6

Finite impulse response filter design

6.1 Classification of digital filters

Digital filters are discrete-time systems. The type of digital filters that we shall design in this course is linear. Therefore, they possess all the properties of linear discrete-time systems discussed in Chapter 3. All linear discrete-time operations on an input sequence can be viewed as a filtering of the sequence to produce an output sequence. This is the reason why digital filters are so important in DSP.

Non-linear filters are also commonly used, especially in areas such as image processing. The median filter discussed in section 5.4 for image enhancement is a typical non-linear digital filter.

Linear systems are characterized by their impulse responses. An impulse response can either have a finite or an infinite duration. A finite impulse response $h(n)$ has its non-zero values extending over a finite time interval and is zero beyond that interval. The following finite impulse response

$$h(n) = \{h_0, h_1, h_2, ..., h_N, 0, 0, 0, ...\}$$

has non-zero values in the interval

$$0 \leq n \leq N$$

and is referred to as a finite impulse response (FIR) filter or system of order N. So an Nth order FIR digital filter has an impulse response with a length of $(N+1)$ samples. The samples of the impulse response function (h_0, h_1, etc) are usually called filter coefficients, filter weights, and filter tap coefficients/weights.

If the impulse response function has an infinite duration, we have an infinite impulse response (IIR) filter. It is obvious that IIR filters cause computational problems since we cannot compute an infinite number of terms. But the type of IIR filters that are designed have their input and output samples interrelated through a linear difference equation. The

output sequence can then be computed recursively. This is the reason why IIR filters are also known as recursive filters and FIR filters as non-recursive filters.

In this chapter, we shall concentrate on FIR filters. IIR filters will be discussed in detail in the next chapter.

6.2 Filter design process

The general digital filter design process can be broken up into four main steps:

- Approximation
- Synthesis and realization
- Performance analysis
- Implementation

These steps are illustrated in diagram form in Figure 6.1.

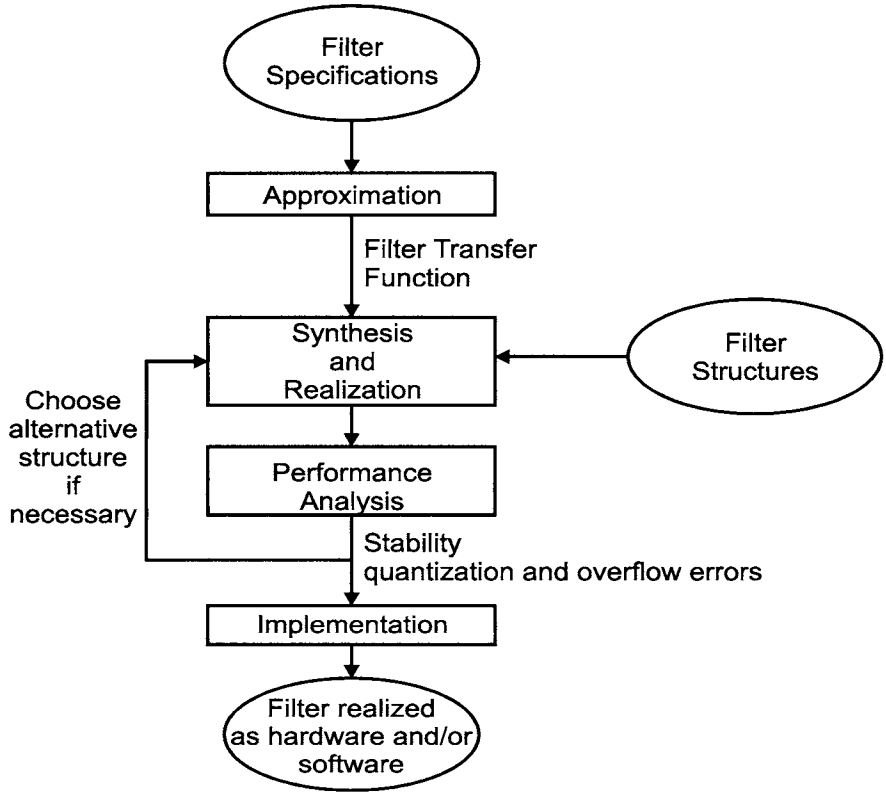

Figure 6.1
The filter design process

6.2.1 Approximation

The design process normally starts with the specifications and requirements of the filter, which are intimately related to the application at hand. These specifications may include frequency domain characteristics such as magnitude and phase responses. There may also be some time domain requirements such as maximum delay.

Most specifications define the upper and lower limits to each of these characteristics. Typical examples can be found in many communication system standards documents. The pre-modulation filter of the ERMES standard for paging systems is shown in Figure 6.2. Alternatively, a desired or ideal response may be given with the maximum amount of deviations from the ideal specified.

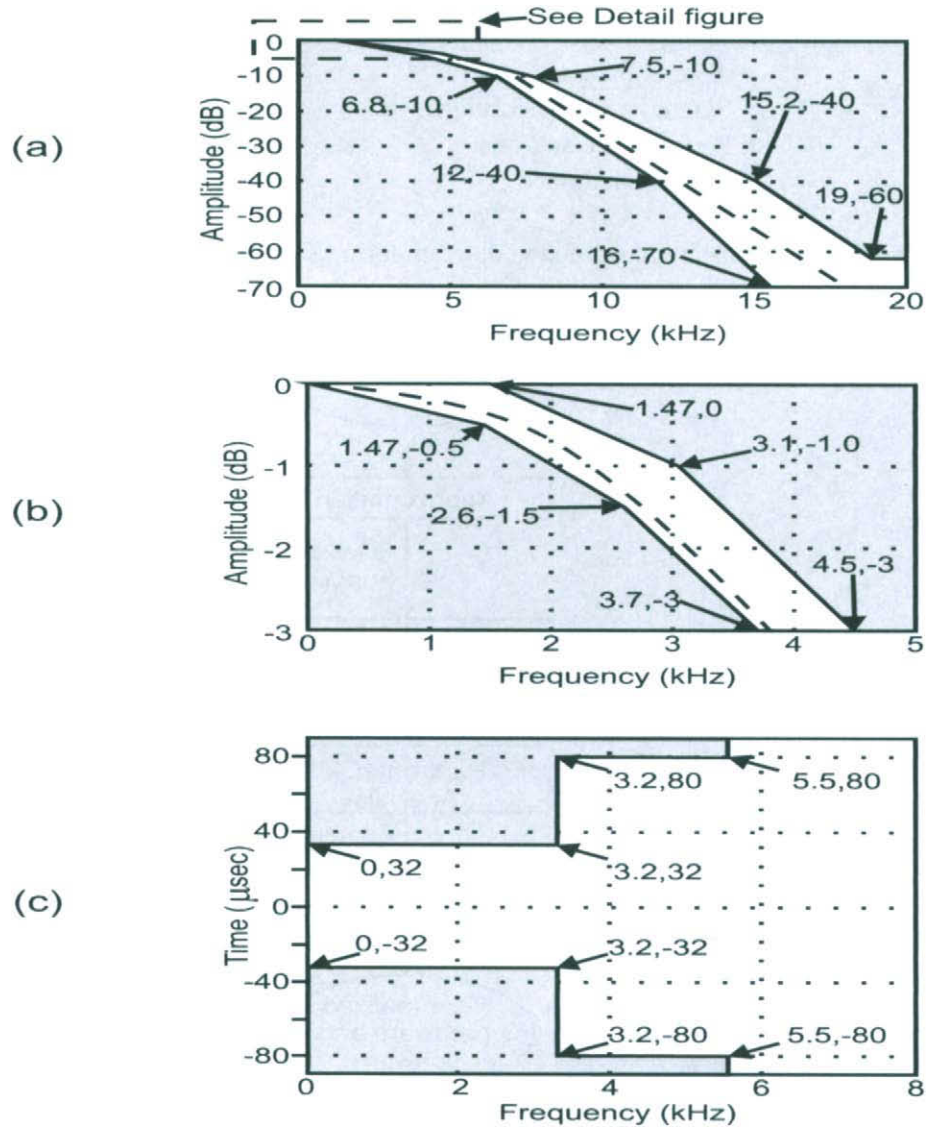

Figure 6.2
Pre-modulation filter specifications in the ERMES system

Given the filter specifications, the first step of the design process is to find a filter transfer function that will satisfy these specifications. This process is called approximation. It is so called because what we are doing is in fact finding a transfer function that approximates the ideal response that is specified.

The methods for solving the approximation problem for digital filters can be classified as direct or indirect. With direct methods, the problem is solved in the discrete-time (and hence discrete-frequency) domain. For indirect methods, a continuous-time transfer function is first obtained using well-established methods in analog filter design. This

transfer function is then transformed into a discrete-time transfer function. Indirect methods are more commonly used for IIR filters, whereas FIR filter design methods are mostly direct ones.

These solution methods can also be classified as closed-form or iterative. Closed form methods make use of closed-form formulas and are usually completed in a definite number of steps. Iterative methods make use of optimization techniques that start with an initial solution and the solution is refined progressively until some pre-determined performance criteria are satisfied. The number of iterations is unknown and depends on the initial solution and the effectiveness of the optimization techniques employed.

6.2.2 Synthesis and realization

Once the transfer function has been determined, it has to be realized into a discrete-time linear network. This procedure is analogous to the filter realization procedure for analog filters where suitable circuit topology and circuit element values are chosen to realize a certain filter transfer function. A number of realization methods has been proposed and studied in the past. The best realization of a given transfer function depends very much on the application. General considerations include the number of adders and multipliers required, and the sensitivity of the network to finite precision arithmetic effects.

Digital filter realization will be discussed in detail in Chapter 8.

6.2.3 Performance analysis

Even though the filter coefficients are determined to a high degree of precision in the approximation step, digital hardware has a finite precision. The accuracy of the output will depend on the type of arithmetic used: fixed-point or floating-point. This is particularly so for fixed-point arithmetic. The designer must ensure that the error introduced by finite precision will not cause violations of the filter specifications. Furthermore, arithmetic overflow and underflow effects must be examined.

It cannot be over-emphasized how important this design step is, especially for IIR filters. While FIR filters are guaranteed to be stable, IIR filters can exhibit instability due to quantization errors introduced in the computational process.

Finite precision effects in digital filters will be discussed in detail in Chapter 8.

6.2.4 Implementation

Digital filters can be implemented either in software or hardware or a combination of both. Software implementations require a decision to be made on the type of computer or microprocessor the software will eventually run on. DSP chips, which are designed specifically for DSP type of operations, are very effective. In Chapter 9 we shall outline the architectures and characteristics of some of the more commonly used and commercially available devices on the market.

Note that the ease of software development depends very much on the quality of the development tools. While the performance of some DSP chips may be similar, the quality of tools available may be very different. The DSP designer should be aware of this fact. Some software tools are developed by the DSP chip manufacturers while others are third party. Some of these tools are described in Chapter 10.

In very demanding applications, the filter may need to be hard-wired or implemented as an application specific integrated circuit (ASIC) in order to obtain the speed required. It may also be necessary that some of the other functions such as analog-to-digital conversion and digital-to-analog conversion be integrated on the same device. However, development time will generally be longer and the cost is much higher.

6.3 Characteristics of FIR filters

Since FIR filters are linear discrete-time systems, the output sequence is related to the input and the impulse response of the filter by the convolution sum:

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

This equation indicates that any particular output sample is only dependent on N input samples for an Nth order filter. Therefore FIR filters are also known *as* non-recursive filters. Also note that the summation on the right-hand side is a convolution between $x(n)$, the input sequence and $h(n)$, the impulse response of the filter. Hence they are also called convolution filters. From the statistical viewpoint, the output sample is a weighted average of the N input sample values. Thus the name moving-average (MA) filter is also used. But the name 'FIR' is most commonly seen in publications.

One of the major advantages of FIR filters is the ease with which exact linear phase filters can be designed. A filter with linear phase characteristics will not distort the input signal and is desirable in a number of applications such as digital communications. Design methods for FIR filters are generally linear and efficient. Another important property of FIR filters is that they are guaranteed to be stable. Furthermore, they can be efficiently realized on general and special purpose hardware. For instance, most DSP chips have special instructions to facilitate the implementation of an FIR filter.

6.3.1 Frequency response

The frequency response of an Nth order FIR filter is given by

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

where ω is in radians per second. Strictly speaking, the exponent should be $(-j\omega Tn)$ where T is the sampling period. But we shall assume that $T = 1$ for simplicity, unless otherwise stated.

Notice that even though the filter is a discrete-time system, the frequency variable is continuous and is periodic with period 2π. This is an important point to remember especially if we are evaluating the frequency response using DFT. For a length-N impulse response, the DFT equation will give us N frequency points. If N is small, we may not get an accurate picture of the response. In these cases, the original impulse response $h(n)$ may need to be padded with an appropriate number of zeros in order to provide us with a more accurate frequency response curve.

Recall that the frequency response of a digital system is generally complex valued and consists of a magnitude and a phase. The function can be written as

$$H(\omega) = A(\omega)e^{j\theta(\omega)}$$

where $A(\omega)$ is the amplitude function/response and $\theta(\omega)$ is the phase function/response. The magnitude response is therefore given by

$$M(\omega) = |H(\omega)| = |A(\omega)|$$

The DFT of a length-N impulse response $h(n)$ is defined as

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

Example 6.1
An FIR filter has impulse response

$$h(n) = \{1, 3, 5, 3, 1\}$$

The magnitude and phase responses are shown in Figure 6.3.
The five DFT coefficients are given by

$$C(k) = \begin{Bmatrix} 13.00, \\ -4.2361 - j3.0777, \\ 0.2361 + j0.7256, \\ 0.2361 - j0.7256, \\ -4.2361 + j3.0777 \end{Bmatrix}$$

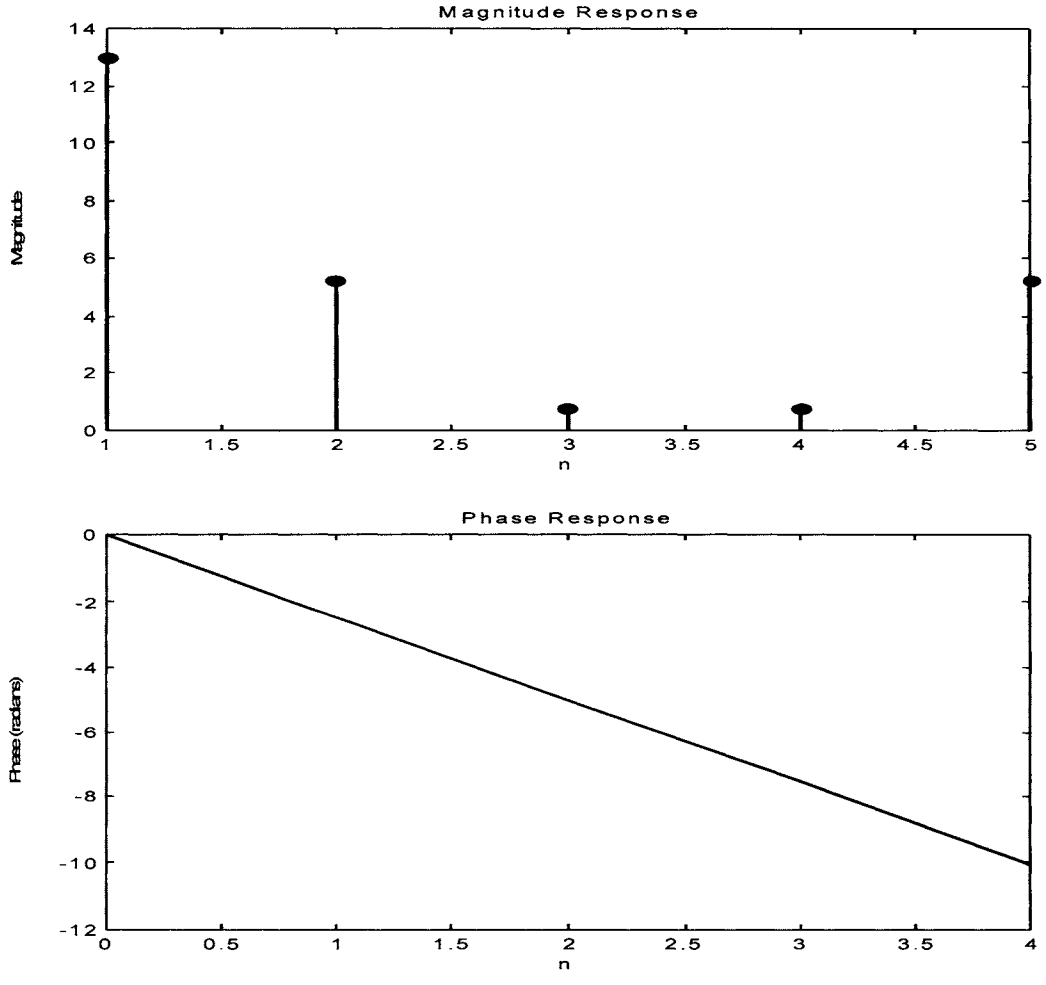

Figure 6.3
Magnitude and phase responses in Example 6.1

6.3.2 Linear phase filters

Linear phase refers to the phase response being a linear function of frequency. The FIR filter in the example given in the previous section has linear phase characteristics as shown in the phase response in Figure 6.4.

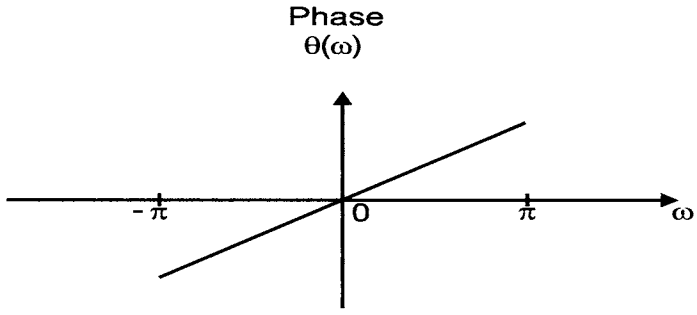

Figure 6.4
A linear phase response

Another way of saying that a filter has linear phase response is to say that it has a constant group delay response.

It can be shown mathematically that an FIR digital filter possesses exact linear phase properties if its impulse response is either symmetric (with even symmetry) or anti-symmetric (with odd symmetry) about the midpoint. Since the length of the impulse response of a digital filter can either be odd or even, there are in total four types of linear phase FIR filters.

- **Type 1:**
  The impulse response has odd length (N is odd) and is even symmetric about its midpoint. Thus

$$h(n) = h(N-n-1)$$

The amplitude response has even symmetry about $\omega = 0$ and $\omega = \pi$. It is also periodic with period 2π. That is,

$$A(\omega) = A(-\omega)$$
$$A(\pi + \omega) = A(\pi - \omega)$$
$$A(\omega + 2\pi) = A(\omega)$$

Figure 6.5 shows the impulse response and amplitude spectrum of a typical type 1 linear phase FIR filter.

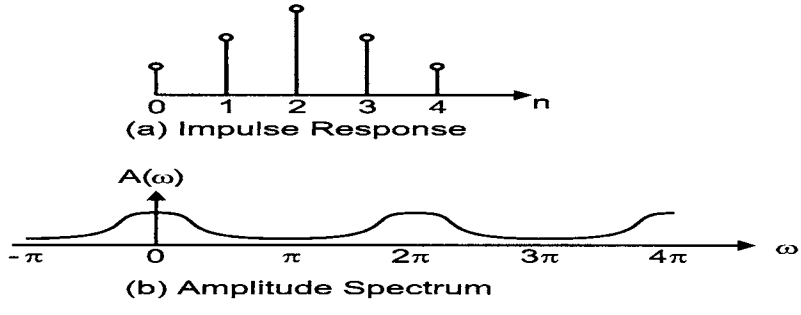

Figure 6.5
Type 1 linear phase responses

- **Type 2:**
  The impulse response has even length and is even symmetric about its midpoint M. Note that in this case M is not an integer. The amplitude spectrum is even about $\omega = 0$ and odd about $\omega = \pi$. The spectrum is also periodic with a period of 4π, instead of 2π.

$$A(\omega) = A(-\omega)$$
$$A(\pi + \omega) = A(\pi - \omega)$$
$$A(\omega + 2\pi) = A(\omega)$$

An example is shown in Figure 6.6.

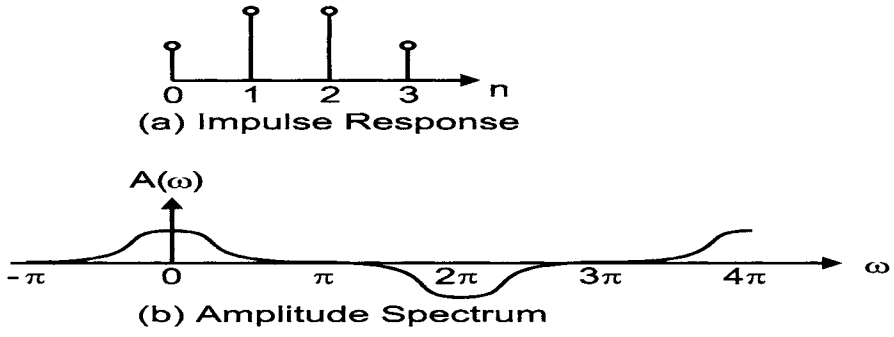

Figure 6.6
An example of type 2 linear phase responses

The frequency response of this type of filter must be zero at $\omega = \pi$. They will make good low-pass filters but are unsuitable for high-pass designs.

- **Type 3:**
  The impulse response has odd length and odd symmetry about the midpoint. The amplitude spectrum is odd about $\omega = 0$ and $\omega = \pi$. It has a period of 2π.

$$A(\omega) = A(-\omega)$$
$$A(\pi + \omega) = -A(\pi - \omega)$$
$$A(\omega + 4\pi) = A(\omega)$$

Figure 6.7 shows an example.

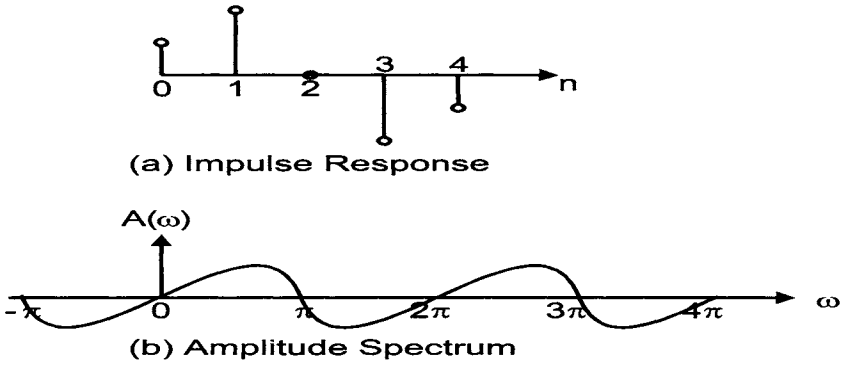

Figure 6.7
An example of type 3 linear phase responses

Note that this type of filter has frequency responses which must be zero at zero frequency ($\omega = 0$) and at $\omega = \pi$. They are therefore not suitable for low-pass and high-pass designs. Furthermore, they introduce a phase shift of 90°.

- **Type 4:**
The impulse response has even length and odd symmetry about the midpoint. The amplitude spectrum is odd about $\omega = 0$ and even about $\omega = \pi$. It has a period of 4π.

$$A(\omega) = -A(-\omega)$$
$$A(\pi + \omega) = -A(\pi - \omega)$$
$$A(\omega + 2\pi) = A(\omega)$$

See Figure 6.8 for an example.

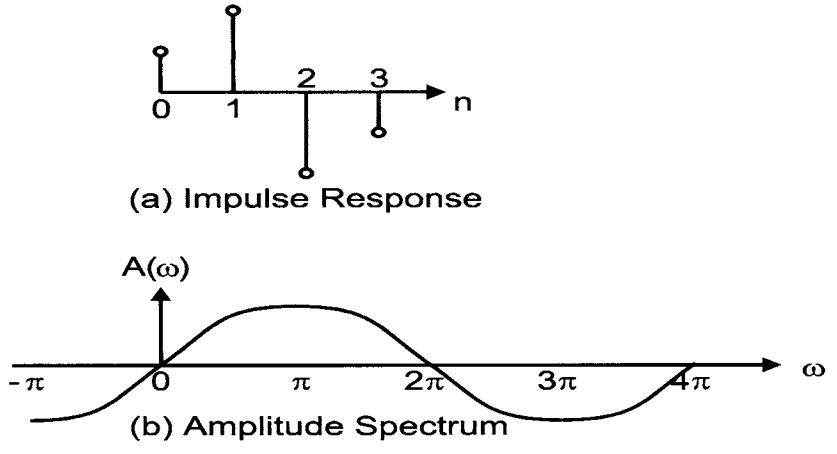

Figure 6.8
Example of type 4 linear phase responses

The frequency response of this type of filters must be zero at zero frequency but not necessarily so at $\omega = \pi$. Therefore they should not be used for low-pass designs but they can make good high-pass filters. Like type 3 filters, they also introduce a phase shift of 90°.

6.4 Window method

The window method is particularly useful for designing filters with simple desired frequency response curves, such as an ideal low-pass, high-pass, band-pass and band-reject filters. Examples of these four ideal filter responses are shown in Figure 6.9.

Notice that in Figure 6.9, the filter frequency responses are only specified over the frequency interval

$$-\pi \leq \omega \leq \pi$$

This is because for digital filters, the frequency response is periodic in ω with a period of 2π. This interval is also called the Nyquist interval in the literature.

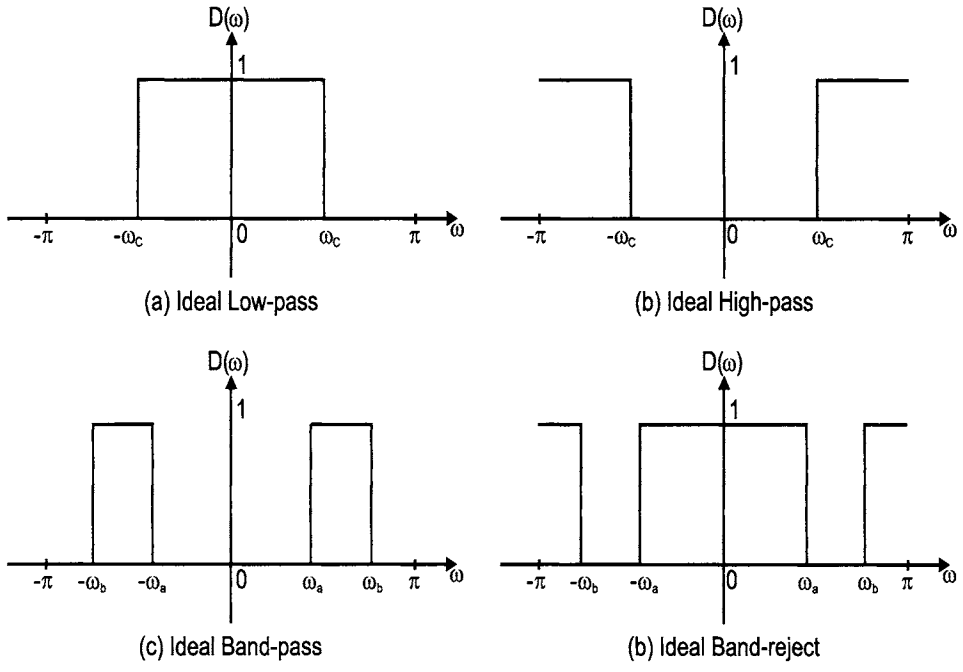

Figure 6.9
Four ideal filter frequency responses

The (continuous) frequency response and the (discrete-time) impulse response are related by the discrete-time Fourier transform (DTFT) relationships:

$$D(\omega) = \sum_{k=-\infty}^{\infty} d(k)e^{-j\omega k}$$

$$d(k) = \int_{-\pi}^{\pi} D(\omega)e^{j\omega k}\frac{d\omega}{2\pi}$$

So, given the desired frequency response, the filter impulse response can be obtained by using the inverse DTFT equation. The filter coefficients will simply be the impulse response samples. However, the impulse response obtained by using the inverse DTFT will in general have two undesirable properties: non-causal and infinite duration. For instance, consider a desired low-pass filter response given by

$$D(\omega) = \begin{cases} 1, & \text{if } |\omega| \le \omega_c \\ 0, & \text{if } \omega_c \le |\omega| \le \pi \end{cases}$$

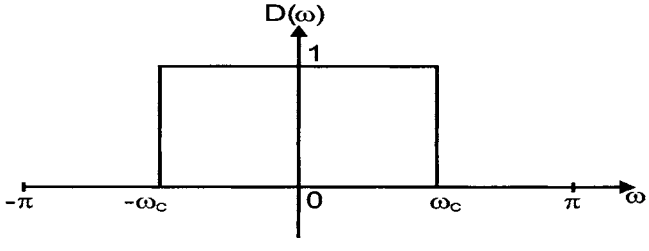

Figure 6.10
An ideal low-pass filter response

This is shown in Figure 6.10. The impulse response will be

$$d(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} D(\omega)e^{j\omega k}\,d\omega$$

$$= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} 1\cdot e^{j\omega k}\,d\omega$$

$$= \frac{1}{2\pi jk}\left[e^{j\omega_c k} - e^{-j\omega_c k}\right]$$

$$\frac{\sin(\omega_c k)}{\pi k} \qquad -\infty < k < \infty$$

with

$$d(0) = \frac{\omega_c}{\pi}$$

This impulse response is plotted in Figure 6.11.

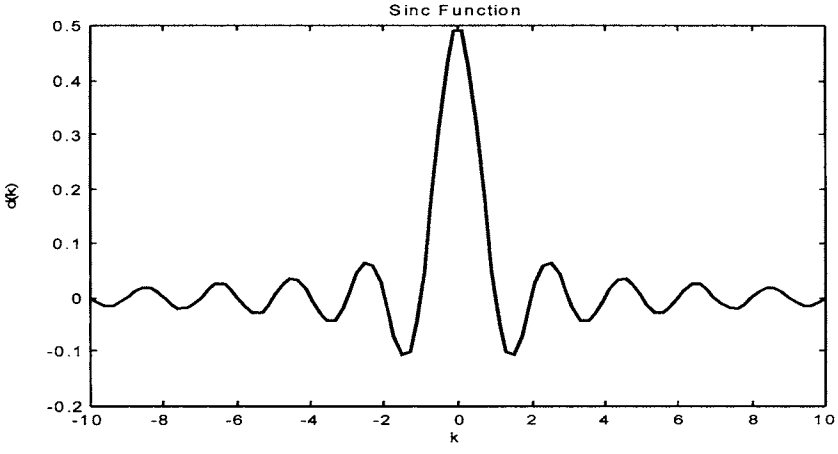

Figure 6.11
Impulse response of the ideal low-pass filter

In order to truncate the impulse response to a finite duration, a window function can be used. The ideal impulse response $d(k)$ is multiplied by a window function, which has a finite duration, resulting in a truncated impulse response. A number of different window functions have been proposed. We shall examine four of them in order to illustrate the effects of windowing and the relative merits of these functions.

It is worth pointing out that in the above example, the frequency response is symmetric about $\omega = 0$ and is real. This results in an even symmetric impulse response that is also real-valued. The phase response is zero for all frequencies.

6.4.1 Rectangular window

The most direct and simple way to truncate the ideal impulse response $d(k)$ is to keep the values of $d(k)$ within a certain interval, say, $-M$ to M. This is equivalent to multiplying $d(k)$ by a rectangular function given by

$$w(n) = \begin{cases} 1, & |n| \le M \\ 0, & \text{otherwise} \end{cases}$$

as shown in Figure 6.12.

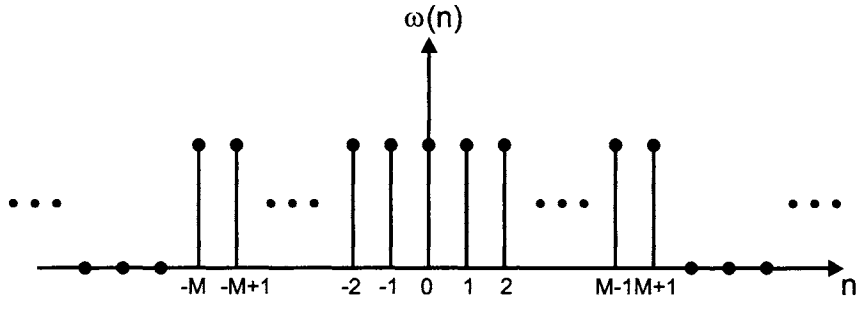

Figure 6.12
A rectangular window

The resulting impulse response $h_w(n)$ has either $N = 2M$ or $N = 2M+1$ non-zero values. In our following discussions, we shall assume that N is odd. The arguments can easily be extended to the case where N is even.

$$h_w(n) = [d_{-M}, d_{M+1}, ..., d_{-1}, d_0, d_1, ..., d_{M-1}, d_M]$$

The windowed impulse response $h_w(n)$ is still non-causal, i.e. it has non-zero values before the time origin n = 0. To make it causal we can simply shift the time origin to the first non-zero sample and re-index the entries. The impulse response (and hence the filter coefficients) of the FIR filter is therefore

$$h(n) = d_w(n-M) \qquad n = 0,1,...,N-1$$

This process is illustrated in Figure 6.13.

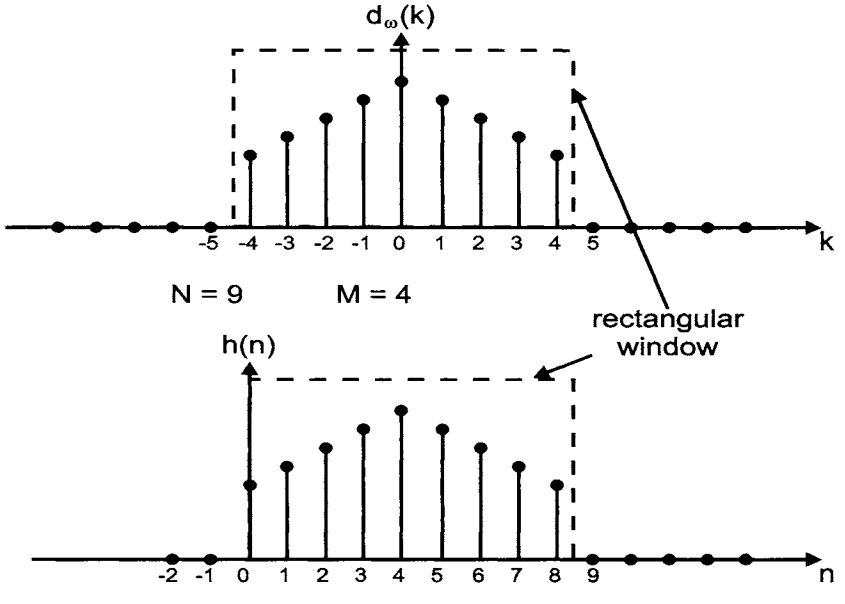

Figure 6.13
Rectangular windowed impulse response

Example 6.2
Find the rectangularly windowed impulse response of an ideal low-pass filter with cut-off frequency

$$\omega_c = \pi/4$$

Assume $N = 11$.
Solution:
Since $N = 11$, $M = (N-1)/2 = 5$.

$$d_w(n) = \frac{\sin(\pi n/4)}{\pi n} \qquad -5 \le n \le 5$$

$$= \left[\frac{\sqrt{2}}{10\pi}, 0, \frac{\sqrt{2}}{6\pi}, \frac{1}{2\pi}, \frac{\sqrt{2}}{2\pi}, \frac{1}{4}, \frac{\sqrt{2}}{2\pi}, \frac{1}{2\pi}, \frac{\sqrt{2}}{6\pi}, 0, \frac{\sqrt{2}}{10\pi}\right]$$

The filter impulse response $h(n)$ is plotted in Figure 6.14.

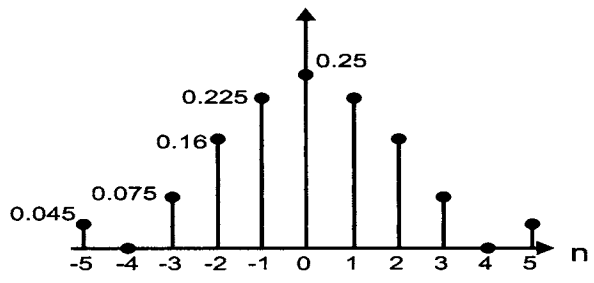

Figure 6.14
Filter impulse response of Example 6.2

6.4.1.1 Performance evaluation

How good is the design using rectangular windows? In other words, how close is the resulting FIR filter frequency response approximation, to the original ideal response $D(\omega)$?

To answer this question, we performed the DTFT of $h(n)$, denoted by $H(\omega)$ and plotted the magnitude response against $D(\omega)$ in Figure 6.15. Note that since $h(n)$ is no longer symmetric about the origin, its DTFT will be complex-valued. Hence we only compare the magnitude response in the frequency range 0 to π.

Notice the ripples in both the passband and the stopband. This can be more clearly seen if we re-design the filter using large values of N. Figures 6.16 and 6.17 show the rectangularly windowed impulse responses and the corresponding magnitude responses for $N = 51$ and $N = 101$.

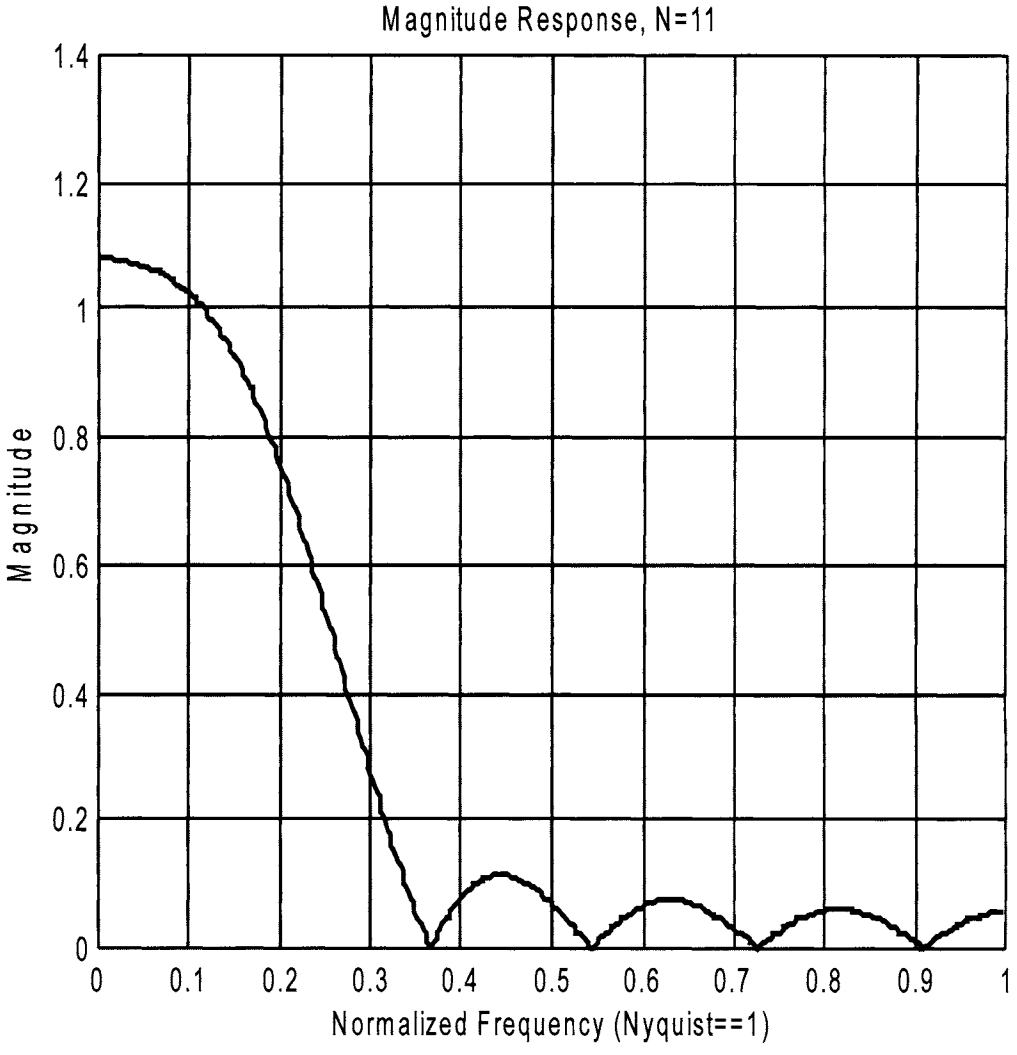

Figure 6.15
Magnitude response of rectangular windowed filter with N=11

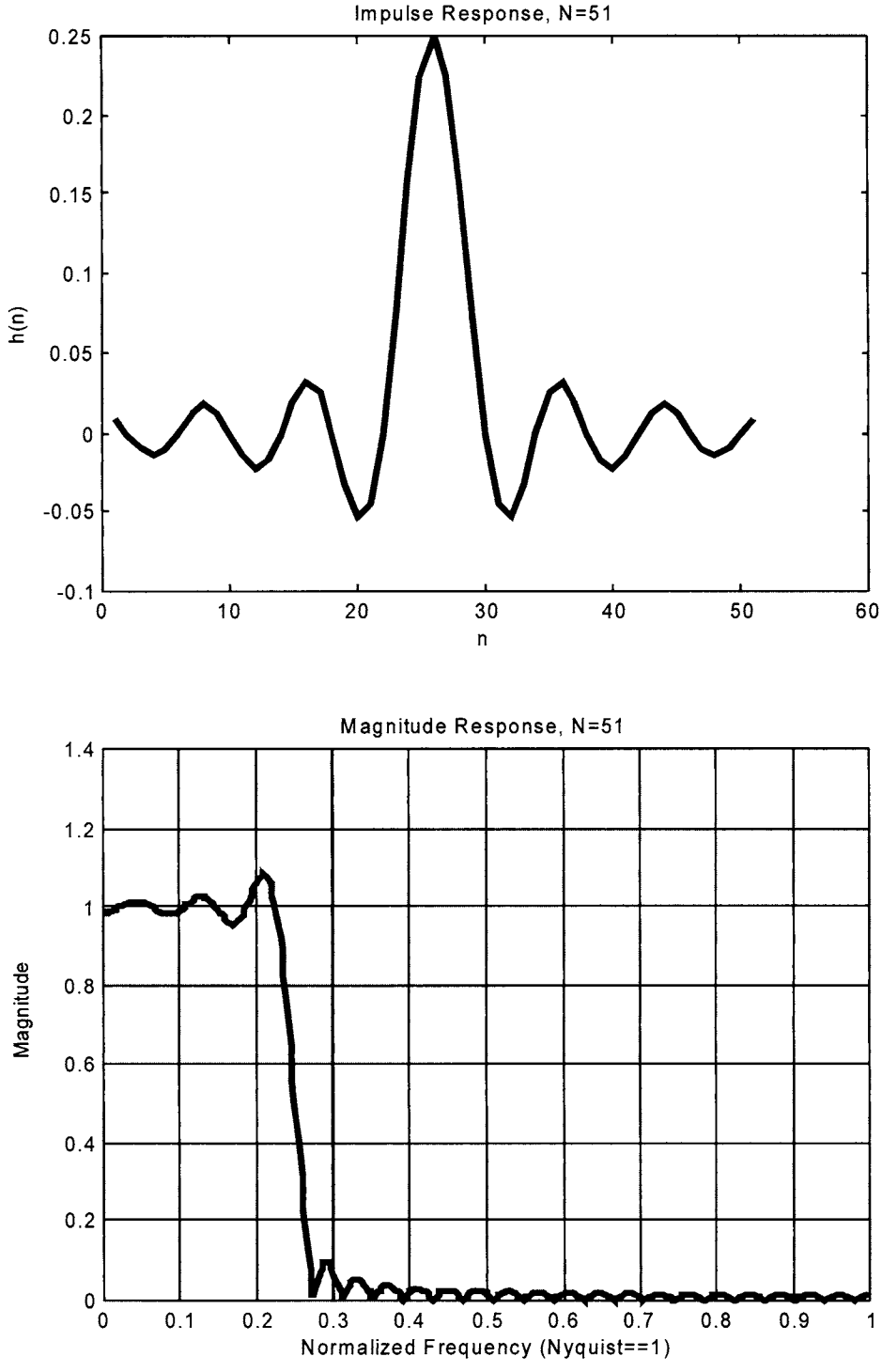

Figure 6.16
Impulse and magnitude responses of truncated ideal LPF with N=51

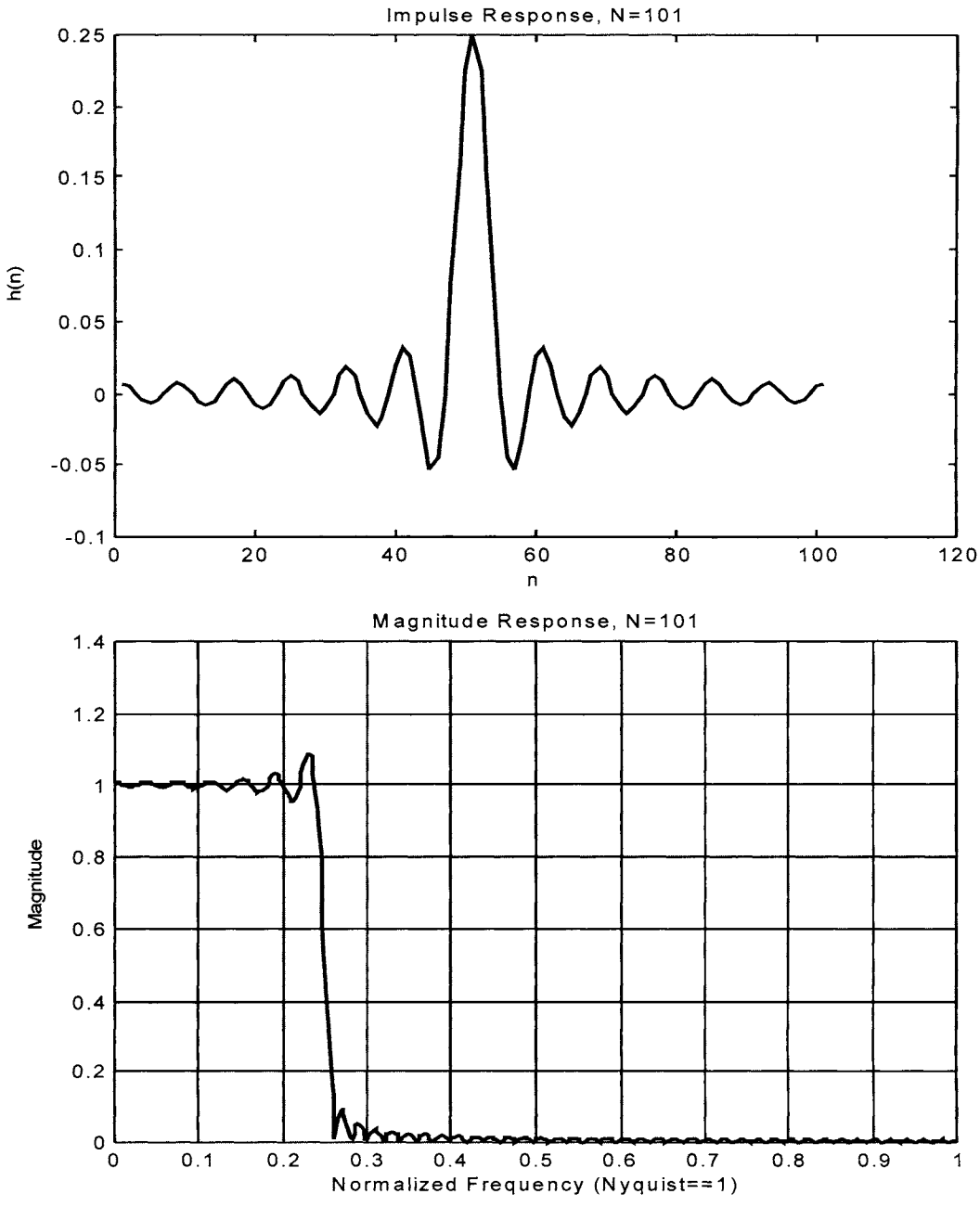

Figure 6.17
Impulse and magnitude responses of truncated ideal LPF with N=101

One would expect that as N increases, the approximation would become better. This is indeed the case except for the region close to the transition between passband and stopband. This area corresponds to a discontinuity in the ideal desired frequency response. The truncation of the Fourier series introduces ripples in the frequency response due to the non-uniform convergence of the Fourier series at a discontinuity. This phenomenon is known as the Gibb's phenomenon. For this reason, the approximation at the band edge will always be poor for the rectangular window design regardless of how large N is.

6.4.1.2 Another interpretation

We can interpret the magnitude response of the FIR filter by using our knowledge of linear systems discussed in the earlier chapter. This is illustrated in Figure 6.18.

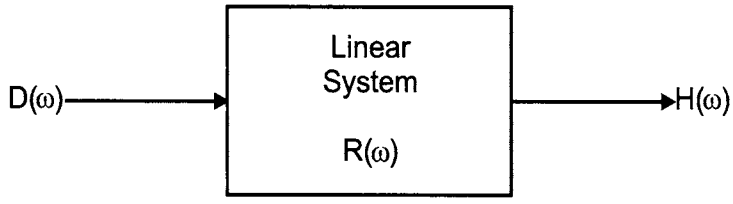

Figure 6.18
Frequency domain interpretation

The original desired magnitude spectrum (or response) is the input to a linear system having the magnitude response of a rectangular window. The output of this system is then the magnitude response of the FIR filter. According to linear system theory, the output is the product of the input and the system responses:

$$H(\omega) = R(\omega) = D(\omega)$$

But the magnitude response of a rectangular window has the form shown in Figure 6.19. This interpretation shows that the rectangular window introduces the ripples and ringing in the FIR filter response.

If we want to design filters with better approximation in the transition region, this interpretation tells us that we need to use a window with better magnitude response. This is the reason why a number of different window functions with different frequency characteristics are proposed.

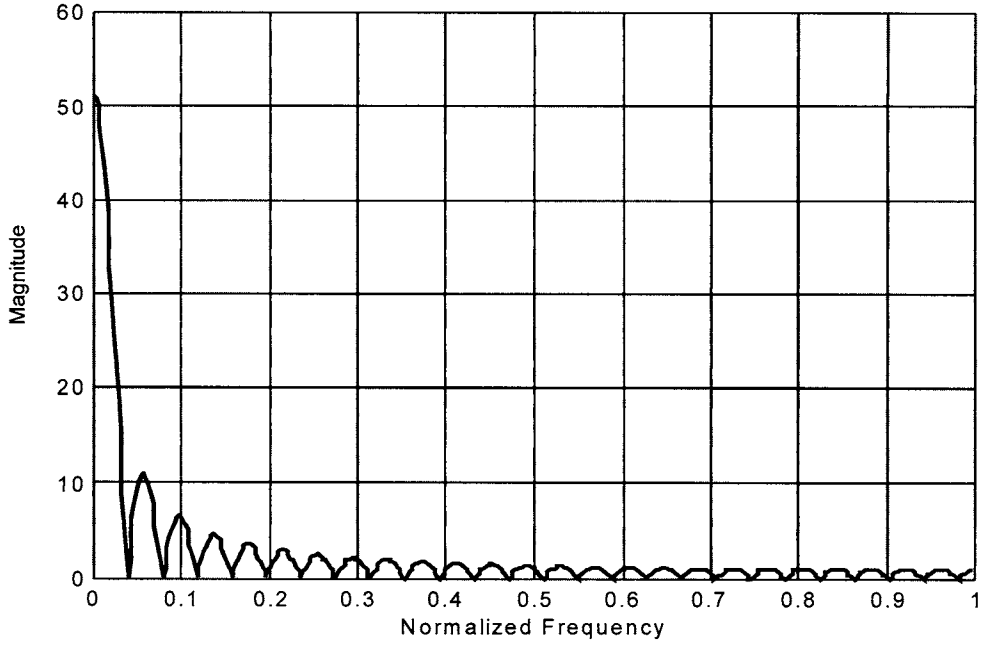

Figure 6.19
Magnitude response of a rectangular window

6.4.1.3 Summary of filter characteristics

- The ripple size decreases with increasing filter order N. Approximation well within the passband and stopband becomes better as the filter order increases.
- The transition width decreases with increasing filter order. For any order N, the filter response is always equal to 0.5 at the cut-off frequency.
- Ripple size near the passband to stopband transition remains roughly the same as N increases. The maximum ripple size is about 9%. This is known as the Gibb's phenomenon.

6.4.2 Hamming window

Since discontinuities in the time function give rise to ringing in the frequency response, we can replace the rectangular window with a window function that tapers off smoothly at both ends. This will reduce the ripple effect. The Hamming window is a popular one in this class of window functions.

The Hamming window is defined mathematically as

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \qquad n = 0,1,...,N-1$$

Figure 6.20 plots this window function.

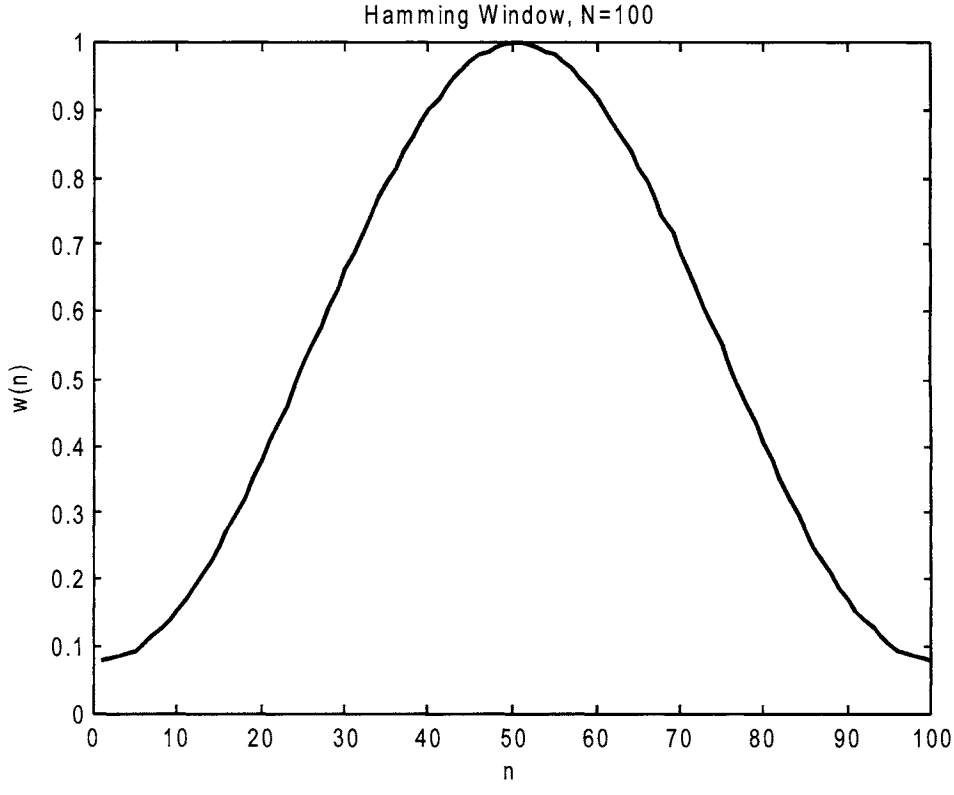

Figure 6.20
The Hamming window

Notice that this equation defines the window samples as already shifted (indices from 0 to N–1). So the impulse response of the FIR low-pass filter designed using the Hamming window is

$$h(n) = w(n)d(n-M)$$
$$= \left[0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)\right] \cdot \frac{\sin[(n-M)\omega_c]}{(n-M)\pi}$$

Figure 6.21 shows a length-51 low-pass filter with cut-off at $\pi/4$ as in the example in the previous section.

Comparing this response with that shown in Figure 6.16, which was designed using the rectangular window, it is obvious that the Hamming window design is better. The ripples in the both the passband and the stopband are virtually eliminated. The cost involved is a wider transition width.

The Hamming window function has the same form as the raised cosine function familiar to digital communication engineers. The only differences are in the scalar value in the constant and cosine terms. The Hamming window does not taper to end values of zero. Instead it goes to a value of 0.08. The maximum stopband ripple is about 53 dB below the passband gain.

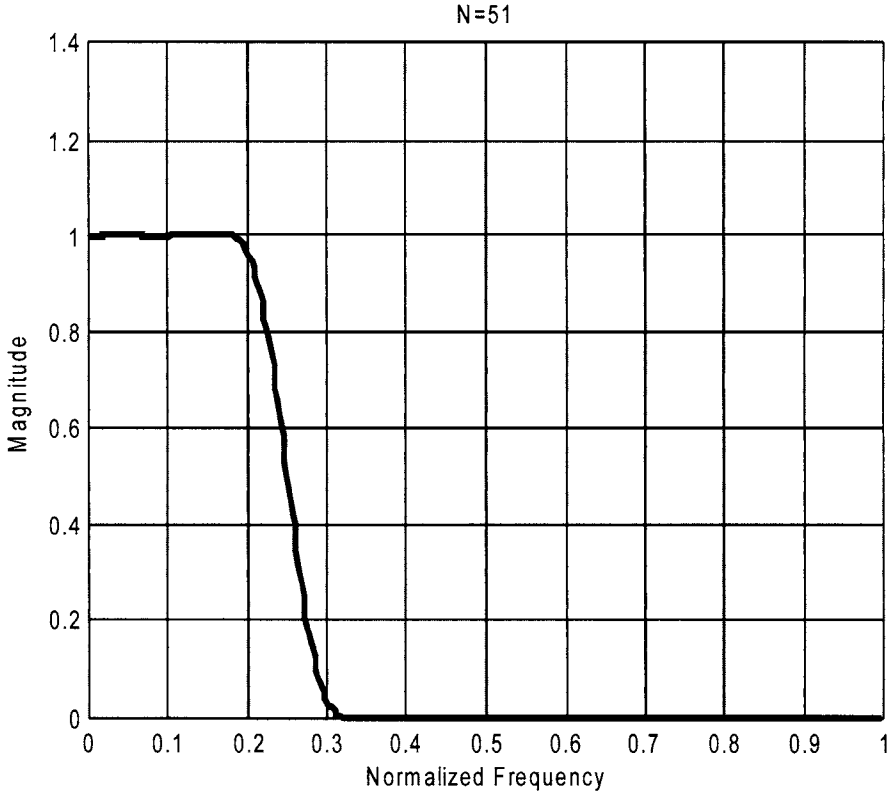

Figure 6.21
Hamming windowed magnitude response

6.4.3 Blackman window

The Blackman window exhibits an even lower maximum stopband ripple (about 74 dB down) in the resulting FIR filter than the Hamming window. It is defined mathematically as

$$w(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right) \qquad n = 0,1,\ldots,N-1$$

Its magnitude and impulse responses are plotted in Figure 6.22. Note that the width of the main lobe in the magnitude response is about 50% wider than that of the Hamming window.

A length-51 low-pass FIR filter is designed using this window and the responses shown in Figure 6.23. This can be compared to the one designed using Hamming window in Figure 6.21.

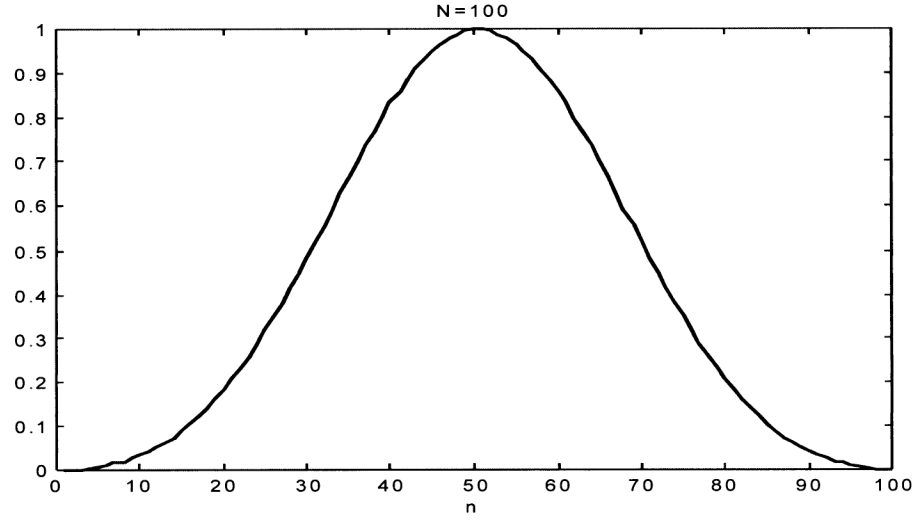

Figure 6.22
The Blackman window

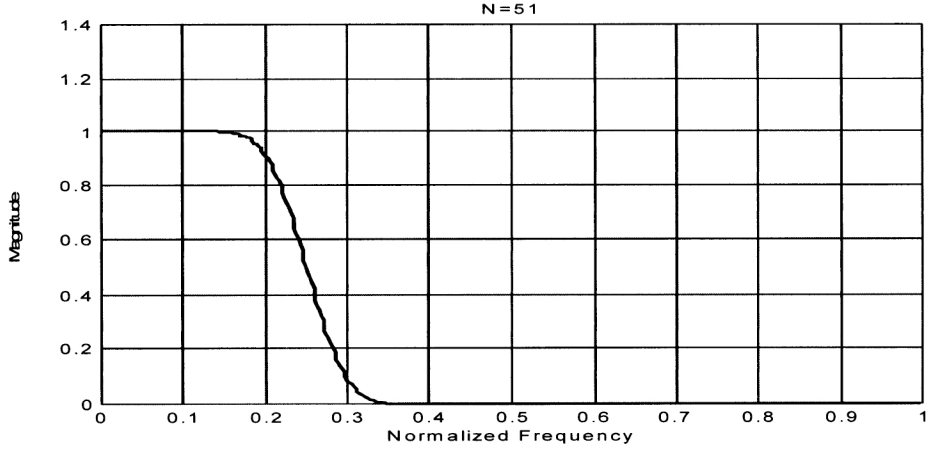

Figure 6.23
Low-pass FIR filter designed using the Blackman window

6.4.4 Kaiser window

The main advantage of the previous three window functions is that they are simple to apply and the resulting filter characteristics are reasonably good. For a large number of applications, Hamming or Blackman window designs will be sufficient to satisfy the specifications. The major drawback of these window functions is that its characteristics such as maximum stopband attenuation and the amount of overshoot are basically fixed. So if the filter specifications include the amount of overshoot and passband-to-stopband transition width, for instance and if the above window functions do not produce designs that can satisfy them, then we are stuck.

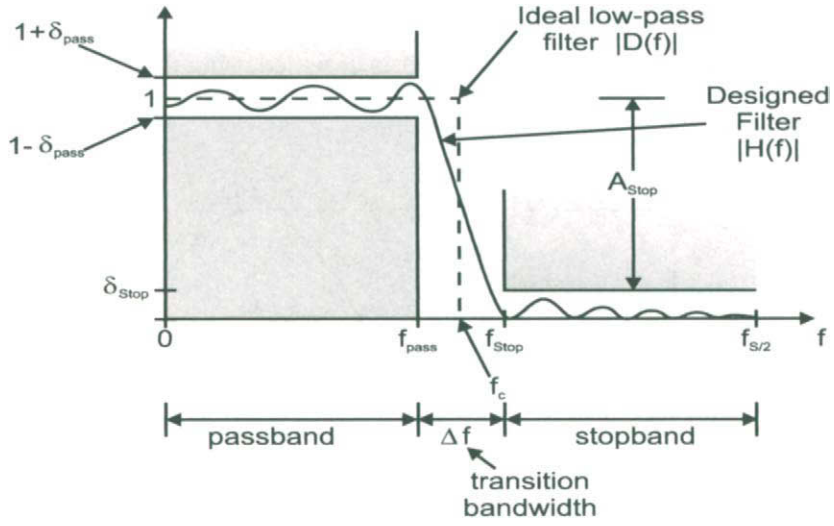

Figure 6.24
Magnitude response characteristics of a low-pass filter

Consider the magnitude response characteristics of a low-pass filter as shown in Figure 6.24. The ideal cut-off frequency is at the midpoint between the passband and stopband edge frequencies.

$$f_c = \frac{1}{2}\left(f_{\text{pass}} + f_{\text{stop}}\right)$$

The transition width is defined as

$$\Delta f = f_{\text{stop}} - f_{\text{pass}}$$

The normalized frequencies are the digital frequencies:

$$\omega_{\text{pass}} = \frac{2\pi f_{\text{pass}}}{f_s}$$

$$\omega_{\text{stop}} = \frac{2\pi f_{\text{stop}}}{f_s}$$

$$\omega_c = \frac{2\pi f_c}{f_s}$$

$$\Delta\omega = \frac{2\pi\Delta f}{f_s}$$

The maximum passband and stopband ripples are usually expressed in decibels (dB) in practice:

$$A_{\text{pass}} = 20\log_{10}\left(\frac{1+\delta_{\text{pass}}}{1-\delta_{\text{pass}}}\right)$$

$$A_{\text{stop}} = -20\log_{10}\delta_{\text{stop}}$$

These equations relate the two sets of specifications $\{f_{\text{pass}}, f_{\text{stop}}, A_{\text{pass}}, A_{\text{stop}}\}$ and $\{f_c, \Delta f, \delta_{\text{pass}}, \delta_{\text{stop}}\}$.

If δ_{pass} is small, then we can use a first order approximation to get

$$A_{\text{pass}} = 17.372\delta_{\text{pass}}$$

Note that it is a property of all window designs that δ_{pass} and δ_{stop} are equal in the filter designed. Therefore, instead of dealing with two variables, we can choose the maximum ripple to be the smaller of the two:

$$\delta = \min(\delta_{\text{pass}}, \delta_{\text{stop}})$$

Practical choices of passband and stopband attenuation will usually result in the stopband ripple being smaller than the passband one.

6.4.4.1 Kaiser window design

Kaiser has developed a flexible family of window functions. This family of window functions has adjustable shape parameters that allow the designer to achieve the specified ripple and attenuation. It is mathematically defined as

$$w(n) = \frac{I_0\left(\beta\sqrt{1-(n-M)^2/M^2}\right)}{I_0(\beta)} = \frac{I_0\left(\beta\sqrt{n(2M-n)/M}\right)}{I_0(\beta)}$$

for $n = 0,1,\ldots,N-1$ where $I_0(x)$ is the zero-th order modified Bessel function of the first kind. Here we assumed as before that N is odd. The second format is more convenient for numerical computations. β is called the shape parameter.

The Kaiser window is symmetric about its midpoint and has a maximum value of 1 at that point. Since $I_0(0)=1$, the end-points have the value $1/I_0(\beta)$. Typical values of β are in the range of

$$4 < \beta < 9$$

Figure 6.25 shows three Kaiser windows with $N = 51$ and $\beta = 5,7,9$.

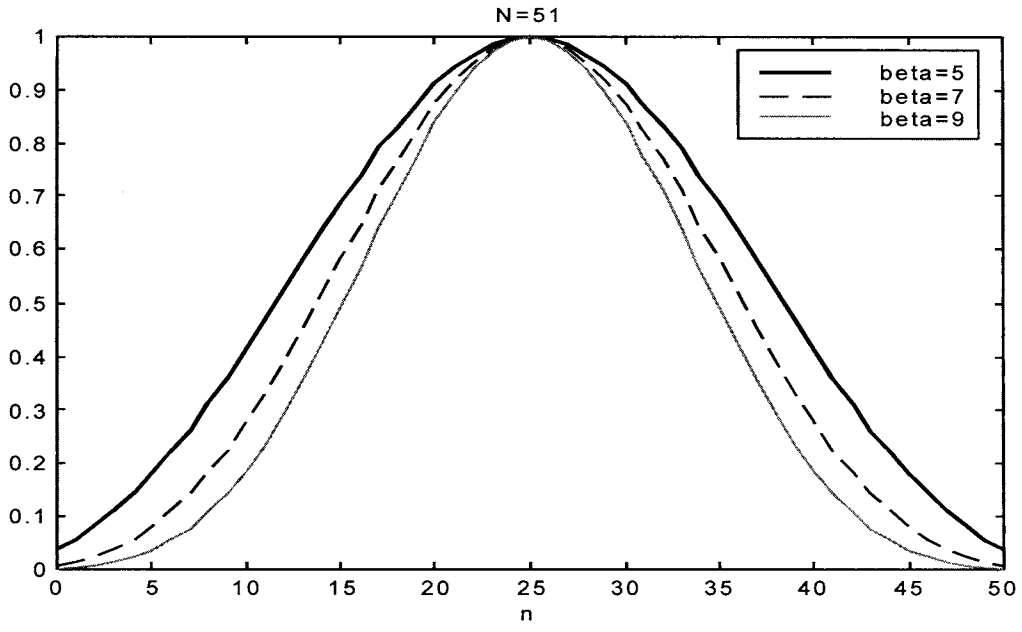

Figure 6.25
Kaiser windows

The Kaiser window is reduced to a rectangular one for $\beta = 0$. It resembles the Hamming window for $\beta = 5$, except near the end-points.

Table 6.1 compares the transition bandwidth and the maximum stopband ripple for various values of the shape parameter.

β	Transition bandwidth	Max. stopband ripple (dB)
4.0	2.6	–45
5.0	3.2	–54
6.0	3.8	–63
7.0	4.5	–72
8.0	5.1	–81
9.0	5.7	–90

Table 6.1
Comparison of transition bandwidth and maximum stopband ripple

The filter order N and the shape parameter can be calculated from the specifications. Kaiser has derived empirical design formulas as follows:

$$\beta = \begin{cases} 0.1102(A-8.7), & A \geq 50 \\ 0.5842(A-21)^{0.4} + 0.07886(A-21), & 21 < A < 50 \\ 0, & A \leq 21 \end{cases}$$

where A is the ripple in dB.

The filter length or order is inversely related to the transition bandwidth:

$$\Delta f = \frac{Df_s}{N-1} \Leftrightarrow N-1 = \frac{Df_s}{\Delta f}$$

where D is a factor computed from A:

$$D = \begin{cases} \dfrac{A-7.95}{14.36} & A > 21 \\ 0.922 & A \leq 21 \end{cases}$$

6.4.4.2 Design steps

- Compute f_c and Δf and then the normalized digital frequencies.
- Compute the passband and stopband ripples and hence δ and A.
- Compute β and D.
- Compute the filter length required and round it up to the next odd integer.
- Compute the window function $w(n)$.
- Compute the windowed impulse response $h(n)$.

Note that the parameters N and β depend only on A and Δf and not on f_c. However, $h(n)$ does depend on f_c.

Example 6.3
Design a low-pass FIR filter using Kaiser windows with the following specifications:

$$f_s = 44.1\,\text{kHz}$$
$$f_{pass} = 12\,\text{kHz}$$
$$f_{stop} = 18\,\text{kHz}$$
$$A_{pass} = 0.2\,\text{dB}$$
$$A_{stop} = 50\,\text{dB}$$

Solution:
Now we have A = 50. Therefore,

$$\begin{aligned} \beta &= 0.1102(A-8.7) \\ &= 0.1102(50-8.7) \\ &= 4.55126 \end{aligned}$$

and

$$\begin{aligned} D &= \frac{A-7.95}{14.36} \\ &= 2.9283 \end{aligned}$$

Since

$$\begin{aligned} \Delta f &= f_{stop} - f_{pass} \\ &= 18-12 \\ &= 6\ \text{kHz} \end{aligned}$$

we have

$$N-1=\frac{Df_s}{\Delta f}$$
$$=\frac{2.9283(44.1)}{6}$$
$$=21.5$$

and so the order of the filter is

$$N=22.5$$

We can try to use $N = 22$ or $N = 23$.

MATLAB functions provide a better approximation for β and N. Figure 6.26 shows the magnitude response of the resulting filter.

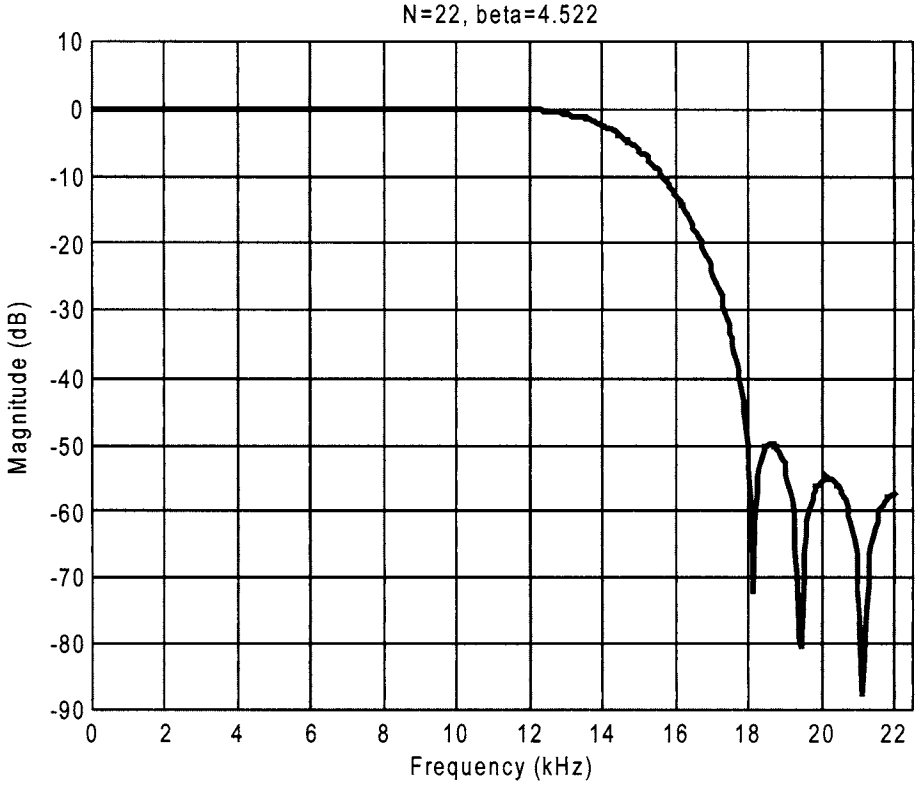

Figure 6.26
Magnitude response of Kaiser window designed FIR filter

6.4.4.3 High-pass filter design

High-pass filter design using Kaiser windows is very similar to low-pass filter design. The only change in the steps is simply define

$$\Delta f = f_{\text{pass}} - f_{\text{stop}}$$

since the role of f_{pass} and f_{stop} are interchanged.

The ideal high-pass impulse response is obtained from the inverse DTFT of the ideal high-pass frequency response. It is given by

$$d(k) = \delta(k) - \frac{\sin(\omega_c k)}{\pi k}$$

The windowed filter impulse response is therefore

$$h(n) = w(n) \cdot \left[\delta(n-M) - \frac{\sin((n-M)\omega_c)}{(n-M)\pi} \right]$$

$$= \delta(n-M) - w(n) \cdot \frac{\sin((n-M)\omega_c)}{(n-M)\pi}$$

The second term on the right-hand side of this equation is the impulse response of the low-pass filter with the same cut-off frequency.

Note that with the same value of ω_c, the low-pass and high-pass filters are complementary. That is,

$$h_{LP}(n) + h_{HP}(n) = \delta(n-M) \qquad n = 0,1,...,N-1$$

Example 6.4

Two-way crossover filters. Conventional loudspeakers make use of an analog crossover network to split the audio signal into its low frequency and high frequency components. The low frequency signal drives the woofer and the high frequency one drives the tweeter. Digital loudspeaker systems use digital filters to perform the same function on the incoming digitized audio signal. The digital signals in the two frequency bands are then converted to analog signals, amplified and drive the corresponding parts of the loudspeaker.

Let the crossover frequency be 1 kHz. The low-pass filter has specifications:

$$f_{pass} = 800\,\text{Hz}$$

$$f_{stop} = 1200\,\text{Hz}$$

$$A_{pass} = 0.1\,\text{dB}$$

$$A_{stop} = 60\,\text{dB}$$

Note that we only need to design the low-pass filter. The high-pass filter is complementary.

The resulting low-pass and high-pass filter magnitude responses are shown in Figure 6.27.

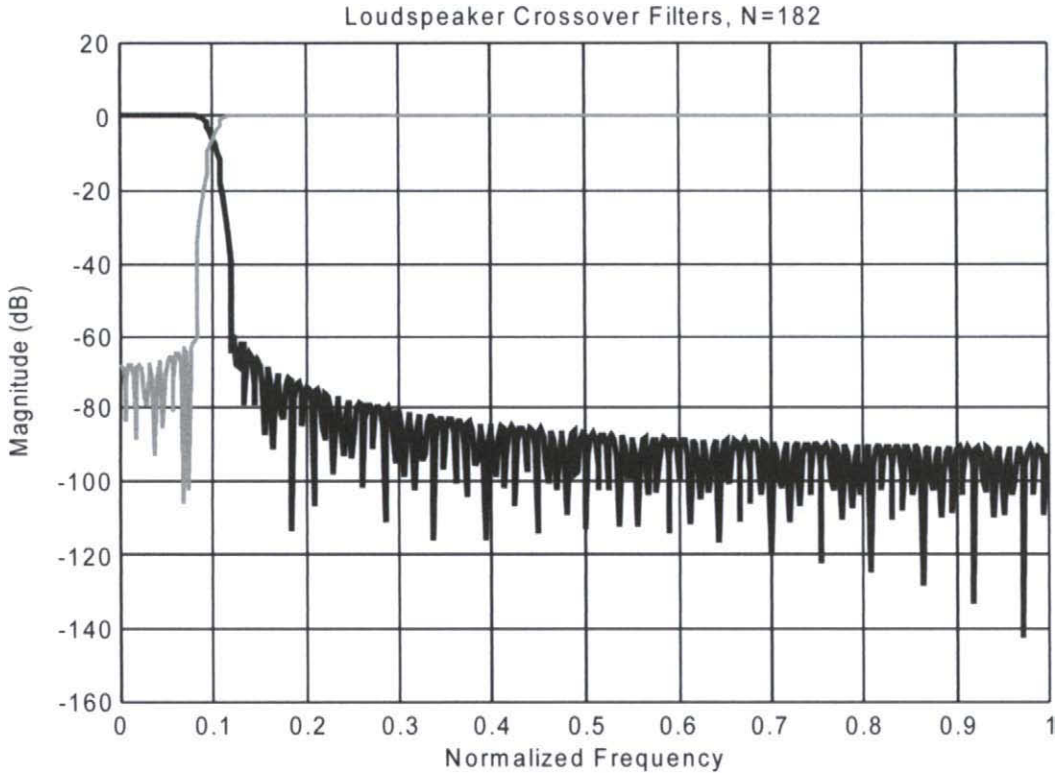

Figure 6.27
Crossover filter responses

The complementary relationship between the two filters leads to a very efficient implementation shown in Figure 6.28.

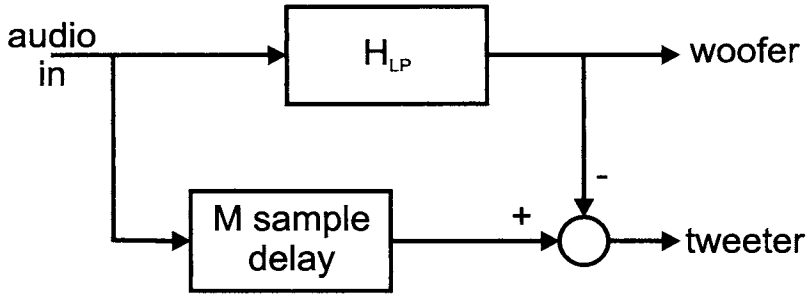

Figure 6.28
Implementation of complementary relationship between two filters

In fact, only one filter is needed; the high-pass output is obtained by combining the low-pass output and the input delayed by M sampling instants.

6.4.4.4 Band-pass filter design

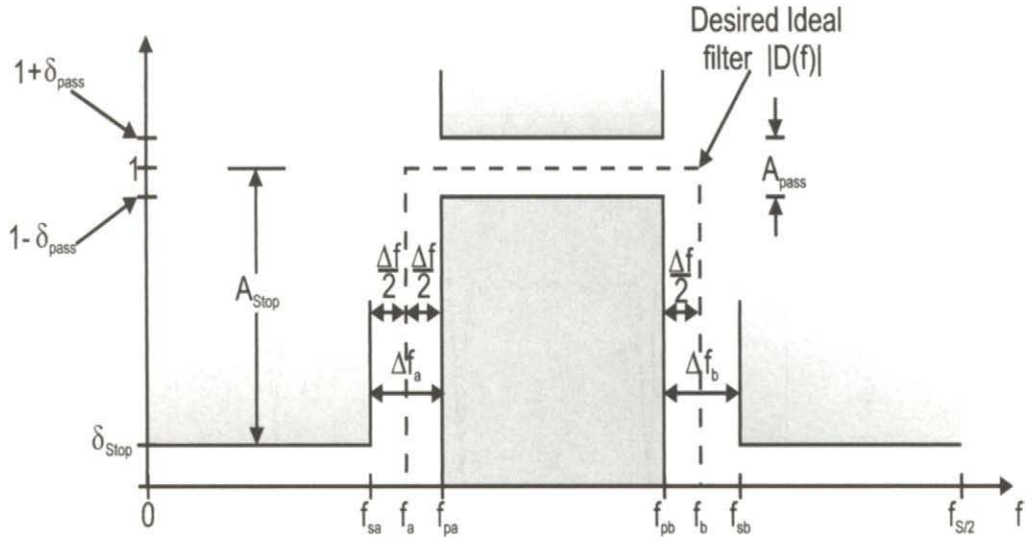

Figure 6.29
Typical specifications for a band-pass filter

Typical specifications for a band-pass filter are shown in Figure 6.29. There are two stopbands and two transition bands. The final design will have equal transition bandwidths. If the transition bandwidths of the original specifications are not equal, the smaller one will be used.

The ideal cut-off frequencies are defined in the same way as in the low-pass case:

$$f_a = f_{pa} - \frac{1}{2}\Delta f$$

$$f_b = f_{pb} + \frac{1}{2}\Delta f$$

The window parameters can then be calculated. The band-pass impulse response is given by

$$h(n) = w(n) \cdot \frac{\sin((n-M)\omega_b) - \sin((n-M)\omega_a)}{(n-M)\pi}$$

for $n = 0,1,\ldots,N-1$ where

$$h(M) = \frac{\omega_b - \omega_a}{\pi}$$

Example 6.6
Five-band graphic equalizer. The crossover frequencies of the 5 bands are 3 kHz, 7 kHz, 11 kHz, and 15 kHz.

The resulting filter magnitude responses are shown in Figure 6.30.

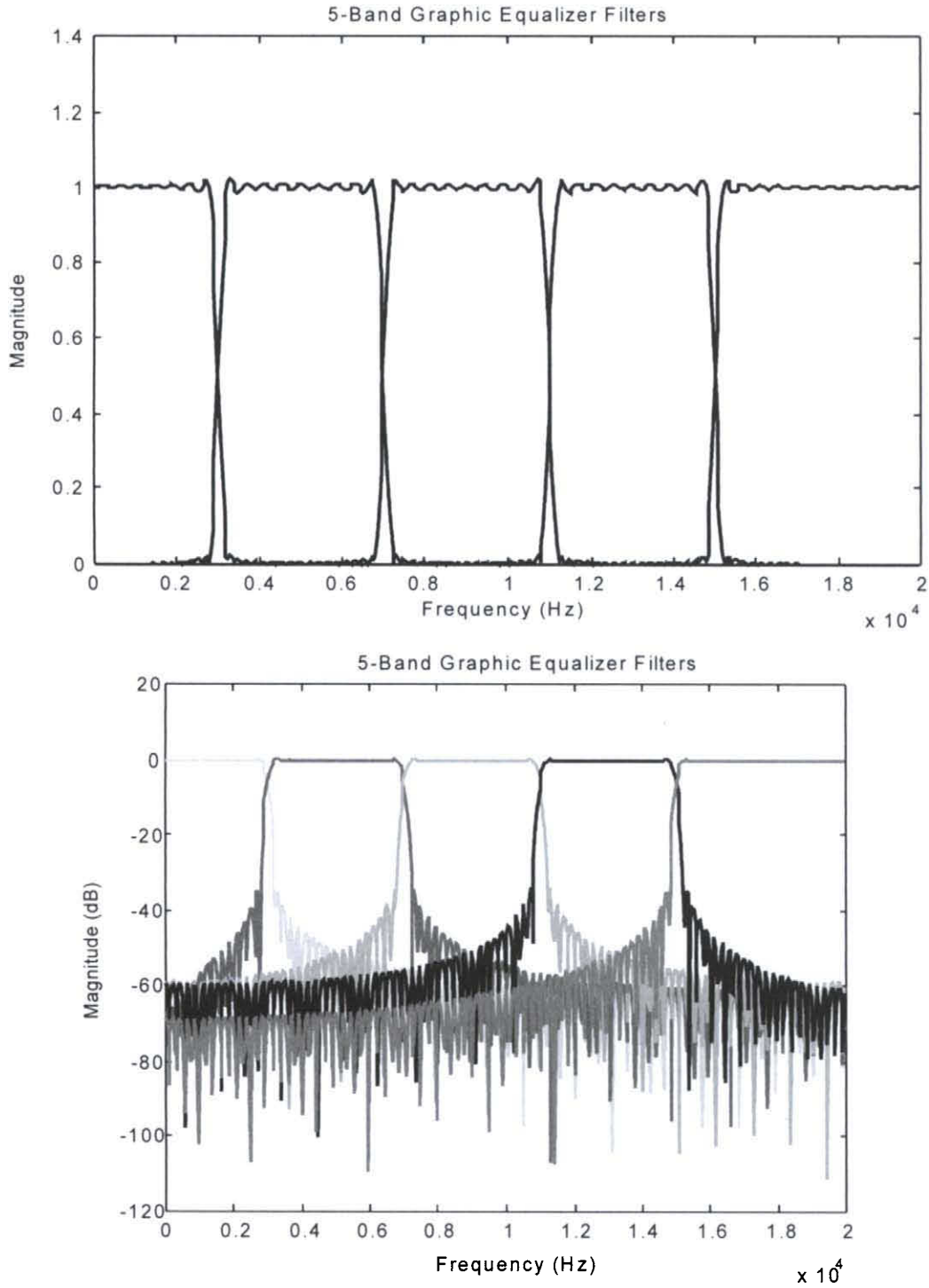

Figure 6.30
Magnitude responses of the five filters in the graphic equalizer

The final high-pass filter is complementary to the sum of the first 4 filters. An implementation using four filters is shown in Figure 6.31.

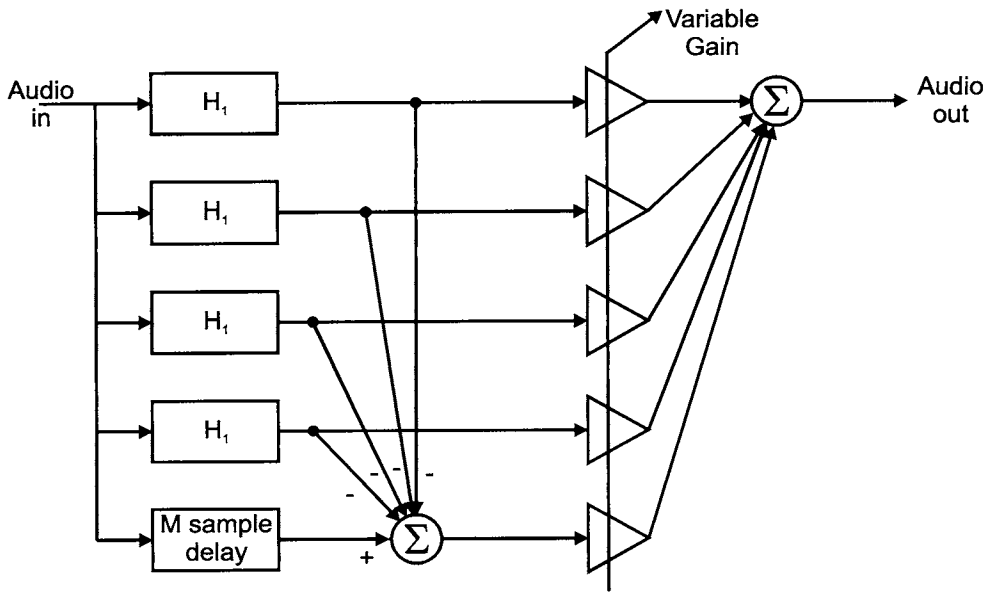

Figure 6.31
An implementation of the 5-band graphic equalizer

In practice, the digital filters employed for digital graphic equalizers are typically second order IIR filters.

6.4.4.5 Remarks

This is a particular example of a filter bank where the input is split into a number of non-overlapping frequency bands. Filter banks have been used very successfully in speech coding – a technique known as sub-band coding. The output of each filter is quantized to a different resolution. The allocation of bits in each band is usually governed by psychoacoustic perceptual criteria. To put simply, fewer bits are assigned to bands that are less audible.

There have been a lot of research activities in the design of filter banks in the last decade. This research has led to the development of wavelet transformation and wavelet filter banks, which is still a very active research area. However, it is beyond the scope of this introductory course.

6.4.4.6 Computation of Bessel functions

To complete our discussions on Kaiser window design method, we shall consider the computation of the Bessel function. The zero-th order Bessel function of the first kind can be defined by its Taylor series expansion:

$$I_0(x) = \sum_{k=0}^{\infty} \left[\frac{(x/2)^k}{k!} \right]^2$$

Evaluation of $I_0(x)$ is over the range limited by the shape parameter and

$$0 \le x \le \beta$$

The Taylor series can be recursively computed to a desired level of accuracy. Define the partial sum

$$S_n = \sum_{k=0}^{n} \left[\frac{(x/2)^k}{k!} \right]^2$$

and the term

$$D_n = \left[\frac{(x/2)^n}{n!} \right]^2$$

Algorithm:

- Initialize

$$S_0 = 1$$
$$D_0 = 1$$

- For n ≥ 1, compute

$$D_n = \left(\frac{x}{2n} \right)^2 D_{n-1}$$
$$S_n = S_{n-1} + D_n$$

- Above step is repeated for successive values of n until the ratio

$$\frac{D_n}{S_n} = \frac{S_n - S_{n-1}}{S_n} < \varepsilon$$

where ε is a small number (say, 10^{-9}).

6.5 Frequency sampling method

The window method of FIR filter design requires the inverse DTFT of the desired frequency response. While the calculations may be straightforward for simple ideal low-pass, band-pass and high-pass responses, it may not be the case for an arbitrary filter response such as the ERMES pre-modulation filter specifications described in the previous chapter. Instead of considering the continuous frequency response, we can take samples of it and deal with the discrete spectrum.

Samples of the desired frequency response $D(\omega)$ are taken at N uniformly spaced frequencies ω_k within the interval $(0,2\pi)$. If N is also the order of the FIR filter to be designed, then the coefficients $h(n)$ can be found by solving the N simultaneous equations:

$$\sum_{n=0}^{N-1} h(n) e^{-j2\pi nk/N} = D\left(\frac{2\pi k}{N} \right) \qquad k = 0,1,\ldots,N-1$$

This approach makes sure that the frequency response of the FIR filter will pass through those sampled frequency points.

The disadvantage of this direct approach is the computational complexity. Solving N simultaneous equations requires on the order of N^3 arithmetic operations. While this is acceptable for small values of N, it becomes prohibitive when N increases.

Recalling that the relationship between the uniformly spaced discrete frequency samples and the discrete-time impulse response is given by the DFT, the inverse DFT (IDFT) of the frequency samples will give the impulse response. IDFT only requires approximately N^2 arithmetic operations. If FFT is used, the number of operations is reduced to $N \log N$.

6.5.1 Design formulas

Explicit formulas can be derived for the four types of linear phase FIR filters described in section 6.3.2. These formulas are simplified from the IDFT equation by making use of the fact that the impulse responses of linear phase FIR filters are real-valued and symmetric (or anti-symmetric).

- **Type 1**

  N is odd and $M = (N-1)/2$.

  where A_k are the equally spaced DFT samples at frequencies

$$\omega_k = \frac{2\pi k}{N} \qquad k = 0,1,...,N-1$$

  If no sample at $\omega = 0$ is included, then

$$\omega_k = \frac{(2k+1)\pi}{N} \qquad k = 0,1,...,N-1$$

  and the design formula becomes

$$h(n) = \frac{1}{N}\left[\sum_{k=0}^{M-1} 2A_k \cos\left(\frac{2\pi(n-M)\left(k+\frac{1}{2}\right)}{N}\right) + A_M \cos\pi(n-M)\right]$$

- **Type 2**

  N is even.

  If a zero frequency sample is available, the design formula is:

$$h(n) = \frac{1}{N}\left[A_0 + \sum_{k=1}^{N/2-1} 2A_k \cos\left(\frac{2\pi(n-M)k}{N}\right)\right]$$

  This formula is essentially the same as that for type 1 filters except for the upper limit of the summation and

$$A_{N/2} = 0$$

  If no zero frequency sample is available, the design formula becomes

$$h(n) = \frac{1}{N}\left[\sum_{k=0}^{N/2-1} 2A_k \cos\left(\frac{2\pi(n-M)\left(k+\frac{1}{2}\right)}{N}\right)\right]$$

$$h(n)=\frac{1}{N}\left[A_0+\sum_{k=1}^{M}2A_{\mathrm{k}}\cos\left(\frac{2\pi(n-M)k}{N}\right)\right]$$

- **Type 3**
  The design formulas for anti-symmetric impulse responses involve terms with the sine function instead of the cosine function. The design formulas are

$$h(n)=\frac{1}{N}\left[\sum_{k=1}^{M}2A_k\sin\left(\frac{2\pi(M-n)k}{N}\right)\right]$$

$$h(n)=\frac{1}{N}\left[\sum_{k=0}^{M-1}2A_k\sin\left(\frac{2\pi(M-n)\left(k+\frac{1}{2}\right)}{N}\right)\right]$$

  respectively for the cases where a zero frequency sample is and is not available.

- **Type 4**
  The corresponding design formulas when N is even are

$$h(n)=\frac{1}{N}\left[\sum_{k=1}^{N/2-1}2A_{\mathrm{k}}\sin\left(\frac{2\pi(M-n)k}{N}\right)+A_{\mathrm{N/2}}\sin\pi(M-n)\right]$$

$$h(n)=\frac{1}{N}\left[\sum_{k=0}^{N/2-1}2A_{\mathrm{k}}\sin\left(\frac{2\pi(M-n)\left(k+\frac{1}{2}\right)}{N}\right)\right]$$

6.5.2 Transition region

Let us consider the design of a linear phase FIR filter to approximate the ideal low-pass response with a passband from 0 to 0.4π (normalized). The frequency samples are given by

$$|D(\omega_{\mathrm{k}})|=\begin{cases}1, & k=0,1,...,P\\ 0, & k=P+1,...,M\end{cases}$$

With N = 40, P = 8. The DFT samples are therefore

$$A_{\mathrm{k}}=\begin{cases}(-1)^k/N, & k=0,1,...,P\\ 0, & k=P+1,...,M\end{cases}$$

Choosing type 2 design since N is even and a zero frequency sample is available, we arrive at a FIR filter with magnitude response shown in Figure 6.32.

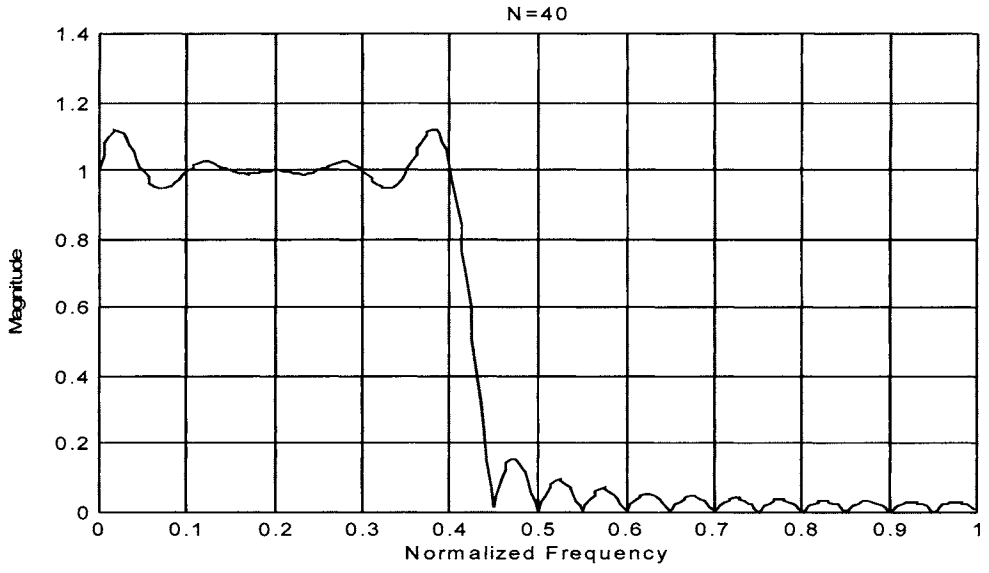

Figure 6.32
FIR filter designed using frequency sampling

The resulting filter response is very similar to the one we arrived at using the rectangular window. The amount of overshoot is relatively large near the band edge and the minimum stopband attenuation is particularly disappointing (at around –20 dB). The reason is that the transition bandwidth is too narrow. If a transition sample is added which has a magnitude that is halfway between the passband and stopband

$$A_p = 0.5(-1)^p / N$$

then the magnitude response is greatly improved as shown in Figure 6.33.

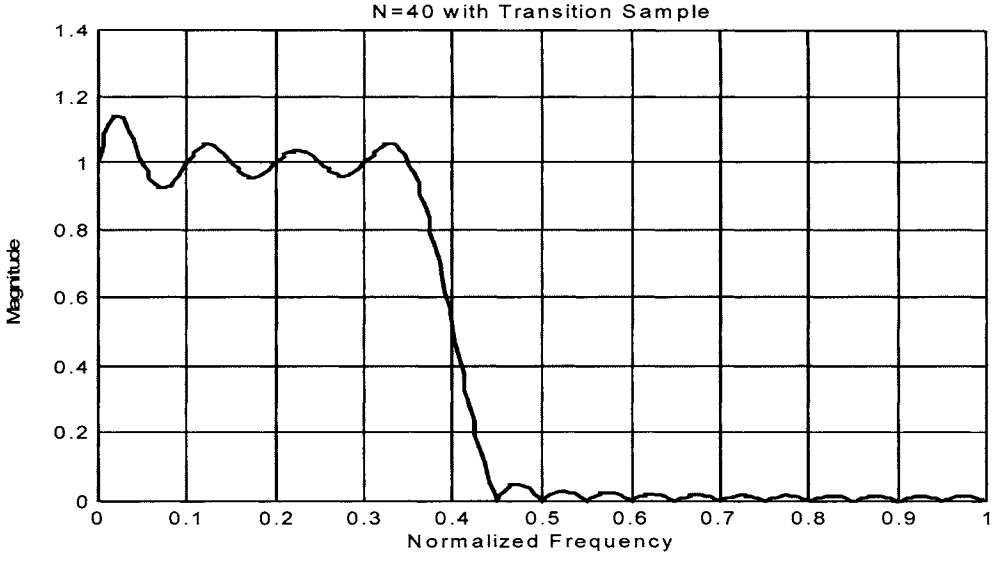

Figure 6.33
Re-design of the same filter as Figure 6.32

The stopband attenuation is now about –30 dB. Adjusting the value of A_P can make further improvements. With

$$A_p = 0.4(-1)^p / N$$

a stopband attenuation of around –40 dB can be achieved.

This example shows that a transition region is very important in the resulting filter performance. In practice, the transition bandwidth is usually dictated by other considerations. However, if there is a freedom of choice, it can be adjusted to give optimal performance given a certain filter length.

Example 6.6
Consider one more filter design with the following specifications:
Passband: 0 to 0.08π
Stopband: 0.2π to π
Minimum stopband attenuation is 40 dB
The transition bandwidth in this case is $0.12\ \pi$; the minimum spacing between frequency samples is $0.06\ \pi$. Hence the length of the filter is

$$N \geq \frac{2\pi}{0.06\pi} = 34$$

The minimum spacing is not sufficient since the stopband will have to start at the fourth sample (0.18π). In order to put a sample at 0.2π, a frequency spacing of 0.05π can be chosen which corresponds to $N = 40$. The passband will be extended to 0.1π and the transition sample will be at 0.15π.
The DFT samples are:

$$A_0 = A_2 = \frac{1}{40}$$

$$A_1 = -\frac{1}{40}$$

$$A_3 = -\frac{0.4}{40} = -0.01$$

$$A_k = 0 \quad \text{for } k \geq 4$$

Type 2 design with zero frequency samples results in a filter response shown in Figure 6.34. The filter specifications are satisfied.

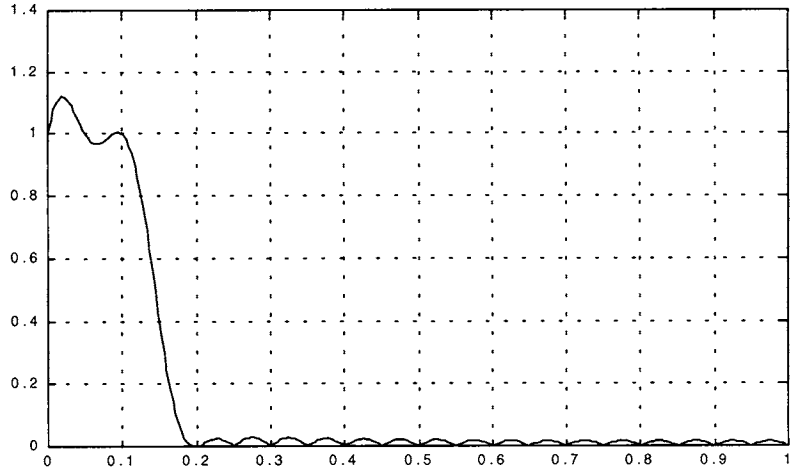

Figure 6.34
Filter response of Example 6.7

Alternatively, we can choose not to include a zero frequency sample. If the frequency sample spacing of 0.06π is used, the passband will be extended to 0.09π but the stopband edge will be at 0.21π, which is too high. Setting

$$\omega_4 = 3.5\Delta\omega = 0.2\pi$$

where $\Delta\omega$ is the sample spacing, we arrive at

$$\Delta\omega = 0.0571\pi$$

or

$$N \geq \frac{2\pi}{0.0571\pi} = 35$$

The passband edge is now at 0.0857π. The DFT samples are now

$$A_0 = \frac{1}{35}$$

$$A_1 = -\frac{1}{35}$$

$$A_2 = \frac{0.4}{35}$$

$$A_k = 0 \qquad \text{for } k \geq 3$$

The resulting filter magnitude response is shown in Figure 6.35. All specifications are satisfied.

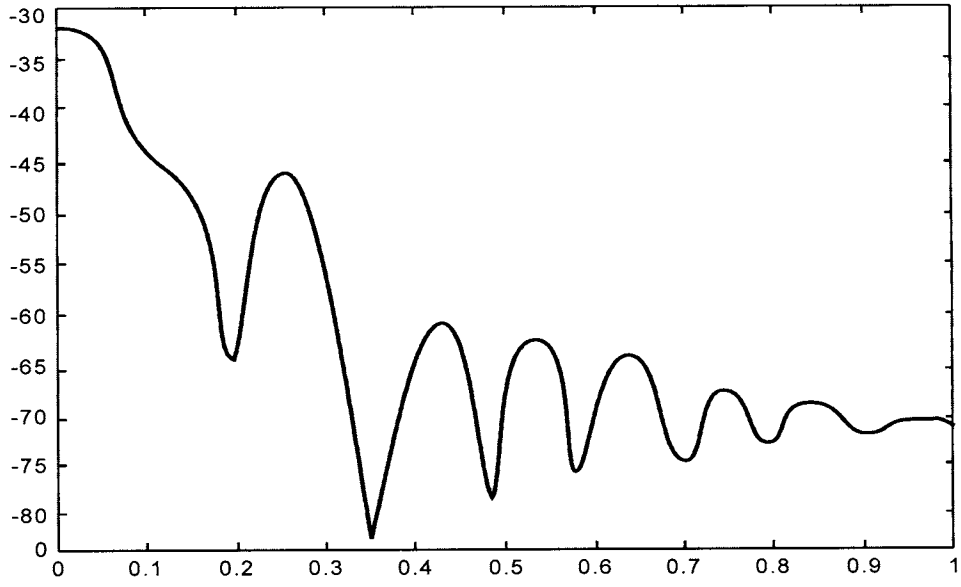

Figure 6.35
Filter response

We can see that in this case a more efficient filter (in terms of filter length) can be obtained by not including a sample at zero frequency. In general, for each design, we need to examine alternative design methods. By comparison, the most efficient windowing design is obtained through the Kaiser window with a minimum length of 38.

6.6 Parks-McClelland method

Early in the chapter we said that the first step in the filter design process is approximation. The frequency sampling design method discussed above is, strictly speaking, not an approximation approach but an interpolation approach. It produces a filter with frequency response that passes through the frequency sample points exactly but there is no constraint on the response between sample points. Consequently we cannot guarantee the behavior of the frequency response apart from that at the sample points. Peaks and overshoots can occur at various parts of the response. For low-pass filters, examples have shown that the transition bandwidth affects the resulting design to a large extent. By carefully optimizing the placement and values of the samples at the transition region, better designs are obtained. The question is how far can the maximum error be reduced?

The answer to this question lies in a technique that was used widely for analog filter approximation, known as Chebyshev approximation. This approximation, when applied to filter design, minimizes the maximum error over a set of frequencies. This type of filter exhibits equiripple behavior in the frequency responses. Thus the filter designed using this approximation are called equiripple FIR filters. They are also called optimum and minimax filters.

Closed form design formulas are not available for these filters, however. An iterative algorithm has to be used. A very efficient one is the Remez exchange algorithm. It was first developed in the early 1970s.

6.6.1 The approximation problem

A typical specification for a low-pass filter suitable for Chebyshev approximation is shown in Figure 6.36.

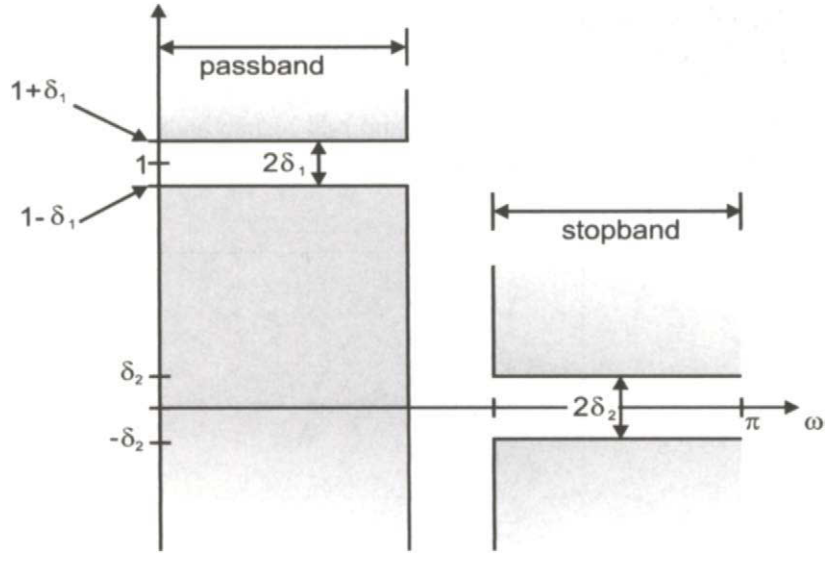

Figure 6.36
Typical low-pass filter specification for Chebyshev approximation

In the passband, the maximum deviation of the magnitude response from unity is $\pm\delta_1$. In the stopband, it is $\pm\delta_2$.

The desired frequency response $D(\omega)$ is assumed to be zero phase which means it is purely real-valued. The form of the frequency response of the final filter is

$$H(\omega) = Q(\omega)\sum_{k=0}^{N-1} h(k)\cos\left(\frac{2\pi k}{N}\right)$$

where

$$Q(\omega) = \begin{cases} 1, & \text{for Type 1 filters} \\ \cos(\omega/2), & \text{for Type 2 filters} \\ \sin\omega, & \text{for Type 3 filters} \\ \sin(\omega/2), & \text{for Type 4 filters} \end{cases}$$

The approximation problem is to minimize the maximum of the weighted error function

$$\left\|E(\omega)\right\| = \max W(\omega)\left|D(\omega) - H(\omega)\right|$$

for all $h(k)$ by choosing a suitable $H(\omega)$. Here Ω is the entire frequency range of interest, which is $[0,\pi]$. $W(\omega)$ is a user-defined weighting function so that more importance can be placed on certain frequency intervals compared with others. For instance, a zero weight can be assigned to the transition frequencies so that the shape of the response in this region will not affect the performance in the passband and stopband which are usually much more important.

6.6.2 The equiripple solution

Solution to the above approximation problem can be found by making use of a theorem, called the alternation theorem, from the theory of approximation. It basically states that the design is optimized for minimum ripple, if and only if, there are at least $N/2+2$ or $(N-1)/2+2$ extrema (for N even and odd respectively) of equal weighted amplitudes and alternating signs in the pass and stopbands. Such extrema are called alternations.

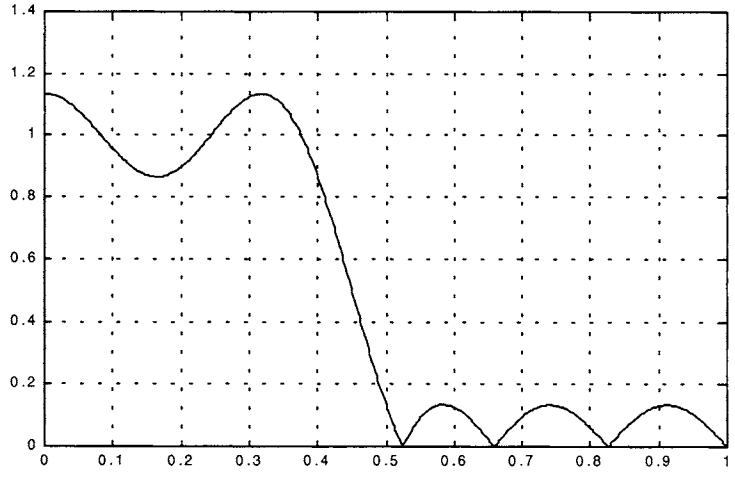

Figure 6.37
A length-13 equiripple FIR filter

Figure 6.37 shows a solution for a length-13 equiripple FIR filter. Since $N=13$, the number of extrema is 8. These eight extrema are also indicated in the figure.

It should be pointed out that the best equiripple design is also unique. For a given set of specifications, the unique best solution may have more than the minimum number of extrema as stated above. Let N be even and $r = (N/2+2)$. If the unique best filter has $r+2$ extrema, then there cannot be another filter with only $r+1$ or r extrema for the same set of specifications. In other words, using the Chebyshev approximation, optimality is guaranteed.

The alternation theorem is useful in that it helps us to establish the form of the optimal solution so that it can be recognized once we have found it. But it does not tell us how to arrive at the optimal solution. One approach is to identify the extremal frequencies. Once the extrema are found, the filter coefficients can be obtained by using the frequency sampling method. Thus the filter design problem becomes one of finding the extrema given a set of specifications. The Remez exchange algorithm is an efficient one for finding these extrema.

6.6.3 The Remez exchange algorithm

Given the order N of the filter, we do not know beforehand the minimum amount of ripples, δ_1 and δ_2, that can be achieved. So they become the additional variables, apart from the filter coefficients, that need to be determined. There are two main ways to handle this.

Parks and McClelland introduced a weight K to the stopband specification so that

$$K\delta_2 = \delta_1 = \delta$$

Hence instead of two variables, only δ will need to be determined together with the filter coefficients. They are evaluated iteratively for a given filter order. If the specifications are not met, then the filter order is increased and the optimization is repeated.

Another method, proposed by Hersey, Lewis and Tufts, is to let δ be equal to either δ_1 or δ_2. In this way the algorithm will still only need to deal with one additional variable but either the passband or stopband constraint will be satisfied exactly.

The Remez exchange algorithm makes use of the fact that the error function

$$E(\omega) = D(\omega) - \sum_{k=0}^{N-1} h(k)\cos k\omega$$

with $0 \le \omega \le \pi$ will always take on values of $\pm\delta$ for a given set of $(N+1)$ normalized frequency points denoted by ω_m for $m = 1,2,\ldots, N+1$. Therefore we have a set of linear equations

$$D(\omega_m) = \sum_{k=0}^{N-1} h(k)\cos k\omega_m + (-1)^m \delta \qquad m = 1,2,\ldots,N+1$$

There are $N+1$ equations with $N+1$ unknowns (N filter coefficients and the ripple amplitude) which we can solve. If the extremal frequency ω_m is known, then the equations can be easily obtained and no iteration is needed.

Obviously the algorithm cannot deal with a continuum of frequencies, even within the Nyquist interval. Parks and McClellan suggested the use of a set of frequencies, which are equally spaced, with a size of about 10 times the order of the filter. Since the number of frequencies is larger than N+1, we cannot directly solve the set of equations set out above. The Remez exchange algorithm starts with a trial set of frequencies and systematically exchange frequencies until the set of extremal frequencies is found.

Remez exchange algorithm:

- Choose an initial set of N+1 frequencies:

$$T^{(0)} = \left\{\omega_1^{(0)}, \omega_2^{(0)}, \ldots, \omega_{N+1}^{(0)}\right\}$$

- Solve the set of linear equations for $T^{(i)}$. The error function has a magnitude of $\delta^{(i)}$ for the i-th iteration.
- Find the frequency response at the whole set of frequencies.
- Search the entire set of frequencies to see where the magnitude of error is larger than that found in the second step. If none exists, then stop.
- Update the set of trial frequencies to be the $N+1$ frequencies where the errors are largest among the errors computed for the whole set of frequencies.

$$T^{(i+1)} = \left\{\omega_1^{(i+1)}, \omega_2^{(i+1)}, \ldots, \omega_{N+1}^{(i+1)}\right\}$$

- Repeat from the second step.

It should be pointed out that in the above discussion, we have assumed that the weight function $W(\omega)$ is unity for all frequencies. But the results apply to a general positive weight function.

6.6.4 Design formulas

For low-pass filters, Kaiser has developed some empirical formulas that helps in estimating the order of the filter required for a given set of specifications.

$$N = \frac{-10\log_{10}(\delta_1\delta_2)}{14.6\Delta f} + 1$$

where Δf is the normalized transition bandwidth given by

$$\Delta f = \frac{\omega_S - \omega_P}{2\pi}$$

and ω_p, ω_s are the passband and stopband edge frequencies respectively. This formula gives a good estimate when the bandwidth is neither extremely wide nor extremely narrow.

For filters with very narrow passbands, the stopband behavior governs the filter order and the following formula can be used for estimation:

$$N = \frac{0.22 - 20\log_{10}\delta_2 / 27}{\Delta f}$$

For very wide passband filters, such as notch filters, the following equation can be used instead:

$$N = \frac{0.22 - 20\log_{10}\delta_1 / 27}{\Delta f}$$

A more accurate estimate can be obtained by:

$$N = \frac{f(\delta_1, \delta_2) - g(\delta_1, \delta_2)(\Delta f)^2}{\Delta f}$$

where

$$f(\delta_1, \delta_2) = (0.005309x_1^2 + 0.07114x_1 - 0.4761)x_2 - (0.00266x_1^2 + 0.5941x_1 + 0.4278)$$

$$g(\delta_1, \delta_2) = 11.012 + 0.51244(x_1 - x_2)$$

and

$$x_1 = \log_{10}\delta_1$$

$$x_2 = \log_{10}\delta_2$$

Example 6.7

An FIR low-pass filter with the following specifications is designed using the Remez exchange algorithm:

Passband: $0 - 0.66\pi$

Stopband: $0.74\pi - \pi$

$\delta_1 = \delta_2 = 0.1$

Figures 6.38 and 6.39 show an odd length and even length filter response that satisfied these specifications.

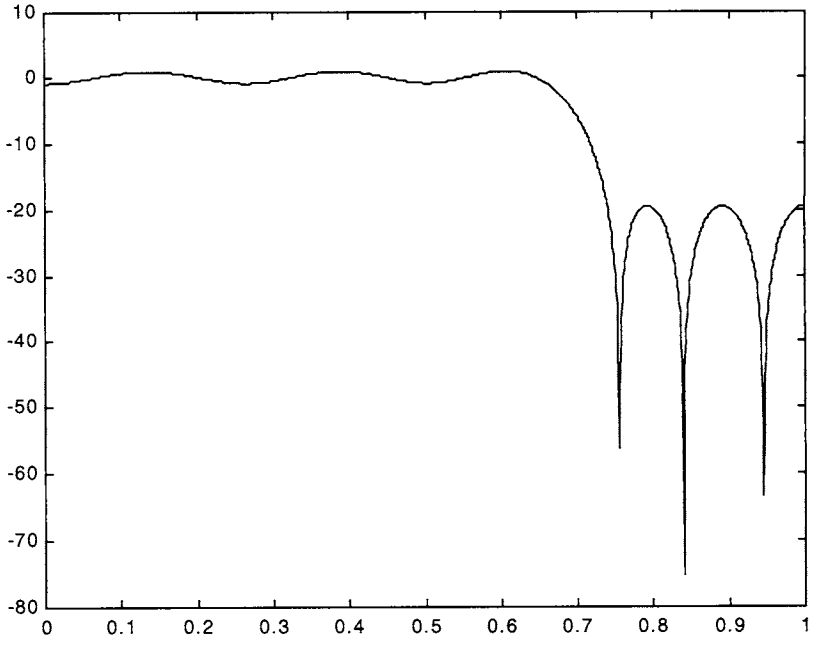

Figure 6.38
Odd length filter response

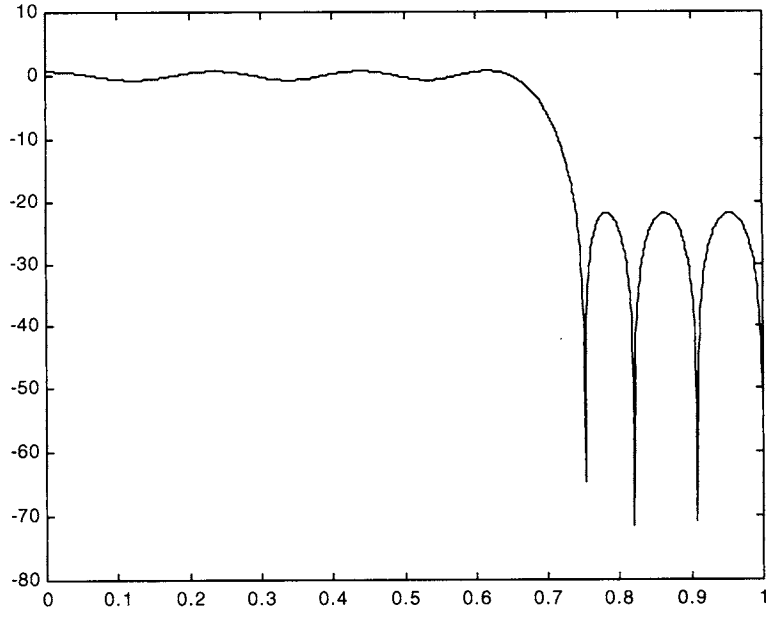

Figure 6.39
Even length filter response

For the length-21 filter, there are 12 extrema as expected, including the two band edges. Note that one of these two band edges will always be an extrema frequency, but not necessarily both. The frequency response is not forced to be zero at either $\omega = 0$ or $\omega = \pi$.

The even length filter has a slightly smaller resulting error than the length-21 filter. There are 11 extrema. Since this is a type 2 filter, $\omega = \pi$ is always zero.

Next we shall design two length-21 bandpass filters. The specifications are:

Passband: $0.36\pi - 0.66\pi$

Stopband: $0 - 0.28\pi$ and $0.74\pi - \pi$

The two transition bandwidths are identical and are the same as in the previous low-pass example. The magnitude response is shown in Figure 6.40.

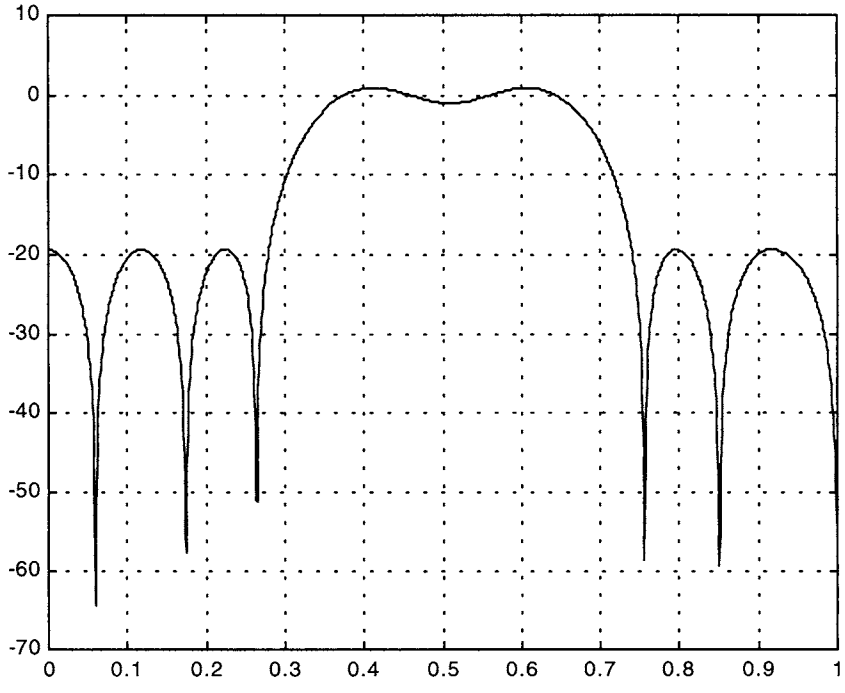

Figure 6.40
Magnitude response of length-21 bandpass filter

If the specifications are altered as below:

Passband: $0.5\pi - 0.74\pi$

Stopband: $0 - 0.16\pi$ and $0.8\pi - \pi$

The transition bandwidths are now unequal. The resulting length-21 filter has an error, which is only slightly larger than the previous equal transition bandwidth case. Examining the filter's magnitude response (Figure 6.41) indicates that it behaves well within the passband and stopbands. But the behavior in one of the transition bands is entirely unexpected. This behavior has been studied extensively.

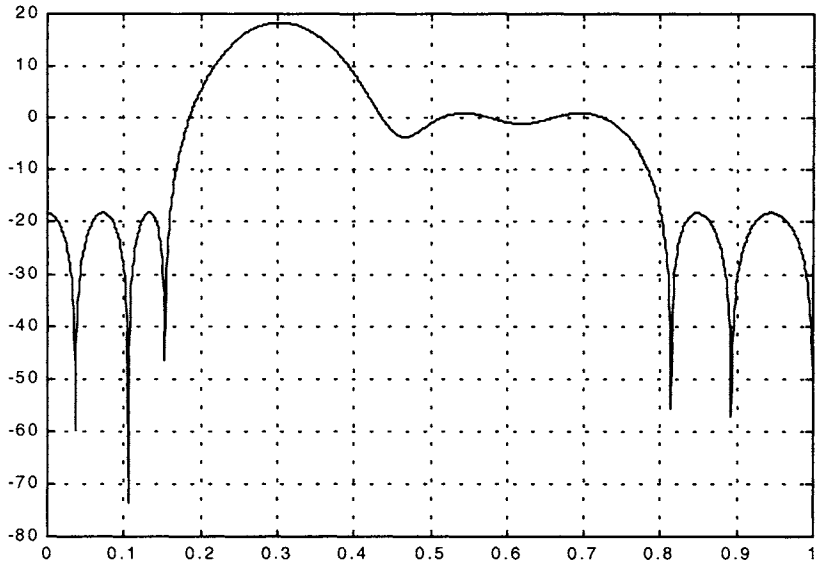

Figure 6.41
Length-21 FIR bandpass filter response

One way of reducing the possibility of transition band peaks is to calculate the following

$$N_1 = \frac{-10\log_{10}(\delta_1\delta_2) - 13}{14.6\Delta f_1} + 1$$

$$N_2 = \frac{-10\log_{10}(\delta_2\delta_3) - 13}{14.6\Delta f_2} + 1$$

where Δf_1 and Δf_2 are the two transition bandwidths. If $N_2 > N_1$, then Δf_1 can be reduced by moving the stopband edge frequency closer to the passband edge so that N_1 is approximately equal to N_2.

Alternatively, the weighing function can be used to control the maximum amount of error in the transition band. However, the appropriate amount of weighing is usually obtained by experience and trial-and-error.

A much better way to control the behavior in the transition band is to use the linear programming design method.

6.7 Linear programming method

One of the most recent approaches to linear phase FIR filter design makes use of the well-known linear programming method. In this case, the desired frequency response is composed of two parts: the upper limit function and the lower limit function. For a set of frequency points (approximately 10 times that of the filter order), the constraints are denoted as

$$H(\omega_k) + x \le U(\omega_k)$$

$$H(\omega_k) - x \ge L(\omega_k) \quad or \quad -H(\omega_k) + x \le -L(\omega_k)$$

where U and L are the upper and lower limit functions respectively. x is a parameter, which represents the distance between the upper and/or lower constraints. x can be zero if we allow the final response to 'hug' one of these two limit functions. Otherwise, the algorithm will maximize x.

With these constraints, we arrive at the linear programming problem:

$$\max \quad x$$

subject to

$$C^{\mathrm{T}}h + ax \leq b$$

The matrix C is determined from the sampled trigonometric functions. Vector h contains the filter coefficients. Vector b contains the limits (or bounds) and vector a is 1 where the parameter x is used and zero where it isn't. The variables h and x are unconstrained in sign. This is called the primal problem.

A form, which is more convenient for numerical solution, is the dual of the primal problem:

$$\min \quad b^{\mathrm{T}}y$$

subject to

$$Cy = 0, \quad a^{\mathrm{T}}y = 1, \quad \text{and} \quad y \geq 0$$

Using the well-known simplex algorithm can easily solve the dual problem. The algorithm will terminate under one of the following conditions:

- Negative cost is obtained which implies that the original design problem is feasible.
- Optimal solution is reached with non-negative cost, which means that the design problem has a feasible solution.
- The dual is unbounded which means that the primal problem is infeasible.
- Dual is infeasible which implies that the primal problem is infeasible or unbounded.

6.8 Design examples

A computer program called METEOR is publicly available which implements this approach. A low-pass filter is designed using METEOR and the result is shown in Figure 6.42. The solid lines are the constraints.

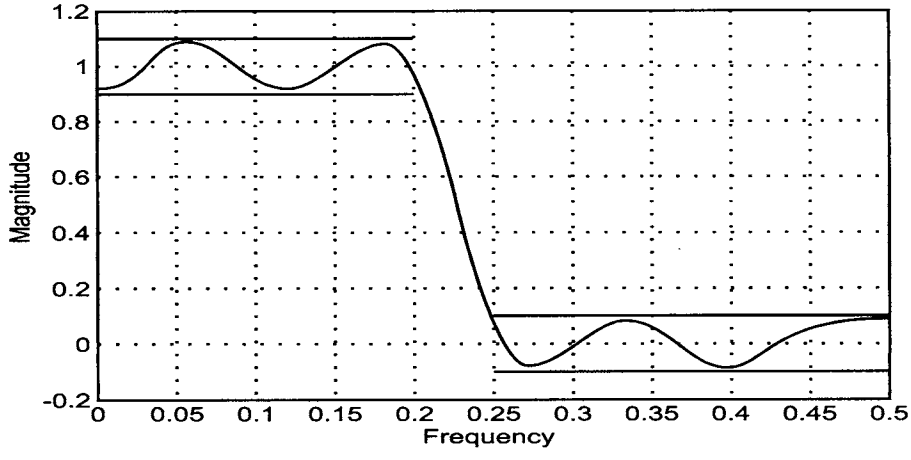

Figure 6.42
Low-pass filter designed using METEOR

Figure 6.43 shows the magnitude response of a length-25 bandpass filter with unequal transition bandwidths.

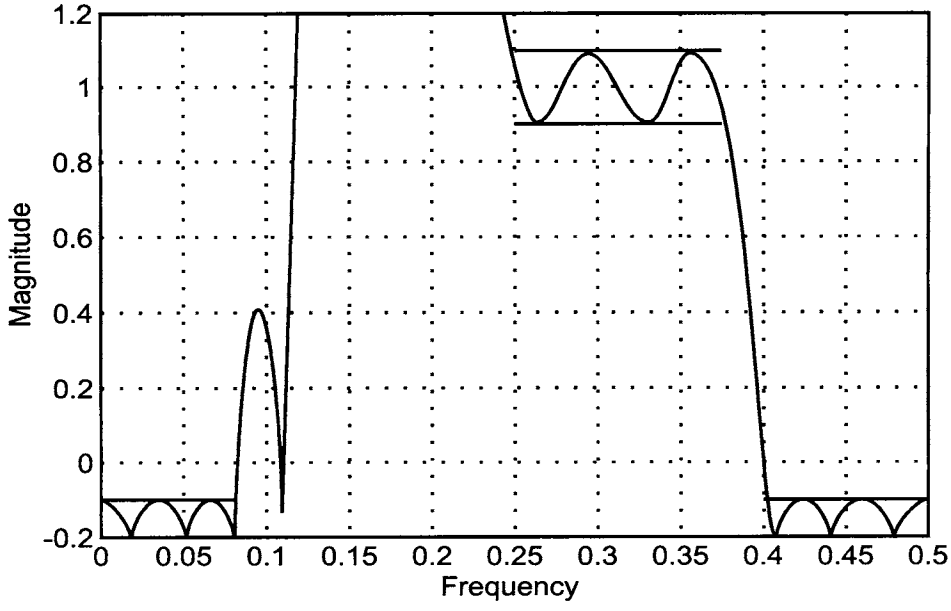

Figure 6.43
Length-25 bandpass filter designed using METEOR

This solution is essentially the same as that obtained using the Parks-McClelland method. The behavior in the first transition band is undesirable. This problem can be overcome by placing an upper limit on the first transition band. The resulting filter response is found in Figure 6.44.

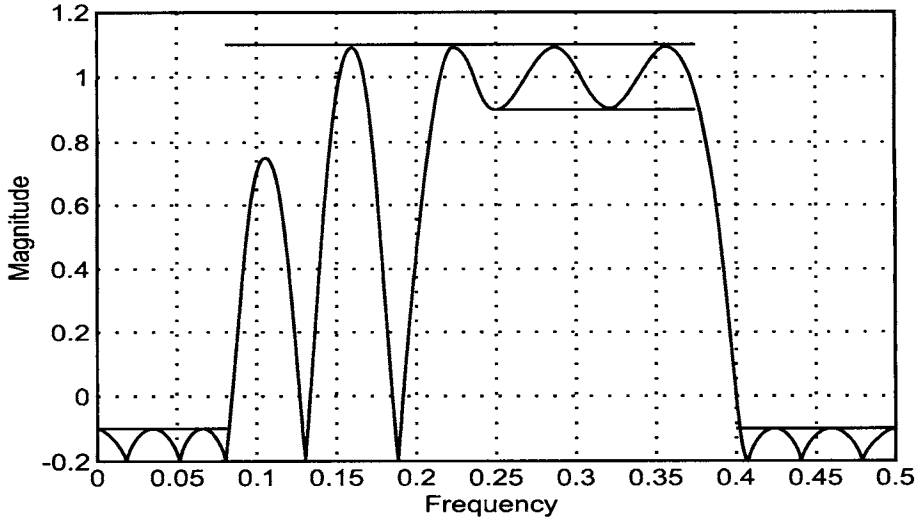

Figure 6.44
Limiting the overshoot in the first transition band

A better response can be obtained by the stopband very much similar to the solution proposed previously.

6.9 To probe further

We have only covered the design of linear phase FIR filters. Obviously, non-linear phase FIR filters can be designed as well. Basically, this means that the desired frequency response is no longer real-valued; it is complex-valued. The optimization will become more complex as a result. This has been the subject of much research in the 1980s and a number of good methods have been proposed. However, it is beyond the scope of this course to cover these topics. The interested reader should consult technical articles appearing in, for instance, IEEE transactions on signal processing in recent years.

Another very interesting topic for filter design is the design of filter banks. Filter banks have been found to be very useful in signal compression, particularly speech and image compression. A specific class of filter bank called perfect reconstruction filter banks (PRFB) guarantee that the signal reconstructed will be exactly the same as the one being decomposed. These filter banks are later found to be linked to the theory of wavelet transformation. Wavelet transformations are essentially Fourier transforms, which are better in capturing local behavior. Since Fourier transformation integrate (or sum) over the entire time axis (from negative infinity to infinity), all local (in time) information will be lost. It essentially 'averages' over all time. If local behavior becomes important, wavelet transforms can be used. This is still a very active area of research and technical articles appear with great regularity in most major technical journals.