Chapter 4

Design of FIR Filters

4.1 Introduction

Digital FIR filters have many favourable properties, which is why they are extremely popular in digital signal processing. One of these properties is that they may exhibit linear phase, which means that signals in the passband will suffer no dispersion. Dispersion occurs when different frequency components of a signal have a different delay through a system.

The simplest design method for FIR filters is *impulse response truncation* (IRT), but unfortunately it has undesirable frequency-domain characteristics, owing to the *Gibb's phenomenon*. The second design method for a FIR filter that we shall cover in this Chapter is the *windowing* technique. The windowing method can be used to mitigate the adverse effects of impulse response truncation.

4.2 Fourier Transform Relationship

The frequency response of a generalised FIR filter is defined by Equation 4.1 below.

$$H(z) = \sum_{k=-\infty}^{\infty} h[n] \cdot z^{-k} \tag{4.1}$$

To find the frequency response, z can be replaced by $e^{j\Omega}$, as follows:

$$z = e^{sT} \equiv e^{jwT} \equiv e^{j\Omega} \tag{4.2}$$

Equation 4.1 can be re-written as Equation 4.3:

$$H(\Omega) = \sum_{k=-\infty}^{\infty} h[n] \cdot e^{-jk\Omega} \tag{4.3}$$

The inverse Fourier transform of $H(\Omega)$ gives an expression for the impulse response of the FIR filter:

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\Omega) \cdot e^{jn\Omega} d\Omega \tag{4.4}$$

Equations (4.3) and (4.4) form a Fourier transform pair. There is a similarity between this pair and the pair for a periodic signal $x(t)$, as shown in equations (4.5):

$$x(t) = \sum_{k=-\infty}^{\infty} X(k) \cdot e^{j2\pi k f_0 t} \qquad X(kf_0) = \frac{1}{T} \int_0^T x(t) \cdot e^{-j2\pi k f_0 t} dt \tag{4.5}$$

Both of the functions $x(t)$ and $H(\Omega)$ are periodic. The Fourier transform of $x(t)$ gives the discrete function of kf_0, – in other words the discrete harmonics of the spectrum. The inverse Fourier transform of $H(\Omega)$ gives the discrete function of n – in other words the discrete samples of the impulse response of the filter.

4.2.1 Low-pass Filter design

Consider the *ideal* low-pass filter frequency response, as illustrated in Figure 4.1 below, with a normalised angular cut–off frequency Ω_c. Usually the subscript D is used to distinguish between the ideal and actual, impulse and frequency responses of a filter.

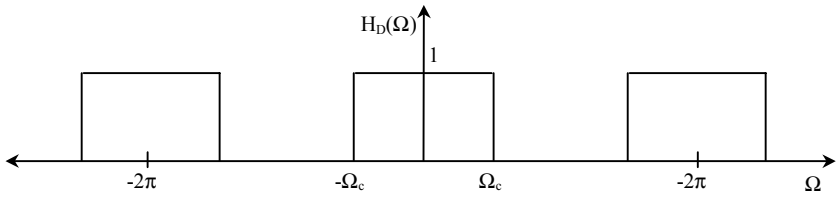

Figure 4.1: Ideal low-pass filter frequency response.

The impulse response of an ideal low-pass filter $h_D[n]$ is found by substituting $H_D(\Omega) = 1$ in Equation 4.5 and integrating between the limits of the cut-off frequencies $[-\Omega_c, \Omega_c]$.

$$h_D[n] = \frac{1}{2\pi}\int_{-\Omega_c}^{\Omega_c} 1 \cdot e^{jn\Omega} d\Omega = \frac{1}{2\pi}\left[\frac{e^{jn\Omega}}{jn}\right]\Bigg|_{-\Omega_c}^{\Omega_c} = \frac{1}{2\pi}\left[\frac{e^{jn\Omega_c}}{jn} - \frac{e^{-jn\Omega_c}}{jn}\right] = \frac{1}{2\pi}\cdot\frac{2j\sin(n\Omega_c)}{jn}$$

Multiplying both the numerator and denominator by Ω_c, the Equation above becomes:

$$h_D[n] = \frac{\Omega_c}{\pi}\cdot\frac{\sin(n\Omega_c)}{(n\Omega_c)}$$

The impulse response of an ideal low-pass filter can also be re-written by replacing $\Omega_c = 2\pi F_c$ in the Equation above, to obtain it in terms of the normalised cut-off frequency F_c. This is illustrated in Equation 4.6 below.

$$h_D[n] = 2F_c \cdot \frac{\sin(n\Omega_c)}{(n\Omega_c)} \tag{4.6}$$

In Chapter 2, we found that the Fourier transform of a rectangular window is a sinc function, which is same for the impulse response of a low-pass filter, as illustrated in Figure 4.2 below.

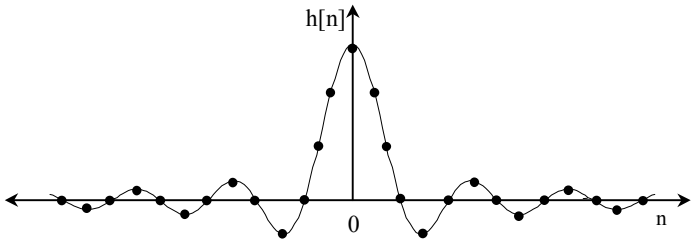

Figure 4.2: Impulse response of an ideal low-pass filter.

4.3 FIR Filter Design by Impulse Response Truncation (IRT)

With reference to Figure 4.2, although $h[n]$ decays to either side of $n = 0$ it theoretically continues for ever in both directions. This reflects a general antithesis between band limitation and time limitation; since we have chosen a frequency response with a sharp cut-off (or brick wall response), then the time-domain response continues forever. To realise such a filter the impulse response is *truncated* in some way or other. One approach is to ignore the small sample values at the ends and shift $h[n]$ to begin at $n = 0$, giving a *causal filter* as depicted in Figure 4.3.

In general, the more samples we include of $h[n]$ the closer we get to the desired form of $H_D(\Omega)$, but the less economic the filter becomes due to the relative number of computations. In practice we must settle for an approximation to the ideal frequency response. It is usually customary to truncate the impulse response to $N = (2M + 1)$ terms. In Figure 4.3, the impulse response of the ideal low-pass filter is truncated to $M = 9$ samples and is delayed by M samples.

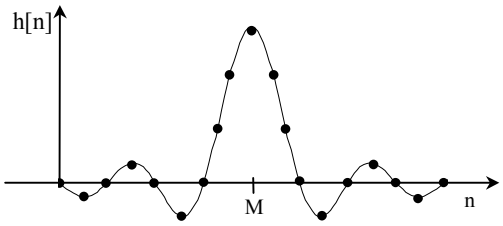

Figure 4.3: Truncated impulse response to 2M+1 samples, with a delay of M samples.

The z-transfer function of the filter now becomes:

$$H(z) = \sum_{n=-M}^{M} h[n] \cdot z^{-(n+M)} \tag{4.7}$$

The ideal impulse responses for a low-pass, high-pass, band-pass and band-stop filters are depicted in Table 4.1 below.

Filter type	$h_D[n]$, $n \neq 0$	$h_D[n]$, $n = 1$
Low-pass	$2F_c \frac{\sin(n\Omega_c)}{n\Omega_c}$	$2F_c$
High-pass	$1 - 2F_c \frac{\sin(n\Omega_c)}{n\Omega_c}$	$1 - 2F_c$
Band-pass	$2F_2 \frac{\sin(n\Omega_2)}{n\Omega_2} - 2F_1 \frac{\sin(n\Omega_1)}{n\Omega_1}$	$2F_2 - 2F_1$
Band-stop	$1 - \left[2F_2 \frac{\sin(n\Omega_1)}{n\Omega_1} - 2F_1 \frac{\sin(n\Omega_2)}{n\Omega_2} \right]$	$1 - [2F_2 - 2F_1]$

Table 4.1: Ideal impulse responses for various FIR filter types.

4.4 Summary of FIR Filter Design Using The IRT Method

- Choose the ideal frequency response $H_D(\Omega)$, depending on the type of filter (e.g. low-pass, high-pass etc), from Table 4.1.
- Calculate the impulse response of the ideal filter $h_D[n]$, using the inverse Fourier transform formula of Equation 4.5.
- Finally, truncate the ideal impulse response to N = (2M + 1) terms.

4.5 Optimality of the IRT Method

A measure of the error between the frequency response of the ideal filter $H_D(\Omega)$ and that of the designed filter $H(\Omega)$ is the integral of the squared error, defined by Equation 4.8.

$$E(\Omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| H_D(\Omega) - H(\Omega) \right|^2 d\Omega \tag{4.8}$$

Filters designed by the IRT method are considered optimal in the sense of minimising the integral of the squared error. However, the IRT approach is not considered practical because of the oscillatory nature of the frequency response near the discontinuity points of the desired response $H_D(\Omega)$. It is the case that as *N* increases, the magnitude of the oscillations remains the same, but they become more localised to the discontinuity, with the result that $E(\Omega)$ decreases with increasing *N*. This property is known as the *Gibb's phenomenon* and is described in detail in Section 4.6.

4.6 Gibb's Phenomenon

Truncating the impulse response introduces undesirable ripples and overshoots in the frequency response. This effect is known as the Gibb's phenomenon and is illustrated in Figure 4.4. As an example, the impulse response for a low-pass filter is truncated with M = 9, 25 and an infinite number of samples. Notice how the undesirable ripples and overshoots are clearly visible in the frequency response.

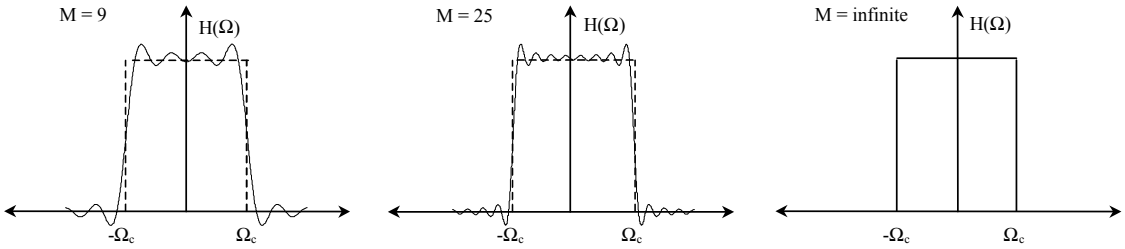

Figure 4.4: Effects on the frequency response of truncating the ideal impulse response.

In the time-domain truncation is achieved by multiplying the impulse response with a window function $w(n)$. Conversely, in the frequency-domain truncation is achieved by convolving the frequency response $H(\Omega)$ with $W(\Omega)$, where $W(\Omega)$ is the Fourier transform of the window function. In this example, the truncation was achieved by multiplying the impulse response with a rectangular window function. The height of the ripples is *not* dependent on the value of M, it only affects the width. The heights of the ripples are however controlled by the type of window function used for the time-domain multiplication. Table 4.2 illustrates the mathematical definitions, and Figure 4.5 shows the characteristic shape and amplitude responses of various window functions.

Name of window function $w(n)$	Mathematical definition
Rectangular	1
Hanning	$0.5 - 0.5\cos\left[\frac{2\pi n}{N-1}\right]$
Hamming	$0.54 - 0.46\cos\left[\frac{2\pi n}{N-1}\right]$
Blackman	$0.42 - 0.5\cos\left[\frac{2\pi n}{N-1}\right] + 0.08\cos\left[\frac{2\pi n}{N-1}\right]$
Kaiser	$\frac{I_0\left[\beta\sqrt{1-\left(\frac{\lvert 2n-N+1\rvert}{N-1}\right)^2}\right]}{-I_o(\beta)}$ Where, $I_0(x) = \sum_{k=0}^{\infty}\left(\frac{x^k}{2^k k!}\right)^2$

Table 4.2: Mathematical definitions for various window functions.

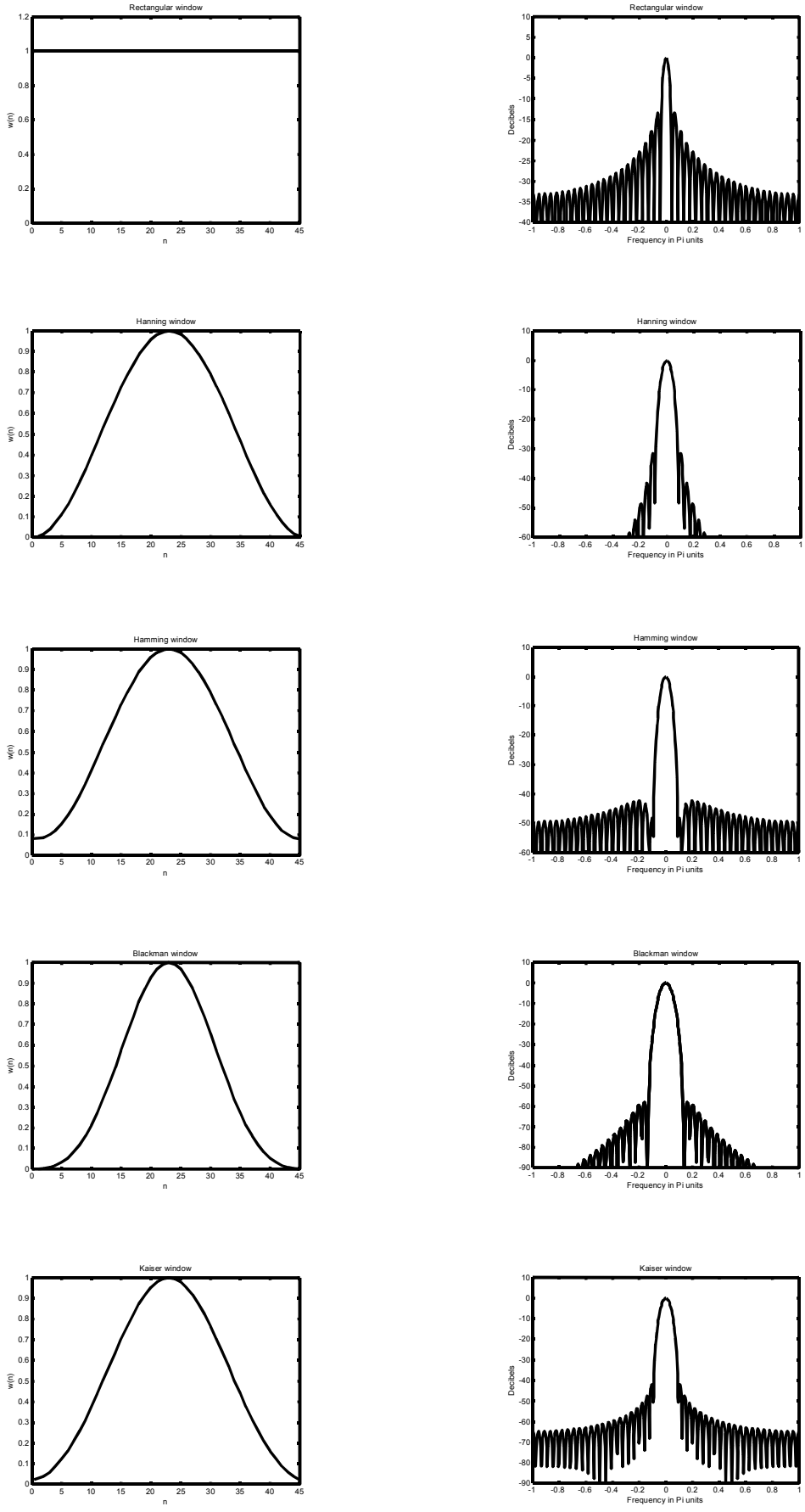

Figure 4.5: Window functions and their respective amplitude responses in dB.

4.7 Filter Specification Requirements

Figure 4.6 provides a graphical description of the specifications of a normalised low-pass filter. The shaded areas in the pass band and in the stop band indicate the forbidden magnitude values in these bands. In the transition band there are no forbidden values but it is usually required that the magnitude decreases monotonically in this band.

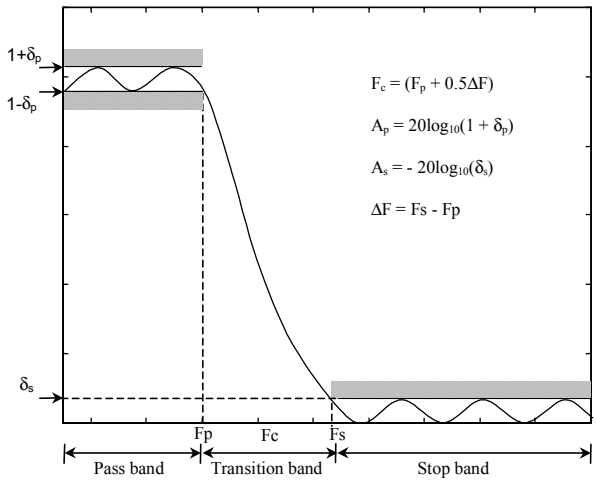

Figure 4.6: A normalised low-pass filter specification.

The parameter δ_s is the tolerance of the magnitude response in the stop-band and the desired magnitude response is always close to zero. The quantity δ_s is known as the *stop-band attenuation* and is sometimes expressed in terms of dBs using Equation 4.9.

$$\text{Minimum stop-band attenuation, } A_s = -20\log_{10}(\delta_s) \tag{4.9}$$

The parameter δ_p is the tolerance of the magnitude response in the pass-band and the desired magnitude response is close to unity. The quantity δ_p is called the *pass-band ripple* and can also be expressed in terms of dBs using Equation 4.10 below.

$$\text{Pass-band ripple, } A_p = 20\log_{10}(1 + \delta_p) \tag{4.10}$$

4.8 FIR Filter Design Using Windows

The IRT design method is only suitable for filters whose tolerances are about 21dB in the stop-band and 0.75dB in the pass-band. Practical filters are almost always required to have smaller tolerances, so the IRT method is impractical. This is due to the Gibb's oscillations, as seen in the amplitude response of Figure 4.4. However, in Chapter 2 windows were shown to reduce the side-lobe interference resulting from truncating an infinite-duration signal to a finite length. In a similar way, windows can also be applied to the impulse response of a FIR filter to attenuate the Gibb's oscillations.

When designing a FIR filter using the window method, it is customary to start by defining the ideal frequency response $H_D(\Omega)$ of the filter, as in the IRT method. Once the ideal frequency response has been obtained, it is then necessary to obtain the ideal impulse response $h_D[n]$ from the inverse Fourier transform of the frequency response (Equation 4.5). The next step in the design process is to select an appropriate window function based on the specification of the pass-band δ_p and stop-band δ_s tolerances of the filter. The tolerances of a FIR filter ultimately depend upon the type of window function used for the windowing but in the design process we always start by assuming that $\delta_s \approx \delta_p$. In Table 4.3 below, it illustrates various window functions and their corresponding properties to enable in the design of a filter. Once the appropriate window function has been determined by the designer based on the filter specification, then the required number of filter coefficients N can then be calculated. The final procedure of the design process is to determine the actual filter coefficients $h[n]$ by multiplying the coefficients of the window function with the corresponding ideal impulse response. This is mathematically shown in Equation 4.12 below:

$$h[n] = w[n] \cdot h_D[n] \tag{4.12}$$

Name of window function $w[n]$	Transition width ΔF in (Hz), (normalised)	Pass-band ripple A_p in (dB)	Ripple δ_p, δ_s	Side-lobe level in (dB)	Stop-band attenuation A_s in (dB)
Rectangular	0.9/N	0.741	0.089	-13	21
Hanning	3.1/N	0.0546	0.063	-31	44
Hamming	3.3/N	0.0194	0.0022	-41	53
Blackman	5.5/N	0.0017	0.000196	-57	74
Kaiser β=4.54	2.93/N	0.0274			50
β=5.65	3.63/N	0.00867			60
β=6.76	4.32/N	0.00275			70
β=8.96	5.71/N	0.000275			90

Table 4.3: Window functions and their corresponding properties.

4.9 Summary of FIR Filter Design Using The Window Method

- Specify the ideal or desired frequency response of the filter $H_D(\Omega)$.
- Obtain the impulse response $h_D[n]$ by evaluating the inverse Fourier transform.
- Select an appropriate window function $w[n]$, that satisfies pass-band and attenuation specifications, and determine the number of coefficients required from the relationship between N and ΔF.
- Determine the values of the window function and calculate the *actual* FIR filter coefficients by multiplying the impulse response with the window function.

4.10 Linear Phase Response

The frequency response of the causal FIR filter given in equation (4.7) is:

$$H(j\omega) = \sum_{n=-M}^{M} h[n] \cdot e^{-j\omega T(n+M)} = e^{-j\omega TM} \sum_{n=-M}^{M} h[n] \cdot e^{-j\omega Tn} \tag{4.13}$$

where T is the sampling period. If the coefficients $h[n]$ are symmetrical around h=0, this can be written:

$$\begin{aligned} H(j\omega) &= e^{-j\omega TM} \left\{ h(0) + \sum_{n=1}^{M} h[n] \cdot \left(e^{j\omega Tn} + e^{-j\omega Tn} \right) \right\} \\ &= e^{-j\omega TM} \left\{ h(0) + \sum_{n=1}^{M} h[n] \cdot 2\cos(\omega Tn) \right\} \end{aligned} \tag{4.14}$$

The phase of the filter is given by the argument of the complex function in (4.14). Since the term in the curly brackets is purely real it therefore contributes nothing to the phase response. The term $e^{-j\omega TM}$ on the other hand has a phase of $-\omega TM$ *rad*, which is clearly linear in ω.

We can therefore conclude that filters with symmetrical impulse response (i.e. symmetrical coefficients) have linear phase; those with asymmetrical impulse response do not.

4.11 Example of Window Design

A FIR low-pass filter is required to have the following specifications:

1. Pass-band edge frequency f_p = 2 kHz
2. Transition band Δf = 200 Hz
3. Pass-band ripple A_p = 0.1dB

4. Minimum stop-band attenuation A_s = 50dB

5. Sampling frequency of f_s = 10 kHz

Using the window method, determine an appropriate window function and calculate the required number of filter coefficients to design this filter. Furthermore, ascertain the corresponding filter coefficient values $h[n]$ for $-10 \le n \le 10$.

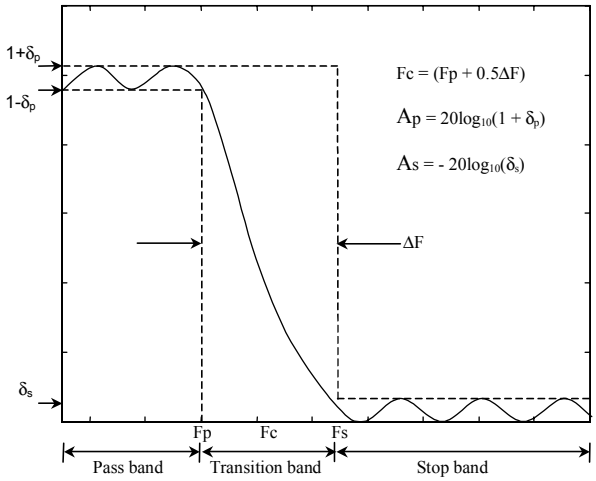

Figure 4.7: A normalised low-pass filter specification.

Pass-band ripple, $A_p = 20\log_{10}(1 + \delta_p)$

$$\delta_p = \log_{10}^{-1}\left[\frac{0.1}{20}\right] - 1 = 0.0116$$

Minimum stop-band attenuation $A_s = -20\log_{10}(\delta_s)$

$$\delta_s = \log_{10}^{-1}\left[\frac{-50}{20}\right] = 0.00316$$

The normalised pass-band edge frequency

$$F_p = f_p / f_s = \frac{2x10^3}{10x10^3} = 0.2$$

The normalised transition width

$$\Delta F = \Delta f / f_s = \frac{200}{10x10^3} = 0.02$$

First of all we need to choose an appropriate window function from Table 4.3, which has the narrowest ΔF but meets the δ_s specification of 0.00316. With reference to Table 4.3, the closest specification is the Hamming window with a δ_s = 0.0022. From the Table, the normalised transition width ΔF = 3.3/N, where *N* is the number of filter coefficient. By re-arranging this Equation to make *N* the subject and inserting the value of the normalised transition width, then the required number of filter coefficients can be calculated. For FIR filters the number of coefficients must be odd, so don't forget to round up to the nearest odd number.

$$N = \frac{3.3}{0.02} = 165$$

The mathematical definition of the hamming window to achieve the desired filter specification is defined from Table 4.2.

$$w[n] = 0.54 - 0.46\cos\left[\frac{2\pi n}{164}\right]$$

From Table 4.1, the ideal impulse response of a low-pass filter is defined by the following equation.

$$h_D[n] = 2F_c \cdot \frac{\sin(n\Omega_c)}{n\Omega_c} = \frac{\sin(n\Omega_c)}{n\pi}$$

Where the normalised angular cut-off frequency Ω_c is defined by the equation below:

$$\Omega_c = 2\pi(F_P + 0.5 \cdot \Delta F) = 2\pi(0.2 + 0.01) = 2\pi(0.21)$$

Hence, the ideal impulse response of the low-pass filter can be re-written as follows:

$$h_D[n] = \frac{\sin(2\pi \cdot 0.21 \cdot n)}{\pi n}$$

In the time-domain, truncation is achieved by multiplying the impulse response with the window function $w[n]$. Hence the filter coefficients $h[n]$ = -10, 10 can be obtained by:

$$h[10] = \frac{\sin(2\pi \cdot 0.21 \cdot 10)}{\pi 10} \times \left[0.54 - 0.46\cos\left[\frac{2\pi 10}{164}\right]\right] = 0.0187 \times 0.1133 = 0.0021$$

Since the coefficients of a FIR filter are symmetric, by definition h[-10] = h[10].

4.12 Optimal FIR Filter Design

An alternative method of FIR filter design is to allow an unconstrained optimisation of the filter coefficients to minimise a cost function. The cost function generally used is:

$$E(\Omega) = W(\Omega)[H_D(\Omega) - H(\Omega)] \tag{4.15}$$

where $H_D(\Omega)$ and $H(\Omega)$ are the ideal and actual frequency response functions respectively, and $W(\Omega)$ is a function to weight different parts of the band differently. One approach is to then determine the $h(n)$ such that the maximum weighted value of $E(\Omega)$ is minimised. Thus determine:

$$\min_{h[n]}\left[\max_{\Omega}|E(\Omega)|\right]$$ over the passbands and stopbands.

For the optimum design filter method it is necessary to first estimate the number of coefficients required. The starting point is the empirical formula given in Equation (4.16).

$$\hat{N} = \frac{2}{3}\frac{1}{\Delta F}\log_{10}\left[\frac{1}{10\delta_p\delta_s}\right] \tag{4.16}$$

Having calculated this estimate, the values of the coefficients are then determined using a suitable DSP simulation package (e.g. *REMEZ* in Matlab). Using these coefficient values, the frequency response is compared with that required; if it is outside the original specification then the value of N is increased, and the design process repeated.

The advantage of this technique over the window-based method is that a smaller value can be specified for δ_s than for δ_p, and this can lead to a significant saving in the number of coefficients required, as illustrated in the example below.

Example:

The following shows a particular example:

Passband	8-12 kHz
Stopband ripple	0.001
Peak passband ripple	0.01

Sampling frequency	44.14 kHz
Transition width	3 kHz

For the *window* method, the following parameters apply for the *Kaiser* window:

Required passband ripple	$20\log(1+0.01)=0.199$ dB
Required stopband attenuation	$-20\log(0.001)=60$ dB
Cutoff frequencies	6.5 kHz, 13.5 kHz
Ripple parameter, β	5.653
Number of filter coefficients, N	3.63/(3/44.14) = 53.4 (rounds up to 55)
Sampling frequency	44.14 kHz

For the *optimal method,* the *remez* function in Matlab would require the following vectors as input:

F=[0, 5/22.07, 8/22.07, 12/22.07, 15/22.07, 1]
A=[0, 0, 1, 1, 0, 0]
W=[(1/0.001), (1/0.01), (1/0.001)]

$$\hat{N}=\frac{2}{3}\frac{1}{\Delta F}\log_{10}\left[\frac{1}{10\delta_p\delta_s}\right]=\frac{2}{3}\frac{1}{(3/44.14)}\log_{10}\left[\frac{1}{10\times 0.01\times 0.001}\right]=39$$

The filter length of 39 is significantly less than that required for the *Kaiser window* method.

N.B. In Matlab the functions *FIR1* and *REMEZ* use frequency normalised to $f_s/2$. Hence the frequencies required for *REMEZ* in Matlab will be 0, 5/(44.14/2), 8/(44.14/2), etc. However, just to confuse things, the Matlab function *FREQZ* returns the frequency vector *W* as an angular frequency normalised to $f_s/2$. Hence, a value of W=1 corresponds to a frequency of ($f_s/2\pi$) Hz.

It is also possible to design the above two filters using the *sptool* in Matlab. This provides an interactive method of adjusting the filter characteristics.

4.13 Finite Word Length Effects in FIR Filters

So far we have assumed that all variables can be represented exactly in the computer, and all arithmetic operations can be performed to an infinite precision. However, in practice numbers can only be represented to a finite precision, and arithmetic operations are subject to errors, since a computer word has a finite number of bits. These problems are more commonly referred to as *finite word length effects* and it can affect a digital filter in several respects by causing the filter coefficients to deviate from their ideal values. As a result the frequency response is altered and the filter may fail to meet the required specification.

In general there are four sources of noise present due to: 1) *A/D conversion*, 2) *coefficient quantisation*, 3) *arithmetic round off*, and 4) *arithmetic overflow*. The first type of noise, A/D conversion, gives rise to quantisation errors that reduce the SNR of the output. However, by increasing the number of bits used for the A/D stage, or by using multirate techniques (Chapter 8) can alleviate this problem.

For the second type of noise, coefficient quantisation, it changes the parameters of the transfer function H(z) and consequently the frequency response of the filter changes. This problem can be mitigated by using a sufficient number of bits for the fixed-point arithmetic. Or alternatively, the use of floating point arithmetic will usually bypass the problem of coefficient quantisation.

In fixed-point arithmetic, when two numbers each of wordlength N are multiplied together they produce a result of wordlength $2N$. To avoid wordlengths becoming gradually bigger and bigger, it is common to round the result to wordlength N by dropping the least significant half of the word. In two's complement, this arithmetic round off leads to an error in the final result so it is like adding quantisation noise to the output (i.e. it reduces the SNR). This problem can be avoided by either utilising double word length to store the intermediate results, or by optimising the structure of the filter.

The fourth type of noise, arithmetic overflow, occurs when a digital signal exceeds its expected value. An overflow in two's complement arithmetic leads to a polarity reversal, in that a number slightly larger than 1 changes to a number slightly larger than –1. Overflows in digital filters are potentially harmful because they give an incorrect output and care is necessary to prevent them from occurring. One such method is to scale the filter coefficients to ensue that overflow does not occur. Another method is to detect when an overflow has occurred and to use the maximum value instead. Most DSP microprocessors have a saturation mode, in which this operation is performed automatically after the

addition. A summary of these four sources of noise and their corresponding affects on the filter performance are shown in Table 4.4 below.

Source of noise	Affect on performance	Reduction techniques
A/D conversion.	Quantisation noise limits the signal-noise-ratio (SNR).	• Increase number of bits. • Use multirate techniques.
Coefficient quantisation.	Modifies desired frequency response e.g. limits max attenuation.	• Use sufficient Nos. of bits in fixed-point representation. • Optimise selection of filter coefficients. • Use floating-point arithmetic.
Arithmetic round off.	Reduces SNR.	• Use double word length for intermediate results. • Optimise filter structure.
Arithmetic overflow.	Incorrect output signal.	• Scale filter coefficients (at cost of reduced SNR). • Detect and use "maximum" rather than "overflowed" value. • Use floating-point arithmetic.

Table 4.4: Finite word length effects in FIR filters.