EN2063 – SIGNALS AND SYSTEMS

FIR and IIR Filter Design Project

Index no: 200398D

Abstract

This report comprise the complete process of designing a FIR and IIR Digital filters for required specifications. For the design of FIR filter, windowing method (in collaborate with the Kaiser window) is used while for IIR filters, bilinear transformation method is used. For the programming of the design and visualizations of results MATLAB R2021b is used.

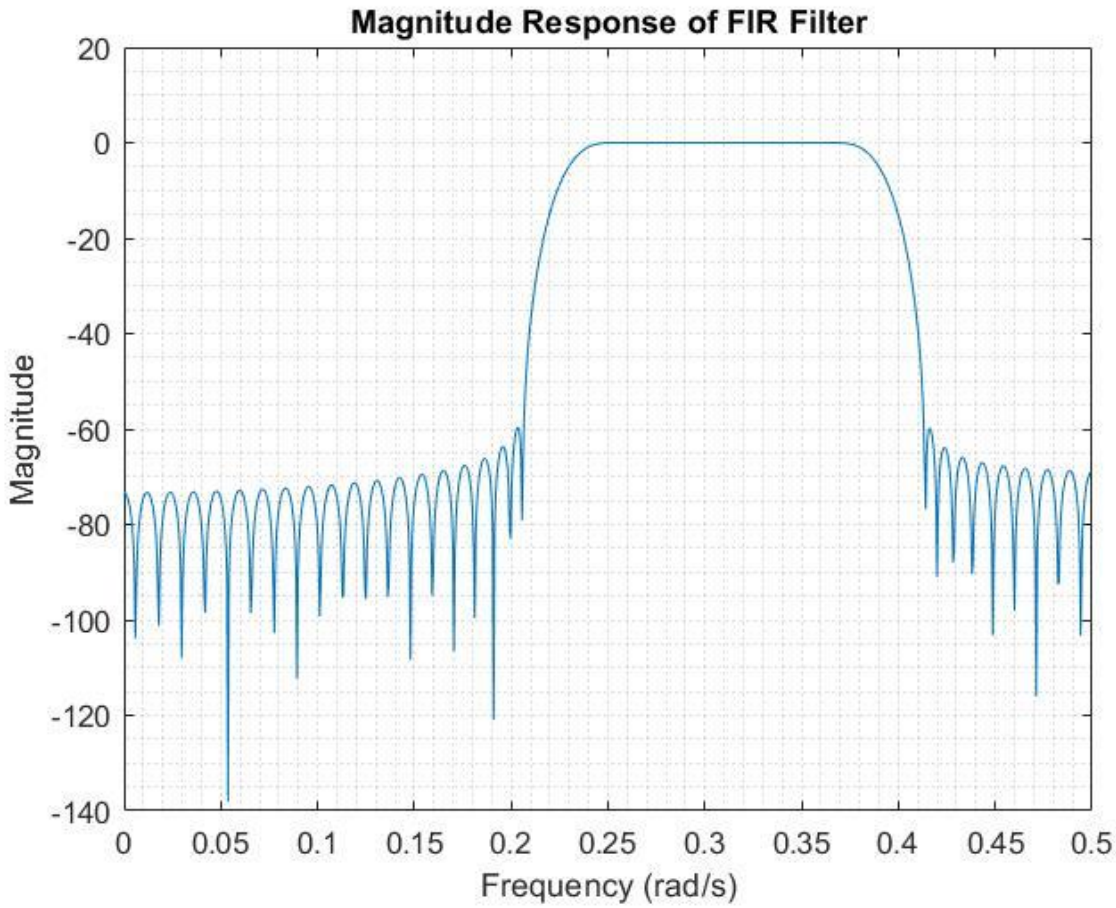
Filter specifications

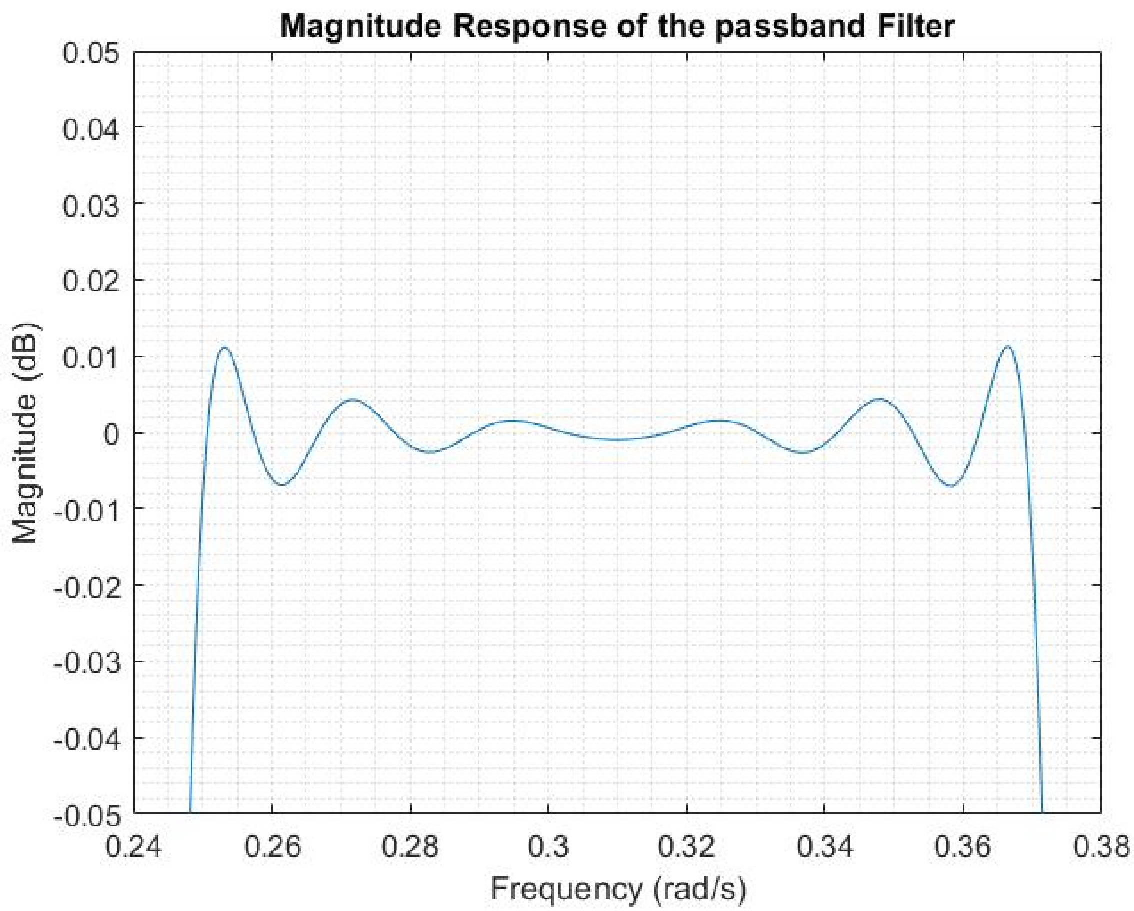
Parameter	Notations	Value
Maximum passband ripple	$\tilde{A}_p$	0.13 dB
Minimum stopband attenuation	$\tilde{A}_s$	59 dB
Lower passband edge	Ω_{p1}	1200 rad/s
Upper passband edge	Ω_{p2}	1700 rad/s
Lower stopband edge	Ω_{s1}	900 rad/s
Upper stopband edge	Ω_{s2}	1900 rad/s
Sampling frequency	Ω_{sm}	4600 rad/s

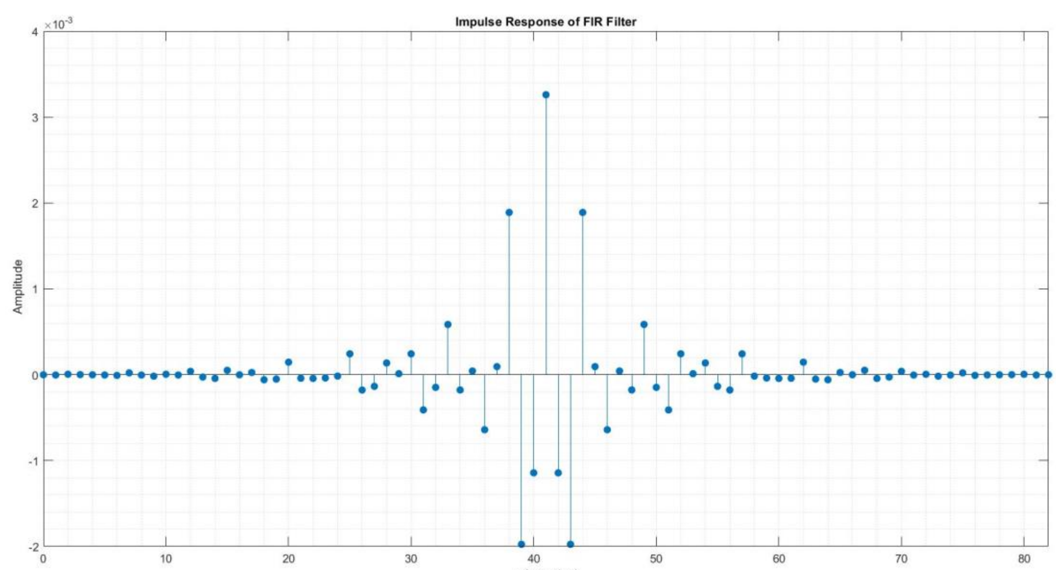
Design procedures of the FIR filters

- FIR bandpass digital filter designing using windowing method in conjunction with the Kaiser window using MATLAB
  1. Defining filter specifications.
  2. Determining the order (n) of the filter, edge frequencies and shape factor beta that specify a Kaiser window using MATLAB code **Kaiserord**.
  3. Designing an n^{th} order FIR bandpass filter using the Kaiser window using MATLAB code **fir1**.
  4. Kaiser window is establish using **kaiser**
  5. Formation of the complex frequency response vector related to the digital filter to plot the responses, using MATLAB code **freqz.**
  6. Generating the impulse response vector corresponding to the filter to plot the response, using MATLAB code **impz.**
  7. Plotting the impulse response of the FIR filter.

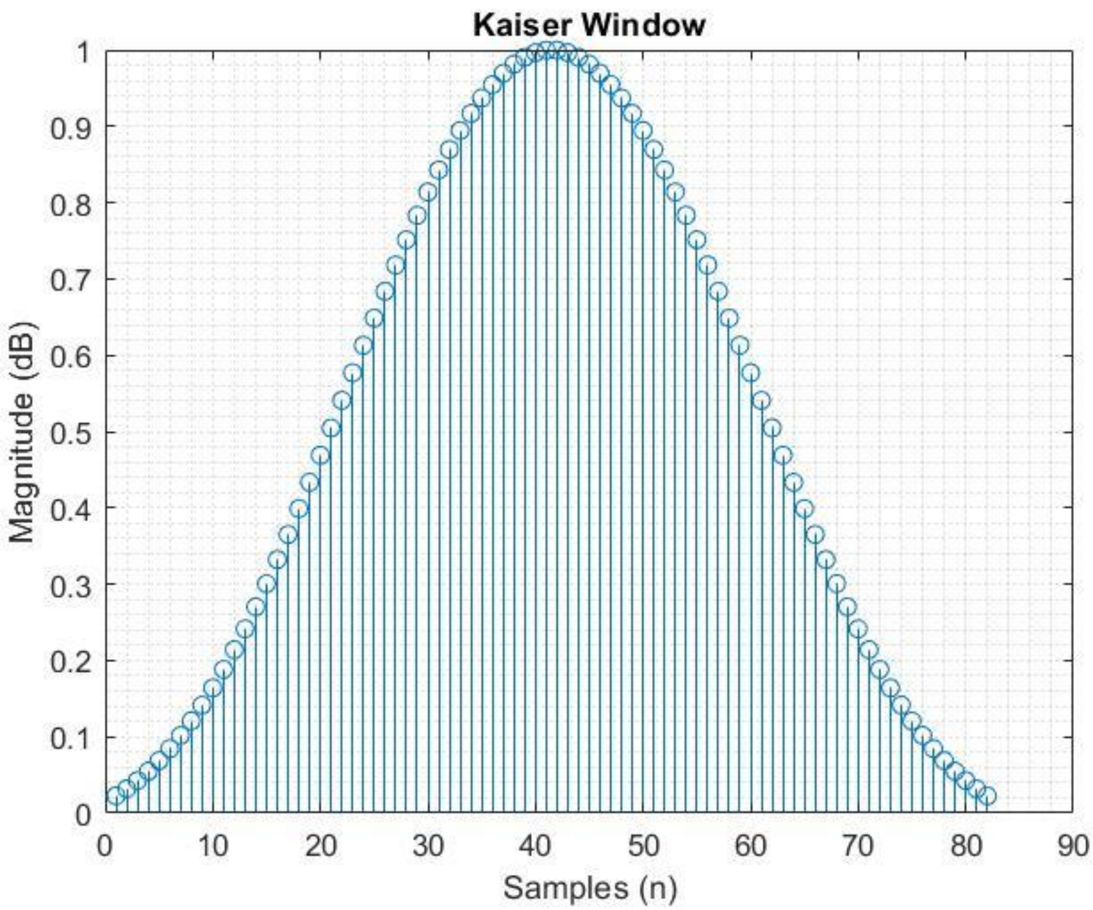
Plotted Graphs







Impulse response of the FIR Filter


Kaiser Window
Magnitude (dB)
Samples (n)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90

Design Procedures of the IIR Filters

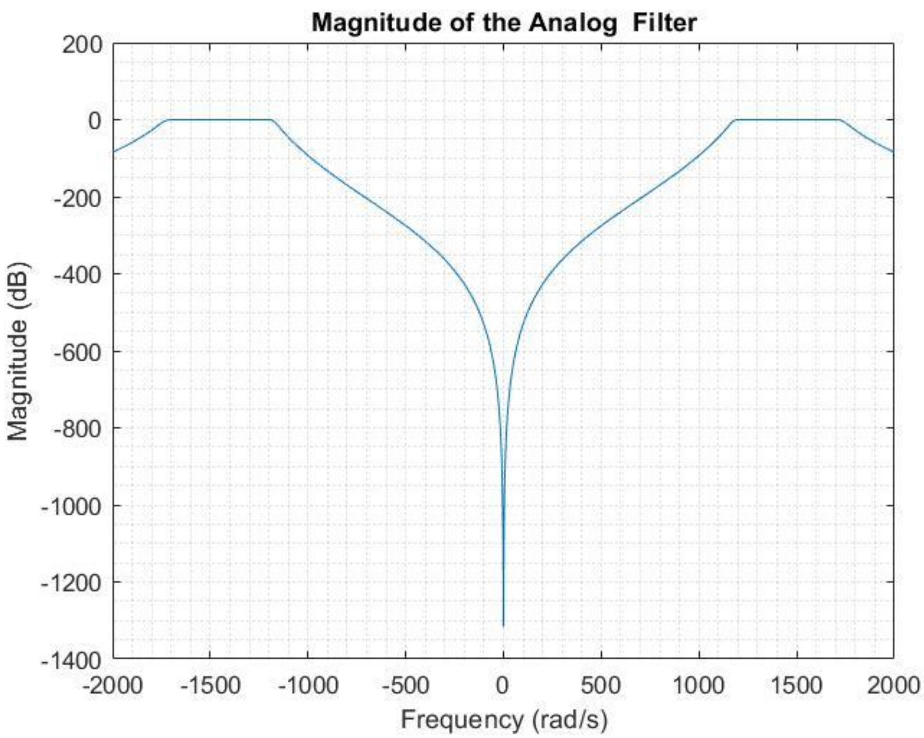
IIR bandpass digital filter (Butterworth) designing using bilinear transformation method using MATLAB.

1. Defining the specifications.
2. Prewarping the critical frequencies using **prewarp**
3. Designing analog Butterworth filter. For that, order is defined by **buttord**
4. Implementing lowpass filter prototype using the **buttap.**
5. Then conversion of the prototype to state space form using **zp2ss.**
6. Transformation of low pass filter into bandpass filter using **1p2bp.**
7. Tramsformation of the S domain into the Z domain by **bilinear.**
8. Finally plotting the graphs

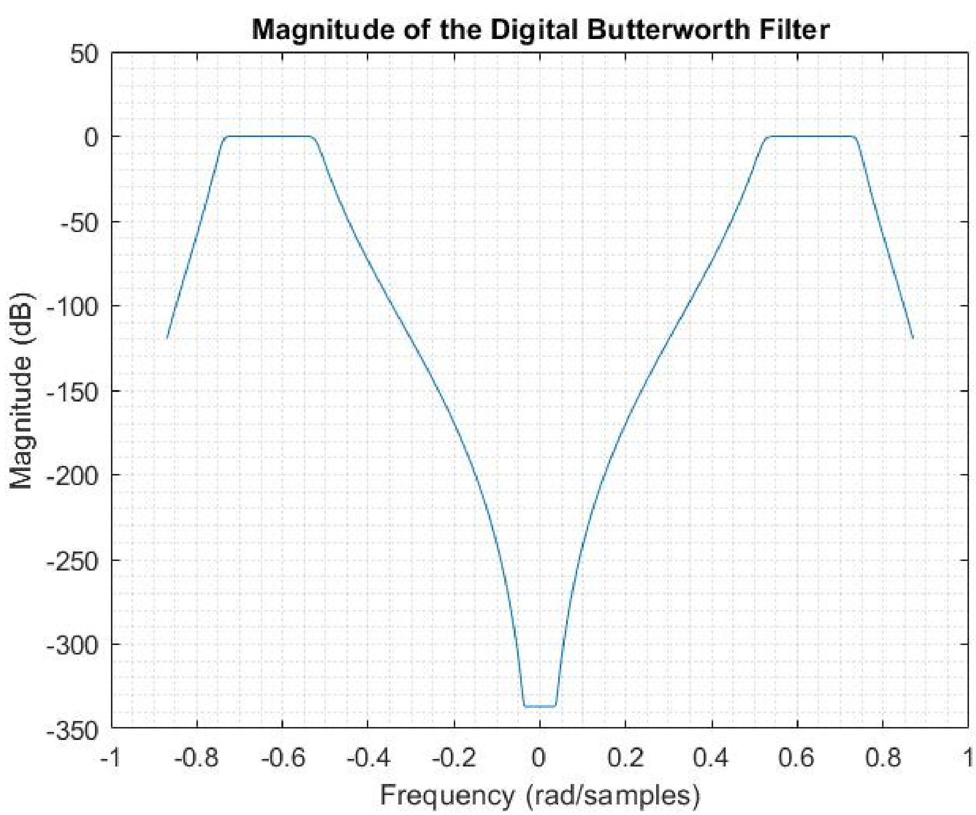
Coefficients of the Transfer function

Numerator	Denominator
9.73E-07	1
9.00E-17	7.274843867
-1.07E-05	30.38213864
1.15E-15	90.0473206
5.35E-05	209.2505654
5.11E-15	399.5753225
-0.000160606	646.5323106
1.17E-14	902.9115609
0.000321212	1102.704055
1.61E-14	1187.358498
-0.000449697	1133.589799
1.41E-14	962.2255706
0.000449697	726.9318775
8.07E-15	488.1126342
-0.000321212	290.4227898
3.01E-15	152.2125192
0.000160606	69.68332065
7.08E-16	27.4953807
-5.35E-05	9.179903654
9.67E-17	2.514761219
1.07E-05	0.539837738
5.60E-18	0.082216748
-9.73E-07	0.007257212

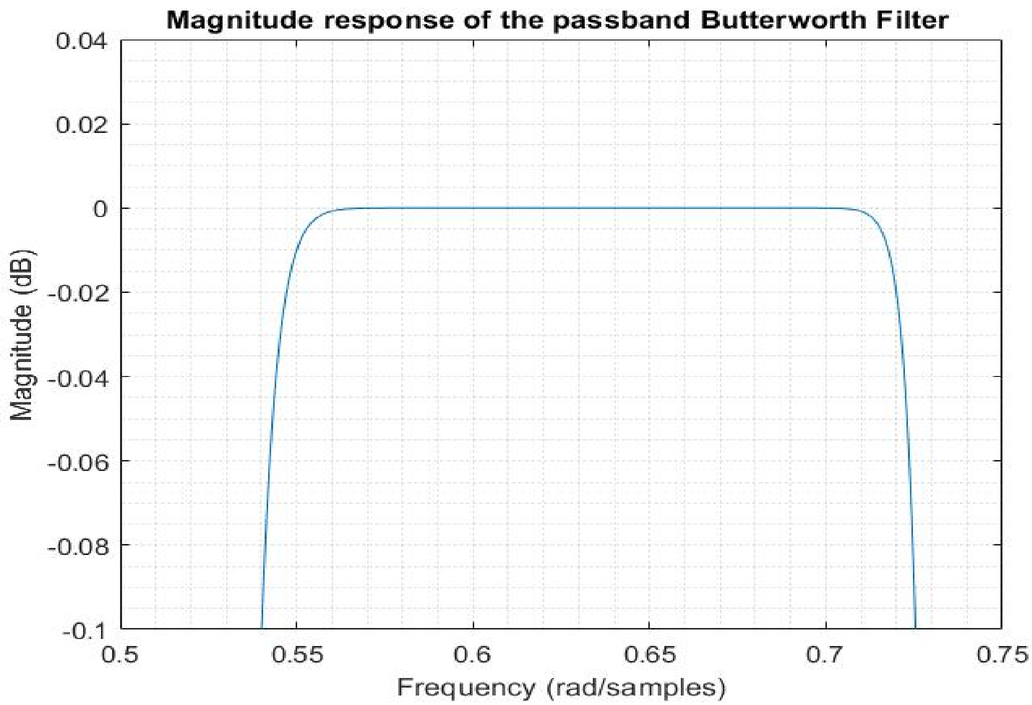
Graphs



Magnitude of the Analog Filter
Magnitude (dB)
Frequency (rad/s)
200
0
-200
-400
-600
-800
-1000
-1200
-1400
-2000
-1500
-1000
-500
0
500
1000
1500
2000



Magnitude of the Digital Butterworth Filter
Magnitude (dB)
Frequency (rad/samples)
50
0
-50
-100
-150
-200
-250
-300
-350
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1



Comparing orders and no. of multiplications and additions performed in above filters

	Order	No.of Multiplications	No. of Additions
FIR	82	83	82
IIR	22	46	44

Appendix

MATLAB code for FIR filter

```
%% Specifications of the Filter
sampling_frequency = 4600;
passband_edges = [900 1200 1700 1900];
magnitudes = [0 1 0];
deviations = [db2mag(-59) db2mag(0.13) db2mag(-59)];

%% Designing the Kaiser window
[n,Wn,beta,ftype] =
kaiserord(passband_edges,magnitudes,deviations,sampling_frequency);
n
n = n + rem(n,2);
my_filter = fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');

%% Magnitude Response of the Filter  plotting

figure;
[H,f] = freqz(my_filter,1,1024,sampling_frequency);
f1=f/4600;
plot(f1,(20*log10(abs(H))))
title('Magnitude Response of the Filter')
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
grid("minor")

%% Kaiser Window
figure;
w = kaiser(n,beta);
stem(w)
title('Kaiser Window')
xlabel('Samples (n)')
ylabel('Magnitude (dB)')
grid("minor")

%% Impulse response of the filter  plotting
figure;
impz(my_filter,100)
title('Impulse Response of the Filter')
xlabel('Samples (n)')
ylabel('Amplitude')
grid("minor")

%% Passband Attenuation response plotting
figure;
[H,f] = freqz(my_filter,1,1024,sampling_frequency);
plot(f1,(mag2db(abs(H))))
axis ([ 0.24 , 0.38 , -0.05 , 0.05]);
title('Magnitude Response of the passband Filter')
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
grid ("minor")
```

MATLAB code for IIR filter

```
%% Filter Specifications Wp Ws Rp Rs
Wp = [1200 1700];
Ws = [900 1900];
Rp = 0.13;
Rs = 59;

%% Designing the Analog Filter Butterworth
[n,Wn] = buttord(Wp,Ws,Rp,Rs,"s");

[b,a]=butter(n,Wn,"s");
filter = tf(b,a);
Wsm=4600;
fsm=2*pi/Wsm;      % sampling frequency


%% Magnitude Response of the Analog Filter
W=linspace(-2000,2000,4000);
H = freqs(b,a,W);
magnitude = abs(H);

figure(1)
plot(W,mag2db(magnitude))
title("Magnitude of the Analog Filter")
xlabel("Frequency (rad/s)")
ylabel("Magnitude (dB)")
grid("minor")
%% Prewrapping frequencies

Wp(1)=2/fsm*tan(Wp(1)*fsm/2); Wp(2)=2/fsm*tan(Wp(2)*fsm/2);
Ws(1)=2/fsm*tan(Ws(1)*fsm/2); Ws(2)=2/fsm*tan(Ws(2)*fsm/2);

%% normalizing frequencies
Wp=[Wp(1)/(Wsm/2) Wp(2)/(Wsm/2)];
Ws=[Ws(1)/(Wsm/2) Ws(2)/(Wsm/2)];
%% Transforming into Digital filter Butterworth
[n,Wc] = buttord(Wp,Ws,Rp,Rs,'s');
n
[z,p,k] = buttap(n);
[A,B,C,D] = zp2ss(z,p,k);
[At,Bt,Ct,Dt] = lp2bp(A,B,C,D,sqrt(Wp(1)*Wp(2)),Wp(2)-Wp(1));  %converting lowpass to
bandpass

W=linspace(-2000/(Wsm/2),2000/(Wsm/2),4000);
[Ad,Bd,Cd,Dd] = bilinear(At,Bt,Ct,Dt,1/pi); % Bilinear Transformation
filter = ss2sos(Ad,Bd,Cd,Dd);
[b,a] = sos2tf(filter);
filter = tf(b,a);               % getting Coefficients of the transfer function
[numerator,denominator]= tfdata(filter);
filter;
[hd,f] = freqz(b,a,W,2);

%% magnitude response of the Digital Filter
magd = abs(hd);
```

```
Wp=[1200/(Wsm/2) 1700/(Wsm/2)];
Ws=[900/(Wsm/2) 1900/(Wsm/2)];
figure(2)
plot(W,mag2db(magd))
title("Magnitude of the Digital Butterworth Filter")
xlabel("Frequency (rad/samples)")
ylabel("Magnitude (dB)")
grid("minor")

%% Magnitude response of the passband
figure(3)
plot(W,mag2db(magd))
axis ([ 0.5 , 0.75 , -0.1 , 0.04]);
title("Magnitude response of the passband Filter ")
xlabel("Frequency (rad/samples)")
ylabel("Magnitude (dB)")
grid("minor")
```