Homework 3: Noise Removal using Digital Filters

CMPE362: Intro. to Signal Processing

Objective

The purpose of this assignment is to design and analyze digital bandstop filters for noise removal in audio signals. You will explore FIR and IIR filters using MATLAB (please also download Signal Processing Toolbox), and investigate their performance.

Scenario

Due to an unknown electrical malfunction in the recording equipment, audio recordings of a mathematics lecture have been corrupted with noise. The noise seems to be localized in a particular frequency range. You are tasked with designing a filter that can eliminate this noise.

For analysis, a sample file `sample.wav` is provided. If you have a hard time hearing the noise in the background, note that the noise resembles the sound of cicadas that you might hear during the summer season.

Instructions

1. Using the `spectrogram()` function in MATLAB, analyze the spectrogram of the original audio signal. Based on your inspection, decide on a frequency range (f_1, f_2) where the noise is concentrated (Hint: direct your attention to the silent region in the audio). Note that there is no exact answer as the boundaries of this range are fuzzy.

   The function `spectrogram()` accepts many arguments. Therefore, it will not be directly useful. Please check the examples provided in MATLAB documentation for this function to find a suitable parameter set.

   Report: Include the spectrogram in your report along with 1-2 sentences explaining how you selected the values (f_1, f_2). For the spectrogram: the x-axis should be time in seconds and the y-axis should be frequency in Hz or kHz.

2. Design a 256th-order FIR **bandstop** filter using the `fir1()` function. Use $(f_1 - 500, f_2 + 500)$ as cutoff frequencies. If your estimated frequency range (f_1, f_2) is sufficiently accurate, this filter should significantly reduce the noise (almost inaudible).

3. For the same frequency range as previous step, design IIR **bandstop** filters using the following functions:

- Butterworth filter: `butter()`
- The Chebyshev Type I filter: `cheby1()`
- Elliptic filter: `ellip()`

For `cheby1()` and `ellip()`, set the R parameter as 0.1.

4. For each of the three IIR filters from the previous step, start with an arbitrary n value such as $n = 3$. Plot the log(10)-magnitude of the frequency responses of FIR, Buttersworth, Chebyshev and Elliptic filters as exemplified in Figure 1. (This figure is just an example and not the final answer.)

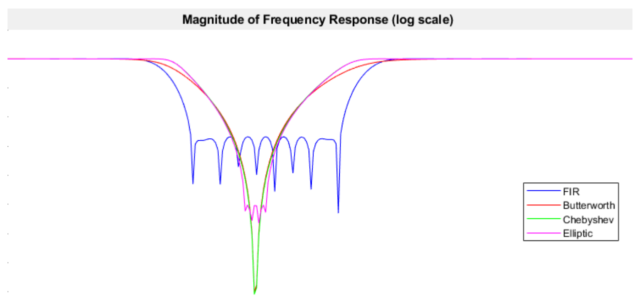

Figure 1: Example plot for magnitude of the frequency response of each filter

Play around various n values for IIR filters, check the freq. response plot and also apply the filter to the audio and check if the noise is audible. Through this trial-and-error back and forth process, try to find low n values for each IIR filter that performs similar to the FIR filter from step 2. Here the term "similar performance" is not very well defined and subjective. However, a similar performing filter might have similar frequency response and also it should reduce the noise as well as the FIR filter.

Report: After you decided on an order value n for each IIR filter (they might be different):

- Include n values you decided for each filter in your report. An extra information: IIR filter design functions that you have used in step 3 will generate filters of order $2n$ when used with parameter `"stop"`.
- Include the final frequency response plot in your report. It should be similar in style to the one shown in Figure 1. The y-axis should contain log(10) of the magnitude of frequency responses. The x-axis should contain the frequencies between 2000 and 8000 Hertz.
- Include the pole-zero plots of the three IIR filters in your report as exemplified in Figure 2. (This figure is just an example and not the final answer.)

5. **Report**: Apply your filters (1 FIR and 3 IIR filters) to the noisy signal `sample.wav` separately and plot the resulting spectrograms. Also save the resulting audio files. We expect to see that the noise is visibly significantly suppressed (or nearly eliminated) in the targeted frequency range. Include the spectrograms in your report.

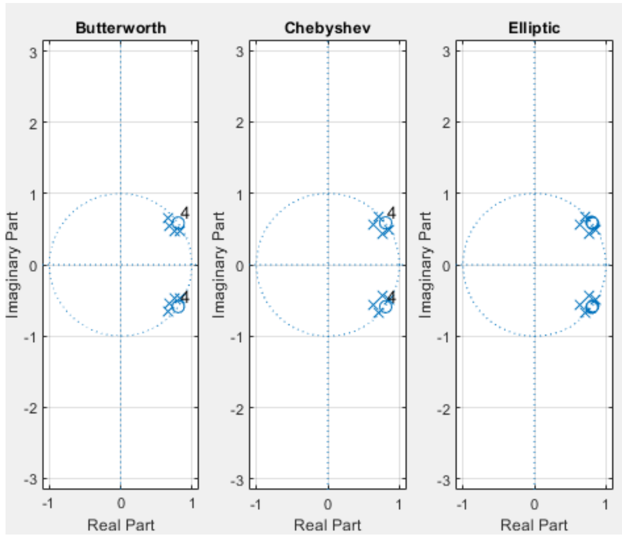

Figure 2: Example zero pole plots for IIR filters.

6. As you increase n there's a point at which the IIR filter becomes unstable and generate numerical stability issues. Identify the smallest parameter n for which the corresponding IIR filter becomes unstable. You can observe that using such a filter on `sample.wav` will corrupt the original audio unrecognizably.

   Report: After finding these n values for each IIR filter, generate the same plot as in the Figure 1 for IIR filters with these n values. Include this plot in your report along with corresponding n values. Investigate the plot for signs of instability. Briefly discuss any visual or numerical clues that could indicate instability.

Helpful MATLAB Functions

You may find the following MATLAB functions useful in this homework:

- `audioread`, `audiowrite`
- `spectrogram`, `abs`, `log10`
- `fir1`, `butter`, `cheby1`, `ellip`
- `freqz`, `zplane`
- `filter`

Submission and Report Summary

Your report should include:

- Spectrograms before and after filtering

- Filter design parameters and comparison
- All plots (z-plane, frequency response)
- Brief explanations and interpretations for each step

Please submit your MATLAB code, your report in PDF format and resulting .wav files in step 5. If have file size issues with Moodle, send an e-mail to me.