NAMAL UNIVERSITY MIANWALI
DEPARTMENT OF ELECTRICAL ENGINEERING

EE 345 (L) – Digital Signal Processing (Lab)
LAB # 10
REPORT

Title : FIR and IIR Filter Design in MATLAB

Name	*Fahim Ur Rehman Shah*
Roll No	*NIM-BSEE-2021-24*
Intructor	*Zulaikha Kiran*
Lab Engineer	*Engr. Faizan Ahmad*
Date Performed	*May 23, 2024*
Marks	

Introduction

The purpose of this lab is to enable the students to design FIR and IIR filters using MATLAB Filter Designer App, and apply it on a signal in their code.

Course Learning Outcomes

CLO1: Develop algorithms to perform signal processing techniques on digital signals using MATLAB and DSP Kit DSK6713

CLO3: Deliver a report/lab notes/presentation/viva, effectively communicating the design and analysis of the given problem

Equipment

- Software
  - MATLAB

Instructions

1. This is an individual lab. You will perform the tasks individually and submit a report.
2. Some of these tasks are for practice purposes only while others (marked as 'Exercise') have to be answered in the report.
3. When asked to display an image/ graph in the exercise either save it as jpeg or take a screenshot, in order to insert it in the report.
4. The report should be submitted on the given template, including:
   a. Code (copy and pasted, NOT a screenshot)
   b. Output values (from command window, can be a screenshot)
   c. Output figure/graph (as instructed in 3)
   d. Explanation where required
5. The report should be properly formatted, with easy to read code and easy to see figures.
6. Plagiarism or any hint thereof will be dealt with strictly. Any incident where plagiarism is caught, both (or all) students involved will be given zero marks, regardless of who copied whom. Multiple such incidents will result in disciplinary action being taken.

Background

The goal of filtering is to perform frequency-dependent alteration of a signal. A simple design specification for a filter might be to remove noise above a certain cut-off frequency. A more complete specification might call for a specific amount of pass band ripple (Rp, in decibels). Stop band attenuation (Rs, in decibels), or transition width (Wp-Ws, in hertz).

Digital filter specifications are often given is terms of the loss function (in dB),

$$A(\omega) = -20\log 10|G(e^{jw})|$$

The Filter Designer tool of MATLAB allows to design and edit IIR and FIR filters of various lengths and types, with lowpass, highpass, bandpass, and bandstop configurations. It also allows export of the designed filter to workspace or Simulink.

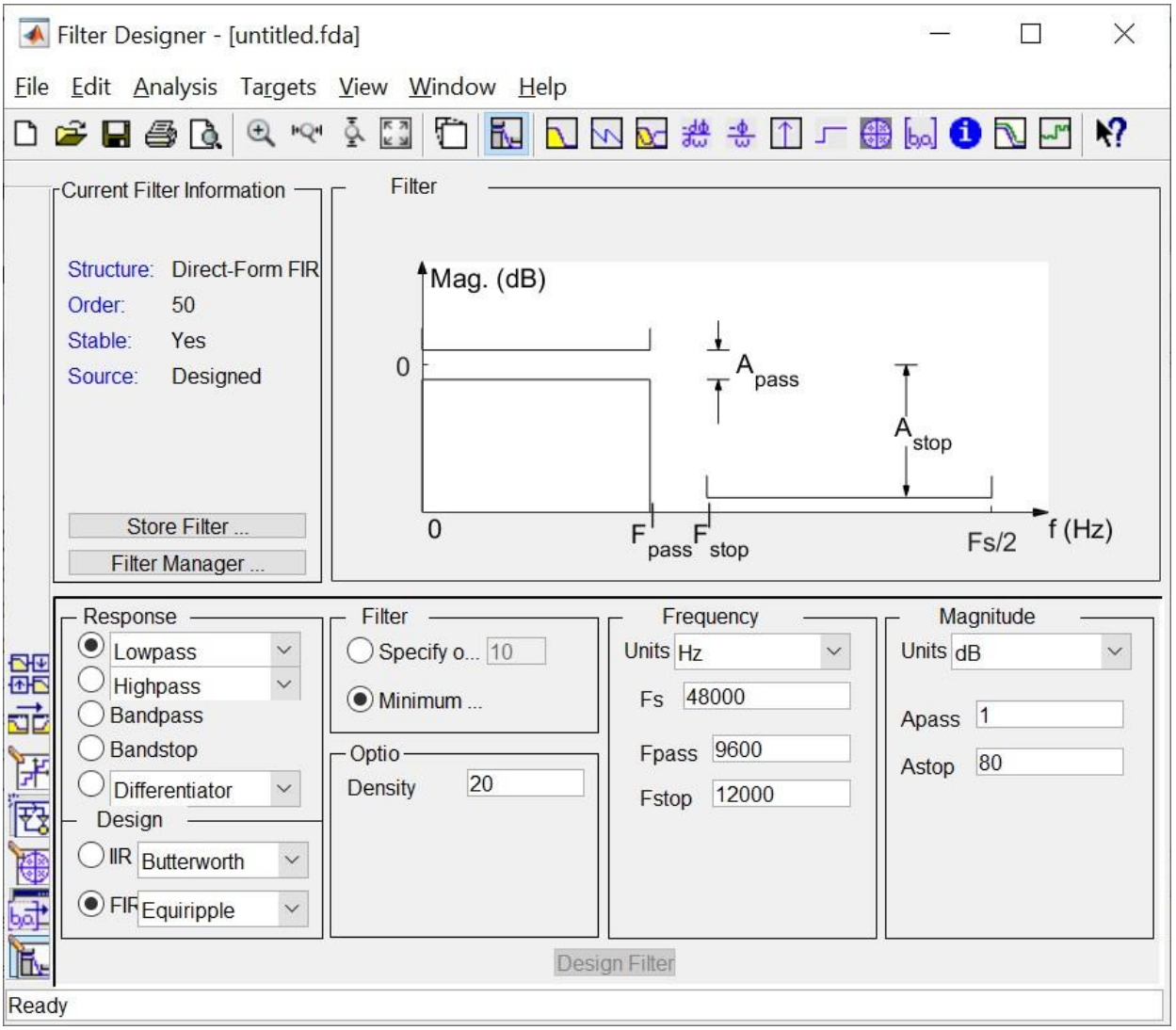
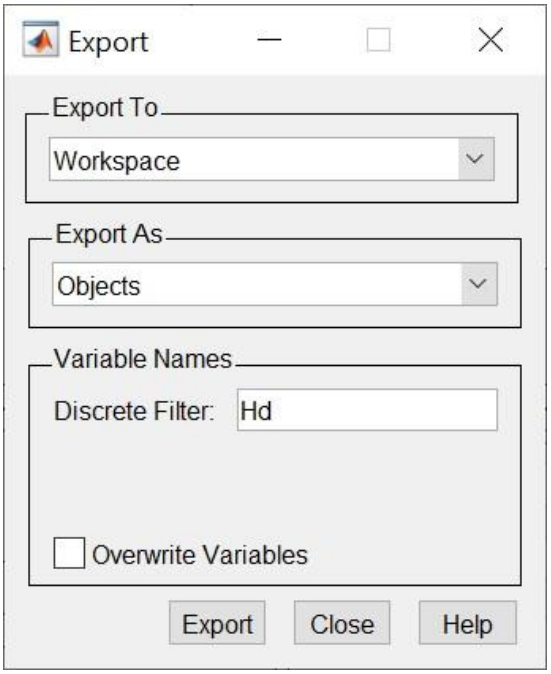


Figure 1: Filter Designer GUI

Once a filter has been designed using this GUI, it can be exported to the workspace by going to File > Export and choosing to export it as objects, so that it can be applied directly in the filter function.

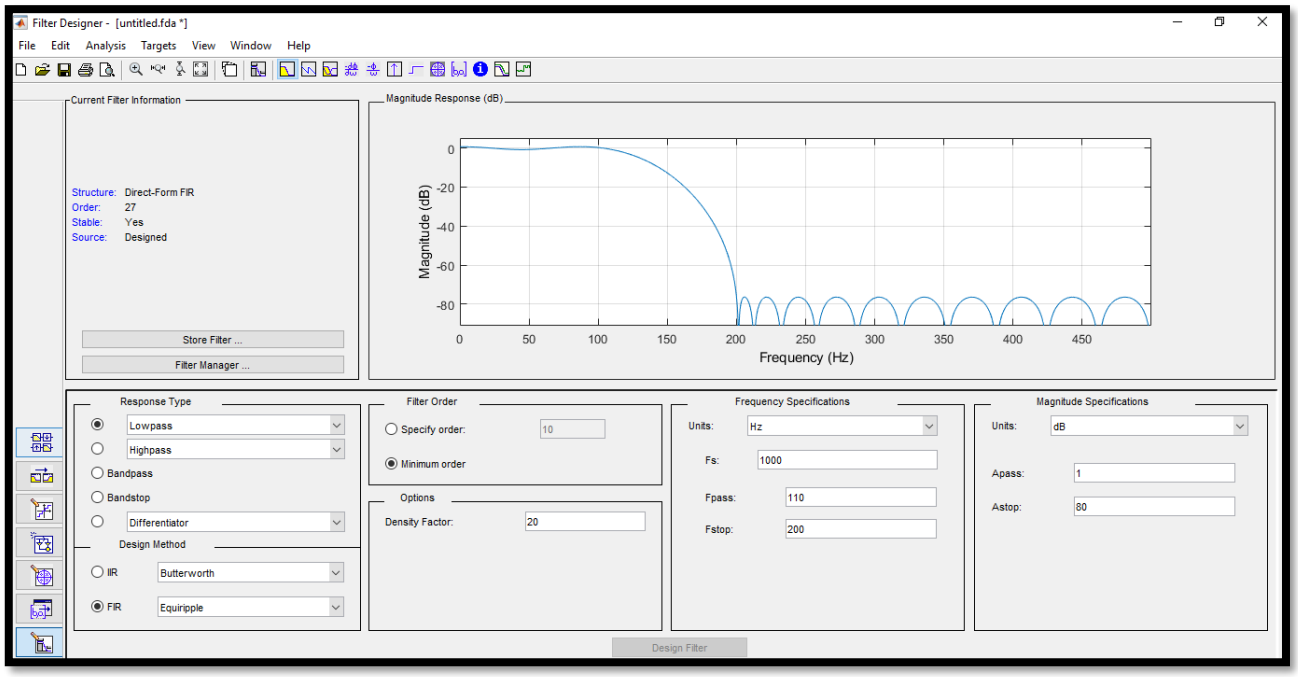


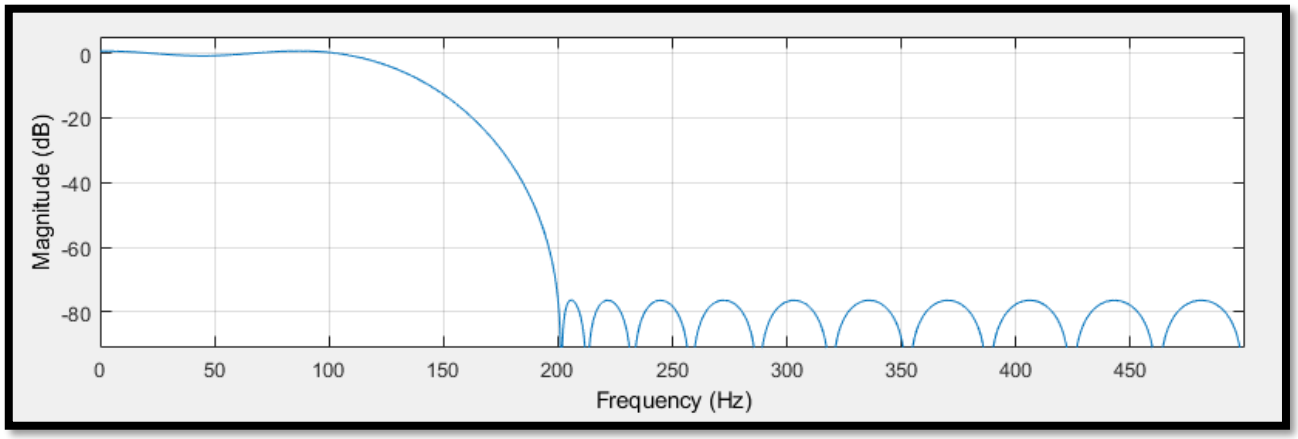
Open the filter designer in MATLAB either from the Apps, or by entering filterDesigner in the command window.

Exercise 1

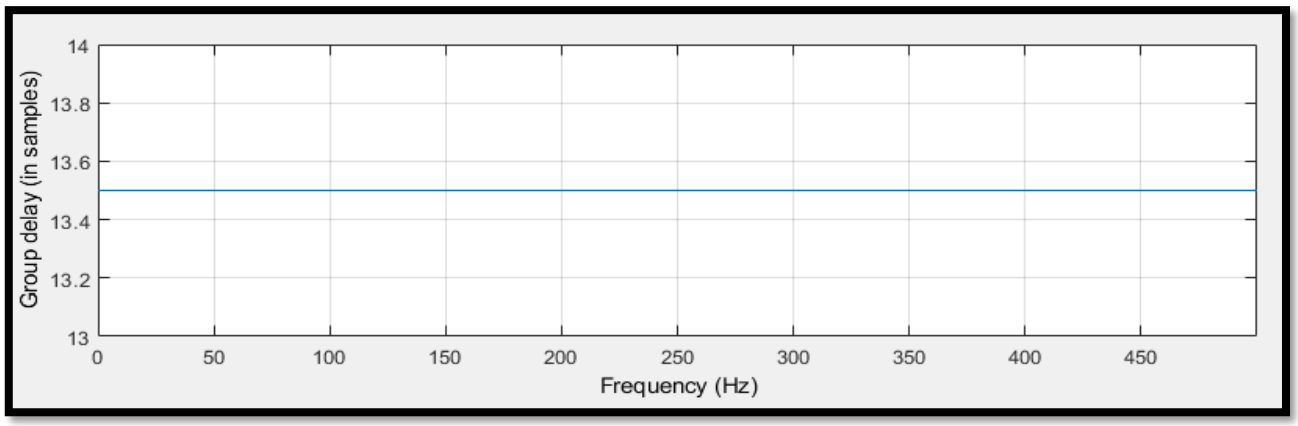
1. Design a low pass, FIR Equiripple filter with minimum order keeping sampling frequency Fs=1000 Hz, Fpass=110, Fstop=200, Apass =1dB, Astop =80dB.
2. Plot the magnitude response. Describe it.
3. Look at dφ/dω called the group delay. Describe the group delay and its significance.
4. Plot the impulse response of designed filter and write your comments.
5. Plot the pole zero map of designed filter and write your comments.
6. Export the filter to your workspace.

Magnitude Response:

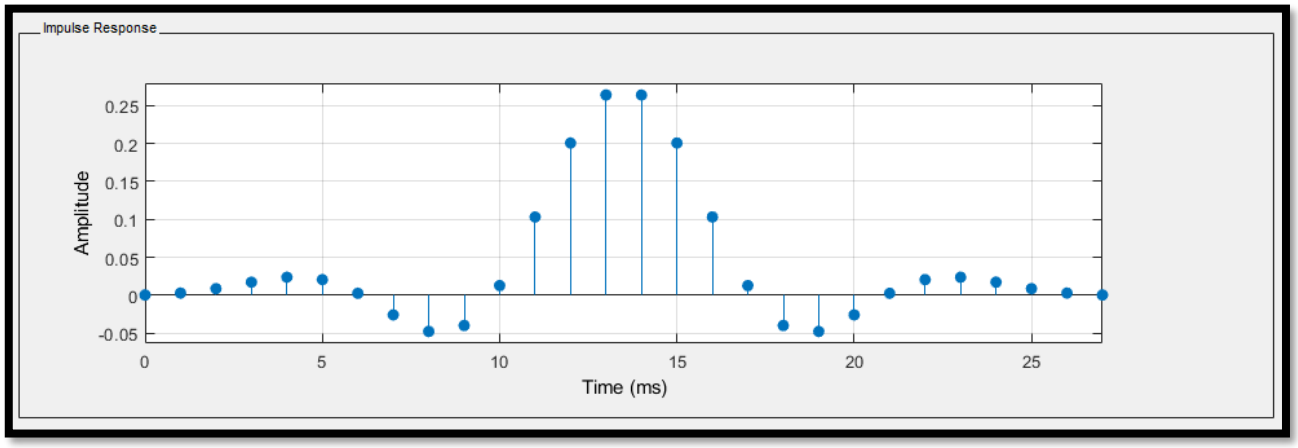




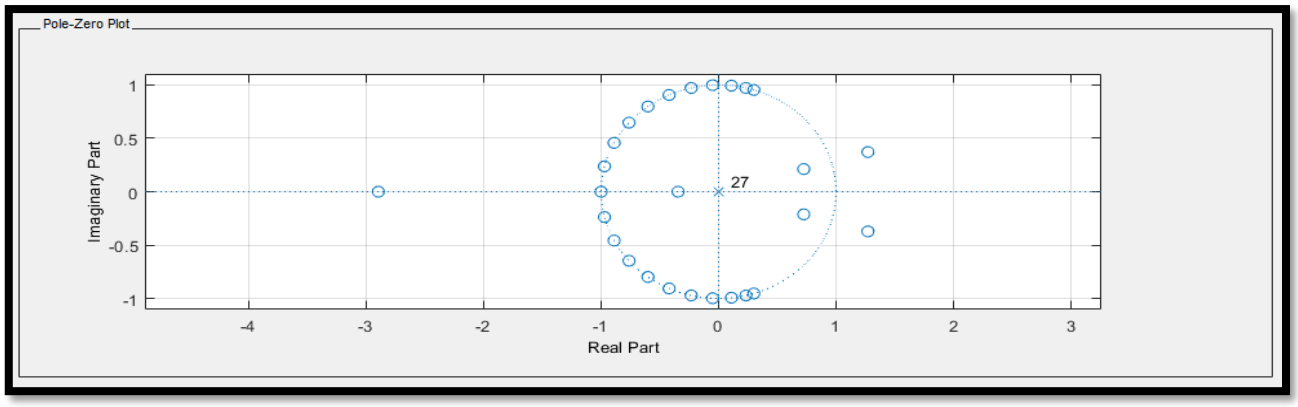
Group Delay dφ/dω:



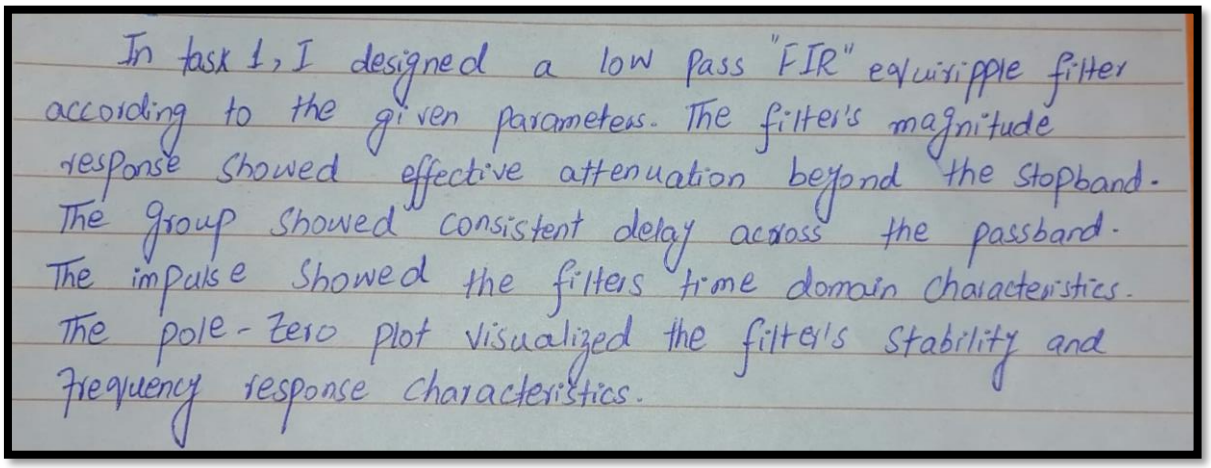
Impulse response:



Pole-Zero Plot:



Explanation:



Exercise 2

Define a signal y using the following code:

```
fs = 1000;
f1 = 100;
f2 = 200;
f3 = 300;
t = 0:(1/fs):(100/f3)-(1/fs);
x1=1*cos(2*pi*f1*t);
x2=2*cos(2*pi*f2*t);
x3=3*cos(2*pi*f3*t);
y = x1+x2+x3;
```

Plot the signal, as well as its frequency response.

Apply the filter designed in Task 1 to this signal using the filter(_) function.

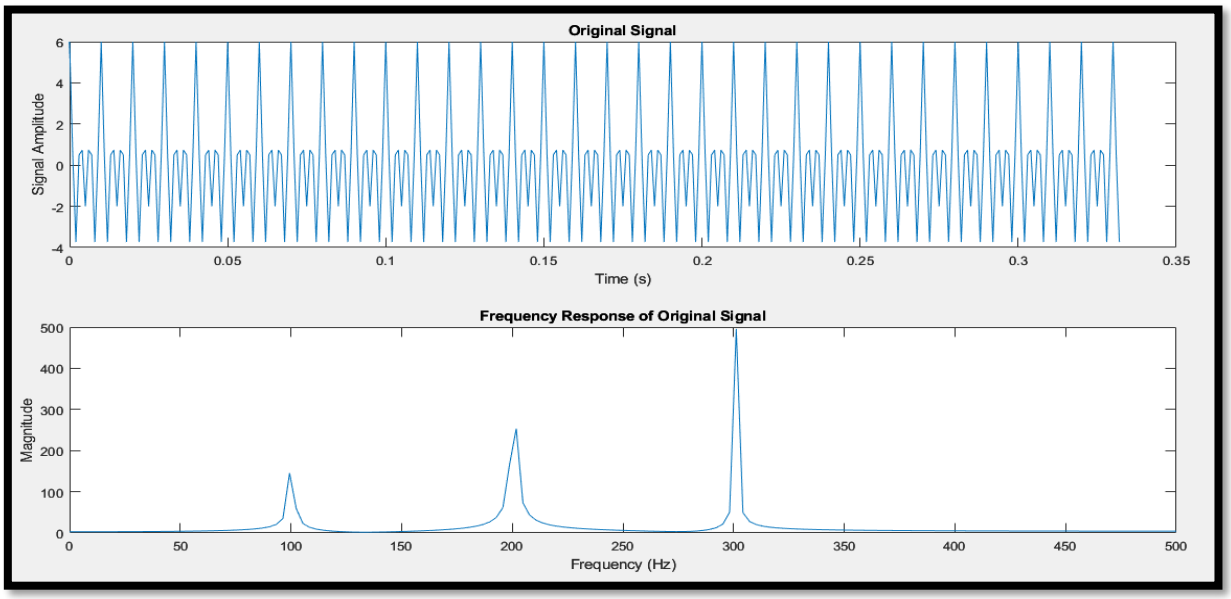
Explain the output using time as well as frequency domain representations.

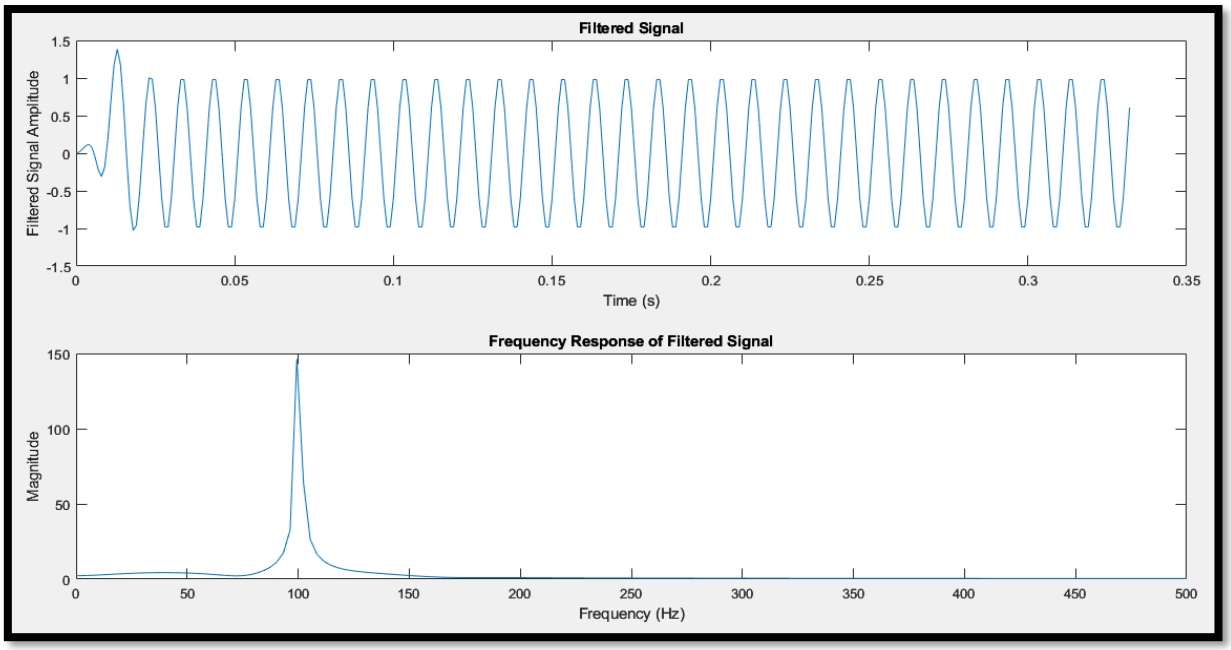
MATLAB Code:

```
clc, clear all:

% Define parameters
fs = 1000;
f1 = 100;
f2 = 200;
f3 = 300;

% Time vector
t = 0:1/fs:(100/f3)-(1/fs);

% Generate signals
x1 = 1*cos(2*pi*f1*t);
x2 = 2*cos(2*pi*f2*t);
x3 = 3*cos(2*pi*f3*t);
```

```
y = x1 + x2 + x3;

% Plot original signal in time domain
figure(1);
subplot(2, 1, 1);
plot(t,y);
xlabel('Time (s)');
ylabel('Signal Amplitude');
title('Original Signal');

% Calculate frequency response of original signal (FFT)
Y = fft(y);
f = linspace(0, fs/2, length(Y)/2+1);

% Plot frequency response of original signal
subplot(2, 1, 2);
plot(f, abs(Y(1:length(f))))
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Frequency Response of Original Signal');

% Apply filter
filtered_y = filter(Hd, y);

% Plot filtered signal in time domain
figure(2);
subplot(2, 1, 1);
plot(t,filtered_y);
xlabel('Time (s)');
ylabel('Filtered Signal Amplitude');
title('Filtered Signal');

% Calculate frequency response of filtered signal (FFT)
Y_filtered = fft(filtered_y);

% Plot frequency response of filtered signal
subplot(2, 1, 2);
plot(f, abs(Y_filtered(1:length(f))))
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Frequency Response of Filtered Signal');
```

Output:





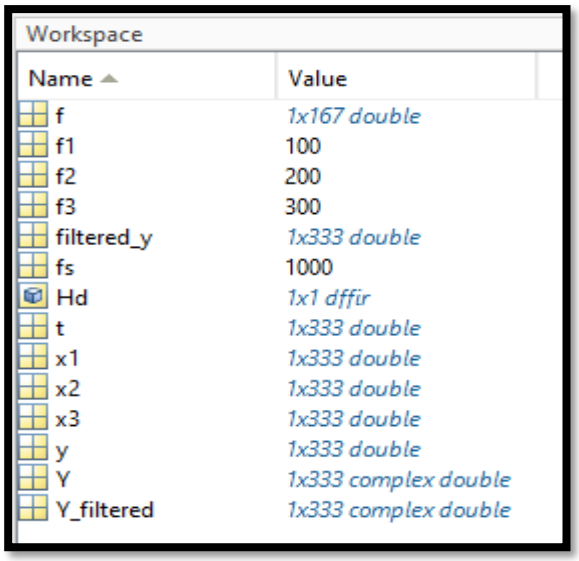
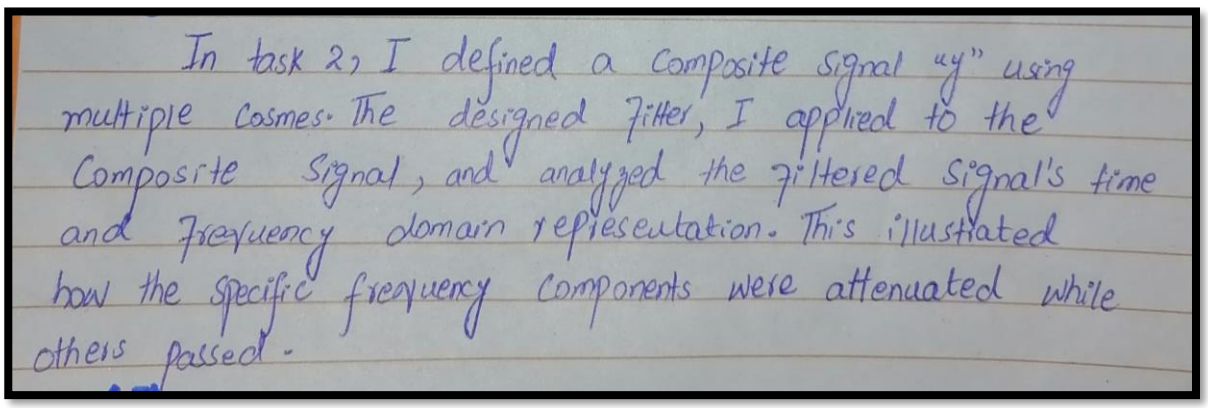


Figure 1: Workspace

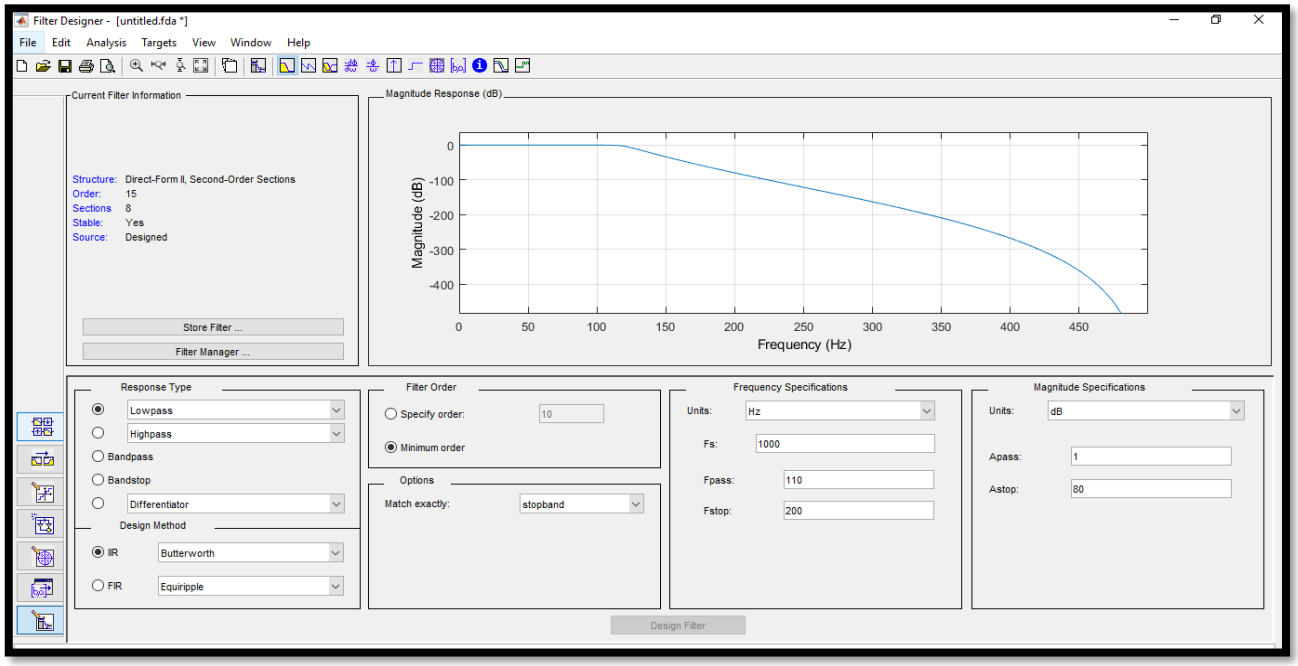
Explanation:



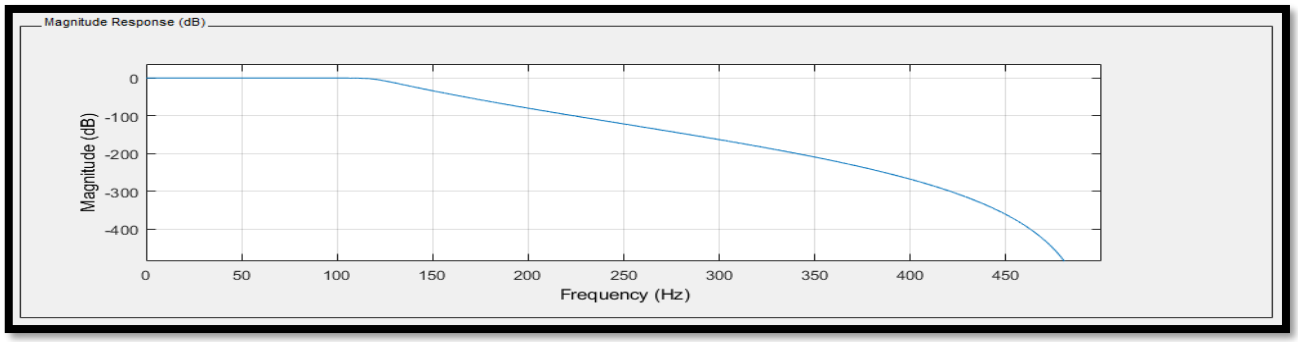
Exercise 3

Design an IIR Butterworth filter with the same specifications as the one in Exercise 1. Compare the two with respect to their filter order, magnitude response, group delay, impulse response, and pole-zero plot.

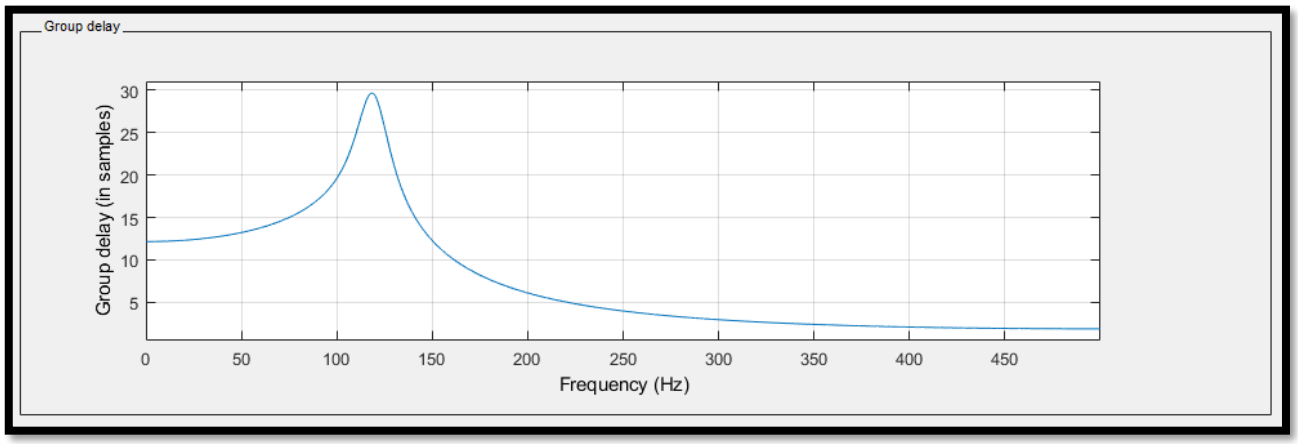
IIR Filter:



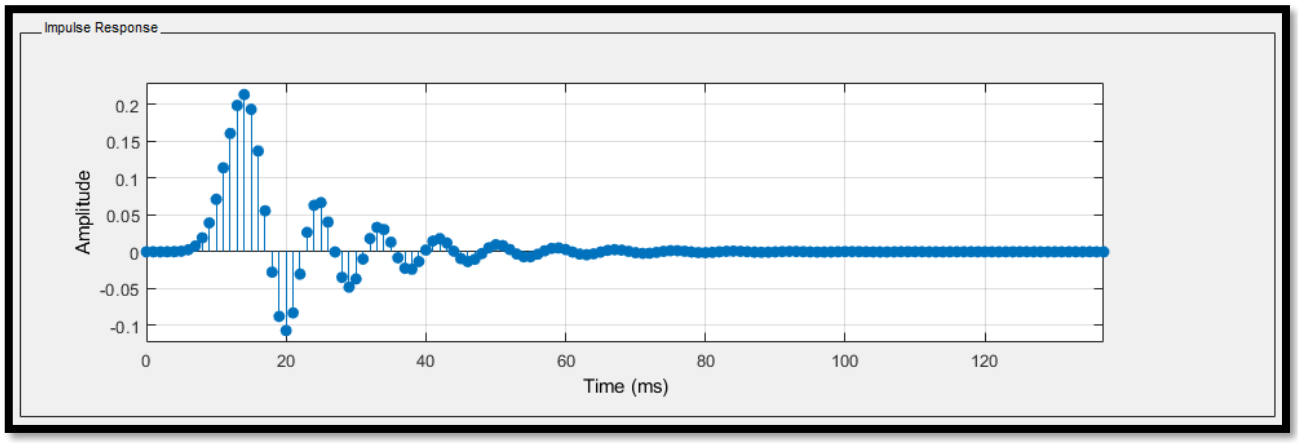
Magnitude Response:



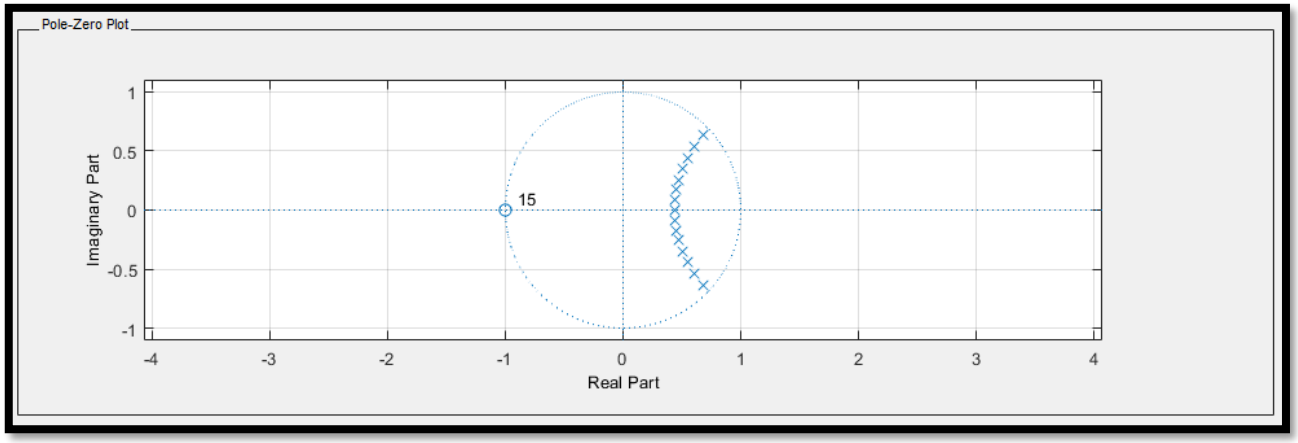
Group Delay:



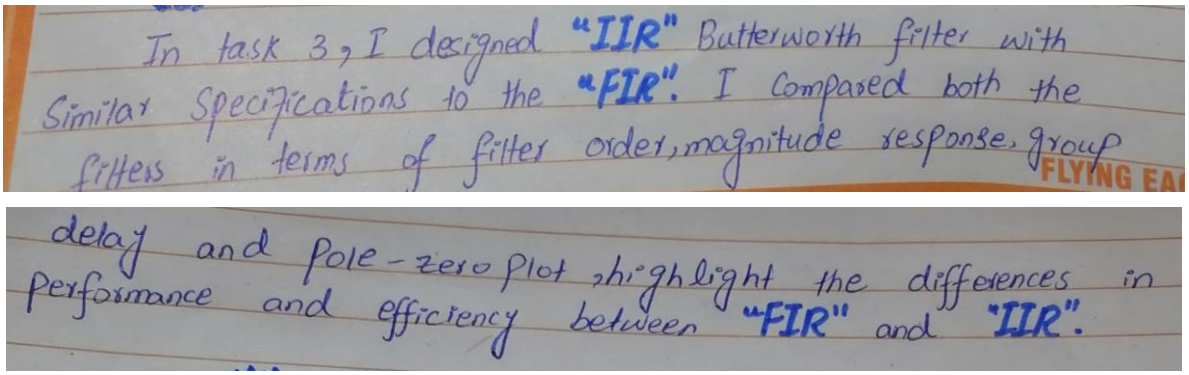
Impulse Response:



Pole-Zero Plot:



Explanation:

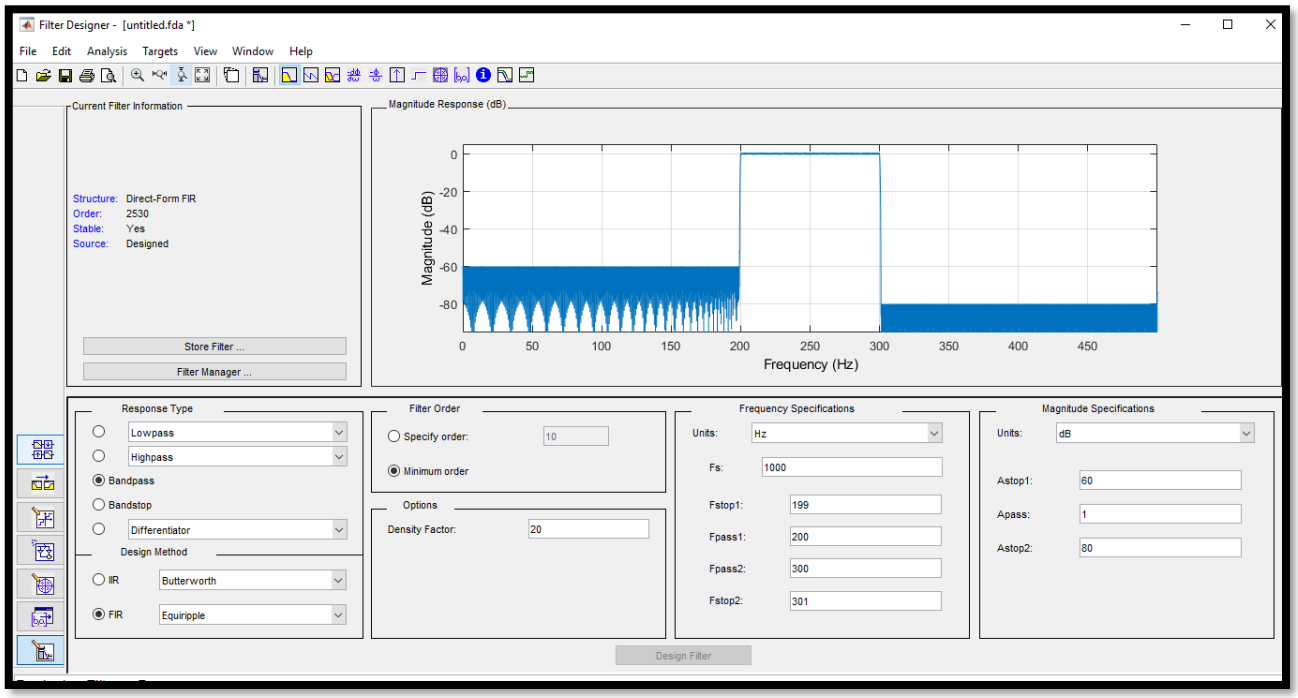


Exercise 4

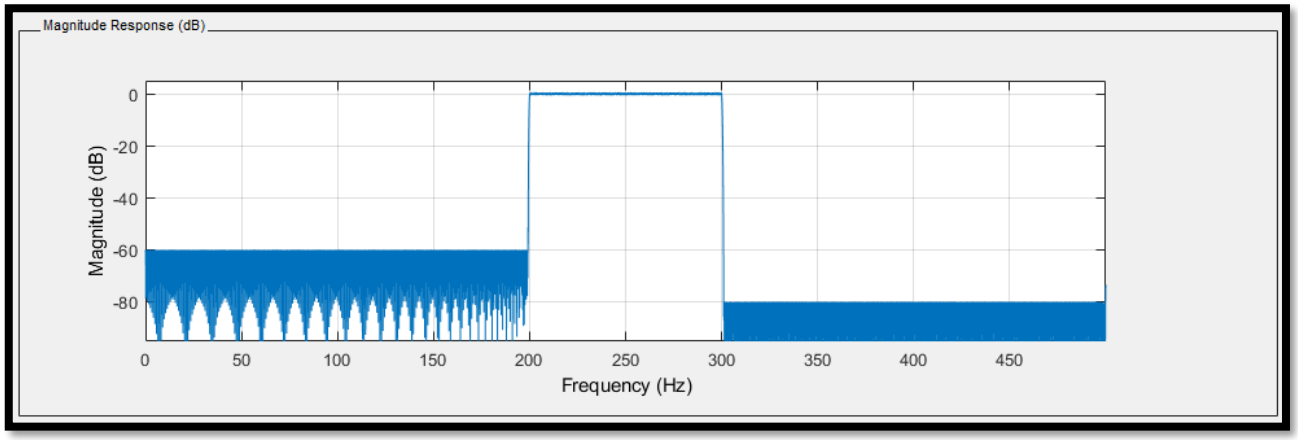
Design a bandpass FIR filter that would only pass a band of frequencies from 200 to 300Hz of the signal in Exercise 2.

Apply the filter to the signal and show that the other frequency components are suppressed.

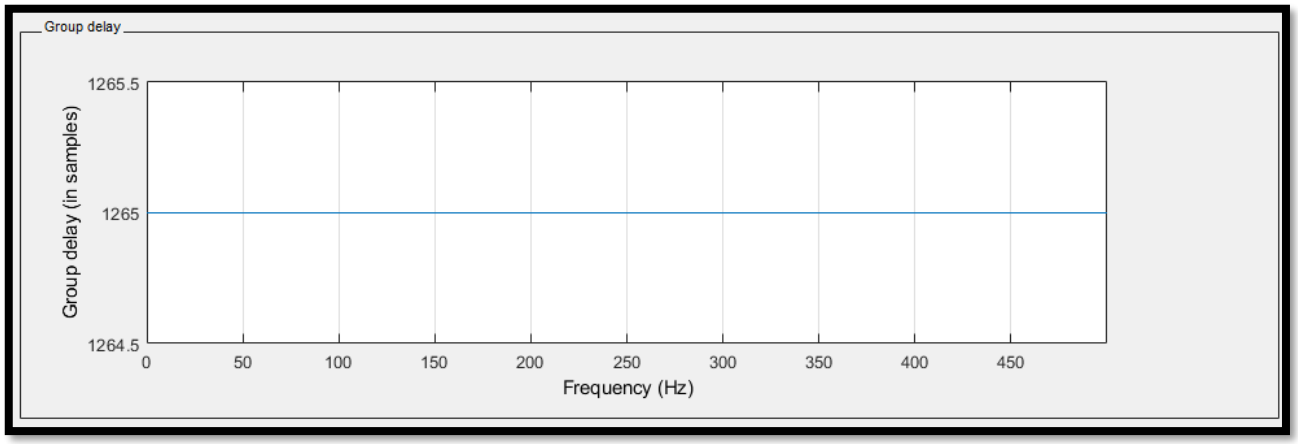
Bandpass FIR Filter:



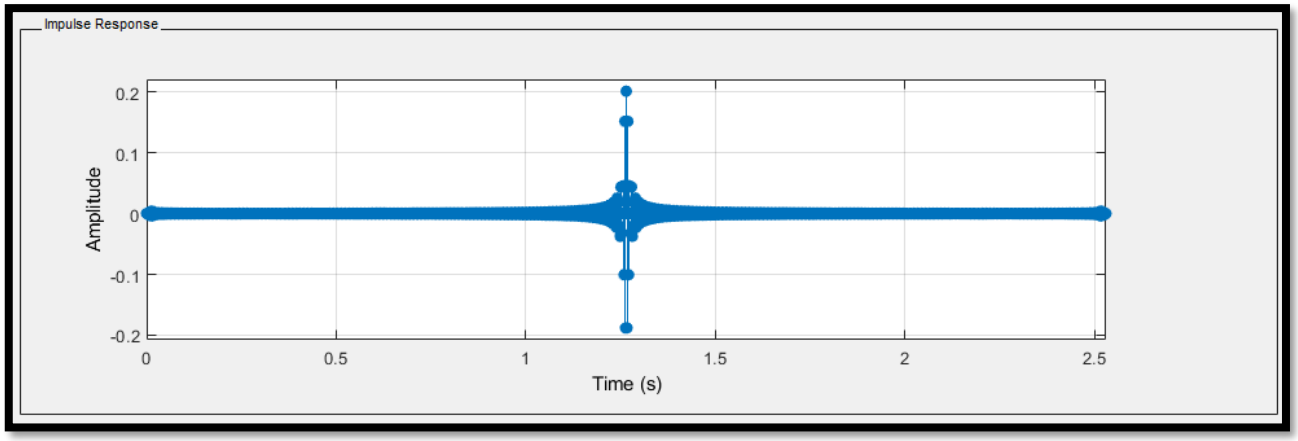
Magnitude Response



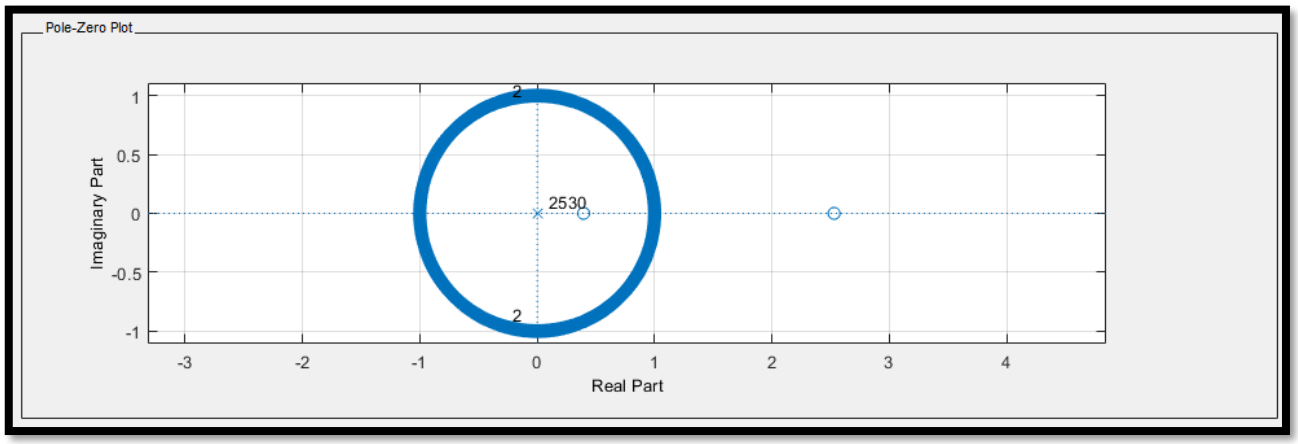
Group Delay:



Impulse Response:



Pole-Zero Plot:

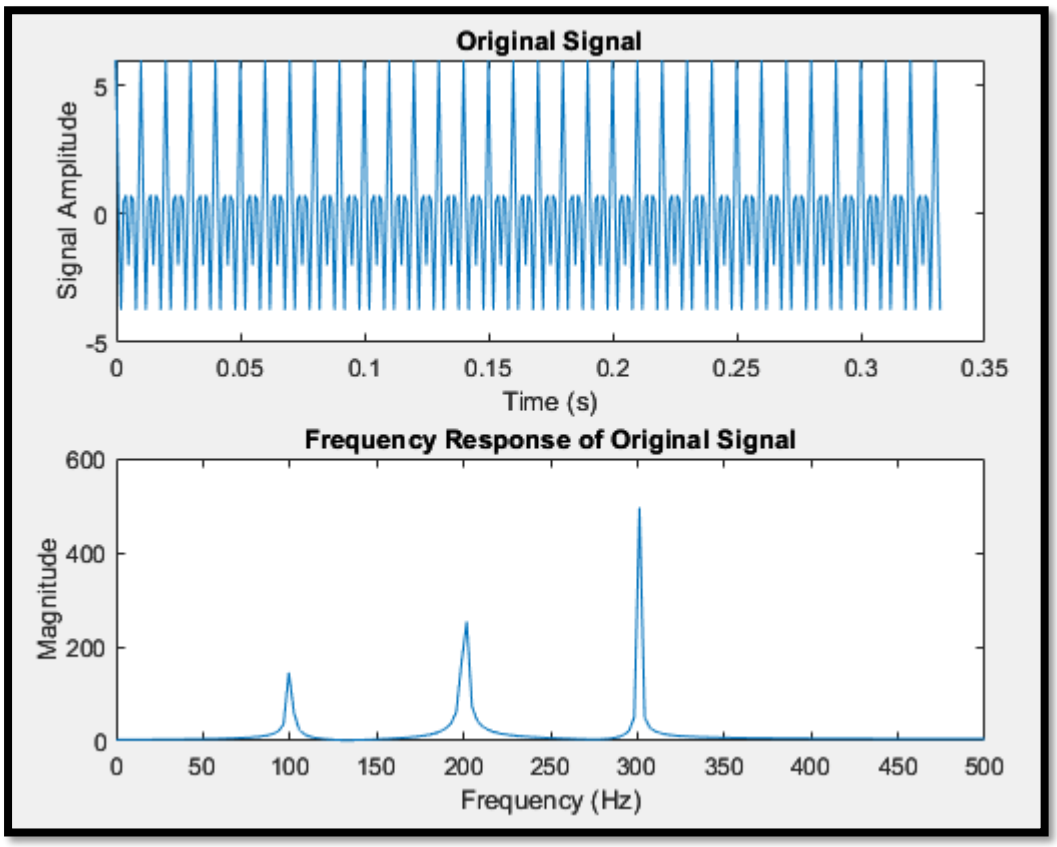


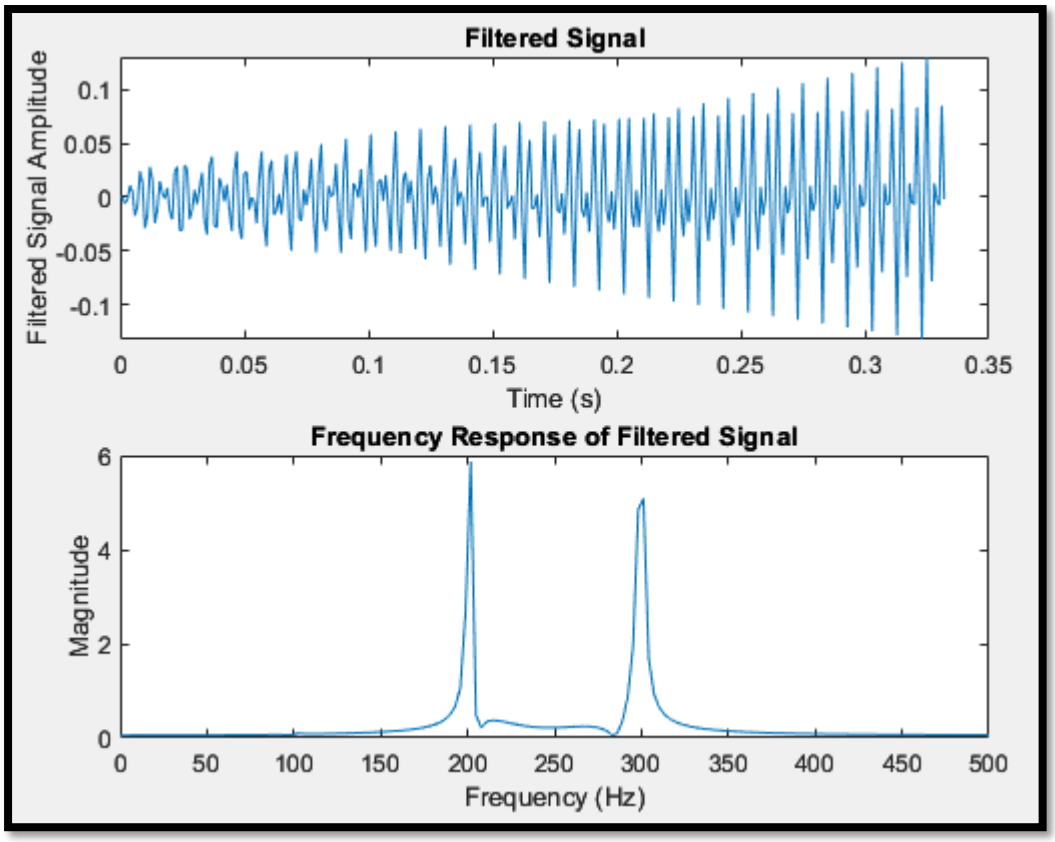
MATLAB Code:

```
clc, clear all:

% Define parameters
fs = 1000;
f1 = 100;
f2 = 200;
f3 = 300;

% Time vector
t = 0:1/fs:(100/f3)-(1/fs);

% Generate signals
x1 = 1*cos(2*pi*f1*t);
x2 = 2*cos(2*pi*f2*t);
x3 = 3*cos(2*pi*f3*t);
y = x1 + x2 + x3;

% Plot original signal in time domain
figure(1);
subplot(2, 1, 1);
plot(t,y);
xlabel('Time (s)');
ylabel('Signal Amplitude');
title('Original Signal');

% Calculate frequency response of original signal (FFT)
Y = fft(y);
f = linspace(0, fs/2, length(Y)/2+1);
```

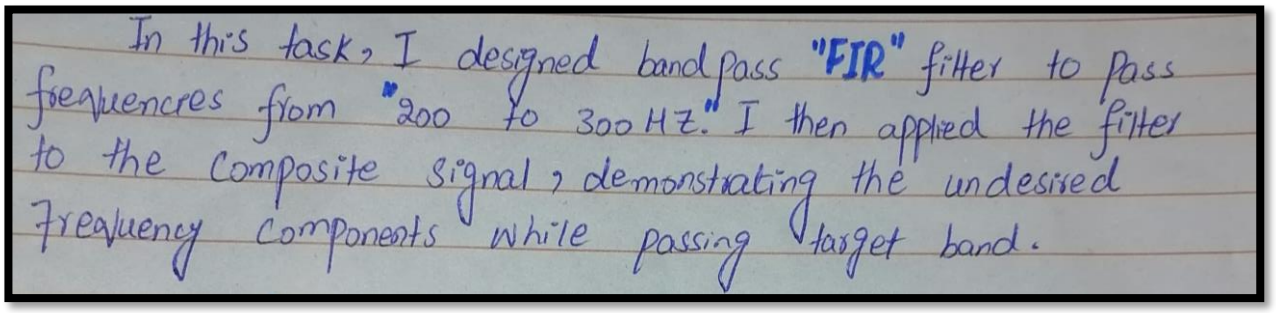
```
% Plot frequency response of original signal
subplot(2, 1, 2);
plot(f, abs(Y(1:length(f))))
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Frequency Response of Original Signal');

% Apply filter
filtered_y = filter(Fahim y);

% Plot filtered signal in time domain
figure(2);
subplot(2, 1, 1);
plot(t,filtered_y);
xlabel('Time (s)');
ylabel('Filtered Signal Amplitude');
title('Filtered Signal');

% Calculate frequency response of filtered signal (FFT)
Y_filtered = fft(filtered_y);

% Plot frequency response of filtered signal
subplot(2, 1, 2);
plot(f, abs(Y_filtered(1:length(f))))
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Frequency Response of Filtered Signal');
```

Output:





Explanation:

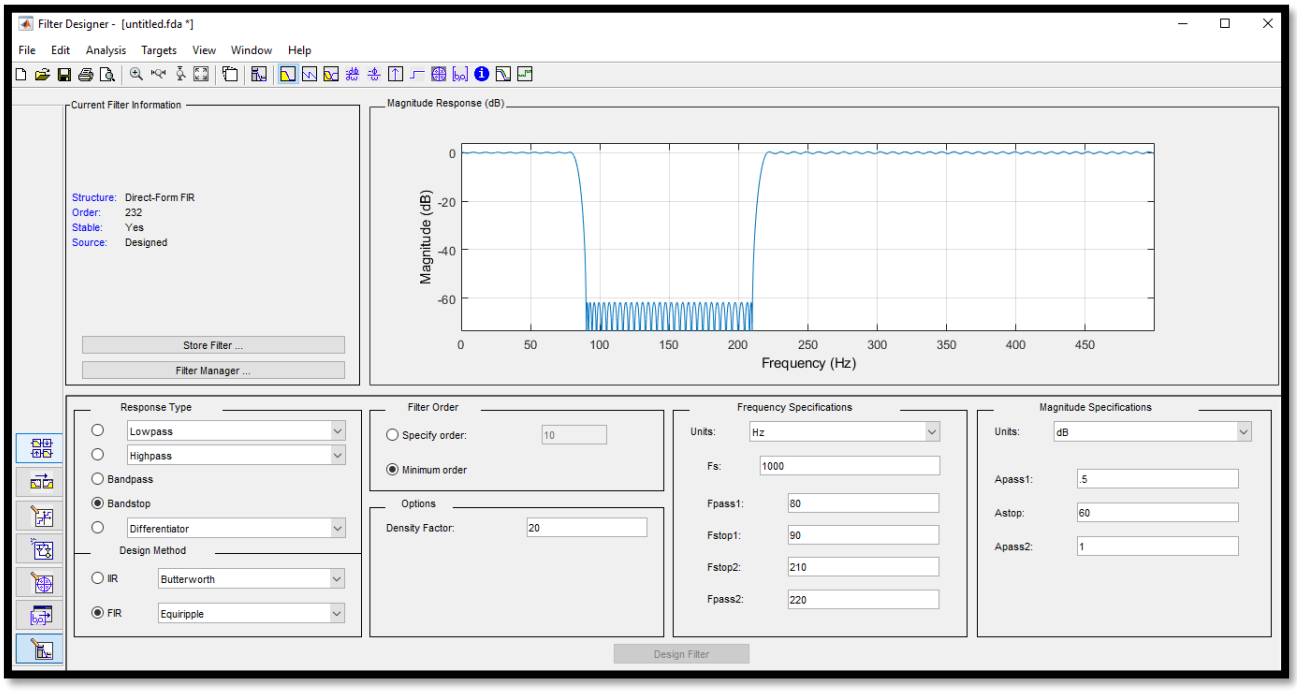


Exercise 5

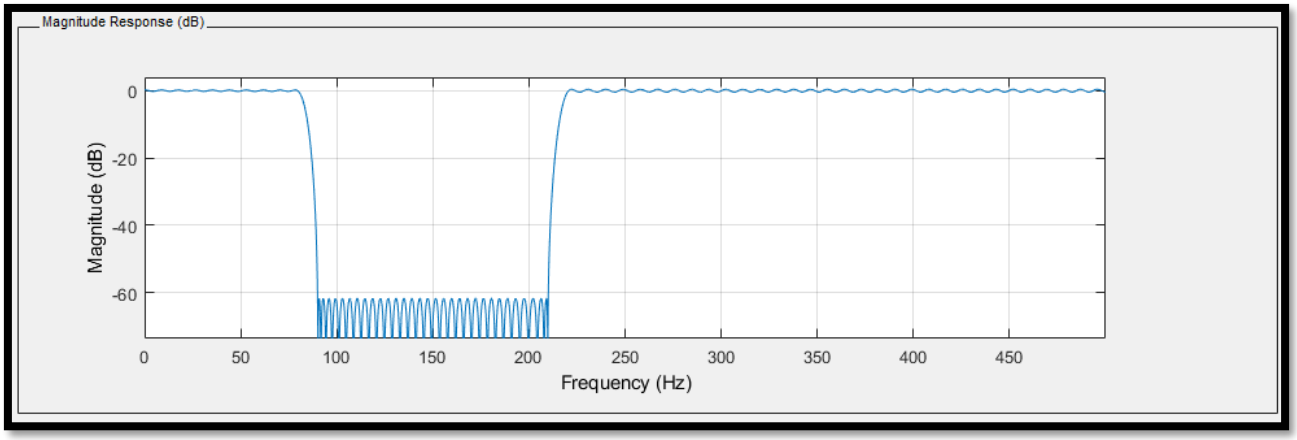
Design a bandstop FIR filter that would stop the band of frequency 100 to 200Hz of the signal in Exercise 2.

Apply the filter to the signal and show that the frequency component is suppressed, while the others are passed.

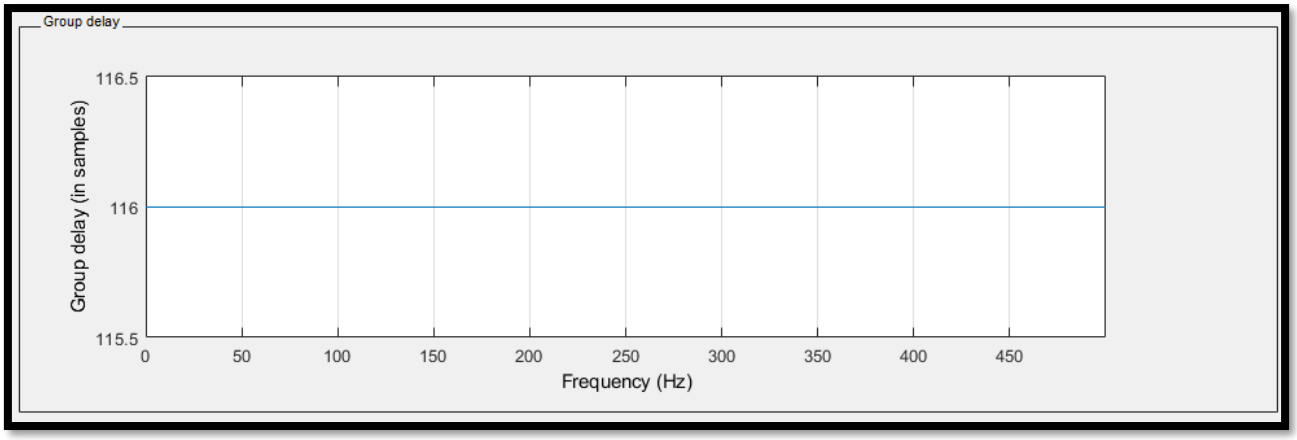
Bandstop FIR Filter:



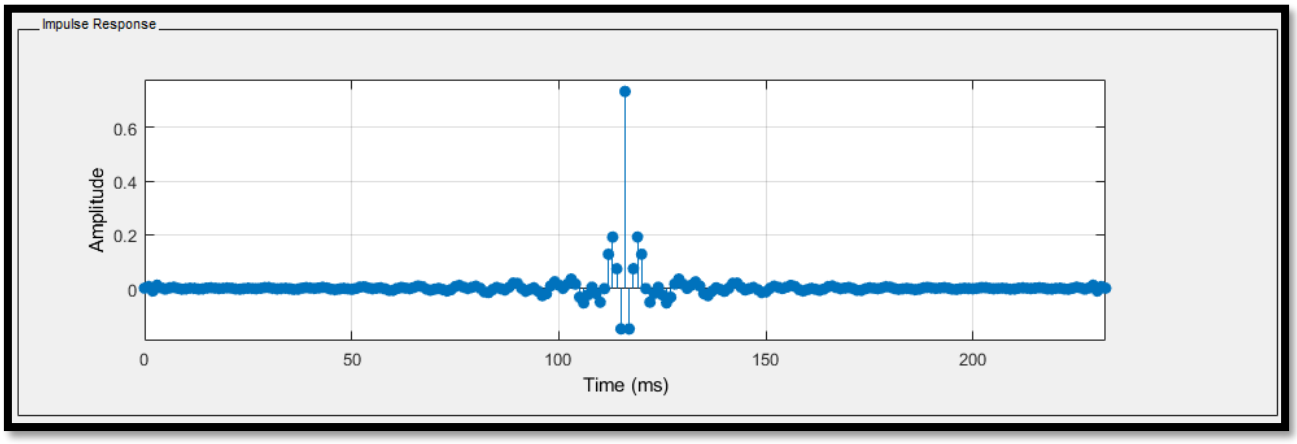
Magnitude Response:



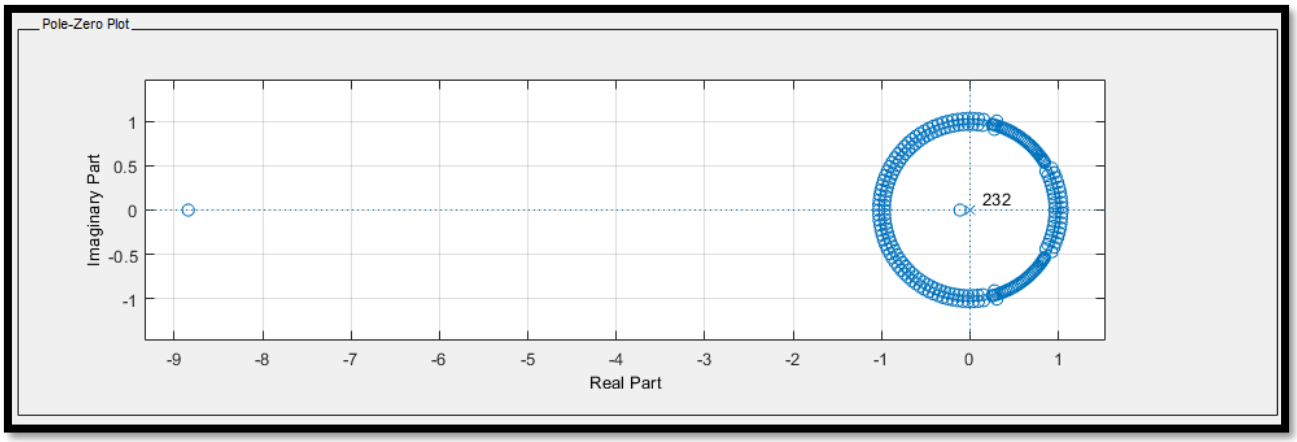
Group Delay:



Impulse Response:



Pole-Zero Plot:

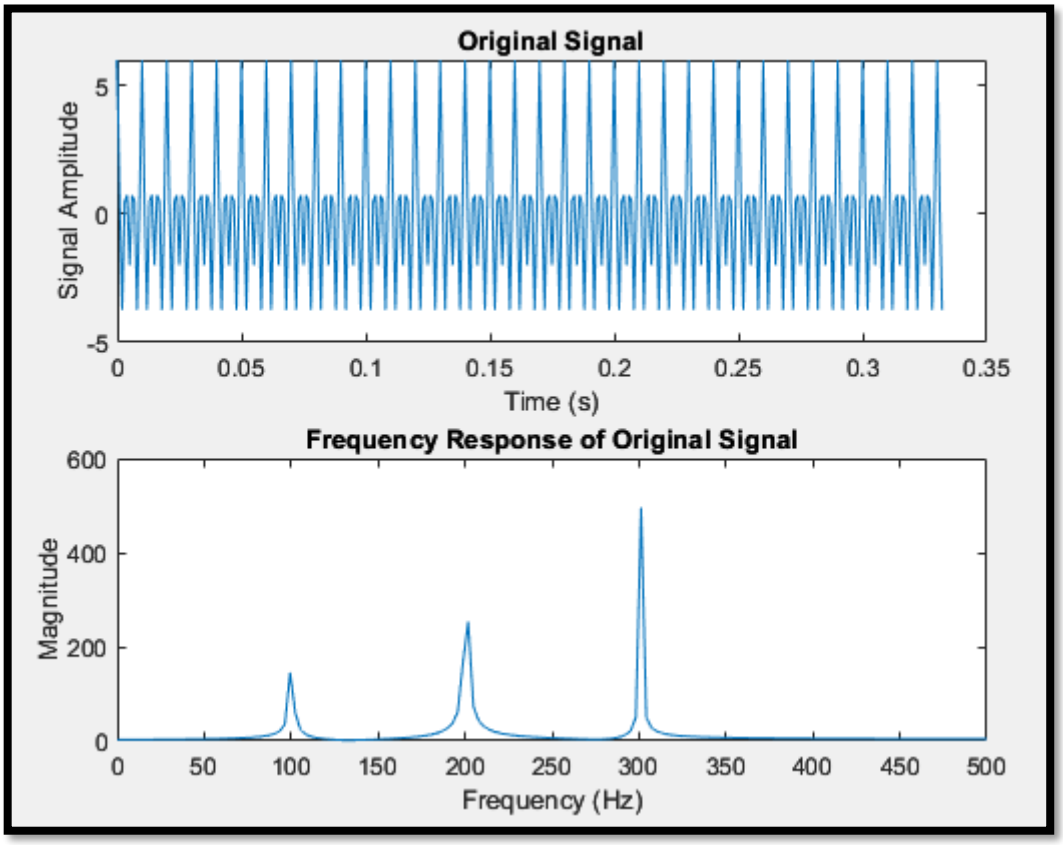


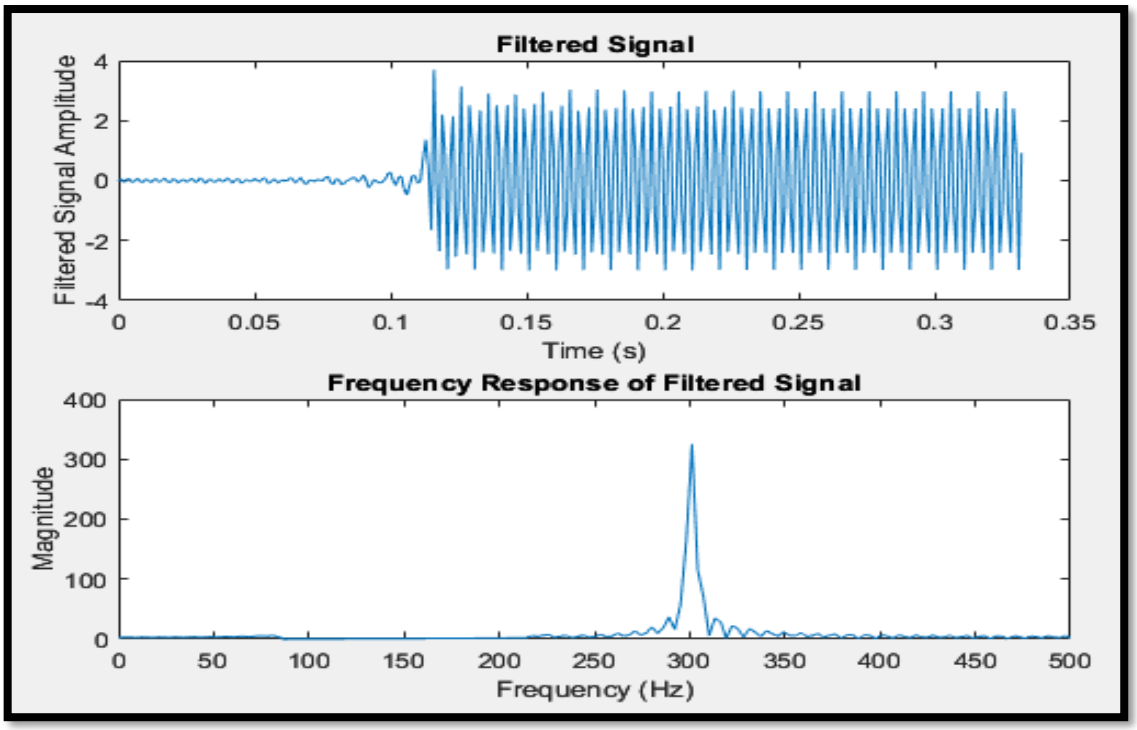
MATLAB Code:

```
clc, clear all:

% Define parameters
fs = 1000;
f1 = 100;
f2 = 200;
f3 = 300;

% Time vector
t = 0:1/fs:(100/f3)-(1/fs);

% Generate signals
x1 = 1*cos(2*pi*f1*t);
x2 = 2*cos(2*pi*f2*t);
x3 = 3*cos(2*pi*f3*t);
y = x1 + x2 + x3;

% Plot original signal in time domain
figure(1);
subplot(2, 1, 1);
plot(t,y);
xlabel('Time (s)');
ylabel('Signal Amplitude');
title('Original Signal');

% Calculate frequency response of original signal (FFT)
Y = fft(y);
f = linspace(0, fs/2, length(Y)/2+1);
```

```
% Plot frequency response of original signal
subplot(2, 1, 2);
plot(f, abs(Y(1:length(f))))
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Frequency Response of Original Signal');

% Apply filter
filtered_y = filter(Shah, y);

% Plot filtered signal in time domain
figure(2);
subplot(2, 1, 1);
plot(t,filtered_y);
xlabel('Time (s)');
ylabel('Filtered Signal Amplitude');
title('Filtered Signal');

% Calculate frequency response of filtered signal (FFT)
Y_filtered = fft(filtered_y);

% Plot frequency response of filtered signal
subplot(2, 1, 2);
plot(f, abs(Y_filtered(1:length(f))))
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Frequency Response of Filtered Signal');
```

Output:





Explanation:

I designed a bandstop "FIR" filter to stop frequencies from 100 to 200 HZ and applied it to the signal and it effectively suppressed the specified frequency band while allowing other components to pass.

Conclusion:

In this lab, I explored the design and implementation of "FIR" and "IIR" filters using MATLAB. The basic purpose was to understand how different filters affects signals in both time and frequency domains. I designed many filters using MATLAB's Filter Designer App and applied them to Composite signals to observe their effects.

This lab provided hands-on experience with digital filter design and applications. "FIR" filters, with their phase characteristics, are essential for applications requiring phase preservation, while "IIR" filters offer efficiency with lower cost but may introduce phase distortions.

Evaluation Rubric

- **Method of Evaluation**: In-lab marking by instructors, Report submitted by students
- **Measured Learning Outcomes**:

CLO1: Develop algorithms to perform signal processing techniques on digital signals using MATLAB and DSP Kit DSK6713
CLO3: Deliver a report/lab notes/presentation/viva, effectively communicating the design and analysis of the given problem

	Excellent 10	Good 9-7	Satisfactory 6- 4	Unsatisfactory 3- 1	Poor 0	Marks Obtained
Tasks (CLO1)	All tasks completed correctly. Correct code with proper comments.	Most tasks completed correctly.	Some tasks completed correctly.	Most tasks incomplete or incorrect.	All tasks incomplete or incorrect.	
Output (CLO1)	Output correctly shown with all Figures/Plots displayed as required and properly labelled	Most Output/Figures/Plots displayed with proper labels	Some Output/Figures/Plots displayed with proper labels OR Most Output/Figures/Plots displayed but without proper labels	Most of the required Output/Figures/Plots not displayed	Output/Figures/Plots not displayed	
Answers (CLO1)	Meaningful answers to all questions. Answers show the understanding of the student.	Meaningful answers to most questions.	Some correct/ meaningful answers with some irrelevant ones	Answers not understandable/ not relevant to questions	Not Written any Answer	
Report (CLO3)	Report submitted with proper grammar and punctuation with proper conclusions drawn and good formatting	Report submitted with proper conclusions drawn with good formatting but some grammar mistakes OR proper grammar but not very good formatting	Some correct/ meaningful conclusions. Some parts of the document not properly formatted or some grammar mistakes	Conclusions not based on results. Bad formatting with no proper grammar/punctuation	Report not submitted	
Total						