George Mason University

Signals and Systems I Fall 2016

Laboratory Project #6

Assigned: Week of November 21, 2016
Due Date: Laboratory Section on Week of December 5, 2016

Report Format and Guidelines for Laboratory Projects

Your lab report for each lab is a formal report in the sense that it should be well-prepared, carefully written, and follow the format given below. It must include descriptions and analysis of all parts of the lab that have been assigned. It may help in the preparation of your report to assume that your report is being written for your boss at the place where you now work as a newly-hired engineer, and that this report represents your response to his request for an analysis of a problem that you have been given to work on.

General Guidelines

- Your report should be typewritten, neat and well-organized.
- The report must be well-organized and follow the format given below. It must include all analytical work, MATLAB plots and code, and relevant explanations. The analytical work and calculations should be complete and clearly explained.
- It is expected that the report will be grammatically correct with no spelling errors. Part of your grade will be based on the grammatical correctness of your report, and points will be deducted for spelling errors. If you prepare your report in Word, there is a spell checker that you are encouraged to use.
- Explanations should be given that describe your work, and plots must include properly labeled axes and a title. Each plot should be a figure, with a figure number and a caption. Although it is important to be complete and thorough in your report, points will be deducted if you present too many unnecessary graphs and plots.
- Within the text of your report, when referring to a particular plot, refer to it by number. For example, you may write something such as

  > "The frequency response of the bandstop filter is shown in Figure 2, and from this plot we see that the center frequency of the stop-band is 10 kHz and the stop-band width is 2 kHz as desired."

Detailed Report Format

1. Title Page: This page contains an identification of the Laboratory by title and number. It also has the name and G-Number of the author as well as the date of submission.

2. Introduction: This section contains a description of the purpose and objectives of the laboratory project. Do not simply copy test from the assignment. It should also summarize the topics covered in the laboratory and a brief summary of the key results obtained.

3. Main body: This section is the most detailed part in your report. Note that it must **not** contain any MATLAB code. It does contain a description of the results obtained including any figures that were generated with MATLAB or other sources. The report body should contain subsections that are organized in a manner similar to the laboratory assignment. All figures must have a caption that is used when referring to them in the body of the report. If theoretical calculations are required, these should be detailed and used in comparisons with the experimental portion of the lab.

4. Observations and Recommendations: In this section, you should summarize the knowledge you gained in this lab, any comments you may have on the lab including how it could be improved.

5. References/Appendices: If you used any references (e.g. web pages) identify them here. You should also include appendices labelled Appendix A, Appendix B, Appendix C, etc. for each experiment for which you wrote MATLAB code. In these appendices, do not list every MATLAB instruction that you typed in the MATLAB command line. Only provide those that would be necessary to repeat any of your experiments or to create the plots that you present in the report. Each MATLAB script file, or sequence of MATLAB commands must be fully documented, and before each listing in the appendix, you should provide a description of what the script file or MATLAB commands are used for.

General Lab Description:

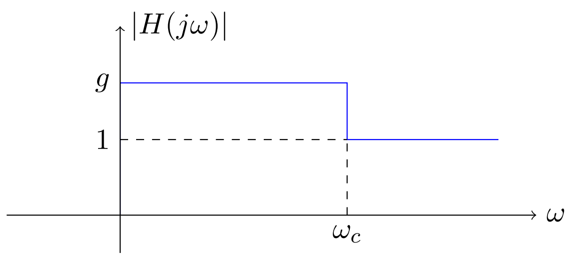
The objective of this laboratory is to reinforce your understanding of the frequency response of LTI systems. Your will explore how LTI systems alter the frequency content of a signal. In parts 2 and 3 of the lab, you will design low and high shelf filters, study their frequency-domain properties, and apply them to single frequency sinusoids. In part 4, you will use your shelf filters to perform equalization in order to balance the the bass (low frequency) and treble (high frequency) parts of a short piece of music, and thus to improve the perceptual sound quality.

1 Prelab

Review relevant sections of your text (Section 7.5 and others) as well as the lecture notes on low-pass and high-pass filters. Review the MATLAB documentation on functions `tf`, `lsim` and `freqs`. Read the MATLAB help files on `pzplot`. You will be using the MATLAB m-file `spectrogram`. Read the help file on this function as well as documentation that can be found on the internet. One purpose of this lab is to ensure that you understand the `spectrogram` function.

2 Design of a Second Order *Low-Shelf Filter*

In general, *shelf filters* are used to amplify or attenuate the signals over a defined frequency range while leaving signals unchanged that are at other frequencies. The *low-shelf filter* is designed to have a gain of g for $|f| < f_c$ where f_c is the cut-off frequency of the filter. Depending on the type of filtering required, the gain g may be either greater than one or less than one. The filter response $|H(j\omega)|$ shown below is an example for $g > 1$.



The general form of the transfer function of a second order low shelf filter is given by

$$H_{ls}(s) = \frac{(s + \omega_c\sqrt{g}e^{j\pi/4})(s + \omega_c\sqrt{g}e^{-j\pi/4})}{(s + \omega_c e^{j\pi/4})(s + \omega_c e^{-j\pi/4})} = \frac{b_2 s^2 + b_1 s + b_0}{a_2 s^2 + a_1 s + a_0} \tag{1}$$

With $s = j\omega$, it is easy to show that $|H_{ls}(j0)| = g$, $|H_{ls}(j\infty)| = 1$, and

$$|H_{ls}(j\omega_c)| = \sqrt{\frac{g^2 + 1}{2}}$$

(a) Write a MATLAB function named `design_shelf` to compute the numerator and denominator coefficients of the transfer function. The inputs to your function should include the gain, g, and the cut-off frequency, ω_c (in rad/sec). The outputs will be the vectors of numerator and denominator coefficients: $B = [b_2\ b_1\ b_0]$ and $A = [a_2\ a_1\ a_0]$

Make sure you include a help section at the top of the m-file that contains a block of comments explaining what the input and output arguments are, and what the function is expected to do. The help section will show up in the command window when you type `help design_shelf`.

Note that the MATLAB function `poly` can be used to compute the coefficients of a polynomial from its roots. For example, the 2^{nd}-order polynomial with roots 2 and 3 is found from `poly([2 3])` as: `ans = 1 -5 6`.

(b) Use your function `design_shelf` to design a low-shelf filter with a cut-off frequency of 300 Hz, i.e. $\omega_c = 2\pi(300)$, for two different gain values: $g = 10$ and $g = 0.1$. Use the MATLAB function `pzplot` to plot the poles and zeros of your low-shelf filter. Comment on how the two plots (for the two different gains) are different and verify in each case that the poles and zeros are in the correct locations.

(c) Use the MATLAB function `freqs` to plot the frequency responses of your filters in db. Specifically, plot of $20\log_{10}|H(j\omega)|$ versus ω. Use at least 1000 points to ensure smooth plots. Place both plots in the same figure using the `subplot` command. Verify that $|H_{ls}(j\infty)|$ is 0 dB and that $|H_{ls}(j\omega_c)|$ is 3 db below the maximum gain. Compare the plots and comment on the relationship between gain value and the frequency response of the filter.

(d) Generate three sinusoidal signals of frequencies 30 Hz, 300 Hz, and 3000 Hz. Select the range of each time vector so that it contains at least ten periods of the signal. Use small sample spacing in order to generate smooth curves. Use the MATLAB command `lsim` to filter each sinusoid with the low-shelf filter designed in part (b) with both gain values.For each gain, make a figure with 6 subplots (3 rows and 2 columns). In the first subplot of each row plot one of the three original sinusoids and in the second subplot plot the corresponding filtered version. Comment on your observations and compare the behavior of the low-shelf filters with two different gains.

3 Design of a Second Order *High-Shelf Filter*

A *high-shelf filter* is the inverse of a *low shelf-filter*. The *high-shelf filter* amplifies or attenuates all frequency components *above* a cut-off frequency, ω_c, and leaves those components below ω_c unchanged. The transfer function of a second order high shelf filter can be derived from Eq. (1) by replacing s/ω_c by ω_c/s. The result, after some simplifying steps, is

$$H_{hs}(s) = \frac{g(s + \frac{\omega_c}{\sqrt{g}}e^{-j\pi/4})(s + \frac{\omega_c}{\sqrt{g}}e^{j\pi/4})}{(s + \omega_c e^{-j\pi/4})(s + \omega_c e^{j\pi/4})} = \frac{b_2 s^2 + b_1 s + b_0}{a_2 s^2 + a_1 s + a_0}. \tag{2}$$

As was the case for the *low-shelf filter*, it is straightforward to verify that the *high-shelf filter* has a gain of 1 at $\omega = 0$, a gain of g at $\omega = \infty$, and a maagnitude spectrum given by:

$$|H_{hs}(j\omega_c)| = \sqrt{\frac{g^2+1}{2}}$$

Note that the poles of the transfer function are the same as the poles of a low-shelf filter.

(a) Modify the function `design_shelf` you wrote in part 2(a) to allow it to find the coefficients of either a low-shelf or a high-shelf filter based on the user's input. One approach is to add a third input for the function that defines the desired filter type. This input can be either numerical (e.g. 0 for low-shelf and 1 for high-shelf) or a string (e.g. 'L' for low shelf and 'H' for high shelf). Within your function, use either an *if statement* or a *switch-case* structure, conditioned on the filter-type input, in order to decide which set of filter coefficients will be assigned to the function outputs.

(b) Use your function to design a high-shelf filter with a cut-off frequency of 300 Hz, $\omega_c = 2\pi(300)$, and two different gains values $g = 10$ and $g = 0.1$. Use the `pzplot` function to plot the poles and zeros of the high-shelf filter. Comment on how the plots for the two different gain values differ and verify that the poles and zeros are in the right place in both cases. Compare the pole-zero plots of the high shelf filter with those of a low-shelf filter you generated in part 2(b) and comment on the differences/similarities.

(c) Use the `freqs` function to compute the frequency response of your two high-shelf filters. Plot the log magnitudes of the frequency response in db of these filters in the same window using two subplots. Verify that $|H_{hs}(0)|$ maps to 0 dB and $|H_{hs}(j\omega_c)|$ is 3 dB below the maximum gain. Comment on how the gain affects the frequency response of the filter. Compare the plots to the plots of the frequency responses of the low-shelf filters you generated in part 2(c).

(d) Repeat part 2(d) for the high shelf filter. Comment on your observations and compare the outputs of the two high-shelf filters (different gains) to each other and to the outputs of the two low-shelf filters in part 2(d).

4 Music Equalization - Bass/Treble Boost and Cut

Low-shelf and high-shelf filters are commonly used in audio equalization, and are usually referred to as bass and treble filters respectively. Equalization is a process in which the perceptual sound quality of an instrument is improved or certain instruments are made more/less prominent. For example, an equalizer can be used to make the vocal part of a song (high pitch) louder while making the bass guitar (low pitch) quieter. Depending on the gain, a bass filter can amplify or attenuate the bass part in a piece of audio. The amplification of the bass is usually referred to as *bass boost* and the attenuation as *bass cut*. Similarly, a treble filter can apply *treble boost* or *treble cut* to the high-pitch part of a piece of audio. In this section of the project, you will use the shelf filters you designed in the previous two sections to improve the perceptual quality of a very short piece of music, so that it sounds more balanced. This exercise is a highly simplified example of what recording engineers do in music recording studios.

(a) Four audio files, `bass`, `piano`, `mixture2`, and `music` are provided with this lab together with the sampling frequency `fs` used to create the audio files. all variables are contained in `Lab6.mat` which you should load into your MATLAB workspace. Listen to each audio signal using the MATLAB function `soundsc`. For example, to hear the `bass` file, use:

```
>> soundsc(bass,fs);
```

The files `bass` and `piano` contain around eight seconds of piano and bass loops synthesized using Apple Garageband, and the file `mixture2` contains a mixture of the two sounds. The file `music` is a tone sequence that will be used later in the lab.

(b) Comment on the perceptual sound quality of the sound mixture. Do you find it well-balanced? Does one of the sound sources dominate the other one? If one of the instruments is dominant, try to determine qualitatively, by listening to the piece, whether it is high-pitch or low-pitch. Discuss how you would use a shelf filter to improve the sound mixture quality. What type of shelf filter would you use?

(c) The spectrogram of a signal is a two-dimensional plot that shows how the frequency content of a signal changes over time. Use the MATLAB function `spectrogram` with the following suggested input values to generate a spectrograms of the piano and bass signals. For example, to plot the spectrogram of the `bass` waveform, use:

```
>> win = 512; noverlap = 256; nfft = 512;
>> spectrogram(bass, win, noverlap, nfft, fs, 'yaxis');
>> colorbar;
```

The command `colorbar` invokes the color scale on the side of the spectrogram as a reference for the strength of the frequency components.

(d) Read the approximate range of active frequency components of the two instruments from the spectrograms, and use this information and the results of your listening experiments in part (a) to determine the proper filter type, an appropriate gain, and the cut-off frequency to modify the frequency content of the mixture. The goal in modifying the mixture is to balance the sounds of the two instruments so that both of them can be heard clearly and so that neither one dominates the other. Use the `design_shelf` function to design your filter and then apply the filter to the mixture signal. Feel free to experiment with both filter types and to tweak filter parameters (gain and cut-off frequency) until you are satisfied with the sound quality.

(e) Listen to the output of the shelf filter. Describe how different it sounds from the input signal. Make spectrograms of the original mixture and the filtered version (use the same parameter values that you used in part (c)). Compare the two spectrograms while paying special attention to the color scale.

(f) The third sound file, `music.wave`, is an ascending tone sequence followed by a chord of the tones. Listen to this file using `soundsc`:

```
>> soundsc(music,fs);
```

and calculate a spectrogram of the file. Comment on what you see.

Design a shelf filter that enhances the third tone (the highest) and diminishes the two lower tones. Calculate a spectrogram of the filtered file you have created. Compare this with the spectrogram of the original `music` file. Provide a detailed description of your approach and discuss the overall effectiveness of the shelf filter that you have employed.

Lab #6
ECE 220: Fall 2016
Instructor Verification Sheet

Name: ____________________________ Date of Lab: ______________

For each verification, be prepared to explain your answer and respond to other related questions that the lab TAs or professors might ask. Turn this page in along with your lab report.