University of Washington ECE Department
EE 242 Lab 4 – Digital Filtering

In this lab, we will consider different types of digital filters (specifically discrete-time, linear, time-invariant filters) and look at their characterization in both time and frequency. This will give you some insight into how digital filters are implemented and into the properties of different digital filter design algorithms. You'll also learn about some of the signal processing functions available from the signal module in the scipy package which will be useful for designing and implement filters. You will work with examples that show you how filtering can be useful to remove noise and reshape the frequency content of a signal. Specifically, we'll revisit the lab 2 problem of removing noise from signals (or smoothing signals), then explore filter design methods, and finally implement a simple audio equalizer. This is a 2-week lab. It is recommended to work on the first X assignments in the first week and the remaining assignments in the second week.

Lab 4 Turn-in Checklist

- 3 pre-lab exercises, to be completed individually and uploaded to canvas before the first lab
- Lab report (Jupyter notebook), completed and submitted as a team
- Jupyter notebook for individual assignment to be uploaded to canvas

Pre-lab

Read the Lab 4 Background document, then complete the following exercises.

1. In lab 2, assignment 2, you implemented a smoothing system using convolution with a box of length N and height 1/N: $h_1[n] = (u[n] - u[n-N])/N$. Find the coefficients {a, b} of the linear constant coefficient difference equation (LCCDE) describing this system for N=10. Find the coefficients for the system $h_1[n] = 0.8^n u[n]$
2. Find the normalized frequency (for use in the filter design functions) that corresponds to a DT filter cut-off of $\pi/4$. What normalized frequency corresponds a cut-off of 500Hz if the sampling frequency is 11,025 Hz?
3. Provide an equation for converting gain G in dB to a linear scale. For example, a 6dB gain translates to a factor of 2, so y[n]=2x[n] is an allpass filter with a gain G=6dB.

Lab Assignments

This lab has 4 team assignments. Each should be given a separate code cell in your Notebook, and each should be associated with a markdown cell with subtitle and discussion. You should aim to get at least the first two assignments done in the first week. As in previous labs, your

notebook should start with a markdown title and overview cell, which should be followed by an import cell that has the import statements for all assignments. For this assignment, you will need to import: *numpy*, the *wavfile* package from *scipy.io*, *simpleaudio*, *matplotlib.pyplot*, and the *signal* package from *scipy*.

Assignment 1: Different Filter Implementations

In this lab, we will be using standard tools to design filters, and we'll want to view them in both the time and frequency domain. In this assignment, you will write and test functions for plotting the frequency response and the impulse response of a system given the filter coefficients {a, b}. This assignment will have four parts, A-C.

A. The response that is most often illustrated is the magnitude frequency response on a dB scale. Write a function that takes as input the filter coefficients, an optional flag for plotting both the magnitude and phase of the frequency response, and an optional sampling frequency. The function should generate either a plot of the magnitude or both the magnitude and the phase side-by-side, depending on the flag, with the default being magnitude only. The magnitude of the frequency response should be plotted on a dB scale with a range of [-100,0]. If no sampling frequency is provided, use radians for the frequency axis; otherwise use a Hz scale.

B. Write a second function that takes as input the filter coefficients and a desired impulse response length, computes and returns the impulse response, and also plots the impulse response using a stem plot.

C. Test the functions by plotting the magnitude, phase and impulse responses of two lowpass filters with a frequency cut-off of 0.15. One should be an FIR filter designed using the signal.firwin function (order 20) and the other should be an IIR filter with the signal.butter function (order 10).

Report discussion: Comment on the differences between the two filters in terms of the magnitude, phase and impulse responses. What are the tradeoffs associated with these differences?

Assignment 2: Different Filter Implementations for Smoothing Signals

In lab 2, assignment 2, you experimented with smoothing a noisy signal using a moving average window and a convolution. The convolution used an impulse response h[n] that was a causal version of the moving average window. In this problem, you will implement the smoothing function using the both convolution and the signal.lfilter command, to see that they give the same result. This assignment will have three parts, A-C.

A. Using the code from lab 2, create a **base** time signal and a noisy version of it by adding random noise generated with the numpy.random.randn() function (the standard normal

distribution, which is zero mean and unit variance). Plot the original and noisy signals together with the original overlaid on the noisy version, with the time axis labeled assuming a sampling rate of 1000 Hz. Constrain the y-axis to be [0,25] for all plots. Include a legend with the plot.

B. Create one smoothed version of the signal called **filtsig1** by using the convolve function from lab 2 with the box impulse response and k=10. Create a second version called **filtsig2** by using the signal.lfilter function. Recall that for the FIR filter, the impulse response is equal to the b coefficient vector. Plot the two filtered signals overlaid. Recall that the convolve function will change the length, so you will need to define a new time vector for that. You should find that the two methods give the same result except for edge effects.

C. Use the function that you wrote in assignment 1 to plot the magnitude and phase for the frequency response of this filter. It should look like a low pass filter.

Report discussion: The moving window average (and its causal version) is an FIR filter, so the phase should be linear. How might the result change if you used a Butterworth filter?

Assignment 3: Filtering an Audio Signal

In this assignment, we'll explore different filtering problems using the horn signal that you worked with in Labs 2 and 3. You will need the audio packages that you used in previous labs. This assignment will have three parts, A-C.

A. Read in the horn sound *horn11short.wav*. Design a highpass filter with a cutoff of 550Hz to remove the first harmonic. A Butterworth filter of order 8 should work. Plot the magnitude response and the impulse response of your filter. Using the same section of the signal as in lab 3, compute the FFT of the original and the filtered signal, and plot the log magnitude frequency content in a 2x1 plot. Play the original and the filtered waveforms.

B. Create a noisy version of the horn sound by generating a sequence of random noise, as in lab 2, but with a scaling factor of 1000, and then adding that to the horn signal. Plot the log magnitude of the frequency content for the two signals, as in part A. Play the resulting signal and confirm that the noise is audible.

C. Design a lowpass filter with a cut-off corresponding to 1800Hz to remove the high frequency noise. (You can use a lower cut-off to remove more noise, but then you start removing audible harmonics.) Plot the log magnitude response and the impulse response of the filter. Plot the frequency content of the filtered signal.

Report discussion: In part A, filtering out the first harmonic of the horn signal reduces the loudness, but it doesn't change the perceived note. Explain why this is. Discuss the differences in the impulse responses of the HPF and LPF.

Assignment 4: Implementing a 3-Band Audio Equalizer

In this assignment, you will design a music equalizer, which breaks the sound file into multiple frequency bands by filtering, weighting, and summing to reconstruct the signal**.** To keep it simple, the equalizer will have only 3 bands, as illustrated in the background document: an LPF, a BFP and an HPF. This assignment will have four parts, A-D.

A. Design 3 5th-order Butterworth filters using the cut-off frequencies: $\boldsymbol{\pi/4}$ and $\boldsymbol{\pi/2}$. Plot the magnitude response of the 3 filters together in one plot

B. Write a function that takes as input a sound signal and three gains in dB, and outputs an equalized version of the signal. The function should use the filters that you designed to create 3 different components, then multiply each component by its respective gain (converted from dB to linear scale using your prelab result), and then sum up the re-weighted components to get the equalized result.

C. Read in music.wav file provided. Apply the equalizer to the music with $\boldsymbol{G_1 = G_2 = G_3 = 0\ dB}$ and play the output. Verify that it sounds the same as the original input.

D. Experiment with different sets of gains (for example $G_1 = G_2 = 0\ dB$ and $G_3 = -40\ dB$). Create two examples where the filtered version sounds different.

Report discussion: Discuss what types of gains lead to an audible difference. Are there any constraints you need to put on the gains?

Team Report

When you've tested and cleaned up all your code (remember, you should only submit code for the Assignments, each in their own cell), go to 'File' then 'Download as', then select '.ipynb'. The file you download is a Notebook that your TA will be able to open and grade for you, once you submit it on Canvas. Remember, only one notebook per team! Make sure that your notebook is titled Lab4-XYZ.ipynb, where XYZ are the initials of the lab partners.

Individual Assignment:

In this assignment, you will design your own 5-band audio equalizer.