Department of Electronic & Telecommunication Engineering

University of Moratuwa

EN 2570 – Digital Signal Processing

DESIGN OF A FINITE-DURATION IMPULSE RESPONSE (FIR) BAND PASS FILTER

<u>Project Report</u>

Name : Rathnayaka R.G.H.V.

Index No : 180529E

March 5, 2021

This Report is submitted as a partial fulfillment of the requirements

for the module EN2570 – Digital Signal Processing

ABSTRACT

This report mainly outlines the designing process of a non- recursive Finite duration Impulse Response (FIR) band pass filter which is satisfying some set of characteristics. The complete designing process is based on Software implementation on MATLAB 2018a which provide a better programming environment for this designing project. The filter design process basically falls upon the provided specifications by using Fourier series method (windowing method) in conjunction with Kaiser window. Here, Kaiser window is used for impulse response truncation. The performance of the filter is evaluated then with the aid of analyzing the obtained outputs for a test input consisting of sinusoidal signals. In the following pages of the project report the basic underlying theoretical construct, results and conclusions are brought out.

Keywords : Non Recursive (FIR) Bandpass Filter, Kaiser Window Function, Windowing Method, MATLAB 2018a Package.

INTRODUCTION

Digital filters are considered as an essential element in telecommunication engineering sector now a day. They are commonplace in everyday electronics. A digital band pass filter is a filter which is capable on attenuating all the frequency components outside a certain frequency range in a sampled discrete time signal.

The report is mainly focusing on the designing of a band pass Finite Impulse Response (FIR) Digital Filter with a step by step approach for given specifications using the windowing technique (Fourier series method) in conjunction with the Kaiser window. Software implementations related to the designing process have been done with the use of MATLAB 2018a programming environment.

There are two classical methods in designing non recursive digital filters with varying levels of complexity and accuracy.

- <u>Windowing method – Fourier Series Method</u>
  Fourier series is used in designing process in conjunction with a class of functions which are named as window functions.
- <u>Weighted – Chebyshev Method</u>
  A multivariable optimization method used in digital filter design.

In this project implementation, the closed form direct approach is used by Fourier series method in defining and filter implementation. Kaiser window is used in windowing. The Kaiser window parameters are used to tune the filter to given characteristics.

The time domain and frequency domain representation of the filter at different stages will evaluate these characteristics. The Fast Fourier Transform (FFT) function of MATLAB is used in obtaining the frequency response of the filter in this project which is an implementation of Discrete Fourier Transforms (DFTs). A sinusoidal signal consisted of three frequencies are used in order to test the resulting filter as input. One of it is in the pass band. The output is then compared with the original sinusoidal components of the input in the pass band. The results are analyzed to identify the drawbacks and improvements that can be made to make the filter efficient and effective.

BASIC THEORY

Implementation of a band pass filter and evaluating its performance is the main task associated with this project. Here, the methodology associated with these two tasks are discussed widely. Filter design process is of several stages. Choosing the correct design parameter is the first stage associated with this design process. Afterward, the ideal frequency response of the band pass filter is obtained. After that, a Kaiser window function is obtained such that the ideal frequency response is truncated preserving the required parameters in filtering. As the final stage, the time domain representation and the frequency domain representation of the designed band pass filter are obtained. Evaluation task can be done with generation of input signals where output signal is received from the convolution process of input and the designed filter. Comparisons are done with the expected output and the filter response with the use of frequency domain representations and time domain representations of input and output signals.

I. Filter Implementation

I.I Prescribed Filter Specifications

Required filter parameters for implementing the bandpass filter were calculated with the given filter specifications as given in the *Table (1)* below.

Specification	Symbol	Value
Maximum passband ripple	A_p	0.08 dB
Minimum stopband attenuation	A_a	47 dB
Lower passband edge	ω_{p1}	1200 rad/s
Upper passband edge	ω_{p2}	1600 rad/s
Lower stopband edge	ω_{a1}	1050 rad/s
Upper stopband edge	ω_{a2}	1700 rad/s
Sampling frequency	ω_s	4200 rad/s

Table (1) – Prescribed Filter Specifications

I.II Derived Filter Specifications

In designing the digital bandpass filter and Kaiser window it is required some filter characteristics to be derived from the above prescribed filter specifications mentioned in *Table (1)*. The newly derived Filter specifications are listed in the *Table (2)* below.

Specification	Symbol	Derivation	Value
Lower transition width	B_{t1}	$\omega_{p1} - \omega_{a1}$	150 rad/s
Upper transition width	B_{t2}	$\omega_{a2} - \omega_{p2}$	100 rad/s
Critical transition width	B_t	min (B_{t1} , B_{t2})	100 rad/s
Lower cutoff frequency	ω_{c1}	$\omega_{p1} - \frac{B_t}{2}$	1150 rad/s
Upper cutoff frequency	ω_{c2}	$\omega_{p2} + \frac{B_t}{2}$	1650 rad/s
Sampling period	T	$\frac{2\pi}{\omega_s}$	0.0015 s

Table (2) – Derived Filter Specifications

I.III Derivation of the Kaiser Window Parameters

With the use of above derived filter specifications, Kaiser window can be obtained in a form as,

$$w_k(nT) = \begin{cases} \dfrac{I_0(\beta)}{I_0(\alpha)} & for\ |n| \le (N-1)/2 \\ 0 & Otherwise \end{cases}$$

Where $\beta = \alpha\sqrt{1-\left(\frac{2n}{N-1}\right)^2}$ and $I_0(\alpha) = 1 + \sum_{k=1}^{\infty}\left[\frac{1}{k!}\left(\frac{\alpha}{2}\right)^k\right]^2.$

Still, α and N remains unknown. We can define a parameter called δ such that the actual passband ripple,A_p is equal to or less than specified passband ripple $\tilde{A}_p$, and the actual minimum stopband attenuation,A_a is equal or greater than the specified minimum stopband attenuation, $\tilde{A}_a$.

A suitable value is received when,

$$\delta = \min(\delta_p,\ \delta_a)$$

Where $\delta_p = \frac{10^{0.05A_p}-1}{10^{0.05A_p}+1}$ and $\delta_a = 10^{-0.05A_a}$.

With the required δ defined, the actual stopband attenuation A_a can be calculated as,

$$A_a = -20\log(\delta)$$

Parameter α can be chosen as,

$$\alpha = \begin{cases} 0 & for\ A_a \le 21 \\ 0.5842(A_a-21)^{0.4} + 0.07886(A_a-21) & for\ 21 < A_a \le 50 \\ 0.07886(A_a-21) & for\ A_a > 50 \end{cases}$$

Then choose parameter D as in the below equation to obtain a value for N.

$$D = \begin{cases} 0.9222 & for\ A_a \le 21 \\ \dfrac{A_a - 7.95}{14.36} & for A_a > 21 \end{cases}$$

Then, the smallest odd integer value is chosen for N that is satisfying the inequality,

$$N \ge \frac{\omega_s D}{B_t} + 1 \qquad \text{where} \qquad B_t = \omega_a - \omega_p$$

The Kaiser window parameters obtained through above set of equations according to the derived filter specification are given in the *Table (3)* below.

Parameter	Value
δ	$\min(\delta_p,\ \delta_a)$ = min (0.0046, 0.0045) = 0.0045
A_a	47 dB
α	4.2009
D	2.7194
N	117

Table (3) – Kaiser Window Parameters

I.IV Derivation of the ideal impulse response

The frequency response for an ideal band pass filter with cut-off frequencies ω_{c1} and ω_{c2} are derived from,

$$H\left(e^{j\omega t}\right) = \begin{cases} 1 \;\; ; -\omega_{c2} \leq \omega \leq -\omega_{c1} \\ 1 \;\; ; \omega_{c2} \leq \omega \leq \omega_{c1} \\ 0 \;\; ; Otherwise \end{cases}$$

Ideal Impulse response of $H\left(e^{j\omega t}\right)$, $H[nT]$ can be obtained by using Inverse Fourier Transform as below.

$$H[nT] = \begin{cases} \dfrac{2}{\omega_s}(\omega_{c1} - \omega_{c2}) & for\ n = 0 \\ \dfrac{1}{n\pi}(\sin\omega_{c2}nT - \sin\omega_{c1}nT) & Otherwise \end{cases}$$

I.V Derivation of the causal impulse response of windowed filter

We obtain the finite order non causal impulse response of the windowed filter $H_w(nT)$ by multiplying the Ideal Impulse Response $\mathrm{h(nT)}$ by the Kaiser window function $w_k(nT)$.

$$H_w(nT) = w_k(nT).\mathrm{h(nT)}$$

The $H_w(nT)$ in time domain is the final filter. In the aim of doing comparisons, we are also obtaining a filter windowed with a rectangular window.

We obtain the Ƶ transform of $H_w(nT)$, $H_w(z)$ as,

$$H_w(z) = Ƶ[\, H_w(nT)]$$

$$H_w(z) = Ƶ[w_k(nT).\mathrm{h(nT)}]$$

Which upon shifting for causality becomes,

$$H'_w(z) = z^{-(N-1)/2} H_w(z)$$

II. Filter Evaluation

The performance of the generated filter can be evaluated using an input signal $\mathrm{x(nT)}$ which is a sum of three sinusoidal signals, each of which has a frequency related to lower stopband, passband and upper stopband as depicted in *Table (4)* below.

$$\mathrm{x(nT)} = \sum_{i=1}^{3} \sin(\omega_i nT)$$

Frequency component	Derivation	Value
ω_1	$\frac{\omega_{c1}}{2}$	575 rad/s
ω_2	$\frac{\omega_{c1} + \omega_{c2}}{2}$	1400 rad/s
ω_3	$\frac{\omega_{c2} + \omega_s/2}{2}$	1875 rad/s

Table (4) – Input Frequency Components

The $\sin(\omega_2 nT)$ component should be the ideal output as it is located inside the passband region while ω_1 and ω_3 components are located outside the passband. The output can be resulted by the convolution of filter impulse response $H_w(nT)$ and $\mathrm{x(nT)}$. In order to avoid the use of convolution operation, the Discrete Fourier Transform (DFT) of the two signals are obtained and by multiplying them in frequency domain, the frequency domain representation of the output is taken. Converting that frequency domain representation into time domain representation by Inverse Discrete Fourier Transform (IDFT) is help in avoiding the hard computation of convolution operation. There are respective methods in MATLAB programming environment to deal with these DFTs and IDFTs.

RESULTS

Results of this bandpass filter design can be demonstrated in two main forms. Initially, the set of frequency and impulse response plots will provide a reading on the characteristics of the filter design in each stage of the process. Then the performance of the filter in the filter evaluation stage can be demonstrated with a comparison between input and output signals of the filter.

I. Time and Frequency domain plots for the Filter

Given below are some plots obtained during the filter design process. *Figure (1)* below demonstrate the time domain representation (impulse response) of the Kaiser window.

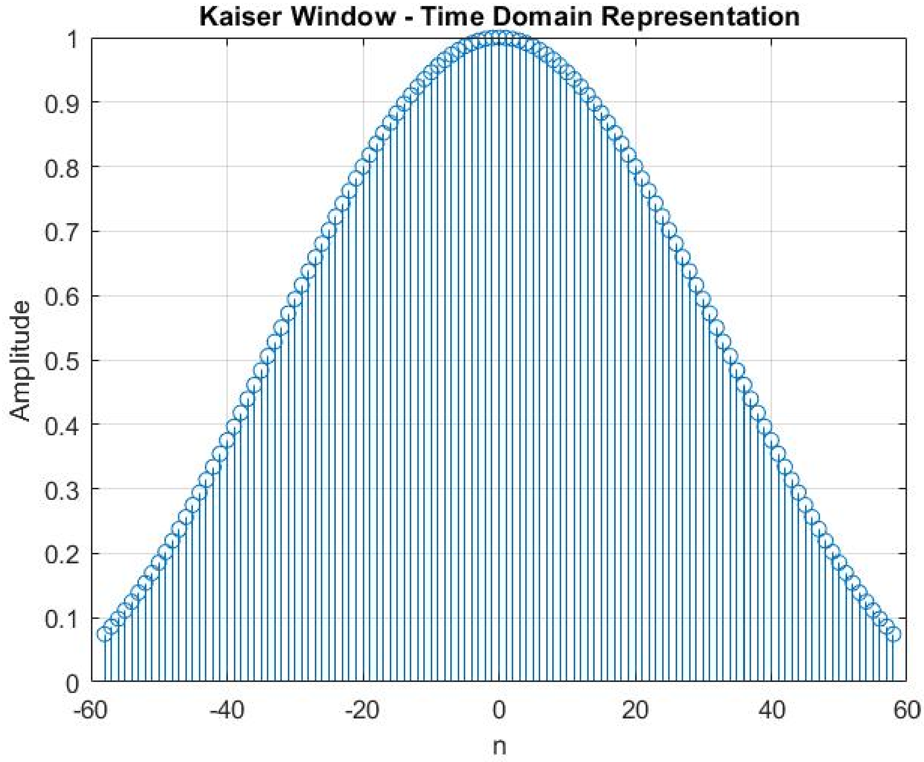

Figure 1 – Impulse response of Kaiser Window

The frequency domain representation of the filter when there is a rectangular window is depicted in the *Figure (2).*

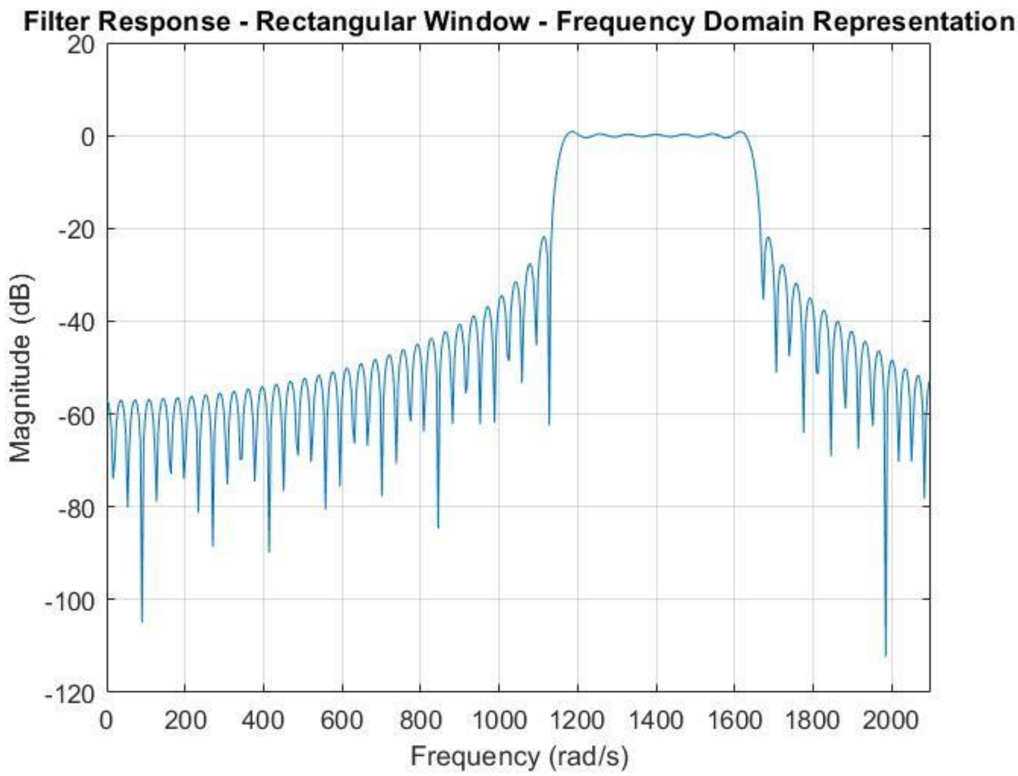

Figure (2) – Frequency domain representation of the filter with a rectangular window function

The impulse response of the filter with the above rectangular window function is given below under *Figure (3).*

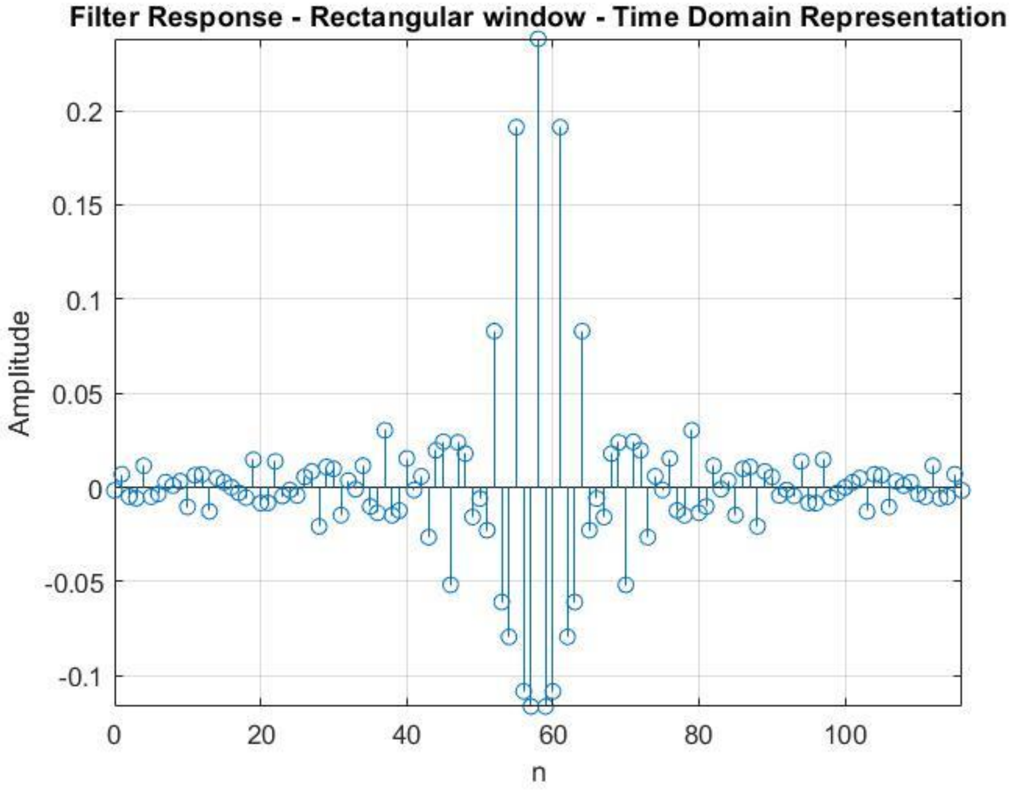

Figure (3) – Time domain representation of the filter with a rectangular window function

The frequency response of the filter with the use of a Kaiser window function is given below under *Figure (4).*

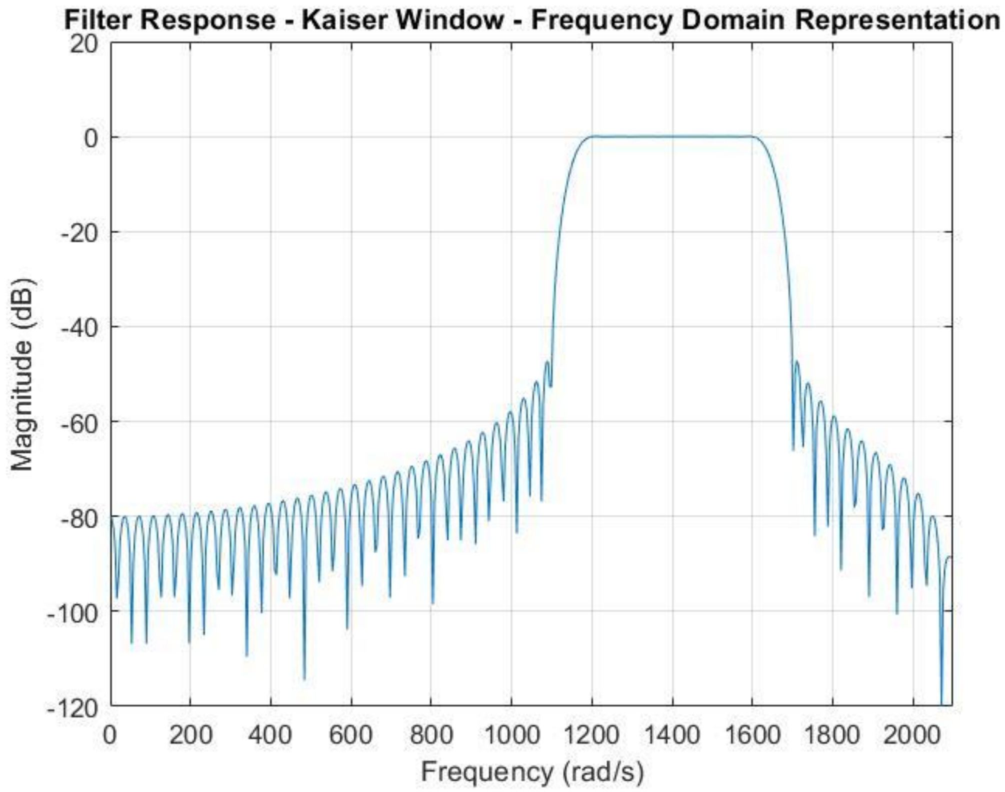

Figure (4) – Frequency domain representation of the filter with a Kaiser window function

The time domain representation of above plot is in *Figure (5)* below.

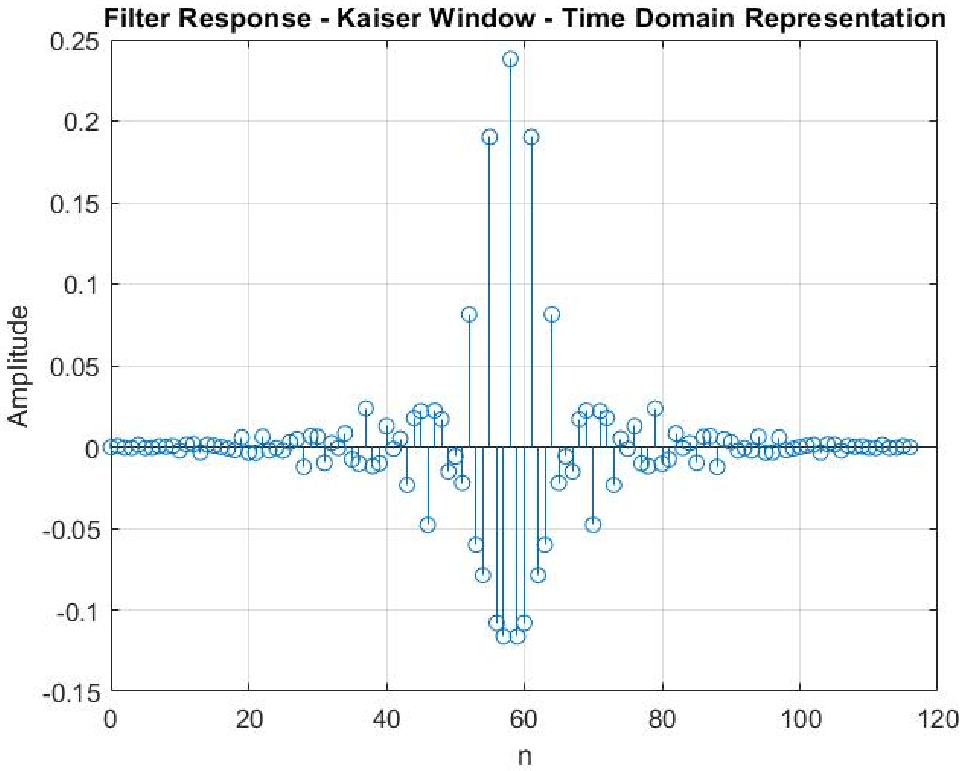

Figure (5) – Time domain Representation of the filter with a Kaiser window function

The passband of the above *Figure (4)* is zoomed to get a clear vision on pass band ripples. It is given in *Figure (6)* below.

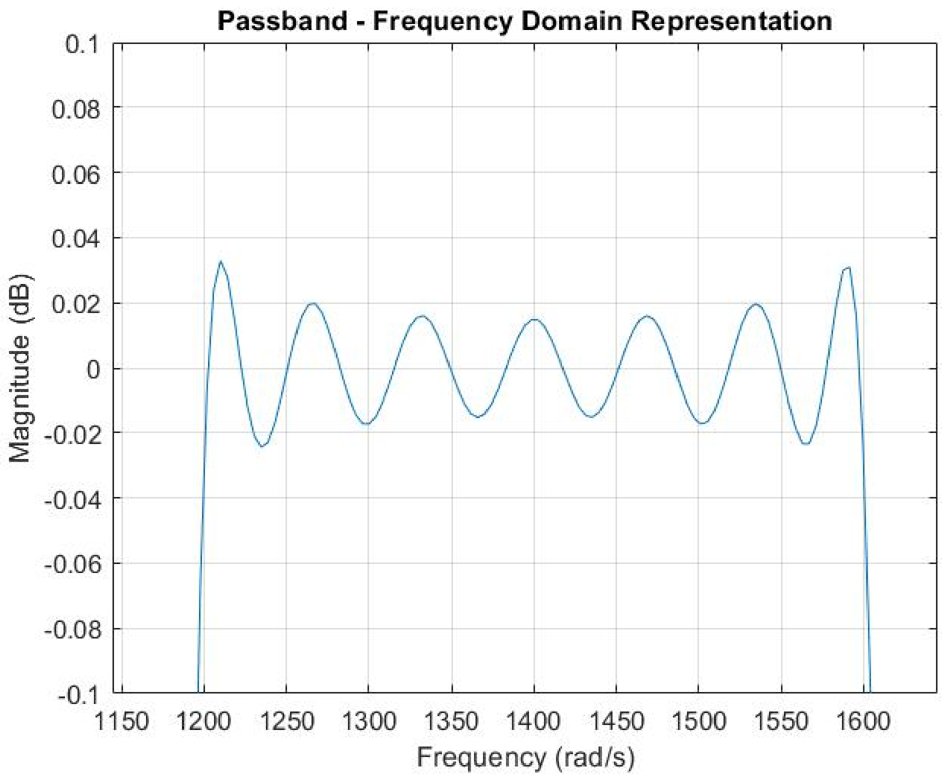

Figure (6) – Frequency Representation of the Zoomed pass band of the filter with a Kaiser window

II. The Input and Output signals of the Filter

Initially we are testing for few sample points in plotting the input signal and input signal which is used for ideal filtering with only the mid frequency. The plots are like the illustrations given under the *Figure (7)* below.

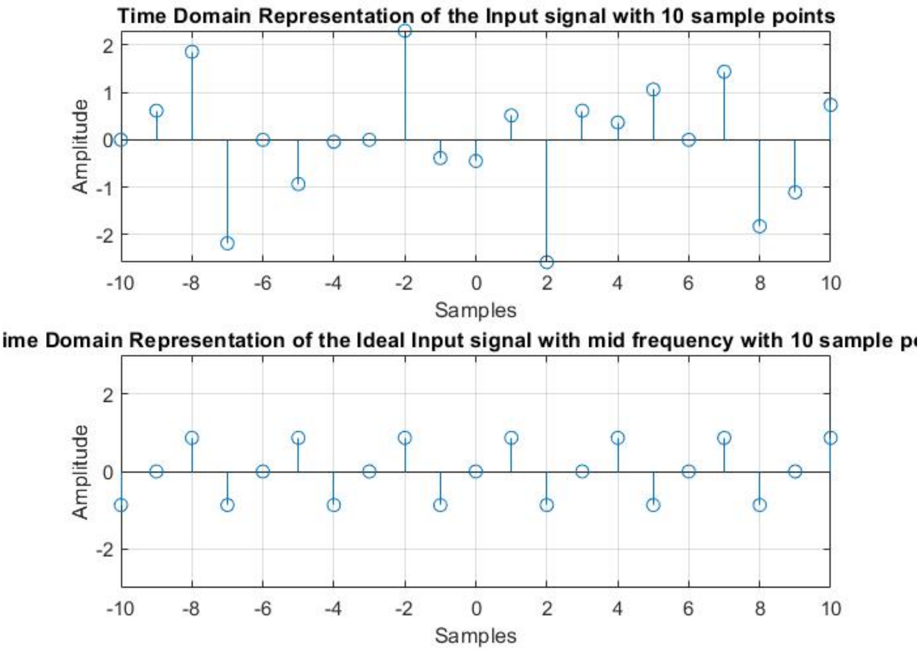

Figure (7) – Time Domain Representations of input signal and the input signal with mid frequency only with 10 samples.

It is given that to get a steady state response there must be 200 - 300 samples. Let us consider by taking 301 samples. The new input signal and input signal which is used for ideal filtering with only the mid frequency are as given below in *Figure (8).*

The

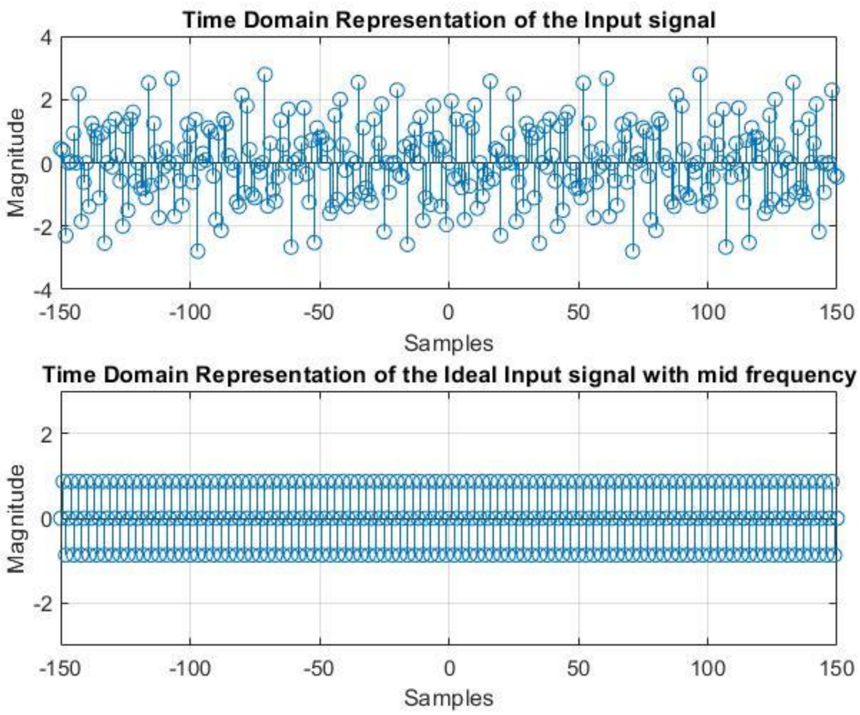

Figure (8) - Time Domain Representations of input signal and the input signal with mid frequency only with 301 samples.

Input output relationship of the filter is observed in relation to the time domain representations and frequency domain representations given in the *Figure (9)* and *Figure (10)* respectively. You will able to discover the similarity between the expected output which is hopefully received by the use of an ideal filter with unit gain in the passbands and 0 gain in the stopband.

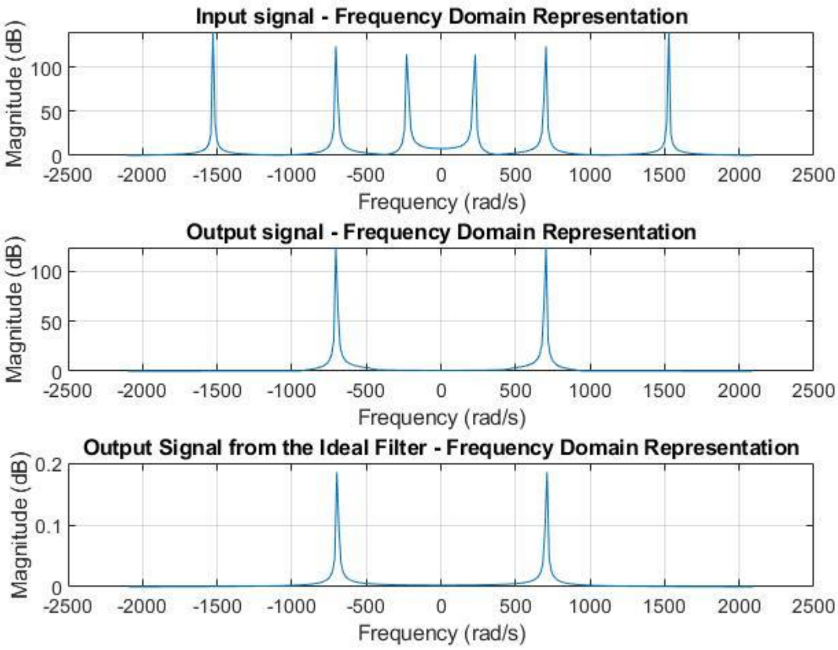

Figure (9) – Frequency Domain Representation of the input & output signals

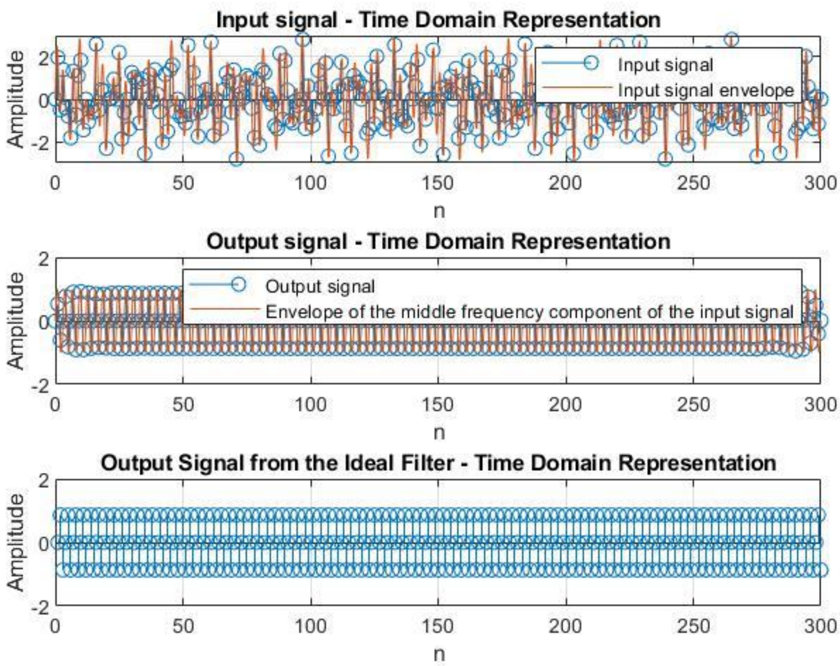

Figure (10) – Time Domain Representation of the input and output signals

To get a clear vision, it is very versatile zooming the above *Figure (11)* as below.

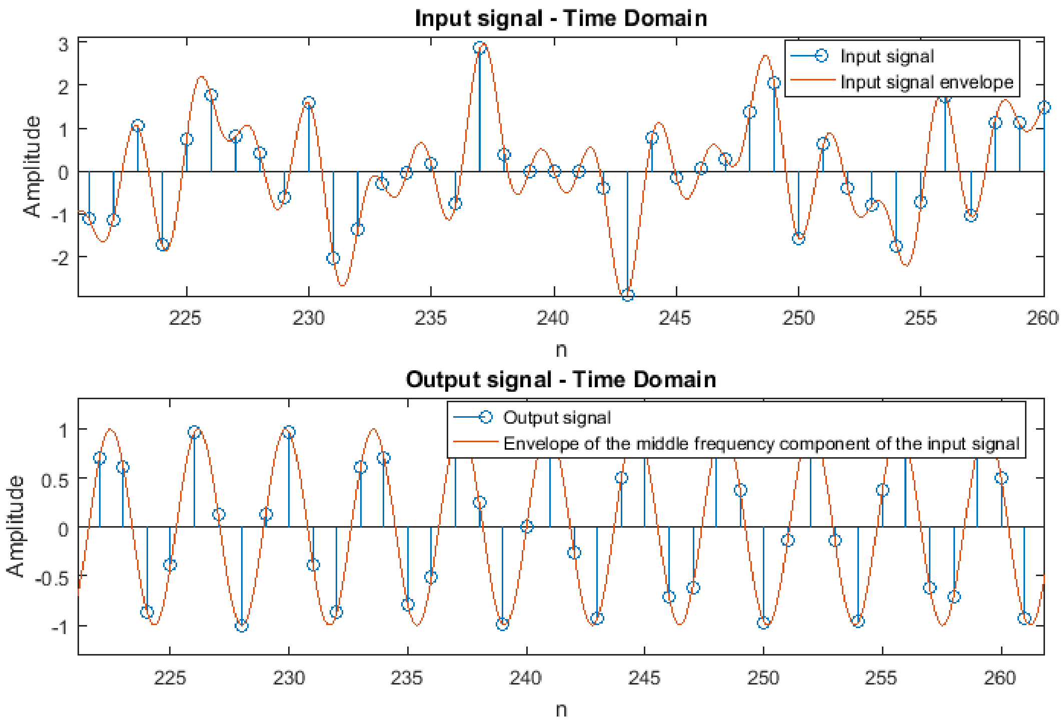

Figure (11) – Magnified Time Domain Representation of the input and output signals

According to the *Figure (11),* it is very clear that the inputs are sampled at T intervals which is depicted in blue color while the orange-colored wave line depicts the envelope. The envelope drawn in orange color in the output plot represents that the envelope of the input sinusoidal signal whose frequency component is expected to be in the passband region.

DISCUSSION

We are able to identify a significant difference between the filters obtained by Kaiser window and the rectangular window. It can be clearly noticed by inspecting the frequency domain representations in *Figure (2)* and *Figure (4).* There is a significantly high amount of ripple in the passband region of the rectangular windowed filter than the Kaiser windowed filter. The preserved stop band attenuation of -47 dB is only achieved in the Kaiser windowing. Th pass band ripple of Kaiser windowing is negotiable comparing it with that of rectangular windowing method. In Decibels, it is nearly on maximum 0.03 dB. It is clearly represented through *Figure (6).* At the edge of the windows Kaiser windowing gradually attenuate the samples while rectangular method is truncating the impulse response abruptly. It can be clearly seen through the time domain representations in *Figure (2)* and *Figure (4).* This will result in considering the Kaiser windowing filter more and more smooth. As given in the *Figure (11),* the frequency response of the filter has clearly demonstrated the stop band components with frequencies 575 rad/s and 1875 rad/s have been attenuated while the 1400 rad/s component will pass via the filter. This will provide a very versatile way in testing filter designs. Whenever the output is of single frequency valued the time domain representation of the input and output signals will forecast the behavior.

The flexibility of the Kaiser window is depicted through above results. As the ideal filters are never implemented practically, it is very important to be being able to control the limitations very easily. Since small imperfections like passband ripple may not cause a considerable difference in filtered output this is a good practical approach. We are able to design filters as very much close to the ideal filters through this Kaiser windowing until difference is indistinguishable.

CONCLUSION

The filter designed using the Kaiser window method shows a approximation to the filter response of an ideal filter. This shows that the Kaiser window method is a good practically acceptable approach. Designing a FIR filter can be easily done with the help of Kaiser window method very easily under the closed form direct approach. The designing process accommodates number of designing requirements due to its flexibility. The computational effort of designing a filter is very low in this method. The main drawback associated with this method is that is the requirement of high order of filter. There may be optimal filters from lower orders can easily achieve the specifications. Hardware implementation of high order filters may show a high inefficiency due to the use of many unit delays and multipliers. Software implementation of high order filters may require more computations per sample. But optimality can be obtained by using many advanced design techniques.

ACKNOWLEDGEMENT

This Design project was completed as a partial fulfillment of the module EN2570 – Digital Signal Processing. I should pay my kind gratitude to the module coordinating lecturer Dr.Chamira Edirisooriya, for inspiring me to do this task. It would never be successful without your contribution in building a good theoretical basis necessary for this project on Finite Impulse Response Digital Filter Design during the lectures. I would like to pay my homage to my batchmates also for helping me to complete this task successfully.

REFERENCES

- "Digital Signal Processing: Signals, Systems, and Filters," *Uvic.ca*, 2021. https://ece.engr.uvic.ca/~dsp/ (accessed Mar. 01, 2021).

- Wikipedia Contributors, "Kaiser window," *Wikipedia*, Feb. 25, 2021. https://en.wikipedia.org/wiki/Kaiser_window (accessed Mar. 02, 2021).

APPENDIX

MATLAB code for the software implementation

This is an attachment of the codes that were used in designing the finite impulse response digital bandpass filter and stimulation and validating the results. All the codes are compatible with MATLAB 2018a software version.

1. Time and the Frequency Domain Plots for the Filter

1.1 Prescribed Filter Specifications

1.2 Derived Filter Specifications

1.3 Kaiser Window Parameters

1.4 Computing I($\boldsymbol{\alpha}$)

1.5 Computing I($\boldsymbol{\beta}$)

1.6 Plotting Kaiser Window

1.7 Generating Causal Impulse Response

1.8 Applying Window to the filter

1.9 Plotting the Passband

2. Input Output Relationship of the filter

2.1 Generation of Input signals

- Time Domain Representation of the input signal with 10 sample points.
- Time Domain Representation of the input signal with mid frequency with 10 sample points.
- Plotting the input signal in Time Domain.
- Plotting the ideal input signal in Time Domain.

2.2 Checking the filtering with Discrete Fourier Transform

- Filtering using frequency domain multiplication.
- Frequency domain representation of the input signal.
- Frequency domain representation of the output signal from the BPF.
- Frequency domain representation of the output signal from the ideal BPF.
- Time domain representation of input signal.
- Time domain representation of the output signal from the BPF.
- Time domain representation of the output signal from the ideal BPF.

EN2570 - Digital Signal Processing

Design Project

Design of a Finite Duration Impulse Response (FIR)

Bandpass Filter

Name : Rathnayaka R.G.H.V.

Index Number : 180529E

Prescribed Filter Specifications

```
A = 5;
B = 2;
C = 9;

A_P = 0.03+0.01*A; %maximum passband ripple (dB)
A_a = 45+B; %minimum stopband attenuation (dB)
wp1 = C*100+300; %lower passband edge (rad/s)
wp2 = C*100+700; %upper passband edge (rad/s)
wa1 = C*100+150; %lower stopband edge (rad/s)
wa2 = C*100+800; %upper stopband edge (rad/s)
ws = 2*(C*100+1200); %sampling freqency (rad/s)
```

Derived Filter Specifications

```
Bt1 = wp1-wa1; %lower transition width (rad/s)
Bt2 = wa2-wp2; %upper transisiton width (rad/s)
Bt = min(Bt1,Bt2); %critical transition width (rad/s)
wc1 = wp1-Bt/2; %lower cutoff frequency (rad/s)
wc2 = wp2+Bt/2; %upper cutoff frequency (rad/s)
T = 2*pi/ws; %sampling period (s)
```

Kaiser Window Parameters

```
delta_P = (10^(0.05*A_P) - 1)/ (10^(0.05*A_P) + 1); %computing deltaP
delta_A = 10^(-0.05*A_a); %computing deltaA
delta = min(delta_P,delta_A); %computing delta

Aa = -20*log10(delta); %Actual stopband attenuation

if Aa <= 21 %Calculating alpha
    alpha = 0;
elseif Aa > 21 && Aa <= 50
    alpha = 0.5842*(Aa-21)^0.4 + 0.07886*(Aa-21);
else
```

```
    alpha = 0.1102*(Aa-8.7);
end

if Aa <= 21 %Calculating D
    D = 0.9222;
else
    D = (Aa-7.95)/14.36;
end

N = ceil(ws*D/Bt +1); %Order of the filter
if mod(N,2) == 0
    N = N+1;
end

n = -(N-1)/2:1:(N-1)/2; %Length of the filter

beta = alpha*sqrt(1-(2*n/(N-1)).^2); %Computing Beta
```

Computing I(α)

```
bessel_limit = 50;
Ialpha = 1;
for k = 1:bessel_limit
    term_k = (1/factorial(k)*(alpha/2).^k).^2;
    Ialpha = Ialpha + term_k;
end
```

Computing I(β)

```
Ibeta = 1;
for k = 1:bessel_limit
    term_k = (1/factorial(k)*(beta/2).^k).^2;
    Ibeta = Ibeta + term_k;
end
```

Plotting Kaiser Window

```
wknt = Ibeta/Ialpha;

figure
stem(n,wknt)
grid on;
xlabel('n')
ylabel('Amplitude')
title('Kaiser Window - Time Domain Representation');
```

Generating Causal Impulse Response

```
nleft = -(N-1)/2:-1;
hntleft = 1./(nleft*pi).*(sin(wc2*nleft*T)-sin(wc1*nleft*T));
nright = 1:(N-1)/2;
hntright = 1./(nright*pi).*(sin(wc2*nright*T)-sin(wc1*nright*T));
hnt0 = 2/ws*(wc2-wc1);
hnt = [hntleft,hnt0,hntright];
n_shifted = (0:1:N-1);
figure
stem(n_shifted,hnt);axis tight;
grid on;
xlabel('n')
ylabel('Amplitude')
title('Filter Response - Rectangular window - Time Domain Representation');
```

```
figure
[h,w] = freqz(hnt);
w = w/T;
h = 20*log10(abs(h));
plot(w,real(h))
axis([0,2100,-120,20]);
grid on;
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Filter Response - Rectangular Window - Frequency Domain Representation');
```

Applying the window to the filter

```
filter = hnt.*wknt;
figure
stem(n_shifted,filter)
grid on;
xlabel('n')
ylabel('Amplitude')
title('Filter Response - Kaiser Window - Time Domain Representation');
```

```
figure
[h,w] = freqz(filter);
w = w/T;
h = 20*log10(abs(h));
plot(w,h)
```

```
axis([0,2100,-120,20]);
grid on;
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Filter Response - Kaiser Window - Frequency Domain Representation');
```

Plotting the Passband

```
figure
start = round(length(w)/(ws/2)*wc1);
finish = round((length(w)/(ws/2)*wc2));
wpass = w(start:finish);
hpass = (h(start:finish));
plot(wpass,hpass)
grid on;
axis([-inf, inf, -0.1, 0.1]);
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Passband - Frequency Domain Representation');
```

Generation of Input signals

```
w1 = wc1/2; %component frequencies of the input
w2 = wc1 + (wc2-wc1)/2;
w3 = wc2 + (ws/2-wc2)/2;
wi = [w1,w2,w3];
```

Time Domain Representation of the input signal with 10 sample points

```
showSamples = 10;
figure
subplot(2, 1, 1)
stem((-showSamples:showSamples), xnt(151-showSamples:151+showSamples))
title('Time Domain Representation of the Input signal with 10 sample points')
xlabel('Samples')
ylabel('Amplitude')
grid on;
```

Time Domain Representation of the ideal input signal with mid frequency with 10 sample points

```
subplot(2, 1, 2)
stem((-showSamples:showSamples), x_out_ideal_filter(151-showSamples:151+showSamples))
title('Time Domain Representation of the Ideal Input signal with mid frequency with 10 sample p
xlabel('Samples')
ylabel('Amplitude')
ylim([-3 3])
grid on;
```

```
n1 = 0:1:300; %x axis for discrete signal
n2 = 0:0.1:300; %x axis for envelope
samples = 301;

xnt = zeros(1, samples);
x_out_ideal_filter = zeros(1, samples);
for k=1:samples
    n = k - ceil(samples/2);
    nT = n * T;
    xnt(k) = sin(w1 * nT) + sin(w2 * nT) + sin(w3 * nT);
    x_out_ideal_filter(k) = sin(w2 * nT);
end
```

Plotting the input signal in Time Domain

```
figure, subplot(2, 1, 1)
stem((1-ceil(samples/2):samples-ceil(samples/2)), xnt);
title('Input signal')
ylabel('Magnitude')
xlabel('Samples')
title('Time Domain Representation of the Input signal')
grid on;
```

Plotting the ideal input signal in the Time Domain

```
subplot(2, 1, 2)
stem((1-ceil(samples/2):samples-ceil(samples/2)), x_out_ideal_filter);
title('Ideal signal')
ylabel('Magnitude')
xlabel('Samples')
ylim([-3 3])
title('Time Domain Representation of the Ideal Input signal with mid frequency')
grid on;
```

Checking the filtering with Discrete Fourier Transform

Filtering using frequency domain multiplication

```
xnt = 0;
xdash = 0;
for each = wi
    xnt = xnt+ sin(each.*n1.*T); %generate discrete signal
    xdash = xdash+sin(each.*n2.*T); %generate envelope
end
Npoint = length(xnt) + length(filter) - 1; %length for fft in x dimension
xfft = fft(xnt,Npoint);
filterfft = fft(filter,Npoint);
outfft = filterfft .* xfft;
out = ifft(outfft,Npoint);
out1 = out(floor(N/2)+1:length(out)-floor(N/2)); %shifting delay
```

Frequency domain representation of input signal

```
figure
subplot(3,1,1)
Npoint = length(xnt);
xfft = fft(xnt,Npoint);
x1 = n1/length(n1)*ws-ws/2;
plot(x1,abs(xfft))
grid on;
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Input signal - Frequency Domain Representation');
```

Frequency domain representation of Output signal from the BPF

```
subplot(3,1,2)
Npoint = length(out1);  % length for fft in x dimension
xffout = fft(out1,Npoint);
x1 = n1/length(n1)*ws-ws/2;
plot(x1,abs(xffout))
grid on;
xlabel('Frequency (rad/s)')
ylabel('Magnitude (dB)')
title('Output signal - Frequency Domain Representation');
```

Frequency domain representation of Output signal from the ideal BPF

```
[~, c] = size( abs( fft(x_out_ideal_filter) ) );
y = (-floor(c/2):floor(c/2))/c*w_s;
subplot(3,1,3);
plot(y ,  abs( T*fft(x_out_ideal_filter) ) )
title('Output Signal from the Ideal Filter - Frequency Domain Representation')
ylabel('Magnitude (dB)')
xlabel('Frequency (rad/s)')
grid on;
```

time domain representation of input signal

```
figure
subplot(3,1,1)
stem(n1,xnt)
xlabel('n')
ylabel('Amplitude')
title('Input signal - Time Domain Representation');
hold on
plot(n2,xdash)
grid on;
legend('Input signal','Input signal envelope');
```

Time domain representation of the Output signal from the BPF

```
subplot(3,1,2)
stem(n1,out1)
xlabel('n')
ylabel('Amplitude')
title('Output signal - Time Domain Representation');
hold on
plot(n2,sin(w2.*n2.*T))
xlim([0,300])
ylim([-2,2])
grid on;
legend('Output signal','Envelope of the middle frequency component of the input signal');
```

Time Domain Representation of the Output signal from the ideal BPF

```
subplot(3,1,3);
```

```
stem(x_out_ideal_filter);
title('Output Signal from the Ideal Filter - Time Domain Representation')
ylabel('Amplitude')
xlabel('n')
xlim([0,300])
ylim([-2 2])
grid on;
```