Full Name: ______________________________

EEL 4750 / EEE 5502 (Fall 2018) – Code #04 **Due Date:** **Oct. 4, 2018**

Question #1: *(1 pts)* How many hours did you spend on this homework?

Question #2: *(8 pts)* *Orthogonality and the Discrete Fourier Transform*
Consider the definition of the inner product for two complex signal $x[n]$ and $y[n]$

$$c = \sum_{n=-\infty}^{\infty} x[n](y[n])^*$$

where $(\cdot)^*$ represents a complex conjugate operation. Now also consider the definition for two orthogonal complex signals $x[n]$ and $y[n]$. That is $x[n]$ and $y[n]$ are orthogonal if

$$\sum_{n=-\infty}^{\infty} x[n](y[n])^* = 0$$

where $(\cdot)^*$ represents a complex conjugate operation. Let's explore properties of orthogonality with respect to the discrete Fourier transform.

Note: Some solutions may need the generalized geometric series:

$$\sum_{n=0}^{N-1} r^n = \frac{1-r^N}{1-r}$$

(a) Let $x[n]$ and $y[n]$ be defined by $x[n] = e^{-j\omega_x n}$ and $y[n] = e^{-j\omega_y n}$ for $\omega_x = \omega_y$. Compute the inner product of $x[n]$ and $y[n]$. Are these vectors orthogonal?

(b) Let $x[n]$ and $y[n]$ be defined by $x[n] = e^{-j\omega_x n}$ and $y[n] = e^{-j\omega_y n}$ for $\omega_x \neq \omega_y$. Compute the inner product of $x[n]$ and $y[n]$. Are these vectors orthogonal?

(c) Let $x[n]$ and $y[n]$ be defined by $x[n] = e^{-j\omega_x n}$ and $y[n] = e^{-j\omega_y n}$ for $\omega_x \neq \omega_y$. Compute the convolution $x[n] * (y[n])^*$. How is this related to orthogonality?

(d) Let $x[n] = e^{-j\omega_x n}(u[n] - u[n-N])$ and $y[n] = e^{-j\omega_y n}(u[n] - u[n-N])$ for $\omega_x \neq \omega_y$. Show that if $\omega_x = (2\pi/N)k_x$ and $\omega_y = (2\pi/N)k_y$ (where k_x and k_y are integers), then $x[n]$ and $y[n]$ are orthogonal.

(e) Let $x[n] = e^{-j\omega_x n}(u[n] - u[n-N])$ and $y[n] = e^{-j\omega_y n}(u[n] - u[n-N])$ for $\omega_x \neq \omega_y$. Show that if $\omega_x = (2\pi/K)k_x$ and $\omega_y = (2\pi/K)k_y$ (where k_x and k_y are integers and $K \neq N$ is an integer), then $x[n]$ and $y[n]$ are *not* orthogonal.

(f) Consider the definition of the discrete Fourier transform (DFT)

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} .$$

Use the results above to compute $X[k]$ for $x[n] = \cos(\frac{2\pi}{N}k_x n)$.

Question #3: *(6 pts)* *The Undercomplete and Overcomplete Discrete Fourier Transforms*

For each of these questions, create a $N = 100$ length vector corresponding to $x[n] = \cos((\pi/2)n)$.

(a) Compute the discrete Fourier Transform (DFT) of $x[n]$:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} .$$

Let the horizontal axis correspond to $\frac{2\pi}{N}k$ for $0 \leq k \leq N-1$. Do this with a for-loop rather than the FFT (since you cannot use the FFT on the next two questions).

Submit the .m file and a plot the magnitude of $X[k]$. Let the horizontal axis correspond to $\frac{2\pi}{N}k$ for $0 \leq k \leq N-1$. **Remember to label all of your axes**.

(b) Note that we do not need to compute every frequency. Now compute an undercomplete DFT (i.e., we cannot represent any time-domain signal with the undercomplete basis) for $K = 10$:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{K}kn}$$

Submit the .m file and a plot the magnitude of $X[k]$. Let the horizontal axis correspond to $\frac{2\pi}{N}k$ for $0 \leq k \leq N-1$. **Remember to label all of your axes**.

(c) We can also compute additional frequencies. Now compute an overcomplete DFT (i.e., there are now many frequency combinations that can make a time-domain signal) for $K = 1000$:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{K}kn}$$

Submit the .m file and a plot the magnitude of $X[k]$. Let the horizontal axis correspond to $\frac{2\pi}{N}k$ for $0 \leq k \leq N-1$. **Remember to label all of your axes**. Also answer: how does this change your result from (b) and why. (*Hint*: Look at Question #2.)

Question #4: *(6 pts)* *Filtering Audio Data*

Load the audio .mp4 file rudenko_01.mp4 into MATLAB using

```
[x, Fs] = audioread(['rudenko_01.mp4']);
```

Listen to the audio recording using

```
sound(x, Fs)
```

Note that you can always stop the sound play by typing the commander `clear sound`.

Just as in coding assignment #3, create a vector corresponding to the impulse response of a 10000-point running average filter

$$h[n] = \frac{1}{M}\sum_{k=0}^{M-1}\delta[n-k] \ .$$

Let `M = 10000`. Use `conv` to filter $x[n]$ and get an output $y[n]$.

Submit a plot the magnitude of the DFT of $x[n]$, $h[n]$, and $y[n]$. For computational and memory efficiency, use the `fft` function to compute the DFT. In your plots, **remember to label all of your axes**.

Also, listen to the new audio with

```
sound(y, Fs)
```

You may need to amplify the signal by about 100 times before playing the audio. (note that the maximum volume for ‘sound‘ is 1)

Answer the questions: What is the relationship between $x[n]$, $h[n]$, and $y[n]$ in the frequency domain? How did the convolution change the signal? Does $h[n]$ act similar to a low-pass filter, bandpass filter, or high-pass filter?