Tribhuvan University

Institute of Engineering

Pulchowk Campus

DIGITAL SIGNAL ANALYSIS AND PROCESSING

Lab 6

IIR and FIR Filter Design

SUBMITTED BY:

Bishal Katuwal

075BCT028

SUBMITTED TO:

Department of Electronics and Computer Engineering

Pulchowk Campus

SUBMITTED ON:

23rd August, 2022

Title
IIR and FIR Filter Design

Background Theory

Design of IIR Filters

There are several methods that can be used to design digital filters having an infinite duration unit sample response. One of the popular methods is based on converting an analog filter into a digital filter. In this method we begin the design of digital filter in the analog domain and then convert the design into the digital domain. For this purpose, depending on the specifications of the required digital filter the various approximations like butterworth, chebyshev, chebyshev2 and elliptic filters are used. In this lab we deal with following approximations:

- Butterworth Approximation
- Chebyshev Approximation
  - Type I
  - Type II
- Elliptic Approximation

Among the different approaches used in the design of digital IIR filters this lab session deals with the,

- Impulse Invariance method
  In Impulse Invariance method, the objective is to design an IIR filter having an unit sample response h(n) that is the sampled version of the impulse response of the analog filter.That is

$$\begin{aligned} h(n) &= h(nT) \\ & n = 0, 1, 2, ... \\ & T \textit{ is the sampling interval} \end{aligned} \tag{1}$$

- Bi-Linear Transformation
  In Bi-Linear transformation a conformal mapping from s plane to z plane is carried out with the relation:

$$s = \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \tag{2}$$

Design of FIR Filters

The major difficulty lies in the implementation of the non-iterative direct design method for IIR filters. However FIR filters are almost entirely restricted to discrete-time implementations. Thus the design techniques for FIR filters are based on directly approximating the desired frequency response of the discrete time system. Furthermore, most techniques for approximating the magnitude response of the FIR system assume a linear phase constraint; thereby avoiding the problem of spectrum factorization that complicates the direct design of IIR filters.

The simplest method of FIR design is called the window method. This lab deals with following windows:

- Rectangular window
- Bartlett window
- Blackmann window
- Hamming window
- Hanning window
- Kaiser window

MATLAB
Signal package has following functions for filter design.
For IIR

- **besselap** : Return bessel analog filter prototype.
- **besself** : Generate a Bessel filter.
- **bilinear** : Transform a s-plane filter specification into a z-plane specification.
- **buttap** : Design lowpass analog Butterworth filter.
- **butter** : Generate a Butterworth filter.
- **buttord** : Compute the minimum filter order of a Butterworth filter with the desired response characteristics.
- **cheb** : Returns the value of the nth-order Chebyshev polynomial calculated at the point x.
- **cheb1ap** : Design lowpass analog Chebyshev type I filter.
- **cheb1ord** : Compute the minimum filter order of a Chebyshev type I filter with the desired response characteristics.
- **cheb2ap** : Design lowpass analog Chebyshev type II filter.
- **cheb2ord** : Compute the minimum filter order of a Chebyshev type II filter with the desired response characteristics.
- **cheby1** : Generate a Chebyshev type I filter with RP dB of passband ripple.
- **cheby2** : Generate a Chebyshev type II filter with RS dB of stopband attenuation.
- **ellip** : Generate an elliptic or Cauer filter with RP dB of passband ripple and RS dB of stopband attenuation.
- **ellipap** : Design lowpass analog elliptic filter.
- **ellipord** : Compute the minimum filter order of an elliptic filter with the desired response characteristics.

- **iirlp2mb** : IIR Low Pass Filter to Multiband Filter Transformation
- **impinvar** : Converts analog filter with coefficients B and A to digital, conserving impulse response.
- **invimpinvar** : Converts digital filter with coefficients B and A to analog, conserving impulse response.
- **ncauer** : Analog prototype for Cauer filter.
- **pei_tseng_notch** : Return coefficients for an IIR notch-filter with one or more filter frequencies and according (very narrow) bandwidths to be used with 'filter' or 'filtfilt'.
- **sftrans** : Transform band edges of a generic lowpass filter (cutoff at W=1) represented in splane zero-pole-gain form.

  For FIR

- **barthannwin** : Return the filter coefficients of a modified Bartlett-Hann window of length M.
- **blackmanharris** : Return the filter coefficients of a Blackman-Harris window of length M.
- **blackmannuttall** : Return the filter coefficients of a Blackman-Nuttall window of length M.
- **boxcar** : Return the filter coefficients of a rectangular window of length M.
- **chebwin** : Return the filter coefficients of a Dolph-Chebyshev window of length M.
- **hann** : Return the filter coefficients of a Hanning window of length M.
- **kaiser** : Return the filter coefficients of a Kaiser window of length M.
- **rectwin** : Return the filter coefficients of a rectangular window of length M.
- **triang** : Return the filter coefficients of a triangular window of length M.
- **window** : Create an M-point window from the function F.

Activity

1. Butterworth approximation

```
pkg load signal
wp=0.2*pi;
ws=0.5*pi;
As=30;
Ap=0.4;
[N,wn]=buttord(wp/pi,ws/pi,Ap,As);
[b,a]=butter(N,wn);
freqz(b,a);
```

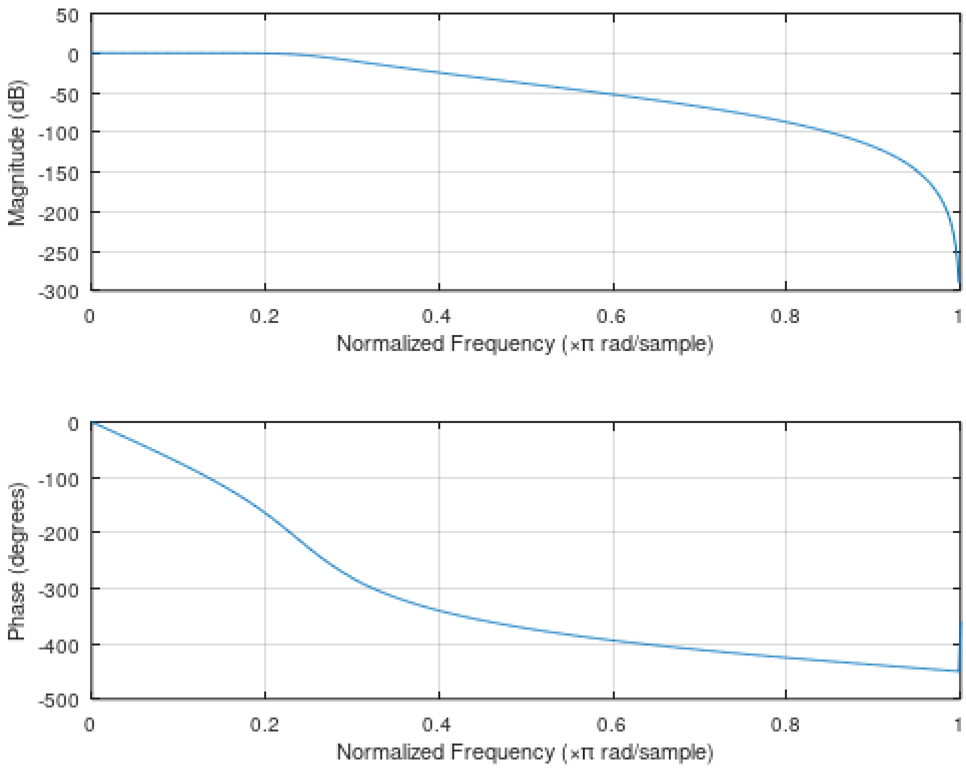

Figure 1: Butterworth approximation

2. For data sampled at 1000 Hz, design a lowpass filter with less than 3 dB of ripple in the passband defined from 0 to 40 Hz and at least 60 dB of ripple in the stopband defined from 150 Hz to the Nyquist frequency (500 Hz):

   - Chebyshev I

```
pkg load signal

Wp = 40/500;
Ws = 150/500;
Rp = 3;
Rs = 60;
```

```
[n,Wn] = cheb1ord(Wp,Ws,Rp,Rs);
[b,a] = cheby1(n,Rp,Wn);
freqz(b,a,512,1000);
title('Chebyshev Type I Lowpass Filter')
```

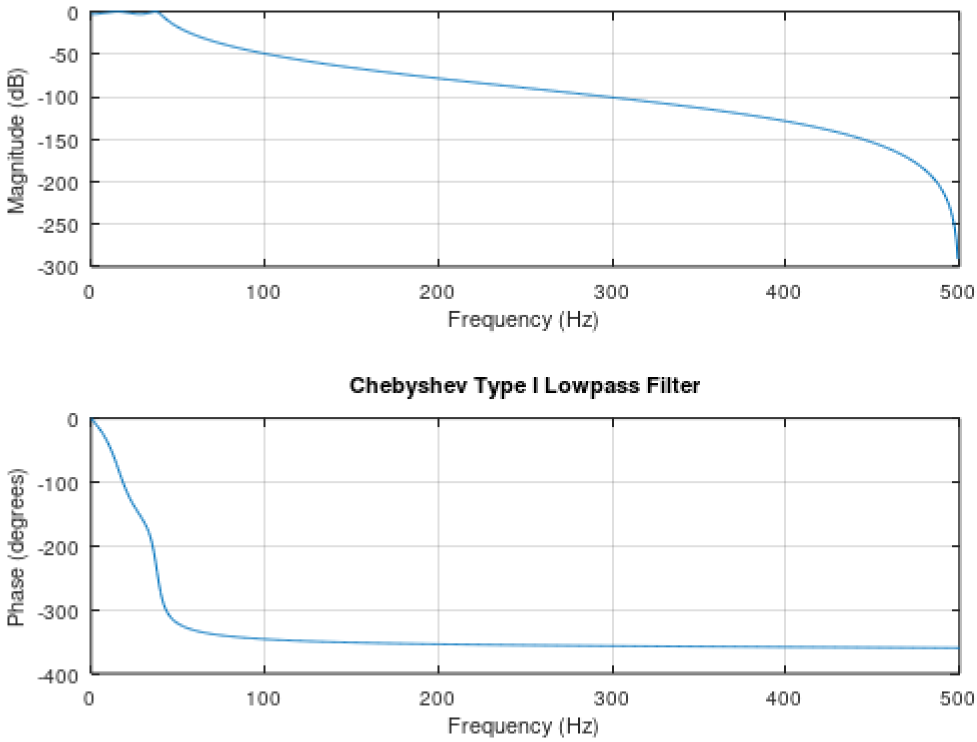

Figure 2: Chebyshev I approximation

- Chebyshev II

```
pkg load signal

Wp = 40/500;
Ws = 150/500;
Rp = 3;
Rs = 60;

[n,Wn] = cheb2ord(Wp,Ws,Rp,Rs);
[b,a] = cheby2(n,Rp,Wn);
freqz(b,a,512,1000);
title('Chebyshev Type II Lowpass Filter')
```

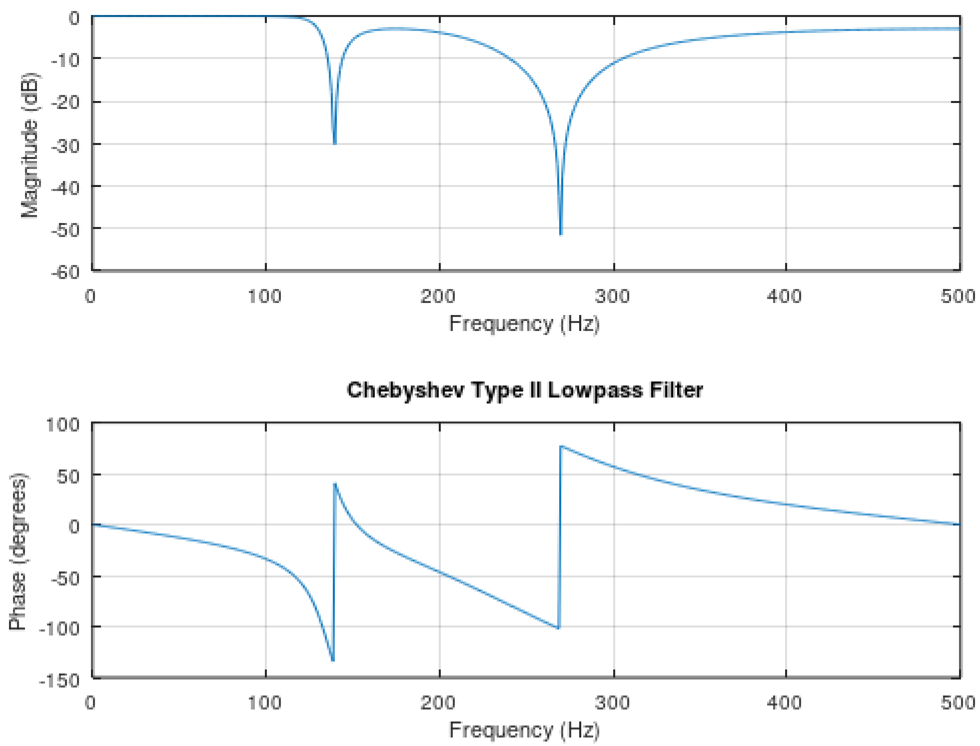

Figure 3: Chebyshev II approximation

3. Design a bandpass filter with a passband of 60 Hz to 200 Hz, with less than 3 dB of ripple in the passband, and 40 dB attenuation in the stopbands that are 50 Hz wide on both sides of the passband:

   - Chebyshev I

```
pkg load signal

Wp = [60 200]/500;
Ws = [50 250]/500;
Rp = 3; Rs = 40;
[n,Wn] = cheb1ord(Wp,Ws,Rp,Rs);
[b,a] = cheby1(n,Rp,Wn);
freqz(b,a,512,1000);
title('Chebyshev Type I Bandpass Filter')
```

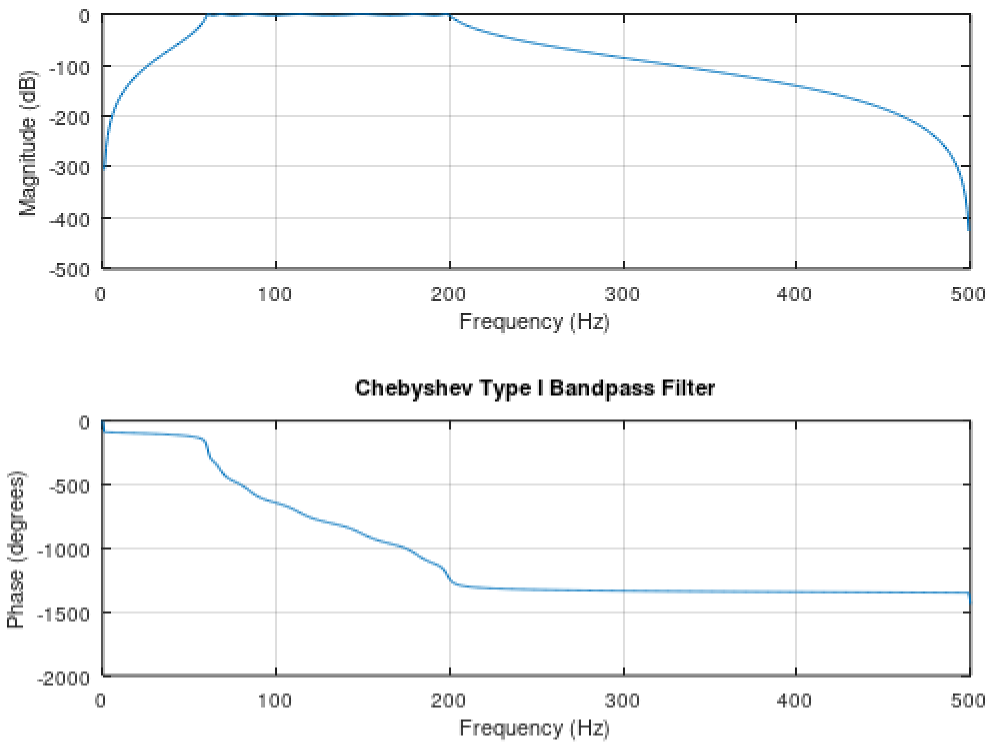

Figure 4: Chebyshev I approximation

- Chebyshev II

```
pkg load signal

Wp = [60 200]/500;
Ws = [50 250]/500;
Rp = 3; Rs = 40;
[n,Wn] = cheb2ord(Wp,Ws,Rp,Rs);
[b,a] = cheby2(n,Rp,Wn);
freqz(b,a,512,1000);
title('Chebyshev Type II Bandpass Filter')
```

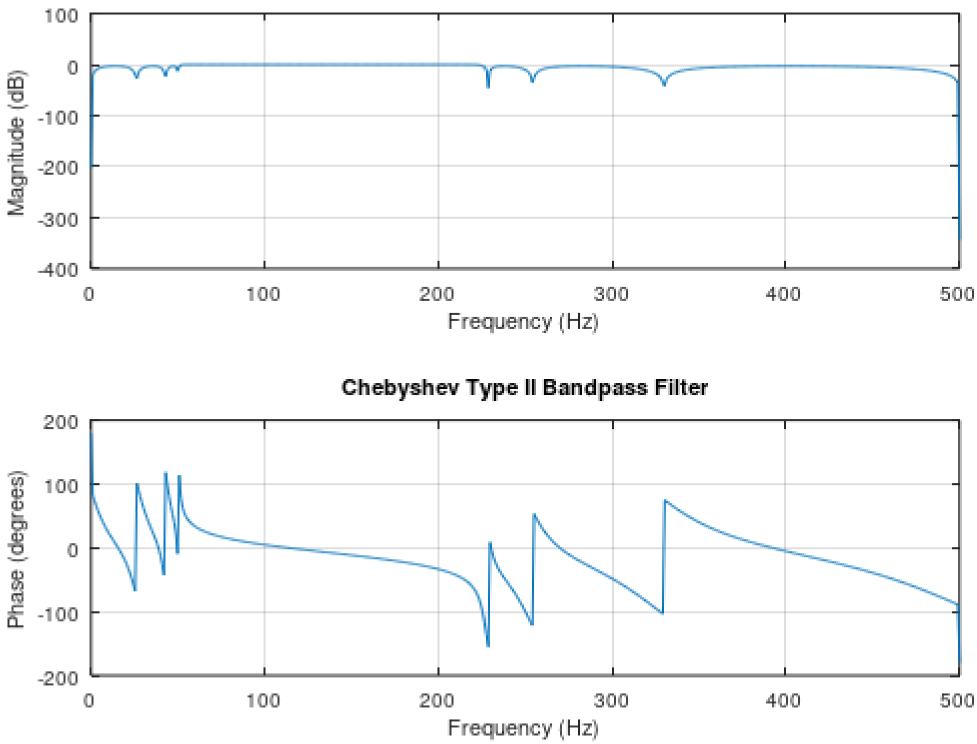

Figure 5: Chebyshev II approximation

4. Comparison of Butterworth, Chebyshev (I and II) and Elliptical approximations:

```
pkg load signal

fc = 3e6; %cut-off frequency
fs = 10e6; %sampling frquency
n = 6; %order of the filter
As = 40; %Stopband attenuation in dB
Rp = 5; %Passband ripple in dB
[b,a] = butter(n,fc/(fs/2));
[h_butter,w_butter] = freqz(b,a,512);
[b,a] = cheby1(n,Rp,fc/(fs/2));
[h_cheby1,w_cheby1] = freqz(b,a,512);
[b,a] = cheby2(n,As,fc/(fs/2));
[h_cheby2,w_cheby2] = freqz(b,a,512);
[b,a] = ellip(n,Rp,As,fc/(fs/2));
[h_ellip,w_ellip] = freqz(b,a,512);
figure;
plot(w_butter/pi,20*log10(abs(h_butter)),'lineWidth',1.2);
hold on
plot(w_cheby1/pi,20*log10(abs(h_cheby1)),'lineWidth',1.2);
hold on
plot(w_cheby2/pi,20*log10(abs(h_cheby2)),'lineWidth',1.2);
hold on
```

```
plot(w_ellip/pi,20*log10(abs(h_ellip)),'lineWidth',1.2)
ylim([-100 10]);grid on;
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')
legend('Butterworth','Cheby1','Cheby2','Elliptic','FontSize',10)
figure;
plot(w_butter/pi,unwrap(angle(h_butter)),'lineWidth',1.2);
hold on
plot(w_cheby1/pi,unwrap(angle(h_cheby1)),'lineWidth',1.2);
hold on
plot(w_cheby2/pi,unwrap(angle(h_cheby2)),'lineWidth',1.2);
hold on
plot(w_ellip/pi,unwrap(angle(h_ellip)),'lineWidth',1.2);
hold on
grid on;
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Phase (radians)')
legend('Butterworth','Cheby1','Cheby2','Elliptic','FontSize',10)
```

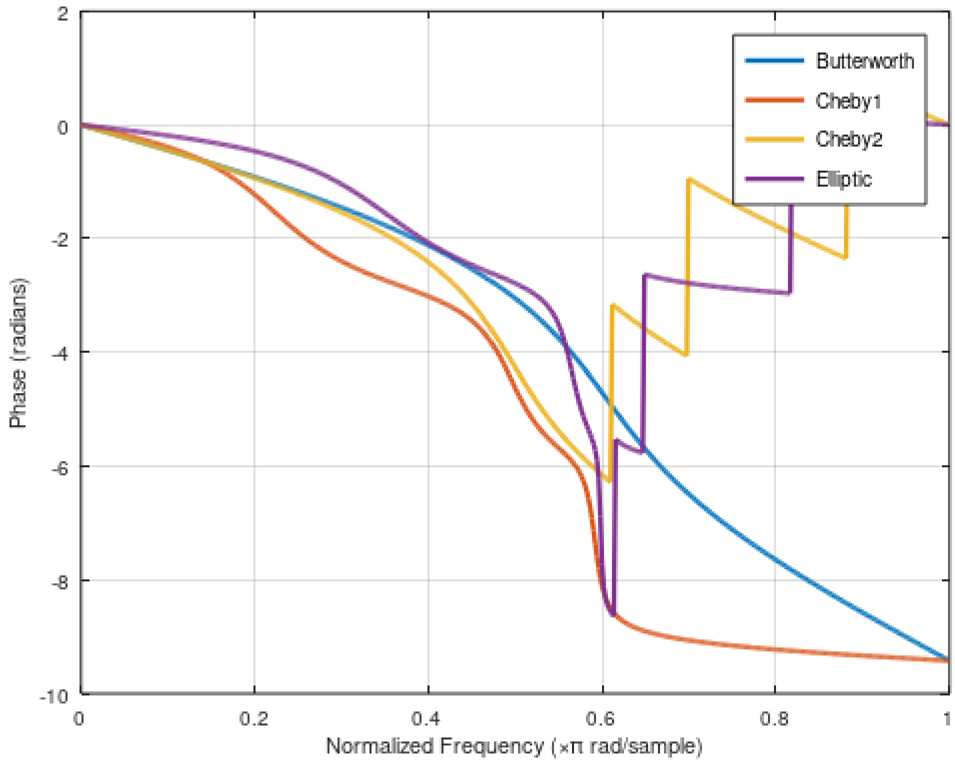

Figure 6: Phase comparisonn

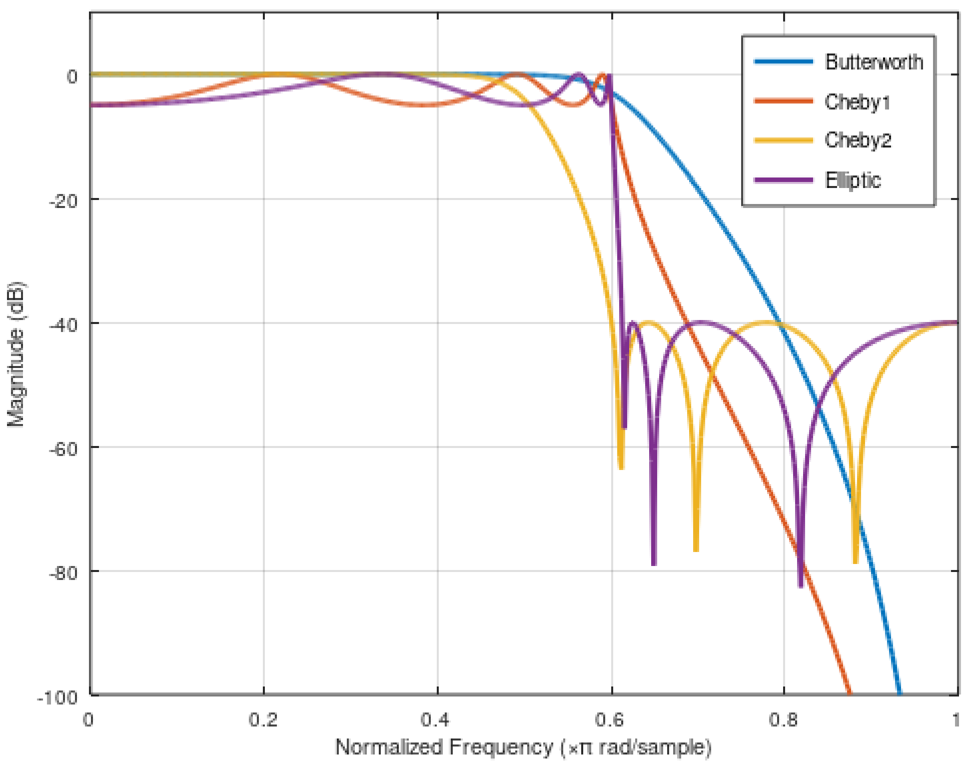

Figure 7: Magnitude comparison

5. FIR filters

(a) Rectangular window

```
N=6;
wc=0.4*pi;
h=fir1(N,wc/pi,rectwin(N+1));
freqz(h);
```

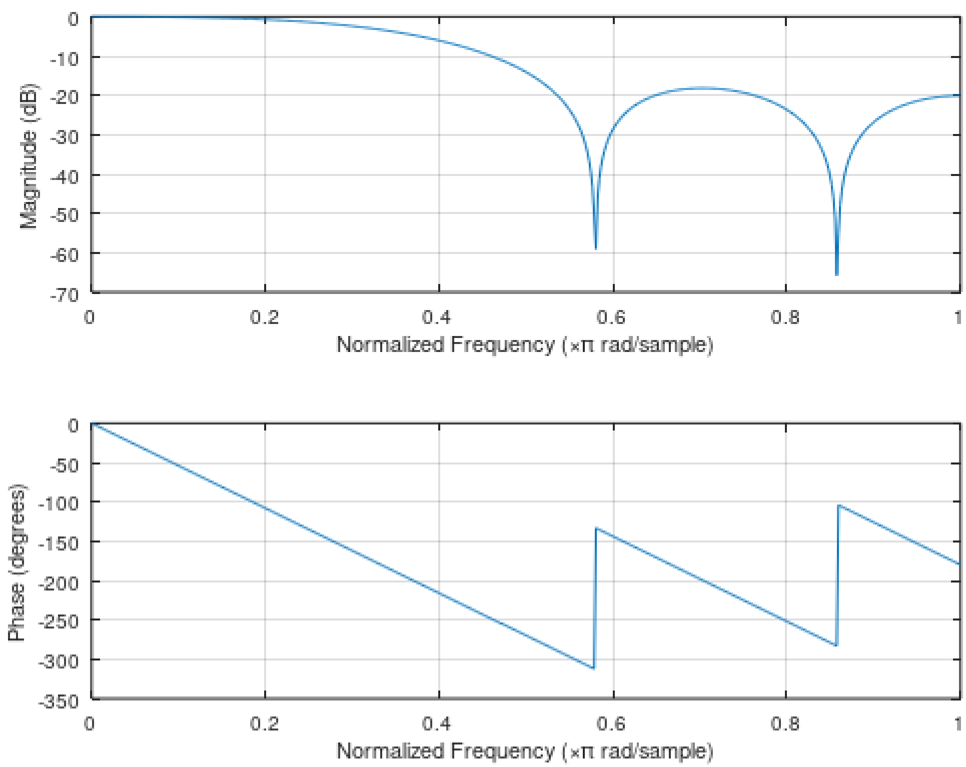

Figure 8: Rectangular window

(b) Hanning window

```
N=6;
wc=0.4*pi;
h=fir1(N,wc/pi,hanning(N+1));
freqz(h);
```

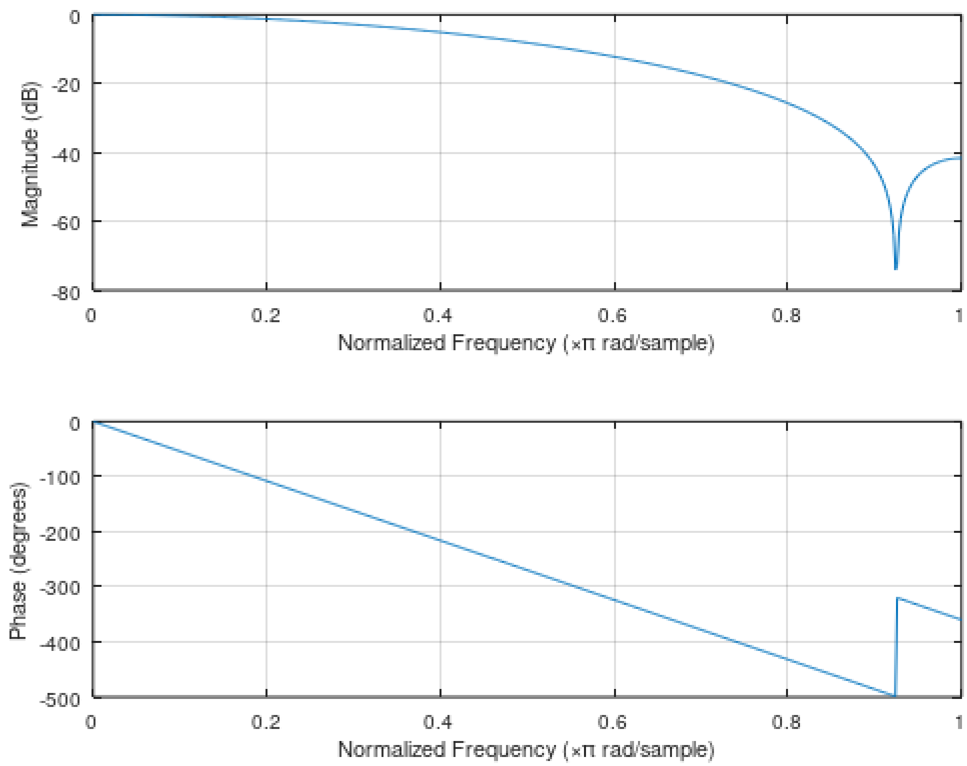

Figure 9: Hanning window

(c) Hamming window

```
N=6;
wc=0.4*pi;
h=fir1(N,wc/pi,hamming(N+1));
freqz(h);
```

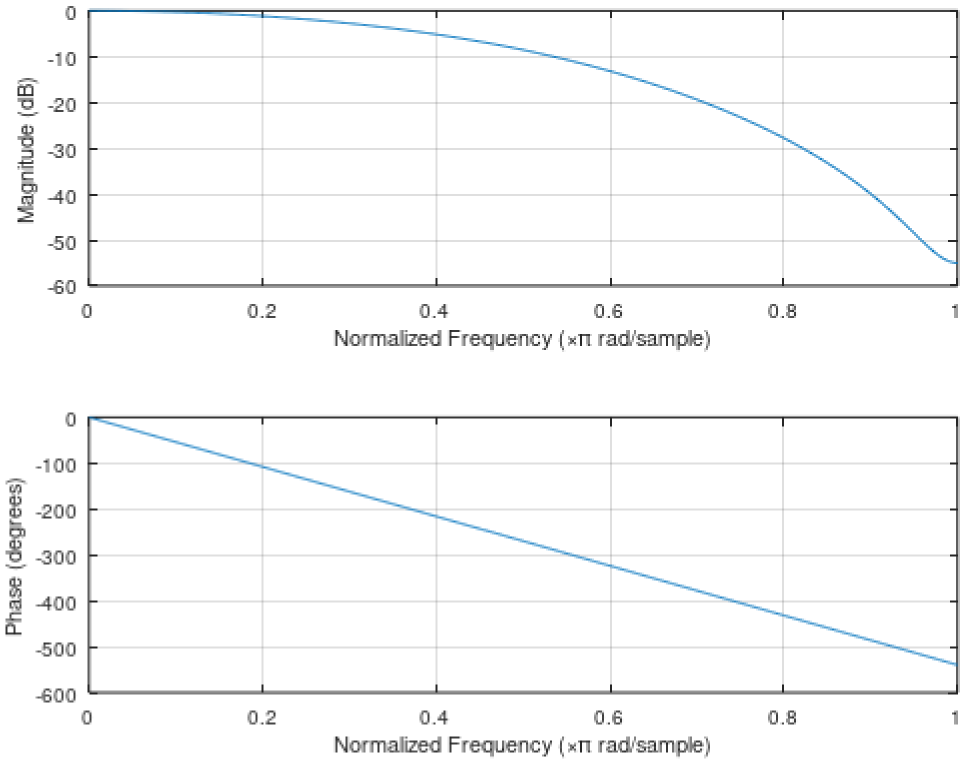

Figure 10: Hamming window

(d) Bartlett window

```
N=6;
wc=0.4*pi;
h=fir1(N,wc/pi,bartlett(N+1));
freqz(h);
```

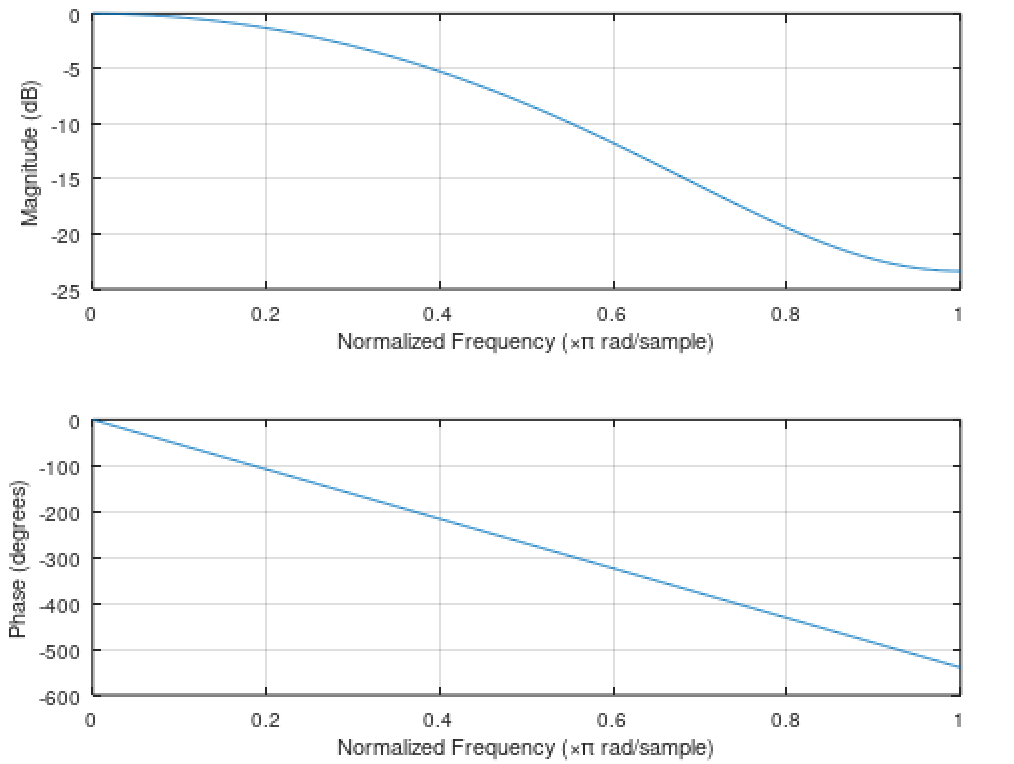

Figure 11: Bartlett window

(e) Blackman window

```
N=6;
wc=0.4*pi;
h=fir1(N,wc/pi,blackman(N+1));
freqz(h);
```

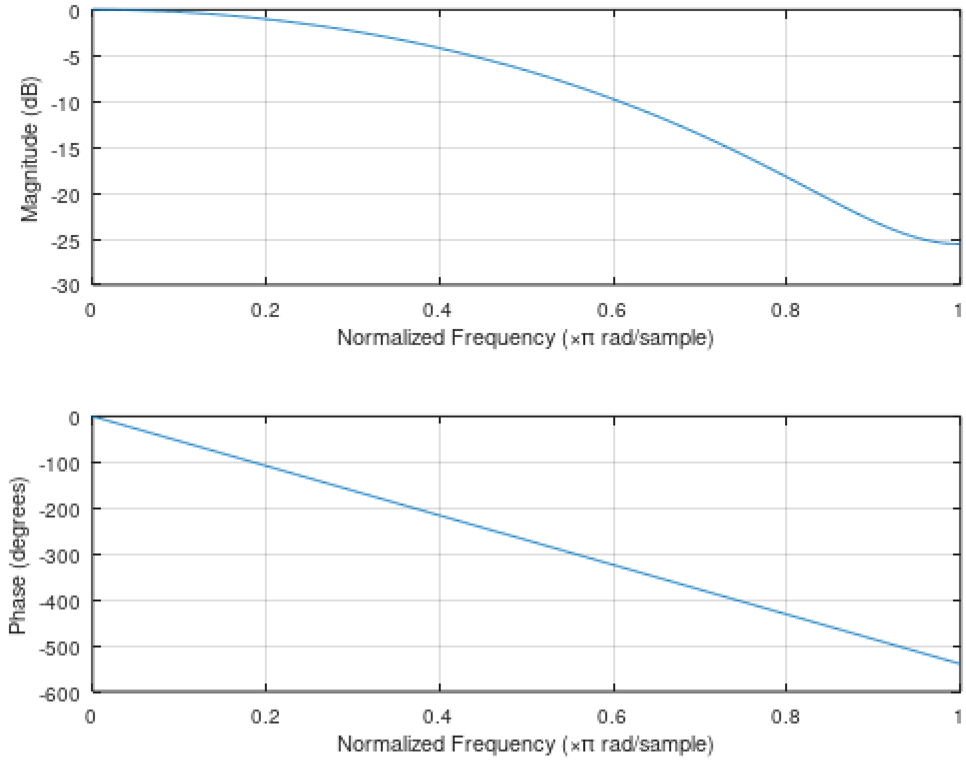

Figure 12: Blackman window

(f) Kaiser window

```
N=6;
wc=0.4*pi;
h=fir1(N,wc/pi,kaiser(N+1));
freqz(h);
```

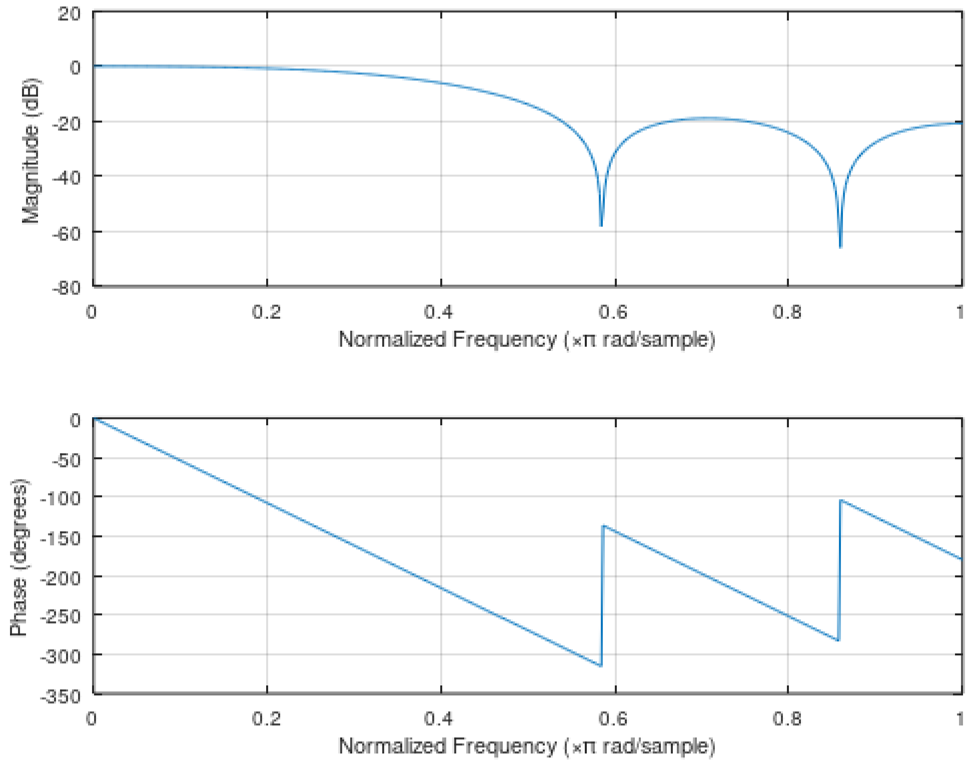

Figure 13: Kaiser window

Conclusion

In this way, "Lab 6 : IIR and FIR Filter Design" was completed with the help of MATLAB. FIR filter was designed using Butterworth, Chebyshev approximation, and IIR filter was designed using windowing method. Different windows like Blackman, Hanning, Hamming, Rectangular, and Bartlett windows were used to design the FIR filter.