Lab 2

Igor Fedorov
Yihan Hu

February 18, 2021

1 Grading

Each lab will consist of a series of tasks, which you must complete in order to receive full credit. Your work must be summarized in a written report, which will be graded by your TA. You may use your favorite word processor. We will give 5% extra credit per lab for reports completed in LaTex. You may use NeurIPS templates in LATEX and Word, available from the 2015 NeurIPS format site. The page limits mentioned there don't apply.

For **each** task, generate the required figures/plots/etc. and provide a brief explanation or summary of your findings if asked. You may also be asked to attach Matlab/C code to the end of your report.

2 Objectives

In this lab, you will learn how to

1. Understand and implement the echo effect
2. Understand the concept of designing a notch filter from a lowpass filter
3. Design and implement a simple FIR notch filter
4. Design and implement a simple IIR notch filter

3 Instructions

As usual, we will begin by implementing a series of DSP operations in MATLAB and then port our implementation to C.

4 Instructions

Build upon the code from Lab 0 to complete the following tasks

4.1 Task 1

Read the provided audio file `speech.wav` into MATLAB and store it in a variable called x. Make sure to min-max normalize x so that it lies in the range $[-1, 1]$. Create a new variable called y, which is constructed by x with an echo, using

$$y[n] = 0.7x[n] + 0.3x[n - N_e] \tag{1}$$

where N_e is the delay of the echo, in samples. Generate y such that N_e corresponds to a $0.5s$ delay at $F_s = 48\ kHz$ using the provided file. Describe what you hear in the report.

4.2 Task 2

In this task, we will add a sinusoidal interference to x. You can use your code from Lab 0 `generate_sigs()` to create a sinusoidal signal that has the same length as x with frequency 2.4 kHz, sampled at 48 kHz, denoted by $v[n]$. Generate the corrupted signal as the following:

$$w[n] = x[n] + v[n]. \tag{2}$$

Generate spectrograms of x and w. Explain the difference between the two spectrograms and how it relates to your knowledge of the signal model.

4.3 Task 3

Find the filter coefficients of a 14th order notch filter with notch at 2.4 kHz using only an ideal low-pass filter. Explain the reasoning behind your method and show your derivation if you can (review class lecture). Use `freqz` to plot the magnitude response of your notch filter and verify your design.

4.4 Task 4

Write a MATLAB function `fir_filter(w,h)` which takes two arguments w and h. It applies an FIR filter with given coefficients h to a given signal w. Pass the noisy signal w you generated in Task 2 through this function with the filter coefficients you found in Task 3 to obtain the filtered signal s. Generate the spectrogram of s and compare with the spectrogram of the noisy signal v. Comment on the spectrogram results.

4.5 Task 5

Find the poles and zeros of a 2nd order IIR notch filter with a notch at 2.4 kHz. Explain where you would place the zeros and poles of a 2nd order system to obtain a notch at 2.4 kHz. After you find the coefficients of the IIR filter, reuse `iir_filter(x,a,b)` from Lab1 to filter a given signal. Use this function to filter w and obtain m. Describe what you hear and generate a spectrogram of the filtered output m. Comment on the results.

4.6 Task 6

Write a C program to implement the echo effect. Pass the provided speech audio file through your code and write the echo version of that file.

4.7 Task 7

Write a C program to implement the FIR notch filter. Pass the provided noisy speech audio file through your code and write the FIR filtered version of that file.

4.8 Task 8

Write a C program to implement the IIR notch filter. Pass the provided noisy speech audio file through your code and write the IIR filtered version of that file.

5 Report Outline with Rubric (25 pts)

1. Title (1pt): The title has to be informative and not generic like 'ECE161B Lab0 Report'. Also you have to include author's list.
2. Introduction (1pt): A short description of what you did. Please mention any key findings and interesting insights.

3. Task 1 (2 pts): A short description of what your hear. Include MATLAB code in the appendix of the report.
4. Task 2 (2+1 pts): Two spectrograms figure with correct axes and labels. A short description of the spectrogram and the signal model. Include MATLAB code in the appendix of the report.
5. Task 3 (2+1 pts): A short description of your filter design approach. Include math derivation as needed. Attach the magnitude response of your FIR filter.
6. Task 4 (2+1+1 pts): A spectrogram of the target signal and describe the difference. Include MATLAB code in the appendix of the report.
7. Task 5 (2+2+2 pts): A short description of your filter design approach. A spectrogram of the target signal and discuss what your hear as well as the results. Include MATLAB code in the appendix of the report.
8. Task 6 (2 pts): Full completion of programming and submit the code to gradescope.
9. Task 7 (2 pts): Full completion of programming and submit the code to gradescope.
10. Task 8 (2 pts): Full completion of programming and submit the code to gradescope.
11. For the source code-the most points will be given for well-documented and clean code.
12. Please submit your C source code as well as tex file if you'd like the extra credit.