Signals and Systems
Quiz #2

E3 upload, Due 23:59 Sunday, June 11

Delivery list

(a) All MATLAB codes
(b) A file containing plots and your answers (PDF file format is preferred).
(c) Output audio files with the correct file name (see below).

Put all files in a zipped single file with the name of your student ID and UPLOAD to e3.

Grading: For some questions, there are no standard answers. Thus our grading will be

(1) Codes are runnable and the output sounds make sense 50 %
(2) Handwritten (PDF file) answers 50%
(3) Bonus (If you provide more detailed analyses or extend your results) 10%

MATLAB assignments to understand how to process an audio signal (e.g. music) using digital signal processing. For the following questions, please refer to the corresponding MATLAB's exemplary codes. What you need to do is to COMPLETE the codes and GENERATE all results (either figures or audio outputs).

(CAUTION: MAKE SURE the volume is moderately low to avoid hurting your hearing. 請注意測試輸出檔案時, 小心音量控制, 避免傷到聽力, 此音訊最大音量振幅為1, 人的靈敏度可以聽到0.01-0.001的聲音，請小心測試).

(a) Read the music (2-channel stereo music) using MATLAB' audioread. For simplicity, we will process **ONE** channel only (i.e., the first channel. The first row in the raw data. This is MONO tone).
   (i) PLOT the time waveform and its frequency-domain representation using FFT. Please also indicate the **correct** time and frequency index based on the sampling rate and data length. (Paste your plots here)
   (ii) Write this MONO music to a new file and save it as .OGG format using MATLAB's audiowrite. (Output a file named as org+your student ID.ogg)

(b) Noise the music (To test your hearing sensitivity).
   (i) Add a Gaussian white noise (using 'randn', **See HW#1**) to the original signal such that the signal-to-noise ratio is 30 dB. The maximum signal amplitude is assumed to be 1. PLOT and compare the frequency-domain representation with and without adding noise. (Paste your plots here)
   (ii) Same as (i). What do you hear? (可以中英文白話敘述)
   (iii) Output a file named as noise30dB+your student ID.ogg
   (iv) How high the SNR is so that you hear the same quality as the original music? (這部分只要回答SNR多少妳/你就感覺不到noise 的影響, 請實測看看並注意音量控制小心影響聽力)

(c) Watermark the music (To test your hearing frequency band or generate inaudible signals). Add a single tone (i.e., a single frequency, cosine signal) to the original data so that you **CANNOT** hear this single tone. That is, you hear exactly the same as the original music.

(i) How to add this single tone and what is the frequency of it? PLOT and compare the frequency-domain representation with and without adding noise. (Paste your plots here). Note that aliasing might be present when the frequency is higher than half of the sampling rate.

(ii) Output a file named as inaudible+your student ID.ogg

(d) Filter the music. Apply the low-pass filter and high-pass filter to listen what is the difference

(i) Use MATLAB's fir1 to generate a low-pass filter with a 64-order FIR type and a variable cut-off frequency (You need to **specify** what the cut-off frequency is). Plot and compare the frequency presentation before and after filtering (Paste your result here). Output the audio file called lpf+your student ID.ogg

(Note: Please carefully read the MATLAB fir1 about the meanings of input parameters)

(ii) Use the same function fir1 to generate a high-pass filter with a 64-order FIR type and a variable cut-off frequency. You need to **specify** what the cut-off frequency is. Plot (Paste your result here) and output the audio file called hpf+your student ID.ogg.

(v) State what is the difference of the filtered music between low-pass and high-pass filtering? (可以中英文白話敘述)

(e) Downsample the music. The original music has the sampling rate of 44.1kHz.

(i) We are trying to make the sampling rate "HALF" (i.e., 22.05kHz). What should it be done without aliasing? (hint: apply an anti-aliasing filtering before downsampling. A factor of 2 means sampling every other sample.)
Complete the code and output the file called ds2+your student ID.ogg.

(f) Compress the music. Use the above filtering or downsampling or **other skills** you can "compress" the music as much as possible in terms of file size without losing the audio quality too much. Complete the code and output the file called compress+your student ID.ogg. Specify your compression ratio by comparing the output file size with the original .ogg file size (i.e., the file generated in (a)(ii))