Signals Systems and Communication Lab

Laboratory report submitted for the partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Electronics and Communication Engineering

by

Mohit Akhouri - 19ucc023

Course Coordinator
Dr. Akash Gupta

Department of Electronics and Communication Engineering
The LNM Institute of Information Technology, Jaipur

February 2021

Contents

Chapter *5*

Experiment - 5

5.1 Aim of the experiment

1. To verify the convolution property of Fourier transform
2. To verify time shifting property of Fourier transform
3. To verify frequency shifting property of Fourier transform

5.2 Software Used

MATLAB

5.3 Theory

5.3.1 About Fourier transform :

In mathematics, a Fourier transform (FT) is a mathematical transform that decomposes functions depending on space or time into functions depending on spatial or temporal frequency, such as the expression of a musical chord in terms of the volumes and frequencies of its constituent notes. The term Fourier transform refers to both the frequency domain representation and the mathematical operation that associates the frequency domain representation to a function of space or time.

The Fourier transform of a function of time is a complex-valued function of frequency, whose magnitude (absolute value) represents the amount of that frequency present in the original function, and whose argument is the phase offset of the basic sinusoid in that frequency. The Fourier transform is not limited to functions of time, but the domain of the original function is commonly referred to as the time domain. There is also an inverse Fourier transform that mathematically synthesizes the original function from its frequency domain representation, as proven by the Fourier inversion theorem.

5.3.2 Convolution property of Fourier transform :

The convolution of two functions $x_1(t)$ and $x_{2](t)}$ is written as h(t)=x_1*x_2 defined by :

$$h(t) = \int_{-\infty}^{\infty} x_1(\tau)x_2(t-\tau) \tag{5.1}$$

The convolution property states that convolution in time domain results in multiplication in frequency domain.

$$x_1(t) * x_2(t) = X_1(\omega).X_2(\omega) \tag{5.2}$$

5.3.3 Time shifting property of Fourier transform :

Time shifting property is as follows :

$$x(t) <-> X(\omega) \tag{5.3}$$

$$x(t-t_o) <-> e^{-j\omega t_o}X(\omega) \tag{5.4}$$

5.3.4 Frequency shifting property of Fourier transform :

Frequency shifting property is as follows :

$$x(t) <-> X(\omega) \tag{5.5}$$

$$e^{j\omega t_o}.x(t) <-> X(\omega-\omega_o) \tag{5.6}$$

5.4 Code and Results

5.4.1 Exercise 1 : Verifying convolution property of Fourier transform

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% This code will verify the convolution property of Fourier transform

to = 25;
ts = 1/10;
N = to/ts;
t = ts:ts:to; % defining range for t
fs = 1/ts;
f = linspace(-fs,fs,N);
xt = (t>=0 & t<10); % defining first pulse x(t)
ht = (t>=0 & t<10); % defining second pulse h(t)

figure;
subplot(2,1,1);
plot(t,xt,'Linewidth',1.5);
xlabel('Time(t)->');
ylabel('x(t) ->');
title('Plotting rectangular signal x(t) of length 10');
grid on;
subplot(2,1,2);
plot(t,ht,'Linewidth',1.5);
xlabel('Time(t)->');
ylabel('h(t)->');
title('Plotting rectangular signal h(t) of length 10');
grid on;
sgtitle('19ucc023 - Mohit Akhouri');

% Finding convolution in time domain
conv_xh_t = conv(xt,ht); % using function conv to calculate
 convolution in time domain

% plotting figures;
figure;
subplot(2,1,1);
plot(conv_xh_t);
xlabel('time(t) ->');
ylabel('x(t) conv h(t)');
title('Convolution of x(t) and h(t) in time domain');
grid on;

% Finding convolution in frequency domain
conv_xh_f = abs(ifft(fft(xt).*fft(ht))); % taking inverse fft of
 multiplication of fft
% of x(t) and h(t) in frequency domain

% plotting figures;
subplot(2,1,2);
plot(conv_xh_f);
xlabel('frequency(Hz) ->');
```

Figure 5.1 Part 1 of Code for verification of convolution property of Fourier transform

```
ylabel('x(\omega) x h(\omega)');
title('Multiplication (Convolution) of x(\omega) and h(\omega) in
 Frequency domain');
grid on;
sgtitle('19ucc023 - Mohit Akhouri');
```

Figure 5.2 Part 2 of Code for verification of convolution property of Fourier transform

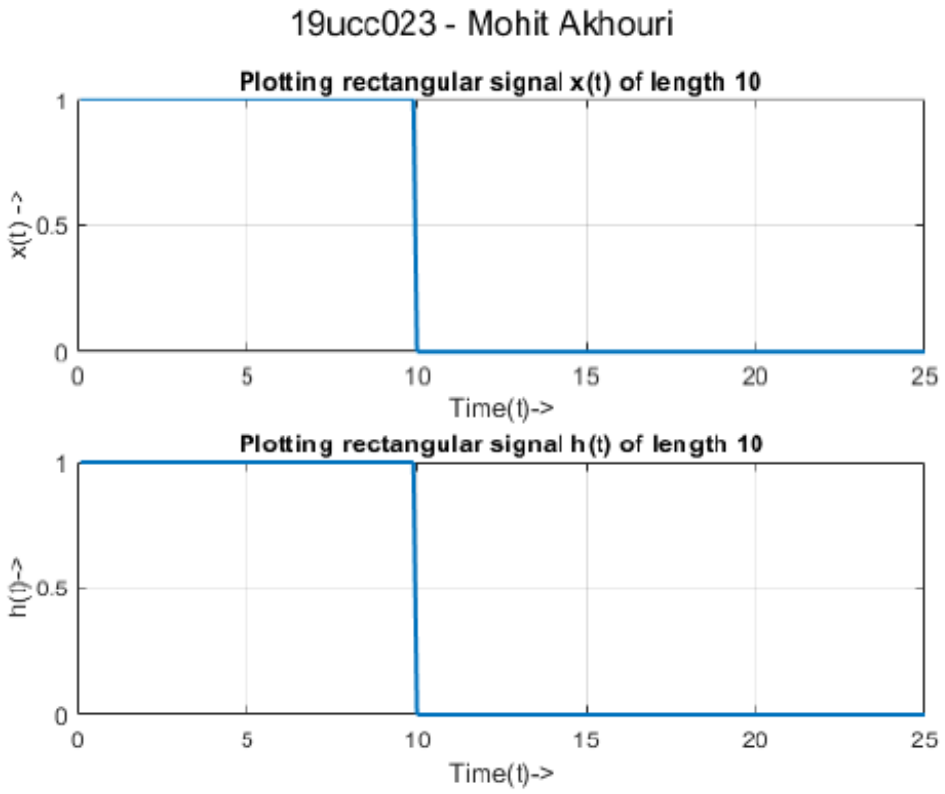

Figure 5.3 Graph of plotted pulses x(t) and h(t) of length 10

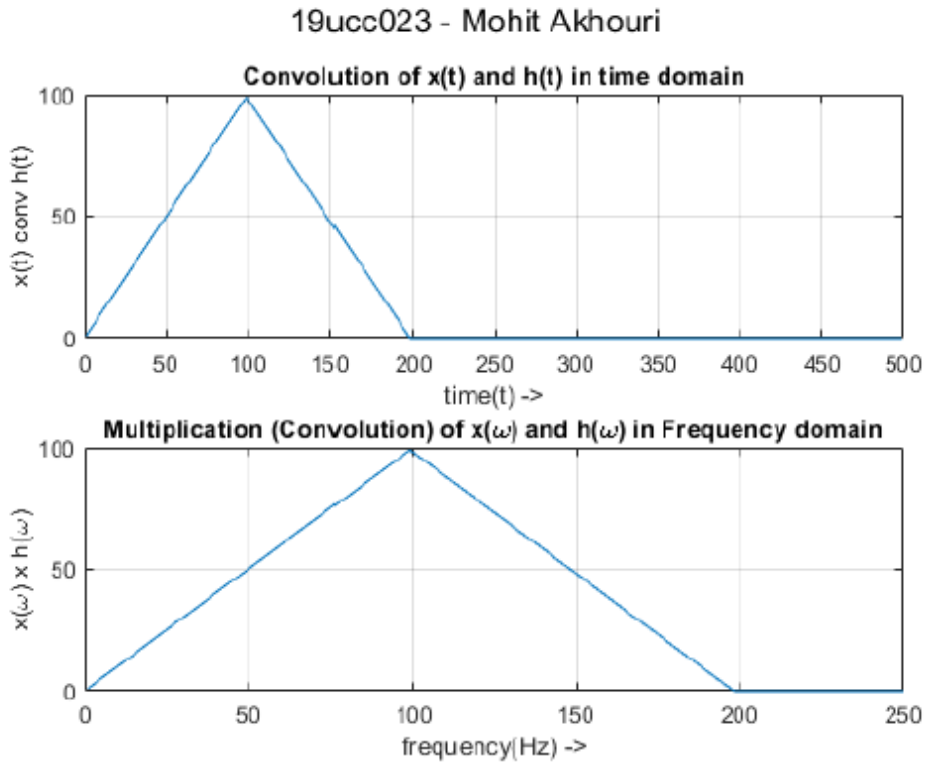

Figure 5.4 Graph of verification of convolution property of Fourier transform

5.4.2 Exercise 2: Verifying Time shifting property of Fourier transform

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% This code will verify the Time shifting property of Fourier
 transform

to = 25; % defining To
ts = 1/10; % defining Ts
N = to/ts; % defining range N
t = ts:ts:to; % defining range for t
fs = 1/ts; % defining frequency fs
f = linspace(-fs,fs,N); % defining frequency f
Xt = (t>=0 & t<10); % defining pulse X(t)

% plotting original signal X(t)
figure;
subplot(2,1,1);
plot(t,Xt,'Linewidth',1.5);
xlabel('Time(t)->');
ylabel('x(t) ->');
title('Plotting rectangular signal X(t) of length 10');
grid on;
sgtitle('19ucc023 - Mohit Akhouri');

To = 5; % defining value by which signal is to be plotted
Ft = fftshift(fft(Xt)); % taking fft of X(t)
shift_Ft = exp(-1j*2*pi*f*To).*Ft; % Shifting fourier transform
Shift_Ift = abs(ifft(shift_Ft)); % Taking inverse fourier transform
t1 = t + To;
Shift_Ift = Xt;

% plotting X(t-5)
figure;
subplot(2,1,1);
plot(t1,Shift_Ift,'Linewidth',1.5);
xlabel('time(t)->');
ylabel('X(t-5)->');
title('Shifting X(t) by 5 units to right, Plotting X(t-5)');
grid on;


To = -5; % defining value by which signal is to be plotted
Ft = fftshift(fft(Xt)); % taking fft of X(t)
shift_Ft = exp(-1j*2*pi*f*To).*Ft; % Shifting fourier transform
Shift_Ift = abs(ifft(shift_Ft)); % Taking inverse fourier transform
t2 = t + To;
Shift_Ift = Xt;

% plotting X(t+5)
subplot(2,1,2);
plot(t2,Shift_Ift,'Linewidth',1.5);
```

Figure 5.5 Part 1 of Code for verification of Time shifting property of Fourier transform

```
xlabel('time(t)->');
ylabel('X(t+5)->');
title('Shifting X(t) by 5 units to left, Plotting X(t+5)');
grid on;
sgtitle('19ucc023 - Mohit Akhouri');
```

Figure 5.6 Part 2 of Code for verification of Time shifting property of Fourier transform

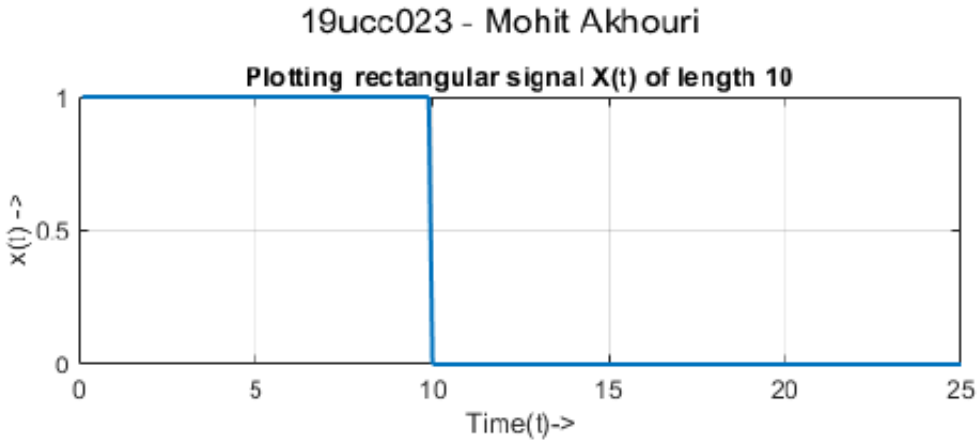

Figure 5.7 Graph of plotted pulse X(t) of length 10

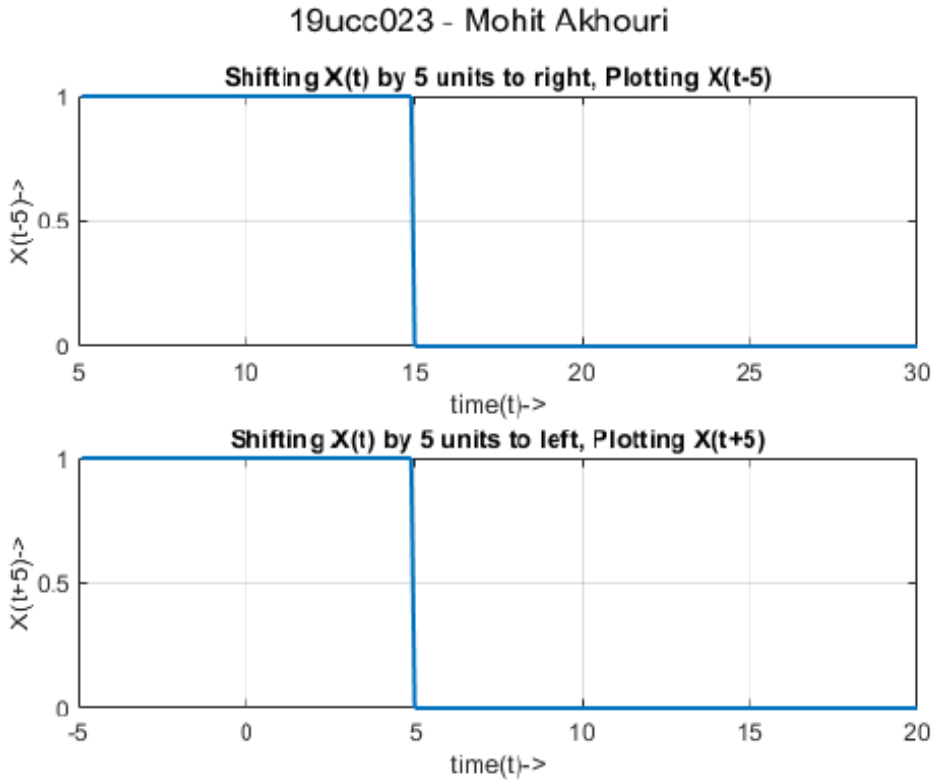

Figure 5.8 Graph of verification of Time shifting property by plotting X(t+5) and X(t-5)

5.5 Conclusion

In this experiment, we learnt about the Time shifting, Convolution and Frequency shifting property of Fourier transform and implemented functions for verification of Time shifting and Convolution property of Fourier transform.

Chapter *6*

Experiment - 6

6.1 Aim of the experiment

1. To add two sinusoidal signals
2. Eliminating the higher frequency component using fifth order low pass Butterworth filter
3. Eliminating the lower frequency component using fifth order high pass Butterworth filter

6.2 Software Used

MATLAB

6.3 Theory

6.3.1 About Butterworth filter:

The Butterworth filter is a type of signal processing filter designed to have a frequency response as flat as possible in the passband. It is also referred to as a maximally flat magnitude filter. It was first described in 1930 by the British engineer and physicist Stephen Butterworth in his paper entitled ”On the Theory of Filter Amplifiers”.
The frequency response of the Butterworth filter is maximally flat (i.e. has no ripples) in the passband and rolls off towards zero in the stopband. When viewed on a logarithmic Bode plot, the response slopes off linearly towards negative infinity. A first-order filter’s response rolls off at 6 dB per octave (20 dB per decade) (all first-order lowpass filters have the same normalized frequency response). A second-order filter decreases at 12 dB per octave, a third-order at 18 dB and so on. Butterworth filters have a monotonically changing magnitude function with , unlike other filter types that have non-monotonic ripple in the passband and/or the stopband.

Compared with a Chebyshev Type I/Type II filter or an elliptic filter, the Butterworth filter has a slower roll-off, and thus will require a higher order to implement a particular stopband specification, but

Butterworth filters have a more linear phase response in the pass-band than Chebyshev Type I/Type II and elliptic filters can achieve.

To use butterworth filter in MATLAB , use commands **butter(ord,Wn,'low')** for low pass filtering and **butter(ord,Wn,'high')** for high pass filtering and then pass them through function **filter()** to get the filtered output signal.

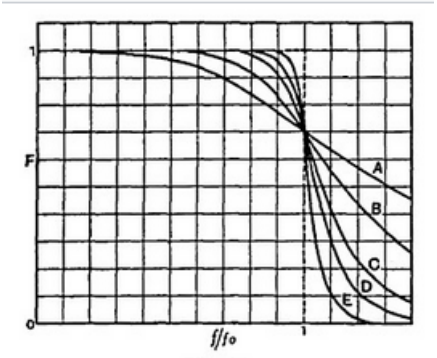

Figure 6.1 Frequency response plot from Butterworth's 1930 paper

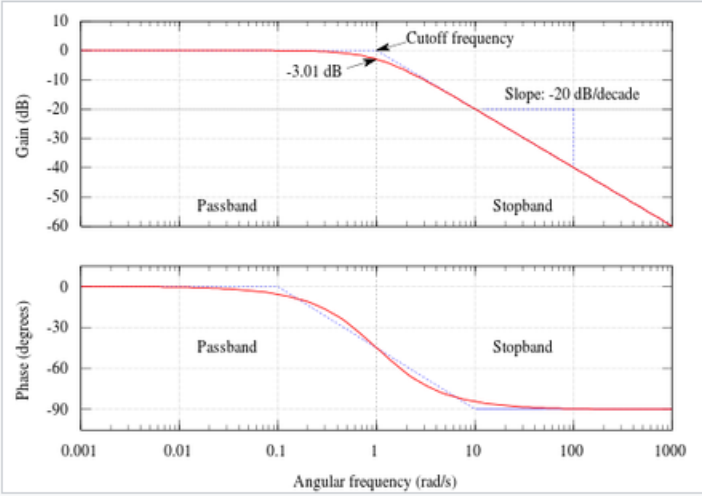

Figure 6.2 The Bode plot of a first-order Butterworth low-pass filter

6.4 Code and Results

6.4.1 Illustration of use of high pass and low pass Butterworth filter

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% This code will plot two sinusoidal signals and make use of
 butterworth
% filter

Fs = 50; % defining frequency Fs
Ts = 1/Fs; % defining Ts ( Sampling time period )
NyqFreq = Fs/2; % defining nyquist frequency
To = 10; % defining total duration ( To )
t = Ts:Ts:To; % defining time interval ( t )
N = length(t); % defining total number of samples N

% defining constants A1,A2,freq1,freq2
A1 = 20;
A2 = 30;
freq1 = 2;
freq2 = 20;

x1 = A1*sin(2*pi*freq1*t); % defining signal x1(t)
x2 = A2*sin(2*pi*freq2*t); % defining signal x2(t)

sum = x1+x2; % defining sum of x1(t) and x2(t)

% plotting figures - x1,x2 and sum
figure;
subplot(2,1,1);
plot(t,x1);
xlabel('time(t)->');
ylabel('x_{1}(t)->');
title('plotting x_{1}(t)');
grid on;
subplot(2,1,2);
plot(t,x2);
xlabel('time(t)->');
ylabel('x_{2}(t)->');
title('plotting x_{2}(t)');
sgtitle('19ucc023 - Mohit Akhouri');
grid on;

figure;
plot(t,sum);
xlabel('time(t)->');
ylabel('x_{1}(t) + x_{2}(t)->');
title('19ucc023 - Mohit Akhouri','Plotting the sum of x_{1}(t) and
 x_{2}(t)');
grid on;

f=linspace(-NyqFreq,NyqFreq,N); % defining range for frequency
Magn_Sum=fftshift(fft(sum,N)); % taking fft of signal sum and fftshift
```

Figure 6.3 Part 1 of Code for working of Butterworth filter and plotting of Frequency and Time domain spectra

```
Magn_Sum_Modified=10*log10(abs(Magn_Sum)); % taking log scale (dB
 scale) of fft of signal sum
figure;
% plotting the magnitude spectra of signal sum
plot(f,Magn_Sum_Modified);
xlabel('Frequency(Hz)->');
ylabel('Magnitude->');
title('19ucc023 - Mohit Akhouri','Magnitude plot of fourier transform
 of signal sum (x_{1}(t)+x_{2}(t))');
grid on;

f_low=2; % defining lower cutoff frequency
f_high=20; % defining upper cutoff frequency

Wn_low=(2*f_low/Fs); % defining Wn for lower cutoff frequency
Wn_high=(2*f_high/Fs); % defining Wn for upper cutoff frequency

order=5; % defining the order of butterworth filter

[b_low,a_low] = butter(order,Wn_low,'low'); % low pass butterworth
 filter
[b_high,a_high] = butter(order,Wn_high,'high'); % high pass
 butterworth filter

nv_low=filter(b_low,a_low,sum); % filtering low pass frequency
nv_high=filter(b_high,a_high,sum); % filtering high pass frequency

mag_lowpass=fftshift(fft(nv_low,N)); % taking magnitude of fft of low
 pass signal
mag_lowpass_modified=10*log10(abs(mag_lowpass)); % taking db scale of
 magnitude of fft of low pass signal

mag_highpass=fftshift(fft(nv_high,N)); % taking magnitude of fft of
 high pass signal
mag_highpass_modified=10*log10(abs(mag_highpass)); % taking db scale
 of magnitude of fft of high pass signal

figure;
% plotting low pass and high pass signals
subplot(2,1,1);
plot(f,mag_lowpass);
xlabel('Frequency (Hz)->');
ylabel('Magnitude->');
title('Magnitude plot of low pass butterworth signal');
grid on;
subplot(2,1,2);
plot(f,mag_highpass);
xlabel('Frequency (Hz)->');
ylabel('Magnitude->');
title('Magnitude plot of high pass butterworth signal');
grid on;
sgtitle('19ucc023 - Mohit Akhouri');
```

Figure 6.4 Part 2 of Code for working of Butterworth filter and plotting of Frequency and Time domain spectra

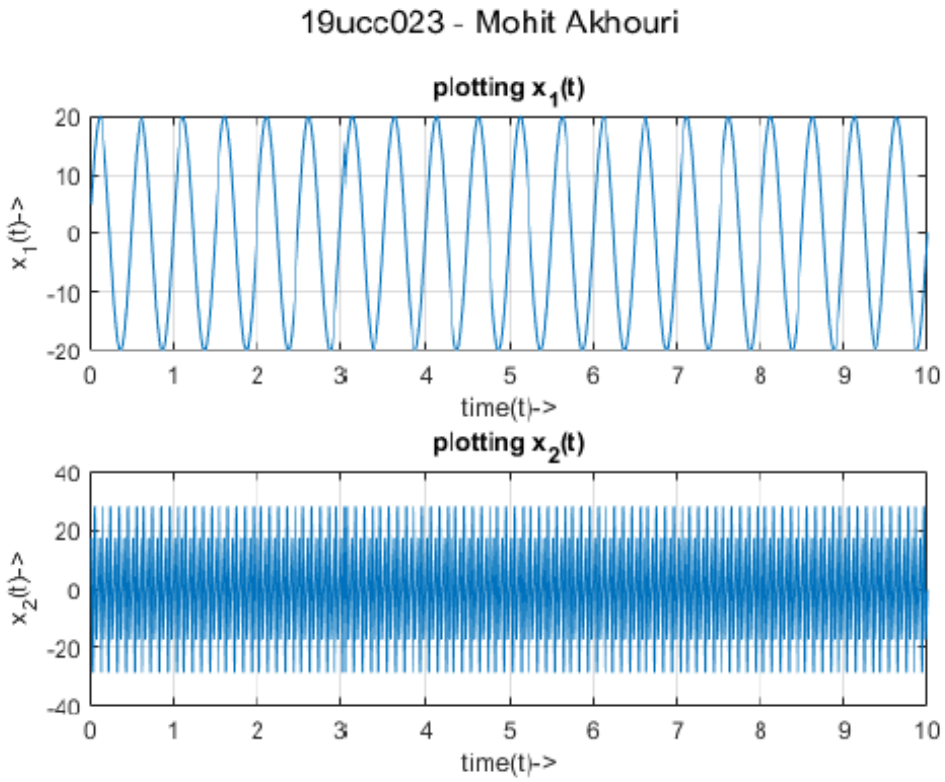

Figure 6.5 Graph of Plotted sinusoidal signals $x_1(t)$ and $x_2(t)$

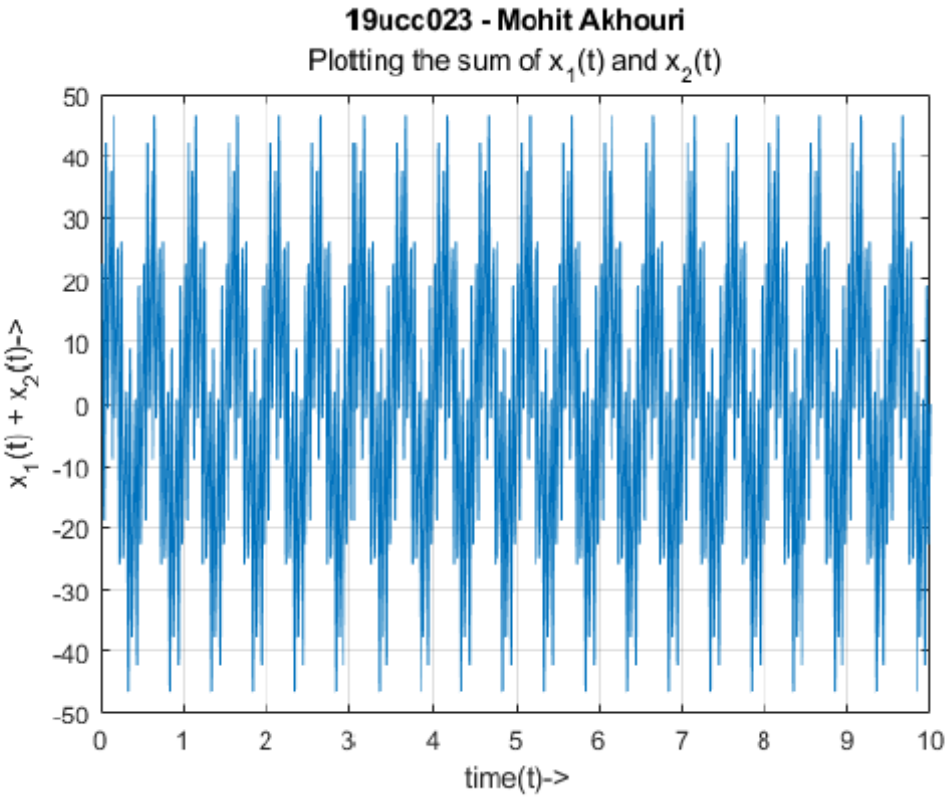

Figure 6.6 Graph of Plotted signal sum = $x_1(t)$ + $x_2(t)$

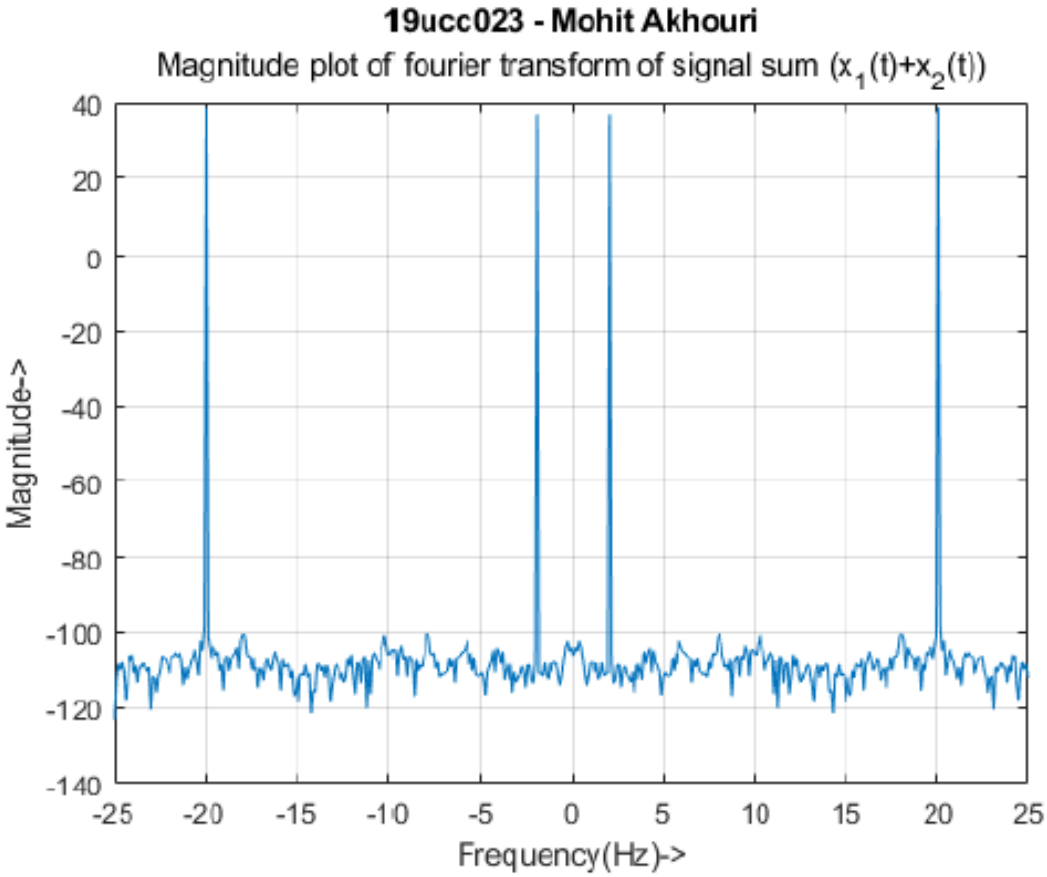

Figure 6.7 Magnitude vs. Frequency plot for the signal sum = $x_1(t)$ + $x_2(t)$

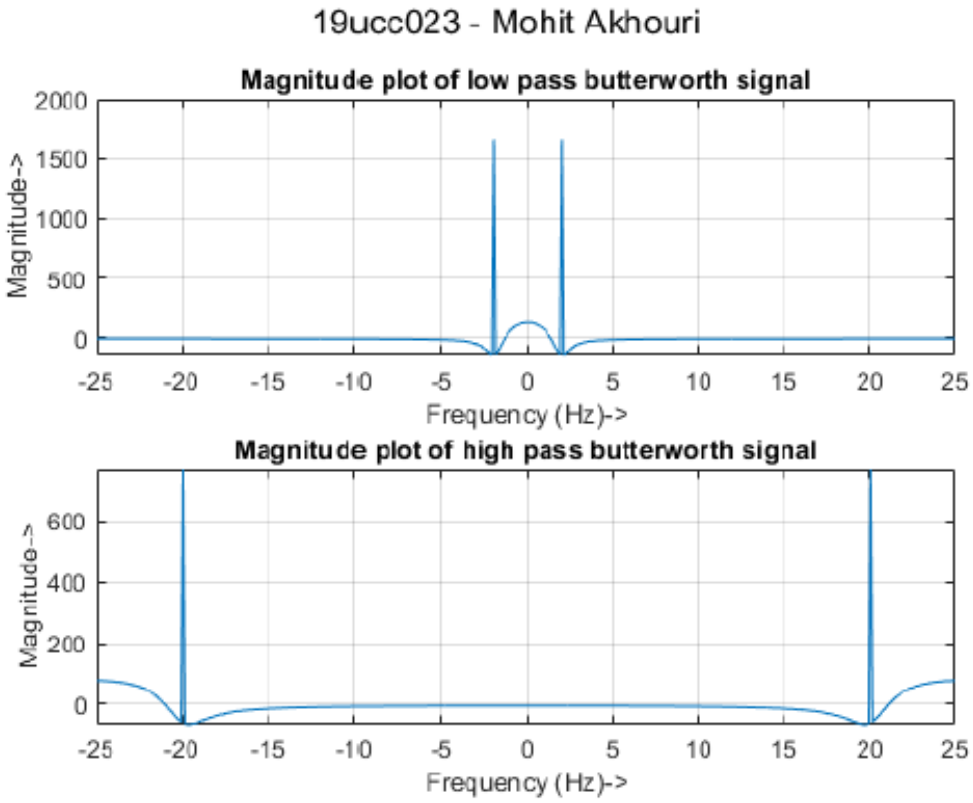

Figure 6.8 Graph of low pass and high pass signal obtained on passing signal **sum** through Butterworth filter

6.5 Conclusion

In this experiment, We have studied about the **sinusoidal signals** and how to plot them . We have used two frequencies for signals **2 Hz** and **20 Hz** and plotted their time domain representation. We calculated the sum of the two sinusoidal signals and plotted its time domain and frequency domain representation.

We also studied about **Low pass** and **High pass Butterworth filter** and how it can be used for filtering process, that is elimination of **lower frequency component** and **higher frequency component** . We studied MATLAB commands to use butterworth filter, **butter( )** and **filter( )**.