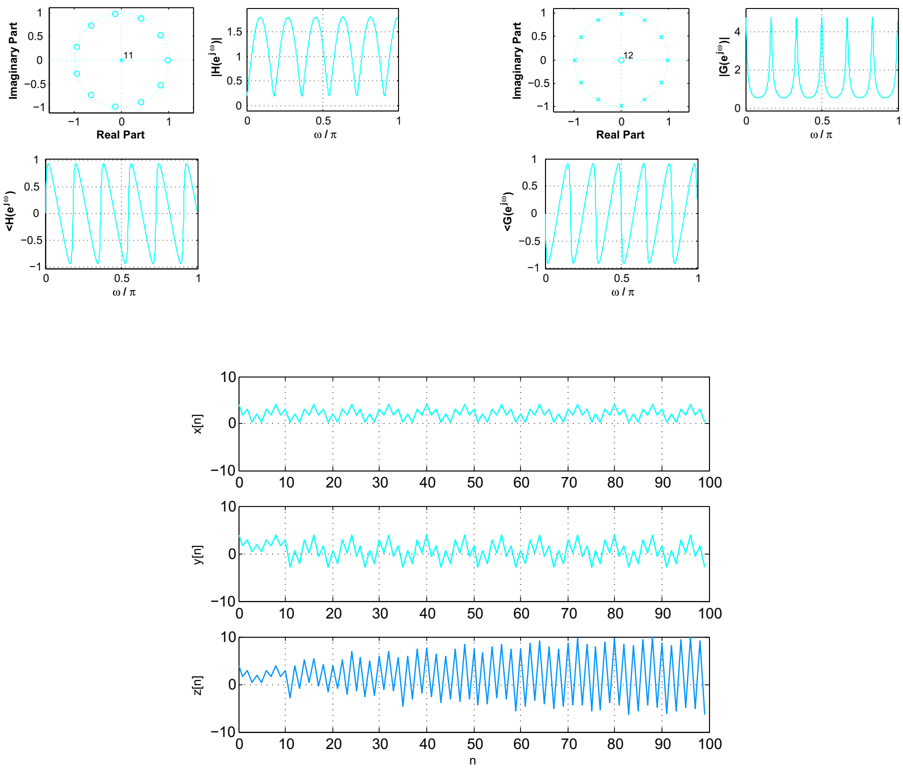

FIGURE 12.4

Poles and zeros, and frequency response of the FIR comb filter $H(z) = (z^{11} - 0.8)/z^{11}$ (top left) and the estimated inverse IIR comb filter $\hat{G}(z) = z^{12}/(z^{12} - 0.79)$ (top right). Bottom plot shows the message $x[n] = 2 + \cos(\pi n/4) + \cos(\pi n)$, the output $y[n]$ of channel $H(z)$ and output $z[n]$ of the estimated inverse filter $\hat{G}(z)$.

$H(z)$, $y[n]$, will not be a delayed version of the input and cannot be recovered by shifting it back. To recover the input, we use an **inverse filter** $G(z)$ such that cascaded with $H(z)$ the overall filter is an all-pass, i.e., $H(z)G(z) = 1$. Thus

$$G(z) = \frac{z^{11}}{z^{11} - 0.8}$$

The poles and zeros, and the magnitude and phase response of $H(z)$ are shown in the top left plot in Figure 12.4. The filters with transfer functions $H(z)$ and $G(z)$ are called **comb filters** given the shape of their magnitude responses.

If the delay is estimated to be 11 and the attenuation 0.8, the input signal $x[n]$ (the message) is recovered exactly. However, if we have slight variations on these values the message might not be recovered. When the delay is estimated to be 12 and the attenuation 0.79, the inverse filter is

$$\hat{G}(z) = \frac{z^{12}}{z^{12} - 0.79}$$

having the poles, zeros and magnitude and phase responses shown in Figure 12.4, top right plot. In the bottom figure in Figure 12.4, the effect of these changes is illustrated. The output of the inverse filter $z[n]$ does not resemble the sent signal $x[n]$. The signal $y[n]$ is the output of the channel with transfer function $H(z)$. ■

12.3 FILTER SPECIFICATIONS

There are two ways to specify a discrete filter: in the frequency domain and in the time domain. Frequency specification of the magnitude and phase for the desired filter are more common in IIR filter design, while the time specification in terms of the desired impulse response of the filter is used in FIR filter design.

12.3.1 Frequency Specifications

Conventionally, when designing IIR filters a prototype low-pass filter is obtained first and then converted into the desired filter using frequency transformations. The magnitude specifications of a discrete low-pass filter are given for frequencies $[0, \pi]$ due to the periodicity and the evenness of the magnitude. Typically, the phase is not specified but it is expected to be approximately linear.

For a low-pass filter, the desired magnitude $|H_d(e^{jw})|$ is to be close to unity in a passband frequency region, and close to zero in a stopband frequency region. To allow a smooth transition from the passband to the stopband a transition region where the filter is not specified is needed. Thus, the magnitude specifications are displayed in Figure 12.5. The **passband** $[0, \omega_p]$ is the band of frequencies for which the attenuation specification is the smallest; the **stopband** $[\omega_{st}, \pi]$ is the band of frequencies where the attenuation specification is the

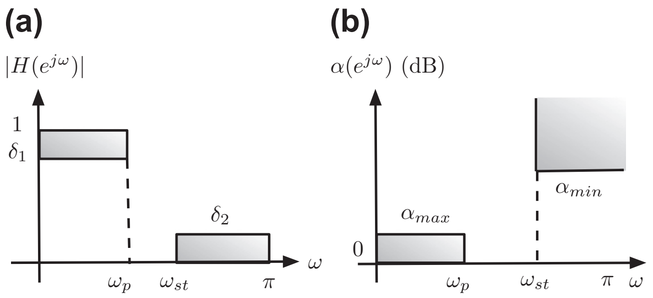

FIGURE 12.5
Low-pass magnitude specifications for an IIR filter: (a) linear scale, (b) loss scale.

greatest; and the **transition band** (ω_p, ω_{st}) is the frequency band where the filter is not specified. The frequencies ω_p, and ω_{st} are called the **passband** and the **stopband** frequencies.

12.3.1.1 Loss Function

As for analog filters, the linear-scale specifications shown in Figure 12.5(a), do not give the sense of the attenuation and thus the loss or log-specification in decibels (dBs) is preferred. This logarithmic scale also provides a greater resolution of the magnitude.

The magnitude specifications of a discrete low-pass filter in a linear scale (Figure 12.5a)) are:

$$\begin{aligned} &\textit{Passband}: \quad \delta_1 \leq |H(e^{j\omega})| \leq 1 \quad && 0 \leq \omega \leq \omega_p \\ &\textit{Stopband}: \quad 0 < |H(e^{j\omega})| \leq \delta_2 \quad && \omega_{st} \leq \omega \leq \pi \end{aligned} \tag{12.6}$$

for $0 < \delta_2 < \delta_1 < 1$.
Defining the **loss function** for a discrete filter as

$$\alpha(e^{j\omega}) = -10\log_{10}|H(e^{j\omega})|^2 = -20\log_{10}|H(e^{j\omega})| \qquad \text{dBs} \tag{12.7}$$

equivalent magnitude specifications for a discrete low-pass filter (Figure 12.5(b)) are:

$$\begin{aligned} &\textit{Passband}: \quad 0 \leq \alpha(e^{j\omega}) \leq \alpha_{max} \quad && 0 \leq \omega \leq \omega_p \\ &\textit{Stopband}: \quad \alpha_{min} \leq \alpha(e^{j\omega}) < \infty \quad && \omega_{st} \leq \omega \leq \pi \end{aligned} \tag{12.8}$$

where $\alpha_{max} = -20\log_{10}\delta_1$ and $\alpha_{min} = -20\log_{10}\delta_2$ (which are positive since both δ_1 and δ_2 are positive and smaller than 1). The frequency specification ω_p is the passband frequency and ω_{st} is the stopband frequency, both in radians.
Increasing the loss 20 dB makes the filter attenuate the input signal by a factor of 10^{-1}.

Remarks

1. *The dB scale is an indicator of attenuation: if we have a unit magnitude the corresponding loss is* 0 *dB and for every* 20 *dB in loss this magnitude is attenuated by* 10^{-1}, *so that when the loss is* 100 *dB the* unit magnitude *would be*

attenuated to 10^{-5}*. The dB scale also has the physiological significance of being a measure of how humans detect levels of sound.*

2. *Besides the physiological significance, the loss specifications have intuitive appeal. It indicates that in the passband, where minimal attenuation of the input signal is desired, the "loss" is minimal as it is constrained to be below a maximum loss of* α_{max} *dBs. Likewise, in the stopband where maximal attenuation of the input signal is needed, the "loss" is set to be larger than* α_{min}.
3. *When specifying a high-quality filter the value of* α_{max} *should be small, the* α_{min} *value should be large and the transition band as narrow as possible, i.e., approximating as much as possible the frequency* response of an ideal *low-pass filter. The cost of this is a large order for the resulting filter making the implementation expensive computationally and requiring large memory space.*

■ Example 12.3

Consider the following specifications for a low-pass filter:

$$0.9 \leq |H(e^{j\omega})| \leq 1.0 \qquad 0 \leq \omega \leq \pi/2$$
$$0 < |H(e^{j\omega})| \leq 0.1 \qquad 3\pi/4 \leq \omega \leq \pi$$

Determine the equivalent loss specifications.

Solution

The loss specifications are then

$$0 \leq \alpha(e^{j\omega}) \leq 0.92 \qquad 0 \leq \omega \leq \pi/2$$
$$\alpha(e^{j\omega}) \geq 20 \qquad 3\pi/4 \leq \omega \leq \pi$$

where $\alpha_{max} = -20\log_{10}(0.9) = 0.92$ dB, and $\alpha_{min} = -20\log_{10}(0.1) = 20$ dBs. These specifications indicate that in the passband the loss is small, or that the magnitude would change between 1 and $10^{-\alpha_{max}/20} = 10^{-0.92/20} = 0.9$ while in the stopband we would like a large attenuation, at least α_{min}, or that the magnitude would have values smaller than $10^{-\alpha_{min}/20} = 0.1$ ■

12.3.1.2 Magnitude Normalization

The specifications of the low-pass filter in Figure 12.5 are **normalized in magnitude**: the dc gain is assumed to be unity (or the dc loss is 0 dB), but there are many cases where that is not so.

Not normalized magnitude specifications (Figure 12.6): In general, the loss specifications are

$$\alpha_1 \leq \hat{\alpha}(e^{j\omega}) \leq \alpha_2 \quad 0 \leq \omega \leq \omega_p$$
$$\alpha_3 \leq \hat{\alpha}(e^{j\omega}) \qquad \omega_{st} \leq \omega \leq \pi$$

Writing these loss specifications as

$$\hat{\alpha}(e^{j\omega}) = \alpha_1 + \alpha(e^{j\omega}) \tag{12.9}$$

we have that

- the normalized specifications are given as

$$0 \leq \alpha(e^{j\omega}) \leq \alpha_{max} \quad 0 \leq \omega \leq \omega_p$$
$$\alpha_{\min} \leq \alpha(e^{j\omega}) \qquad \omega_{st} \leq \omega \leq \pi$$

  where $\alpha_{\max} = \alpha_2 - \alpha_1$ and $\alpha_{\min} = \alpha_3 - \alpha_1$.
- the dc loss of α_1 is achieved by multiplying a magnitude-normalized filter by a constant K such that

$$\hat{\alpha}(e^{j0}) = \alpha_1 = -20\log_{10} K \text{ or } K = 10^{-\alpha_1/20} \tag{12.10}$$

Example 12.4

Suppose the loss specifications of a low-pass filter are

$$10 \leq \hat{\alpha}(e^{j\omega}) \leq 11 \qquad 0 \leq \omega \leq \frac{\pi}{2}$$
$$\hat{\alpha}(e^{j\omega}) \geq 50 \qquad \frac{3\pi}{4} \leq \omega \leq \pi$$

Determine the loss specifications that can be used to design a magnitude-normalized filter. Find a gain K that when it multiplies the normalized filter the resulting filter satisfies the specifications.

Solution

If we let:

$$\hat{\alpha}(e^{j\omega}) = 10 + \alpha(e^{j\omega})$$

the loss specifications for a normalized filter would be

$$0 \leq \alpha(e^{j\omega}) \leq 1 \qquad 0 \leq \omega \leq \frac{\pi}{2}$$
$$\alpha(e^{j\omega}) \geq 40 \qquad \frac{3\pi}{4} \leq \omega \leq \pi$$

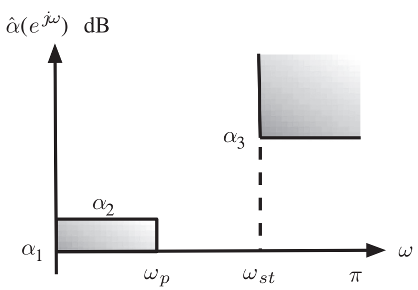

FIGURE 12.6
Not normalized loss specifications for a low-pass filter.

Then the dc loss is 10 dBs and $\alpha_{max} = 11 - 10 = 1$ and $\alpha_{min} = 50 - 10 = 40$ dBs. Suppose that we design a filter $H(z)$ that satisfies the normalized filter specifications. If we let $\hat{H}(z) = KH(z)$ be the filter that satisfies the given loss specifications, at the dc frequency we must have

$$-20\log_{10}|\hat{H}(e^{j0})| = -20\log_{10}K - 20\log_{10}|H(e^{j0})| \text{ or } 10 = -20\log_{10}K + 0$$

so that $K = 10^{-0.5} = 1/\sqrt{10}$. ■

12.3.1.3 Frequency Scales

Given that discrete filters can be used to process continuous as well as discrete-time signals, there are different equivalent ways the frequency of a discrete filter can be expressed (see Figure 12.7).

> In the discrete processing of continuous-time signals the sampling frequency (f_s in Hz or Ω_s in rad/sec) is known, and so we have the following possible scales:
>
> - the f (Hz) scale from 0 to $f_s/2$ (the fold-over or Nyquist frequency) derived from the sampling theory,
> - the scale $\Omega = 2\pi f$ (rad/sec), where f is the previous scale, the frequency range is then from 0 to $\Omega_s/2$,
> - the discrete frequency scale $\omega = \Omega T_s$ (rad) ranging from 0 to π,
> - a normalized discrete frequency scale ω/π (no units) ranging from 0 to 1.
>
> If the specifications are in the discrete domain, the scales used are the ω (rad) or the normalized ω/π.

Other scales are possible, but less used. One of these consists in dividing by the sampling frequency either in Hz or in rad/sec: the f/f_s (no units) scale goes from 0 to 1/2, and so does the Ω/Ω_s (no units) scale. It is clear that when the specifications are given in any scale, it can be easily transformed into any other desired scale. If the filter is designed for use in the discrete domain only the scales in radians and the normalized ω/π are the ones to use.

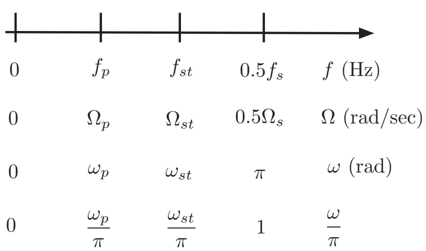

FIGURE 12.7
Frequency scales used in discrete filter design.

12.3.2 Time-Domain Specifications

Time-domain specifications consist in giving a desired impulse response $h_d[n]$. For instance, when designing a low-pass filter with cutoff frequency ω_c and linear phase $\phi(\omega) = -N\omega$, the desired frequency response in $0 \leq \omega \leq \pi$ is

$$H_d(e^{j\omega}) = \begin{cases} 1e^{-j\omega N} & 0 \leq \omega \leq \omega_c \\ 0 & \omega_c < \omega \leq \pi \end{cases}$$

The desired impulse response for this filter is then found from

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} 1e^{-j\omega N} e^{j\omega n} d\omega$$

The resulting $h_d[n]$ will be used as the desired impulse response to approximate.

■ Example 12.5

Consider an FIR filter design with the following desired magnitude response

$$|H_d(e^{j\omega})| = \begin{cases} 1 & 0 \leq \omega \leq \pi/4 \\ 0 & \pi/4 < \omega \leq \pi \end{cases}$$

and zero phase. Find the desired impulse response $h_d[n]$ that we wish to approximate.

Solution

Since the magnitude response is even function of ω, the desired impulse response is computed as follows:

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\pi/4}^{\pi/4} e^{j\omega n} d\omega = \begin{cases} \sin(\pi n/4)/\pi n & n \neq 0 \\ 0.25 & n = 0 \end{cases}$$

which corresponds to the impulse response of a non-causal system. As we will see later, windowing and shifting of $h_d[n]$ are needed to make it into a causal, finite length filter. ■

12.4 IIR FILTER DESIGN

Two possible approaches in the design of IIR filters are:

- using analog filter design methods and transformations between the s-plane and the Z-plane, and
- using optimization techniques.

The first is a **frequency transformation approach**. Using a mapping between the analog and the discrete frequencies we obtain the specifications for an analog filter from the discrete filter specifications. Applying well-known analog filter design methods, we then design the analog filter from the transformed specifications. The discrete filter is finally obtained by transforming the designed analog filter.

The **optimal approach designs** the filter directly, setting the rational approximation as a non-linear optimization. The added flexibility of this approach is diminished by the need to insure stability of the designed filter. Stability is guaranteed, on the other hand, in the transformation approach.

12.4.1 Transformation Design of IIR Discrete Filters

To take advantage of analog filter design, a common practice is to design discrete filters by means of analog filters and mappings of the s-plane into the Z-plane. Two mappings used are:

- the sampling transformation $z = e^{sT_s}$, and
- the bilinear transformation

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

Recall the transformation $z = e^{sT_s}$ was found when relating the Laplace transform of a sampled signal with its Z-transform. Using this transformation, we convert the analog impulse response $h_a(t)$ of an analog filter into the impulse response $h[n]$ of a discrete filter and obtain the corresponding transfer function. The resulting design procedure is called the **impulse invariant method**. Advantages of this method are:

- it preserves the stability of the analog filter, and
- given the linear relation between the analog frequency Ω and the discrete frequency ω, the specifications for the discrete filter can be easily transformed into the specifications for the analog filter.

Its drawback is possible frequency aliasing. Sampling of the analog impulse response requires that the analog filter be band-limited which might not be possible to satisfy in all cases. Due to this we will concentrate on the approach based on the bilinear transformation.

12.4.1.1 The Bilinear Transformation

The bilinear transformation results from the **trapezoidal rule approximation of an integral**. Suppose that $x(t)$ is the input and $y(t)$ the output of an integrator with transfer function

$$H(s) = \frac{Y(s)}{X(s)} = \frac{1}{s} \tag{12.11}$$

Sampling the input and the output of this filter using a sampling period T_s, we have that the integral at time nT_s is

$$y(nT_s) = \int_{(n-1)T_s}^{nT_s} x(\tau)d\tau + y((n-1)T_s)$$

where $y((n-1)T_s)$ is the integral at time $(n-1)T_s$. If T_s is very small, the integral between $(n-1)T_s$ and nT_s can be approximated by the area of a trapezoid with bases $x((n-1)T_s)$ and $x(nT_s)$ and height T_s (this is called the trapezoidal rule approximation of an integral):

$$y(nT_s) \approx \frac{[x(nT_s) + x((n-1)T_s)]T_s}{2} + y((n-1)T_s)$$

with a Z-transform given by

$$Y(z) = \frac{T_s(1+z^{-1})}{2(1-z^{-1})} X(z)$$

The discrete transfer function is thus

$$H(z) = \frac{Y(z)}{X(z)} = \frac{T_s}{2}\frac{1+z^{-1}}{1-z^{-1}} \tag{12.12}$$

which can be obtained directly from $H(s)$ in Equation (12.11) by letting

$$s = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}} \tag{12.13}$$

Thinking of the above transformation as a transformation from the z to the s variable, solving for the variable z in that equation we obtain a transformation from the s to the z variable:

$$z = \frac{1+(T_s/2)s}{1-(T_s/2)s} \tag{12.14}$$

The **bilinear transformation** (linear in the numerator and in the denominator) that transforms from the s-plane into the z-plane is

$$z = \frac{1+s/K}{1-s/K} \qquad K = \frac{2}{T_s} \tag{12.15}$$

and it maps

- the $j\Omega$-axis in the s-plane into the unit circle in the Z-plane,
- the open left-hand s-plane $\mathcal{R}e[s] < 0$ into the inside of the unit circle in the Z-plane, or $|z| < 1$, and
- the open right-hand s-plane $\mathcal{R}e[s] > 0$ into the outside of the unit circle in the Z-plane, or $|z| > 1$.

The bilinear transformation from the z-plane to the s-plane is

$$s = K\frac{1-z^{-1}}{1+z^{-1}} \tag{12.16}$$

Thus as shown in Figure 12.8, the point (A), $s = 0$ or the origin of the s-plane, is mapped into $z = 1$ on the unit circle; points (B) and (B′), $s = \pm j\infty$, are mapped into $z = -1$ on the unit circle; point (C), $s = -1$, is mapped into $z = (1-1/K)/(1+1/K) < 1$ which is inside the unit circle; finally point (D),

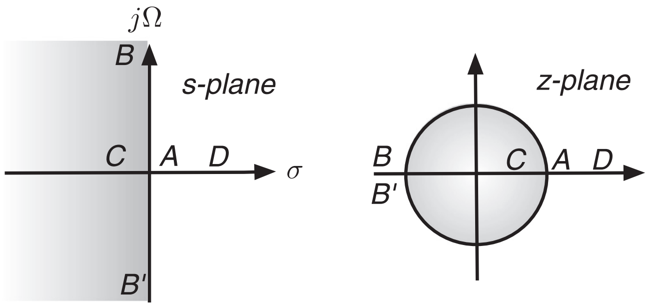

FIGURE 12.8
Bilinear transformation mapping of **s**-plane into **Z**-plane.

$s = 1$, is mapped into $z = (1 + 1/K)/(1 - 1/K) > 1$, located outside the unit circle.

In general, by letting $z = re^{j\omega}$ and $s = \sigma + j\Omega$ in (12.15) we obtain

$$\begin{aligned} r &= \sqrt{\frac{(1+\sigma/K)^2 + (\Omega/K)^2}{(1-\sigma/K)^2 + (\Omega/K)^2}} \\ \omega &= \tan^{-1}\left(\frac{\Omega/K}{1+\sigma/K}\right) + \tan^{-1}\left(\frac{\Omega/K}{1-\sigma/K}\right) \end{aligned} \tag{12.17}$$

from which we have that

- In the $j\Omega$ -axis of the s-plane, i.e., when $\sigma = 0$ and $-\infty < \Omega < \infty$, we obtain $r = 1$ and $-\pi \leq \omega < \pi$ which corresponds to the unit circle of the Z-plane.
- In the open left-hand s-plane, or equivalently when $\sigma < 0$ and $-\infty < \Omega < \infty$, we obtain $r < 1$ and $-\pi \leq \omega < \pi$, or the inside of the unit circle in the Z-plane.
- Finally, in the open right-hand s-plane, or equivalently when $\sigma > 0$ and $-\infty < \Omega < \infty$, we obtain $r > 1$ and $-\pi \leq \omega < \pi$, or the outside of the unit circle in the Z-plane.

The above transformation can be visualized by thinking of a giant who puts a nail in the origin of the s-plane, grabs the plus and minus infinity extremes of the $j\Omega$-axis and pulls them together to make them agree into one point, getting a magnificent circle, keeping everything in the left-hand s-plane inside, and keeping out the rest. If our giant lets go, we get back the original s-plane!

The bilinear transformation maps the whole s-plane into the whole Z-plane, differently from the transformation $z = e^{sT_s}$ that only maps a slab of the s-plane into the Z-plane (see Chapter 10 on the Z-transform). Thus a stable analog filter with poles in the open left-hand s-plane will generate a discrete filter that is also stable, as it has all its poles inside the unit circle.

12.4.1.2 Frequency Warping

A minor drawback of the bilinear transformation is the non-linear relation between the analog and the discrete frequencies. Such a relation creates a warping that needs to be taken care of when specifying the analog filter using the discrete filter specifications.

> The analog frequency Ω and the discrete frequency ω according to the bilinear transformation are related by
>
> $$\Omega = K\tan(\omega/2) \tag{12.18}$$
>
> which when plotted displays a linear relation around the low frequencies but it warps as we get into larger frequencies (see Figure 12.9).

The relation between the frequencies is obtained by letting $\sigma = 0$ in the second equation in (12.17). The linear relationship at low frequencies can be seen using the expansion of the $\tan(.)$ function

$$\Omega = K\left[\frac{\omega}{2} + \frac{\omega^3}{24} + \cdots\right] \approx \frac{\omega}{T_s}$$

for small values of ω or $\omega \approx \Omega T_s$. As frequency increases the effect of the terms beyond the first one makes the relation non-linear. See Figure 12.9.

To compensate for the non-linear relation between the frequencies, or the warping effect, the following steps to design a discrete filter are followed:

1. Using the frequency warping relation (12.18) the specified discrete frequencies ω_p, and ω_{st} are transformed into specified analog frequencies Ω_p and Ω_{st}. The magnitude specifications remain the same in the different bands—only the frequency is being transformed.
2. Using the specified analog frequencies and the discrete magnitude specifications an analog filter $H_N(s)$ that satisfies these specifications is designed.

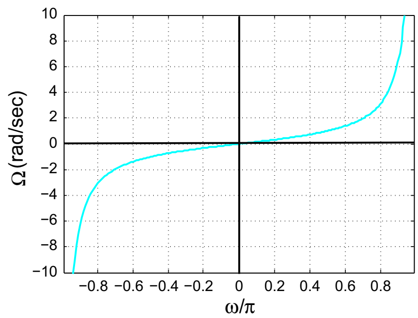

FIGURE 12.9
Relation between Ω and ω for $K = 1$

3. Applying the bilinear transformation to the designed filter $H_N(s)$, the discrete filter $H_N(z)$ that satisfies the discrete specifications is obtained.

12.4.2 Design of Butterworth Low-pass Discrete Filters

Applying the warping relation between the continuous and the discrete frequencies

$$\Omega = K\tan(\omega/2) \tag{12.19}$$

to the magnitude squared function of the Butterworth low-pass analog filter

$$|H_N(j\Omega')|^2 = \frac{1}{1+(\Omega')^{2N}} \qquad \Omega' = \frac{\Omega}{\Omega_{hp}}$$

gives the magnitude squared function for the Butterworth low-pass discrete filter

$$|H_N(e^{j\omega})|^2 = \frac{1}{1+\left[\frac{\tan(0.5\omega)}{\tan(0.5\omega_{hp})}\right]^{2N}} \tag{12.20}$$

As a frequency transformation (no change to the loss specifications) we directly obtain the minimum order N and the half-power frequency bounds by replacing

$$\frac{\Omega_{st}}{\Omega_p} = \frac{\tan(\omega_{st}/2)}{\tan(\omega_p/2)} \tag{12.21}$$

in the corresponding formulas for N and Ω_{hp} of the analog filter giving

$$N \geq \frac{\log_{10}[(10^{0.1\alpha_{min}}-1)/(10^{0.1\alpha_{max}}-1)]}{2\log_{10}\left[\frac{\tan(\omega_{st}/2)}{\tan(\omega_p/2)}\right]} \tag{12.22}$$

$$2\tan^{-1}\left[\frac{\tan(\omega_p/2)}{(10^{0.1\alpha_{max}}-1)^{1/2N}}\right] \leq \omega_{hp} \leq 2\tan^{-1}\left[\frac{\tan(\omega_{st}/2)}{(10^{0.1\alpha_{min}}-1)^{1/2N}}\right] \tag{12.23}$$

The normalized half-power frequency $\Omega'_{hp} = 1$ in the continuous domain is mapped into the discrete half-power frequency ω_{hp}, giving the constant in the bilinear transformation

$$K_b = \left.\frac{\Omega'}{\tan(0.5\omega)}\right|_{\Omega'=1,\omega=\omega_{hp}} = \frac{1}{\tan(0.5\omega_{hp})} \tag{12.24}$$

The bilinear transformation $s = K_b(1-z^{-1})/(1+z^{-1})$ is then used to convert the analog filter $H_N(s)$, satisfying the transformed specifications, into the desired discrete filter

$$H_N(z) = H_N(s)|_{s=K_b(1-z^{-1})/(1+z^{-1})}$$

The basic idea of this design is to convert an **analog frequency-normalized** Butterworth magnitude squared function into a discrete function using the relationship (12.19). To understand why this is an efficient approach consider the following issues that derive from the application of the bilinear transformation to the Butterworth design:

- Since the discrete magnitude specifications are not changed by the bilinear transformation, we only need to change the analog frequency term in the formulas obtained before for the Butterworth low-pass analog filter.
- It is important to recognize that when finding the minimum order N and the half-power relation the value of K is not used. This constant is only important in the final step where the analog filter is transformed into the discrete filter using the bilinear transformation.
- When considering that $K = 2/T_s$ depends on T_s, one might think that a small value for T_s would improve the design, but that is not the case. Given that the analog frequency is related to the discrete frequency as

$$\Omega = \frac{2}{T_s} \tan\left(\frac{\omega}{2}\right) \tag{12.25}$$

  for a given value of ω if we choose a small value of T_s the specified analog frequency Ω is large, and if we choose a large value of T_s the analog frequency Ω decreases. In fact, in the above equation we can only choose either Ω or T_s. To avoid this ambiguity, we ignore the connection of K with T_s and concentrate on K.
- An appropriate value for K for the Butterworth design is obtained by connecting the normalized half-power frequency $\Omega'_{hp} = 1$ in the analog domain with the corresponding frequency ω_{hp} in the discrete domain. This allows us to go from the discrete domain specifications *directly* to the analog normalized frequency specifications. Thus we map the normalized half-power frequency $\Omega'_{hp} = 1$ into the discrete half-power frequency ω_{hp}, by means of K_b.
- Once the analog filter $H_N(s)$ is obtained, using the bilinear transformation with the K_b we transform $H_N(s)$ into a discrete filter

$$H_N(z) = H_N(s)|_{s=K_b \frac{z-1}{z+1}}$$

- The filter parameters (N, ω_{hp}) can also be obtained directly from the discrete loss function obtained from equation (12.20) as

$$\alpha(e^{j\omega}) = 10 \log_{10} \left[1 + (\tan(0.5\omega)/\tan(0.5\omega_{hp}))^{2N}\right] \tag{12.26}$$

  and the loss specifications

$$\begin{aligned} 0 \leq \alpha(e^{j\omega}) &\leq \alpha_{\max} \quad 0 \leq \omega \leq \omega_p \\ \alpha(e^{j\omega}) &\geq \alpha_{\min} \qquad \omega \geq \omega_{st} \end{aligned}$$

  just as we did in the continuous case. The results coincide with those where we replace the warping frequency relation.

Example 12.6

The analog signal $x(t) = \cos(40\pi t) + \cos(500\pi t)$ is sampled using the Nyquist frequency and processed with a discrete filter $H(z)$ which is obtained from a second-order, high-pass analog filter

$$H(s) = \frac{s^2}{s^2 + \sqrt{2}s + 1}$$

The discrete-time output $y[n]$ is then converted into analog. Apply MATLAB's *bilinear* function to obtain the discrete filter with half-power frequencies $\omega_{hp} = \pi/2$. Use MATLAB to plot the poles and zeros of the discrete filters in the Z-plane and the corresponding magnitude and phase responses. Use the function *plot* to plot the sampled input and the filter output and consider these approximations to the analog signals. Change the frequency scale of the discrete filter into f in Hz and indicate what is the corresponding half-power frequency in Hz.

Solution

The coefficients of the numerator and denominator of the discrete filter are found from $H(s)$ using the MATLAB function *bilinear*. The input F_s in this function equals $K_b/2$, where K_b corresponds to the transformation of the discrete half-power frequency, ω_{hp}, into the normalized analog half-power frequency $\Omega_{hp} = 1$.
The following script is used:

```
%%
% Example 12.6
%%
b=[1 0 0]; a=[1 sqrt(2) 1];  % coefficients of analog filter
whp=0.5 * pi;                % desired half-power frequency
Kb=1/tan(whp/2); Fs=Kb/2;
[num, den]=bilinear(b,a,Fs); % bilinear transformation
Ts=1/500;                    % sampling period
n=0:499; x1=cos(2*pi*20*n*Ts)+cos(2*pi*250*n*Ts);  % sampled signal
zplane(num, den)              % poles/zeros of discrete filter
[H,w]=freqz(num,den);         % frequency response discr. filter
phi=unwrap(angle(H));         % unwrapped phase discr. filter
y=filter(num,den,x1);         % output of discr. filter, input x1
```

We find the transfer function of the discrete filter to be

$$H(z) = \frac{0.2929(1 - z^{-1})^2}{1 + 0.1715z^{-2}}$$

The poles and zeros of $H(z)$ can be found with the MATLAB function *roots* and plotted with *zplane*. The frequency response is obtained using *freqz*. To have the frequency scale in Hz we consider that $\omega = \Omega T_s$, letting $\Omega = 2\pi f$ then

$$f = \frac{\omega}{2\pi T_s} = \left(\frac{\omega}{\pi}\right)\left(\frac{f_s}{2}\right)$$

so we multiply the normalized discrete frequency ω/π by $f_s/2 = 250$, resulting in a maximum frequency of 250 Hz. The half-power frequency in Hz is thus 125 Hz. The magnitude and phase responses of $H(z)$ are shown in Figure 12.10. Notice that phase is approximately linear in the passband, despite the fact that no phase specifications are considered.

Since the maximum frequency of $x(t)$ is 250 Hz we choose the sampling period to be $T_s = 1/(2f_{max}) = 1/500$. As a high-pass filter, when we input $x(nT_s)$ into $H(z)$ its low-frequency component $\cos(40\pi nT_s)$ is attenuated. The input and corresponding output of the filter are shown in Figure 12.10. ■

Remarks

1. *The Butterworth low-pass filter design is simplified by giving as a specification a desired half-power frequency ω_{hp}. We only need then to calculate the order of the filter by using the stopband constraint. In such a case, we could let $\alpha_{\max} = 3$ dB and $\omega_p = \omega_{hp}$ and use (12.22) to find N.*
2. *A very important consequence of using the bilinear transformation is that the resulting transfer function $H_N(z)$ is guaranteed to be BIBO stable. This transformation maps the poles of a stable filter $H_N(s)$ in the open left-hand s-plane into poles inside the unit circle corresponding to $H_N(z)$, making the discrete filter stable.*
3. *The bilinear transformation creates a pole and a zero in the Z-plane for each pole in the s-plane. Analytic calculation of the poles of $H_N(z)$ is not as important as*

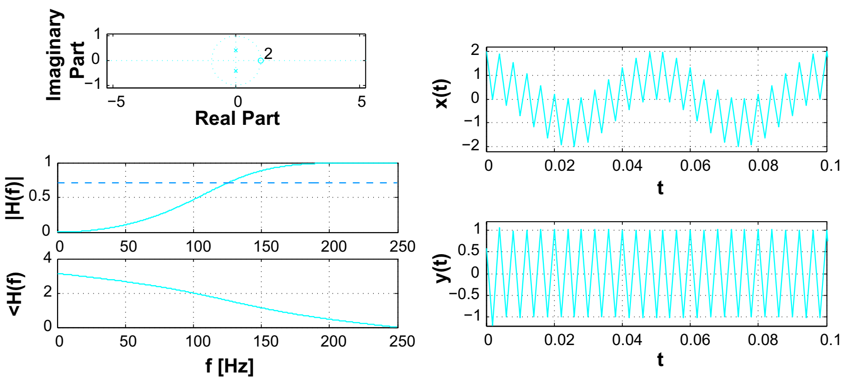

FIGURE 12.10

Bilinear transformation of a high-pass analog filter into a discrete filter with half-power frequencies $\omega_{hp} = \pi/2$ or $f_{hp} = 125$ Hz. Poles and zeros, and magnitude and phase response of the discrete filter are shown on left plots. The plots on the right show the analog input and output obtained using the MATLAB function ***plot*** to interpolate the sampled signal $x(nT_s)$ and the output of the discrete filter $y(nT_s)$ into $x(t)$ and $y(t)$.

in the analog case, the MATLAB function zplane can be used to plot its poles and zeros, and the function roots to find the values of the poles and zeros.

4. *Applying the bilinear transformation by hand to filters of order higher than 2 is cumbersome. When doing so, $H_N(s)$ should be expressed as a product or sum of first- and second-order transfer functions before applying the bilinear transformation to each, i.e.,*

$$H_N(s) = \prod_i H_{N_i}(s) \quad \text{or} \quad H_N(s) = \sum_\ell H_{N_\ell}(s)$$

where $H_{Ni}(s)$ and $H_{N\ell}(s)$ are first- or second-order functions with real coefficients. Applying the bilinear transformation to each of the $H_{Ni}(s)$ or $H_{N\ell}(s)$ components to obtain $H_{Ni}(z)$ and $H_{N\ell}(z)$, the discrete filter becomes

$$H_N(z) = \prod_i H_{N_i}(z) \quad \text{or} \quad H_N(z) = \sum_\ell H_{N_\ell}(z)$$

5. *The zeros of the discrete low-pass Butterworth filter are at $z = -1$; this is due to the rationalization of the analog filter. If the analog low-pass Butterworth filter has a transfer function*

$$H_N(s) = \frac{1}{a_0 + a_1 s + \cdots + a_N s^N}$$

letting $s = K_b(1 - z^{-1})/(1 + z^{-1})$ we obtain a discrete filter

$$\begin{aligned} H_N(z) &= \frac{1}{a_0 + a_1 K_b (1 - z^{-1})/(1 + z^{-1}) + \cdots + a_N K_b^N (1 - z^{-1})^N/(1 + z^{-1})^N} \\ &= \frac{(1 + z^{-1})^N}{a_0 (1 + z^{-1})^N + a_1 K_b (1 - z^{-1})(1 + z^{-1})^{N-1} + \cdots + a_N K_b^N (1 - z^{-1})^N} \end{aligned}$$

with N zeros at $z = -1$.

6. *Since the resulting filter has normalized magnitude, a specified dc gain can be attained by multiplying $H_N(z)$ by a constant value G so that $|GH(e^{j0})|$ equals the desired dc gain.*

■ Example 12.7

The specifications of a low-pass discrete filter are

$$\begin{aligned} &\omega_p = 0.47\pi\,(\text{rad}) \qquad \alpha_{max} = 2\text{ dB} \\ &\omega_{st} = 0.6\pi\,(\text{rad}) \qquad \alpha_{min} = 9\text{ dB} \\ &\alpha(e^{j0}) = 0\text{ dB} \end{aligned}$$

Use MATLAB to design a discrete low-pass Butterwoth filter using the bilinear transformation.

Solution

Normalizing the frequency specifications to ω_p/π and ω_{st}/π we use these values directly in the MATLAB function *buttord* as inputs along with α_{max}, and α_{min}. The function gives the minimum order N and the normalized half-power frequency ω_{hp}/π of the filter. With these as inputs of the function *butter* we obtain the coefficients of the numerator and denominator of the designed filter $H(z) = B(z)/A(z)$. The function *roots* is used to find the poles and zeros of $H(z)$, while *zplane* plots them. The magnitude and phase responses are found using *freqz* aided by the functions *abs*, *angle*, and *unwrap*. Notice that *butter* obtains the normalized analog filter and transforms it using the bilinear transformation. The script used is:

```
%%
% Example 12.7
%%
% LP Butterworth
alphamax=2; alphamin=9;                     % loss specifications
wp=0.47;ws=0.6;                             % passband and
                                              stopband
                                              frequencies
[N,wh]=buttord(wp,ws,alphamax,alphamin)     % minimum order,
                                            half-power frequency
[b,a]=butter(N,wh);                         % coefficients of
                                            designed filter
[H,w]=freqz(b,a);w=w/pi;N=length(H);        % frequency response
spec1=alphamax*ones(1,N);
spec2=alphamin*ones(1,N);                   % specification lines
hpf=3.01 *  ones(1,N);                      % half-power
                                              frequency line
disp('poles')                               % display poles
roots(a)
disp('zeros')                           % display zeros
roots(b)
alpha=-20*log10(abs(H));                % loss in dB
```

The results of the design are shown in Figure 12.11. The minimum order of the designed filter is $N = 3$ and the half-power frequency is $\omega_{hp} = 0.499\pi$. The poles are along the imaginary axis of the Z-plane ($K_b = 1$) and there are three zeros at $z = -1$. The transfer function of the designed filter is

$$H(z) = \frac{0.166 + 0.497z^{-1} + 0.497z^{-2} + 0.166z^{-3}}{1 - 0.006z^{-1} + 0.333z^{-2} - 0.001z^{-3}}$$

Finally, to verify that the specifications are satisfied we plot the loss function $\alpha(e^{j\omega})$ along with three horizontal lines corresponding to $\alpha_{max} = 2$ dB, the 3 dB for the half-power frequency and $\alpha_{min} = 9$ dB. The crossings of these

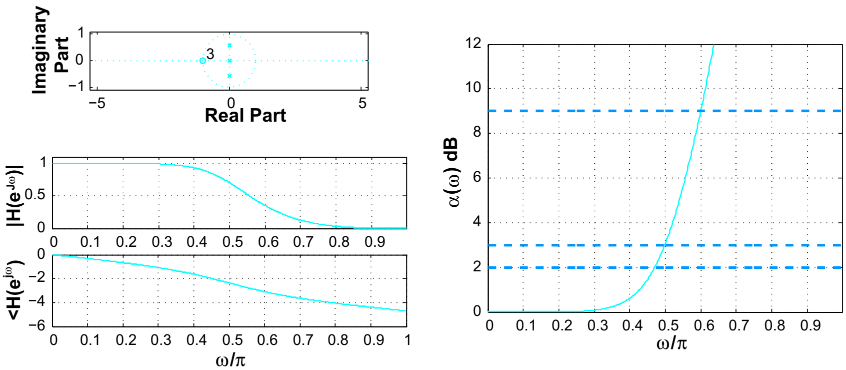

FIGURE 12.11
Design of low-pass Butterworth filter using MATLAB: poles and zeros, magnitude and phase response (left); verification of specifications using loss function $\alpha(\omega)$.

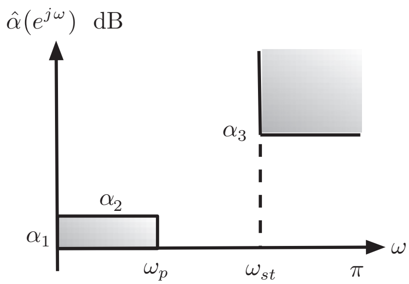

FIGURE 12.12
Loss specifications for a discrete low-pass filter used to process analog signals.

lines with the filter loss function indicate that at the normalized frequencies [0 0.47) the loss is less than 2 dB, at the normalized frequencies (0.6 1] the loss is bigger than 9 dB as desired, and that the normalized half-power frequency is about 0.5 (see Figure 12.12). ■

■ Example 12.8

In this example we consider designing a Butterworth low-pass discrete filter for processing an analog signal. The filter specifications are:

$$f_p = 2250 \text{ Hz passband frequency}, \quad \alpha_1 = -18 \text{ dB dc loss}$$
$$f_{st} = 2700 \text{ Hz stopband frequency}, \quad \alpha_2 = -15 \text{ dB loss in passband}$$
$$f_s = 9000 \text{ Hz sampling frequency}, \quad \alpha_3 = -9 \text{ dB loss in stopband}$$

Solution

The specifications are not normalized. Normalizing them we have that

$$\hat{\alpha}(e^{j0}) = -18 \text{ dB, dc gain,} \quad \alpha_{max} = \alpha_2 - \alpha_1 = 3 \text{ dB,} \quad \alpha_{min} = \alpha_3 - \alpha_1 = 9 \text{ dB,}$$
$$\omega_p = \frac{2\pi f_{hp}}{f_s} = 0.5\pi, \quad \omega_{st} = \frac{2\pi f_{st}}{f_s} = 0.6\pi$$

Note that $\omega_p = \omega_{hp}$ since the difference in the losses at dc and at ω_p is 3 dB.

Since the sample period is

$$T_s = 1/f_s = (1/9) \times 10^{-3} \text{ sec/sample} \qquad \Rightarrow K_b = \cot(\pi f_{hp} T_s) = 1$$

Given that the half-power frequency is known, only the minimum order of the filter is needed. The loss function for the Butterworth filter is then

$$\alpha(e^{j\omega}) = 10\log_{10}\left(1 + \left[\frac{\tan(0.5\omega)}{\tan(0.5\omega_{hp})}\right]^{2N}\right) = 10\log_{10}(1 + (\tan(0.5\omega))^{2N})$$

since $0.5\omega_{hp} = \pi/4$. For $\omega = \omega_{st}$, letting $\alpha(e^{j\omega_{st}}) \geq \alpha_{min}$ solving for N we get

$$N = \left\lceil \frac{\log_{10}(10^{0.1\alpha_{min}} - 1)}{2\log_{10}(\tan(0.5\omega_{st}))} \right\rceil$$

or the integer that is larger than the argument of $\lceil . \rceil$, which by replacing α_{min} and ω_{st} gives as minimum order $N = 4$. The MATLAB script used is:

```
%%
% Example 12.8---filtering of analog signal
%%
wh=0.5*pi; ws=0.6*pi; alphamin=9; Fs=9000;       % filter
                                                  specifications
N=log10((10^(0.1*alphamin)-1))/(2*log10(tan(ws/2)/tan(wh/2)));
N=ceil(N);
[b,a]=butter(N,wh/pi);
[H,w]=freqz(b,a);w=w/pi;N=length(H);f=w*Fs/2;
alpha0=-18;
G=10^(-alpha0/20);H=H*G;
spec2=alpha0+alphamin*ones(1,N);
hpf=alpha0+3.01*ones(1,N);
disp('poles'); p=roots(a); disp('zeros'); z=roots(b)
alpha=-20*log10(abs(H));
```

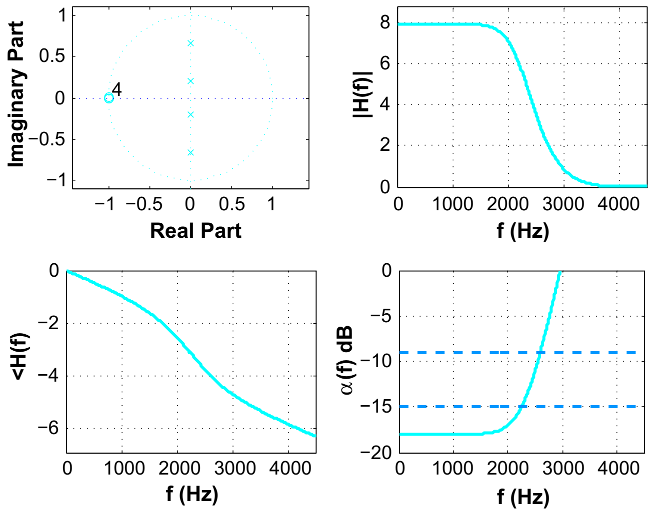

FIGURE 12.13
Low-pass filter for filtering of analog signal.

To insure that the loss is -18 dB at $\omega = 0$, if the transfer function of the normalized loss (0 dB) filter is $H(z)$, we include a gain G in the numerator so that $H'(z) = GH(z)$ has the desired dc loss of -18 dB. The gain G is found from $\alpha(e^{j0}) = -18 = -20\log_{10} G$ to be $G = 7.94$. Thus the final form of the filter is

$$H'(z) = GH(z) = \frac{(z+1)^4}{(z^2+0.45)(z^2+0.04)}$$

which satisfies the loss specifications. Notice that when $K_b = 1$, the poles are imaginary.

Since this filter is used to filter an analog signal the frequency scale of the magnitude and phase response of the filter is given in Hz (see Figure 12.13). To verify that the specifications are satisfied the loss function is plotted and compared with the losses corresponding to f_{hp} and f_{st}. The loss at $f_{hp} = 2250$ (Hz) coincides with the dc loss plus 3 dB, and the loss at $f_{st} = 2700$ (Hz) is above the specified value. ■

12.4.3 Design of Chebyshev Low-pass Discrete Filters

The constant K_c of the bilinear transform for the Chebyshev filter is calculated by transforming the normalized pass frequency $\Omega'_p = 1$ into the discrete frequency ω_p giving

$$K_c = \frac{1}{\tan(0.5\omega_p)} \qquad (12.27)$$

Using the frequencies relation in the bilinear transformation we have that

$$\frac{\Omega}{\Omega_p} = \frac{\tan(0.5\omega)}{\tan(0.5\omega_p)} \tag{12.28}$$

and replacing it in the magnitude squared function for the Chebyshev analog filter, the magnitude squared function of the discrete Chebyshev low-pass filter is

$$|H_N(e^{j\omega})|^2 = \frac{1}{1 + \varepsilon^2 C_N^2(\tan(0.5\omega)/\tan(0.5\omega_p))} \tag{12.29}$$

where $C(.)$ are the Chebyshev polynomials of the first kind encountered before in the analog design. The ripple parameter remains the same as in the analog design (since it does not depend on frequency)

$$\varepsilon = (10^{0.1\alpha_{\max}} - 1)^{1/2} \tag{12.30}$$

Replacing (12.28) in the analog formulas gives the order of the filter

$$N \geq \frac{\cosh^{-1}\left([(10^{0.1\alpha_{\min}} - 1)/(10^{0.1\alpha_{\max}} - 1)]^{1/2}\right)}{\cosh^{-1}[\tan(0.5\omega_{st})/\tan(0.5\omega_p)]} \tag{12.31}$$

and the half-power frequency

$$\omega_{hp} = 2\tan^{-1}\left[\tan(0.5\omega_p)\cosh\left(\frac{1}{N}\cosh^{-1}\left(\frac{1}{\varepsilon}\right)\right)\right] \tag{12.32}$$

After calculating these parameters, the transfer function of the Chebyshev discrete filter is found by transforming the Chebyshev analog filter of order N into a discrete filter using the bilinear transformation:

$$H_N(z) = H_N(s)|_{s=K_c(1-z^{-1})/(1+z^{-1})} \tag{12.33}$$

Remarks

1. *Just as with the Butterworth filter, the equations for the filter parameters (N, ω_{hp}) can be obtained from the analog formulas by substituting*

$$\frac{\Omega_{st}}{\Omega_p} = \frac{\tan(0.5\omega_{st})}{\tan(0.5\omega_p)}$$

   Since the analog ripple factor ε only depends on the magnitude specifications, it is not affected by the bilinear transformation—a frequency only transformation.

2. *The filter parameters $(N, \omega_{hp}, \varepsilon)$ can also be found from the loss function obtained from the discrete Chebyshev squared magnitude:*

$$\alpha(e^{j\omega}) = 10\log_{10}\left[1 + \varepsilon^2 C_N^2\left(\frac{\tan(0.5\omega)}{\tan(0.5\omega_p)}\right)\right] \tag{12.34}$$

   This is done by following a similar approach to the one in the analog case.

3. *Like in the discrete Butterworth, for Chebyshev filters the dc gain (i.e., gain at $\omega = 0$) can be set to any desired value by allowing a constant gain G in the numerator such that*

$$|H_N(e^{j0})| = |H_N(1)| = G\frac{|N(1)|}{|D(1)|} = \text{desired gain} \tag{12.35}$$

4. *MATLAB provides two functions to design Chebyshev filters. The function cheby1 is for designing the filters covered in this section, while cheby2 is to design filters with a flat response in the passband and with ripples in the stopband. The minimum order of the filter is found using cheb1ord and cheb2ord. The functions cheby1 and cheby2 give the filter coefficients.*

Example 12.9

Consider the design of two low-pass Chebyshev filters. The specifications for the first filter are

$$\alpha(e^{j0}) = 0\ dB$$
$$\omega_p = 0.47\pi \text{ rad}, \quad \alpha_{max} = 2 \text{ dBs}$$
$$\omega_{st} = 0.6\pi \text{ rad}, \quad \alpha_{min} = 6 \text{ dBs}$$

For the second filter change ω_p to 0.48π rad, keep the other specifications. Use MATLAB.

Solution

In example 12.7, we obtained a third-order Butterworth low-pass filter that satisfies the specifications of the first filter. According to the results in this example, a second-order Chebyshev filter satisfies the same specifications. It is always so that a Chebyshev filter satisfies the same specifications as a Butterworth filter using a lower minimum order. For the second filter we narrow the transition band by 0.01π radians, and so the minimum order of the Chebyshev filter increases by one, as we will see. The following is the script for the design of the two filters.

```
%%
% Example 12.9 --- LP Chebyshev
%%
alphamax=2; alphamin=9;                    % loss specs
figure(1)
for i=1:2,
  wp=0.47+(i-1)*0.01; ws=0.6;              % normalized frequency specs
  [N,wn]=cheb1ord(wp,ws,alphamax,alphamin)
  [b,a]=cheby1(N,alphamax,wn);
  wp=wp * pi;
  % magnitude and phase
```

```
    [H,w]=freqz(b,a); w=w/pi; M=length(H);H=H/H(1);
    % to verify specs
    spec0=zeros(1,M); spec1=alphamax*ones(1,M)* (-1)^(N+1);
    spec2=alphamin*ones(1,M);
    alpha=-20 *log10(abs(H));
    hpf=(3.01+alpha(1))*ones(1,M);
    % epsilon and half-power frequency
    epsi=sqrt(10^(0.1*alphamax)-1);
    whp= 2*atan(tan(0.5*wp)*cosh(acosh(sqrt(10^(0.1*3.01)-1)/
    epsi)/N));
    whp=whp/pi
    % plotting
    subplot(221); zplane(b,a)
    subplot(222)
    plot(w,abs(H)); grid; ylabel('|H|');
    axis([0 max(w) 0 1.1*max(abs(H))])
    subplot(223)
    plot(w,unwrap(angle(H))); grid;
    ylabel('<H (rad)'); xlabel('\omega/pi)')
    subplot(224)
    plot(w,alpha); ylabel('alpha(\omega) dB'); xlabel('\omega/pi')
  hold on; plot(w,spec0,'r'); hold on; plot(w,spec1,'r')
  hold on; plot(w,hpf,'k'); hold on
  plot(w,spec2,'r'); grid;
  axis([0 max(w) 1.1*min(alpha) 1.1* (alpha(1)+3)]); hold off
  figure(2)
end
```

The transfer function of the first filter is

$$H_1(z) = \frac{0.224 + 0.449z^{-1} + 0.224z^{-2}}{1 - 0.264z^{-1} + 0.394z^{-2}}$$

and its half-power frequency is $\omega_{hp} = 0.493\pi$ rad. The second filter has a transfer function

$$H_2(z) = \frac{0.094 + 0.283z^{-1} + 0.283z^{-2} + 0.094z^{-3}}{1 - 0.691z^{-1} + 0.774z^{-2} - 0.327z^{-3}}$$

and a half-power frequency $\omega_{hp} = 0.4902\pi$. The poles and zeros as well as the magnitude and phase responses of the two filters are shown in Figure 12.14. Notice the difference in the gain (or losses) in the passband of the two filters. In order for the dc gain to be unity, the magnitude response in the even-order filter $H_1(z)$ has values above 1, while the odd-order filter $H_2(z)$ does the opposite.

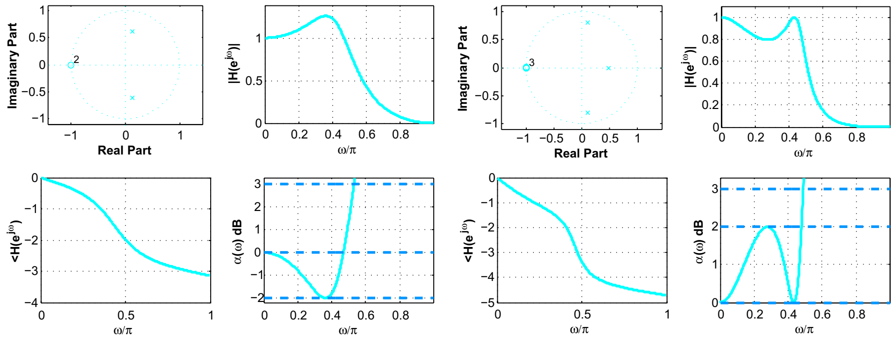

FIGURE 12.14

Two Chebyshev filters with different transition bands: even-order filter for $\omega_p = 0.47\pi$ on the left, and odd-order filter for $\omega_p = 0.48\pi$ (narrower transition band) on the right.

It is important to indicate that the output frequency ω_n given by *cheb1ord* and that *cheby1* uses as input is the passband frequency ω_p. Since the half power is not given by *cheb1ord* the half-power frequency of the filter can be calculated using the minimum order N, the ripple factor ε and the passband frequency ω_p in Equation (12.32). See the script. ■

■ Example 12.10

Consider the following specifications of a filter that will be used to filter an acoustic signal

dc gain = 10, half-power frequency $f_{hp} = 4$ kHz

band-stop frequency $f_{st} = 5$ kHz, $\alpha_{min} = 60$ dBs

sampling frequency $f_s = 20$ kHz

Design a Butterworth and a Chebyshev low-pass filters of the same order and compare their frequency responses.

Solution

The specifications of the discrete filter are

$$\begin{aligned}
&\text{dc gain} = 10 \Rightarrow \alpha(e^{j0}) = -20 \text{ dBs}\\
&\text{half-power frequency:} \quad \omega_{hp} = 2\pi f_{hp}(1/f_s) = 0.4\pi \text{ rad}\\
&\text{band-stop frequency:} \quad \omega_{st} = 2\pi f_{st}(1/f_s) = 0.5\pi \text{ rad}
\end{aligned}$$

When designing the Butterworth filter we only need to find the minimum order N given that the half-power frequency is specified. We find that $N = 15$ satisfies the specifications. Using this value and the given discrete

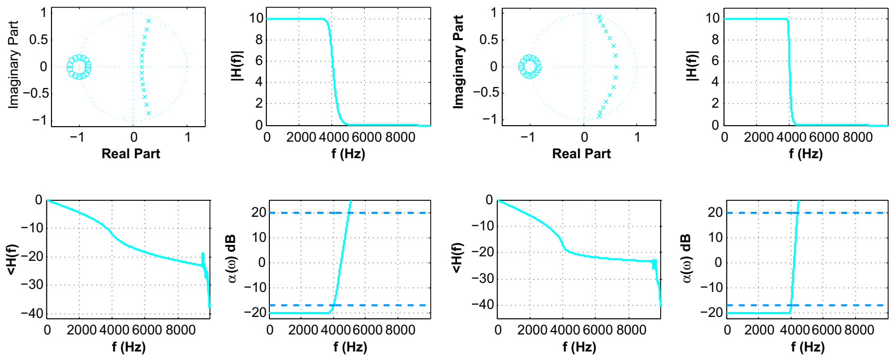

FIGURE 12.15
Equal-order ($N = 15$) Butterworth (left) and Chebyshev filters for filtering of acoustic signal.

half-power frequency the function *butter* gives the coefficients of the filter. See Figure 12.15 for results.

The design of the Chebyshev filter with order $N = 15$ and half-power frequency $\omega_{hp} = 0.4\pi$ cannot be done directly with the function *cheby1*, as we do not have the passband frequency ω_p. Using Equation (12.32) to compute the half-power frequency, we solve for ω_p after we give a value to α_{max} (arbitrarily chosen to be 0.001 dB) which allows us to compute the ripple factor ε (see Equation (12.30)). See the script part corresponding to the Chebyshev design. The function *cheby1* with inputs N, α_{max}, and ω_p gives the coefficients of the designed filter. Using these coefficients we plot the poles/zeros, magnitude and phase responses, and the loss function as shown in Figure 12.15. According to the loss function plots the Chebyshev filter displays a sharper response in the transition band than the Butterworth filter, as expected. ■

```
%%
% Example 12.10---Butterworth/Chebyshev filters for analog signal
%%
wh=0.4*pi;ws=0.5*pi; alphamin=40;Fs=20000;
% Butterworth
  %N=log10((10^(0.1*alphamin)-1))/(2*log10(tan(ws/2)/tan(wh/2))); N=ceil(N)
  % [b,a]=butter(N,wh/pi); % to get Butterworth filter get rid of '%'
% Chebyshev
  alphamax=0.001;
  epsi=sqrt(10^(0.1*alphamax)-1);
  % computation of wp for Chebyshev design
  wp=2*atan(tan(0.5*wh)/(cosh(acosh(sqrt(10^(0.1*3.01)-1)/
epsi)/N)));
  wp=wp/pi; [b,a]=cheby1(N,alphamax,wp);
```

```
% magnitude and phase
  [H,w]=freqz(b,a);w=w/pi;M=length(H); f=w*Fs/2;
  alpha0=-20;H=H*10;
% verifying specs
  spec2=alpha0+alphamin*ones(1,M);
  hpf=alpha0+3.01*ones(1,M);
  alpha=-20 * log10(abs(H));
  Ha=unwrap(angle(H));
```

12.4.4 Rational Frequency Transformations

As indicated before, the conventional approach to filter design is to obtain first a prototype low-pass filter and then to convert it into different types of filters by means of frequency transformations. When using analog filters to design IIR discrete filters the frequency transformation could be done in two ways:

- transform a prototype low-pass analog filter into a desired analog filter which in turn is converted into a discrete filter using the bilinear, or other transformation, into a discrete filter,
- design a prototype low-pass discrete filter and then transform it into the desired discrete filter.

The first approach has the advantage that the analog frequency transformations are available and well understood. Its drawback appears when applying the bilinear transformation as it may cause undesirable warping in the higher frequencies. So the second approach will be used.

Given a prototype low-pass filter $H_{\ell p}(Z)$, we wish to transform it into a desired filter $H(z)$, typically another low-pass, band-pass, high-pass, or band-stop filter. The transformation

$$G(z^{-1}) = Z^{-1} \tag{12.36}$$

should preserve the rationality and the stability of the low-pass prototype. Accordingly, $G(z^{-1})$ should

- be rational, to preserve rationality,
- map the inside of the unit circle in the Z-plane into the inside of the unit circle in the $z$plane, to preserve stability, and
- map the unit circle $|Z| = 1$ into the unit circe $|z| = 1$, so that the frequency response of the prototype filter is mapped into the frequency response of the desired filter.

If $Z = Re^{j\theta}$ and $z = re^{j\omega}$, the third condition on $G(z^{-1})$ corresponds to

$$G(e^{-j\omega}) = |G(e^{-j\omega})|e^{j\angle(G(e^{-j\omega}))} = \underbrace{1e^{-j\theta}}_{\text{unit circle in Z-plane}} \tag{12.37}$$

indicating that the frequency transformation $G(z^{-1})$ has the characteristics of an all-pass filter, with magnitude $|G(e^{-j\omega})| = 1$ and phase $\angle G(e^{-j\omega}) = -\theta$.

Using the general form of the transfer function of an all-pass filter (ratio of two equal order polynomials with poles and zeros being the inverse conjugate of each other) the general form of the rational transformation is

$$Z^{-1} = G(z^{-1}) = A \prod_k \frac{z^{-1} - \alpha_k}{1 - \alpha_k^* z^{-1}} \tag{12.38}$$

where the values of A and $\{\alpha_k\}$ are obtained from the prototype and the desired filters.

12.4.4.1 Low-pass to Low-pass Transformation

We wish to obtain the transformation $Z^{-1} = G(z^{-1})$ to convert a prototype low-pass filter into a different low-pass filter. The all-pass transformation should be able to expand or contract the frequency support of the prototype low-pass filter but keep its order. Thus it should be a ratio of linear transformations:

$$Z^{-1} = A\frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \tag{12.39}$$

for some parameters A and α. Since the zero frequency in the Z-plane is to be mapped into the zero frequency in the Z-plane, if we let $Z = z = 1$ in the transformation we get $A = 1$. To obtain α, we let $Z = 1e^{j\theta}$ and $z = 1e^{j\omega}$ in (12.39) to obtain

$$e^{-j\theta} = \frac{e^{-j\omega} - \alpha}{1 - \alpha e^{-j\omega}} \tag{12.40}$$

The value of α that maps the cutoff frequency θ_p of the prototype into the desired cutoff frequency ω_d is found as follows. First, from (12.40) we have

$$\begin{aligned}\alpha &= \frac{e^{-j\omega} - e^{-j\theta}}{1 - e^{-j(\theta+\omega)}} = \frac{e^{-j\omega} - e^{-j\theta}}{2je^{-j0.5(\theta+\omega)}\sin((\theta+\omega)/2)} \\ &= \frac{e^{j0.5(\theta-\omega)} - e^{-j0.5(\theta-\omega)}}{2j\sin((\theta+\omega)/2)} = \frac{\sin((\theta-\omega)/2)}{\sin((\theta+\omega)/2)}\end{aligned}$$

and then replacing θ and ω by θ_p and ω_d gives

$$\alpha = \frac{\sin((\theta_p - \omega_d)/2)}{\sin((\theta_p + \omega_d)/2)} \tag{12.41}$$

Notice that if the prototype filter coincides with the desired filter, i.e., $\theta_p = \omega_d$, then $\alpha = 0$, and the transformation is $Z^{-1} = z^{-1}$. For different values of α

between 0 and 1 the transformation shrinks the support of the prototype low-pass filter, and conversely for $-1 \leq \alpha < 0$ the transformation expands the support of the prototype. (In Figure 12.16, the frequencies θ and ω are normalized to values between 0 and 1, i.e., both are divided by π.)

Remarks

1. *The LP-LP transformation then consists in:*
   - *given θ_p and ω_d find the corresponding α value (Equation (12.41)), and then*
   - *use the found α in the transformation (12.39) with $A = 1$.*
2. *Even in this simple low-pass to low-pass case the relation between the frequencies θ and ω is highly non-linear. Indeed, solving for $e^{-j\omega}$ in the transformation (12.40) we get*

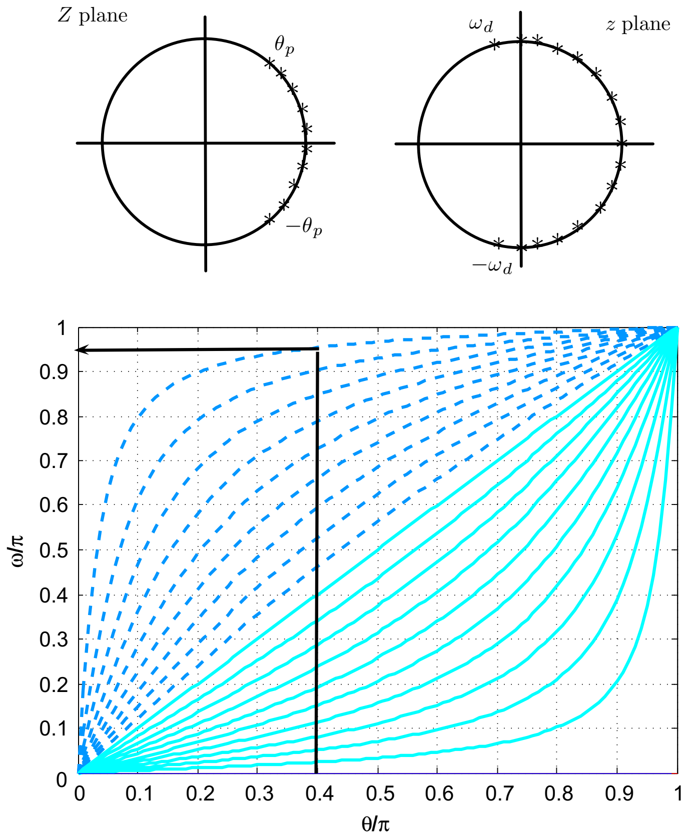

FIGURE 12.16

Frequency transformation from a prototype low-pass filter with cutoff θ_p into a low-pass filter with desired cutoff frequency ω_d (top figure). Mapping of θ/π into ω/π frequencies in low-pass to low-pass frequency transformation: the continuous lines correspond to $0 < \alpha \leq 1$, while the dashed lines correspond to values $-1 \leq \alpha \leq 0$. The arrow shows the transformation of $\theta_p = 0.4\pi$ into $\omega_d \approx 0.95\pi$ when $\alpha = -0.9$.

$$
\begin{aligned}
e^{-j\omega} &= \left(\frac{e^{-j\theta}+\alpha}{1+\alpha e^{-j\theta}}\right)\left(\frac{1+\alpha e^{j\theta}}{1+\alpha e^{j\theta}}\right) = \frac{e^{-j\theta}+2\alpha+\alpha^2 e^{j\theta}}{1+2\alpha\cos(\theta)+\alpha^2} \\
&= \underbrace{\frac{2\alpha+(1+\alpha^2)\cos(\theta)}{1+2\alpha\cos(\theta)+\alpha^2}}_{B} - j\underbrace{\frac{(1-\alpha^2)\sin(\theta)}{1+2\alpha\cos(\theta)+\alpha^2}}_{C}
\end{aligned}
$$

and since $e^{-j\omega} = \cos(\omega) - j\sin(\omega)$ *we find that* $\cos(\omega) = B$ *and* $\sin(\omega) = C$ *so that* $\tan(\omega) = C/B$ *and thus*

$$
\omega = \tan^{-1}\left[\frac{(1-\alpha^2)\sin(\theta)}{2\alpha+(1+\alpha^2)\cos(\theta)}\right] \tag{12.42}
$$

which when plotted for different values of α *gives Figure 12.16. These curves clearly show the mapping of a frequency* θ_p *into* ω_d *and the value of* α *needed to perform the correct transformation.*

12.4.4.2 Low-pass to High-pass Transformation

The duality between low-pass and high-pass filters indicates that this transformation, like the LP-LP, should be linear in both numerator and denominator. Also notice that the prototype low-pass filter can be transformed into a high-pass filter with the same bandwidth, by changing Z^{-1} into $-Z^{-1}$. Indeed, complex poles or zeros $R_1 e^{\pm j\theta_1}$ of the low-pass filter are mapped into $-R_1 e^{\pm j\theta_1} = R_1 e^{j(\pi\pm\theta_1)}$ (i.e., let $\theta_1 \to \pi - \theta_1$) corresponding to a high-pass filter. For instance, a low-pass filter

$$
H(Z) = \frac{Z+1}{Z-0.5}
$$

with a zero at -1 and a pole at 0.5 becomes when letting $Z^{-1} \to Z^{-1}$:

$$
H_1(Z) = \frac{-Z+1}{-Z-0.5} = \frac{Z-1}{Z+0.5}
$$

with a zero at 1 and a pole at -0.5 or a high-pass filter.

The LP-HP transformation is then

$$
Z^{-1} = -\left(\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}\right)
$$

and to obtain α we replace θ_p by $\pi - \theta_p$ in Equation (12.41):

$$
\begin{aligned}
\alpha &= \frac{\sin(-(\theta_p+\omega_d)/2+\pi/2)}{\sin(-(\theta_p-\omega_d)/2+\pi/2)} \\
&= \frac{\cos(-(\theta_p+\omega_d)/2)}{\cos(-(\theta_p-\omega_d)/2)} = \frac{\cos((\theta_p+\omega_d)/2)}{\cos((\theta_p-\omega_d)/2)}
\end{aligned} \tag{12.43}
$$

As before, θ_p is the cutoff frequency of the prototype low-pass filter and ω_d the desired cutoff frequency of the high-pass filter.

The above transformation confirms that when the low-pass and the high-pass filters have the same bandwidth, i.e., $\omega_d = \pi - \theta_p$, we have that $\theta_p + \omega_d = \pi$ and so $\alpha = 0$ giving as transformation $Z^{-1} = -z^{-1}$ which as we indicated converts a low-pass filter into a high-pass filter, both of the same bandwidth.

12.4.4.3 Low-pass to Band-pass and Low-pass to Band-stop Transformations

By being linear in both the numerator and the denominator, the LP-LP and LP-HP transformations preserve the number of poles and zeros of the prototype filter. To transform a low-pass filter into a band-pass or a band-stop filter, the number of poles and zeros must be doubled. For instance, if the prototype is a first-order low-pass filter (with real valued poles/zeros) we need a quadratic, rather than a linear, transformation in both numerator and denominator to obtain band-pass or band-stop filters from the low-pass filter since band-pass and band-stop filters cannot be first-order.

The low-pass to band-pass (LP-BP) transformation is

$$Z^{-1} = -\left(\frac{z^{-2} - bz^{-1} + c}{cz^{-2} - bz^{-1} + 1}\right) \tag{12.44}$$

while the low-pass to band-stop (LP-BS) transformation is

$$Z^{-1} = \frac{z^{-2} - (b/k)z^{-1} - c}{-cz^{-2} - (b/k)z^{-1} + 1} \tag{12.45}$$

where and

$$b = 2\alpha k/(k+1) \qquad c = (k-1)/(k+1)$$

$$\alpha = (\cos((\omega_{du} + \omega_{d\ell})/2))/(\cos((\omega_{du} - \omega_{d\ell})/2)) \quad k = \cot((\omega_{du} - \omega_{d\ell})/2))\tan(\theta_p/2)$$

The frequencies $\omega_{d\ell}$ and ω_{du} are the desired lower and higher cutoff frequencies in the band-pass and band-stop filters. Notice that when $\omega_{du} + \omega_{d\ell} = \pi$ (i.e., there is symmetry of the magnitude response around $\pi/2$) the above equations become simpler because $\alpha = 0$.

12.4.5 General IIR Filter Design with MATLAB

The following function *buttercheby*1 can be used to design low-pass, high-pass, band-pass, and band-stop Butterworth and Chebyshev filters. One important thing to remember when designing band-pass and band-stop filters is the order of the low-pass prototype is half that of the desired filter.

```
function [b,a,H,w]=buttercheby1(lp_order,wn,BC,type)
%
%   Design of frequency discriminating filters
%   using Butterworth and Chebyshev methods, the bilinear
    transformation and
%   frequency transformations
%
%   lp_order : order of lowpass filter prototype
%   wn : vector containing the cutoff normalized frequency(ies)
%    (entries must be normalized)
%   BC: Butterworth (0) or Chebyshev1 (1)
%   type : type of filter desired
%         1 = lowpass
%         2 = high-pass
%         3 = band-pass
%         4 = stopband
%    [b,a] : numerator, denominator coefficients of designed filter
%    [H,w] : frequency response, frequency range
%   USE:
%   [b,a,H,w]=buttercheby1(lp_order,wn,BC,type)
 if BC==0;                                          % Butterworth filter
    if type == 1
    [b,a]=butter(lp_order,wn);                      % lowpas
    elseif type == 2
    [b,a]=butter(lp_order,wn,'high');               % high-pass
    elseif type == 3
    [b,a]=butter(lp_order,wn);                      % band-pass
    else
    [b,a]=butter(lp_order,wn,'stop');               % stopband
    end
[H,w]=freqz(b,a,256);
else % Chebyshev1 filter
R=0.01;
if type == 1,
        [b,a]=cheby1(lp_order,R,wn);            % lowpas
        elseif type == 2,
        [b,a]=cheby1(lp_order,R,wn,'high');    % high-pass
        elseif type == 3,
        [b,a]=cheby1(lp_order,R,wn);            % band-pass
```

```
        else
        [b,a]=cheby1(lp_order,R,wn,'stop');    % stopband
end
[H,w]=freqz(b,a,256);
end
```

To illustrate the design of filters other than low-pass filters, consider the design of a Butterworth and Chebyshev band-stop filters of order $N = 30$ and half-power frequencies $[0.4\pi\, 0.6\pi]$. For the filter design we use the following script. Figure 12.17 shows the results.

```
%%
% Band-stop Butterworth
%%
figure(1)
[b1,a1]=buttercheby1(15,[0.4 0.6],0,4)
%%
% Band-stop Chebyshev
%%
figure(2)
[b2,a2]=buttercheby1(15,[0.4 0.6],1,4)
```

There are other filters that can be designed with MATLAB, following a procedure similar to the previous cases. For instance, to design a bandpass elliptic filter with cut-off frequencies $[0.45\pi\ \ 0.55\pi]$ of order 20 and with losses specifications of 0.1 and 40 dB in the passband and in the stopband, respectively, we use the script shown below. Likewise, to design a high-pass filter using the *cheby2* function we specify the order 10, the loss in the stopband and the cut-off frequency 0.55π and indicate it is a high-pass filter. The results are shown in Figure 12.18.

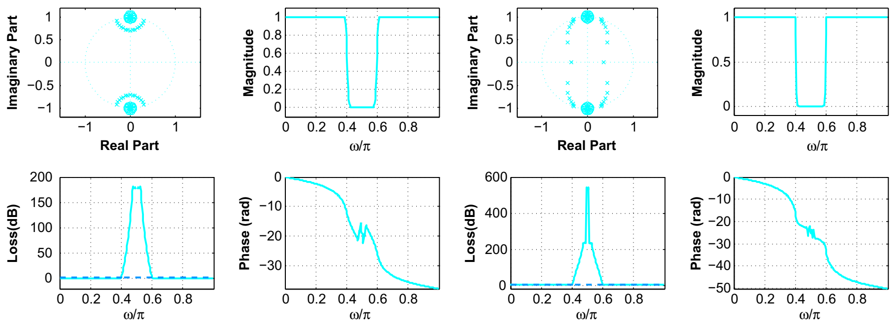

FIGURE 12.17

Bandstop Butterworth (left) and Chebyshev (right) filters: (clockwise for each side from top left) poles/zeros, magnitude, phase frequency responses, loss.

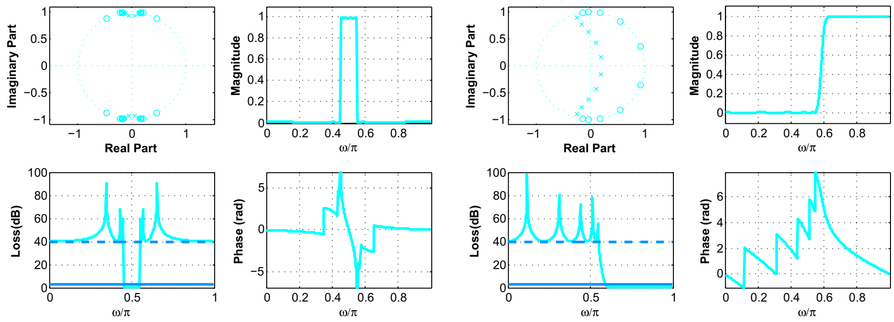

FIGURE 12.18
Elliptic band-pass filter (left) and high-pass using ***cheby2***: (clockwise for each side from top left) poles/zeros, magnitude, phase frequency responses, loss.

```
%%
% Elliptic and Cheby2
%%
[b1,a1]=ellip(10,0.1,40,[0.45 0.55]);
[b2,a2]=cheby2(10,40, 0.55,'high');
```

12.5 FIR FILTER DESIGN

The design of FIR filters is typically discrete. The specification of FIR filters is usually given in the time rather than in the frequency domain. FIR filters have three definite advantages: (i) stability, (ii) possible linear phase, and (iii) efficient implementation. Indeed, the poles of an FIR filter are at the origin of the Z-plane, thus FIR filters are stable. An FIR filter can be designed to have linear phase, and since the input/output equation of an FIR filter is equivalent to a convolution sum, FIR filters are implemented using the Fast Fourier Transform (FFT). A minor disadvantage is the storage required; typically FIR filters have a large number of coefficients.

■ Example 12.11

A moving average filter has an impulse response $h[n] = 1/M,\ 0 \le n \le M-1$, and zero otherwise. The transfer function of this filter is

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

Consider the stability of this filter, determine if the phase of this filter is linear and what type of filter it is.

Solution

The impulse response $h[n]$ is absolutely summable given its finite length M, thus the filter is BIBO stable. Notice that $H(1)$ is $0/0$, according to the final expression above, indicating that a pole and a zero at $z = 1$ exist, but also from the sum $H(1) = 1$, so there are no poles at $z = 1$. The zeros of $H(z)$ are values that make $z^M - 1 = 0$, or $z_k = e^{j2\pi k/M}$ for $k = 0, \ldots, M-1$. When $k = 0$ the zero $z = 1$ cancels the pole at 1, thus

$$H(z) = \frac{(z-1)\prod_{k=1}^{M-1}(z - e^{j2\pi k/M})}{Mz^{M-1}(z-1)} = \frac{\prod_{k=1}^{M-1}(z - e^{j2\pi k/M})}{Mz^{M-1}}.$$

To convince yourself of the pole zero cancelation let $M = 3$ for which

$$H(z) = \frac{1}{3}\frac{z^3 - 1}{z^2(z-1)} = \frac{1}{3}\frac{(z^2+z+1)(z-1)}{z^2(z-1)} = \frac{z^2+z+1}{3z^2}$$

showing the pole-zero cancelation.

Since the zeros of the filter are on the unit circle, the phase of this filter is not linear. Although the filter is considered a low-pass filter, it is of very poor quality in terms of its magnitude response. ■

12.5.1 Window Design Method

The usual filter specifications of magnitude and linear phase can be translated into a time-domain specification (i.e., a desired impulse response) by means of the discrete-time Fourier transform. In this section, we will show how to design FIR filters using this specification with the **window method**. You will see that this is a trial-and-error method, as there is no measure of how close the designed filter is to the desired response, and that using different windows we obtain different results.

Let $H_d(e^{j\omega})$ be the desired frequency response of an ideal discrete low-pass filter. Assume that the phase of $H_d(e^{j\omega})$ is zero. The desired impulse response is given by the inverse discrete-time Fourier transform:

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}\, d\omega \quad -\infty < n < \infty \tag{12.46}$$

which is non-causal and of infinite length. The filter

$$H_d(z) = \sum_{n=-\infty}^{\infty} h_d[n]z^{-n} \tag{12.47}$$

is thus not an FIR filter. To obtain an FIR filter that approximates $H_d(e^{j\omega})$ we need to window the impulse response $h_d[n]$ to get finite length, and then to delay the resulting windowed impulse response to achieve causality.

For an odd integer N, define

$$h_w[n] = h_d[n]w[n] = \begin{cases} h_d[n] & -(N-1)/2 \le n \le (N-1)/2 \\ 0 & \text{elsewhere} \end{cases}$$

where w$[n]$ is a **rectangular window**

$$\mathrm{w}[n] = \begin{cases} 1 & -(N-1)/2 \leq n \leq (N-1)/2 \\ 0 & \text{otherwise} \end{cases} \tag{12.48}$$

that causes the truncation of $h_d[n]$. The windowed impulse response $h_{\mathrm{w}}[n]$ has a discrete-time Fourier transform

$$H_{\mathrm{w}}(e^{j\omega}) = \sum_{n=-(N-1)/2}^{(N-1)/2} h_{\mathrm{w}}[n]e^{-j\omega n}$$

For a large value of N we have that $H_{\mathrm{w}}(e^{j\omega})$ must be a good approximation of $H_d(e^{j\omega})$, i.e., for a certain large N we could have

$$|H_{\mathrm{w}}(e^{j\omega})| \approx |H_d(e^{j\omega})|, \quad \angle H_{\mathrm{w}}(e^{j\omega}) = \angle H_d(e^{j\omega}) = 0 \tag{12.49}$$

It is not clear how the value of N should be chosen—this is what we meant by that this design is a trial-and-error method.

For the N value that makes possible (12.49), to convert $H_{\mathrm{w}}(z)$ into a causal filter, we delay the impulse response $h_{\mathrm{w}}[n]$ by $(N-1)/2$ samples to obtain

$$\begin{aligned} \hat{H}(z) &= H_{\mathrm{w}}(z)z^{-(N-1)/2} = \sum_{m=-(N-1)/2}^{(N-1)/2} h_{\mathrm{w}}[m]z^{-(m+(N-1)/2)} \\ &= \sum_{n=0}^{N-1} h_d[n-(N-1)/2]\mathrm{w}[n-(N-1)/2]z^{-n} \end{aligned}$$

after letting $n = m + (N-1)/2$. For a large value of N, we have

$$\begin{aligned} |\hat{H}(e^{j\omega})| &= |H_{\mathrm{w}}(e^{j\omega})e^{-j\omega(N-1)/2}| = |H_{\mathrm{w}}(e^{j\omega})| \approx |H_d(e^{j\omega})| \\ \angle\hat{H}(e^{j\omega}) &= \angle H_{\mathrm{w}}(e^{j\omega}) - \frac{N-1}{2}\omega = -\frac{N-1}{2}\omega \end{aligned} \tag{12.50}$$

since $\angle H_{\mathrm{w}}(e^{j\omega}) = \angle H_d(e^{j\omega}) = 0$, according to (12.49). That is, the magnitude response of the FIR filter $\hat{H}(z)$ is approximately (depending on the value of N) the desired response and its phase response is linear. These results can be generalized as follows.

General Window Method For FIR Design

- If the desired low-pass frequency response has a magnitude

$$|H_d(e^{j\omega})| = \begin{cases} 1 & -\omega_c \leq \omega \leq \omega_c \\ 0 & \text{otherwise} \end{cases} \tag{12.51}$$

and a linear phase

$$\theta(\omega) = -\omega M/2$$

The corresponding impulse response is given by

$$h_d[n] = \begin{cases} \sin(\omega_c(n - M/2)/(\pi(n - M/2)) & n \neq M/2 \\ \omega_c/\pi & n = M/2 \text{ M even, else not defined} \end{cases} \tag{12.52}$$

Using a window w$[n]$ of length M and centered at $M/2$, the windowed impulse response is

$$h[n] = h_d[n]\mathrm{w}[n]$$

and the designed FIR filter is

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

- The design using windows is a trial-and-error procedure. Different tradeoffs can be obtained by using various windows and various lengths of the windows.
- The symmetry of the impulse response $h[n]$ with respect to $M/2$, independent of whether this is an integer or not, guarantees the linear phase of the filter.

12.5.2 Window Functions

In the previous section, the windowed impulse response $h_{\mathrm{w}}[n]$ was written $h_{\mathrm{w}}[n] = h_d[n]\mathrm{w}[n]$, where

$$\mathrm{w}[n] = \begin{cases} 1 & -(N-1)/2 \leq n \leq (N-1)/2 \\ 0 & \text{otherwise} \end{cases} \tag{12.53}$$

is a **rectangular window** of length N. If we wish $H_{\mathrm{w}}(e^{j\omega}) = H_d(e^{j\omega})$, we would need a rectangular window of infinite length so that the impulse responses $h_{\mathrm{w}}[n] = h_d[n]$, i.e., no windowing. This ideal rectangular window has a discrete-time Fourier transform

$$W(e^{j\omega}) = 2\pi\delta(\omega) \quad -\pi \leq \omega < \pi \tag{12.54}$$

Since $h_{\mathrm{w}}[n] = \mathrm{w}[n]h_d[n]$, a product of functions of n, then $H_{\mathrm{w}}(e^{j\omega})$ is the convolution of $H_d(e^{j\omega})$ and $W(e^{j\omega})$ in the frequency domain, i.e.,

$$H_{\mathrm{w}}(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\theta})W(e^{j(\omega-\theta)})d\theta = \int_{-\pi}^{\pi} H_d(e^{j\theta})\delta(\omega-\theta)d\theta = H_d(e^{j\omega})$$

Thus for $N \to \infty$, the result of this convolution is $H_d(e^{j\omega})$, but if N is finite the convolution in the frequency domain would give a distorted version of $H_d(e^{j\omega})$. Thus to obtain a good approximation of $H_d(e^{j\omega})$ using a finite window w$[n]$ the window must have a spectrum approximating that of the ideal rectangular window, an impulse in frequency in $-\pi \leq \omega < \pi$ as in (12.54) with most of the energy concentrated in the low frequencies. The smoothness of the window makes this possible. Example of windows that are smoother than the rectangular window are:

Triangular or Bartlett Window

$$w[n] = \begin{cases} 1 - \frac{2|n|}{N-1} & -(N-1)/2 \leq n \leq (N-1)/2 \\ 0 & \text{otherwise} \end{cases} \tag{12.55}$$

Hamming Window

$$w[n] = \begin{cases} 0.54 + 0.46\cos(2\pi n/(N-1)) & -(N-1)/2 \leq n \leq (N-1)/2 \\ \\ 0 & \text{otherwise} \end{cases} \tag{12.56}$$

Kaiser Window. This window has a parameter β which can be adjusted. It is given by

$$w[n] = \begin{cases} \frac{I_0\left(\beta\sqrt{1-(n/(N-1)^2)}\right)}{I_0(\beta)} & -(N-1)/2 \leq n \leq (N-1)/2 \\ 0 & \text{otherwise} \end{cases} \tag{12.57}$$

where $I_0(x)$ is the zeroth-order Bessel function of the first kind which can be computed by the series

$$I_0(x) = 1 + \sum_{k=1}^{\infty}\left(\frac{(0.5x)^k}{k!}\right)^2 \tag{12.58}$$

When $\beta = 0$ the Kaiser window coincides with a rectangular window, since $I_0(0) = 1$. As β increases the window becomes smoother.

The above definitions are for windows symmetric with respect to the origin. Figures 12.19 and 12.20 show the causal rectangular, Bartlett, Hamming, and Kaiser windows, and their magnitude spectra. Given that the side lobes for the Kaiser window have the largest loss, the Kaiser window is considered the best of these four, followed by the Hamming, the Bartlett, and the rectangular windows. Notice that the width of the first lobe is the widest for the Kaiser and the narrowest for the rectangular, indicating the Kaiser window is very smooth and most of its energy is in the low frequencies and the opposite for the rectangular window.

12.5.3 Linear Phase and Symmetry of the Impulse Response

The linear phase is a result of the symmetry of the impulse response of the designed filter. As we show next, if the impulse response $h[n]$ of the FIR filter is even or odd symmetric with respect to a middle sample the filter has linear phase.

Consider the following cases for an FIR filter, of order $M-1$ or length M, with transfer function

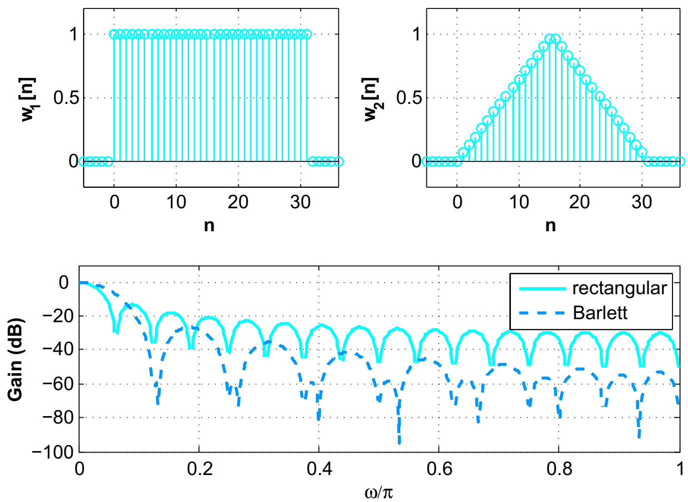

FIGURE 12.19

Rectangular and Bartlett causal windows and their spectra.

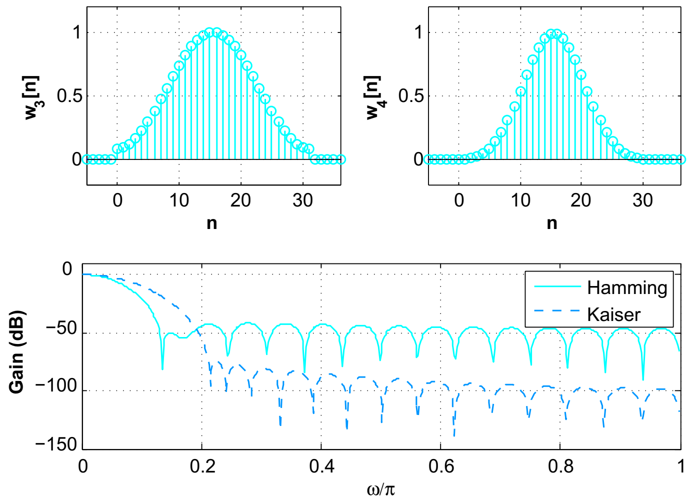

FIGURE 12.20

Hamming and Kaiser causal windows and their spectra.

$$H(z) = \sum_{n=0}^{M-1} h[n] z^{-n} \tag{12.59}$$

1. M is odd, i.e., $M = 2N + 1$. Assume that the impulse response $h[n]$ is symmetric with respect to the N value:

$$h[n] = h[2N - n] \quad n = 0, \ldots, N \tag{12.60}$$

or $h[0] = h[2N], h[1] = h[2N-1], \ldots, h[N]$ equal to itself, then we rewrite (12.59) as

$$\begin{aligned} H(z) &= z^{-N}\sum_{n=0}^{2N} h[n]z^{N-n} = z^{-N}\left[\sum_{n=0}^{N-1} h[n]z^{N-n} + h[N] + \sum_{n=N+1}^{2N} h[n]z^{N-n}\right] \\ &= z^{-N}\left[\sum_{n=0}^{N-1} h[n]z^{N-n} + h[N] + \sum_{m=0}^{N-1} h[2N-m]z^{-N+m}\right] \\ &= z^{-N}\left[h[N] + \sum_{n=0}^{N-1} h[n](z^{N-n} + z^{-(N-n)})\right] \end{aligned}$$

where we let $m = 2N - n$ and used the symmetry of the impulse response. The frequency response is then

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

and since $A(e^{j\omega})$ is real, then

$$\angle H(e^{j\omega}) = \begin{cases} -\omega N & A(e^{j\omega}) \geq 0 \\ -\omega N - \pi & A(e^{j\omega}) < 0 \end{cases} \tag{12.61}$$

that is, the group delay $\tau(\omega) = -d\angle H(e^{j\omega})/d\omega = N$ is constant so the phase is linear.

2. M is even, i.e., $M = 2N$. Assume the impulse response in this case is symmetric with respect to the value in between samples $N-1$ and N, or $N - 0.5$:

$$h[n] = h[2N-1-n] \quad n = 0, \ldots, N-1 \tag{12.62}$$

or $h[0] = h[2N-1], h[1] = h[2N-2], \ldots, h[N-1] = h[N]$. Using a similar approach to the even case, we find the frequency response to be

$$H(e^{j\omega}) = e^{-j(N-0.5)\omega}\sum_{n=0}^{N-1} 2h[n]\cos((N-0.5-n)\omega) = e^{-j(N-0.5)\omega}B(e^{j\omega})$$

and since $B(e^{j\omega})$ is real then the phase of the FIR filter in this case is

$$\angle H(e^{j\omega}) = \begin{cases} -(N-0.5)\omega & B(e^{j\omega}) \geq 0 \\ -(N-0.5)\omega - \pi & B(e^{j\omega}) < 0 \end{cases} \tag{12.63}$$

The group delay $\tau(\omega) = -d\angle H(e^{j\omega})/d\omega = N - 0.5$ is constant so that the phase is considered linear.

3. It is possible to have odd symmetry in the impulse response for the above two cases:

- M is odd, i.e., $M = 2N + 1$. Assume that the impulse response $h[n]$ is odd symmetric with respect to the N value:

$$h[n] = -h[2N - n] \quad n = 0, \ldots, N \tag{12.64}$$

or $h[0] = -h[2N], h[1] = -h[2N - 1], \ldots, h[N] = h[-N] = 0$. As above, the frequency response of the FIR filter is

$$H(e^{j\omega}) = -je^{-j\omega N}\left[\sum_{n=0}^{N-1} 2h[n]\sin((N - n)\omega)\right] = -je^{-j\omega N}C(e^{j\omega})$$

and since $C(e^{j\omega})$ is real, then

$$\angle H(e^{j\omega}) = \begin{cases} -\omega N - \pi/2 & C(e^{j\omega}) \geq 0 \\ -\omega N - 3\pi/2 & C(e^{j\omega}) < 0 \end{cases} \tag{12.65}$$

that is, the phase of the filter is linear.

- M is even, i.e., $M = 2N$. Assume the impulse response in this case is odd symmetric with respect to the value in between samples $N - 1$ and N or $N - 0.5$:

$$h[n] = -h[2N - 1 - n] \quad n = 0, \ldots, N - 1 \tag{12.66}$$

or $h[0] = -h[2N - 1], h[1] = -h[2N - 2], \ldots, h[N - 1] = -h[N]$. The frequency response is found to be

$$H(e^{j\omega}) = je^{-j(N-0.5)\omega}\sum_{n=0}^{N-1} 2h[n]\sin((N - 0.5 - n)\omega) = je^{-j(N-0.5)\omega}D(e^{j\omega})$$

and since $D(e^{j\omega})$ is real then the phase of the FIR filter in this case is

$$\angle H(e^{j\omega}) = \begin{cases} -(N - 0.5)\omega + \pi/2 & D(e^{j\omega}) \geq 0 \\ -(N - 0.5)\omega - \pi/2 & D(e^{j\omega}) < 0 \end{cases} \tag{12.67}$$

or the phase is linear.

■ Example 12.12

Design a low-pass FIR filter of length $M = 21$ to be used in filtering analog signals and that approximates the following ideal frequency response

$$H_d(e^{jf}) = \begin{cases} 1 & -125 \leq f \leq 125 \text{ Hz} \\ 0 & \text{elsewhere in } -f_s/2 < f \leq f_s/2 \end{cases}$$

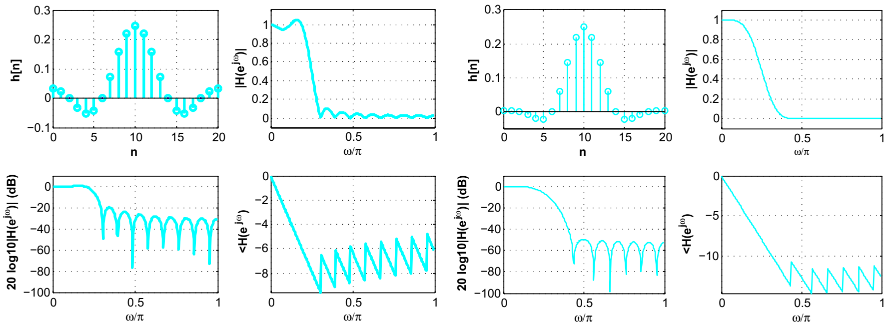

FIGURE 12.21
Low-pass FIR filters using rectangular (left) and Hamming windows.

and $f_s = 1000$ Hz is the sampling rate. Use first a rectangular window, and then a Hamming window. Compare the designed filters.

Solution

Using $\omega = 2\pi f/f_s$, the discrete frequency response is given by

$$H_d(e^{j\omega}) = \begin{cases} 1 & -\pi/4 \le \omega \le \pi/4 \text{ rad} \\ 0 & \text{elsewhere in } -\pi < \omega \le \pi \end{cases}$$

The desired impulse response is thus

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega = \frac{1}{2\pi}\int_{-\pi/4}^{\pi/4} e^{j\omega n}d\omega = \begin{cases} \sin(\pi n/4)/(\pi n) & n \neq 0 \\ 0.25 & n = 0 \end{cases}$$

Using a rectangular window, the FIR filter is then of the form (since $M = 2N + 1 = 21$, the delay is $N = 10$):

$$\hat{H}(z) = H_w(z)z^{-10} = \sum_{n=0}^{20} h_d[n-10]z^{-n} = 0.25z^{-10} + \sum_{n=0,n\neq 10}^{20} \frac{\sin(\pi(n-10)/4)}{\pi(n-10)}z^{-n}$$

The magnitude and phase of this filter are shown in Figure 12.21 when we use a rectangular (left) and a Hamming window (right).

The magnitude and phase responses of the filter designed using the Hamming window are much improved over the ones obtained using the rectangular window. Notice that the second lobe in the stopband for the Hamming window design is at about -50 dB while for the rectangular window design is at about -20 dB, a significant difference. In both cases, the phase response is linear in the passband of the filter, corresponding to the impulse response $h[n]$ being symmetric with respect to the $n = 10$ sample. ■

■ Example 12.13

Design a high-pass filter of order $M-1=14$, and cutoff frequency 0.2π using the Kaiser window. Use MATLAB.

Solution

Let $h_{lp}[n]$ be the impulse response of an ideal low-pass filter

$$H_{lp}(e^{j\omega}) = \begin{cases} 1 & -\omega_c \leq \omega \leq \omega_c \\ 0 & \text{otherwise in } [-\pi, \pi) \end{cases}$$

According to the modulation property of the DTFT, we have that

$$2h_{lp}[n]\cos(\omega_0 n) \Leftrightarrow H_{lp}(e^{j(\omega+\omega_0)}) + H_{lp}(e^{j(\omega-\omega_0)})$$

If we let $\omega_0 = \pi$ then the right term gives a high-pass filter, and so $h_{hp}[n] = 2h_{lp}[n]\cos(\pi n) = 2(-1)^n h_{lp}[n]$ is the desired impulse response of the high-pass filter. The following script shows how to use our function *fir* to design the high-pass filter

```
%%
% Example 12.13---FIR filter from 'fir'
%%
No=14;wc=0.2;wo=1;wind=4;
[b]=fir(No,wc,wo,wind);
[H,w]=freqz(b,1,256);
```

The results are shown in Figure 12.22. Notice the symmetry of the impulse response with respect to sample at $n=7$ gives linear phase in the passband of the high-pass filter. The second lobe of the gain in dB is about -50 dB. ■

Our function *fir* can be used to design low-pass, high-pass, and band-pass FIR filters using different types of windows. When designing high-pass and band-pass FIRs, *fir* first designs a prototype low-pass filter and then uses the modulation property to shift it in frequency to a desired center frequency.

```
function [b]=fir(N,wc,wo,wind)
%
% FIR filter design using window method and
% frequency modulation
%
% N : order of the FIR filter
% wc : normalized cutoff frequency (between 0 and 1)
% of low-pass prototype
```

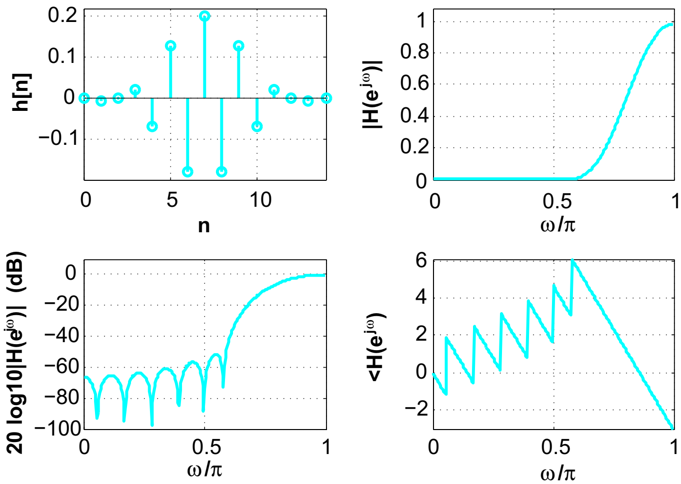

FIGURE 12.22
High-pass FIR filter design using Kaiser window.

```
%  wo : normalized center frequency (between 0 and 1)
%  of high-pass, band-pass filters
%  wind : type of window function
%     1 : rectangular
%     2 : hanning
%     3 : hamming
%     4 : kaiser
%  [b] : coefficients of designed filter
%
%  USE:
%  [b]=fir(N,wc,wo,wind)
%
n=0:N;
if wind ==1
window=boxcar(N+1);
disp('***** RECTANGULAR WINDOW *****')
elseif wind ==2
window=hanning(N+1);
disp('*****HANNING WINDOW *****')
elseif wind == 3
window=hamming(N+1);
disp('***** HAMMING WINDOW *****')
```

```
else
window=kaiser(N+1,4.55);
disp('***** KAISER WINDOW *****')
end
% calculation of ideal impulse response
den=pi * (n-N/2);
num=sin(wc * den);
% if N even, this prevents 0/0
if fix(N/2)==N/2,
num(N/2+1)=wc;
den(N/2+1)=1;
end
b=(num./den).  * window';
% frequency shifting
[H,w]=freqz(b,1,256);  % low-pass
if wo>0 & wo<1,
b=2 * b. * cos(wo * pi *  (n-N/2))/H(1);
elseif wo==0,
    b=b/abs(H(1));
elseif wo==1;
    b=b. * cos(wo * pi * (n-N/2));
end
```

MATLAB provides the function fir1 to design FIR filters with the window method. As expected, the results are identical for either fir1 and ours. The reason for writing fir is to simplify the code and to show how the modulation property can be used in the design of filters different from low-pass.

12.6 REALIZATION OF DISCRETE FILTERS

The realization of a discrete filter can be done in hardware or in software. In either case, the implementation of the transfer function $H(z)$ of a discrete filter requires delays, adders, and constant multipliers as actual hardware or as symbolic components. Figure 10.13 in Chapter 10, depicts the operation of each of these components as block diagrams.

In choosing a structure over another to realize a filter, two factors to consider are:

1. **Computational complexity** which relates to the number of operations (mainly multiplications and additions), but more importantly to the number of delays used. The aim is to obtain minimal realizations.

2. **Quantization effects** or the representation of filter parameters using finite length registers. The aim is to minimize quantization effects on parameters and on operations.We will consider here the computational complexity of the structures seeking to obtain minimum realizations, i.e., to optimize the number of delays used. The quantization effects are not considered.

12.6.1 Realization of IIR Filters

The structures commonly used to realize IIR filters are:

1. Direct Form
2. Cascade
3. Parallel

The direct form represents the difference equation resulting from the transfer function of the IIR filter while attempting to minimize the number of delays. The cascade and parallel structures are based on the product or sum of first- and second-order filters to express the filter transfer function, which are in turn implemented using a direct form.

12.6.1.1 Direct Form Realization

Given the transfer function of a causal IIR filter

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{1 + \sum_{k=1}^{N-1} a_k z^{-k}}, \quad M \leq N \tag{12.68}$$

where $Y(z)$ and $X(z)$ are the z-transforms of the output $y[n]$ and the input $x[n]$, the input/output relationship is given by the difference equation

$$y[n] = -\sum_{k=1}^{N-1} a_k y[n-k] + \sum_{k=0}^{M-1} b_k x[n-k] \tag{12.69}$$

The direct form attemps to realize this equation with no more than $N-1$ delays. Assuming the input $x[n]$ is available, then $M-1$ delays are needed to generate the delayed inputs $\{x[n-k]\}$, $k = 1, \ldots, M-1$, and realizing the output components requires additional $N-1$ delays. Thus a direct realization requires $M+N-2$ delays for an $(N-1)$th-order difference equation. Such a realization is not minimal.

The direct form provides minimal realizations—the number of delays coincides with the order of the system—for filters with a constant numerator transfer function. Indeed, if

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0}{1 + \sum_{k=1}^{N-1} a_k z^{-k}} \tag{12.70}$$

the input/output relationship is given by the difference equation

$$y[n] = -\sum_{k=1}^{N-1} a_k y[n-k] + b_0 x[n] \tag{12.71}$$

which only requires $N-1$ delays for the output, and none for the input. This is a minimal realization of $H(z)$ as only $N-1$ delays are needed.

If the numerator of the transfer function $H(z)$ of the filter we wish to realize is not a constant, but a polynomial

$$B(z) = \sum_{k=0}^{M-1} b_k z^{-k}$$

as in Equation (12.68), we need to factor $H(z)$ into a transfer function $B(z)$ and a constant numerator transfer function $1/A(z)$, where

$$A(z) = \sum_{k=0}^{N-1} a_k z^{-k}$$

is the denominator of $H(z)$. We then have that

$$Y(z) = H(z)X(z) = B(z)\left[\frac{X(z)}{A(z)}\right] \tag{12.72}$$

Defining an output $w[n]$ with $W(z) = X(z)/A(z)$, corresponding to the second term in the last equation, we obtain a constant-numerator IIR filter with transfer function

$$\frac{W(z)}{X(z)} = \frac{1}{A(z)} \Rightarrow w[n] = -\sum_{k=1}^{N-1} a_k w[n-k] + b_0 x[n]$$

or a difference equation which only requires $N-1$ delays for the output, and none for the input. The output $y[n]$ is then obtained from

$$Y(z) = B(z)W(z) \Rightarrow y[n] = \sum_{k=0}^{M-1} b_k w[n-k]$$

an input-output equation which uses the delayed signals $\{w[n-k]\}$ from above. Since the filter is causal, $M \leq N$, and no additional delays are required, the number of delays used corresponds to the order of the denominator $A(z)$ which is the order of the filter.

■ Example 12.14

Consider the transfer function

$$H(z) = \frac{1 + 1.5z^{-1}}{1 + 0.1z^{-1}}$$

of a first-order IIR filter. Find a minimal direct realization for it.

Solution

The transfer function corresponds to a system with a first-order difference equation

$$y[n] = x[n] + 1.5x[n-1] - 0.1y[n-1]$$

so $M = N = 2$ and this equation can be realized with $M + N - 2 = 2$ delays—not a minimal realization.

To obtain a minimal realization we let

$$W(z) = \frac{X(z)}{1 + 0.1z^{-1}} \Rightarrow w[n] = x[n] - 0.1w[n-1]$$
$$Y(z) = (1 + 1.5z^{-1})W(z) \Rightarrow y[n] = w[n] + 1.5w[n-1]$$

which gives the minimal direct realization shown in Figure 12.23.

To obtain the transfer function from the minimal direct realization we need to obtain the transfer function corresponding to the constant numerator IIR filter first, then that of the FIR and use their Z-tranforms. Thus,

$$w[n] = x[n] - 0.1w[n-1]$$
$$y[n] = w[n] + 1.5w[n-1]$$

If we replace the first equation into the second we obtain an expression containing $w[n]$ and $w[n-2]$ and $x[n]$ so that we cannot express $y[n]$ directly in term of the input. Instead, the Z-transforms of the above equations are

$$(1 + 0.1z^{-1})W(z) = X(z)$$
$$Y(z) = (1 + 1.5z^{-1})W(z)$$

from which we obtain the transfer function $H(z)$. ■

Since the cascade and the parallel realizations will connect first and second-order systems to realize a given transfer function, overall minimal realizations can be obtained by using minimal direct realizations for first- and second-order filters. For a second-order filter with a general transfer function with constant coefficients

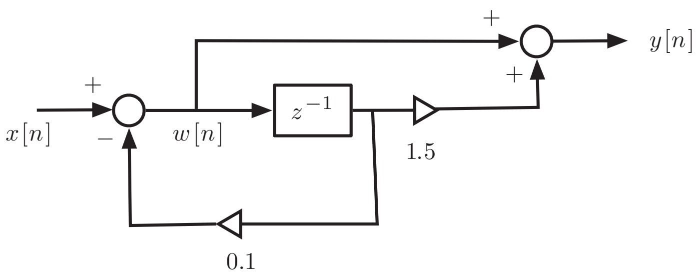

FIGURE 12.23
Minimal direct realization of $H(z) = (1 + 1.5z^{-1})/(1 + 0.1z^{-1})$.

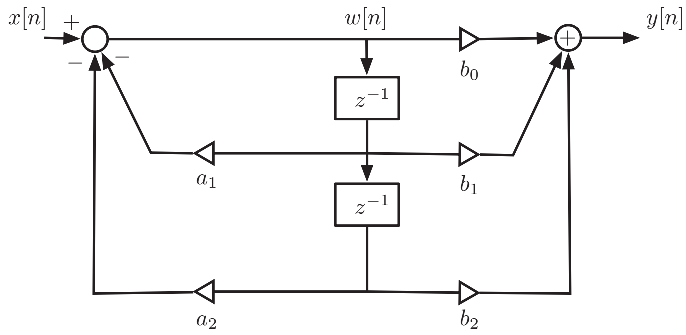

FIGURE 12.24
Minimal direct realization of first- and second-order filters (for the first-order filters let $a_2 = b_2 = 0$, eliminate the constant multipliers and the lower delay).

$$H_2(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{12.73}$$

a minimal direct realization is shown in Figure 12.24. A minimal realization for a first-order filter is obtained from this by getting rid of the constant multipliers corresponding to b_2 and a_2 and the lower delay, corresponding to letting $b_2 = a_2 = 0$ when we obtain a general transfer function for a first-order system from $H_2(z)$.

To obtaining the transfer function from a minimal direct realization, obtain the equations for the IIR and the FIR components and apply the Z-transform.

12.6.1.2 Cascade Realization

The cascade realization is obtained by representing the given transfer function $H(z) = B(z)/A(z)$ as a product of first- and second-order filters $H_i(z)$ with real coefficients:

$$H(z) = \prod_i H_i(z) \tag{12.74}$$

Each transfer function $H_i(z)$ is realized using the minimal direct form and cascaded. Different from the analog case, this cascade realization is not constrained by loading.

■ Example 12.15

Obtain a cascade realization of the filter with transfer function

$$H(z) = \frac{3 + 3.6z^{-1} + 0.6z^{-2}}{1 + 0.1z^{-1} - 0.2z^{-2}}$$

Solution

The poles of $H(z)$ are $z = -0.5$ and $z = 0.4$ and the zeros $z = -1$ and $z = -0.2$ all of which are real. One way of obtaining the cascade realization is to express $H(z)$ as

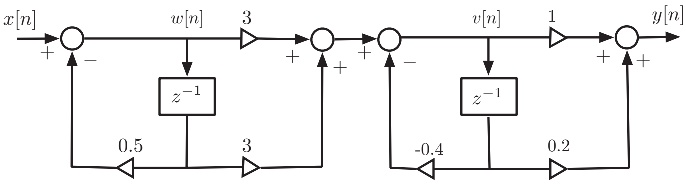

FIGURE 12.25
Cascade realization of $H(z) = (3 + 3.6z^{-1} + 0.6z^{-2})/(1 + 0.1z^{-1} - 0.2z^{-2})$.

$$H(z) = \left[\frac{3(1 + z^{-1})}{1 + 0.5z^{-1}}\right]\left[\frac{1 + 0.2z^{-1}}{1 - 0.4z^{-1}}\right]$$

If we let

$$H_1(z) = \frac{3(1 + z^{-1})}{1 + 0.5z^{-1}}, \quad H_2(z) = \frac{1 + 0.2z^{-1}}{1 - 0.4z^{-1}}$$

realizing each separately and then cascading them we obtain the realization shown in Figure 12.25.

It is also possible to express $H(z)$ as

$$H(z) = \underbrace{\left[\frac{1 + 0.2z^{-1}}{1 + 0.5z^{-1}}\right]}_{\hat{H}_1(z)}\underbrace{\left[\frac{3(1 + z^{-1})}{1 - 0.4z^{-1}}\right]}_{\hat{H}_2(z)}$$

which would give a different but equivalent realization of $H(z)$.

Since loading is not applicable when cascading discrete filters, the product of the transfer functions always gives the overall transfer function. As LTI systems these realizations can be cascaded in different orders with the same result.

Example 12.16

Obtain a cascade realization of

$$H(z) = \frac{1 + 1.2z^{-1} + 0.2z^{-2}}{1 - 0.4z^{-1} + z^{-2} - 0.4z^{-3}}$$

Solution

The zeros of $H(z)$ are $z = -1$ and $z = -0.2$, while its poles are $z = \pm j$ and $z = 0.4$. We can thus rewrite $H(z)$ as either

$$\begin{aligned} H(z) &= \left[\frac{1 + z^{-1}}{1 + z^{-2}}\right]\left[\frac{1 + 0.2z^{-1}}{1 - 0.4z^{-1}}\right] \\ &= \left[\frac{1 + 0.2z^{-1}}{1 + z^{-2}}\right]\left[\frac{1 + z^{-1}}{1 - 0.4z^{-1}}\right] \end{aligned}$$

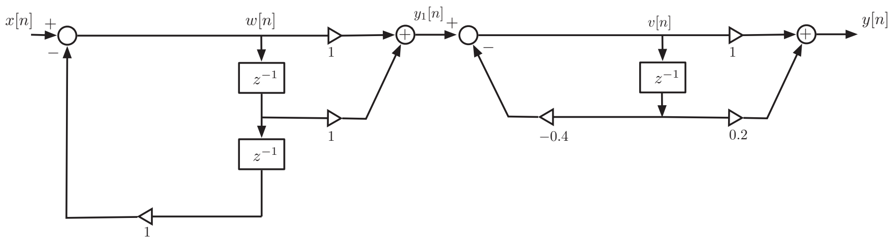

FIGURE 12.26
Cascade realization of $H(z) = [(1+z^{-1})/(1+z^{-2})]\,[(1+0.2z^{-1})/(1-0.4z^{-1})]$.

where the complex conjugate poles give the denominator of the first filter. Realizing each of these components and cascading in any order would give different but equivalent representation of $H(z)$. Figure 12.26 shows the realization of the top form of $H(z)$. ■

12.6.1.3 Parallel Realization

In this case the given transfer function $H(z)$ is represented as a partial fraction expansion

$$H(z) = \frac{B(z)}{A(z)} = C + \sum_{i=1}^{r} H_i(z) \tag{12.75}$$

where C is a constant and the filters $H_i(z)$ are first- or second-order with real coefficients that are implemented with the minimal direct realization.

The constant C in the expansion is needed when the numerator (in positive powers of z) is of larger or equal order than the denominator. If the numerator is of larger order than the denominator, the filter is non-causal. To illustrate this consider a first-order filter with a transfer function where the numerator is of second order (in terms of positive powers of z)

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 z^2 + b_1 z + b_2}{z + a_1} = \frac{b_0 z + b_1 + b_2 z^{-1}}{1 + a_1 z^{-1}}$$

where we multiplied the numerator and denominator by z^{-1} to be able to obtain the difference equation. The difference equation representing this system is

$$y[n] = -a_1 y[n-1] + b_0 x[n+1] + b_1 x[n] + b_2 x[n-1]$$

requiring a future input $x[n+1]$ to compute the present $y[n]$, i.e., corresponding to a non-causal filter.

The cascade and parallel realizations are shown in Figure 12.27.

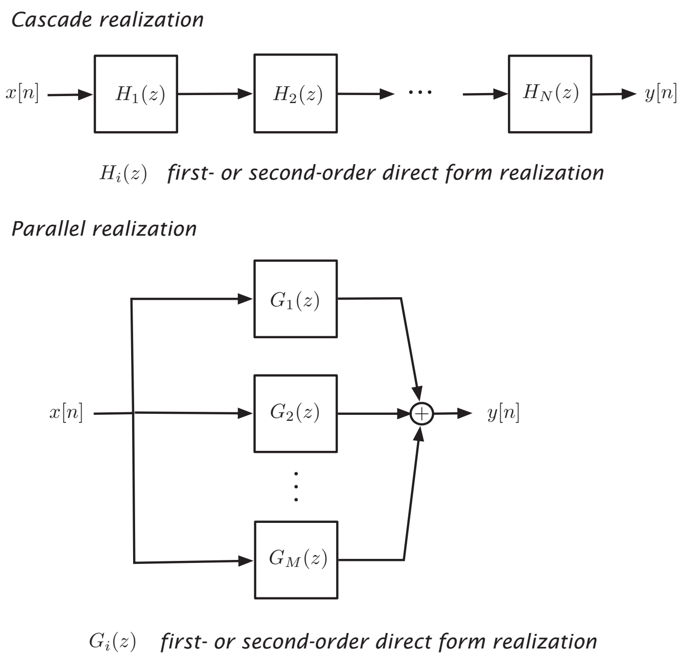

FIGURE 12.27
Cascade and parallel realizations of IIR filters.

■ Example 12.17

Let

$$H(z) = \frac{3 + 3.6z^{-1} + 0.6z^{-2}}{1 + 0.1z^{-1} - 0.2z^{-2}} = \frac{3z^2 + 3.6z + 0.6}{z^2 + 0.1z - 0.2}$$

Obtain a parallel realization.

Solution

The transfer function $H(z)$ is not proper rational, in either positive or negative powers of z, and the poles are $z = -0.5$ and $z = 0.4$. Thus the transfer function can be expanded as

$$H(z) = A_1 + \frac{A_2}{1 + 0.5z^{-1}} + \frac{A_3}{1 - 0.4z^{-1}}$$

In this case we need A_1 because the numerator, in positive as well as in negative powers of z, is of the same order as the denominator. We then have

$$\begin{aligned}
A_1 &= H(z)|_{z=0} = -3 \\
A_2 &= H(z)(1 + 0.5z^{-1})|_{z^{-1}=-2} = -1 \\
A_3 &= H(z)(1 - 0.4z^{-1})|_{z^{-1}=2.5} = 7
\end{aligned}$$

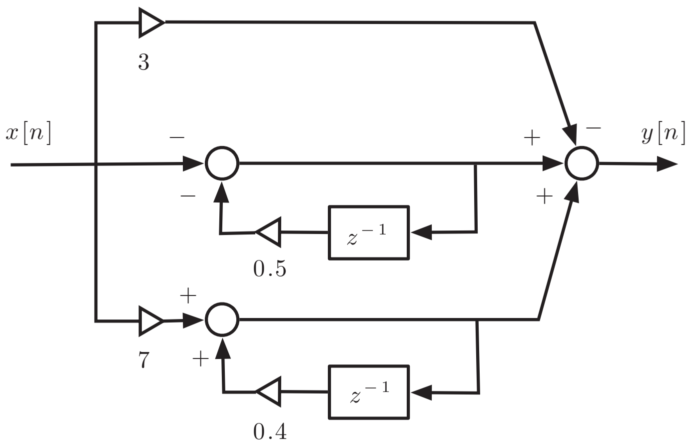

FIGURE 12.28
Parallel realization for $H(z) = (3 + 3.6z^{-1} + 0.6z^{-2})/(1 + 0.1z^{-1} - 0.2z^{-2})$.

Letting

$$H_1(z) = \frac{-1}{1 + 0.5z^{-1}}, \quad H_2(z) = \frac{7}{1 - 0.4z^{-1}}$$

we obtain the parallel realization for $H(z)$ shown in Figure 12.28. ■

12.6.2 Realization of FIR Filters

The realization of FIR filters can be done using direct and cascade forms. Since these filters are non-recursive, there is no way to implement FIR filters in parallel.

The direct realization of an FIR filter consists in realizing the input/output equation using delays, constant multipliers, and summers. For instance, if the transfer function of an FIR filter is

$$H(z) = \sum_{k=0}^{M} b_k z^{-k} \tag{12.76}$$

The Z-transform of the filter output can be written as $Y(z) = H(z)X(z)$ where $X(z)$ is the Z-transform of the filter input. In the time domain we have

$$y[n] = \sum_{k=0}^{M} b_k x(n-k)$$

which can be realized as shown in Figure 12.29 for $M = 3$. Notice that M is the number of delays needed and that there are $M + 1$ taps, which has given the name of tapped filters to FIR filters realized this way.

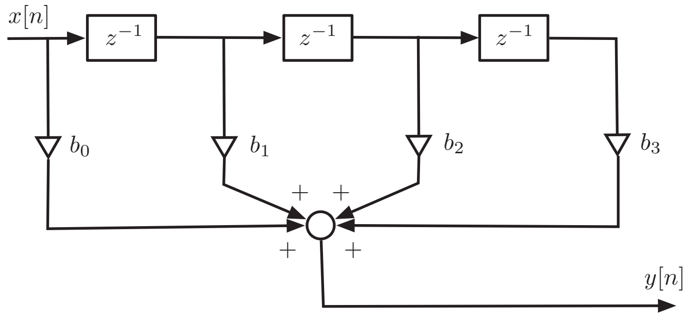

FIGURE 12.29
Direct form realization of FIR filter of order $M = 3$.

The cascade realization of an FIR filter is based on the representation of $H(z)$ in (12.76) as a cascade of first- and second-order filters, i.e., we let

$$H(z) = \prod_{i=1}^{r} H_i(z)$$

where

$$\begin{aligned} H_i(z) &= b_{oi} + b_{1i}z^{-1} \quad \text{or} \\ H_i(z) &= b_{oi} + b_{1i}z^{-1} + b_{2i}z^{-2} \end{aligned}$$

■ Example 12.18

Provide the cascade realization of an FIR filter with transfer function

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

Solution

The transfer function is factored as

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

which can be realized as the cascade of two FIR filters:

$$\begin{aligned} y_1[n] &= x[n] + x[n-1] \\ y[n] &= y_1[n] + 2y_1[n-1] + y_1[n-2] \end{aligned}$$

which are realized as shown in Figure 12.30. ■

12.7 WHAT HAVE WE ACCOMPLISHED? WHERE DO WE GO FROM HERE?

In Chapter 7 and in this chapter you have been introduced to the most important application of linear time-invariant systems: filtering. The design and realization of analog and discrete filters gathers many practical issues in signals and systems. If you pursue this topic, you will see the significance, for instance, of passive and active elements, feedback and operational amplifiers, reactance functions and frequency transformation in analog filtering. The design and realization of discrete

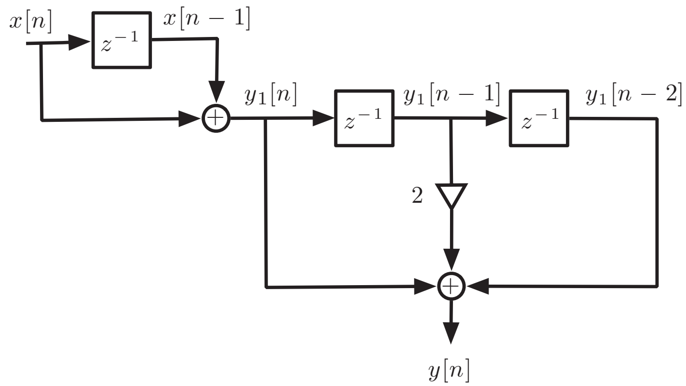

FIGURE 12.30
Cascade realization of FIR filter.

filters brings together interesting topics such as quantization error and its effect on the filters, optimization methods for filter design, stabilization of unstable filters, finite register effects in the implementation of filters, etc. If you pursue filtering deeper, you will find that there is a lot more on filter design that what we have provided you in this chapter. A lot more. Also remember that MATLAB provides you with a large number of functions to design and implement filters. It is our hope you have profited intellectually from the material in this book and that you continue learning more on this subject. There are excellent books in circuits and analog filtering [21,18,14,72,52], in classical control [56,22,38,24], linear control systems [13,39,26], communications [17,30,29,63,70,41], Fourier analysis and numerical analysis [33,9,69], digital filtering [7,2,19], and last but not least excellent books in signals and systems and in digital signal processing [25,31,62,59, 40,48,42,60,68,36,49,58,19,27,43,64,57,12,46] for you to read and learn more.

12.8 PROBLEMS

12.8.1 Basic Problems

12.1 Inputs to an ideal low-pass filter with frequency response

$$H(e^{j\omega}) = \begin{cases} 1e^{-j10\omega} & -\frac{\pi}{2} \leq \omega \leq \frac{\pi}{2} \\ 0 & \text{else in } -\pi \leq \omega \leq \pi \end{cases}$$

are given below. Find the corresponding outputs $y_i[n]$ $i = 1, 2, 3$.

(a) $x_1[n] = 1 + \cos(0.3\pi n) + 23\sin(0.7\pi n), -\infty < n < \infty$.
(b) $x_2[n] = \delta[n] + 2\delta[n-4]$.
(c) $x_3[n]$ has DTFT $X(e^{j\omega}) = H(e^{j\omega/2})$.

Answers: $y_1[n] = 1 - \cos(0.3\pi n)$; $y_2[n] = h[n] + 2h[n-4]$, $h[n]$ impulse response of filter.

12.2 The frequency response of a filter is $H(e^{j\omega}) = 1.5 + \cos(2\omega)$, $-\pi \leq \omega \leq \pi$.

(a) Is $|H(e^{j\omega})| = H(e^{j\omega})$? Is phase zero?
(b) Find the impulse response $h[n]$ of this filter. What type of filter is it? FIR? IIR? Is it causal? Explain.
(c) Let $H_1(e^{j\omega}) = e^{-jN\omega}H(e^{j\omega})$ determine the value of the positive integer N so that $H_1(e^{j\omega})$ has linear phase. Use the impulse response $h[n]$ found above to find the impulse response $h_1[n]$.

Answers: (a) Yes to both; (b) $h[n] = 0.5\delta[n+2] + 1.5 + 0.5\delta[n-2]$.

12.3 The impulse response $h[n]$ of an FIR is given by $h[0] = h[3]$, $h[1] = h[2]$, the other values are zero.

(a) Find the Z-transform of the filter $H(z)$ and the frequency response $H(e^{j\omega})$.
(b) Let $h[0] = h[1] = h[2] = h[3] = 1$, Determine the zeros of the filter. Determine if the phase $\angle H(e^{j\omega})$ of the filter is linear? Explain.
(c) Under what conditions on the values of $h[0]$ and $h[1] - h[0]$ would the phase be linear.
(d) For linear phase, in general, where should the zeros of $H(z)$ be? Explain.

Answers: (a) $H(e^{j\omega}) = e^{-j1.5\omega}(2h[0]\cos(1.5\omega) + 2h[1]\cos(0.5\omega))$; if $h[1] > 3h[0]$ phase is linear.

12.4 An FIR filter has a transfer function $H(z) = z^{-2}(z - e^{j\pi/2})(z - e^{-j\pi/2})$.

(a) Find and plot the poles and zeros of this filter.
(b) Expressing the frequency response at some frequency ω_0 as

$$H(e^{j\omega_0}) = \frac{\vec{Z}_1(\omega_0)\vec{Z}_2(\omega_0)}{\vec{P}_1(\omega_0)\vec{P}_2(\omega_0)}$$

carefully draw the vectors on the pole-zero plot.
(c) Consider the case when $\omega_0 = 0$, find $H(e^{j0})$ analytically and using the vectors.
(d) Repeat the above calculations for $\omega_0 = \pi$ and $\omega_0 = \pm\pi/2$ (verify your result from the given Z-transform). Can you classify this filter?

Answers: $H(e^{j0}) = 2$; $H(e^{j\pi}) = 2$; $H(e^{\pm j\pi/2}) = 0$.

12.5 The impulse response of an FIR filter is $h[n] = \alpha\delta[n] + \beta\delta[n-1] + \alpha\delta[n-2]$, $\alpha > 0$ and $\beta > 0$.

(a) Determine the value of α and β for which this filter has a dc gain $|H(e^{j0})| = 1$, and linear phase $\angle H(e^{j\omega}) = -\omega$.

(b) For the smallest possible β and the corresponding α obtained above find the zeros of the filter, plot them in the Z-plane and indicate their relation. Generalize the relation of the zeros for all possible values of β and corresponding α, and find a general expression for the two zeros.

Answers: $H(e^{j\omega}) = e^{-j\omega}[\beta + 2\alpha\cos(\omega)]$; $\beta \geq 1/2$.

12.6 The transfer function of an FIR filter is $H(z) = z^{-2}(z-2)(z-0.5)$.

(a) Find the impulse response $h[n]$ of this filter and plot it. Comment on any symmetries it might have.
(b) Find the phase $\angle H(e^{j\omega})$ of the filter and carefully plot it. Is this phase linear for all frequencies $-\pi < \omega \leq \pi$?

Answers: $h[n] = \delta[n] - 2.5\delta[n-1] + \delta[n-2]$, phase is linear.

12.7 An FIR filter has a system function $H(z) = 0.05z^2 + 0.5z + 1 + 0.5z^{-1} + 0.05z^{-2}$.

(a) Find the magnitude $|H(e^{j\omega})|$ and phase response $\angle H(e^{j\omega})$ at frequencies $\omega = 0, \pi/2$ and π. Sketch each of these responses for $-\pi \leq \omega < \pi$, and indicate the type of filter.
(b) Determine the impulse response $h[n]$ and indicate if the filter is causal.
(c) If $H(z)$ is non-causal, how would you make it causal? What would be the effect of your procedure on the magnitude and the phase responses obtained before? Explain and plot the magnitude and phase of your causal filter.

Answers: $H(e^{j\omega}) = 1 + \cos(\omega) + 0.1\cos(2\omega)$; filter made causal by delaying $h[n]$ two samples.

12.8 The transfer function of an IIR filter is

$$H(z) = \frac{z+1}{z(z-0.5)}$$

(a) Calculate the impulse response $h[n]$ of the filter.
(b) Would it be possible for this filter to have linear phase? Explain.
(c) Sketch the magnitude response $|H(e^{j\omega})|$ using a plot of the poles and zeros of $H(z)$ in the Z-plane. Use vectors to calculate the magnitude response.

Answers: $h[n] = 0.5^{n-1}u[n-1] + 0.5^{n-2}u[n-2]$; no linear phase possible.

12.9 The transfer function of an IIR filter is

$$H(z) = \frac{(z+2)(z-2)}{(z+0.5)(z-0.5)}$$

Find the magnitude response of this filter at $\omega = 0$, $\omega = \pi/2$, and $\omega = \pi$. From the poles and the zeros of $H(z)$ find geometrically the magnitude response and indicate the type of filter.

Answers: $|H(e^{j\omega})| = 4$ for all ω, i.e., all-pass filter.

12.10 Consider the following problems related to the specification of IIR filters

(a) The magnitude specifications for a low-pass filter are

$$\begin{array}{ll} 1-\delta \leq |H(e^{j\omega})| \leq 1 & 0 \leq \omega \leq 0.5\pi \\ 0 < |H(e^{j\omega})| \leq \delta & 0.75\pi \leq \omega \leq \pi \end{array}$$

i. Find the value of δ that gives the following equivalent loss specifications for this filter

$$\alpha(e^{j0}) = 0 \text{ dB}, \quad \alpha_{max} = 0.92 \text{ dB}, \quad \alpha_{min} = 20 \text{ dB}$$

ii. What are the values of ω_p and ω_{st}?

(b) The following are specifications for a low-pass discrete IIR filter that will be used in processing analog signals

$$\begin{array}{ll} 10 \leq \alpha(e^{j\omega}) \leq 10.1 \text{ dBs} & 0 \leq f \leq 2 \text{ kHz} \\ \alpha(e^{j\omega}) \geq 60 \text{ dBs} & 4 \text{ kHz} \leq f \leq 5 \text{ kHz} \end{array}$$

i. Find the values of α_{max}, α_{min}, and the dc loss in dBs.
ii. What is the sampling frequency f_s in kHz in this filter specifications?
iii. Obtain the equivalent specifications in discrete frequencies for 0 dB dc loss for this filter.

Answers: (a) $\delta = 0.1$; $\omega_p = 0.5\pi$; (b) $\alpha_{max} = 0.1, \alpha_{min} = 50$ dB, and 10 dB dc loss.

12.11 A second-order analog Butterworth filter has a transfer function

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

(a) Is the half-power frequency of this filter $\Omega_{hp} = 1$ rad/sec?
(b) To obtain a discrete Butterworth filter we choose the bilinear transformation

$$s = K\frac{1-z^{-1}}{1+z^{-1}}, K = 1$$

What is the half-power frequency of the discrete filter?

(c) Find the transfer function $H(z)$ when we use the above bilinear transformation.

(d) Plot the poles and zeros of $H(s)$ and $H(z)$. Are both of these filters BIBO stable?

Answers: $|H(j1)| = \frac{1}{\sqrt{2}}|H(j0)|$; $\omega_{hp} = \pi/2$ rad.

12.12 A low-pass IIR discrete filter has a transfer function

$$H(z) = \sum_{n=0}^{\infty} 0.5^n z^{-n}$$

(a) Find the poles and zeros of this filter.

(b) Suppose that you multiply the impulse response of the low-pass filter by $(-1)^n$ so that you obtain a new transfer function

$$H_1(z) = \sum_{n=0}^{\infty} (-0.5)^n z^{-n}$$

Find the poles and zeros for $H_1(z)$. What type of filter is it?

(c) Is it true, in general, that if every z in a low-pass filter transfer function is changed into $-z$ you obtain a high-pass filter? Explain.

Answers: $H(z) = z/(z - 0.5)$; $H_1(z)$ is high-pass filter.

12.13 A first-order low-pass analog filter has a transfer function $H(s) = 1/(s + 1)$.

(a) If for this filter, the input is $x(t)$ and the output is $y(t)$ what is the ordinary differential equation representing this filter.

(b) Suppose that we change this filter into a discrete filter using the bilinear transformation

$$s = K\frac{1 - z^{-1}}{1 + z^{-1}}, \quad K = \frac{2}{T_s}$$

Obtain the transfer function $H(z)$. If for the discrete filter, the input is $x[n]$ and the output $y[n]$ obtain the difference equation representing the discrete filter.

(c) Suppose $x(t) = u(t) - u(t - 0.5)$ find the output of the analog filter.

(d) Let $K = 1000$, so that $T_s = 2/K$ is used to sample $x(t)$ to get the discrete signal $x[n]$. Use the difference equation to solve for the output $y[n]$. Compare your result with the one obtained by solving the ordinary differential equation for the first three values.

Answers: $y(t) = (1 - e^{-t})u(t) + (1 - e^{-(t-0.5)})u(t - 0.5)$;

$y[0] = 0.000999, y[1] = 0.0030$

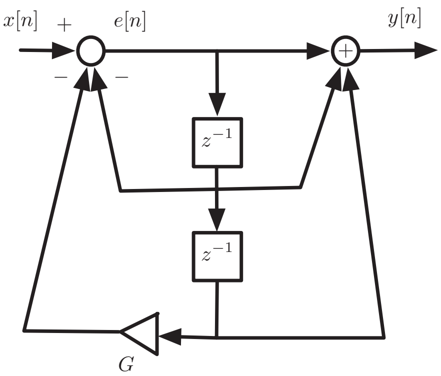

FIGURE 12.31
Problem 14.

12.14 Given the discrete IIR filter realization shown in Figure 12.31 where G is a gain value

(a) Determine the difference equation that corresponds to the filter realization.
(b) Determine the range of values of the gain G so that the given filter is BIBO stable and has complex conjugate poles.

Answers: for complex conjugate poles and system to be BIBO stable $1/4 < G < 1$.

12.15 Consider the following transfer function:

$$H(z) = \frac{2(z-1)(z^2+\sqrt{2}\,z+1)}{(z+0.5)(z^2-0.9z+0.81)}$$

(a) Develop a cascade realization of $H(z)$ using a first-order and a second-order sections. Use minimal direct form to realize each of the sections.
(b) Develop a parallel realization of $H(z)$ by considering first and second-order sections, each realized using minimal direct form.

Answers: Parallel

$$H(z) = -4.94 + 2.16/(1+0.5z^{-1}) + (4.78-1.6z^{-1})/(1-0.9z^{-1}+0.81z^{-2}).$$

12.16 Given the realization in Figure 12.32. Obtain

(a) the difference equations relating $g[n]$ to $x[n]$ and $g[n]$ to $y[n]$,
(b) the transfer function $H(z) = Y(z)/X(z)$ for this filter.

Answers:

$$g[n] = 1.3g[n-1] - 0.8g[n-2] + 2x[n] + 1.8x[n-1] + 0.4x[n-2].$$

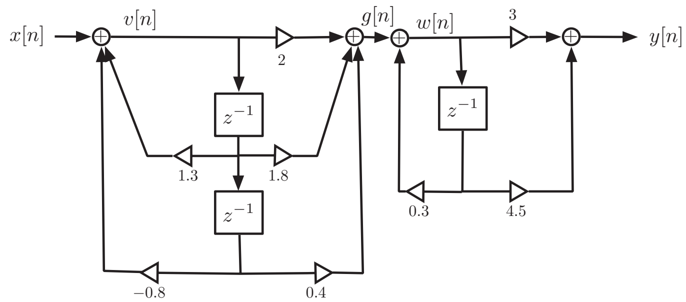

FIGURE 12.32
Problem 16: IIR realization.

12.8.2 Problems using MATLAB

12.17 **FIR filters: causality and phase**—A three-point moving-average filter is of the form:

$$y[n] = \beta\,(\alpha x[n-1] + x[n] + \alpha x[n+1])$$

where α and β are constants, and $x[n]$ is the input and $y[n]$ is the output of the filter.

(a) Determine the transfer function $H(z) = Y(z)/X(z)$ of the filter and from it find the frequency response $H(e^{j\omega})$ of the filter in terms of α and β.

(b) Let $\alpha = 0.5$, find then β so that the dc gain of the filter is unity, and the filter has a zero phase. For the given and obtained values of α and β, sketch $H(e^{j\omega})$ and find the poles and zeros of $H(z)$ and plot them in the Z-plane.

(c) Suppose we let $v[n] = y[n-1]$ be the output of a second filter. Is this filter causal? Find its transfer function $G(z) = V(z)/X(z)$. Use MATLAB to compute the unwrapped phases of $G(z)$ and to plot the poles and zeros of $G(z)$ and $H(z)$ and explain the relation between $G(z)$ and $H(z)$.

Answers: $H(e^{j\omega}) = \beta(1 + 2\alpha\cos(\omega))$; $\beta = 0.5$.

12.18 **FIR and IIR filters: causality and zero-phase**—Let the filter $H(z)$ be the cascade of a causal filter with transfer function $G(z)$ and an anti-causal filter with transfer function $G(z^{-1})$, so that

$$H(z) = G(z)G(z^{-1})$$

(a) Suppose that $G(z)$ is an FIR filter with transfer function

$$G(z) = \frac{1}{3}(1 + 2z^{-1} + z^{-2})$$

Find the frequency response $H(e^{j\omega})$ and determine its phase.

(b) Determine the impulse response of the filter $H(z)$. Is $H(z)$ a causal filter? If not, would delaying its impulse response make it causal? Explain. What would be the transfer function of the causal filter?

(c) Use MATLAB to verify the unwrapped phase of $H(z)$ you obtained analytically, and to plot the poles and zeros of $H(z)$.

(d) How would you use the MATLAB function *conv* to find the impulse response of $H(z)$.

(e) Suppose then that $G(z) = 1/(1-0.5z^{-1})$, find the filter $H(z) = G(z)G(z^{-1})$. Is this filter zero-phase? If so, where are its poles and zeros located? If you think of filter $H(z)$ as causal, is it BIBO stable?

Answers: $H(e^{j\omega})$ has zero phase; $h_1[n] = h[n-2]$ corresponds to causal filter.

12.19 FIR and IIR filters: symmetry of impulse response and linear-phase—Consider two FIR filters with transfer functions

$$H_1(z) = 0.5 + 0.5z^{-1} + 2.2z^{-2} + 0.5z^{-3} + 0.5z^{-4}$$
$$H_2(z) = -0.5 - 0.5z^{-1} + 0.5z^{-3} + 0.5z^{-4}$$

(a) Find the impulse responses $h_1[n]$ and $h_2[n]$ corresponding to $H_1(z)$ and $H_2(z)$. Plot them carefully and determine the sample with respect to which these impulse responses are even or odd.

(b) Show frequency response for $G(z) = z^2 H_1(z)$ is zero-phase, and from it determine the phase of $H_1(e^{j\omega})$. Use MATLAB to find the unwrapped phase of $H_1(e^{j\omega})$ and confirm your analytic results.

(c) Find the phase of $H_2(e^{j\omega})$ by finding the phase of the frequency response for $F(z) = z^2 H_2(z)$. Use MATLAB to find the unwrapped phase of $H_2(e^{j\omega})$. Is it linear?

(d) If $H(z)$ were the transfer function of an IIR filter, according to the above arguments could it be possible for it to have linear phase? Explain.

Answers:
$G(e^{j\omega}) = \cos(2\omega) + \cos(\omega) + 2.2$; $F(e^{j\omega}) = -j(\sin(2\omega) + \sin(\omega))$.

12.20 **Effect of phase on filtering**—Consider two filters with transfer functions

$$\text{(i) } H_1(z) = z^{-100}, \qquad \text{(ii) } H_2(z) = \left(0.5\frac{1-2z^{-1}}{1-0.5z^{-1}}\right)^{10}$$

(a) The magnitude response of these two filters is unity, but that they have different phases. Find analytically the phase of $H_1(e^{j\omega})$ and use MATLAB to find the unwrapped phase of $H_2(e^{j\omega})$ and to plot it.

(b) Consider the MATLAB signal *handel.mat* (a short piece of the Messiah by composer George Handel), use the MATLAB function *filter* to filter it with the two given filters. Listen to the outputs, plot them, and compare them. What is the difference (look at the first 200 samples of the outputs from the two filters)?

(c) Can you recover the original signal by advancing either of the outputs? Explain.

Answers: Use MATLAB function *conv* to find coefficients $H_2(z)$.

12.21 Butterworth versus Chebyshev specifications

A Butterworth low-pass discrete filter of order N has been designed to satisfy the following specifications:

$$\text{Sampling period } T_s = 100\ \mu\text{sec}$$
$$\alpha_{\max} = 0.7 \text{ dB for } 0 \leq f \leq f_p = 1000 \text{ Hz}$$
$$\alpha_{\min} = 10 \text{ dB for } f_{st} = 1200 \leq f \leq f_s/2 \text{ Hz}$$

What should be the new value of the stopband frequency f_{st} so that an N order Chebyshev low-pass filter satisfies the design specifications for $T_s, \alpha_{max}, \alpha_{min}$, and f_p.

Answers: If we choose $f_{st} = 1035$ we have $N_b = N_c = 10$.

12.22 Bilinear transformation and pole location—Find the poles of the discrete filter obtained by applying the bilinear transformation with $K = 1$ to frequency normalized analog second-order Butterworth low-pass filter. Determine the half-power frequency ω_{hp} of the resulting discrete filter. Use the MATLAB function *bilinear* to verify your results.

Answers: Double zero at $z = -1$ and poles at

$$z_{1,2} = \pm j\sqrt{(2-\sqrt{2})/(2+\sqrt{2})}.$$

12.23 Warping effect of the bilinear transformation—The non-linear relation between the discrete frequency ω (rad) and the continuous frequency Ω (rad/sec) in the bilinear transformation causes warping in the high frequencies. To see this consider the following:

(a) Use MATLAB to design a Butterworth analog band-pass filter of order $N = 12$ and with half-power frequencies $\Omega_1 = 10$ and $\Omega_2 = 20$ (rad/sec). Use MATLAB *bilinear function*, with $K = 1$, to transform the resulting filter into a discrete filter. Plot the magnitude and the phase of the discrete filter. Plot the poles and zeros of the continuous and the discrete filters.

(b) Increase the order of the filter to $N = 14$ and keep the other specifications the same. Design an analog bandpass filter and use again the *bilinear function*, with $K = 1$, to transform the analog filter into a discrete filter. Plot the magnitude and the phase of the discrete filter. Plot the poles and zeros of the continuous and the discrete filters. Explain your results.

12.24 **Warping effect of the bilinear transformation on phase**—The warping effect of the bilinear transformation also affects the phase of the transformed filter. Consider a filter with transfer function $G(s) = e^{-5s}$.

(a) Find the transformed discrete frequencies ω (rad) corresponding to $0 \leq \Omega \leq 20$ (rad/sec) using a bilinear transformation with $K = 1$. Plot Ω versus ω

(b) Discretize the continuous frequencies $0 \leq \Omega \leq 20$ (rad/sec) to compute values of $G(j\Omega)$ and use MATLAB functions to plot the phase of $G(j\Omega)$.

(c) Find the function

$$H(e^{j\omega}) = G(j\Omega)|_{\Omega=\tan(\omega/2)},$$

and plot its unwrapped phase using MATLAB for the discrete frequencies corresponding to the analog frequencies to $0 \leq \Omega \leq 20$ (rad/sec). Compare the phases of $G(j\Omega)$ and of $H(e^{j\omega})$.

Answers: $H(e^{j\omega}) = 1e^{-j5\tan(\omega)/2}$ has non-linear phase.

12.25 **Discrete Butterworth filter for analog processing**—Design a Butterworth low-pass discrete filter that satisfies the following specifications:

$$0 \leq \alpha(e^{j\omega}) \leq 3 \text{ dB} \quad \text{for } 0 \leq f \leq 25 \text{ Hz}$$
$$\alpha(e^{j\omega}) \geq 38 \text{ dB} \quad \text{for } 50 \leq f \leq Fs/2 \text{ Hz}$$

and the sampling frequency is $F_s = 2000$ Hz. Express the transfer function $H(z)$ of the designed filter as a cascade of filters. Use first the design formulas and then use MATLAB to confirm your results. Show that the designed filter satisfies the specifications, plotting the loss function of the designed filter.

Answers:

$N = 7$; $H(z) = G\prod_{i=1}^{4} H_i(z)$; $H_1(z) = 0.04(1 + z^{-1})/(1 - 0.92z^{-1})$.

12.26 **All-pass IIR filter**—Consider an all-pass analog filter

$$G(s) = \frac{s^4 - 4s^3 + 8s^2 - 8s + 4}{s^4 + 4s^3 + 8s^2 + 8s + 4}$$

(a) Use MATLAB functions to plot the magnitude and phase responses of $G(s)$. Indicate whether the phase is linear.

(b) A discrete filter $H(z)$ is obtained from $G(s)$ by the bilinear transformation. By trial and error, find the value of K in the bilinear transformation so that the poles and zeros of $H(z)$ are on the imaginary axis of the Z-plane. Use MATLAB functions, to do the bilinear transformation, and to plot the magnitude and unwrapped phase of $H(z)$ and its poles. Is it an all-pass filter? If so, why?

(c) Let the input to the filter $H(z)$ be $x[n] = \sin(0.2\pi n)$, $0 \leq n < 100$, and the corresponding output be $y[n]$. Use MATLAB functions to compute and plot $y[n]$. From these results would you say that the phase of $H(z)$ is approximately linear? Why or why not?

Answers: All-pass filter, $K = 1.4$, phase is approximately linear.

12.27 **Butterworth filtering of analog signal**—We wish to design a discrete Butterworth filter that can be used in filtering a continuous-time signal. The frequency components of interest in this signal are between 0 and 1 kHz, so we would like the filter to have a maximum passband attenuation of 3 dB within that band. The undesirable components of the input signal occur beyond 2 kHz, and we like to attenuate them by at least 10 dB. The maximum frequency present in the input signal is 5 kHz. Finally, we would like the dc gain of the filter to be 10. Choose the Nyquist sampling frequency to process the input signal. Use MATLAB to design the filter. Give the transfer function of the filter, plot its poles and zeros, and its magnitude and unwrapped phase response using an analog frequency scale in kHz.

Answers: Low-pass, with $f_p = 1$ kHz ($f_p = f_{hp}$), $\alpha_{max} = 3$ dB ,
$f_{st} = 2$ kHz, $\alpha_{min} = 10$ dB.

12.28 **Butterworth versus Chebyshev filtering**—If we wish to preserve low frequencies components of the input, a low-pass Butterworth filter could perform better than a Chebyshev filter. MATLAB provides a second Chebyshev filter function *cheby2* that has a flat response in the passband and a rippled one in the stopband. Let the signal to be filtered be the first 100 samples from MATLAB's *train* signal. To this signal add some Gaussian noise to be generated by *randn*, multiply it by 0.1 and add it to the 100 samples of the train signal. Design three discrete filters, each of order 20, and half-frequency (for Butterworth *butter*) and the passband frequency (for the Chebyshev filters) of $\omega_n = 0.5$. For the design with *cheby1* let the maximum passband attenuation be 0.01 dB and for the design with *cheby2* let the minimum stopband attenuation be 60 dB. Obtain the three filters and use them to filter the noisy train signal. Using MATLAB plot the following for each of the three filters:

- Using the *fft* function compute the DFT of the original signal, the noisy signal, and the noise, and plot their magnitudes. Is the cutoff frequency of the filters adequate to get rid of the noise? Explain.
- Compute and plot the magnitude and the unwrapped phase, as well as the poles and zeros for each of the three filters. Comment on the differences in the magnitude responses.
- Use the function *filter* to obtain the output of each of the filters, and plot the original noiseless signal and the filtered signals. Compare them.

12.29 **Butterworth, Chebyshev, and Elliptic filters**—The gain specifications of a filter are

$$-0.1 \leq 20\log_{10}|H(e^{j\omega})| \leq 0 \text{ (dB)} \quad 0 \leq \omega \leq 0.2\pi$$

$$20\log_{10}|H(e^{j\omega})| \leq -60 \text{ (dB)} \qquad 0.3\pi \leq \omega \leq \pi$$

(a) Find the loss specifications for this filter.

(b) Design using MATLAB a Butterworth, a Chebyshev (using *cheby1*), and an Elliptic filters. Plot in one plot the magnitude response of the three filters, compare them and indicate which gives the lowest order.

Answers: The dc loss is 0 dB, $\alpha_{max} = 0.1$ and $\alpha_{min} = 60$ dBs.

12.30 **Notch and all-pass filters**—Notch filters is a family of filters that includes the all-pass filter. For the filter

$$H(z) = K\frac{(1-\alpha_1 z^{-1})(1+\alpha_2 z^{-1})}{(1-0.5z^{-1})(1+0.5z^{-1})}$$

(a) Determine the values of α_1, α_2, and K that would make $H(z)$ an all-pass filter of unit magnitude. Use MATLAB to compute and plot the magnitude response of $H(z)$ using the obtained values for α and K. Plot the poles and zeros of this filter.

(b) If we would like the filter $H(z)$ to be a notch filter, of unit gain at at $\omega = \pi/2$ (rad), and notches at $\omega = 0$ and π, determine the values of α and K to achieve this. Use MATLAB functions to verify that the filter is a notch filter, and to plot the poles and zeros.

(c) Use MATLAB to show that when $\alpha_1 = \alpha_2 = \alpha$ and $1 \leq \alpha \leq 2$, the given $H(z)$ correspond to a family of notch filters with different attenuations. Determine K so that these filters are unity gain at $\omega = \pi/2$.

(d) Suppose we use the transformation $z^{-1} = jZ^{-1}$ to obtain a filter $H(Z)$ using $H(z)$ obtained in the previous item. Where are the notches of this new filter? What is the difference between the all-filters $H(z)$ and $H(Z)$?

Answers: $H(z) = K(1 - \alpha^2 z^{-2})/(1 - 0.25z^{-2})$, $1 \leq \alpha \leq 2$, notch filters with notches at $\omega = 0, \pi$.

12.31 **IIR comb filters**—Consider a filter with transfer function

$$H(z) = K\frac{1 + z^{-4}}{1 + (1/16)z^{-4}}$$

(a) Find the gain K so that this filter has unit dc gain. Use MATLAB to find and plot the magnitude response of $H(z)$, and its poles and zeros. Why is it called a comb filter?

(b) Use MATLAB to find the phase response of the filter $H(z)$. Why is it that the phase seems to be wrapped and it cannot be unwrapped by MATLAB?

(c) Suppose you wish to obtain an IIR comb filter that is sharper around the notches of $H(z)$ and flatter in between notches. Implement such a filter using the function *butter* to obtain two notch filters of order 10 and appropriate cutoff frequencies. Decide how to connect the two filters. Plot the magnitude and phase of the resulting filter, and its poles and zeros.

Answers: Zeros $z_k = e^{j(2k+1)\pi/4}$, poles $z_k = 0.5e^{j(2k+1)\pi/4}$, $k = 0, \ldots, 3$.

12.32 **Three-band discrete spectrum analyzer**—To design a three-band discrete spectrum analyzer for audio signals, we need to design a low-pass, a band-pass, and a high-pass IIR filters. Let the sampling frequency be $F_s = 10$ kHz. Consider the three bands, in kHz, to be $[0\ F_s/4]$, $(F_s/4\ 3F_s/8]$, and $(3F_s/8\ F_s/2]$.

(a) Let all the filters be of order $N = 20$, and choose the cutoff frequencies so that the sum of the tree filters is an all-pass filter of unit gain.

(b) Consider the MATLAB test signal *handel*, use the designed spectrum analyzer to obtain the spectrum in the three bands.

12.33 **FIR filter design with different windows**—Design a causal low-pass FIR digital filter with $N = 21$. The desired magnitude response of the filter is

$$|H_d(e^{jwT})| = \begin{cases} 1 & 0 \leq f \leq 250 \text{ Hz} \\ 0 & \text{elsewhere in } 0 \leq f \leq (f_s/2) \end{cases}$$

and the phase is zero for all frequencies. The sampling frequency $f_s = 2000$ Hz.

(a) Use a rectangular window in your design. Plot magnitude and phase of the designed filter.

(b) Use a triangular window in the design and compare the magnitude and phase plots of this filter with those obtained in part (a).

Answers: $h_d[0] = 0.25, h_d[n] = \sin(\pi n/4)/(\pi n) n \neq 0$.

12.34 **FIR filter design**—Design an FIR low-pass filter with a cuttoff of $\pi/3$ and lengths $N = 21$ and $N = 81$,

(a) using a rectangular window
(b) using Hamming and Kaiser ($\beta = 4.5$) windows, and compare the magnitude of the resulting filters.

Answers: $h_d[0] = 0.33, h_d[n] = \sin(\pi n/3)/(\pi n), n \neq 0$.

12.35 **Modulation property transformation for IIR filters**—The modulation-based frequency transformation of the DTFT is applicable to IIR filters. It is obvious in the case of FIR filters, but requires a few more steps in the case of IIR filters. In fact, if we have that the transfer function of the prototype IIR low-pass filter is $H(z) = B(z)/A(z)$, with impulse response $h[n]$, let the transformed filter be $\hat{H}(z) = \mathcal{Z}(2h[n]\cos(\omega_o n))$ for some frequency ω_0.

(a) Find the transfer function $\hat{H}(z)$ in terms of $H(z)$
(b) Consider an IIR low-pass filter $H(z) = 1/(1 - 0.5z^{-1})$. If $\omega_0 = \pi/2$ determine $\hat{H}(z)$.
(c) How would you obtain a high-pass filter from $H(z)$ given in the previous item? Use MATLAB to plot the resulting filters here and in the past item.

Answers: (b) $\hat{H}(z) = 2/(1 + 0.25z^{-2})$.

12.36 **Down-sampling transformations**—Consider down-sampling the impulse response $h[n]$ of a filter with transfer function $H(z) = 1/(1 - 0.5z^{-1})$.

(a) Use MATLAB to plot $h[n]$ and the down-sampled impulse response $g[n] = h[2n]$.
(b) Plot the magnitude responses corresponding to $h[n]$ and $g[n]$ and comment on the effect of the down-sampling. Is $G(e^{j\omega}) = 0.5H(e^{j\omega/2})$? Explain.

Answers: $g[n] = 0.25^n u[n]$; $G(e^{j\omega}) \neq 0.5H(e^{j\omega/2})$.

12.37 **Modulation property transformation**—Consider a moving average, low-pass, FIR filter

$$H(z) = \frac{1 + z^{-1} + z^{-2}}{3}$$

(a) Use the modulation property to convert the given filter into a high-pass filter.
(b) Use MATLAB to plot the magnitude responses of the low-pass and the high-pass filters.

Answers: $G(e^{j\omega}) = e^{-j\omega}(2\cos(\omega) - 1)/3$.

12.38 **Implementation of IIR rational transformation**—Use MATLAB to design a Butterworth second-order low-pass discrete filter $H(Z)$ with half-power frequency $\theta_{hp} = \pi/2$, and dc gain of 1. Consider this low-pass filter a prototype that can be used to obtain other filters. Implement using MATLAB the frequency transformations $Z^{-1} = N(z)/D(z)$ using the convolution property to multiply polynomials to obtain:

(a) A high-pass filter with a half-power frequency $\omega_{hp} = \pi/3$ from the low-pass filter.
(b) A band-pass filter with $\omega_1 = \pi/2$ and $\omega_2 = 3\pi/4$ from the low-pass filter.
(c) Plot the magnitude of the low-pass, high-pass and band-pass filters.Give the corresponding transfer functions for the low-pass as well as the high-pass and the band-pass filters.

Answers: Use *conv* function to implement multiplication of polynomials.

12.39 **Parallel connection of IIR filters**—Use MATLAB to design a Butterworth second-order low-pass discrete filter with half-power frequency $\theta_{hp} = \pi/2$, and dc gain of 1, call it $H(z)$. Use this filter as prototype to obtain a filter composed of a parallel combination of the following filters.

(a) Assume that we up-sample by $L = 2$ the impulse response $h(n)$ of $H(z)$ to get a new filter $H_1(z) = H(z^2)$. Determine $H_1(z)$, plot its magnitude using MATLAB and indicate the type of filter.
(b) Assume then that we shift $H(z)$ by $\pi/2$ to get a band-pass filter $H_2(z)$. Find the transfer function of $H_2(z)$ from $H(z)$, plot its magnitude and indicate the type of filter.
(c) If the filters $H_1(z)$ and $H_2(z)$ are connected in parallel, what is the overall transfer function $G(z)$ of the parallel connection? Plot the magnitude response corresponding to $G(z)$.

Answers: $H_1(z)$ is band-eliminating filter and $H_2(z)$ is band-pass filter.

Useful Formulas

CONTENTS

TRIGONOMETRIC RELATIONS

Reciprocal

$$\csc(\theta) = \frac{1}{\sin(\theta)} \qquad \sec(\theta) = \frac{1}{\cos(\theta)}$$

$$\cot(\theta) = \frac{1}{\tan(\theta)}$$

Pythagorean

$$\sin^2(\theta) + \cos^2(\theta) = 1$$

Sum and Difference of Angles

$$\begin{aligned}
&\sin(\theta \pm \phi) = \sin(\theta)\cos(\phi) \pm \cos(\theta)\sin(\phi)\\
&\sin(2\theta) = 2\sin(\theta)\cos(\theta)\\
&\cos(\theta \pm \phi) = \cos(\theta)\cos(\phi) \mp \sin(\theta)\sin(\phi)\\
&\cos(2\theta) = \cos^2(\theta) - \sin^2(\theta)
\end{aligned}$$

Multiple-angle

$$\begin{aligned}
&\sin(n\theta) = 2\sin((n-1)\theta)\cos(\theta) - \sin((n-2)\theta)\\
&\cos(n\theta) = 2\cos((n-1)\theta)\cos(\theta) - \cos((n-2)\theta)
\end{aligned}$$

Products

$$\begin{aligned}
\sin(\theta)\sin(\phi) &= \frac{1}{2}[\cos(\theta-\phi) - \cos(\theta+\phi)]\\
\cos(\theta)\cos(\phi) &= \frac{1}{2}[\cos(\theta-\phi) + \cos(\theta+\phi)]\\
\sin(\theta)\cos(\phi) &= \frac{1}{2}[\sin(\theta+\phi) + \sin(\theta-\phi)]\\
\cos(\theta)\sin(\phi) &= \frac{1}{2}[\sin(\theta+\phi) - \sin(\theta-\phi)]
\end{aligned}$$

Signals and Systems Using MATLAB®, Second Edition. http://dx.doi.org/10.1016/B978-0-12-394812-0.00020-6

Euler's Identity (**θ in radians**)

$$e^{j\theta} = \cos(\theta) + j\sin(\theta) \qquad j = \sqrt{-1}$$

$$\cos(\theta) = \frac{e^{j\theta} + e^{-j\theta}}{2} \qquad \sin(\theta) = \frac{e^{j\theta} - e^{-j\theta}}{2j} \qquad \tan(\theta) = -j\left[\frac{e^{j\theta} - e^{-j\theta}}{e^{j\theta} + e^{-j\theta}}\right]$$

Hyperbolic Trigonometry Relations

Hyperbolic cosine $\cosh(\alpha) = \frac{1}{2}(e^{\alpha} + e^{-\alpha})$

Hyperbolic sine $\sinh(\alpha) = \frac{1}{2}(e^{\alpha} - e^{-\alpha})$

$$\cosh^2(\alpha) - \sinh^2(\alpha) = 1$$

CALCULUS

Derivatives

u, v functions of x, α, β constants

$$\frac{d\,uv}{dx} = u\frac{dv}{dx} + v\frac{du}{dx}$$

$$\frac{du^n}{dx} = nu^{n-1}\frac{du}{dx}$$

Integrals

$$\int \phi(y)dx = \int \frac{\phi(y)}{y'}dy, \text{ where } y' = \frac{dy}{dx}$$

$$\int u\,dv = uv - \int v\,du$$

$$\int x^n dx = \frac{x^{n+1}}{n+1} \qquad n \neq 1, \text{integer}$$

$$\int x^{-1}dx = \log(x)$$

$$\int e^{ax}dx = \frac{e^{ax}}{a} \qquad a \neq 0$$

$$\int xe^{ax}dx = \frac{e^{ax}}{a^2}(ax - 1)$$

$$\int \sin(ax)dx = -\frac{1}{a}\cos(ax)$$

$$\int \cos(ax)dx = \frac{1}{a}\sin(ax)$$

$$\int \frac{\sin(x)}{x}dx = \sum_{n=0}^{\infty}(-1)^n\frac{x^{2n+1}}{(2n+1)(2n+1)!} \quad \text{integral of sinc function}$$

$$\int_0^{\infty} \frac{\sin(x)}{x}dx = \int_0^{\infty}\left[\frac{\sin(x)}{x}\right]^2 dx = \frac{\pi}{2}$$