**EC-313 DIGITAL SIGNAL PROCESSING**

**LAB FINAL**

Total Marks: 30

Time: 1 hour

**Write a MATLAB script for the following tasks. Comment the code, give appropriate titles to the figures/plots, and label the axes properly!**

1. Record a mono audio for **5 seconds** at **8kHz** sampling rate and **8 bits per sample** and save it as **‘recAudio.wav’**. **[4]**

recObj = audiorecorder(8000, 8, 1);

disp("Start recording...");

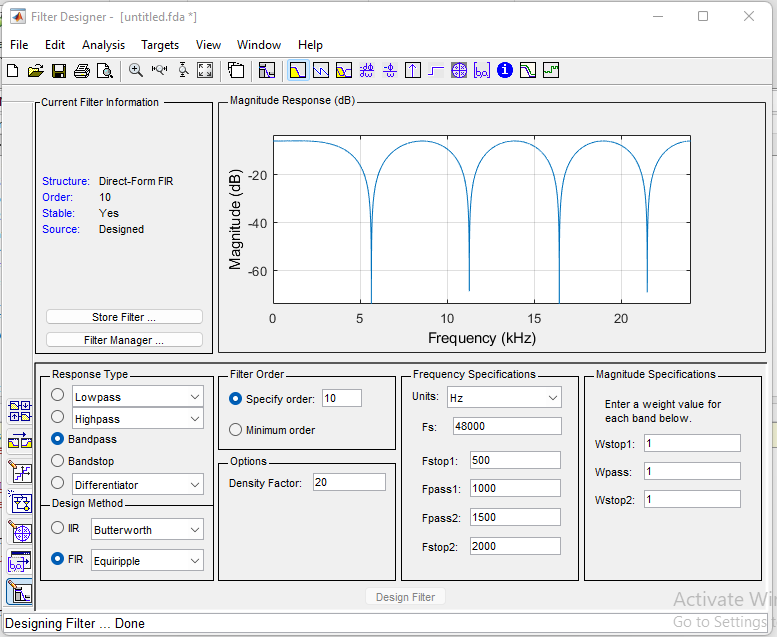
recordblocking(recObj, 5);

disp("Recording complete.");

audioData = getaudiodata(recObj);

audiowrite('recAudio.wav', audioData, 8000);

1. Design a **10th order** FIR Equiriple Bandpass Filter with cutoff frequencies (**1kHz** and **2kHz**) using filter design app or fda toolbox. **[4]**



% designed the filter and obtained the coefficients

filterCoeffs = [0.1511, 0.1009, 0.0718, 0.0570, 0.0525];

% Save the filter coefficients to a MAT-file

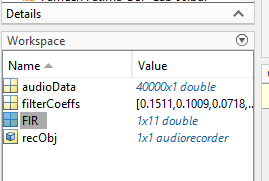
save('filterCoeffs.mat', 'filterCoeffs');

% Display the filter coefficients

disp('Filter coefficients saved to filterCoeffs.mat:');

disp(filterCoeffs);

1. Export the filter coefficients to MAT-file and display them in command window. **[3]**



A screenshot of a computer

Description automatically generated with medium confidence

1. Plot the frequency response, phase response, impulse response, pole-zero plot, and group delay of the filter, using the respective MATLAB commands/functions. **[5]**

Magnitude and phase response:

A screen shot of a graph

Description automatically generated with medium confidence

Phase response:

A picture containing text, screenshot, line, plot

Description automatically generated

Group Delay:  
A picture containing text, screenshot, line, plot

Description automatically generated

A screenshot of a computer screen

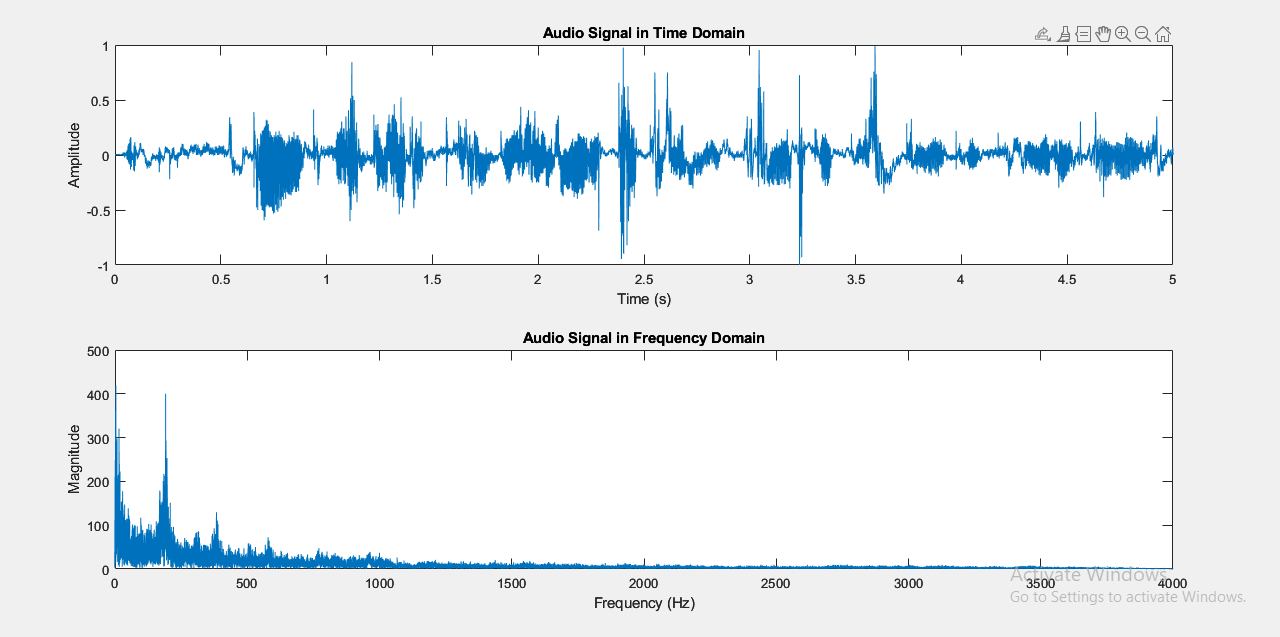
Description automatically generated with medium confidence

Pole-zero plot:

A picture containing text, diagram, line, screenshot

Description automatically generated

1. Read the recorded audio file and plot the signal in time and frequency domain. **[3]**
2. [audioData, sampleRate] = audioread('recAudio.wav');
3. time = (0:length(audioData)-1) / sampleRate;
4. figure;
5. subplot(2, 1, 1);
6. plot(time, audioData);
7. xlabel('Time (s)');
8. ylabel('Amplitude');
9. title('Audio Signal in Time Domain');
10. subplot(2, 1, 2);
11. fftSize = 2^nextpow2(length(audioData));
12. frequency = (0:fftSize/2) \* sampleRate / fftSize;
13. spectrum = abs(fft(audioData, fftSize));
14. plot(frequency, spectrum(1:fftSize/2+1));
15. xlabel('Frequency (Hz)');
16. ylabel('Magnitude');
17. title('Audio Signal in Frequency Domain');



1. Pass the signal through the designed filter and plot the resultant signal in time and frequency domain. **[3]**
2. % Pass the signal through the filter
3. filteredSignal = filter(filterCoeffs, 1, audioData);
4. % Plot the filtered signal in the time domain
5. time = (0:length(filteredSignal)-1) / sampleRate;
6. figure;
7. subplot(2, 1, 1);
8. plot(time, filteredSignal);
9. xlabel('Time (s)');
10. ylabel('Amplitude');
11. title('Filtered Signal in Time Domain');
12. % Plot the filtered signal in the frequency domain
13. fftSize = 2^nextpow2(length(filteredSignal));
14. frequency = (0:fftSize/2) \* sampleRate / fftSize;
15. spectrum = abs(fft(filteredSignal, fftSize));
16. subplot(2, 1, 2);
17. plot(frequency, spectrum(1:fftSize/2+1));
18. xlabel('Frequency (Hz)');
19. ylabel('Magnitude');
20. title('Filtered Signal in Frequency Domain');
21. Resample the filtered signal to **2kHz** and plot the resultant signal in time and frequency domain. **[4]**
22. % Resample the filtered signal to 2kHz
23. resampledSignal = resample(filteredSignal, 2000, sampleRate);
24. % Calculate the new sampling rate after resampling
25. newSampleRate = 2000;
26. % Plot the resampled signal in the time domain
27. timeResampled = (0:length(resampledSignal)-1) / newSampleRate;
28. figure;
29. subplot(2, 1, 1);
30. plot(timeResampled, resampledSignal);
31. xlabel('Time (s)');
32. ylabel('Amplitude');
33. title('Resampled Signal in Time Domain');
34. % Plot the resampled signal in the frequency domain
35. fftSizeResampled = 2^nextpow2(length(resampledSignal));
36. frequencyResampled = (0:fftSizeResampled/2) \* newSampleRate / fftSizeResampled;
37. spectrumResampled = abs(fft(resampledSignal, fftSizeResampled));
38. subplot(2, 1, 2);
39. plot(frequencyResampled, spectrumResampled(1:fftSizeResampled/2+1));
40. xlabel('Frequency (Hz)');
41. ylabel('Magnitude');
42. title('Resampled Signal in Frequency Domain');
43. Play all the three audio signals at their respective sampling rates with a **5 second** pause before each one of them. **[4]**

% Play the original audio signal at its original sampling rate

sound(audioData, sampleRate);

pause(5);

% Play the filtered signal

sound(filteredSignal, sampleRate);

pause(5);

% Play the resampled signal at its new sampling rate

sound(resampledSignal, newSampleRate);

Best of Luck!