EN 2063 Signals and Systems – Digital Signal Processing

FIR and IIR Filter Design

Abstract

This report consists about the whole process of designing a non-recursive or Finite Impulse Response (FIR) bandpass filter and a recursive Infinite Impulse Response (IIR) Bandpass filter. We have used the windowing method in conjunction with the Kaiser window to design the FIR filter and using a bilinear transform to design the IIR filter. MATLAB R2021b is used to program and visualize the design and its results. Filter design, visualization of Kaiser window, visualization of impulse response, magnitude response of the FIR and IIR filters, Magnitude response in the passband of the FIR and IIR filters and methodology have been clearly stated in this report. And while designing the filter no hardcoded codes are used (only predefined functions available in the MATLAB is used)

An Introduction to FIR and IIR Filters

Digital filters are discrete system designed for filtering objectives by a mathematical algorithm implemented in software or/and hardware. Digital filter can be grouped as recursive and non-recursive filters as well as IIR (Infinite Impulse response) and FIR (Finite Impulse Response) filters. There are two ways of designing non-recursive filters,

- Fourier Series method (Window method).
- Weighted Chebyshev method.

Windowing method for digital filters is fast and convenient method to be implemented in FIR filters. There are variety of window functions can be implemented in digital filters.

- Rectangular
- Von Hann
- Blackman
- Hamming
- Kaiser

As well as for the IIR filter design It can be implemented using closed form indirect method. Such like using

- Using Impulse Invariance method
- Using Bilinear Transform

Here because of the limitations of the impulse invariance method we cannot design band unlimited signals such as High pass filter. There fore we can use bilinear transformation to transform the impulse response in S domain into Z domain. However warping issues eventually occur in critical frequencies to get rid of the warping effect we use a particular method of pre-warping the frequencies.

This report gives a brief design explanation about FIR bandpass filter using windowing method in conjunction with Kaiser window And Design explanation about IIR bandpass filter using bilinear transforming methods. All the software implementations of FIR filters are done in MATLAB using predefined codes. No other hardcoded codes are implemented.

Filter Specifications

Specifications for the filter parameters are selected based on the index number as suggested. The specifications of FIR and IIR bandpass filter are

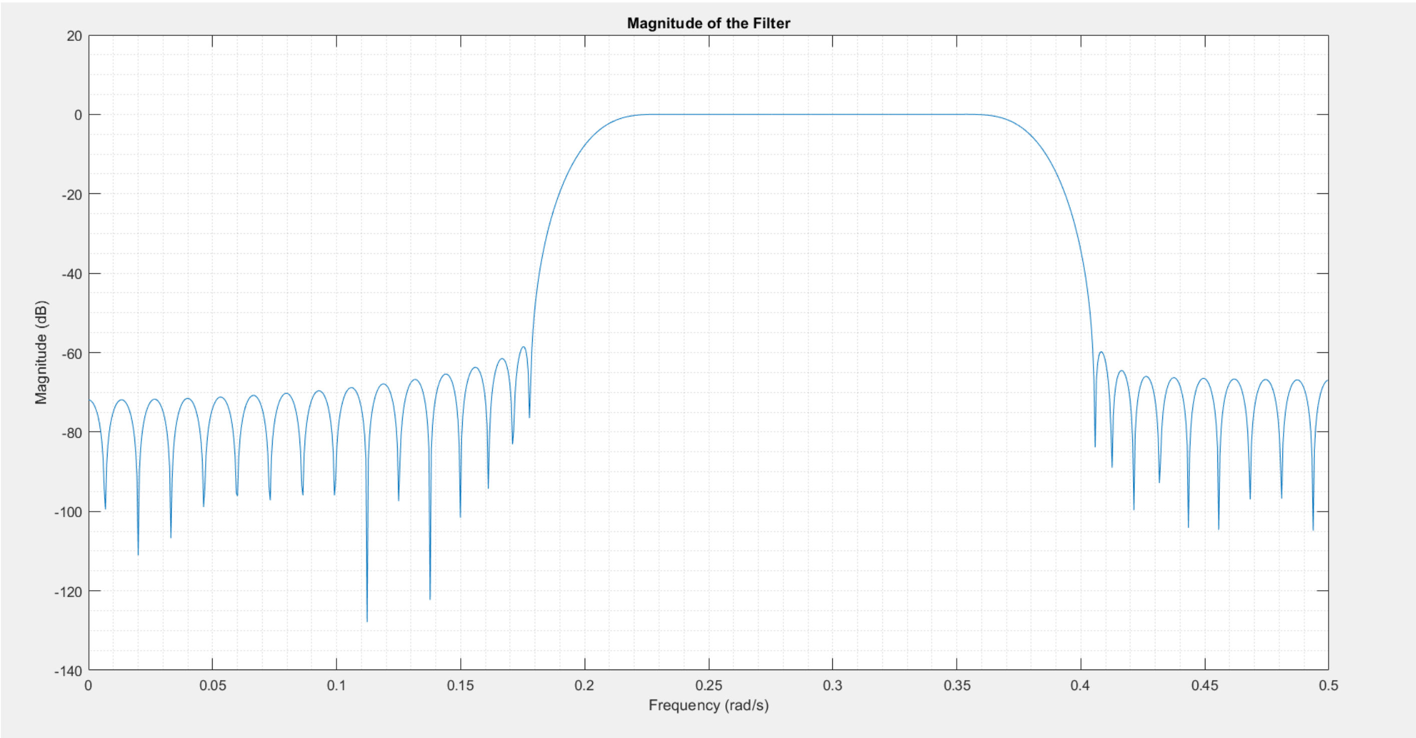
Specifications	*Symbols*	*Value*
Maximum pass band Ripple	A_p	0.16 dB
Maximum Stop Band Attenuation	A_a	-58 dB
Lower Passband edge	Ω_{p_1}	1000 rad/s
Upper Passband edge	Ω_{p_2}	1500 rad/s
Lower Stopband edge	Ω_{s1}	700 rad/s
Upper Stopband edge	Ω_{s2}	1700 rad/s
Sampling Frequency	Ω_{sm}	4200 rad/s

- For IIR filter : Implemented Analog Filter type : Inverse Chebhychev

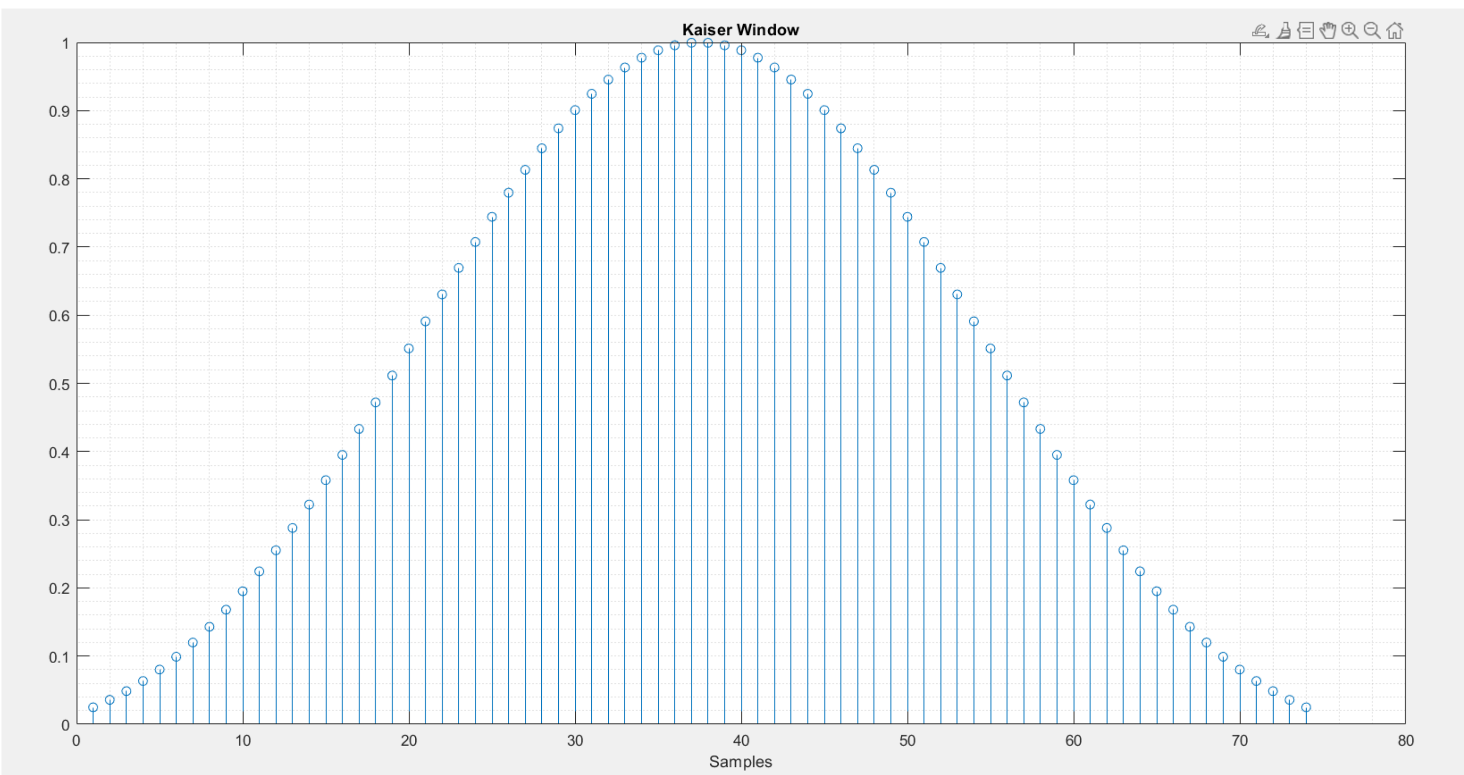
Design Procedures of the FIR filters

a) Define the filter specifications of the required filter using the data given.
b) `Kaiserord` function from the MATLAB returns the order of the filter, edge frequencies and shape factor which is used to specify the required filter
c) `fir1` function is used to create a n^{th} order FIR filter using the defined kaiser window
d) `kaiser` function returns the kaiser window we implemented.
e) `freqz , impz` codes are used to find the complex frequency response as well the impulse response vector corresponding to the digital filter
f) all the required plots are plotted in the respective graphs

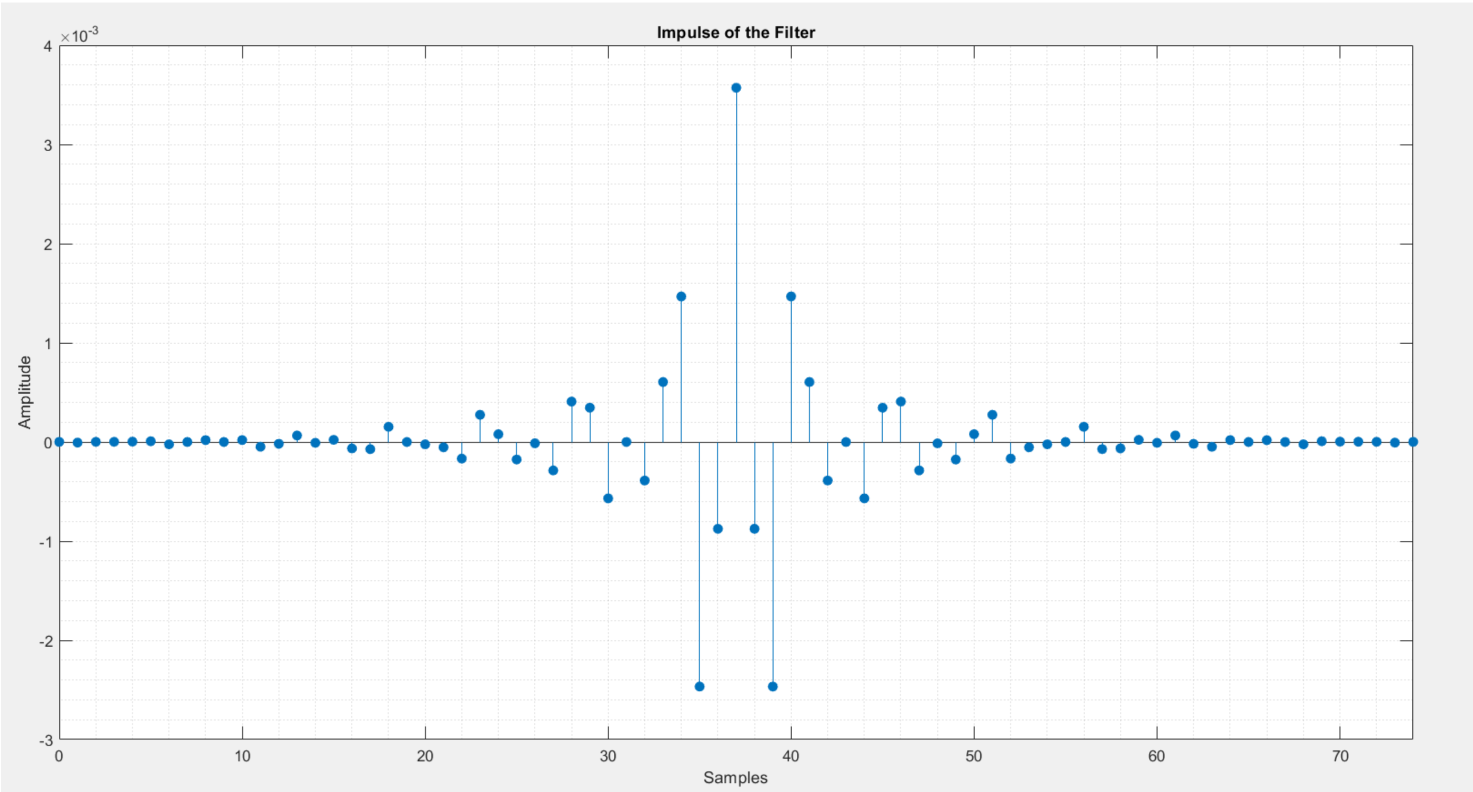
following are the snapshots of the graphs and codes of our required FIR filter.



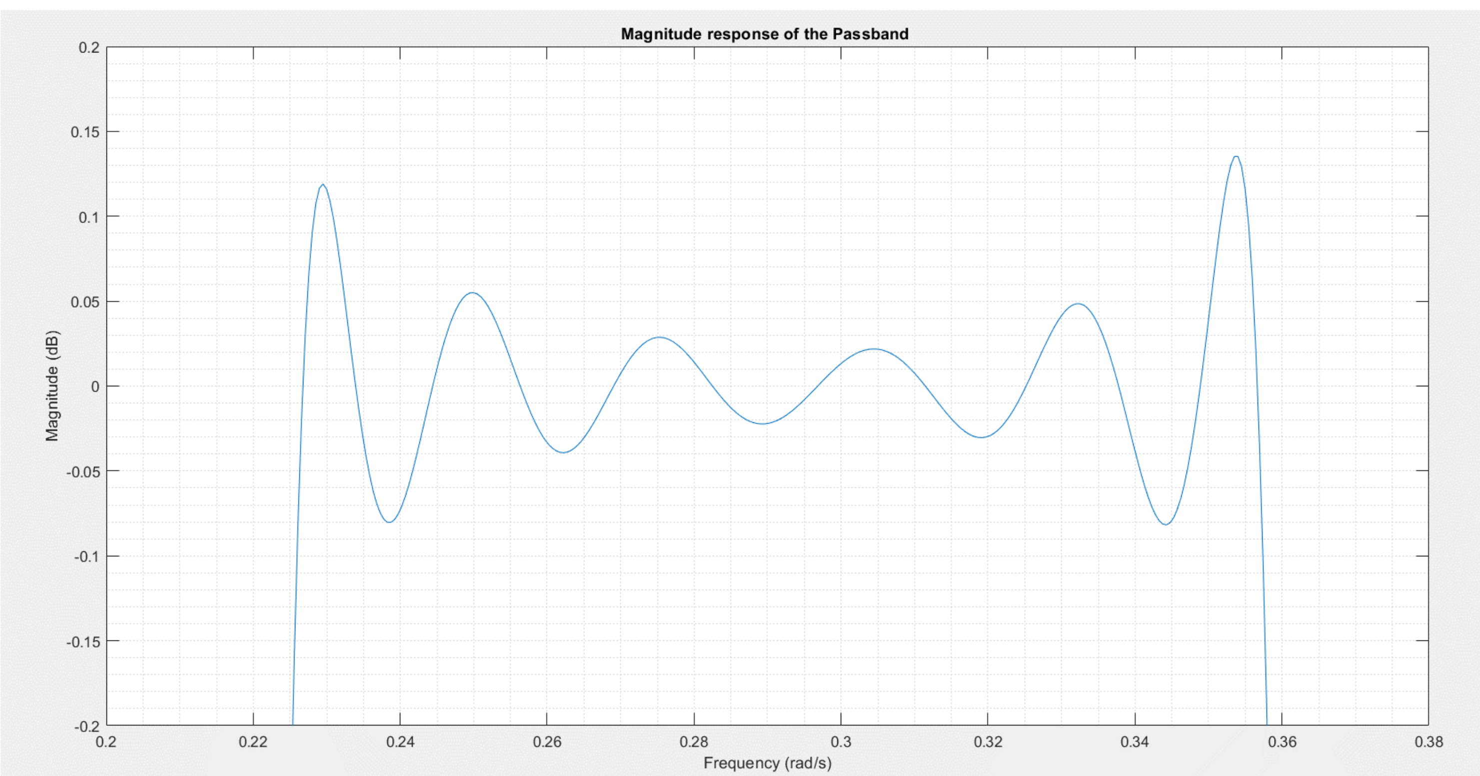
Magnitude Response of the filter



Implemented Kaiser Window



Impulse response of the filter



Magnitude response of the pass band

Design Procedures of the IIR filters

a) First thing as we said earlier due to the warping issues of frequencies due to the bilinear transformation, we need to **prewarp** critical frequencies.

b) Determine the minimum order of the filter according to the data provided. Here we need to design the IIR filter by starting off with designing a analog inverse chebhychev filter. This order of this filter is defined by `cheb2ord` function in MATLAB.

c) Then in MATLAB a prototype lowpass filter is implemented using `cheb2ap.`

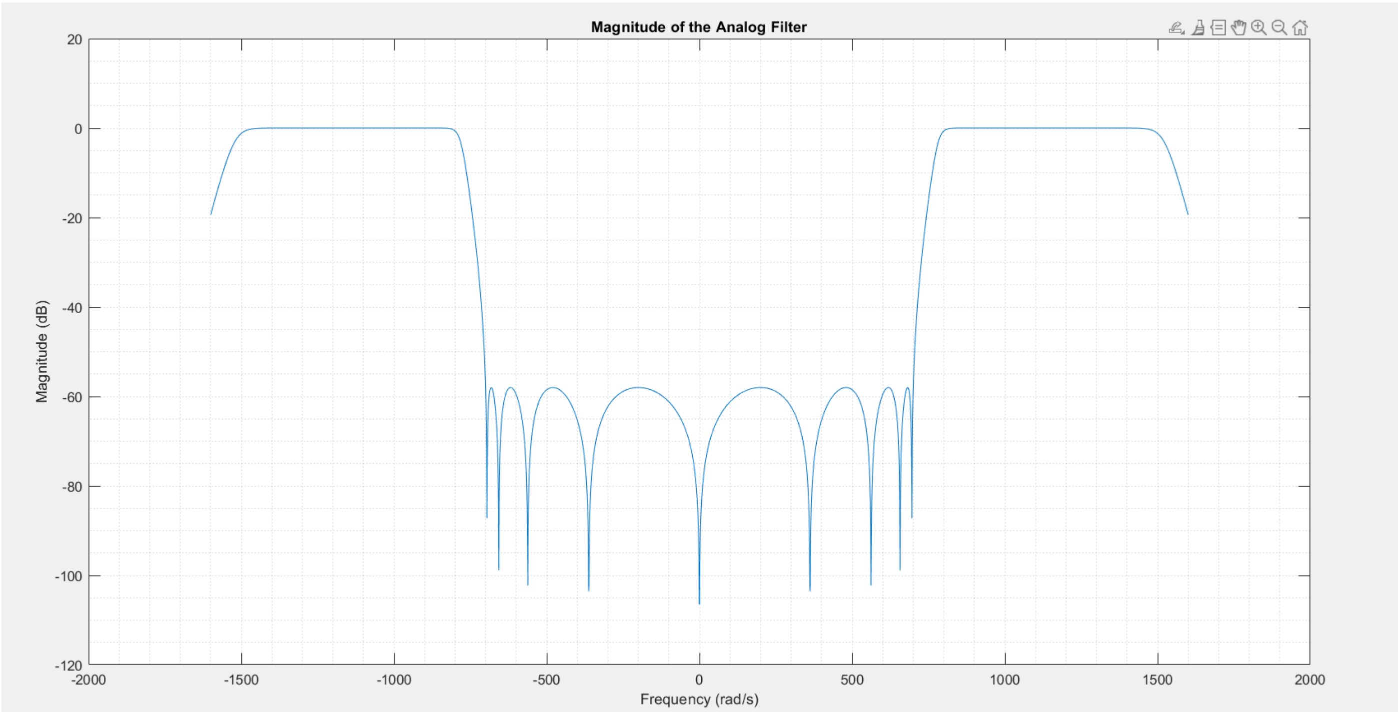
d) Then we sequentially convert the prototype to state space form using the relative MATLAB code `zp2ss`.

e) Then we transform the prototype low pass filter into bandpass filter using `lp2bp`.

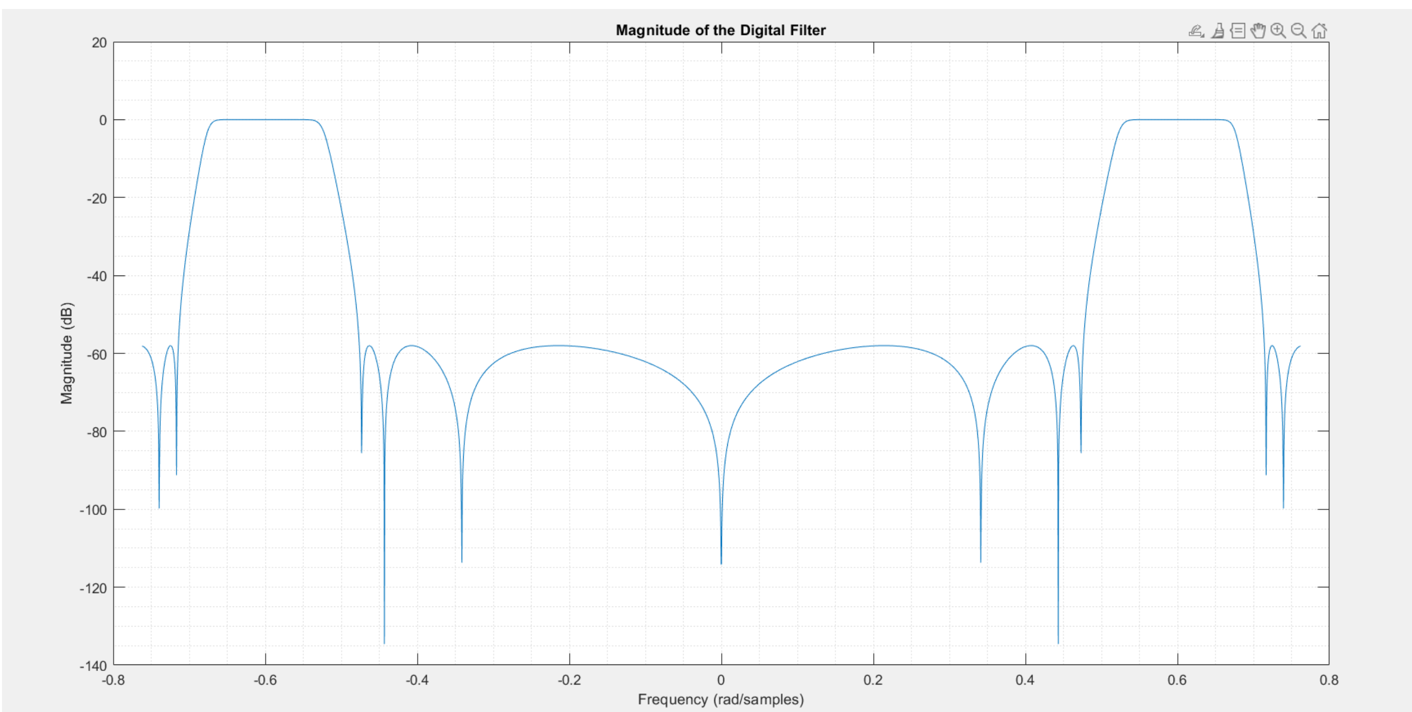
f) Then the most important part of designing the filter transforming the S domain into Z domain using the *Bilinear transformation.* This is implemented through the pre defined MATLAB code `bilinear.`

g) Then the respected transfer function is calculated and respective graphs are plotted using MATLAB.

Numerator	Denominator
0.002	1
0.00792	3.7007
0.0149	10.585
0.0209	20.295
0.0249	32.724
0.0217	41.641
0.012350	45.878
7.719519e-17	41.598
0.0123500	32.907
0.021733	21.407
0.02495	12.05358
0.02098	5.34224
0.01497	1.991663
0.0079	0.49470
0.002831	0.09576

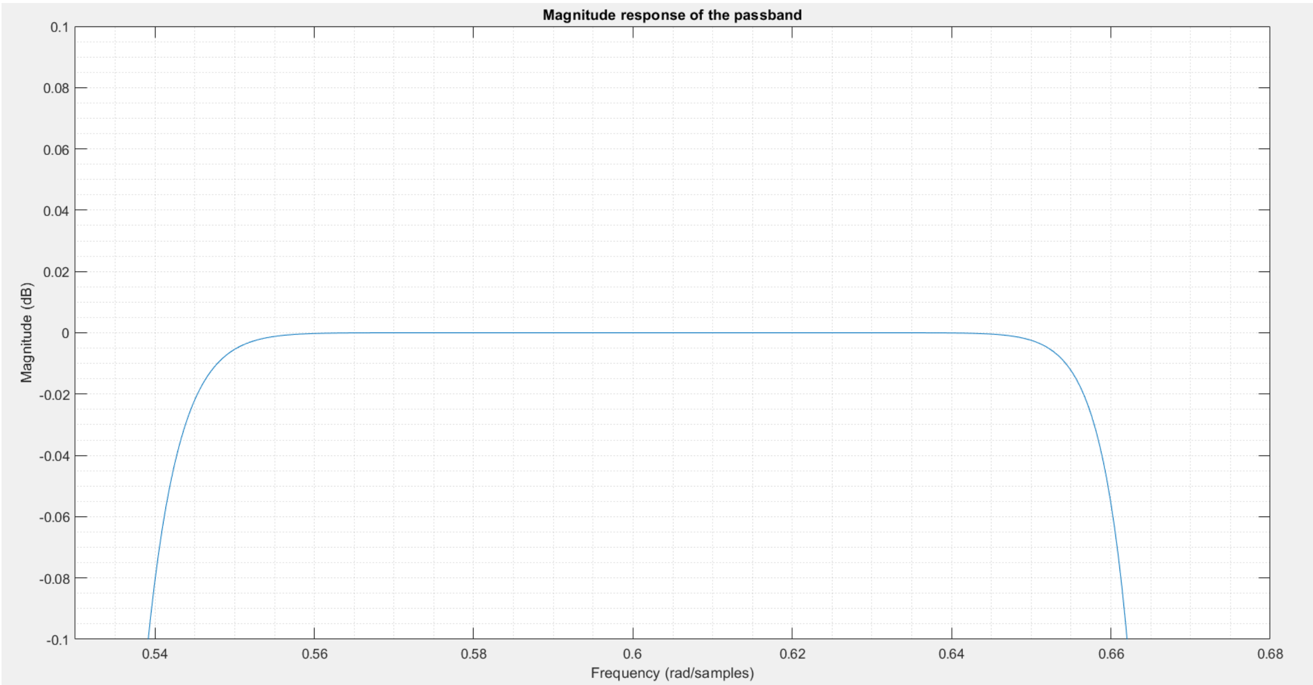


Magnitude response of the Analog filter



Magnitude response of the IIR filter

Magnitude response of the Passband



Comparison between the Two windows

As we see through the responses of the two filters designed above, we can see some distinct characteristics. Here we are going to distinguish both filters according to their order and number of multiplications and additions done. By analyzing the transfer function.

For FIR filters

- Number of multiplications = Number of Co-efficient terms in the Denominator
- Number of Co-efficient terms = Order of the filter +1
- Number of additions = Number of multiplications -1

For IIR filters

- Here there will be co-efficient terms in both numerator and denominator. Therefore
- Number of multiplications = No. of Co-efficient terms in the Denominator &Numerator
- Number of Co-efficient terms = 2*Order of the filter +2
- Number of additions = Number of multiplications -2

And the Following table shows us a brief comparison between FIR and IIR filter relating to the Order and Number of multiplications & additions.

	Order	No. of Multiplication	No. of Addition
FIR filter	74	75	74
IIR filter	14	30	28

Table showing brief comparison between IIR and FIR filters.

Appendix

MATLAB Code for the IIR filter

```
%% Filter Specifications
Wp = [1000 1500];
Ws = [700 1700];
Rp = 0.16;
Rs = 58;

%% Designing the Analog Filter
[n,Wn] = cheb2ord(Wp,Ws,Rp,Rs,"s");
[b,a]=cheby2(n,Rs,Ws,"s");
filtera = tf(b,a);
Wsm=4200;
tsm=2*pi/Wsm;


%% Magnitude Response of the Analog Filter
W=linspace(-1600,1600,3200);
H = freqs(b,a,W);
mag = abs(H);

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

Wp(1)=2/tsm*tan(Wp(1)*tsm/2); Wp(2)=2/tsm*tan(Wp(2)*tsm/2);
Ws(1)=2/tsm*tan(Ws(1)*tsm/2); Ws(2)=2/tsm*tan(Ws(2)*tsm/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
[n,Wc] = cheb2ord(Wp,Ws,Rp,Rs,'s');
[z,p,k] = cheb2ap(n,Rs);
[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));
```

```
W=linspace(-1600/(Wsm/2),1600/(Wsm/2),3200);
[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); % Coefficients of the transfer function
[num,den] = tfdata(filter)
[hd,f] = freqz(b,a,W,2);

%% magnitude response of the Digital Filter
magd = abs(hd);
figure(2)
plot(W,mag2db(magd))
title("Magnitude of the Digital Filter")
xlabel("Frequency (rad/samples)")
ylabel("Magnitude (dB)")
grid("minor")

%% Magnitude response of the passband
figure(3)
plot(W,mag2db(magd))
axis ([ 0.53 , 0.68 , -0.1 , 0.1]);
title("Magnitude response of the passband")
xlabel("Frequency (rad/samples)")
ylabel("Magnitude (dB)")
grid("minor")

-------------------------------------------------------------------------------------
```

MATLAB Code for the FIR filter

```
%% Specifications of the Filter
fsamp = 4200;
fcuts = [700 1000 1500 1700];
mags = [0 1 0];
devs = [db2mag(-58) db2mag(0.16) db2mag(-58)];

%% Designing the Kaiser window
[n,Wn,beta,ftype] = kaiserord(fcuts,mags,devs,fsamp);
n = n + rem(n,2);
hh = fir1(n,Wn,ftype,kaiser(n+1,beta),'noscale');

%% Magnitude Response of the Filter
figure;
[H,f] = freqz(hh,1,1024,fsamp);
f1=f/4200;
plot(f1,(mag2db(abs(H))))
title("Magnitude of the Filter")
xlabel("Frequency (rad/s)")
ylabel("Magnitude (dB)")
grid("minor")
```

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

%% Impulse response of the filter
figure;
impz(hh,100)
title("Impulse of the Filter")
xlabel("Samples")
ylabel("Amplitude")
grid("minor")

%% Passband Magnitude response
figure;
[H,f] = freqz(hh,1,1024,fsamp);
plot(f1,(mag2db(abs(H))))
axis ([ 900 , 1600 , -0.2 , 0.2]);
title("Magnitude response of the Passband ")
xlabel("Frequency (rad/s)")
ylabel("Magnitude (dB)")
grid("minor")
```