

## CSE 3313 – Homework #8

---

Assigned: Monday, April 13, 2020

Due: Monday, April 20, 2020 at end of class

Note the following about the homework:

1. You must show your work to receive credit.
2. For the hand-written problems, submit a hard-copy.

### Assignment:

#### Applications

1. (100 points) **Application Area: Shelving Filter to Change Gain of Low Frequencies**  
Purpose: Learn to amplify or attenuate a particular range of frequencies.

Radios often have dials that allow the bass (low) or treble (high) frequencies to be changed as a group. These can be implemented by a type of IIR filter called a *shelving filter*.

- (a) The course website has a skeleton file called `shelvingFilter.py` that you should complete. It calls a function named `applyShelvingFilter()`, which has the signature

`applyShelvingFilter(inName, outName, gain, cutoff)`

You will create `applyShelvingFilter()`. While it's OK to decompose your code into multiple functions, I should be able to call `applyShelvingFilter()` with these parameters and have everything work. These parameters are:

- **inName**: a string representing the name of an audio file in the WAVE format to read and process
- **outName**: a string representing the name of an audio file that you will write in the WAVE format after applying the filter
- **gain**: an integer that represents the gain  $g$  of the filter. This is how much to increase or decrease the relevant frequencies and therefore could be negative.
- **cutoff**: an integer that represents the cut-off frequency  $f_c$  of the filter. The modified magnitudes will be for frequencies in the range of 0 Hz to the cut-off frequency.

Assume that the audio file to read (**inName**) and the audio file to write to (**outName**) will both be in the current directory.

- (b) On the course website is a file, `P_9_1.wav`, that you can use; it's from [SB96]. You don't necessarily have to use this file, but your program should read a file in the WAVE format using the sound library we have used for previous assignments.
- (c) Also on the course website is a copy of the relevant chapter of [LDKN00], which describes the process that your shelving filter will apply.

(d) Your program should also do the following:

- i. Plot the following side-by-side using subplots:
  - A. the magnitudes of the fft of the original signal with the x-axis in Hertz. This x-axis labels should be their actual frequencies.
  - B. the magnitudes of the fft of the filtered signal with the x-axis in Hertz. This x-axis labels should be their actual frequencies.
- ii. If the fft of the original signal has  $N$  values, plot only the first  $N/4$  values of both the original and filtered signals.
- iii. The y-axis of both plots should be from 0 to the maximum magnitude of the two signals, plus 100. For example, if the maximum magnitude for the original signal is 2000 and the maximum magnitude for the filtered signal is 1500, the y-axis for BOTH plots should be from 0 to 2100. See Figure 1 for an example (your plot will depend on the gain and cut-off chosen).

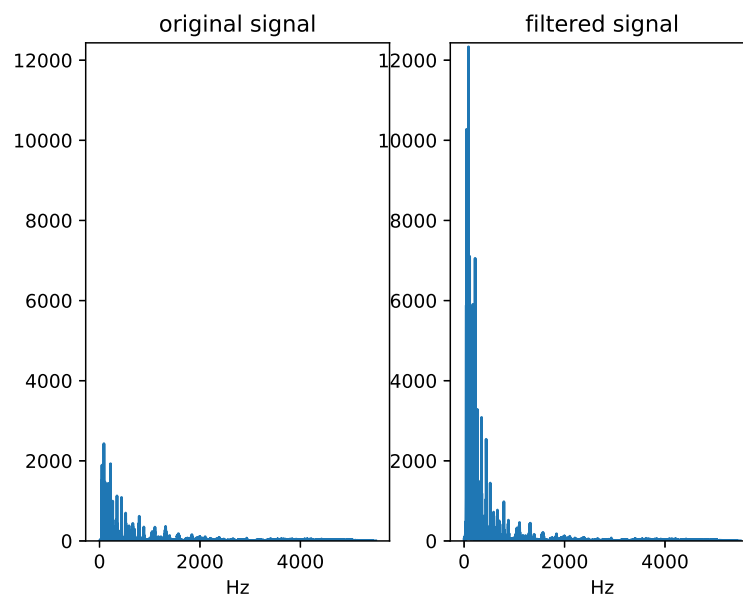


Figure 1: Comparison of signal after applying shelving filter

- (e) Save the processed signal as `outName` using the sound library we have been using in our assignments.
- (f) On your own play around with positive and negative values for the gain and different cut-off frequencies to see how they affect the sound. Also listen to the original signal to see what is happening.

General requirements about the Python problems:

- a) **As a comment in your source code, include your name.**
- b) The Python program should do the work. Don't perform the calculations and then hard-code the values in the code or look at the data and hard-code to this data unless instructed to do so.
- c) The program should not prompt the user for values, read from files unless instructed to do so, or print things not specified to be printed in the requirements.

To submit the Python portion, do the following:

- a) Create a directory using your last name, the last 4 digits of your student ID, and the specific homework, with a hyphen between your ID and the homework number. For example, if John Smith has a student ID of 1000123456 and is submitting hw02, his directory would be named **smith3456-hw02**. Use all lowercase and zero-pad the homework number to make it two digits.

If you have a hyphenated last name or a two-part last name (e.g., Price-Jones or Price Jones), let's discuss what you should do.

- b) Place your .py files in this directory.
- c) Zip the directory, not just the files within the directory. You must use the zip format and the name of the file (using the example above) will be **smith3456-hw02.zip**.
- d) Upload the zip'd file to Canvas.

## References

- [LDKN00] John Lane, Jayant Datta, Brent Karley, and Jay Norwood. *DSP Cookbook*. Prompt Publications, 2000.
- [SB96] Virginia Stonick and Kevin Bradley. *Labs for Signals and Systems Using MATLAB*. PWS Publishing Company, 1996.