CS3570 Introduction to Multimedia

<u>**Homework #2**</u>

Due: 11:59pm, 4/17/2018

Q1. Create your own FIR filters to filter audio signal (40%)

"**HW2_Mix.wav**" is a mix of 3 songs. Based on the **ideal impulse responses** given in <u>slide #72</u> and the **windowing functions** in <u>slide #75</u>, you need to design and apply different FIR filters in time domain to separate the **three audio signals** from the given audio file. You can refer to the algorithm on <u>slide #76</u>. Finally, the output audio signals are too simple so you should apply **one-fold echo** and **multiple-fold echo** (<u>slide #69</u>) to produce wonderful music.

1. Implement **3 different FIR filters** to separate the three audio signals with:
   - **Blackmann** window function (You have to pick the appropriate window size and cut-off frequency).
2. Implement **1-D convolution** on the input signal of the given audio with your filters.
3. Apply **one-fold echo** and **multiple-fold echo** on the audio signal that pass through the **low-pass filter**.
4. **Plot** the result.
   - The spectrum of the input signal
   - The spectrums of the output signals. **(before echo)**
   - The spectrums of the filters
   - The shapes of the filters (time domain)
5. **Store** the filtered audio files and name "**FilterName_[para1]_[para2].wav**".
   Store the echo audio files and name **"Echo_one.wav"** and **"Echo_multiple.wav"**
6. In the report
   - Discuss how you determine the filters.
   - How you implement the filter and convolutions to separate the mixed song and one/multiple fold echo.
   - Compare spectrum and shape of the filters.
   - Anything worth mentioning.

Q2. Audio dithering and noise shaping (60%)

There is an audio signal **"Tempest.wav"**. You are asked to apply **bit reduction** to reduce the signal from 16-bit to 8-bit. Due to the reduction, the audio will contain noise and sharp stair-step effect. To eliminate the noise, you need to apply **audio dithering**, **noise shaping** (slide #18), and use **low-pass filter** you finished in Q1 to filter out the high frequency components. Finally, since the volume is too small, you should apply **audio limiting** (slide #48) and **normalization** (slide #50) to get the louder audio signal. Follow the steps below and you can finish the homework easily.

1. **(Bit reduction)** Reduce the signal from 16-bit to 8-bit.
2. **(Audio dithering)** Add random noise (uniform distribution) into the signal
3. **(Noise shaping)** Apply the first-order feedback loop for noise shaping. Please choose the appropriate coefficient **c** yourself.
4. **(Low-pass filter)** Apply low-pass filter. Please choose the appropriate **cutoff frequency** yourself.
5. **(Audio limiting)** Apply hard clipping to the signal.
6. **(Normalization)** Apply normalization steps to the signal.
7. **Plot** the result
   - The shape and spectrum of input signals
   - The shape and spectrum of output signals
   - The shape and spectrum of the result of audio dithering and noise shaping
8. **Store** the audio file
   - The result of Bit reduction – **"Tempest_8bit.wav"**
   - The final output audio signal – **"Tempest_Recover.wav"**
9. In the report
   - How you implement the bit reduction, audio dithering, noise shaping, low-pass filter, audio limiting and normalization.
   - Discuss the effect of dithering and noise shaping according to the spectrums and shapes you plot
   - Anything worth mentioning.

Reminder

- Follow the instructions (hints) and spec in the sample codes.
- You cannot use Matlab built-in function "conv" in this homework.

- Plot the spectrum and shape in appropriate range for better visualization.
- Please save the report as “[YourID]_report.pdf”.
- Please provide a README file about how to execute your program.
- Please compress all the .m files, output audio files and report into a zip file and name it “HW2_[YourID].zip”.