

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

Copyright © The LNMIIT 2021
All Rights Reserved

Contents

Chapter	Page
6 Experiment - 6	iv
6.1 Aim of the experiment	iv
6.2 Software Used	iv
6.3 Theory	iv
6.3.1 <u>About Butterworth filter:</u>	iv
6.4 Code and Results	vi
6.4.1 <u>Illustration of use of high pass and low pass Butterworth filter</u>	vi
6.5 Conclusion	xii

;

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.

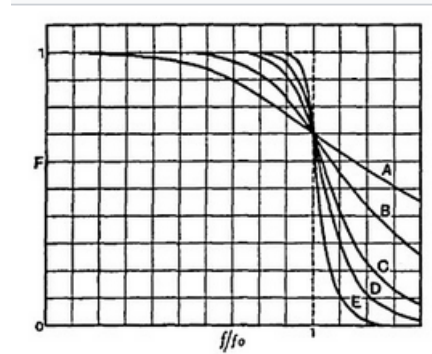


Figure 6.1 Frequency response plot from Butterworth's 1930 paper

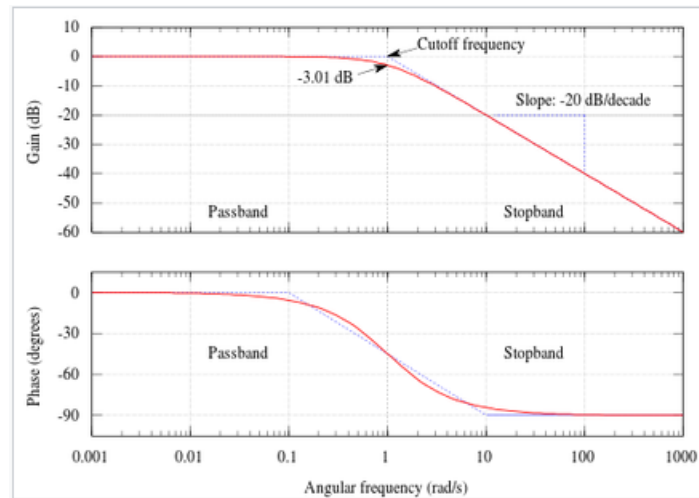


Figure 6.2 The Bode plot of a first-order Butterworth low-pass filter

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

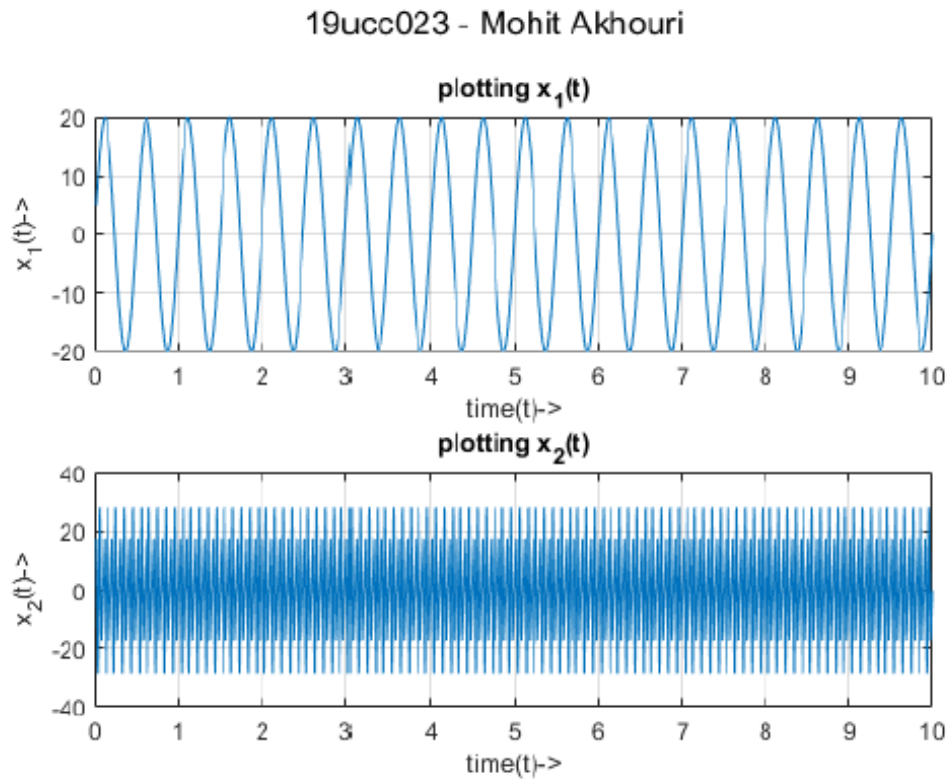


Figure 6.5 Graph of Plotted sinusoidal signals $x_1(t)$ and $x_2(t)$

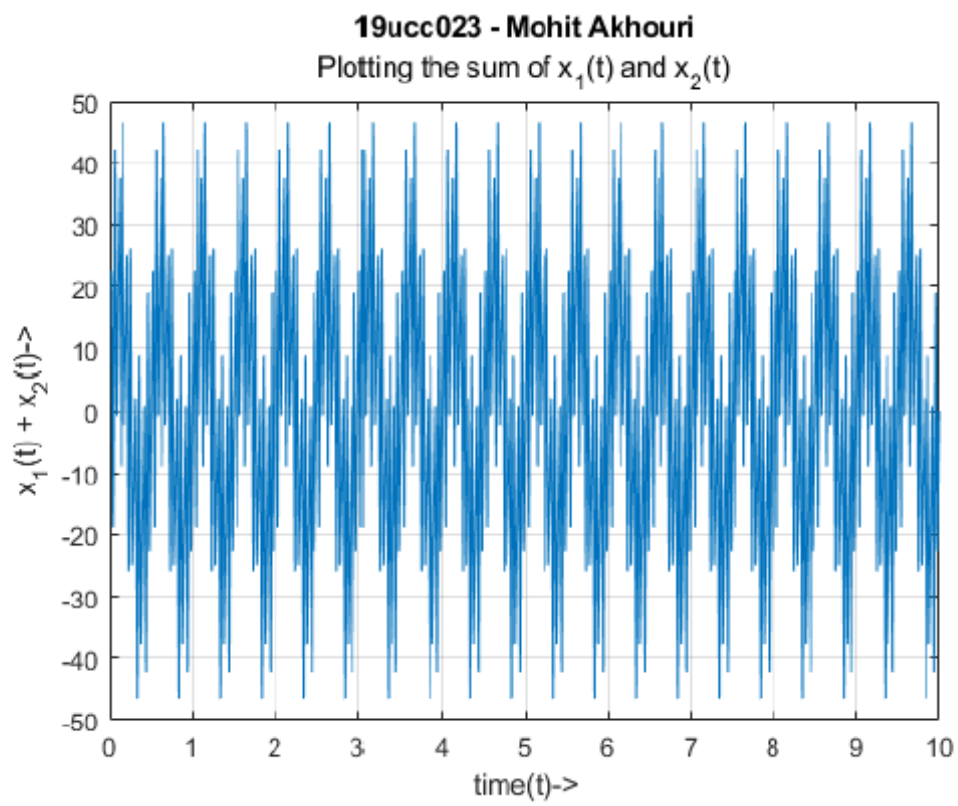


Figure 6.6 Graph of Plotted signal sum = $x_1(t) + x_2(t)$

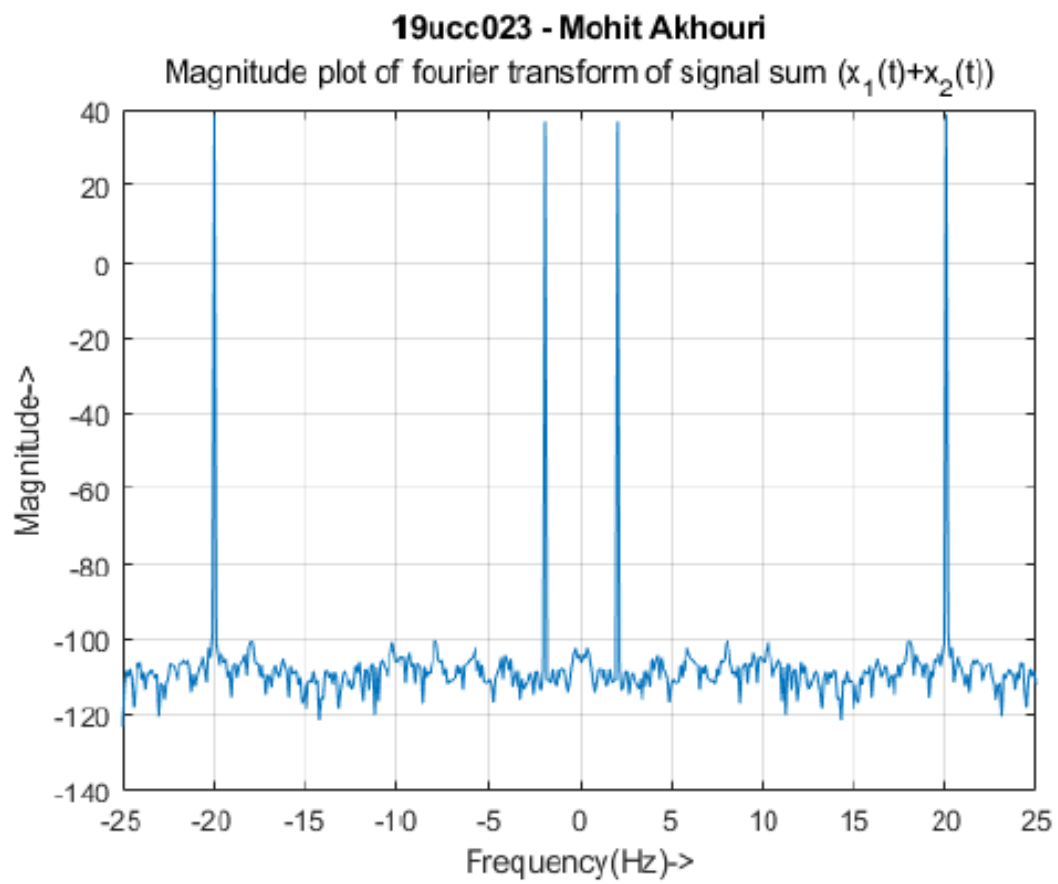
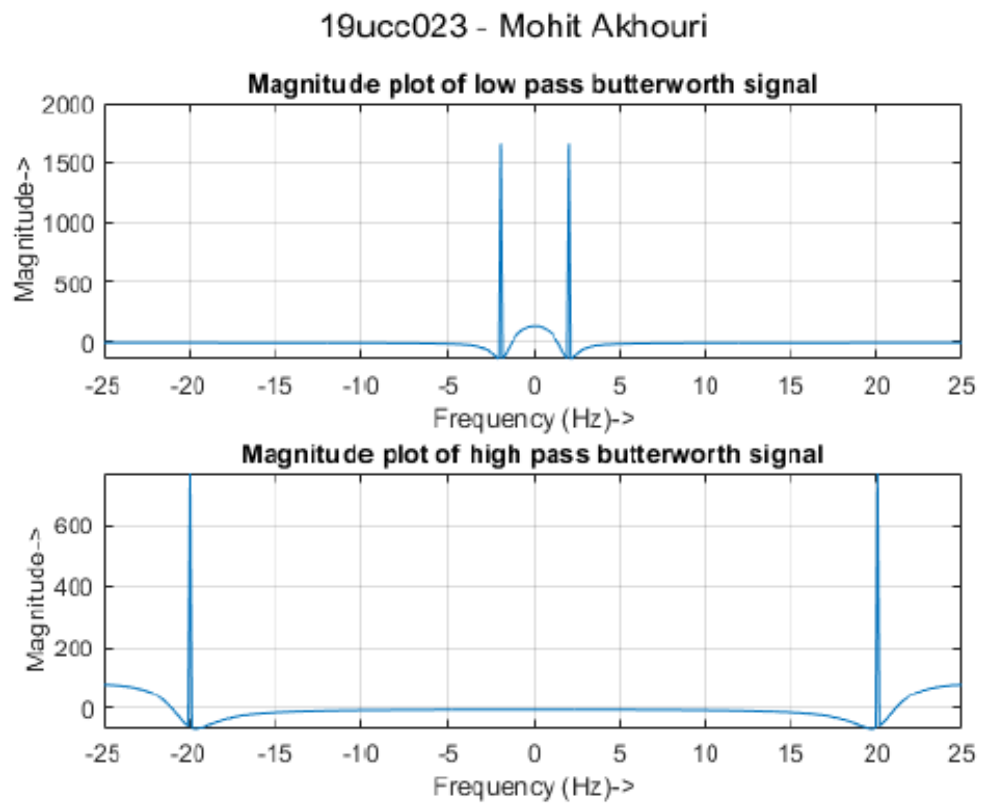


Figure 6.7 Magnitude vs. Frequency plot for the signal sum $= x_1(t) + x_2(t)$



Published with MATLAB® R2020b

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()**.