4 Digital Filtering

In this chapter, we'll study digital filtering methods. Specifically, we'll look into the following:

- Filter specifications
- Filtering in frequency domain
- Filtering in time domain
- Simple filter design – Sum and difference (SD) filters
- Finite Impulse Response (FIR) filters
- Infinite Impulse Response (IIR) filters (using MATLAB functions)

4.1 Filter Specifications

Filtering is the process of keeping components of the signal with certain desired frequencies and removing components of the signal with certain undesired frequencies. Very often, we keep the gain of the required frequency components to 1 or close to 1 and the gain of the undesired frequency components will be 0 or close to 0. In general, there are 4 types of filter: low-pass filter (LPF), high-pass-filter (HPF), band-pass filter (BPF) and band-stop filter (BSF). Each filter will have specific characteristics:

- Passband – the range of frequency components that are allowed to pass
- Stopband – the range of frequency components that are suppressed
- Passband ripple – ripples in the passband, the maximum amount by which attenuation in the passband may deviate from gain (which is normally 1)
- Stopband ripple – ripples in the stopband, the maximum amount by which attenuation in the stopband may deviate from gain (which is normally 0)
- Stopband attenuation – the minimum amount by which frequency components in the stopband are attenuated
- Transition band – the band between the passband and the stopband.

Magnitude frequency responses of ideal filters are shown in Figure 1 where fc is the cut-off frequency with F as the sampling frequency.

Email: haider.jabber@uomus.edu.iq

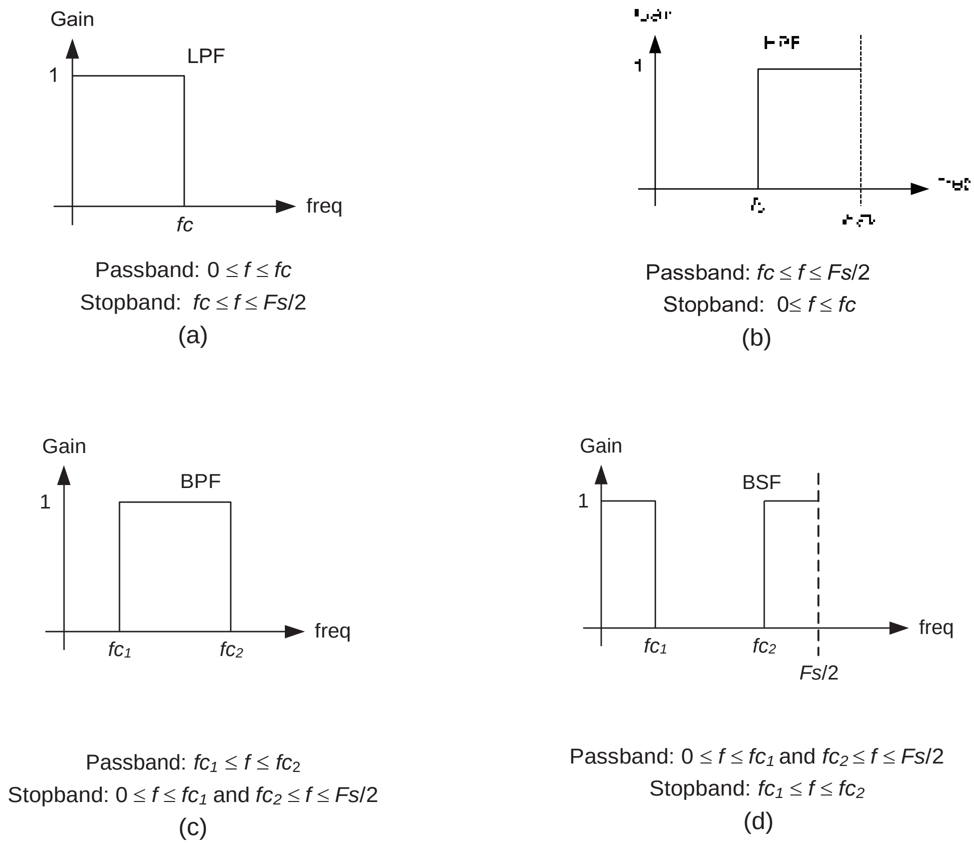


Figure 4.1: Ideal magnitude frequency responses (a) LPF (b) HPF (c) BPF (d) BSF.

Figure 4.1: Ideal magnitude frequency responses (a) LPF (b) HPF (c) BPF (d) BSF.

4.1.1 Low-pass filter

A LPF passes all low-frequency components below the cut-off frequency, *fc* and blocks all higher frequency components above *fc*. Figure 4.2 shows the magnitude frequency response of a LPF in reality, where we can't design 'square' type of filters as shown in Figure 4.1. So, there needs to be transition band between the passband and stopband. The edge frequencies are the end frequencies of passband (*fp*) or stopband (*fs*). So, a practical LPF will allow frequency components below *fp* and remove components higher than *fs*.

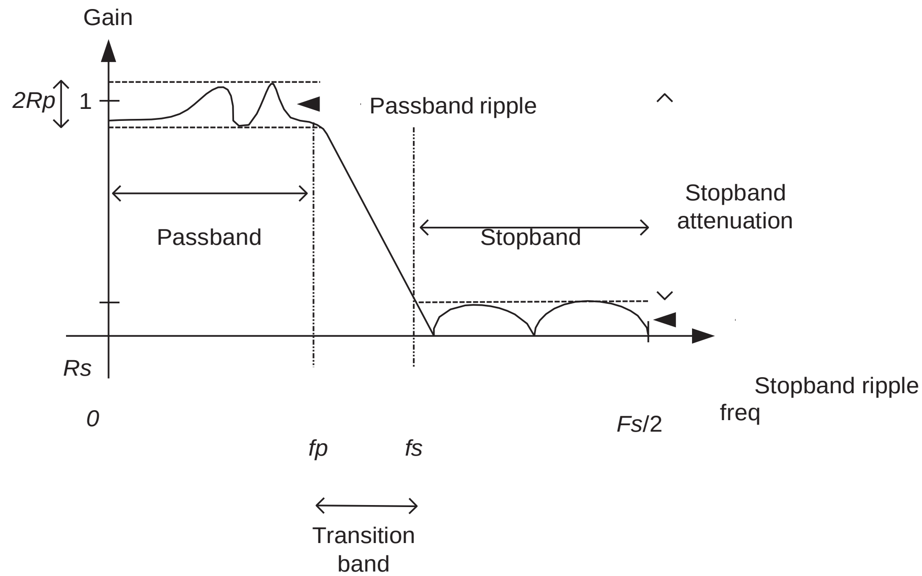
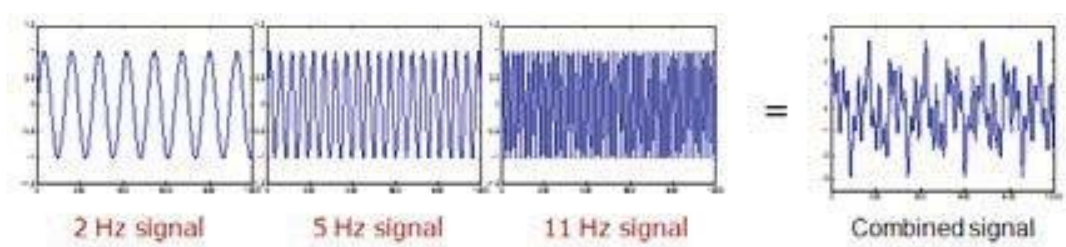


Figure 4.2: Magnitude frequency response of a LPF.

For example, consider a combination of three sinusoidal signals: 2 Hz, 5 Hz and 11 Hz as shown in Figure 4.3



The final output signals after LPF at *fp*=3 Hz with *fs*=4 Hz and *fp*=8 Hz with *fs*=9 Hz are shown in Figure 4.4.

Figure 4.3: A combination of three sinusoidal signals.

The final output signals after LPF at *fp*=3 Hz with *fs*=4 Hz and *fp*=8 Hz with *fs*=9 Hz are shown in Figure 4.4.

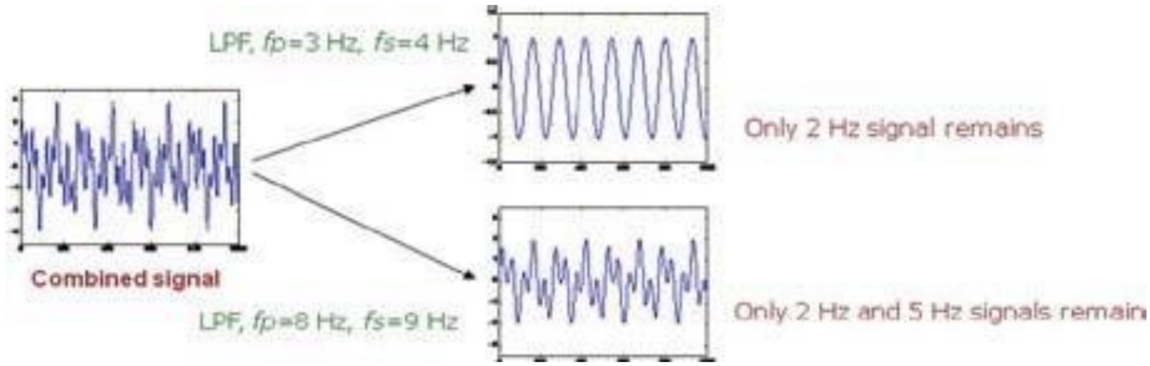


Figure 4.4: LPF of the three sinusoidal signals.

4.1.2 High-pass filter

HPF passes all high-frequency components above the cut-off frequency, *fc* and blocks all lower frequency components below *fc*. The magnitude frequency response of a HPF in reality is shown in Figure 4.5 where it allows frequency components higher than *fp* and remove components below *fs*.

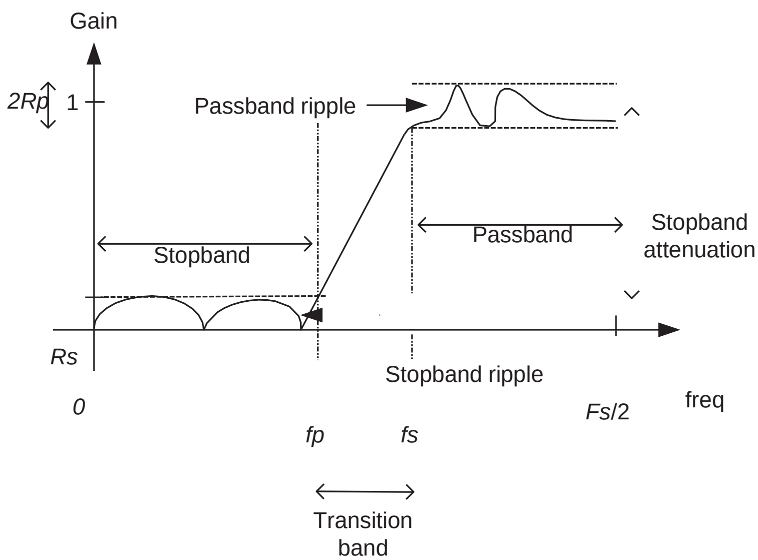


Figure 4.5: Magnitude frequency response of a HPF.

Consider the same combination of three sinusoidal signals: 2 Hz, 5 Hz and 11 Hz as previously. The final output signals after HPF at *fs*=3 Hz with *fp*=4 Hz and *fs*=8 Hz with *fp*=9 Hz are shown in Figure 4.6.

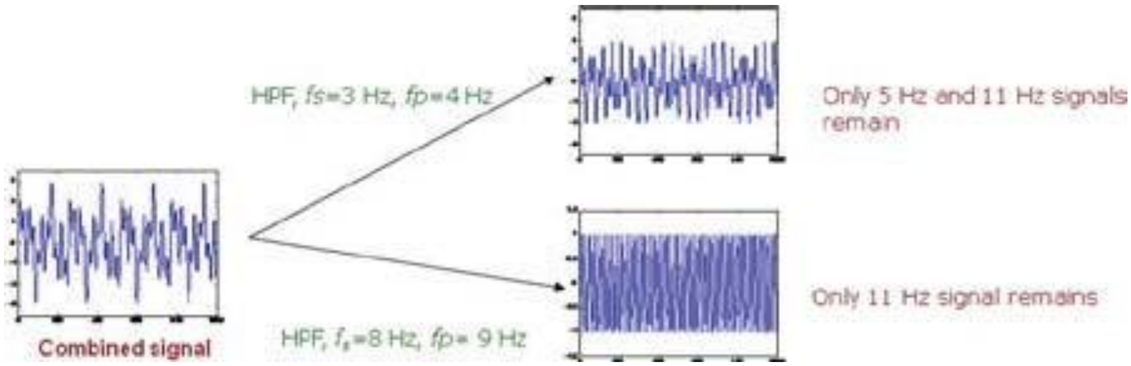


Figure 4.6: HPF of the three sinusoidal signals.

4.1.3 Band-pass and band-stop filters

BPF passes all frequency components between edge passband frequencies, $fp_1<\text{freq}_{(\text{allow})}<fp_2$ and blocks all frequencies below and above edge stopband frequencies, $\text{freq}_{(\text{block})}<fs_1$; $\text{freq}_{(\text{block})}>fs_2$. A BPF can be designed using a LPF and HPF. BSF passes all frequency components lower and higher than edge passband frequencies, $\text{freq}_{(\text{allow})}<\boldsymbol{fp}_1$; $\text{freq}_{(\text{allow})}>\boldsymbol{fp}_2$ and blocks all frequencies between $\boldsymbol{fs}_1<\text{freq}_{(\text{block})}<\boldsymbol{fs}_2$. The magnitude frequency responses of a BPF and a BSF are shown in Figures 4.7 and 4.8.

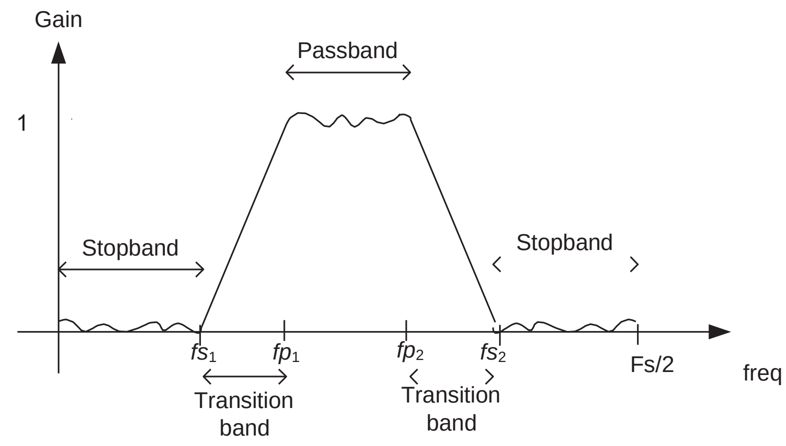


Figure 4.7: Magnitude frequency response of a BPF.

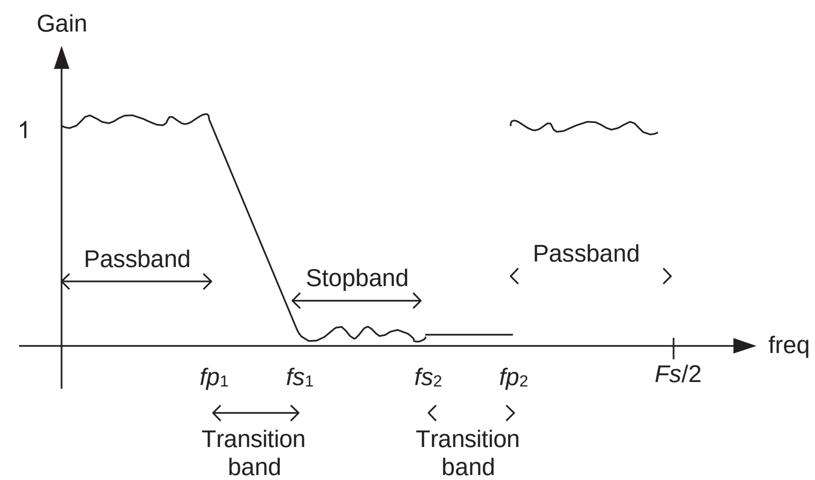
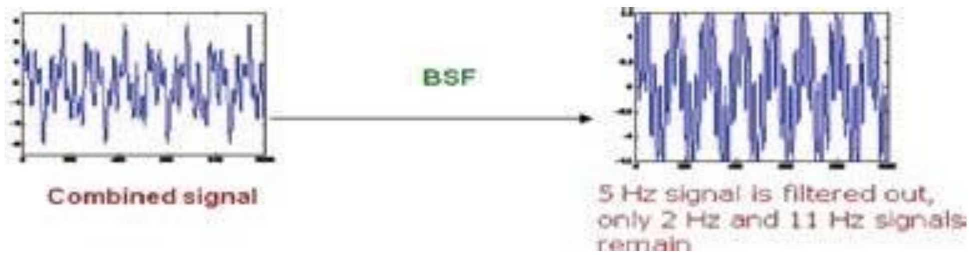


Figure 4.8: Magnitude frequency response of a BSF.

Figure 4.9 shows the output signals after applying BPF at fp_1=4 Hz, fp_2=6 Hz, fs_1=3 Hz, fs_2=7 Hz and BSF fp_1=4 Hz, fp_2=6 Hz, fs_1=3 Hz, fs_2=7 Hz for the combination of the sinusoidal signals.

(a)



(b)

Figure 4.9: (a) BPF (b) BSF of the three sinusoidal signals.

4.2 Direct filtering in frequency domain

Filtering can be done directly in the frequency domain using the following steps:

- Obtain the Discrete Fourier Transform (DFT) of the signal (from 0 to F);
- Set to zero the values that are not in the required frequency range i.e. apply a rectangular window;
- Compute the Inverse Discrete Fourier Transform (IDFT).

For example, let use generate a combination of two sinusoidal signal with f =8 Hz and f =25 Hz with =100, F =200 Hz (shown in Figure 4.10) and say, we wish to design a LPF with fp=10 Hz and fs=12 Hz. Compute `y=fft(x)` in MATLAB and apply the rectangular window, i.e. set the values `y(7:95)=0`. As MATLAB indexing starts from 1, `y(1:6)` represents DFT values from 0 to 10 Hz[16], which represents the passband range, the stopband range from 12 Hz to 100 Hz is represented by `y(7:51)`. Due to symmetry, we also need to create mirror images of the passband and stopband resulting in the rectangular window as shown in Figure 4.11 (a).

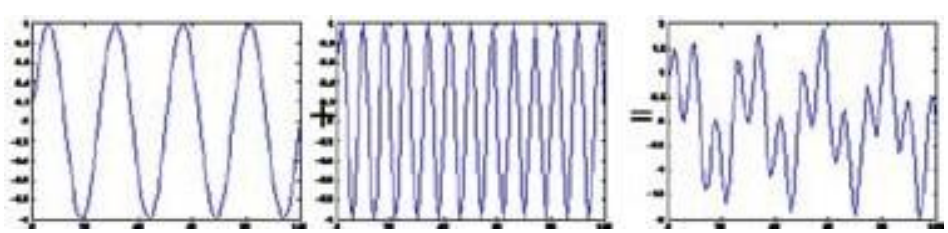


Figure 4.10: Combination of two sinusoidal signals ($f1$=8 Hz and $f2$=25 Hz).

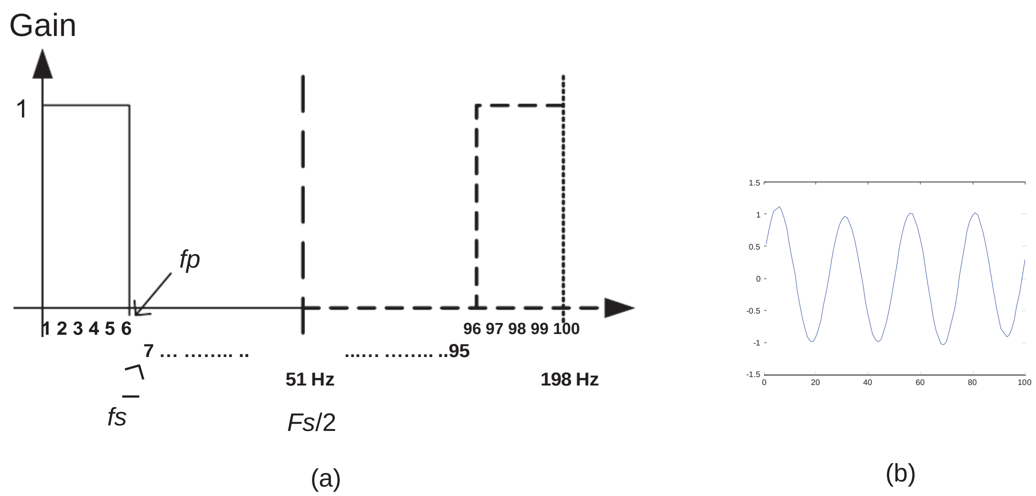


Figure 4.11: Direct LPF (a) rectangular window with *fp*=10 Hz and *fs*=12 Hz (b) filtered output.

Next, compute `yf=ifft(y,'symmetric')` and the low pass filtered signal is obtained as shown in Figure 4.11 (b). In MATLAB, it is useful to force conjugate symmetry, else complex values could be obtained due round-off errors in the `fft` and `ifft` operations. Figure 4.12 shows the whole procedure for the discussed example. This direct filtering method is simplistic to understand but has the disadvantage of high computation cost and requires chunks of data (i.e. real-time filtering is not possible). Thus we normally use finite impulse response (FIR) or infinite impulse response (IIR) filters to perform filtering.

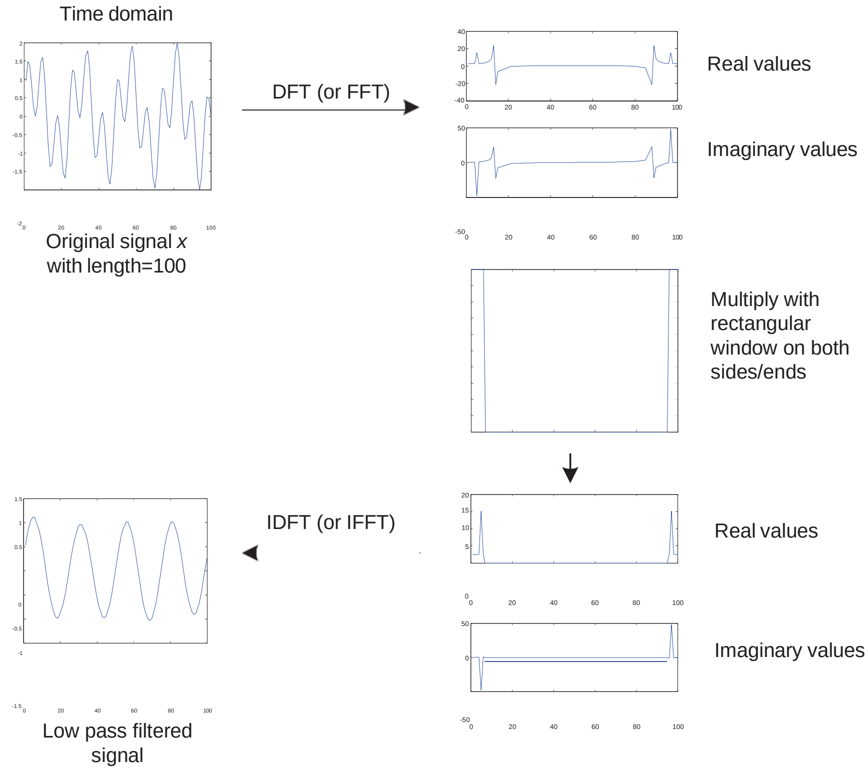


Figure 4.12: LPF example using direct filtering method.

4. Time domain filtering

To solve the problems of direct filtering, we could filter in time domain and there are several time domain filtering methods. We will look at design of simple FIR filters and IIR filters using MATLAB. The output from an IIR digital filter is made up of previous inputs and previous outputs:

$$y[n] = \sum_{k=1}^{M} B[k]x[n-k] + \sum_{j=1}^{N} A[\,j]y[n-j]. \tag{4.1}$$

where and A are the filter coefficients. The output from a FIR digital filter is made up of previous inputs only, so there is no feedback:

$$y[\quad] = \sum_{k=1}^{M} \quad [k]\ [\quad -k]. \tag{4.2}$$

Figure 4.13 shows an example comparing direct filtering in the frequency domain with time domain filtering. It should be obvious from this figure that filtering in time domain is computationally less complicated.

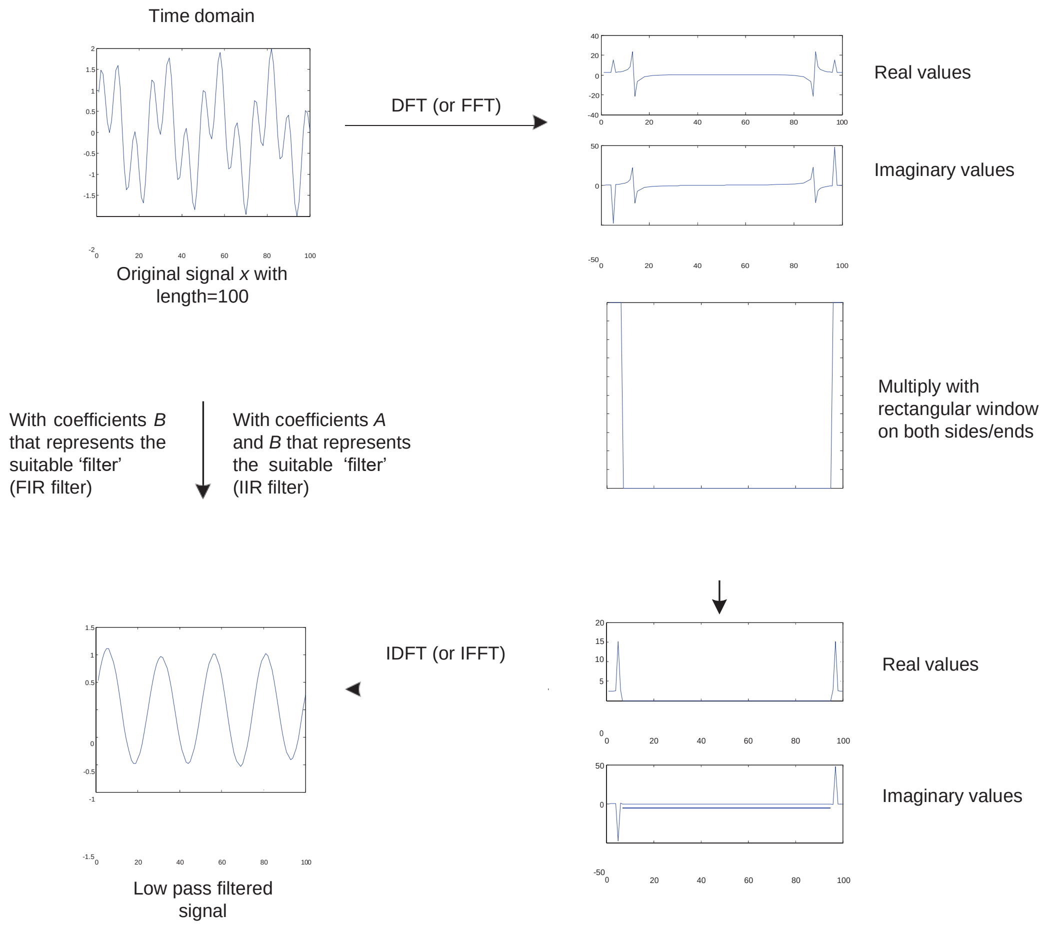


Figure 4.13: LPF comparison using direct frequency filtering and time domain methods.

4.4 Simple FIR filters

Simple FIR filter are also known as sum and/or difference filter. Consider a sum filter

$$y[\;]=\frac{1}{2}(x[n]+\;[\;-1]), \qquad (4.3)$$

for every data $x[n]$ in the signal; this simple addition can be shown using Z-transform to act as a LPF! Considering (4.2), the filter coefficients are $B[1]$=0.5 and $B[2]$=0.5.

For hardware design, the block diagram is shown in Figure 4.14.

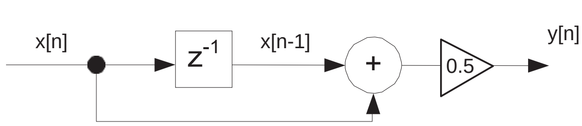


Figure 4.14: Block diagram of a simple sum filter.

The advantages of such a filter are:

- Only one adder and one delay[17] block is needed, so simple design and low cost;
- The filter coefficients are finite values (in this case they are 0.5), so no errors caused by round-off;
- It is an FIR filter, so it is stable[18];
- It's phase response is linear (more on this later).

However, the magnitude (gain) response is not very good as it is far from the ideal 'square' filter (see Figure 4.15).

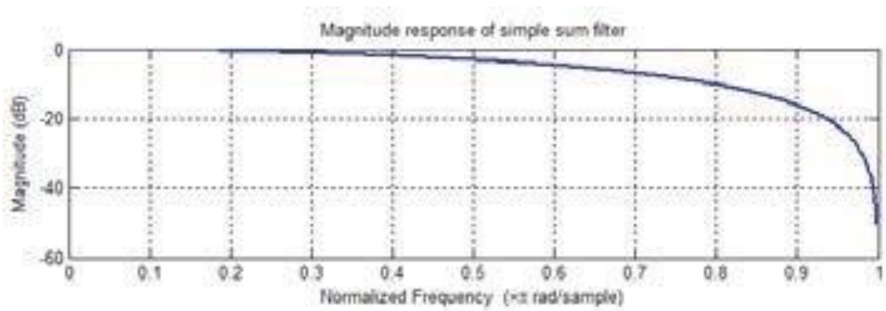


Figure 4.15: Magnitude response of the simple sum filter.

The magnitude (gain) at normalised frequency 0 is 1 (i.e. 0 dB[19]) and the stopband frequency (when gain drops to 0) is at π rad/sample or at $Fs/2$ Hz, where is the sampling frequency. So, it appears that there is no stopband and for these cases, we use the 3 dB cut-off as the passband frequency. The 3 dB cut-off frequency is defined as the frequency when the gain drops 3 dB from maximum gain of 1, which is 0 dB.

So when energy is half, i.e. gain=$(1/2)^{0.5}$=0.7071, we have $20\log_{10}(0.7071)$=-3 dB. From Figure 4.15, we can see that the 3 dB cut-off frequency (when gain=0.7071) is approximately ≈ 0.5 rad/sample or ≈ /4 Hz. This is the edge passband frequency and so the passband is from 0 to $s/4$ Hz and transition band is from 4 to 2.

Similarly, a HPF can be designed using a difference filter:

$$y[n] = \frac{1}{2}(x[n] - x[n-1])\,. \tag{4.4}$$

The block diagram is shown in Figure 4.16.

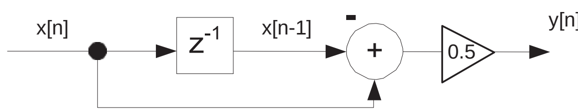


Figure 4.16: Block diagram of a simple difference filter.

And the magnitude response is given in Figure 4.17.

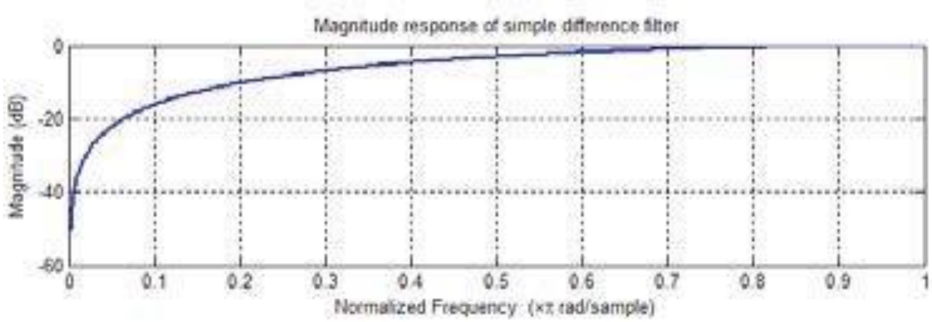


Figure 4.17: Magnitude response of the simple difference filter.

The stopband frequency (when gain=0) is at 0 Hz, i.e. there is no stopband. The gain at $Fs/2$ Hz or π rad/sample is 1 and hence, the edge passband frequency is at $Fs/4$ Hz (using 3 dB cut-off approach). The passband width is from $Fs/4$ to $Fs/2$ Hz.

4.4.1 Increasing the order of the simple filter

The order of the filter can be increased to obtain a smaller passband width and to obtain a frequency response closer to the ideal 'square' filter. The sum filter in (4.3) had an order of 1; if we cascaded another first order sum filter, we will have the block diagram shown in Figure 4.18 (we'll drop the constants for simplicity of discussion):

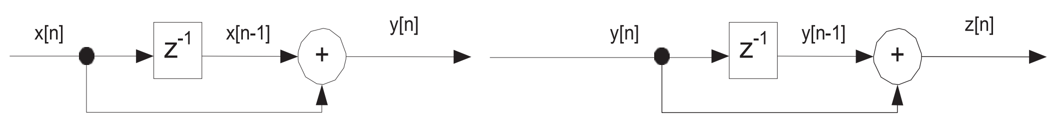


Figure 4.18: Magnitude response of two first order sum filters (effectively a second order sum filter).

Solving for $z[n]$ in Figure 4.18 will give

$$\begin{aligned} z[n] &= y[n] + y[n-1], \\ &= x[n] + x[n-1] + x[n-1] + x[n-2], \\ &= x[n] + 2x[n-1] + x[n-2]. \end{aligned} \tag{4.5}$$

Eq. (4.5) could be easily verified by replacing values for $x[n]$. For example, using $x[1]=3$, $x[2]=2$ and $x[3]=5$ and computing $z[3]$ for the single cascaded second order filter will give 12. Computation using two first order filters (i.e. $y[2]$ and $y[3]$) will give the same result.

It should be noted that $z[n]$ in the example above will be defined only for n=3 onwards if $x[1]$ is the starting point of the signal, i.e. for every order M, M initial data points will be lost in filtering. Likewise $y[n]$ in (4.3) is defined only from $y[2]$ onwards.

For order M, we have

$$y(n) = \sum_{r=0}^{M} {}^{M}C_r \, x[n-r] \quad \text{where} \quad {}^{M}C_r = \frac{M!}{r!(M-r)!}. \tag{4.6}$$

As an example, for order M=3, we will have

$$y(n) = x[n] + 3x[n-1] + 3x[n-2] + x[n-3]. \tag{4.7}$$

The magnitude response is given in Figure 4.19.

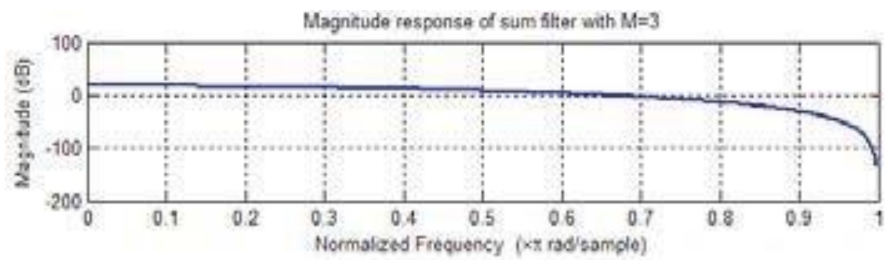


Figure 4.19: Magnitude response of the third order sum filter.

The passband is about 0.302 rad or $\approx F/6$. It could be seen that with increasing order, the passband is becoming smaller without any change in stopband. Also, the response is becoming closer to the ideal 'square'. So, we can increase/decrease M depending on the requirements. The 3-dB cut-off frequency is given by

$$fp = \frac{2}{\pi}\cos^{-1}(2^{-1/2M})\,. \tag{4.8}$$

Similarly for the HPF with order , we will have

$$fp = \frac{2}{\pi}\sin^{-1}(2^{-1/2N})\,, \tag{4.9}$$

and the magnitude response as

$$y(n) = \sum_{r=0}^{N} (-1)^r \left({}^N C_r \, x[n-r] \right). \tag{4.10}$$

4.4.2 BPF design using sum and difference filter

Similarly, a BPF can be designed using a combination of LPF and HPF. This BPF is known as sum and difference (SD) filter. Different orders, M and N can be chosen to obtain the required frequency response [1]:

$$G_{M,N}(f) = (2\cos \pi f T)^M (2\sin \pi f T)^N / Gain_{cf} , \tag{4.11}$$

where $Gain_{cf}$ is the gain at centre frequency given by

$$fcentre = \frac{f_s}{2\pi} \cos^{-1}\left(\frac{M-N}{M+N} \right). \tag{4.12}$$

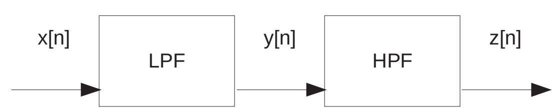


Figure 4.20: BPF design using SD filter.

For example, with filter orders of M=28 and N=8 gives the centre frequency of 40 Hz when Fs=256 Hz. The approximate 3 dB bandwidth is from 32 to 48 Hz (rounded to the nearest integer) and the gain amplification at 40 Hz is approximately 47.21. Figure 4.20 shows this example using different filter orders but with similar centre frequency (which is dependent on ratio of M/N).

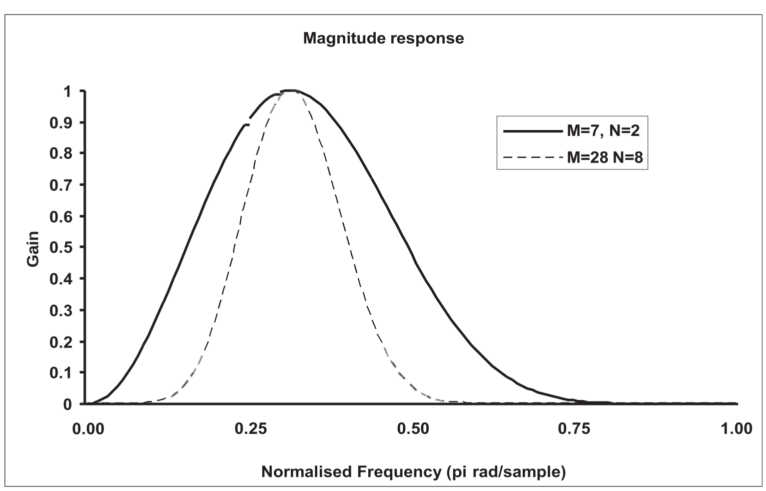


Figure 4.21: BPF magnitude response for M=7, N=2 and M=28 and N=4 (note: ordinate shows gain as actual values and not as dB).

As another example, let us obtain the band pass FIR filter expression for orders, LPF, =4 and HPF, =1, i.e. obtain the band pass FIR equation that expresses *z*[ ] in terms of [ ] and delays of [ ]. Using (4.6), obtain *y*[ ] in term of [ ] and using (4.10), obtain *z*[ ] in terms *y*[ ]. Next, replace *y*[ ] in the latter expression to arrive at

$$z[n] = x[n] + 3x[n-1] + 2x[n-2] - 2x[n-3] - 3x[n-4] - x[n-5]. \qquad (4.13)$$

4.5 FIR filter design using window method

The SD filter that we studied in the previous section is simple to design but for practical purposes, we often need filters that can be tailored to suit our required specifications. Consider doing an inverse DFTof the ideal LPF shown in Figure 4.1 (a) to obtain what is known as the impulse response, which are basically the filter coefficients[20].

The impulse response is actually the *sinc* function given by

$$h_{LPF}[n] = \frac{\sin(2\pi f_c n)}{\pi n}, \qquad -\infty < n < \infty \qquad (4.14)$$

It would be obvious that we will not be able to use h_{LPF} as the filter coefficients as the length is infinite. So we could use a rectangular window, *w*[*n*] to truncate the impulse response. However, by using a finite set of coefficients (i.e. impulse response), the shape of the magnitude response is changed with ripples showing up as in Figure 4.20. This is known as the Gibbs phenomenon – oscillatory behaviour in the magnitude responses caused by truncating the ideal impulse response function (i.e. the rectangular window has an abrupt transition to zero). Gibbs phenomenon can be reduced by

- using a window that tapers smoothly at each end such as Hamming, Hanning, triangular etc (refer to Section 3.5 in the previous chapter);
- providing a smooth transition from passband to stopband in the magnitude specifications.

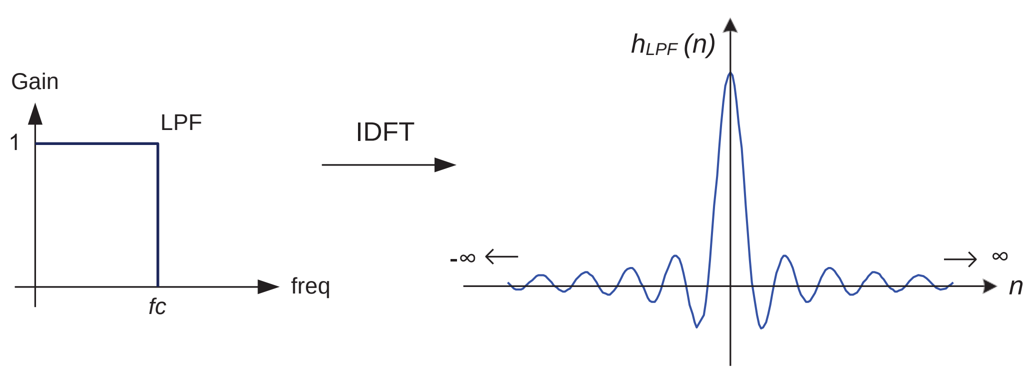


Figure 4.21: Impulse response for LPF.