Name:
Student ID:
Section:

ECE 340
Discrete Time Signals and Systems
Lab. 5
Filtering Multimedia Signals Using an FIR Filter

Please read the following instructions carefully

1) This lab is to be done individually.
2) At the end of the Lab, wait for the lab instructors to come over and monitor your results.
3) Once you have been given a mark for the lab, please leave the lab quietly.
4) Remember to hand-in this sheet of paper to the lab instructor before you leave.

Outline of the Lab

In Lab-4, you used a noisy audio signal and a noisy image signal, and did the power spectrum analysis. In this Lab, you will perform digital filtering on these signals and analyze the effect of filtering. Note that a digital filter is a discrete-time system that transforms the frequency spectrum of a signal in a specified way. In digital signal and image processing, the digital filters are widely used for various tasks such as reshaping of the frequency spectrum and noise removal.

1. (a) Using MATLAB `fir1` function, design a 513-tap lowpass FIR filter with a cut-off frequency of 2500 Hz. Choose an appropriate truncation window such that the stopband ripples of the frequency response do not exceed -50dB. Assume a sampling frequency of 22050 Hz. A sample code is given below. Initialize appropriate filter parameters.

```
wc=fc/(Fs/2);
% fc: The cut-off frequency of the filter
% Fs: Sampling frequency of the audio signal
window = hamming(513);
% Truncation window function, using Hamming window.
% Other truncation window types may also be applicable. Please use
% Matlab help to find more applicable truncation windows.
filter_coeff=fir1(513-1,wc, window);
% filter_coeff: Coefficients of the FIR filter
```

   (b) Plot the frequency response of the filter. You may use the MATLAB `freqz` function to calculate the frequency response. A sample code is given below. Initialize appropriate filter parameters.

```
freqz(filter_coeff,1); %The frequency response of the filter
```

   (c) Read the audio signal `love_mono22.wav` used in Lab-4 into MATLAB workspace.

   (d) Pass the audio signal through the filter and calculate the output signal. A sample code is given below.

```
x_filtered=filter(filter_coeff,1,x);
%x_filtered: The filtered signal
```

   (e) Calculate and compare the spectrum (using `psd` or `pwelch` function) of the input and output signals and compare the two.

   (f) Play the output using MATLAB sound function [You can also save it in .wav format, and play it with system media player].

   (g) Listen to the original and filtered signals and comment on their difference.

2. (a) Using MATLAB `fir1` function, design a 513-tap highpass FIR filter with a cut-off frequency of 5000 Hz. Choose an appropriate truncation window such that the stopband ripples of the frequency response do not exceed -50dB. Assume a sampling frequency of 22050 Hz.

   Repeat steps (b)-(g) in Q1.

3. (a) The audio signal is corrupted by a noise. From the frequency spectrum of the noisy audio signal, determine the frequency range in which most of the noise's energy located (in KHz). Based on your observations, use MATLAB `fir1` function to design a 513-tap FIR filter to remove the noise in the audio signal. Choose an appropriate truncation window such that the stopband ripples of the frequency response do not exceed -50dB. What type of filter (lowpass, highpass, bandpass, bandstop) is it?

   Repeat steps (b)-(g) in Q1.

4. In Lab-4, you used a noisy image `ayantika.tif`. The spectrum analysis in Lab-4 showed that the rectangular grid noise is due to the 4 peaks in the mid-frequency region of the power spectrum. It is possible to eliminate the noise by designing an appropriate bandstop filter which will just eliminate the noise frequencies. In this question, we will use another approach where we will use the following lowpass filter

   `filter_coeff` = [1 2 3 2 1; 2 3 4 3 2; 3 4 5 4 3; 2 3 4 3 2; 1 2 3 2 1]/65 ;

   whose stopband includes the noise frequencies.

   (a) Download the Lab5Q4.m MATLAB function, and run it.
   (b) Go through the various plots and images and try to understand what is going on.
   (c) Comments on the plots and images. Explain, why the filtered image looks somewhat a very smooth version (the edges are not very sharp) of the original image.

Some tips:

1. The sound function can be used as follows:
   `sound(y, fs, nbits);`
   Here `y` is the vector of input signal, `fs` is the sampling frequency and `nbits` is number of bits per sample.
2. When saving a vector as an audio file using `audiowrite`, please ensure that all values in this vector are in the range of [-1, 1]. Otherwise, a warning of data clipping will raise.