**Department of Electronic & Telecommunication Engineering**
**University of Moratuwa**

**Signals and systems** $_{F}IR\&IIRDesign$

# Digital Filter Design Project

Bandara D.M.D.V.
Undergraduate (Biomedical engineering)
Department Electronic and Telecommunications
Faculty of Engineering
University of Moratuwa

# Contents

# 1 Abstract

This report contains a filter design with a unique passband and stopband configured by the university's index number and includes codes and plots obtained. There are two filter designs, FIR and IIR. the windowing method (in conjunction with the Kaiser window) is used, whereas, for IIR filters, the bilinear transformation method is used.

# 2 Filter specification

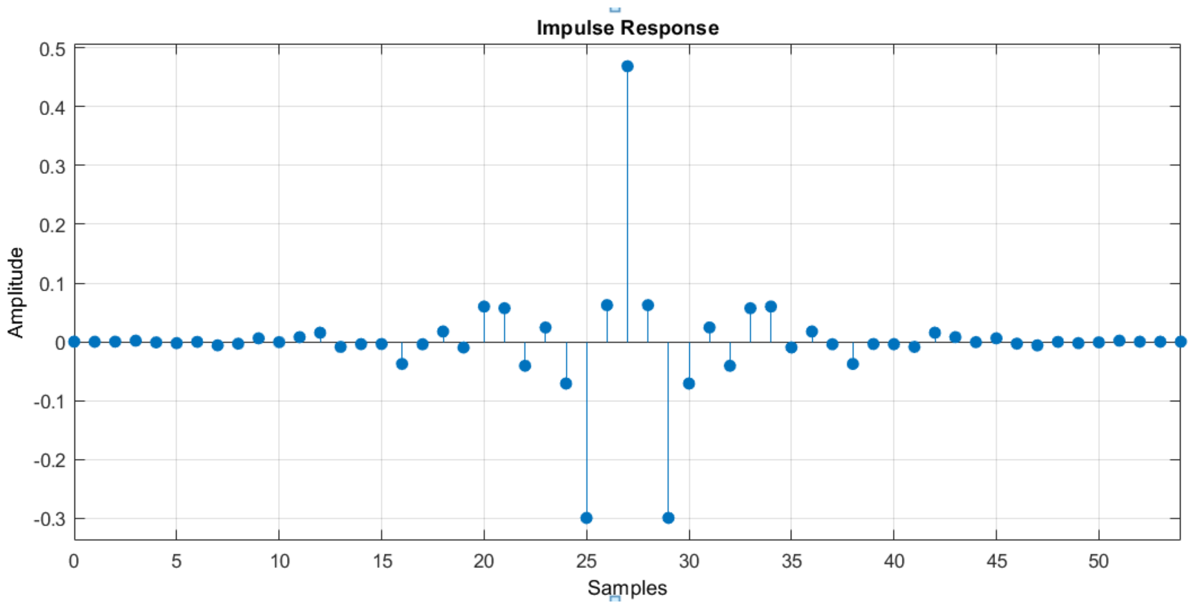
The random 3 digits, A=0, B=6 and C=1

| Parameter | Value |
|---|---|
| Maximum passband ripple, $\tilde{A}_p$ | 0.10 dB |
| Minimum stopband attenuation, $\tilde{A}_a$ | 56 dB |
| Lower passband edge, $\Omega_{p1}$ | 500 $rad/s$ |
| Upper passband edge, $\Omega_{p2}$ | 1000 $rad/s$ |
| Lower stopband edge, $\Omega_{s1}$ | 200 $rad/s$ |
| Upper stopband edge, $\Omega_{s2}$ | 1200 $rad/s$ |
| Sampling frequency, $\Omega_{sm}$ | 3200 $rad/s$ |

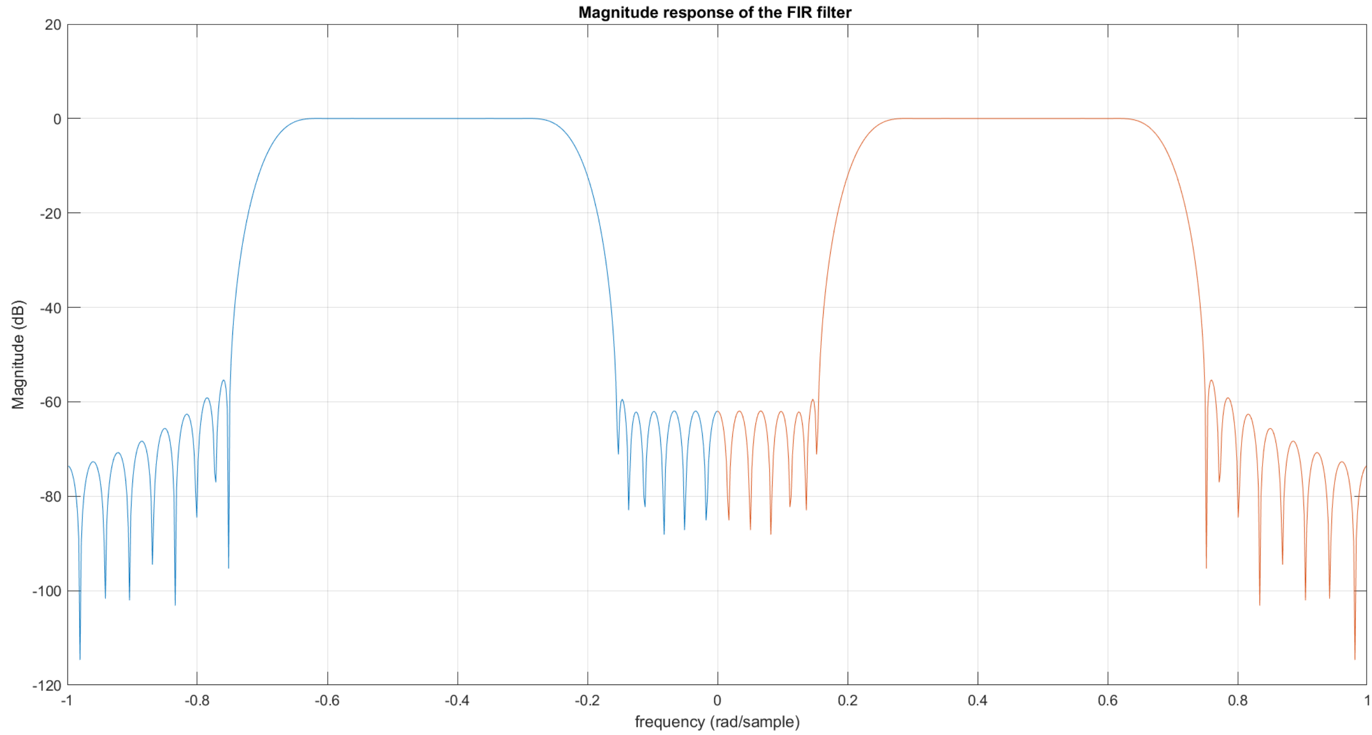
# 3 FIR band-pass digital filter

This particular band-pass filter was designed using kaiser window of order 54.

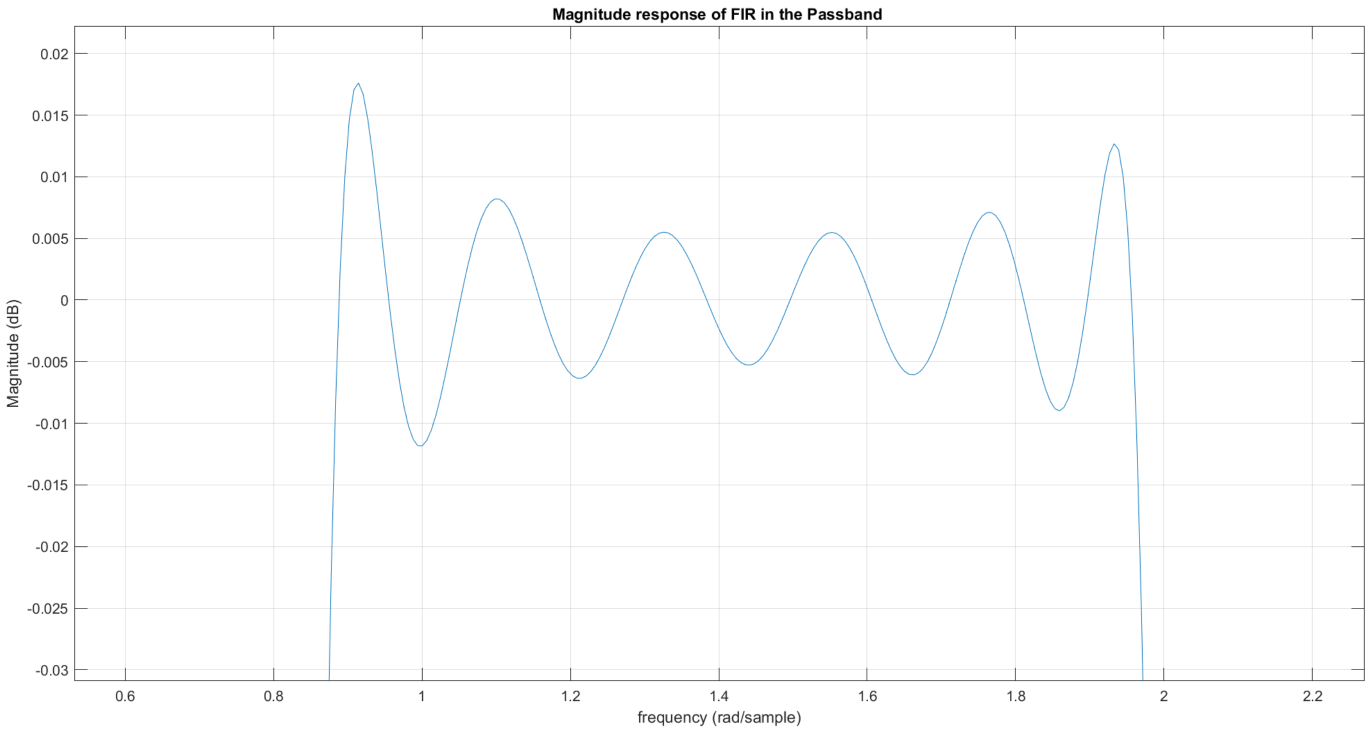
## 3.1 Impulse response



## 3.2 Magnitude response of the digital filter for $-\pi \leq \omega \leq \pi$




## Magnitude response of FIR filter in the passband



# 4 IIR bandpass digital filter

$D/4 = 1$. So the **Chebyshev** approximation method was used.
First, the warping effect is removed by prewarping the frequencies. Then the minimum order of the filter was calculated using **cheb1ord** function. Then, **cheb1ap** function is used to create the desired filter. Finally, we map from S domain to Z domain using the bilinear function and end up with the required digital filter.

## 4.1 Coefficient of the transfer function of the IIR filter

Since this IIR filter is a recursive filter it has the following transfer function.

$$H(z) = \frac{a_0 z^0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + ... + a_N z^{-N}}{b_0 z^0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + ... + b_N z^{-N}}$$

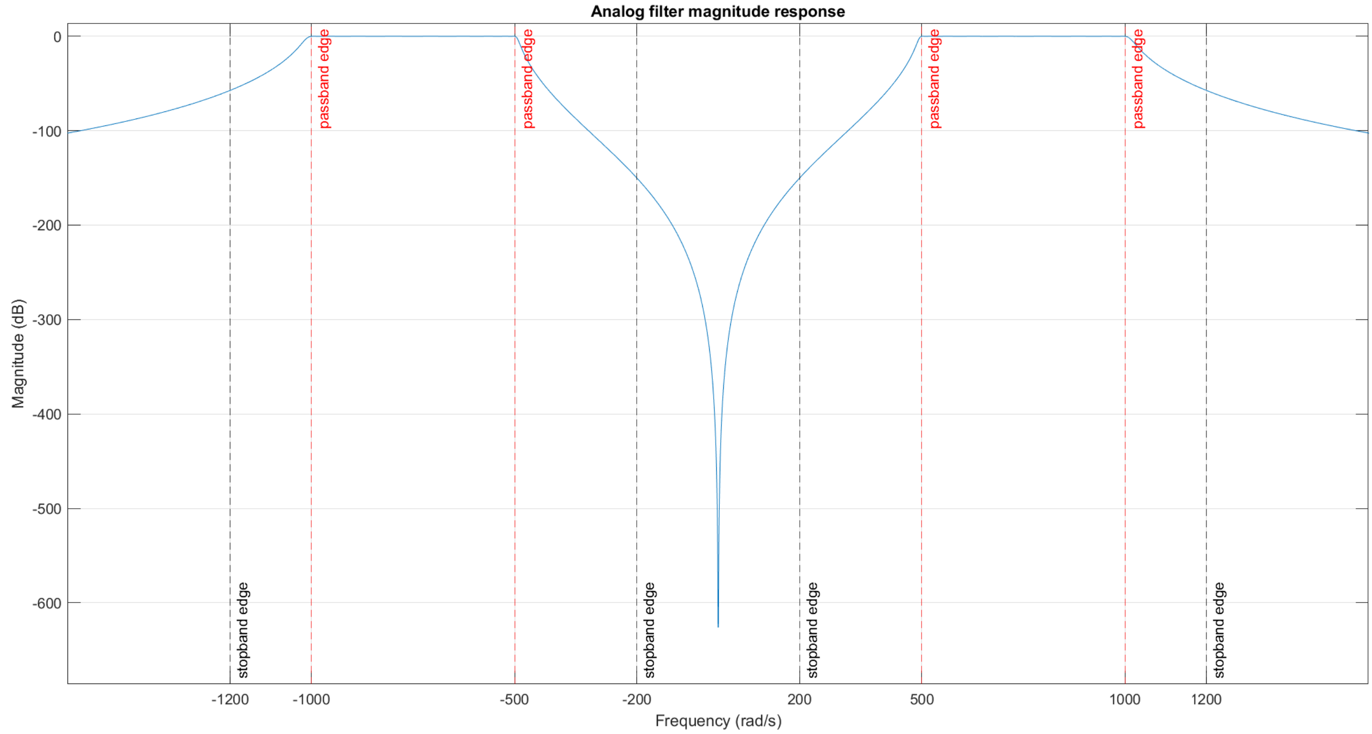
$N$ is the order of the filter.

The designed filter is a filter of order 7 and above transfer function has 15 coefficient in both numerator and denominator.

| $k$ | $a_k$ | $b_k$ |
|---|---|---|
| 0 | $3.5464\times10^{-4}$ | 1 |
| 1 | $-8.6736\times10^{-19}$ | -1.2180 |
| 2 | $-2.4825\times10^{-3}$ | 4.6091 |
| 3 | $2.6888\times10^{-17}$ | -4.5757 |
| 4 | $7.4475\times10^{-3}$ | 9.9134 |
| 5 | $-6.5919\times10^{-17}$ | -8.0621 |
| 6 | $-1.2412\times10^{-2}$ | 12.5905 |
| 7 | $7.8929\times10^{-17}$ | -8.2673 |
| 8 | $1.2412\times10^{-2}$ | 10.1069 |
| 9 | $-6.5919\times10^{-17}$ | -5.1463 |
| 10 | $-7.4474\times10^{-3}$ | 5.0935 |
| 11 | $2.2118\times10^{-17}$ | -1.8338 |
| 12 | $2.4825\times10^{-3}$ | 1.4835 |
| 13 | $-5.3125\times10^{-18}$ | –0.2920 |
| 14 | $-3.5464\times10^{-04}$ | -0.1913 |

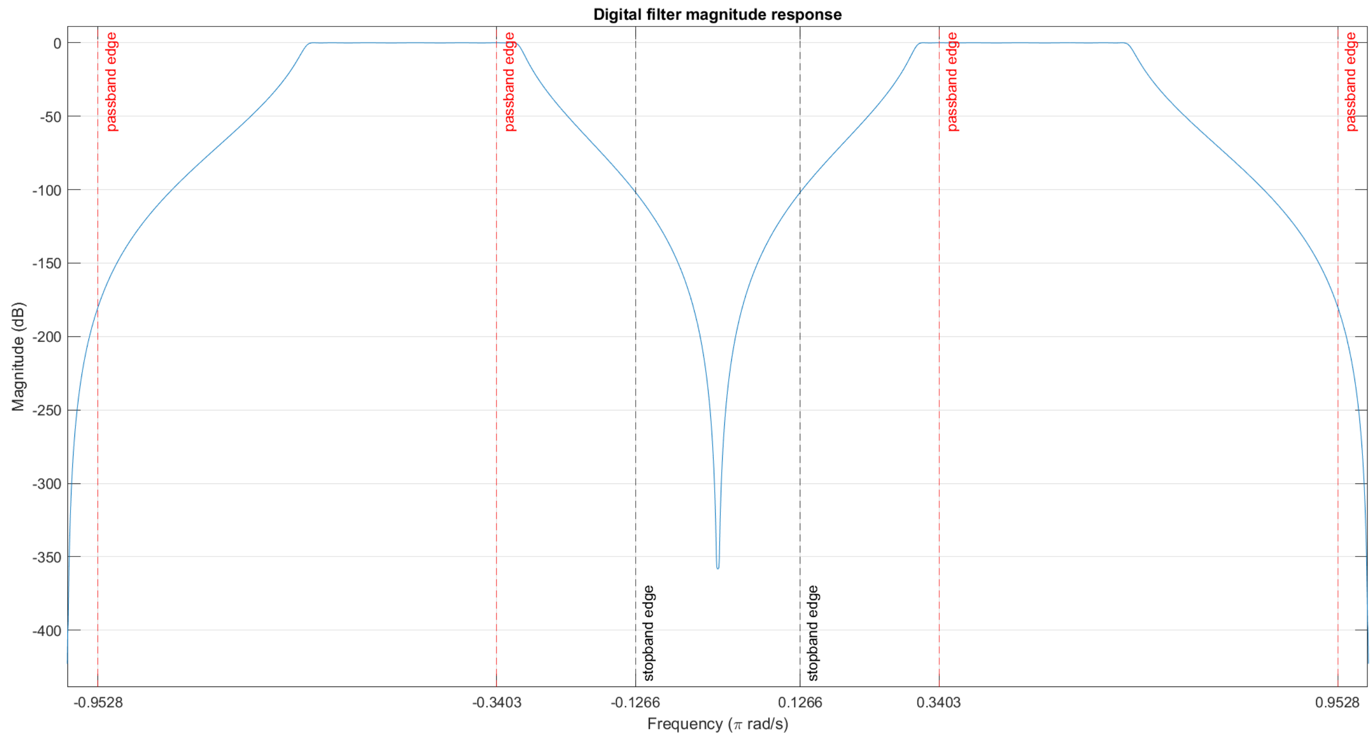
Table 1: Numerator and denominator Coefficients of $Z$ domain transfer function

## 4.2 Magnitude response of the digital filter for $-\pi \leq \omega \leq \pi$

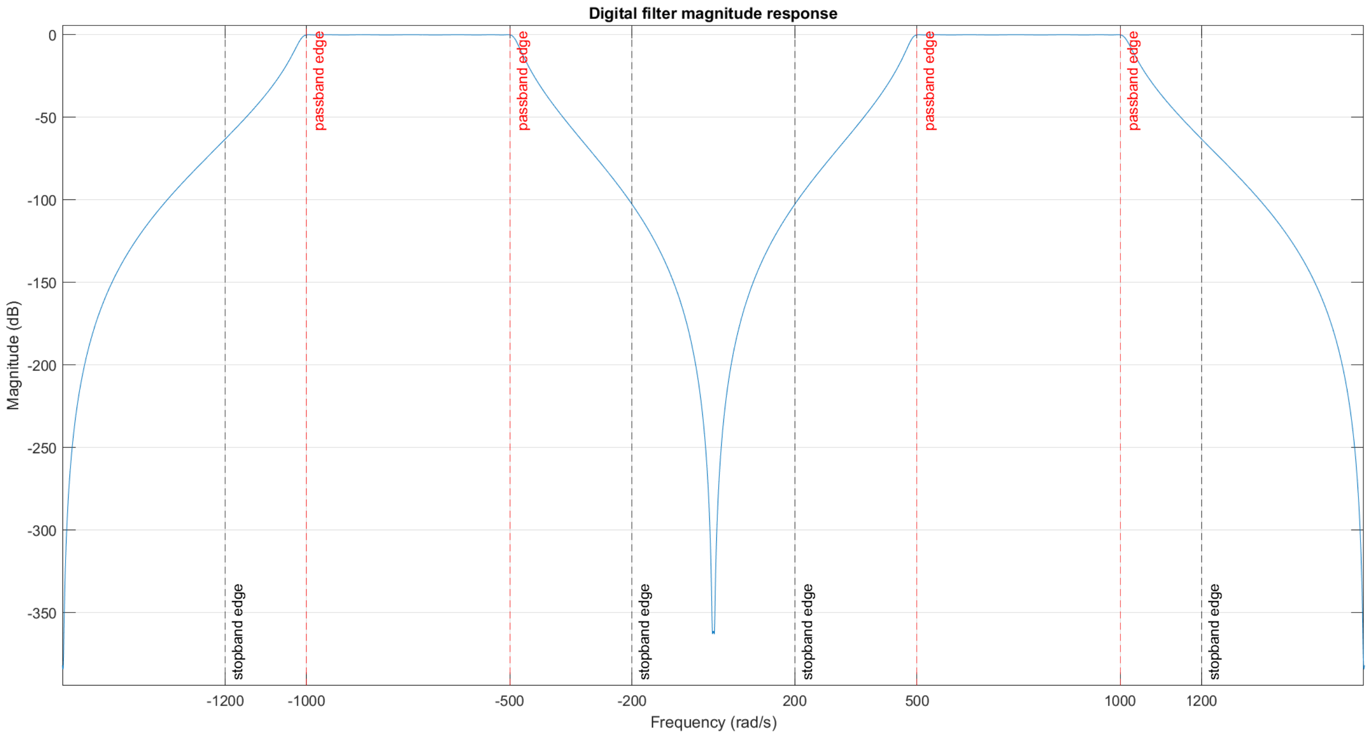



## 4.3 Magnitude response of IIR digital filter in the passband

### With normalizing the frequencies



#### Without normalizing the frequencies




# 5 Comparison between designed FIR and IIR filters

## 5.1 Order

Order of the FIR filter is 54 and 7 in IIR filter.

## 5.2 Processing the samples

### 5.2.1 FIR

FIR is a non-recursive filter.Therefore, its outputs only depend on inputs and given by the following difference equation.

$$y[n] = \sum_{k=0}^{5} 4a_k x[n-k]$$

All the coefficients($a_k$) should be multiply with their corresponding input values, and their sum is taken. As the order($m$) is 54, there should be 55 multipliers and **54 adders** to process a sample using this FIR filter. **However, due to the symmetry** $(a_k = a_{m-k})$ **of the filter only *28 multipliers* are required.**

### 5.2.2 IIR

IIR is a recursive filter and it's outputs given by the following difference equation.

$$y[n] = \sum_{k=0}^{7} a_k x[n-k] + \sum_{k=1}^{7} b_k y[n-k]$$

As mentioned in the table 1, at $k = 0$, $b_k = 1$ and $z^0 = 1$. Therefore, to process a sample using this IIR filter there should be **15 multipliers** and **15 adders**.

# 6 Appendix

## MATLAB code

```
%Filter Specifications
A=0; B=6; C=1;

%% Specifications
Ap      = 0.1 + (0.01*A);        %Maximum passband ripple
As      = 50 + B;                %Minimum stopband attenuation
W_p1    = (C * 100) + 400;
W_p2    = (C * 100) + 900;
W_p     = [W_p1 , W_p2 ];        %Passband
W_s1    = (C * 100) + 100;
W_s2    = (C * 100) + 1100;
W_s     = [W_s1 , W_s2 ];        %Stopband
W_sm    = 2*((C * 100) + 1500); %Sampling frequency
Tsm     = 2*pi/W_sm;

%% Digital FIR FILTER
delta_p = (10.^(0.05*Ap)-1)/(10.^(0.05*Ap)+1); %From lecture notes
delta_s = 10.^(-0.05*As);
desired_freq_ranges = [W_s1 W_p1 W_p2 W_s2];
amplitude = [0 1 0];
deviation = [delta_s delta_p delta_s];

[n,Wn,beta,ftype] = kaiserord(desired_freq_ranges,amplitude,deviation
    ,W_sm);
bandpass = fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');
fv = fvtool(bandpass);

% Magnitude response in range {-pi,pi]
figure(2);
[h, f] = freqz(bandpass);
plot([-f/pi f/pi],mag2db(abs(h)));
title("Magnitude response of the FIR filter ")
xlabel("frequency (rad/sample)");
ylabel("Magnitude (dB)");
grid on;

% Kaiser Window
figure(3);
k_w = kaiser(n,beta);
stem(k_w)
title("Kaiser Window")
xlabel("Sample(n)")
grid on;

% Magnitude response in between [Wp1,Wp2]
```

```
figure(4);
plot(f, mag2db(abs(h)))
xlim ([ W_p1 W_p2 ]*(2* pi / W_sm ) )
title("Magnitude response of FIR in the Passband ")
xlabel("frequency (rad/sample)");
ylabel("Magnitude (dB)");
grid on;

%% IIR filter design
%% Analog Chebyshev Filter design

[n,Wc] = cheb1ord(W_p,W_s,Ap,As,'s'); %minimum order and cut off
[b,a] = cheby1(n,Ap,W_p,'s'); %2*n bandpass analog filter
W=linspace(-1600,1600,3200);

H = freqs(b,a,W); %returns the complex frequency response of the analog filter specified by the coefficient vectors b and a, evaluated at the angular frequencies W
%Plotting magnitude response in dB

figure(5);
plot(W,mag2db(abs(H)));
title('Analog filter magnitude response in dB');
xlim([-1600 1600]);
xline([W_p,-1*W_p],'--','Color',[1,0,0],'Label','passband edge');
xline([W_s,-1*W_s],'--','Color',[0,0,0],'Label','stopband edge','LabelVerticalAlignment','bottom');
xticks([-W_s2,-W_p2,-W_p1,-W_s1,W_s1,W_p1,W_p2,W_s2]);
grid on;
xlabel('Frequency (rad/s)');
ylabel('Magnitude (dB)');

%% Prewrapping
W_p=[(2/T_sm*tan(W_p1*T_sm/2))/(W_sm/2) (2/T_sm*tan(W_p2*T_sm/2))/(W_sm/2)];
W_s=[(2/T_sm*tan(W_s1*T_sm/2))/(W_sm/2) (2/T_sm*tan(W_s2*T_sm/2))/(W_sm/2)];

[n,Wc] = cheb1ord(W_p,W_s,Ap,As,'s');
[z,p,k] = cheb1ap(n,Ap);
[A,B,C,D] = zp2ss(z,p,k); %state-space form
[At,Bt,Ct,Dt] = lp2bp(A,B,C,D,sqrt(W_p(1)*W_p(2)),W_p(2)-W_p(1)); %continous time
[Ad,Bd,Cd,Dd] = bilinear(At,Bt,Ct,Dt,1/pi); %discrete-time

W=linspace(-1600/(W_sm/2),1600/(W_sm/2),3200);
% determines the numerator and denominator of the transfer function
[b,a] = sos2tf(ss2sos(Ad,Bd,Cd,Dd));
sprintf('nuerator = %d \n',b)
```

```
sprintf('denominator = %d \n',a)
hd = freqz(b,a,W,2); %frequency response

%Plotting magnitude response in dB
figure(6);
plot(W,mag2db(abs(hd)));
title('Digital filter magnitude response in dB');
xlim([-1600/(W_sm/2) 1600/(W_sm/2)]);
xline([W_p,-1*W_p],'--','Color',[1,0,0],'Label','passband edge');
xline([W_s,-1*W_s],'--','Color',[0,0,0],'Label','stopband edge','LabelVerticalAlignment','bottom');
xticks([-W_s(2),-W_p(2),-W_p(1),-W_s(1),W_s(1),W_p(1),W_p(2),W_s(2)]);\
grid on;
xlabel('Frequency (\pi rad/s)');
ylabel('Magnitude (dB)');

%% Prewrapping without normalizing
W_p = [2/T_sm*tan(W_p1*T_sm/2) 2/T_sm*tan(W_p2*T_sm/2)];
W_s = [2/T_sm*tan(W_s1*T_sm/2) 2/T_sm*tan(W_s2*T_sm/2)];

[n,Wc] = cheb1ord(W_p,W_s,Ap,As,'s');
[z,p,k] = cheb1ap(n,Ap);
[A,B,C,D] = zp2ss(z,p,k); %state-space form
[At,Bt,Ct,Dt] = lp2bp(A,B,C,D,sqrt(W_p(1)*W_p(2)),W_p(2)-W_p(1)); %continous time
[Ad,Bd,Cd,Dd] = bilinear(At,Bt,Ct,Dt,W_sm/(2*pi)); %discrete-time

W=linspace(-1600,1600,3200);
[b,a] = sos2tf(ss2sos(Ad,Bd,Cd,Dd));
[z,p,k] = tf2zp(b,a);
filter2 = freqz(b,a,W,W_sm); %magnitude response

W_p = [W_p1 , W_p2 ];
W_s = [W_s1 , W_s2 ];
%Plotting magnitude response  in dB
figure(7);
plot(W,mag2db(abs(filter2)));
title('Digital filter magnitude response in dB');
xlim([-1600 1600]);
xline([W_p,-1*W_p],'--','Color',[1,0,0],'Label','passband edge');
xline([W_s,-1*W_s],'--','Color',[0,0,0],'Label','stopband edge','LabelVerticalAlignment','bottom');
xticks([-W_s(2),-W_p(2),-W_p(1),-W_s(1),W_s(1),W_p(1),W_p(2),W_s(2)])
grid on;
xlabel('Frequency (rad/s)');
ylabel('Magnitude (dB)');
```