**UNIVERSITY OF MORATUWA**



**DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION**

**EN2570 – DIGITAL SIGNAL PROCESSING**

PROJECT REPORT

**DESIGNING A FIR BANDPASS FILTER**

R.M. PETHANGODA

180472V

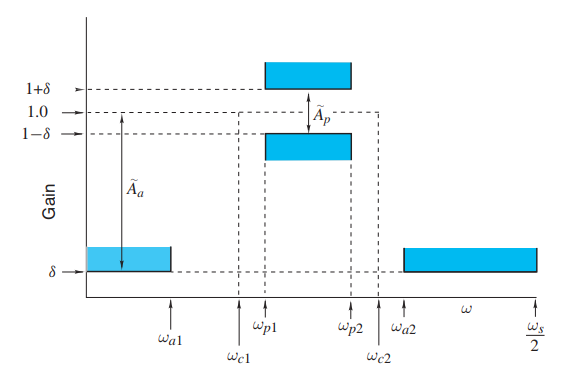
***Abstract***

This is a detailed report about designing a non-recursive finite impulse response (FIR) bandpass filter using the Kaiser window method. Derivations was done using MATLAB 2020a as a programming environment. The FIR filter was designed using the specifications provided and it was evaluated using the given excitation. Performance of the filter was analysed using the results obtained. Following sections are dedicated to analyse the filter performance starting with the basic theory and moving forward with design process used, methodology, evaluation, calculations, results and finally the conclusion.

***Introduction***

In this filter design project, it is supposed to design a FIR bandpass filter for the given specifications using the Kaiser window. Basic theoretical background is discussed in separate sections in this report. Using the window method, we can construct low-pass, high-pass, band-pass and band-stop filters for the given specifications. It’s not a complex method. It’s important to consider about the design procedure of this bandpass filter. Firstly, Kaiser window was formed according to the given specifications and then the ideal impulse response of the filter was obtained. Impulse response of the filter was calculated with those results. Then the designed filter needs to be evaluated. For that a sample signal was used. Basically, it is a summation of three different frequencies of sinusoids. Those three different frequencies are belonging to the two stopbands and passband of the bandpass filter. In the results section performance of the filter will be graphically analysed.

Parameters to consider when designing a FIR bandpass filter can be illustrated graphically like below.



***1.Basic Theory***

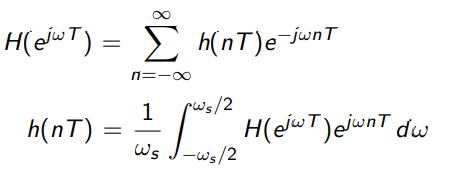
To design non-recursive filters there are two methods available,

* Windowing method (Fourier series method) – Combination of Fourier series concepts with window functions.
* Weighted – Chebyshev method – This is a multivariable optimization method.

Here we consider windowing method.

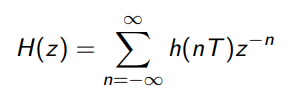
Frequency response of a digital filter is periodic with period of sampling frequency (ωs).





can be represented as a Fourier series.

If = z we can obtain the transfer function.



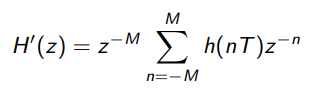
Here, h(nT) is the impulse response.

Since Fourier series coefficients are defined from -∞ to ∞ , obtained non recursive filter has infinite length and it is noncausal. To remove the infinite length, we can truncate the impulse response.

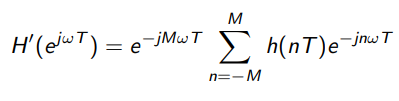
h(nT) = 0 ; |n| < M , M = (N-1)/2 here N is the length of the filter

Delay the impulse response by M samples (MT time period) to obtain a causal filter.

Then the transfer function of a causal filter



Frequency response of the causal filter: when z =



In order to obtain a causal filter, we introduced a delay of M samples. It does not change the amplitude response since || = 1. But phase response will be changed.

Truncated Fourier series creates oscillations in the amplitude response of the filter in both stopband and passband. This is called the Gibbs’ phenomenon. If we increase the length of the filter, amplitude of the oscillation wouldn’t change but it will increase the oscillation frequency. So, the filters that are designed using windowing method are practically create issues.

Truncation of h(nT) can be done using a discrete time sequence called windowing function w(nT).

hw(nT) = h(nT). w(nT) ; hw(nT) is the modified impulse response

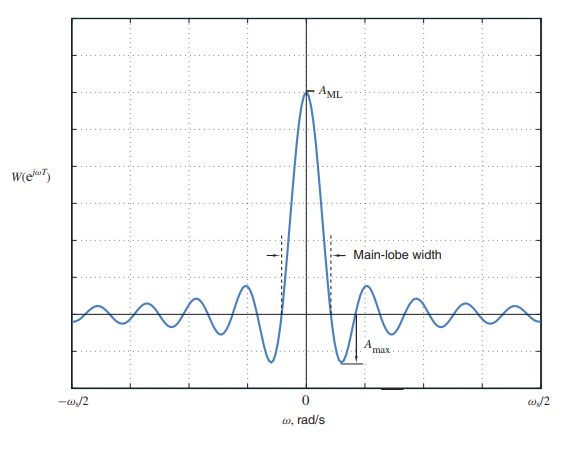
Transfer function is,

Hw(z) = Z[h(nT). w(nT)]

To obtain the frequency response of the modified filter evaluate Hw(z) |z = ejwT(on the unit circle). The convolution integral of it,



Where *W(ejwT)* is the frequency response of the windowing function. Its frequency domain representation is,

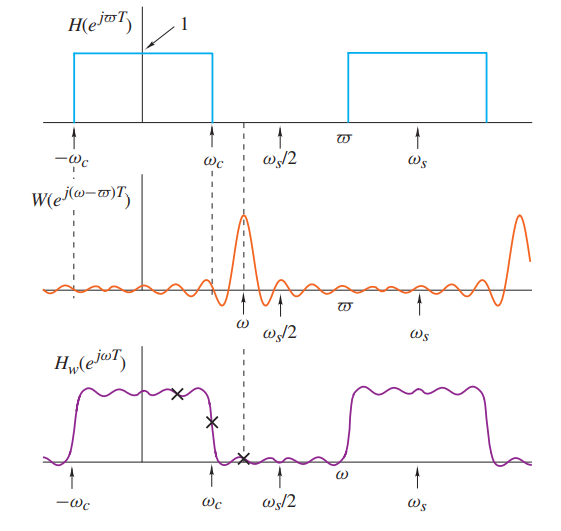


A window can be characterized by two parameters

* Main-lobe width (BML) - Like shown in the graph, it’s the gap between the first positive and negative zero crossings.
* Ripple ratio (r) - This is defined as; (Amax/AML ) x 100% , 20 log(Amax/AML) dB AML is the maximum amplitude of the main-lobe and Amax is the maximum side-lobe amplitude.

To obtain the maximum efficiency of a window spectral energy has to be cantered to the main lobe rather than scattering into side lobes.

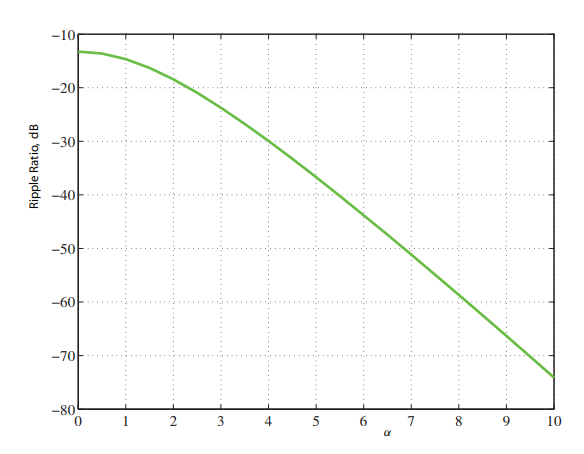
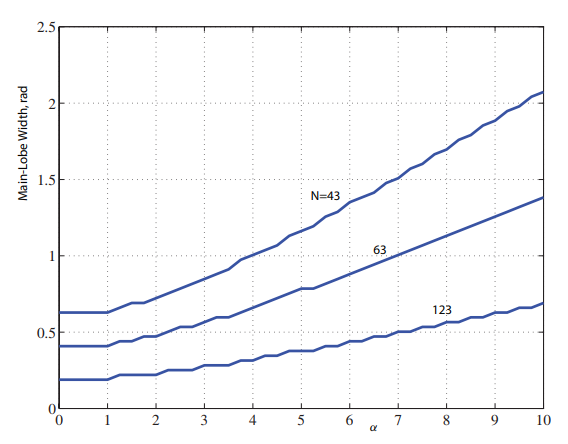
This is an illustration of how the window function contributes in reducing the Gibb’s phenomenon when designing a low pass filter. Consider the convolution integral of *Hw(ejwT)* and the area under the spectrum of window function to be 2π/T.



Here we can see ripples in the passband and stopband and a steep curve in the transition. Main-lobe width and the ripple ratio of the window determines them.

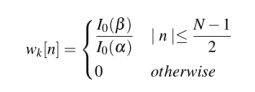
***2.Kaiser Window***

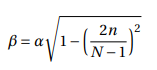
This window is formed basically by changing α and N (length of the filter). First α is selected for the desired ripple ratio. Then based on α, N is increased until the desired main-lobe width is achieved. Here we have to be careful about not to select a large value for N, because it will reduce the efficiency of the filter.



ripple ratio vs α (decreasing graph) main-lobe width vs α (increasing graph)

Kaiser window is defined like this,





Here and

***3.Methodology***

Tasks of this project are, design a bandpass filter for the given specifications using Kaiser window and evaluation of the designed filter output with an ideal bandpass filter output for a given excitation.

First calculations were done to obtain the parameters to form the Kaiser window. Next using the Fourier series method discussed above was used to determine the ideal impulse response. After that the desired bandpass filter was generated by multiplying ideal impulse response and window function. Then it was plotted in frequency and time domains. Finally, the given excitation (sample signal) was sent through the designed bandpass filter and ideal bandpass filter separately, and the outputs are analysed in frequency and time domains.

*3.1 Specifications for the filter design*

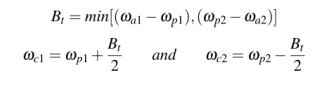
|  |  |  |
| --- | --- | --- |
| Parameter | Value | Unit |
| Maximum passband ripple (Ãp) | 0.07 | dB |
| Minimum stopband attenuation (Ãa) | 52 | dB |
| Lower passband edge (*ωp1*) | 500 | rad/s |
| Upper passband edge (*ωp2*) | 900 | rad/s |
| Lower stopband edge (*ωa1*) | 350 | rad/s |
| Upper stopband edge (*ωa2*) | 1000 | rad/s |
| Sampling frequency (*ωs*) | 2800 | rad/s |

*3.2 Design process of bandpass filter*

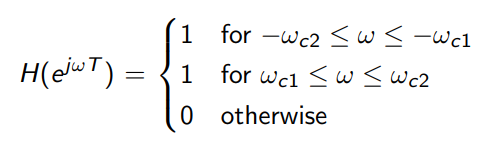
* To convert the given parameter frequencies in rad/s to rad/sample within the range (-π , π) following equation is used,

*ω* = = (rad/sample)

* Obtained the most critical transition width out of the two transitions in bandpass filter and two cut-off frequencies were determined (*ωc1 , ωc2*).

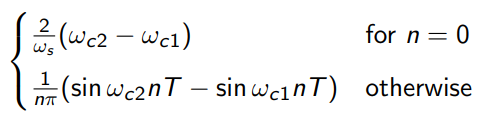


* Ideal frequency response of the bandpass filter is,



*Hi (ejw )* =

* Ideal impulse response was obtained using the inverse fourier transform



hi[n] =

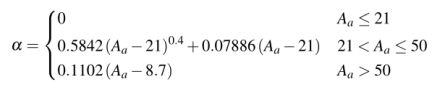
* value for 𝛿 is chosen such that actual passband ripple (Ap) Ãpand actual minimum stopband ripple (Aa) Ãa.

 and

Satisfying above condition value for 𝛿 = min (𝛿p, 𝛿a)

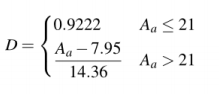
Then, actual stopband attenuation is Aa = -20 log(𝛿)

* Obtain the value of alpha



Next step is to take the N (length of the filter).

* For N, value of D is needed



* Value for N is the minimum odd integer that satisfies the below inequality



Then Kaiser window is generated using zeroth order modified Bessel function (I0(x)) by taking the approximated summation of 500 terms.

* Multiplying the ideal impulse response with window function impulse response of the bandpass filter is taken.

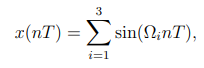
h[n] = hi[n]. w[n]

* Causal impulse response is obtained by delaying the impulse response by (N-1)/2 samples.

hc[n] = h [n- (N-1)/2

*3.3 Evaluation*

The designed filter was evaluated by providing the given excitation.



Here,

Ω1 – Middle frequency of the lower stopband

Ω2 – Middle frequency of the passband

Ω3 – Middle frequency of the upper stopband

After getting outputs from the ideal filter and the designed bandpass filter, they were analysed separately in time and frequency domains. Considering what is the response for a ideal filter and designed bandpass filter.

***4.Calculations***

* Calculated parameters for the filter specifications

|  |  |  |
| --- | --- | --- |
| Parameter | Value | Units |
| Critical Transition width (Bt) | 100 | rad/s |
| Lower cutoff frequency(ωc1) | 450 | rad/s |
| Upper cutoff frequency(ωc2) | 950 | rad/s |
| Sampling Period (T) | 0.002243 | s |

* Calculated parameters to form the Kaiser window

|  |  |  |
| --- | --- | --- |
| Parameter | Value | Units |
| 𝛿p | 0.0040 | - |
| 𝛿a | 0.0025 | - |
| 𝛿 | 0.0025 | - |
| Aa | 52 | dB |
| α | 4.77166 | - |
| D | 3.06754874 | - |
| N | 87 | - |

* Frequencies for the excitation

|  |  |  |
| --- | --- | --- |
| Frequency component | Value | Units |
| Ω1 | 225 | rad/s |
| Ω2 | 700 | rad/s |
| Ω3 | 1175 | rad/s |

***5.Results***

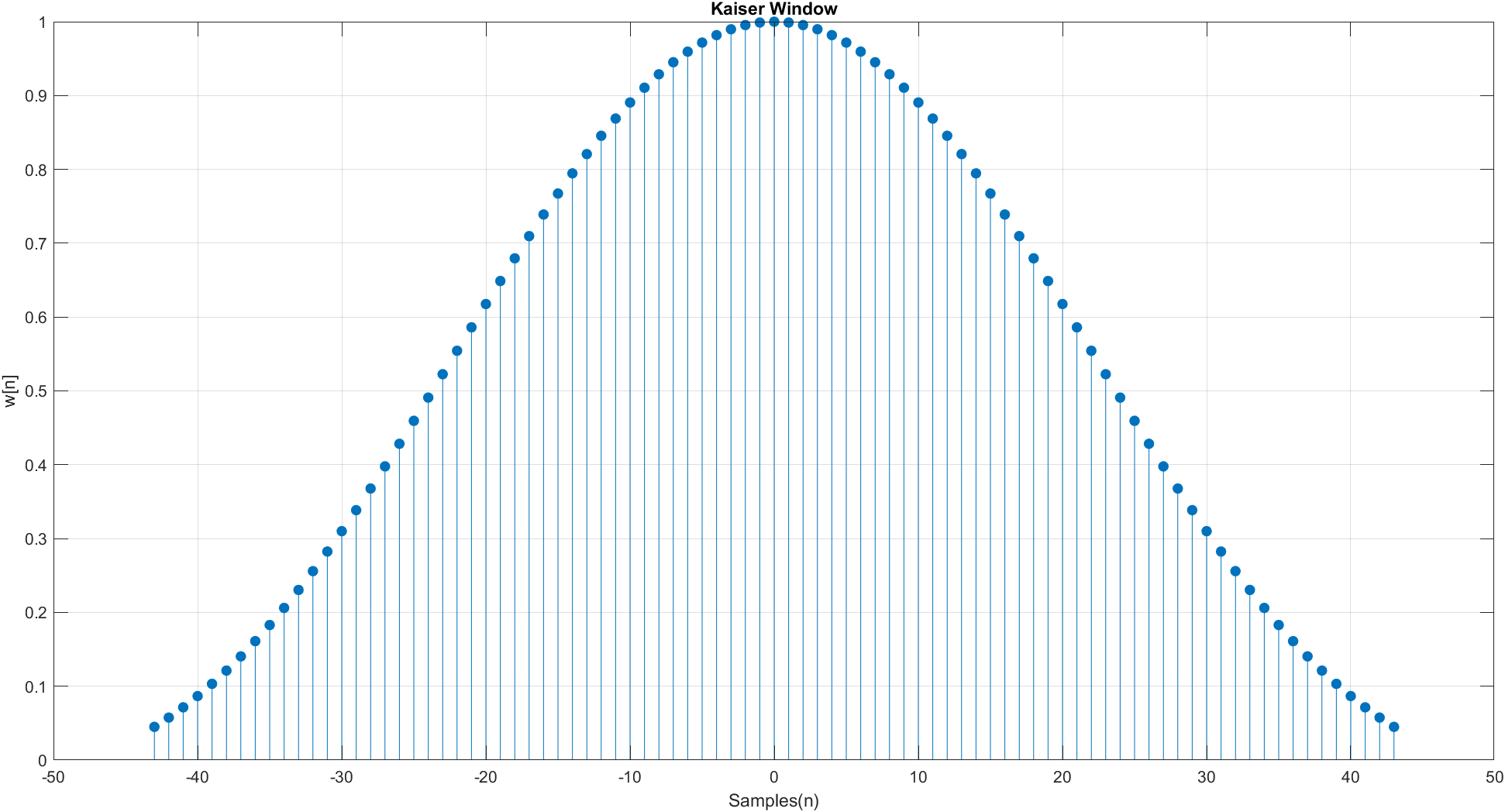
* Designed Kaiser Window

Figure (1)

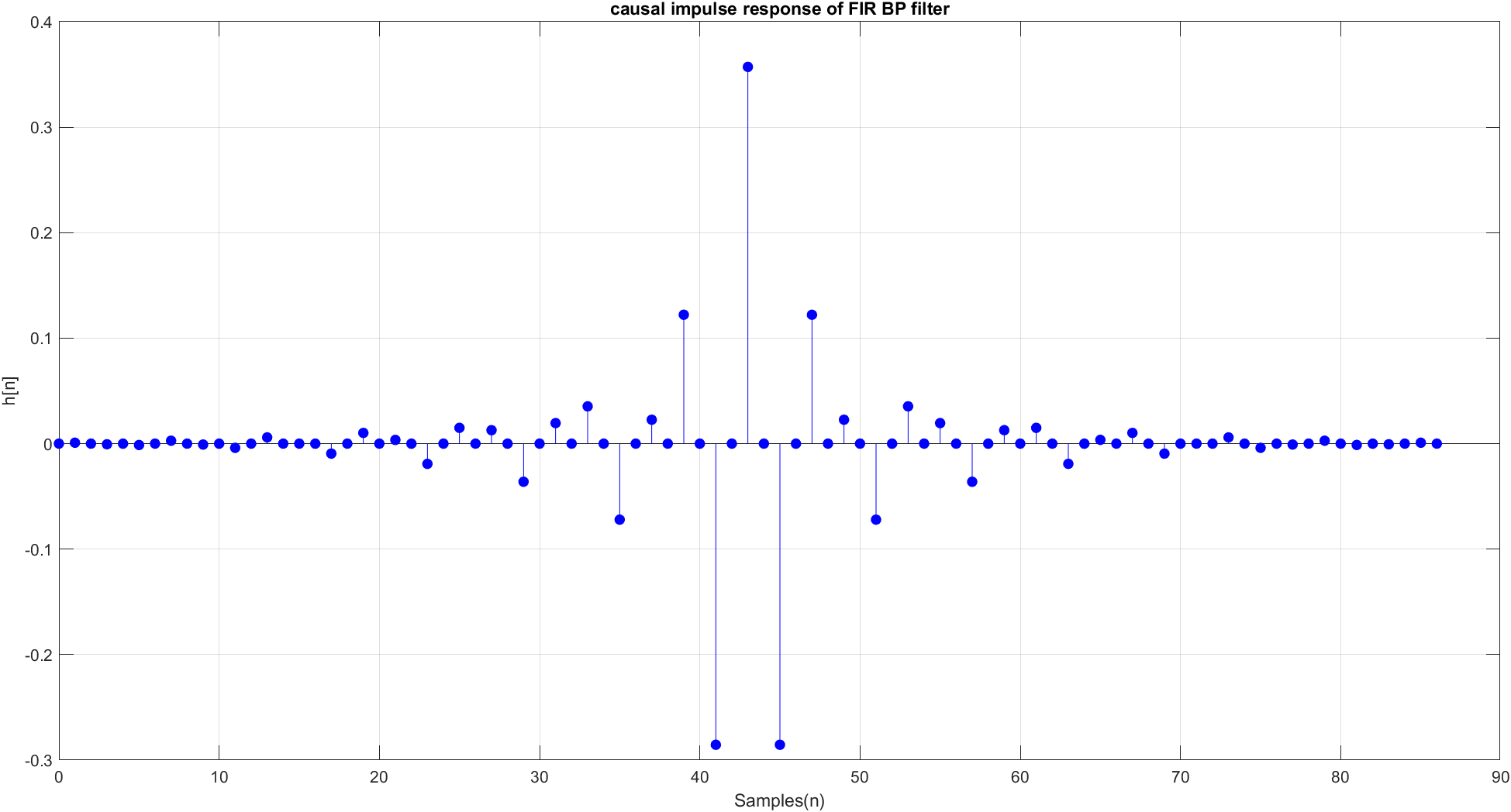
* Causal impulse response of the designed bandpass filter

Figure (2)

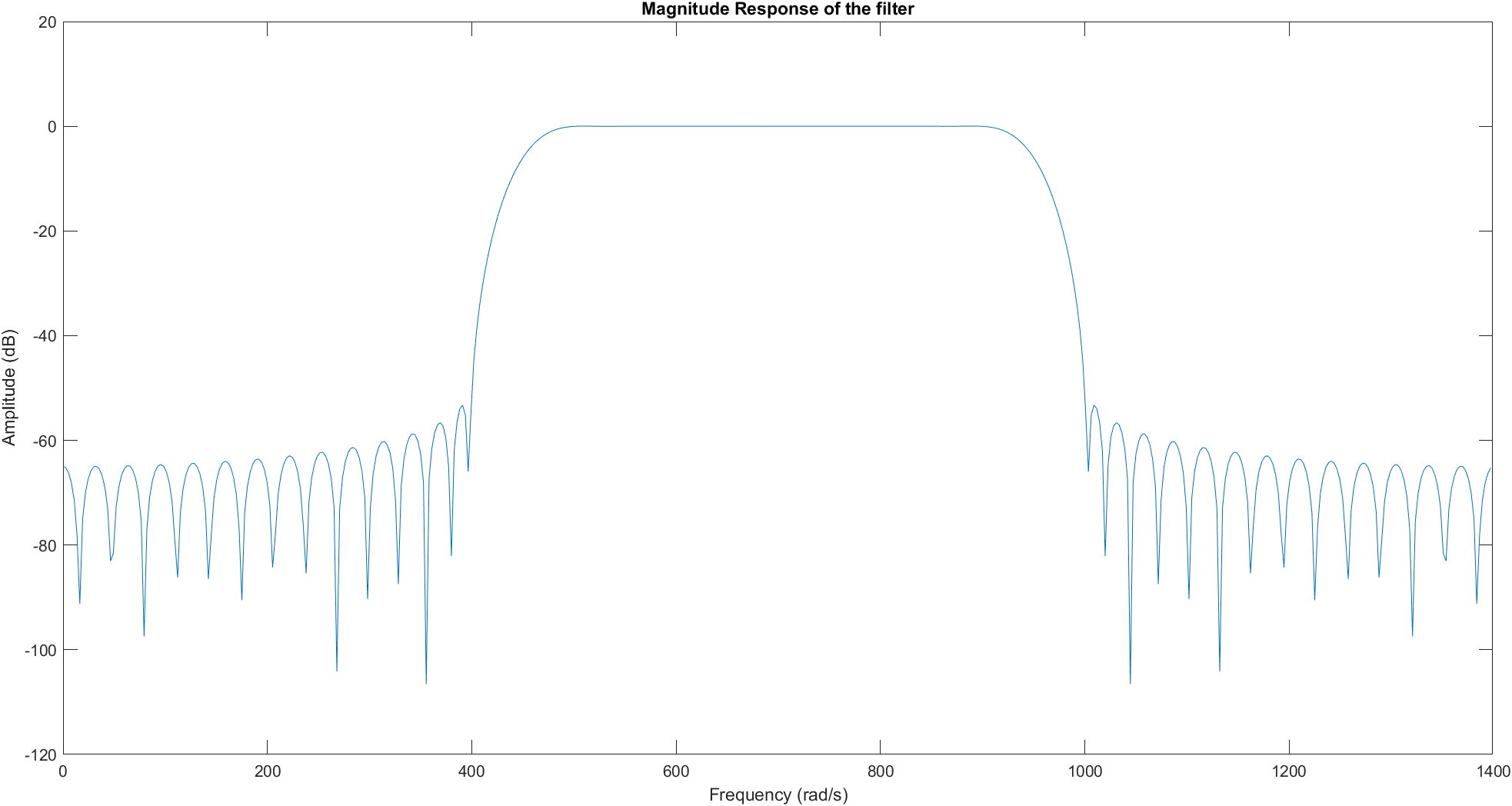
* Magnitude response of the filter for the frequency range 0 to Ωs/2

Figure (3)

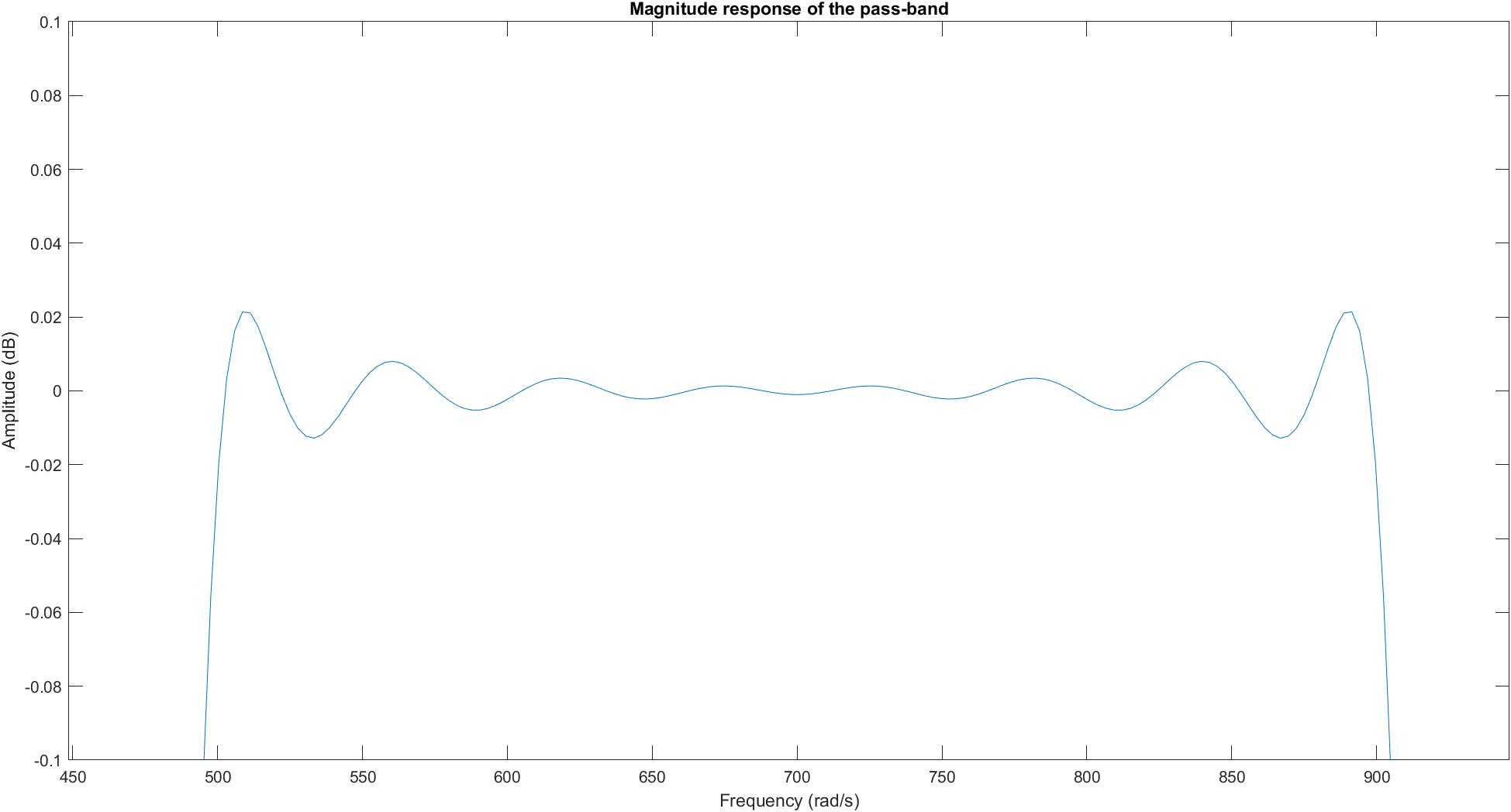
* Magnitude response of the filter for the frequencies in the passband

Figure (4)

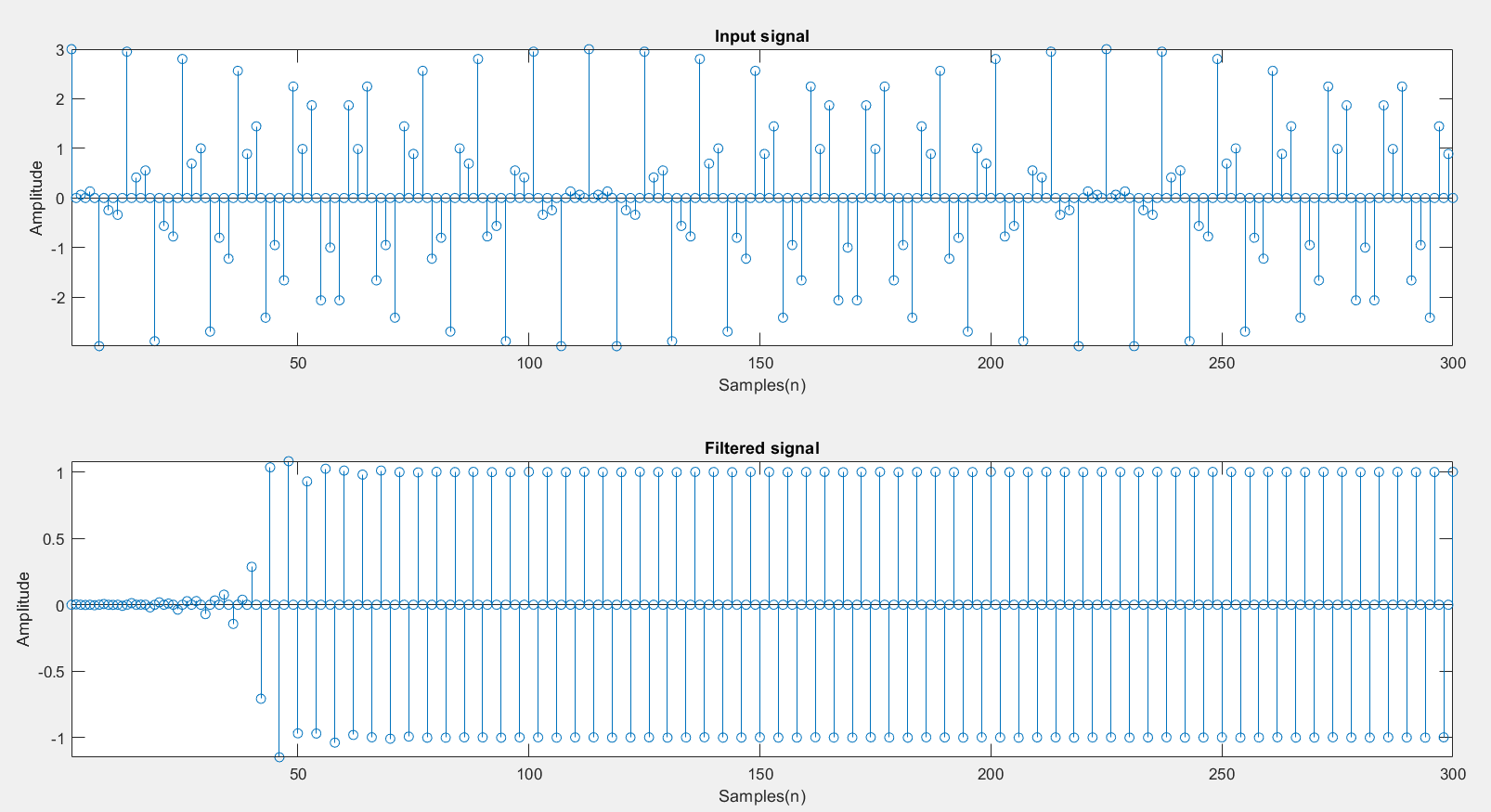
* Time domain response of the filter for the given excitation & corresponding filtered signal.

Figure (5)

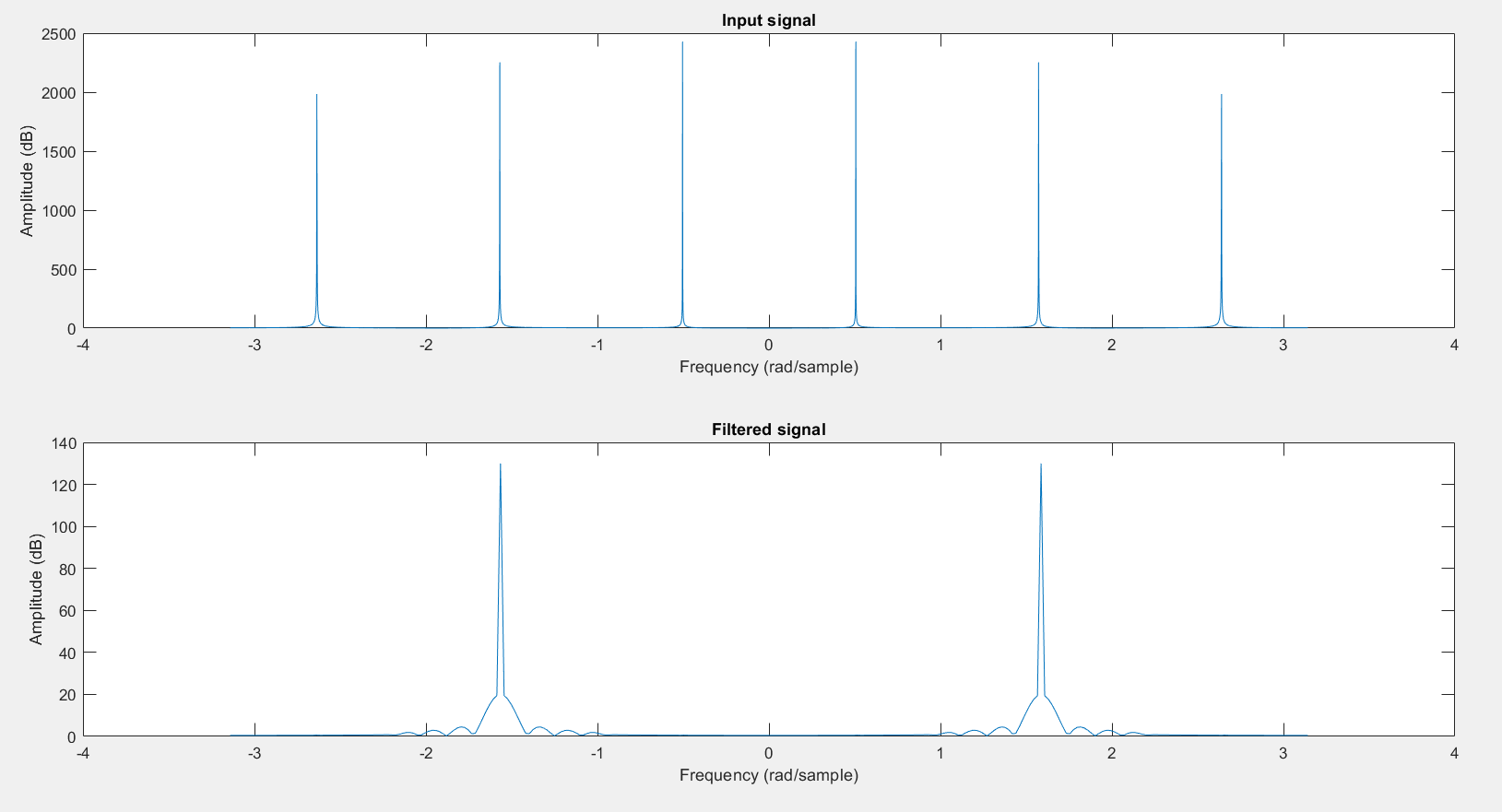
* Frequency domain response of the input signal & it’s corresponding filtered signal

Figure (6)

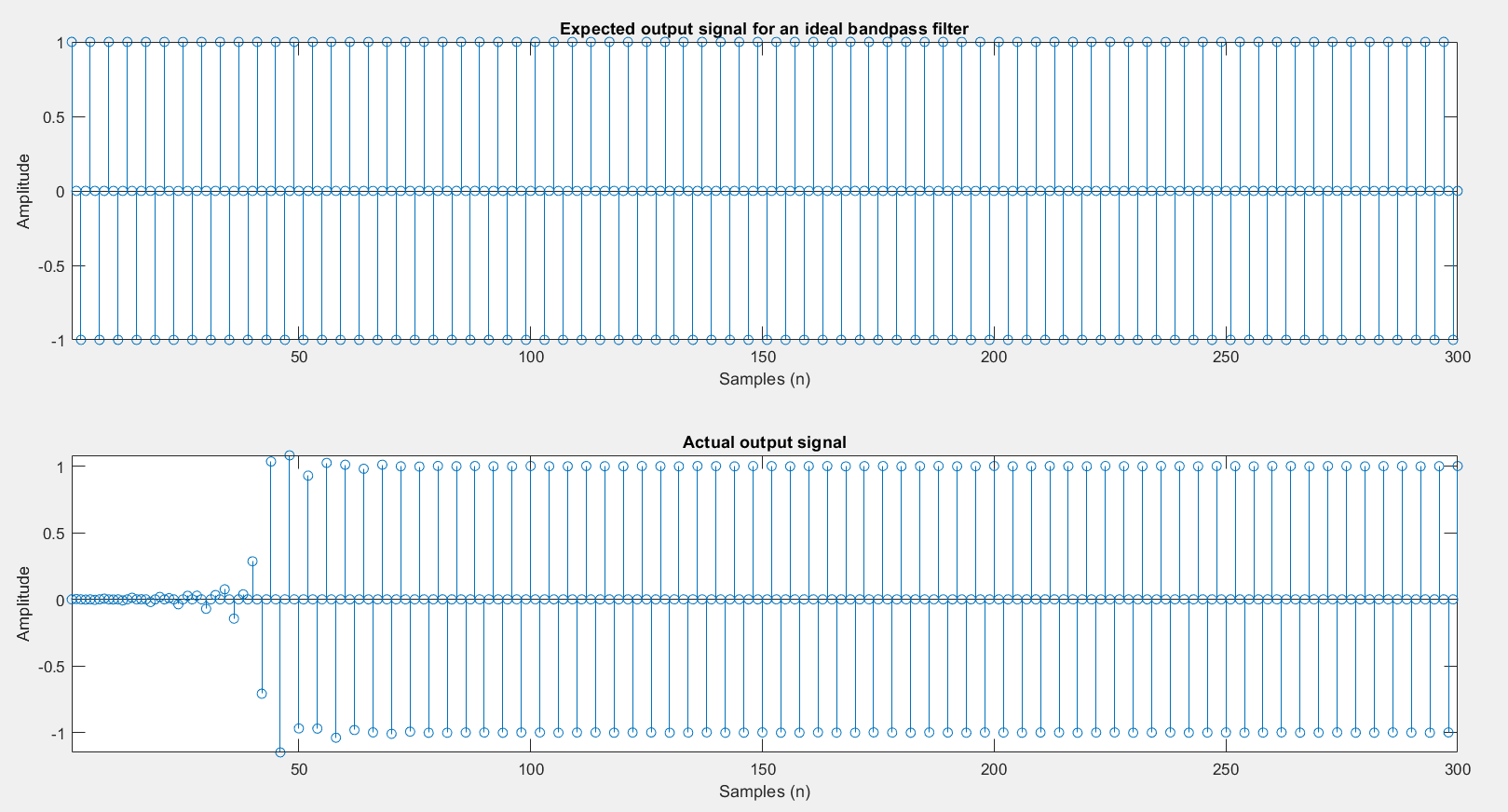
* Comparison of outputs for an ideal filter and designed filter in time domain for the given excitation.

Figure (7)

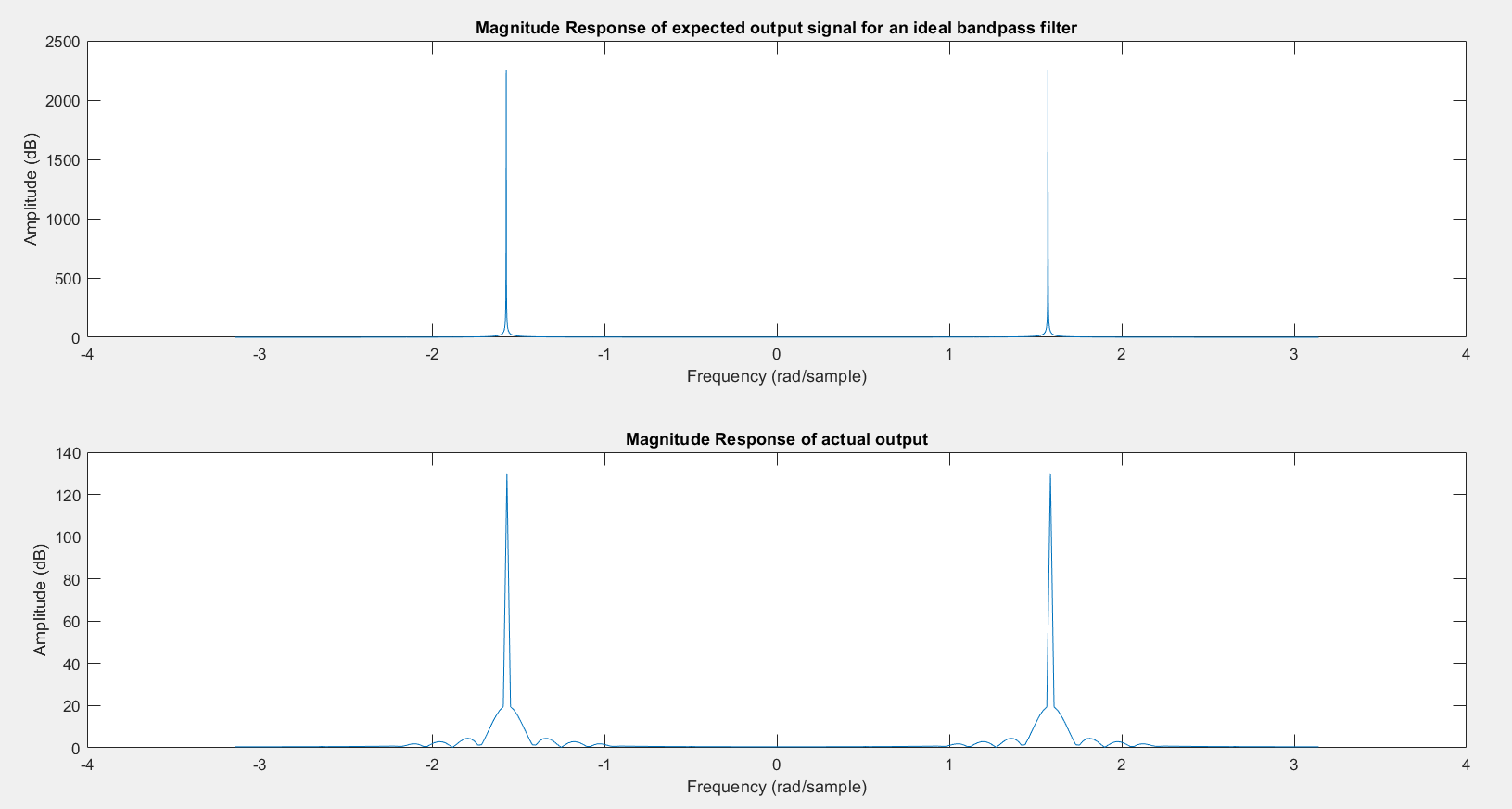
*  Comparison of outputs for an ideal filter and designed filter in frequency domain for the given excitation.

Figure (8)

***6.Conclusion***

* Considering the comparison of outputs for an ideal filter and designed filter in frequency domain for the given excitation (figure (8)), it can be seen that the positions of the two spikes are same in the frequency domain for both occasions. So, it can be verified that the output from the bandpass filter for the given excitation is correct.
* By observing the frequency domain response of the input signal & it’s corresponding filtered signal (figure (6)), it’s clear that frequency components corresponding to the stopband of the filter are absent in the filtered signal and frequency components corresponding to the passband of the filter are present in the filtered signal. Therefore, it’s clear that the correct frequencies have been passed through the filter and output signal is a filtered version of the input signal.
* Results discussed above implies that from the given specifications the required results can be obtained by the design of this bandpass filter.
* Hence, a bandpass filter can be formed successfully using the Kaiser window method.

***7.Discussion***

Using the windowing method infinite impulse response can be truncated to a finite impulse response. It has been successfully implemented in this finite impulse response (FIR) filter design project. The designed bandpass filter can be considered in practical instances where stopband attenuation is not equal unity and passband ripples can be neglected. These are the things to consider when comparing with an ideal filter response. When looking at the results obtained these things are actually neglectable. There were closed to 40 zero samples in figures 5 & 7. Therefore, 200 -300 samples were required to obtain a steady-state response.

***References***

A. Antoniou, *Digital Signal Processing – Chapter 9, Design of nonrecursive (FIR) filters*

**Appendix – MATLAB code**

close all;

clear all;

clc

%% my index number = 180472V

A = 4;

B = 7;

C = 2;

%% declare parameters for the Bandpass Filter

A\_p=0.03+0.01\*A; % Maximum passband ripple

A\_a=45+B; % Minimum stopband attenuation

W\_p1=C\*100+300; % Lower passband edge

W\_p2=C\*100+700; % Upper passband edge

W\_a1=C\*100+150; % Lower stopband edge

W\_a2=C\*100+800; % Upper stopband edge

Ws=2\*(C\*100+1200);

T = (2\*pi)/Ws; %sampling period

fs=Ws/(2\*pi); %sampling frequency

%% get the most critical transition width out of two transitions

Bt=min((W\_p1-W\_a1),(W\_a2-W\_p2));

W\_c1=W\_p1-Bt/2;%ideal lower cutoff frequency

W\_c2= W\_p2+Bt/2;%ideal upper cutoff frequency

%% get actual passband ripple

delta\_P=(10^(0.05\*A\_p)-1)/(10^(0.05\*A\_p)+1);

delta\_A=10^(-0.05\*A\_a);

%% choose delta to be the minimum of delta\_P and delta\_A

delta=min(delta\_P,delta\_A);

%% actual stopband attenumation & passband attenuation calculation

A\_a= - 20\*log10(delta);

%% obtain alpha using A\_a

if A\_a <= 21

alpha=0;

elseif A\_a <= 50

alpha=0.5842\*(A\_a-21)^0.4 + 0.07886\*(A\_a-21);

else

alpha=0.1102\*(A\_a-8.7);

end

%% obtain the parameter D

if A\_a<=21

D=0.9222;

else

D=(A\_a-7.95)/14.36;

end

%% get N (length of the filter)

var=(Ws\*D/Bt)+1;

int\_var = fix(var); % get the integer part of var

%select lowest odd value as N satisties N >= var

if (var-int\_var) ==0

if mod(var,2)==1

N=var;

else

N=var+1;

end

else

if mod(int\_var,2)==0

N=int\_var+1;

else

N=int\_var+2;

end

end

%% generate the kaiser window

M = (N-1)/2;

n=linspace(-M,M,2\*M+1);

beta = alpha\*(1 - ((2\*n)/(N-1)).^2).^0.5;

num = 1;

denom = 1;

for k = 1 : 500

num = num + ((1/factorial(k))\*(beta/2).^k).^2;

denom = denom + ((1/factorial(k))\*(alpha/2).^k).^2;

end

w\_n = num /denom;

figure;

stem(n,w\_n,'fill')

ylabel('w[n]');

xlabel('Samples(n)');

title('Kaiser Window');

grid on;

%% ideal BPF generation

negative\_range = -M:1:-1;

positive\_range = 1:1:M;

%When n != 0

h1\_n=(sin(W\_c2\*negative\_range/fs)-sin(W\_c1\*negative\_range/fs))./(pi\*negative\_range);

h2\_n=(sin(W\_c2\*positive\_range/fs)-sin(W\_c1\*positive\_range/fs))./(pi\*positive\_range);

%When n = 0

h\_0 =(W\_c2-W\_c1)\*2/Ws;

%ideal BP filter

h\_n = [h1\_n h\_0 h2\_n];

%resulting FIR BP Filter

hw\_n=h\_n.\*w\_n;

figure;

stem(n,hw\_n,'fill','blue');

xlabel('Samples(n)');

ylabel('h[n]');

title('FIR BP filter (Impulse response)');

grid on;

%% causal impulse response of FIR BP filter

shifted\_samples = [ 0 : 1 : N-1];

figure;

stem(shifted\_samples,hw\_n,'fill','blue');

xlabel('Samples(n)');

ylabel('h[n]');

title('causal impulse response of FIR BP filter');

grid on;

%% Magnitude response of the filter

[h,f] = freqz(hw\_n);

hn = 20\*log10(abs(h));

f1 = f\*(Ws/(2\*pi)); % frequency domain correction

figure;

plot(f1,hn);

title('Magnitude Response of the filter');

xlabel('Frequency (rad/s)');

ylabel('Amplitude (dB)');

%% Magnitude response for the frequencies in the pass-band

P1 = round(length(f1)\*(2\*W\_c1/Ws));

P2 = round(length(f1)\*(2\*W\_c2/Ws));

Fpass = f1(P1:P2);

Mpass = hn(P1:P2);

figure;

plot(Fpass,Mpass);

axis([-inf, inf, -0.1, 0.1]);

title('Magnitude response of the pass-band');

xlabel('Frequency (rad/s)')

ylabel('Amplitude (dB)')

%% Generate the excitation

nx = 0:1:5000;

omega\_1 = W\_c1/2;

omega\_2 = (W\_c1+W\_c2)/2;

omega\_3 = (W\_c2+Ws/2)/2;

x\_n = cos(omega\_1\*T\*nx)+cos(omega\_2\*T\*nx)+cos(omega\_3\*T\*nx);

%% Time domain response of the excitation (x\_n) and the filtered output

len = length(fft(x\_n)) + length(fft(hw\_n)) - 1;

y\_f = fft(x\_n,len).\*fft(hw\_n,len);

y\_n = ifft(y\_f,len);

y\_n = y\_n(1:300);

figure;

subplot(2,1,1);

stem(x\_n(1:300));

axis tight;

title('Input signal');xlabel('Samples(n)');ylabel('Amplitude');

subplot(2,1,2)

stem(y\_n(1:300));

axis tight;

title('Filtered signal');xlabel('Samples(n)');ylabel('Amplitude');

%% Frequency domain response of the excitation (x\_n) and the filtered output

X\_n = abs(fft(x\_n));

X\_nf = [X\_n(floor(numel(X\_n)/2)+1:numel(X\_n)),X\_n(1:floor(numel(X\_n)/2))];

d1 = linspace(-pi,pi, numel(X\_n));

figure;

subplot(2,1,1)

plot(d1,X\_nf);

title('Input signal');xlabel('Frequency (rad/sample)');ylabel('Amplitude (dB)');

subplot(2,1,2)

y1\_f = abs(fft(y\_n));

Y\_nf=[y1\_f(floor(numel(y1\_f)/2)+1:numel(y1\_f)),y1\_f(1:floor(numel(y1\_f)/2))];

d2 = linspace(-pi,pi, numel(y1\_f));

plot(d2,Y\_nf);

title('Filtered signal');xlabel('Frequency (rad/sample)');ylabel('Amplitude (dB)');

%% Comparing outputs for ideal filter and disigned filter in time domain

y1\_n = cos(omega\_2\*T\*nx); %in the passband this signal only

figure;

subplot(2,1,1)

stem(y1\_n(1:300));

axis tight;

title('Expected output signal for an ideal bandpass filter');xlabel('Samples (n)');ylabel('Amplitude');

subplot(2,1,2)

stem(y\_n(1:300))

axis tight;

title('Actual output signal');xlabel('Samples (n)');ylabel('Amplitude');

%% Comparing outputs for ideal filter and disigned filter in frequency domain

y2\_f = abs(fft(y1\_n));

Y\_n\_plot=[y2\_f(floor(numel(y2\_f)/2)+1:numel(y2\_f)),y2\_f(1:floor(numel(y2\_f)/2))];

d3 = linspace(-pi,pi, numel(y2\_f));

figure;

subplot(2,1,1);

plot(d3, Y\_n\_plot);

title('Magnitude Response of expected output signal for an ideal bandpass filter');xlabel('Frequency (rad/sample)');ylabel('Amplitude (dB)');

subplot(2,1,2)

d4 = linspace(-pi,pi, numel(y1\_f));

plot(d4,Y\_nf);

title('Magnitude Response of actual output');xlabel('Frequency (rad/sample)');ylabel('Amplitude (dB)');