**Full Name:** ______________________________

**EEL 4750 / EEE 5502 (Fall 2021) – HW #4** **Due: 4:00 PM ET, Sept. 20, 2021**

## Concept Questions 4

**Question #1:** Complete the Canvas questions here: `https://ufl.instructure.com/courses/437179/assignments/4812577`

## Theory Questions 4

**Question #1:** Use the definitions of the discrete-time Fourier transform (DTFT) / inverse DTFT to answer the questions below. As a reminder, the DTFT / inverse DTFT is defined by

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \qquad , \qquad x[n] = \frac{1}{2\pi}\int_{2\pi} X(\omega)e^{j\omega n}\, d\omega$$

(a) Show that the DTFT of a signal is always periodic with a period of $2\pi$.

(b) Show that if $x[n]$ is real, then its DTFT $X(\omega)$ has conjugate symmetry:

$$X(\omega) = X^*(-\omega)$$

(c) Show that if $x[n]$ is real (the complex case is not much more difficult), then

$$X_e(\omega) = \sum_{n=-\infty}^{\infty} x_e[n]\cos(\omega n) \qquad , \qquad X_o(\omega) = -j\sum_{n=-\infty}^{\infty} x_o[n]\sin(\omega n)$$

where $x_e[n]$ and $x_o[n]$ are the even and odd parts of $x[n]$, respectively. [1]

**Question #2:** Rectangular functions are used throughout signal processing and other fields (e.g., an aperture of a camera in two-dimensions can be represented by a rectangle and its effect on images is a convolution). Hence, the CTFT or DTFT of the rectangular function is also commonly used. In this problem, let's derive the DTFT of a rectangular function. That is, show that following DTFT pair is true:

$$x[n] = u[n+N] - u[n-N-1]$$
$$X(\omega) = \frac{\sin(\omega(N+1/2))}{\sin(\omega/2)}$$

You may use the DTFT tables to show this, except do not use the row of the table that explicitly gives this relationship. [2]

[1] Recall from Homework #1 that $X(\omega) = X_e(\omega) + X_o(\omega)$ [This proof is similar to another question in HW #1]

[2] Hint: You may want to use something similar to $1 - e^{-j\omega} = e^{-j\omega/2}(e^{+j\omega/2} - e^{-j\omega/2})$

## Implementation Questions 4

**Question #1:** We cannot numerically compute the continuous-time Fourier transform (CTFT) or the discrete-time Fourier transform (DTFT) perfectly because we cannot represent continuous-time signals on the computer. However, we can approximate these transforms.

(a) Write a function `X = DTFT(x,n,w)` that computes the DTFT $X(\omega)$ (`X`) at desired angular frequencies `w` from $x[n]$ (`x`). You can accomplish this with the standard expression:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

(b) Write a function `X = CTFT(x,t,w)` that computes the CTFT $X(\Omega)$ at desired angular frequencies `w` from $x(t)$ (`x`) at times `t`. You can approximate the CTFT as:

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt \approx \Delta T \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega(n\Delta T)} \,,$$

where $\Delta T$ is the difference between each consecutive value in the `t` vector.

**Question #2:** Use the CTFT and DTFT functions from the previous problem to compute and plot the Fourier transforms of the following signals for the $n$, $t$, and $\omega$ be defined in the MATLAB template.

(a) $x(t) = u(t+3) - u(t-3)$
$x[n] = u[n+3] - u[n-4]$

(b) $x(t) = u(t) - u(t-6)$
$x[n] = u[n] - u[n-7]$

(c) $x(t) = (u(t+3) - u(t-3))\cos((\pi/4)t)$
$x[n] = (u[n+3] - u[n-4])\cos((\pi/4)n)$

(d) What is the key difference between the CTFT and DTFT results? Leave a comment in your code with your observation.

**Question #3:**

(a) Included with the MATLAB template is a `cosine.wav` file with a cosine in it. Our template shows you how to load the file and get the time values (since we have not learned about sampling yet). Use your CTFT function to identify the cosine's positive **cyclic frequency** `f` ($f = \Omega/(2\pi)$) (Note: it will be in a different frequency range as Question 2.)

(b) We can now transcribe basic music. Included with the MATLAB template is a `rolemusic.wav` file with an excerpt of music by Rolemusic (`https://freemusicarchive.org/music/Rolemusic`) from the Free Music Archive. The excerpt is from "The Will."

Our template shows you how to load the file and get the time values (since we have not learned about sampling yet). Use your CTFT function and custom code to identify the positive **cyclic frequency** `f` and **amplitude** `A` of each dominant note (`A` can be extracted from the time or frequency domains). There are 50 notes in the excerpt, equally spaced and all of equal length. Included with the template is a synthesizer function that accepts `f` and `A` that you can use to check your results by listening.