Full Name: ______________________________

EEL 4750 / EEE 5502 (Fall 2021) – HW #1 **Due: 4:00 PM ET, Aug. 30, 2021**

Concept Questions 1

Question #1: Complete the Canvas questions here: `https://ufl.instructure.com/courses/437179/assignments/4805163/`

Question #2: *(Slack)* Join the EEL 4750 / EEE 5502 Slack Group. Use the following URL to join the group:

`https://join.slack.com/t/uf-eee5502-fa2021/signup`.

Use your first and last name as the display name. The Slack Group will be used to ask / answer questions. It has been used extensively and appreciated by prior classes.

Theory Questions 1

Question #1: *(Signals Properties)* This question develops theoretical properties in this course.

(a) Show that any signal can be decomposed into the sum of a causal and anti-causal signal:

$$x[n] = x_c[n] + x_a[n]$$

(b) Show that any signal can be decomposed into the sum of an even and odd signal:

$$x[n] = x_e[n] + x_o[n]$$

(c) Assume $x[n] = ax_e[n] + bx_o[n]$ for scalar values a, b. Also assume the energy of $x_e[n]$ is 1 and the energy of $x_o[n]$ is 1. Show that

$$a = \sum_{n=-\infty}^{\infty} x[n]x_e[n] \quad , \quad b = \sum_{n=-\infty}^{\infty} x[n]x_o[n]$$

(d) Briefly discuss how could this property be valuable in computational applications.

Question #2: *(Systems)* This question develops theoretical properties used in this course.

(a) Consider two discrete-time systems,

$$y[n] = \mathcal{H}\{x[n]\} = ax[n]$$
$$y[n] = \mathcal{G}\{x[n]\} = x[n-N]$$

Determine the output of the cascaded systems $y[n] = \mathcal{G}\{\mathcal{H}\{x[n]\}\}$.

(b) Consider three continuous-time systems,

$$y(t) = \mathcal{H}\{x(t)\} = x(\alpha t)$$
$$y(t) = \mathcal{G}\{x(t)\} = x(t-\tau)$$
$$y(t) = \mathcal{T}\{x(t)\} = x(-t)$$

Determine the output of the cascaded systems $y(t) = \mathcal{G}\{\mathcal{T}\{\mathcal{H}\{x(t)\}\}\}$.

Implementation Problems Information

Additional information for starting the assignment.

A Note on MATLAB: Implementation assignments will be done in a development platform from Mathworks called MATLAB. The labs are written for MATLAB's stable version R2019b, although most versions within recent years will also work. MATLAB can also be accessed for free through UF's `https://view.ece.ufl.edu` and `https://apps.ufl.edu` remote login into UF computers. However, we recommend that you download the standalone newest version to use in case access to the internet is not available.

Installing MATLAB: Mathworks offers student version of MATLAB that can be purchased at their website. There are different toolboxes that contains application specific optimized APIs for development. You should not need any toolboxes for this class.

Writing Scripts in the Editor Window: We will be writing and submitting MATLAB scripts during this class. Start a new script by clicking on "New Script" or "New Live Script" in the upper-left in the "HOME" toolbar. This should produce an editor window. You can then sequentially run the script by clicking the "Run" button in the "EDITOR" toolbar.

Publishing MATLAB Live Scripts: This semester, when a document of your code is requested, MATLAB Live Scripts will be use. Live Scripts can be exported as a PDF using the "Save" drop down menu on the top right of the editor window. When using Live Scripts, images and text output through the code will also appear in the exported PDF. You can separate sections using `%%` and display answers to questions that are asked in the homework using commends such as `disp()` and `plot()`. Always export to PDF for submitted assignments.

An explanation on how to use the tool can be found at:

`https://www.mathworks.com/help/matlab/matlab_prog/publishing-matlab-code.html?fbclid=IwAR3REw_0GAC1lzx7QK9Z_UkL3O-9Rki-QS85YtL8En6eK3Bvbg2gFjDN4WY`.

MATLAB Tutorials: If you have little experience with MATLAB, we suggest looking at the tutorials on the course website. For MATLAB syntax basics, please see one of the many online tutorials, such as Mahdi Farahikia's "Introduction to MATLAB Programming for Engineers and Scientists" on YouTube. Please feel free to share useful resources on the course Slack page.

Implementation Questions 1

For these questions, download `2021_eee5502_code01_prob.zip` from the course website. This file includes skeleton code `code1.mlx` (you may convert it to a .m file if you want). Do not change the name of this file. Follow the directions below to submit your completed `code1.mlx` or `code1.m` file.

Also included in the zip file is `code01_grader.p`. With `code01_grader.p` and your solution in the same folder, type `code01_grader` into the command line to check your solutions.

Question #1: *(Signals)* In this question, you will practice constructing discrete signals. Each signal will have values in the range `n=-10:10` and be represented by a row vector of length 21:

(a) $x_1[n] = \delta[n-5]$

(b) $x_2[n] = (0.9)^{n-3}\, u[n-3]$

(c) $x_3[n] = 2\cos(0.1\pi n)\,(u[n+3] - u[n-3])$

Question #2: *(Systems)* In this question, we will build some basic systems in the form of functions (some which we will likely use again). Design the functions to accept input signals of arbitrary length and value. Function parameters are constrained as noted below. Input and output signals lengths should be equal. Assume all values outside of the range of the input signal are zero. Included in the provided skeleton code are signals that you will input into these systems.

(a) Delay (or advance) system, `y = delay(x,M)`[1]:

$$y[n] = x[n-M]$$

(b) A distortion system, `y = distort(x,a)`:

$$y[n] = \mathrm{sign}(x[n])\left(1 - e^{-a|x[n]|}\right)$$

(c) Reverberation system, `y = reverb(x,M,A)`:

$$y[n] = Ay[n-M] + x[n]$$

Question #3: Find a sound bite of your choice and apply distortion and reverberation on it, with parameters of your choice. Use the `audiowrite` function to save the result as a `.wav` file and submit with your code. You do not have to submit additional code for this question.

[1] Do not use convolutions for any of these functions yet (this will be helpful later in the course). Instead, use a for-loop to cycle thru each sample n and an if-then statement to not assign values outside of n's range.