EE 220 Signals and Systems
Assignment 2: MATLAB learning

Instructions: This is a MATLAB exercise for your learning. It doesn't require assistance from your fellow students. Enough hints are already given. The only thing you need to figure out is how you are going to install MATLAB and then you should be able to do it yourself. Make an individual report and submit it in the tutorial class on 18th October. Some mat files are to be uploaded on the Teams page as well. This assignment will be graded for 6% marks.

Question 1
Objective: To perform convolution using MATLAB.

Before Getting Started
This assignment deals primarily with manipulating sound waveforms. As a result, having a pair of headphones or PC speakers will be helpful (but not necessary) for completing this assignment. You will be able to make use of the PC's sound card in MATLAB using either the function `sound` or the function `soundsc`. Please use the `help` command to find out more about these functions.

Loading the Data
You need to obtain the speech waveform from the Teams website called "echo.mat." You should load this data file into MATLAB using the `load` command. Therefore, make sure that you have saved echo.mat to the current working directory of MATLAB and load the data by typing the following at the command prompt.

```
load echo.mat
```

This will load two variables into the workspace, which are
`xx` – the speech waveform
`fs` – the sampling frequency of the speech waveform

Listening to the Speech Waveform
Assuming you have headphones, speakers, or access to the PC's internal speakers, you may listen to this speech waveform using the `sound` or `soundsc` functions. The general form is

```
soundsc(waveform, samplingfreq)
```

and the sound will be played through the computer's sound card. (The only difference between `sound` and `soundsc` is that `soundsc` scales the waveform to maximize its volume without clipping the waveform. I would typically recommend using `soundsc`.) If you are having problems hearing the speech signal, make sure that you have the volume turned up on the PC's Master Volume.

Plotting the Speech Waveform
Plot the speech waveform, as a function of time (*not sample value*). Convert the sample values to their time values using the sampling frequency. Make sure you label all plots. (You may use the `plot` function for this, and not the `stem` function, since you are converting the x-axis to be time instead of sample values).

Create a System Impulse Response

You must create a discrete-time system impulse response that creates an "echo" of the original speech waveform. To do so, the system must pass the present value of the input unattenuated (i.e. gain of 1) and also a delayed version of the input signal that has been attenuated by a factor of 4 (i.e. the "echo" should ¼ the size of the original signal and should be delayed by one second). Another way of stating this is that the resulting output of the system will include the present value of the signal (unattenuated) and a delayed version of the signal (attenuated). Pretend, for example, that you are on the top of a mountain facing a second mountain (the "system" is the mountain range with the output being whatever your ears hear). When you yell, you will hear your voice instantaneously as it comes out of your mouth, and you will also hear the echo one second later. In your report, include the line(s) of code used to create the impulse response in Matlab.

HINT – How many samples are in one second of time?
HINT – Since you first need to create the impulse response, think about what the input would be for an *impulse response*.

Convolution

Use discrete-time convolution in MATLAB to find the output of the "echo" system when using the input contained in echo.mat, and plot this output. Again, this system outputs the present value of the signal and also echo of the signal one second later (that has been attenuated by a factor of 4). Plot this signal as a function of time (again, convert the sample values into time). Also, you should listen to the output to verify that it does, indeed, echo the original speech waveform. In your report, include the line(s) of code used to perform convolution of your impulse response and the input signal.

What to Turn In

Turn in the two resulting plots. You must also turn in the Matlab code that was used to generate your impulse response and how you found the output.

Helpful Hints
- Useful commands
  - `ones`
  - `zeros`
- Remember, you can concatenate multiple signals
  - e.g. if you have `x1`, `x2`, and `x3`
  - `x4 = [x1 x2 x3];`
  - This could be helpful in creating your impulse response, but do not use concatenation to create the output signal.
- If you did not use the `conv` function, then you have done this project incorrectly.
- You can get help on any function by typing `help function_name`
- Do not forget axis labels and titles on your plots
- Do not forget the cover sheet as described on the MATLAB assignments webpage

Question 2 – Guitar Effects

The object of this question is to take this assignment one step further and create a guitar effect known as digital delay. You may download a guitar sample from the class website.

You must create a <u>function</u> that can accept four arguments.

- Input signal
- Time delay (in seconds)
- Attenuation factor of the echo
- Sampling frequency of the input signal

For example, this function may be used in the following manner.

```
[output] = digitaldelay(input, timedelay, echoattenuation, samplingfreq);
```

This function must pass the present value of the input signal and also provide an echo that is attenuated by some factor. Typically, this guitar effect is used to provide a short delay (~0.1sec) that gives the guitar a "fuller" sound; this often makes it sound like multiple guitars are playing at the same time. This function must also be able to vary the amount of time delay and the amount of attenuation of the echo.

What to Turn In
Submit the function used to generate the effect and the code used to call the function. This code must be turned in via paper form (with your report). Play around with the values of the time delay and attenuation to find what you consider the most aesthetically pleasing sound. Save the results of your output signal (after passing through your function); save this in a .mat file. Name your .mat file as *FullName_rollnumber*.mat. You must upload the .mat file you generated from running your function on the Teams page. Use the `save` command to create the .mat file. Please name your output variable `yy`.

I will then take the .mat file and listen to it using

```
soundsc(yy,44100);
```

Make sure you include the call to the function that was actually used to generate the output (to verify the time delay, attenuation, etc.). This call to the function should contain the actual numerical values of the parameters (time delay, echoattenuation, samplingfreq) that you used to generate your final version of the output signal.

I will test all sound files using the following commands, so please make sure you follow the directions closely (e.g. make sure you save your output as variable `yy`).

```
load lastname.mat
soundsc(yy,44100);
```

I will run these two lines of code, exactly as shown (with your "*FullName_rollnumber*" substituted in), so please make sure that your .mat file will work with these two lines of code.