Full Name: ____________________

EEL 3135 (Spring 2021) – Lab #04 Due: 11:59 PM ET, Feb. 16 - Feb. 22 (On Lab Day)

Submission Notes

- Your laboratory solutions should be submitted on Canvas as a single published MATLAB PDF.
- Use the provided skeleton code as the basis for your solutions (easier for you and the graders).

Question #1: *(Convolution)*

Download `EEL3135_lab04_comment.m` from Canvas, replace each of the corresponding comments with the corresponding descriptions. This is designed to show you how to work with audio and images using convolution in MATLAB.

Note: You should run the code to help you understand how it works and help you write your comments. You will use elements of this MATLAB code in the rest of the lab assignment.

Question #2: *(Echo and Tremolo)*

This question will implement both an echo effect and a tremolo effect on a song.

(a) Create a function `y = echo(x, s, A)` that outputs the following signal:

$$y[n] = x[n] + Ax[n-s] ,$$

where `A` is the echo amplitude and `s` is the echo delay. You may want to use the `shift` function in Question #1. Include the `echo` function at the end of the skeleton file.

(b) Apply `echo` to your input `x` in the skeleton code to get output `xr`. Set the delay to $s = 10000$ samples and amplitude to $A = 0.9$. Use `soundsc` to play the output. **Have this audio checked off by a TA.**

(c) **Answer in your comments:** If our sampling frequency is 44100 samples per second, and we are delaying by 10000 samples, how long is the delay on the signal in seconds?

(d) Create a function `y = tremolo(x, m, A)` that outputs the following signal:

$$y[n] = x[n] + A\cos(2\pi mn)x[n] ,$$

where `A` is the tremolo amplitude and `m` is the tremolo modulation frequency (in normalized angular frequency). Include the `tremolo` function at the end of the skeleton file.

(e) Apply `tremolo` to your input `x` in the skeleton code to get output `xt`. Set the amplitude to `A = 0.5` and modulation frequency to `m = 20 / fs`, where `fs` is the cyclic sampling frequency. Use `soundsc` to play the output. **Have this audio checked off by a TA.**

(f) **Answer in your comments:** How does the tremolo affect the audio and why?

(g) Again apply `tremolo` to your input `x`, but now with the amplitude `A = 1` and modulation frequency `m = 1/(length(x))`. Use `soundsc` to play the output. **Answer in your comments:** Why does the signal strength decrease half-way through and then get loud?

(h) A system is time-varying if the system $\mathcal{T}\{\cdot\}$ satisfies

$$\mathcal{T}\{x[n-N]\} \neq y[n-N] \qquad \text{such that} \qquad \mathcal{T}\{x[n]\} = y[n]$$

Use `shift` function in Question #1 to shift outputs from (b) and (g) by $N = 220499$ samples (half the signal length). Use `soundsc` to play these outputs. Then shift your input `x` by $N = 220499$ samples to get `xs`. Input `xs` into `echo` and `tremolo` with the parameters from (b) and (g). Use `soundsc` to play the new outputs. **Have this audio checked off by a TA.**

(i) **Answer in your comments:** Based on your results from (h), is either system time-varying? How do you know?

Question #3: *(Image Filtering)*

Spatial filtering is often used to modify images. We can regard the image as an input signal and the spatial filter is our system. The output of this system is obtained by doing convolution between the input image $g[x, y]$ and the filter impulse response $w[u, v]$. The output pixel $f[x, y]$ with 3×3 filter can be obtained by the 2-dimensional convolution

$$f[x, y] = \sum_{s=-1}^{1} \sum_{v=-1}^{1} w[u, v] g[x-u, y-v].$$

For this question, add all code into skeleton `ee13135_lab04_skeleton.m` from Canvas. Include all code (and functions) in this one file so that everything is published to a single PDF.

(a) Consider the filter impulse response (also known as a kernel) below. Apply this filter to the image `img` in the skeleton code. Use `image` and `subplot` to plot the image before and after applying the filter. Use Question #1 as a guide.

$w[-1,-1]$	$w[-1,0]$	$w[-1,1]$		1/9	1/9	1/9
$w[0,-1]$	$w[0,0]$	$w[0,1]$	=	1/9	1/9	1/9
$w[1,-1]$	$w[1,0]$	$w[1,1]$		1/9	1/9	1/9

(b) **Answer in your comments:** Does this filter blur, sharpen, or extract edges in the image? How / why do you know that from the impulse response / kernel?

(c) Consider the filter impulse response / kernel below. Apply this filter to the image `img`. Use `image` and `subplot` to plot the image before and after applying the filter.

$w[-1,-1]$	$w[-1,0]$	$w[-1,1]$		1	1	1
$w[0,-1]$	$w[0,0]$	$w[0,1]$	=	1	-8	1
$w[1,-1]$	$w[1,0]$	$w[1,1]$		1	1	1

(d) **Answer in your comments:** Does this filter blur, sharpen, or extract edges in the image? How / why do you know that from the impulse response / kernel?

(e) Convolve the original image with a 3×3 2-dimensional impulse function (i.e., w[0,0]=1 and all else is zero). Subtract the output image in (c) from that result. This process is known as *unsharp masking*. Use `image` and `subplot` to plot the image before (the original image) and after applying unsharp masking.

(f) **Answer in your comments:** Does this filter blur, sharpen, or extract edges in the image? How is this result different than the other two filter results?