DSP Semester Report

WORD COUNT: 1172

CHOOSEN TOPIC:

Talk about filter specifications and how to design a digital filter based on the given specifications

Submitted by:

Khawaja Arsalan Habib

Student ID: 201414020101

PHD Student

Submitted to:

Lin Jingran

Filter Requirements and Specification

Filter design is the process of creating the filter coefficients to meet specific filtering requirements. A digital filter is a basic building block in any Digital Signal Processing (DSP) system. The frequency response of the filter depends on the value of its coefficients, or taps. Filter implementation involves choosing and applying a particular filter structure to those coefficients. The goal of filter design is to perform frequency dependent alteration of a data sequence. A possible requirement might be to remove noise above 30 Hz from a data sequence sampled at 100 Hz.

Introduction to FIR Filters

As this report has to be concise and brief, the next sections will only focus of design of FIR filter and related examples. FIR filters are digital filters with finite impulse response. This can be stated mathematically as:

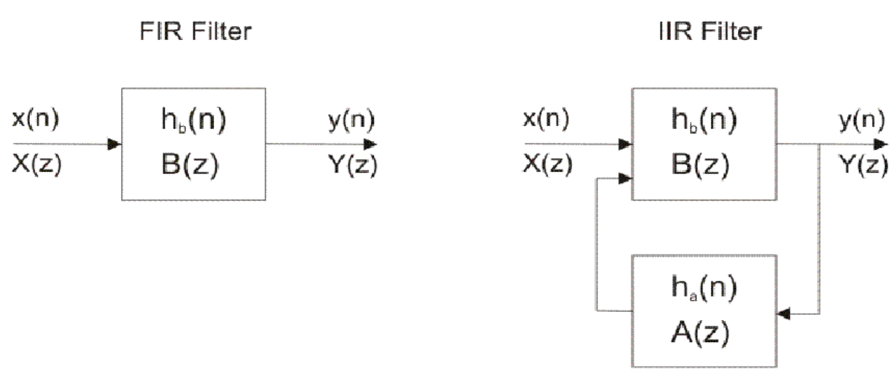
$$h(n) = \begin{array}{l} \{0, n \leq \tau_1 \quad -\infty < \tau_1 < \tau_2 < +\infty \\ \{0, n \geq \tau_2 \end{array}$$

where h(n) denotes the impulse response of the digital filter, nis the discrete time index, and τ1 and τ2 are constants. A difference equation is the discrete time equivalent of a continuous time differential equation. The general difference equation for a FIR digital filter is:

$$y(n) = \Sigma \mathrm{b}_\mathrm{k}\, \mathrm{x}(n-k)$$

where y(n) is the filter output at discrete time instance n, b_k is the k-th feed forward tap, or filter coefficient, and x(nk) is the filter input delayed by k samples. This denotes summation from k= 0 to k = M-1 where Mis the number of feed forward taps in the FIR filter. Note that the FIR filter output depends only on the previous M inputs. This feature is why the impulse response for a FIR filter is finite.

They are also known as non-recursive digital filters as they do not have the feedback as in case of IIR (Infinite Impulse Response) Filters. A diagram shown below depicts FIR and IIR configurations where x (n) is the input signal and y(n) is the output signal in time domain. FIR filters can be designed using different methods but most of them are based on ideal filter approximation. The objective is not to achieve the ideal characteristics, as it is impossible anyway but to achieve sufficient good characteristics of a filter. The transfer function of a FIR filter approaches the ideal as the filter order increases, thus increasing the complexity and amount of time needed for processing input samples of a signal being filtered.



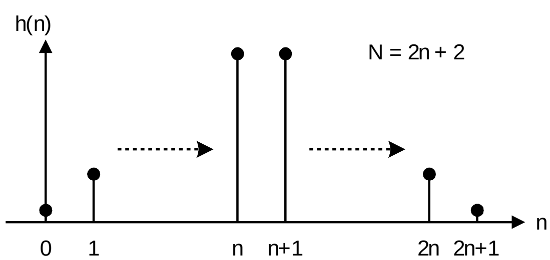
Advantages of FIR Filters

FIR filtering has these advantages over IIR filtering:

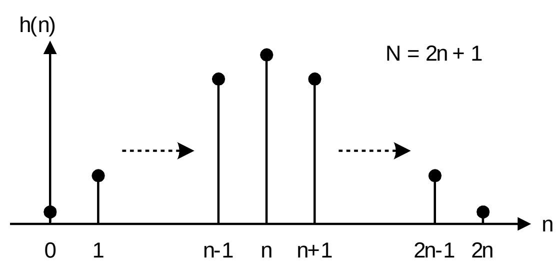
- It can implement linear-phase filtering. This means that the filter has no phase shift across the frequency band. Alternately, the phase can be corrected independently of the amplitude.
- It can be used to correct frequency-response errors in a loudspeaker to a finer degree of precision than using IIRs. However, FIRs can be limited in resolution at low frequencies, and the success of applying FIR filters depends greatly on the program that is used to generate the filter coefficients. Usage is generally more complicated and time-consuming than IIR filters.
- FIR filters are simple to design and they are guaranteed to be bounded input-bounded output (BIBO) stable. By designing the filter taps to be symmetrical about the center tap position, a FIR filter can be guaranteed to have linear phase. This is a desirable property for many applications such as music and video processing. FIR filters also have a low sensitivity to filter coefficient quantization errors. This is an important property to have when implementing a filter on a DSP processor or on an integrated circuit.

Phase Linearity of FIR Filter

One very important property of FIR filter is phase linearity. A casual FIR filter whose impulse response is symmetrical is guaranteed to have a linear phase response.



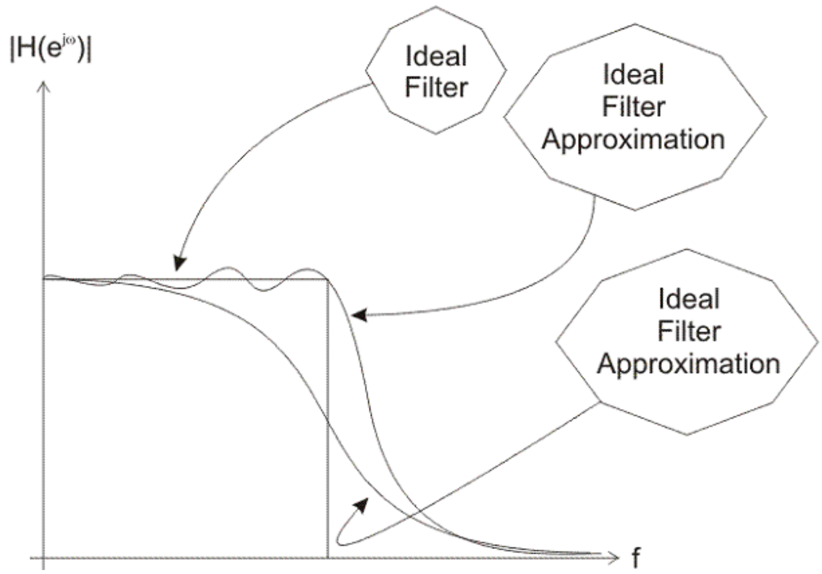
Even Symmetry



Odd Symmetry

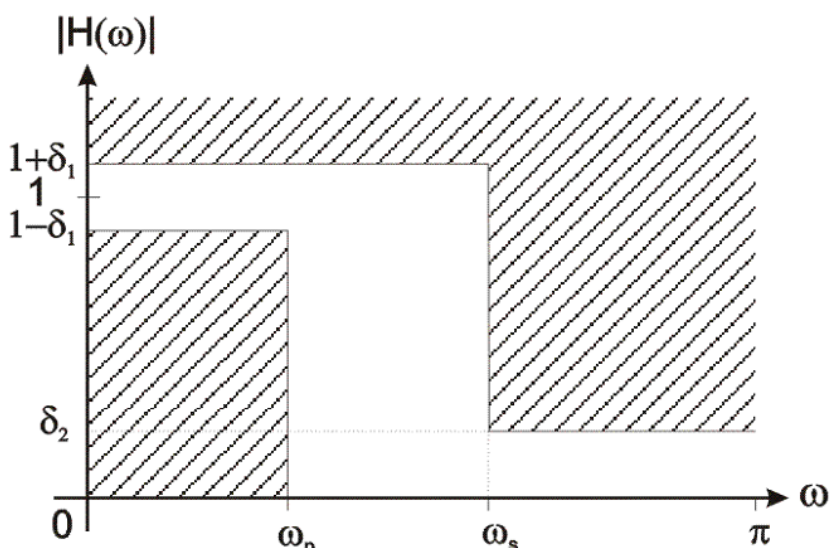
Example of Low Pass Filter to understand Filter Parameters

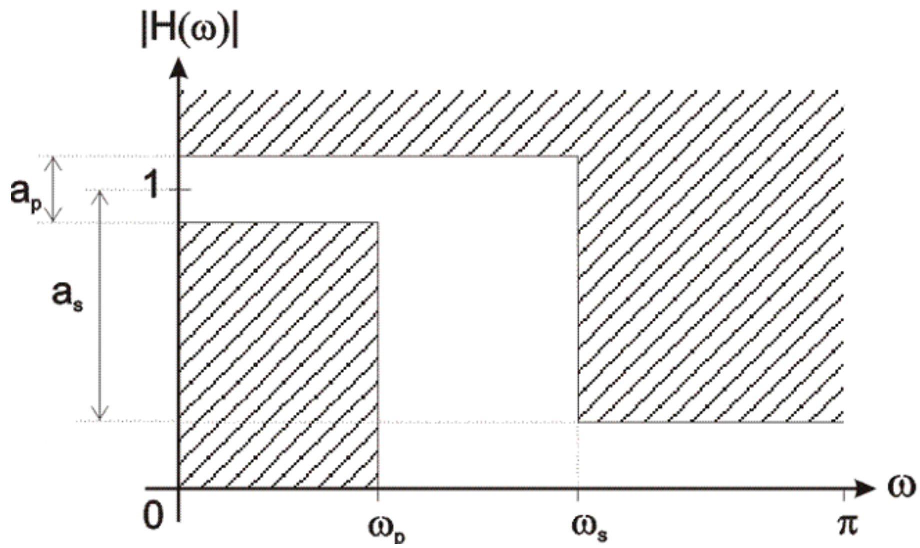
A low pass filter is commonly used to attenuate a certain range of higher frequencies and allow some low frequencies desirable at later stages of signal processing. This filter is most common in applications where removal of noise is vital at early stages of processing. During the design of such a filter, frequency response is the key parameter that determines the efficiency of any filter. The resulting frequency response can be a monotone function or oscillatory function with a certain frequency range. The waveform of frequency response depends on the method used in design process as well as on its parameters. The characteristics of the transfer function as well as its deviation from the ideal frequency response depend on the filter order. Each filter category has its advantages and its disadvantages and due to this reason it is very important to carefully choose category and type of a filter during the design process.



Let us consider an example of a low pass filter, the approximation of a filter response and ideal response of a low pass filter are shown above. Now, we try to approximate various aspects involved in design of such a filter. In the diagrams shown below the parameters are:

ωp – normalized cut -off frequency in the pass band
ωs – normalized cut -off frequency in the stop band
δ1 – maximum ripples in the pass band
δ2 – minimum attenuation in the stop band [dB]
ap – maximum ripples in the pass band
as – minimum attenuation in the stop band [dB]





$$a_p = 20\log_{10}\left(\frac{1+\delta_1}{1-\delta_1}\right)$$

$$a_s = -20\log_{10}\delta_2$$

Frequency normalization can be expressed as follows:

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

Where:

fs is a sampling frequency;
f is a frequency to normalize
ω is normalized frequency

Ideal Low pass filter

The ideal low pass filter is one that allows through all the frequency components of a signal below a designated cutoff frequency wc rejects all frequency components of a signal above wc.

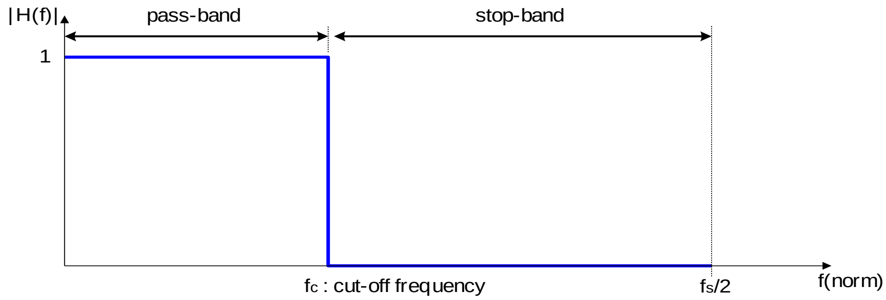
FIR Low pass filter

Because the impulse response required to implement the ideal low pass filter is infinitely long, it is impossible to design an ideal FIR low pass filter. Finite length approximations to the ideal impulse response leads to presence of ripples in both the pass band ($w<w_c$) and the stop band ($w>w_c$) of the filter, as well as to a nonzero transition width between the pass band and stop band of the filter.

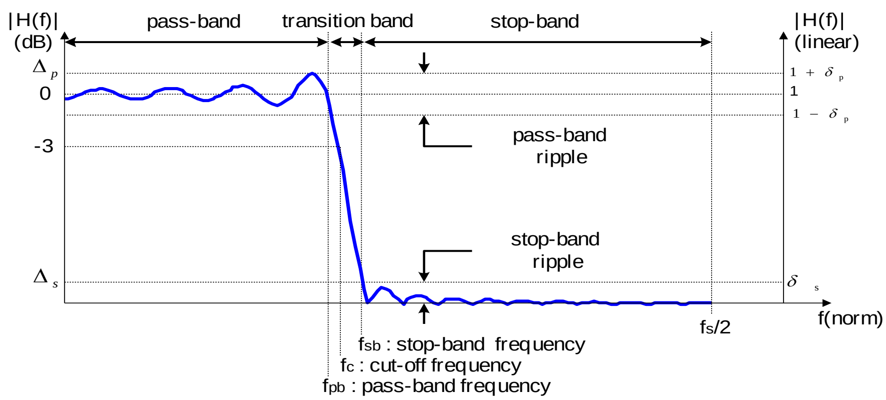
Design Procedure of FIR Filter

To fully design a filter, three steps are required:

1. **Filter specifications**



(a)



(b)

2. **Coefficient Calculation**

There are several popular methods for coefficient calculation but the most popular are:

- Windows Method
- Frequency Sampling

We will consider the **Windows method**

- ***First stage*** of this method is to calculate the coefficients of ideal filter which can be calculated by the equation shown below:

$$
\begin{aligned}
h_d(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)e^{j\omega n}\,d\omega \\
&= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} 1\cdot e^{j\omega n}\,d\omega \\
&= \begin{cases} \dfrac{2f_c \sin(n\omega_c)}{n\omega_c} & \text{for } n \neq 0 \\ 2f_c & \text{for } n = 0 \end{cases}
\end{aligned}
$$

- ***Second stage*** of this method is to select a window function based on the pass band or attenuation specifications, and then determine the filter length based on the required width of the transition band.

Window Type	**Normalised Transition Width (Δf(Hz))**	**Passband Ripple(dB)**	**Stopband Attenuation (dB)**
Rectangular	$\frac{0.9}{N}$	0.7416	21
Hanning	$\frac{3.1}{N}$	0.0546	44
Hamming	$\frac{3.3}{N}$	0.0194	53
Blackman	$\frac{5.5}{N}$	0.0017	74
Kaiser	$\frac{2.93}{N} \rightarrow \beta = 4.54$ $\frac{5.71}{N} \rightarrow \beta = 8.96$	0.0274 0.000275	50 90

By Using the Hamming Window:

$$N = \frac{3.3}{\Delta f} = \frac{3.3}{(1.2-1.4)\,kHz} \cdot 8kHz = 132$$

- ***The third stage*** is to calculate the set of truncated or windowed impulse response coefficients, h[n]:

$$h(n) = h_d(n) \cdot W(n) \text{ for } \begin{array}{ll} -\frac{N-1}{2} \le n \le \frac{N-1}{2} & \text{for N = odd} \\ -\frac{N}{2} \le n \le \frac{N}{2} & \text{for N = even} \end{array}$$

Where:

$$\begin{aligned} W(n) &= 0.54 + 0.46\cos\left(\frac{2\pi n}{N}\right) \\ &= 0.54 + 0.46\cos\left(\frac{2\pi n}{133}\right) \end{aligned} \quad \text{for } -66 \le n \le 66$$

Matlab Simulation Results: