7. Design of FIR Filters

7.1 Introduction

- As noted above, FIR filters cannot be designed using the same techniques as IIR filters.
- This is because there is no analogue "equivalent" to FIR filters
- Special techniques have been developed for FIR filter design:
  - Window method
  - Optimisation Methods
  - Frequency-Sampling Technique

7.2 Linear Phase

- Generally, the main reason for selecting an FIR filter is that the application requires linear phase.
- Not all FIR filters have linear phase – in fact, in order to exhibit linear phase, an FIR filter must have a symmetric impulse response.
- Since the filter coefficients are the same as the samples of the impulse response, this means that the filter coefficients are symmetric.
- Four types of symmetry:
  - Odd-order filter with positive symmetry
  - Even-order filter with positive symmetry

- Odd-order filter with negative symmetry
- Even-order filter with negative symmetry

- Positive symmetry in the impulse response means that:

$$h(n) = \begin{cases} h(N-n-1) & \text{if } 0 \le \mathrm{n} \le \mathrm{N}-1 \\ 0 & \text{otherwise} \end{cases}$$

- For a FIR filter with negative symmetry:

$$h(n) = \begin{cases} -h(N-n-1) & \text{if } 0 \le \mathrm{n} \le \mathrm{N}-1 \\ 0 & \text{otherwise} \end{cases}$$

7.3 FIR Filter Design using Window Method

- The easiest way to design an FIR filter is to truncate the impulse response of an IIR filter.
- This is the basis of the Window Method

$$h_{FIR}(n) = h_{IIR}(n)w(n)$$

- where $h_{IIR}(n)$ is the impulse response of an IIR filter, $w(n)$ is a suitably-chosen window (finite length), and $h_{FIR}(n)$ is the resulting FIR filter impulse response.
- In Section 5, we saw that windowing in the time domain corresponds to convolution in the frequency domain, i.e.:

$$x_w(n) = x(n)w(n)$$

$$X_w(\theta) = X(\theta) * W(\theta)$$

- We have a "smearing" of the frequency response of the IIR filter because of the windowing operation.
- The same principles apply to the choice of window for FIR filter design, as apply to signal analysis – we would like the window function to be as close as possible to an "impulse" in the frequency domain.
- Selection of the window length is a trade-off between the smearing caused by small values of N, versus the additional complexity caused by a larger number of coefficients.
- The Window Method can be summarised as follows:

  - Determine a suitable IIR filter impulse response
  - Choose a value for N, and a suitable window type
  - Apply the window symmetrically to the IIR filter impulse response

- Suppose we want to design an FIR low-pass filter, with linear phase of slope $-\alpha$, and cut off frequency θ_c. Choosing a slope of $-\alpha$ for the phase response means that the FIR filter has an inherent delay of α samples.
- The frequency response of this filter can be described as follows:

$$H(\theta) = \begin{cases} e^{-j\alpha\theta}, & |\theta| \le \theta_c \\ 0, & \theta_c \le |\theta| < \pi \end{cases}$$

- The corresponding (infinite length) impulse response can be obtained by taking the inverse Fourier Transform, and can be shown to be given by:

$$h(n) = \frac{\sin[\theta_c(n-\alpha)]}{\pi(n-\alpha)}$$

- This is the familiar sinc function, centred around $n = \alpha$, and extending to $+\infty$ and $-\infty$. We can obtain a causal FIR filter by multiplying the infinite length sinc function by a window of length N, starting at n=0.
- For the FIR filter to have linear phase, we require that the coefficients be symmetric, so α must be chosen carefully. In particular, we need to have $\alpha = (N\text{-}1)/2$.

Exercise 7.1

Determine the impulse response of an IIR low pass filter with frequency response:

$$H(\theta) = \begin{cases} 1, & |\theta| \leq \dfrac{\pi}{3} \\ 0, & \dfrac{\pi}{3} \leq |\theta| < \pi \end{cases}$$

Hence, design a FIR filter of length N=9.

Exercise 7.2

Design a linear phase FIR high pass filter with cut off frequency equal to $\pi/2$.

- The Matlab function `fir1` can be used to design FIR filters using the Window Method. Various options are available for the design of low pass, high pass, band pass etc.
- As noted above, the choice of window determines the behaviour of the filter in the frequency domain. For a given length of filter, different windows give different amounts of ripple.

Exercise 7.3

Using `fir1`, examine the effect of different windows on the design of the high pass filter in Exercise 7.2.

7.4 Optimisation Methods

- In essence, what we are trying to do in FIR filter design is to approximate a desired frequency response by a "practical" response; this gives rise to an approximation error.
- In the case of the window method, the ripple tends to be concentrated near the band edges (cutoff frequency) – this is often not the best situation.
- It would be better if we could "distribute" the ripple (error) a bit more uniformly across the entire frequency axis.
- The basis of optimisation methods is the "optimal" distribution of the ripple between pass band and stop band in order to achieve the desired magnitude response as closely as possible.

- The most commonly used technique in this class is the *Remez Exchange Algorithm* or Parks-McClellan Optimisation Technique.
- We will not examine the method in detail, since it is widely implemented in the form of computer software (e.g. Matlab function `firpm` (previously known as `remez`)).
- There is some flexibility in the application of the method, e.g. the user can specify a "weighting" function for the error so that some bands in the frequency domain are favoured more than others.

Exercise 7.4

Using Matlab, compare the Window Method and the Remez Exchange Algorithm for the design of an FIR low pass filter.

7.5 Frequency Sampling Design of FIR Filters

- This method allows us to design FIR filters with arbitrary frequency response.
- It also allows us to design filters with *recursive* implementations, thus leading to simpler implementation than with the standard transversal filter architecture. In this case, the filter is implemented as a combination of FIR and IIR filters; in fact, this is one situation where a recursive structure (normally

associated with IIR filters) can be used to give a finite impulse response.

- The filter is designed by sampling the desired frequency response of the filter at N equally spaced points.
- These points can then be processed by an inverse DFT to obtain an N-point impulse response for an FIR filter.
- An important assumption is that the N points chosen in the frequency domain are an adequate representation of the frequency response.
- The basis for the method is as follows. Firstly, the desired frequency response is sampled at N points to give:

$$H(k) = H_{desired}\left(e^{j\frac{2\pi k}{N}}\right) \text{for } k = 0 \ldots N-1$$

- Then, the transfer function of the filter is obtained "indirectly" by using the inverse Discrete Fourier Transform to obtain $h(n)$, then using this to obtain $H(z)$:

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k) e^{jn\frac{2\pi k}{N}}$$

and

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

$$\therefore H(z) = \sum_{n=0}^{N-1} \frac{1}{N} \sum_{k=0}^{N-1} H(k) e^{jn\frac{2\pi k}{N}} z^{-n} = \sum_{k=0}^{N-1} \frac{1}{N} H(k) \sum_{n=0}^{N-1} e^{jn\frac{2\pi k}{N}} z^{-n}$$

$$H(z) = \sum_{k=0}^{N-1} \frac{1}{N} H(k) \sum_{n=0}^{N-1} \left(e^{j\frac{2\pi k}{N}} z^{-1} \right)^{n} = \sum_{k=0}^{N-1} \frac{1}{N} H(k) \frac{1-\left(e^{j\frac{2\pi k}{N}} z^{-1} \right)^{N}}{1-\left(e^{j\frac{2\pi k}{N}} z^{-1} \right)}$$

$$H(z) = \sum_{k=0}^{N-1} \frac{1}{N} H(k) \frac{1-\left(e^{j\frac{2\pi kN}{N}} z^{-N} \right)}{1-\left(e^{j\frac{2\pi k}{N}} z^{-1} \right)} = \sum_{k=0}^{N-1} \frac{1}{N} H(k) \frac{1-z^{-N}}{1-\left(e^{j\frac{2\pi k}{N}} z^{-1} \right)}$$

$$\therefore H(z) = \frac{1 - z^{N}}{N} \sum_{k=0}^{N-1} \frac{H(k)}{1 - e^{j\frac{2\pi k}{N}} z^{-1}}$$

- This transfer function consists of two terms:
  - An FIR term 1-z^{-N}, which introduces N zeros on the unit circle – such a filter is sometimes referred to as a "comb" filter (because of the shape of its magnitude response)
  - A sum of N first-order IIR terms, each of which has a pole located on the unit circle and is therefore marginally stable.
- However, the N zeros located on the unit circle cancels out the poles located on the unit circle, and hence the overall filter is stable.

Exercise 7.5

Show how an FIR filter whose impulse response consists of N samples each of amplitude g, can be implemented efficiently by a cascade of a comb filter and a resonator.

- As it stands, the implementation is as follows:

$$H(z) = \frac{1 - z^{N}}{N} \sum_{k=0}^{N-1} \frac{H(k)}{1 - e^{j\frac{2\pi k}{N}} z^{-1}}$$

- This requires complex coefficients (in the pole terms). However, it turns out that complex conjugate pole terms can be combined in pairs, each of which will result in a term which contains only real coefficients (except for the k = 0 term which is real anyway). Hence:

$$H(z)=\frac{1-z^{N}}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1-e^{j\frac{2\pi k}{N}}z^{-1}}=\frac{1-z^{N}}{N}\left(\frac{H(0)}{1-z^{-1}}+\sum_{k=1}^{\frac{N-1}{2}}\left(\frac{H(k)}{1-e^{j\frac{2\pi k}{N}}z^{-1}}+\frac{H^{*}(k)}{1-e^{-j\frac{2\pi k}{N}}z^{-1}}\right)\right)$$

$$H(z)=\frac{1-z^{N}}{N}\left(\frac{H(0)}{1-z^{-1}}+\sum_{k=1}^{\frac{N-1}{2}}\left(\frac{(H(k)+H^{*}(k))-z^{-1}\left(H(k)e^{-j\frac{2\pi k}{N}}+H^{*}(k)e^{j\frac{2\pi k}{N}}\right)}{1-2Cos\left(\frac{2\pi k}{N}\right)z^{-1}+z^{-2}}\right)\right)$$

$$H(z) = \frac{1 - z^N}{N}\left(\frac{H(0)}{1 - z^{-1}} + \sum_{k=1}^{\frac{N-1}{2}}\left(\frac{2\,\mathrm{Re}(H(k)) - 2\,\mathrm{Re}\left(H(k)e^{-j\frac{2\pi k}{N}}\right)z^{-1}}{1 - 2Cos\left(\frac{2\pi k}{N}\right)z^{-1} + z^{-2}}\right)\right)$$

- The accuracy of the frequency response depends on how many points are taken along the desired frequency response (i.e. N) – the more points that are taken; intuitively, the more accurately the frequency response is represented.
- However, since the number of frequency points equals the number of filter coefficients, there is a trade-off between accuracy and computational requirements.