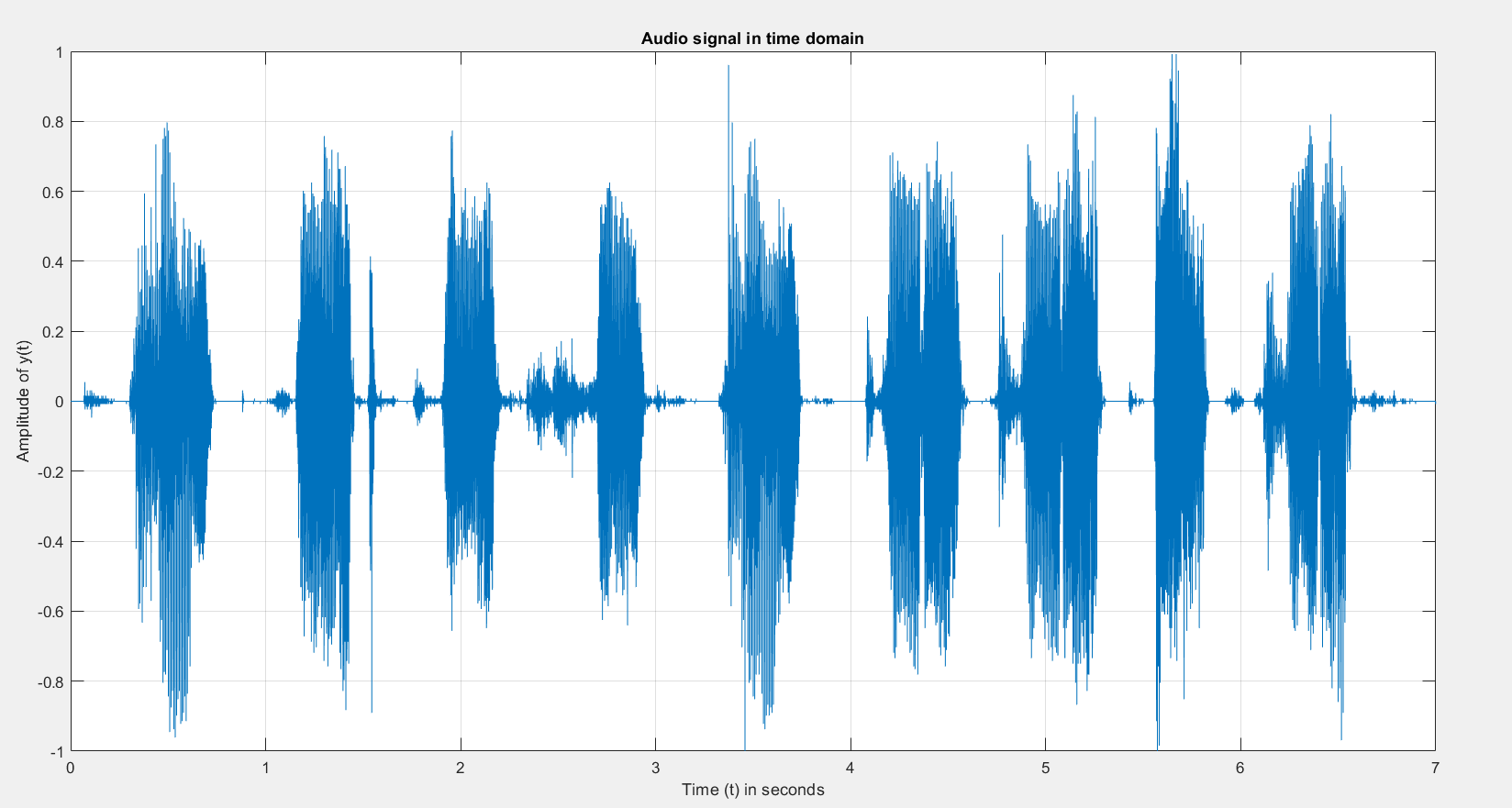
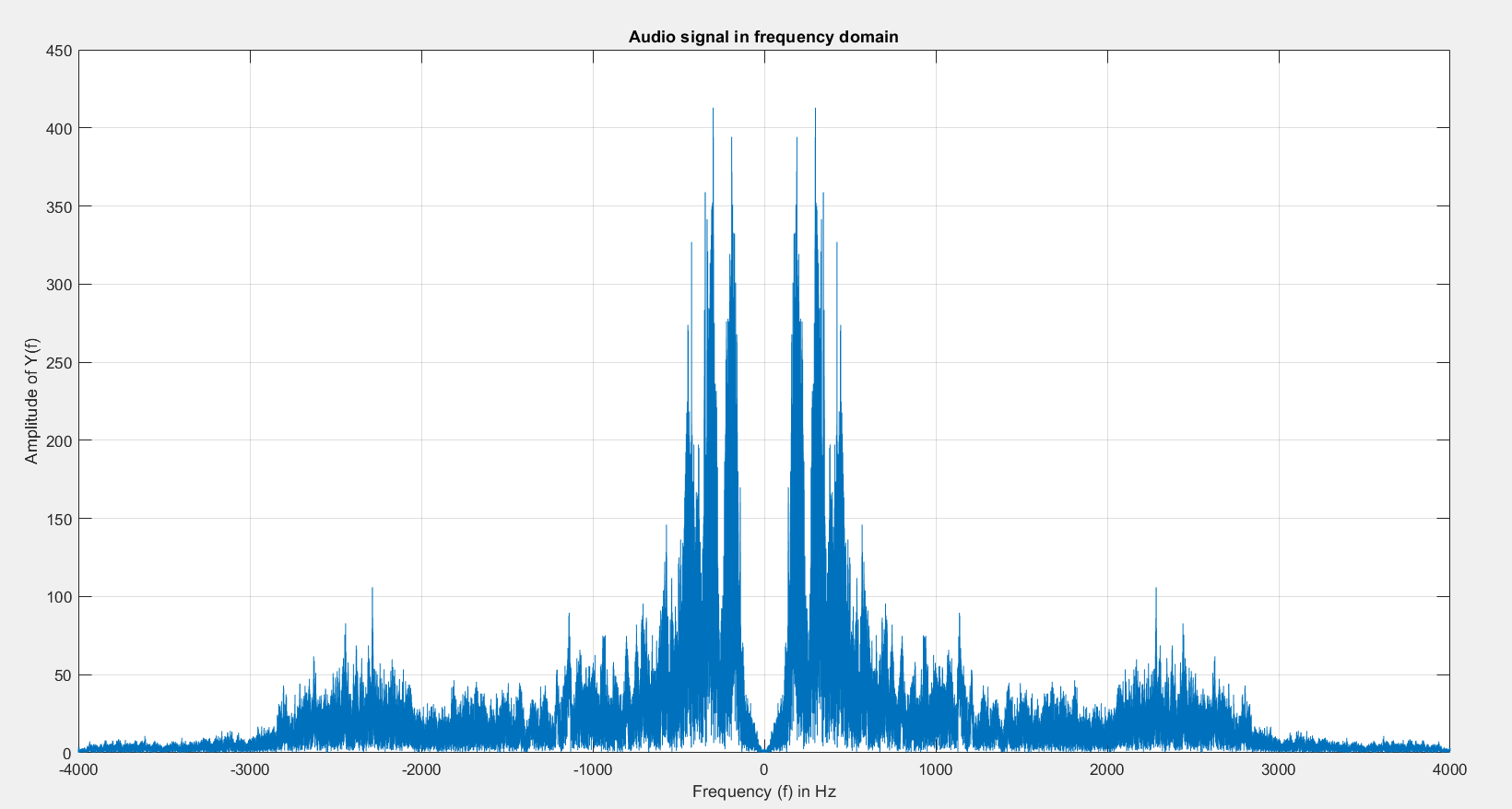
**Digital Signal Processing Laboratory**

**Lab Test**

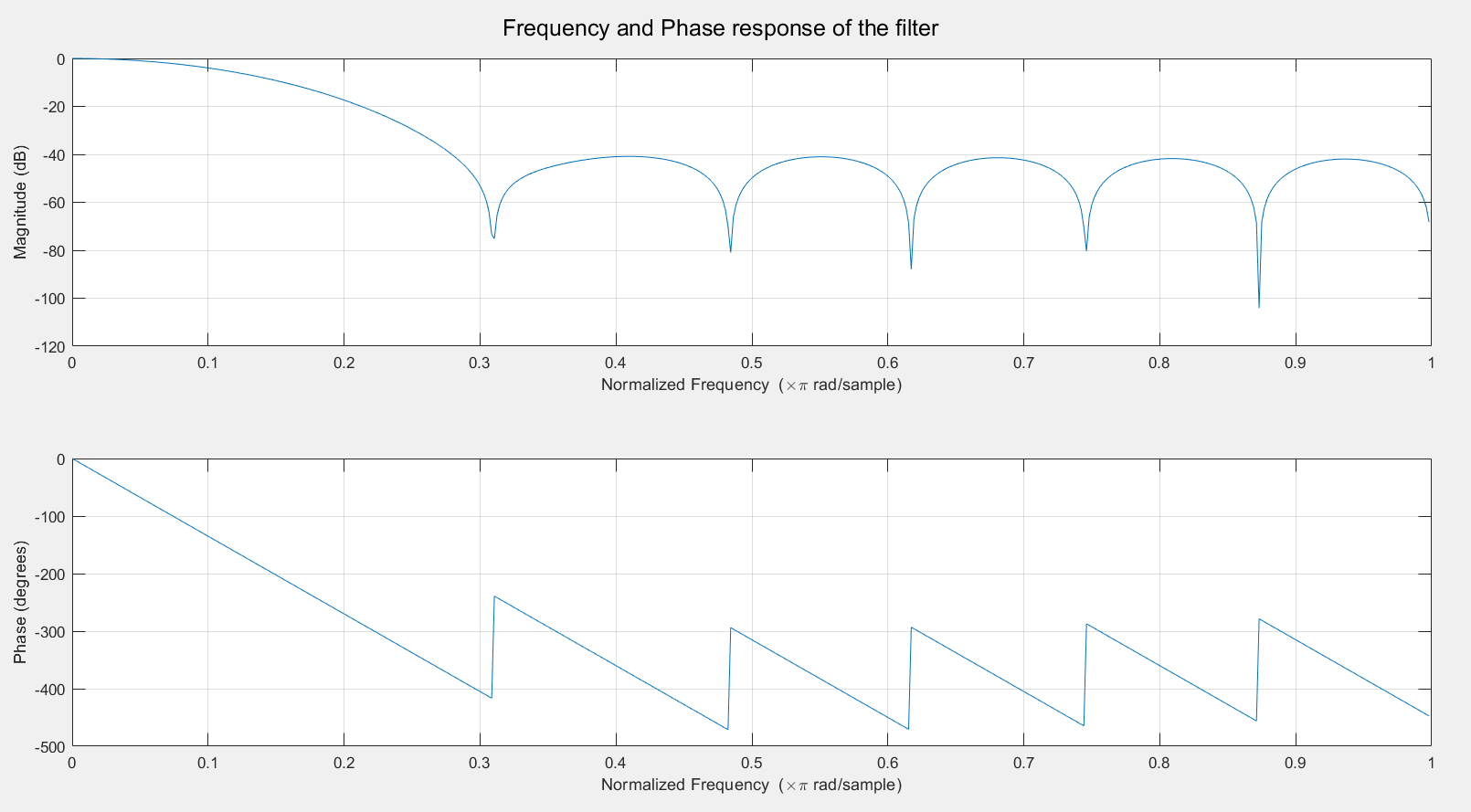
By: Hardik Suryaprakash Tibrewal (18EC10020)

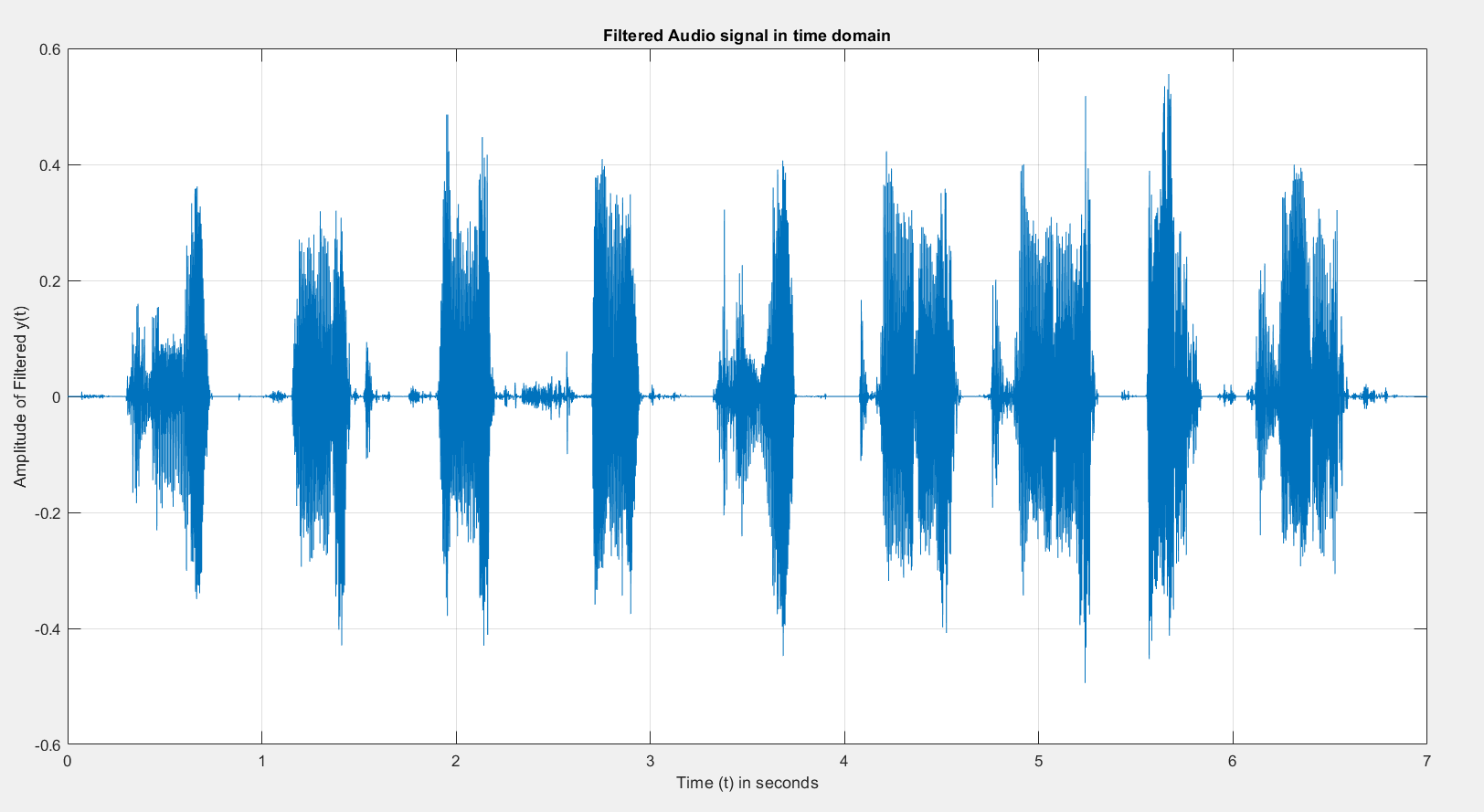
(Date of Birth: 18/11/2000)

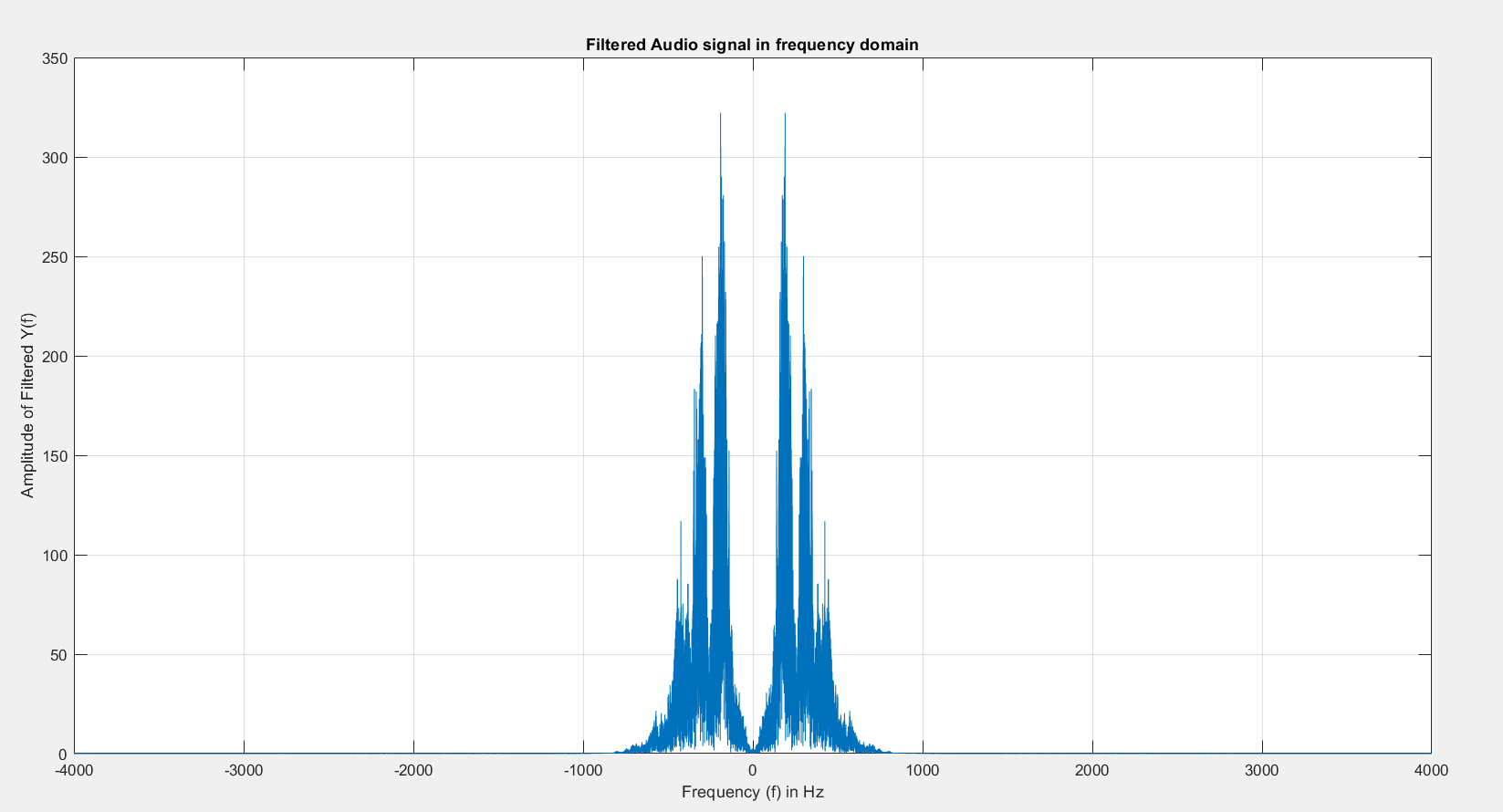
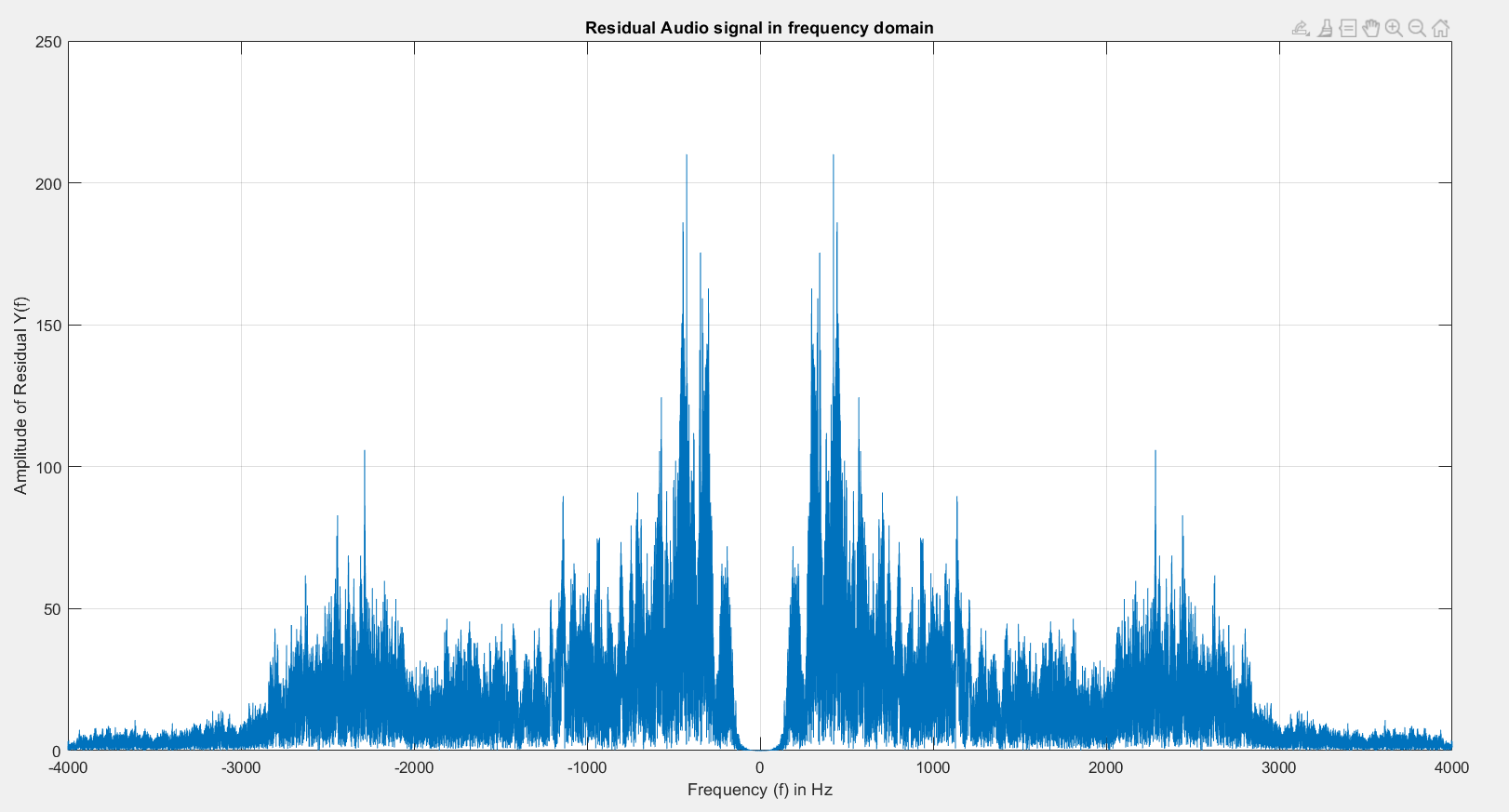
Simulation Results:

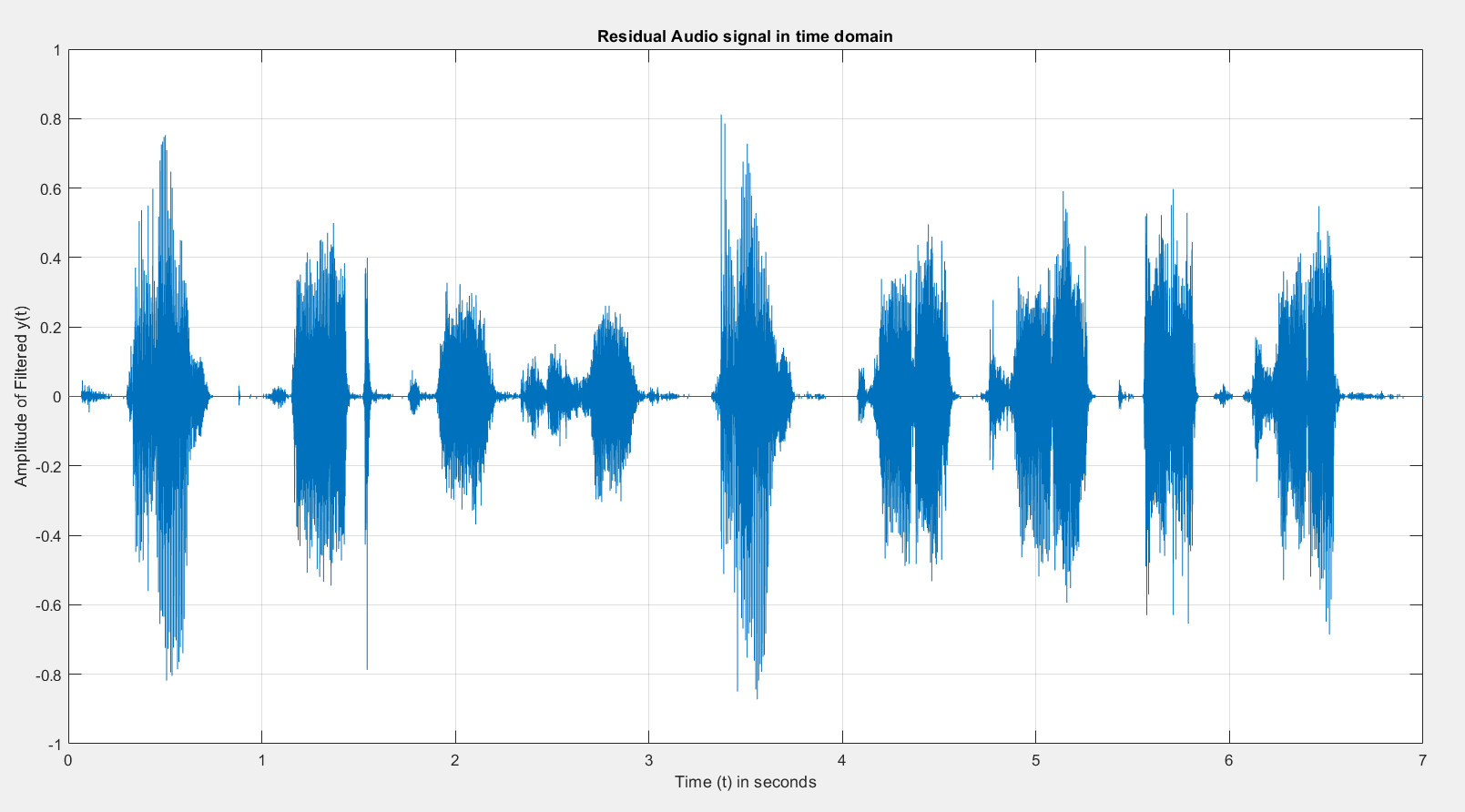
FIR filter equation using generated coefficients:

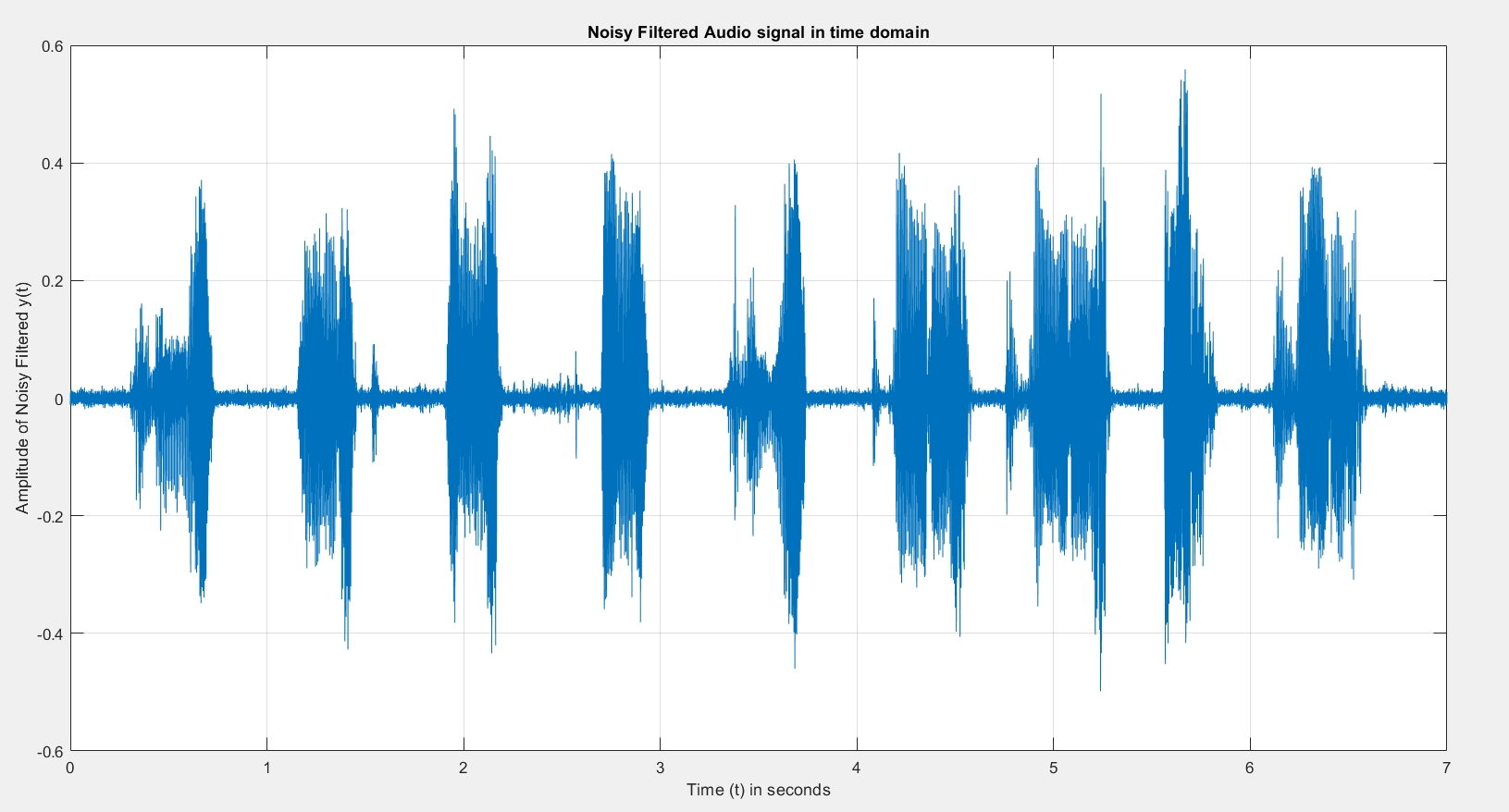
H(z) = 0.008 + 0.01289z-1 + 0.0262z-2 + 0.04667z-3 + 0.0711z-4 + 0.09524z-5 + 0.1145z-6 + 0.1252z-7 + 0.1252z-8 + 0.1145z-9 + 0.09524z-10 + 0.0711z-11 + 0.04667z-12 + 0.0262z-13 + 0.01289z-14 + 0.008z-15

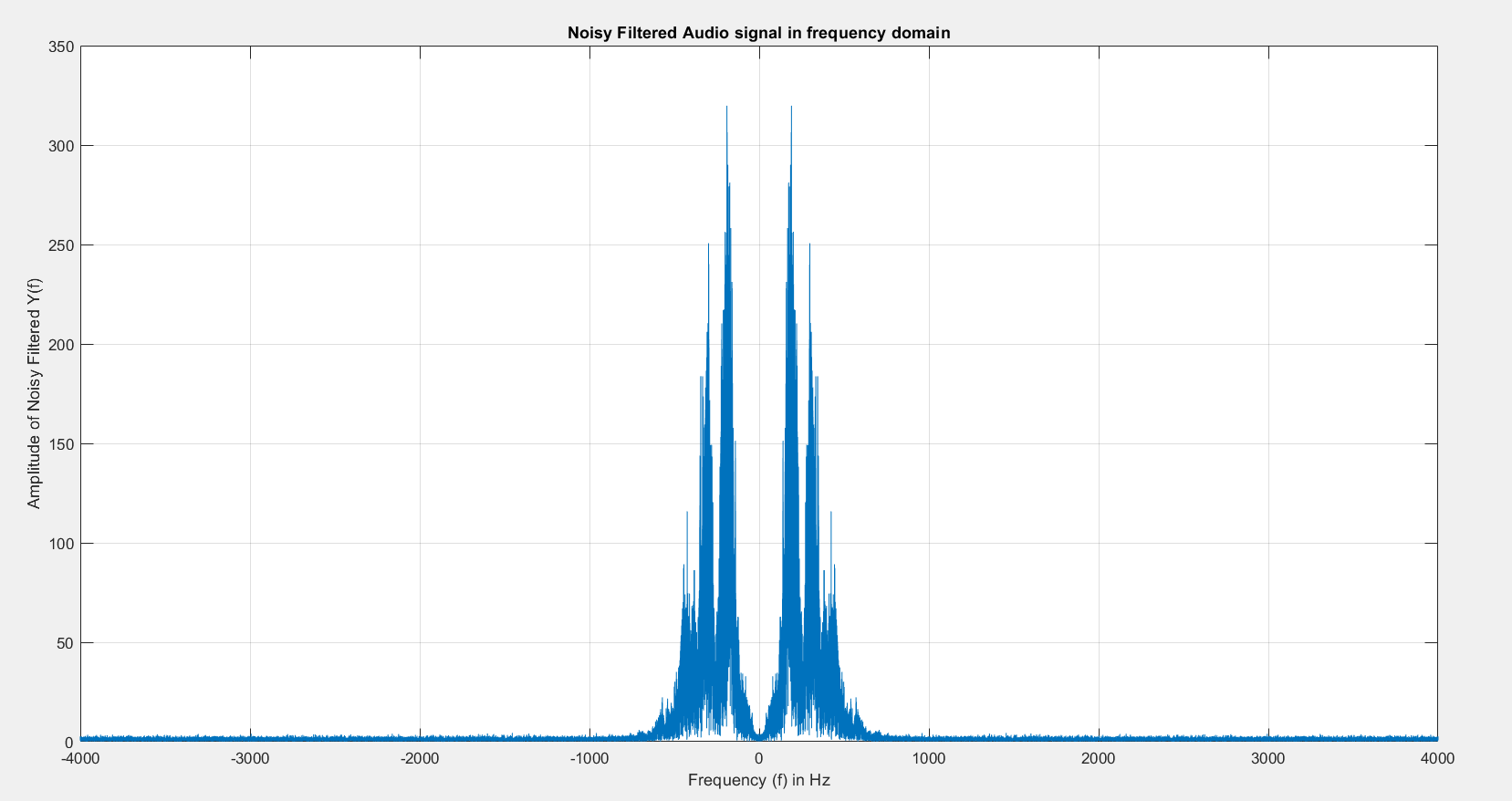
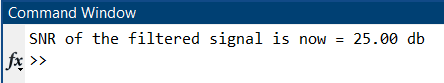












Audio Files: [Input And Output Audio Files](https://drive.google.com/drive/folders/1m5bGAPNbGmTVocYSr63J6IWYiLJEivLm?usp=sharing)

Discussion:

The experiment was to perform a standard low pass filtering on a speech signal. The time domain and frequency domain characteristics of each intermediate signal were observed. The filtered output clearly removed the major high frequency components as desired, but since the roll off is not extremely sharp, some components beyond the cut off were observed. The residual signal was also seen to have low pass components since the filtering in actual filters is imperfect, and has some loss in the pass band as well. The output audio was observed to be slightly noisy(since awgn was added), and fainter. Faint audio is expected since filtering causes attenuation. This is the simulation of an actual process, since noise will get introduced at some stage. Since the SNR is high, the effect of noise was just slightly prominent, but still, modern high quality audio files have SNR of greater than 90 dB.

Appendix

Code:

%%% Name: Hardik Suryaprakash Tibrewal

%%% Roll No.: 18EC10020

%%% DoB: 18/11/2000

clc

clear all

close all

%This is the recording of the audio part

%{

disp('Start speaking');

recObj = audiorecorder;

recordblocking(recObj, 7);

disp('End on recording.');

voice = getaudiodata(recObj);

audiowrite("LabAudio.wav", voice, 8000);

%}

%Reading the audio and plotting in time and frequency domains

[y, Fs] = audioread("LabAudio.wav");

figure();

t = 0:1/Fs:(numel(y)/Fs)-1/Fs;

plot(t,y);

grid on;

xlabel('Time (t) in seconds'); ylabel('Amplitude of y(t)');

title('Audio signal in time domain');

figure();

f\_range = -Fs/2:Fs/numel(y):Fs/2-Fs/numel(y);

Y = fftshift(abs(fft(y)));

plot(f\_range, Y);

grid on;

xlabel('Frequency (f) in Hz'); ylabel('Amplitude of Y(f)');

title('Audio signal in frequency domain');

%Making the required FIR filter using mentioned parameters

%Plotting frequency response of filter

M = mod((1+8+1+1+2+0+0+0),9);

f\_cutoff = 150+10\*M;

B = fir1(15, f\_cutoff/(Fs/2), hamming(16));

figure();

sgtitle("Frequency and Phase response of the filter");

freqz(B,1,512);

%Filtering the signal and making plots

y\_filt = filtfilt(B,1,y);

figure();

plot(t,y\_filt);

grid on;

xlabel('Time (t) in seconds'); ylabel('Amplitude of Filtered y(t)');

title('Filtered Audio signal in time domain');

figure();

Y\_filt = fftshift(abs(fft(y\_filt)));

plot(f\_range, Y\_filt);

grid on;

xlabel('Frequency (f) in Hz'); ylabel('Amplitude of Filtered Y(f)');

title('Filtered Audio signal in frequency domain');

%Residual is the part that was removed by the filter. Finding and plotting

y\_resid = y-y\_filt;

figure();

plot(t,y\_resid);

grid on;

xlabel('Time (t) in seconds'); ylabel('Amplitude of Filtered y(t)');

title('Residual Audio signal in time domain');

figure();

Y\_resid = fftshift(abs(fft(y\_resid)));

plot(f\_range, Y\_resid);

grid on;

xlabel('Frequency (f) in Hz'); ylabel('Amplitude of Residual Y(f)');

title('Residual Audio signal in frequency domain');

%Adding 25 dB average whie gaussian noise to the filter output. Plotting

%relevant plots

y\_filt\_noisy = awgn(y\_filt, 25, 'measured', 'db');

figure();

plot(t,y\_filt\_noisy);

grid on;

xlabel('Time (t) in seconds'); ylabel('Amplitude of Noisy Filtered y(t)');

title('Noisy Filtered Audio signal in time domain');

figure();

Y\_filt\_noisy = fftshift(abs(fft(y\_filt\_noisy)));

plot(f\_range, Y\_filt\_noisy);

grid on;

xlabel('Frequency (f) in Hz'); ylabel('Amplitude of Noisy Filtered Y(f)');

title('Noisy Filtered Audio signal in frequency domain');

%Ensuring the SNR is as desired

fprintf("SNR of the filtered signal is now = %.2f db\n", snr(y\_filt, y\_filt\_noisy-y\_filt));

%Writing audio file

audiowrite("18EC10020\_OUTPUT.wav", y\_filt\_noisy, Fs);