Let's Make it Sound Better
EECE 269 Project - Winter 2016 Term 2

March 2, 2016

1 Overview

To make your music sound better, your mobile phone or computer uses an audio equalizer. The equalizer tunes multiple frequency bands to improve the user's listening experience. These days the equalizers also use special sound enhancement techniques, such as the reverberation effect, where you can configure the ambiance, i.e. room, stadium, concert hall,etc. In this project, you will apply the concepts that you learned throughout this course, to implement selected sound enhancement techniques.

1.1 Requirements

You will need a computer with access to the Internet and MATLAB. You will also need loudspeakers or headphones to listen to the audio files.

1.2 Deliverables

You will submit this project as a hard copy report printed on letter-sized paper with a cover page that includes your name and UBC student number. The report will contain your labeled and ordered answers to the questions and your MATLAB codes. Print out of figures must be included where required.

In order to be graded, your report must be submitted before or during the last lecture of this term.

2 Tasks

For all the taks, use the file "audio_signal_N.wav" for input, where N is the remainder when you divide your student number by 3.

Task 1 In this task, it is required to load, play and process audio signals in MATLAB.

- Load "audio_signal_N.wav" into MATLAB using *wavread* function and assign to x.
- Play the audio signal x in MATLAB using the *sound* function.

Q0 What is the sampling frequency and the number of bits per sample of the loaded audio signal?

- Make a copy y of the signal x and use the MATLAB function *resample* to resample y at half the sampling frequency of the original signal x.
- Make a copy z of the signal x and use the MATLAB function *downsample* to resample z at half the sampling frequency of the original signal x.
- Using MATLAB's Fourier transform function *fft*, obtain the frequency spectra of x, y and z.

Q1 Note that the signals x, y and z are all real signals. What do you observe in the frequency spectra?

Q2 Explain why the frequency spectra of y and z are different.

P0 Plot the absolute value of the frequency spectra of y and z (the x-axis must display the frequencies).

Task 2 One method to add the reverberation effect is by treating the room/stadium as a filter, with the original audio signal as the input and the signal with the reverberation effect as the output.

In this task, it is required to create audio signals with reverberation effects in MATLAB.

- Load "audio_signal_N.wav" into MATLAB using *wavread* function and assign to x.

- Note that a filter is characterized by its impulse response. First load "room_impulse.wav" into MATLAB using *wavread* function and assign to h.

- The output of the filter is obtained by performing the convolution of the input x with the impulse response h. Perform this operation using MATLAB's *conv* function and assign the result to y.

- Convolution in time domain is multiplication in frequency domain. Write your own MATLAB function, $MyConv(x, h)$ to perform the convolution of x and h using the following approach: Obtain the frequency spectra of x and h using MATLAB's *fft* function, multiply them, compute the inverse Fourier transform of the result using MATLAB's *ifft* function and take its real part to get the final signal z. Note that x and h might be of different lengths and the FFT size should be at least equal to the length of the longer signal.

- Save y and z in MATLAB using the *audiowrite* function with appropriate filenames.

P1 Plot the absolute value of the frequency spectra of y and z (the x-axis must display the frequencies).

- Repeat the above procedure using "stadium_impulse.wav".

Q3 How would you approach this problem if the sampling frequency of x and h are different?

Task 3 Bass shelf and treble shelf filters are used to tune the bass and treble frequency range of audio signals, respectively.

The filter coefficients of a "bass-boost" filter are given by

$$\begin{aligned} b_0 &= \frac{(1+\sqrt{G_l}\cdot r\cdot C + G_l\cdot C^2)}{(1+r\cdot C+C^2)} \\ b_1 &= \frac{(2(G_l\cdot C^2-1))}{(1+r\cdot C+C^2)} \\ b_2 &= \frac{(1-\sqrt{G_l}\cdot r\cdot C + G_l\cdot C^2)}{(1+r\cdot C+C^2)} \\ a_0 &= 1 \\ a_1 &= \frac{(2(C^2-1))}{(1+r\cdot C+C^2)} \\ a_2 &= \frac{(1-r\cdot C+C^2)}{(1+r\cdot C+C^2)} \end{aligned} \tag{1}$$

where $C = \tan(\frac{\pi f_c}{f_s})$, with f_c denoting the cut-off frequency and f_s denoting the sampling frequency, G_l is the gain of the filter in the pass-band (in linear scale) and r is a constant. The filter coefficients of a "treble-boost" filter are given by

$$\begin{aligned} b_0 &= \frac{(G_l+\sqrt{G_l}\cdot r\cdot C + C^2)}{(1+r\cdot C+C^2)} \\ b_1 &= \frac{(2(C^2-G_l))}{(1+r\cdot C+C^2)} \\ b_2 &= \frac{(G_l-\sqrt{G_l}\cdot r\cdot C + C^2)}{(1+r\cdot C+C^2)} \end{aligned} \tag{2}$$

with the remaining parameters same as that of the "bass-boost" filter.

In this task, it will be required to implement filters to boost only bass, only treble and both bass and treble.

Q4 What is the order of the boosting filters specified above?

- Load "audio_signal_N.wav" into MATLAB using *audioread* function and assign to x and note the sampling frequency f_s.

- Use the MATLAB *filter* function to implement a "bass-boost" filter with input x, a cut-off frequency of 250 Hz, pass-band gain of 10 dB and $r = \sqrt{2}$ and store the output of the filter in y.

P2 Plot the frequency response of the "bass-boost" filter using the MATLAB *freqz* function.

- Use the MATLAB *filter* function to implement a "treble-boost" filter with input x, a cut-off frequency of 4 kHz, pass-band power gain of 10 dB and $r = \sqrt{2}$ and store the output of the filter in z.

P3 Plot the frequency response of the "treble-boost" filter using the MATLAB *freqz* function.

- Save y and z in MATLAB using the *audiowrite* function with appropriate filenames.

Q5 Identify and explain why one is a low-pass filter and the other is a high-pass filter.

- Using these two filters, obtain an enhanced audio signal s, which has both bass and treble boosted using the following two methods - a) by feeding the output of the "bass-boost" filter to the input of the "treble-boost" filter and b) by evaluating new filter co-efficients $(\bar{b}, \bar{a})$ using the values (b, a) of the two given filters.

P4 Plot the frequency response of the "bass-and-treble-boost" filter using the MATLAB *freqz* function.

- Save s in MATLAB using the *audiowrite* function with appropriate filename.

Q6 What are the values of $(\bar{b}, \bar{a})$? What type of filter is the "bass-and-treble-boost" filter?

Q7 Examine the effect of r on the frequency response of the "bass-boost" and "treble-boost" filters by checking for some values of r lesser than 1 and some greater than 1. Comment on your observations.

Hope you enjoyed listening to the audio enhancements that you just implemented!