Full Name: ____________________

EEL 4750 / EEE 5502 (Fall 2019) – Code #04 **Due Date:** **Oct. 21, 2019**

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

Question #2: *(7 pts)* *The Discrete Fourier Transform over Time*

The DFT or DTFT alone is not particularly useful for describing many complex signals. For example, music contains many frequencies that all occur at different times. The DFT / DTFT of an entire piece of music shows all of those frequencies. As an example, consider the signal

$$x[n] = \cos\left(\frac{2\pi n^2}{10000}\right) .$$

This is known as a linear frequency-modulated "chirp" signal whose instantaneous frequency $\omega[n]$ changes linearly with time. Chirp signals are commonly used in RADAR processing to extract time-varying time-shifts (distance from a target) and frequency/doppler-shifts (velocity of a target). In this context, the instantaneous frequency $\omega[n]$ is defined by

$$x[n] = \cos\left(\left(\omega[n]/2\right) n\right) .$$

More specifically, the instantaneous frequency $\omega[n]$ is the time-derivative of the expression inside the cosine, also known as the instantaneous phase.

(a) Determine the instantaneous frequency of $x[n]$ for $n = 0$, $n = 1000$, and $n = 2500$. Express the frequencies as a function of π (e.g., $\pi/100$).

(b) Plot $x[n]$ with a length of 2501. Label the horizontal axis "Samples" and vertical axis "Amplitude." Also, use the `fft` function to compute the DFT of $x[n]$ (i.e., $X[k]$). Use the `abs` function in MATLAB to plot the magnitude response $|X[k]|$. Label your horizontal axis "Normalized frequency [rad / s]" with values $2\pi k/N$. Label your vertical axis "Magnitude." After plotting, apply the command `axis([0 2*pi 0 max(abs(X))])` to see the figure from 0 to 2π in frequency and 0 to the maximum value in magnitude.

Since the DFT / DTFT is poor for analyzing such signals, we may instead analyze signal **segments**. Yet, this can affect the precision of our results. To demonstrate this, consider the cosine

$$y[n] = \cos((\pi/2)n)$$

and a rectangular window defined by

$$w[n] = u[n] - u[n - N]$$

(c) Compute and sketch the magnitude and phase of the DTFT of $y[n]$.

(d) Compute and sketch the magnitude and phase of the DTFT $Z(\omega)$ of a segment of the cosine signal, defined by $z[n] = y[n]w[n]$ for $N = 10$.

(e) **(EEE 5502 Only)** Show that the "bandwidth" of the $W(\omega)$ (defined as the length between the two locations nearest to $\omega = 0$ where $W(\omega) = 0$) is inversely proportional to the width of the rectangular window N.

(f) **(EEE 5502 Only)** This result implies that the we cannot simultaneously analyze the frequency content of a signal with perfect temporal resolution and perfect frequency resolution. We often refer to this as a time-frequency uncertainty principle. Explain how the previous result shows this.

Question #3: *(6 pts)* *The Discrete Fourier Transform over Time*

As we illustrated in the previous problem, the DFT is not particularly useful for describing chirp signals. In this problem, we will compute the DFT over time (i.e., a short-time Fourier transform) to better describe the chirp's behavior.

(a) Code the following short-time Fourier Transform process for your chirp $x[n]$:

- 1) Compute `M = floor(N/W)`, the number of length-W segments in $x[n]$ (where $N = 2501$ is the length of the chirp).
- 2) Initialize a matrix `STFT = zeros(W, M);`.
- 3) Extract the first W samples (samples 0 to $W - 1$) of the signal
- 4) Compute the DFT (using the `fft` function) of these W samples (note: this is equivalent to computed the windowed signal segment)
- 5) Store the result of the DFT in `STFT(:,m)` where $m = 1$.
- 6) Iteratively repeat steps # 3 to # 5 for the next W samples (i.e., samples W to $2W-1$ and then samples $2W$ to $3W - 1$ ... until you reach samples $(M - 1)W$ to $MW - 1$). Increase `m` with each iteration.

Use the MATLAB code

```
imagesc(0:(M-1), 2*pi*(0:(W-1))/W, abs(STFT))
xlabel('Time [samples]');
ylabel('Normalized Frequency [rad/s]');
```

to plot the short-time Fourier transform magnitude values for a W of your choosing.

(b) Plot the short-time Fourier transform values for 8 different values of W. Specifically, consider W equal to 10, 20, 40, 80, 160, 320, 640, and 1280.

(c) Your plots should illustrate the time-frequency uncertainty principle. For which value of W do you think best illustrates the chirp signal?

Question #4: *(7 pts)* *Audio DFT over Time*

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

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

(a) Compute the short-time Fourier transform of your Rudenko music `x`. Use the MATLAB code

```
imagesc((0:(M-1))*W/Fs, 2*pi/W*(0:(W-1)), abs(STFT))
xlabel('Time [seconds]')
ylabel('Normalized Frequency [rad/s]');
```

to plot the short-time Fourier transform magnitude values for $W = 10,000$. Use the `axis` command from Question #2 to focus only on the lower frequencies from 0 to $\pi/20$. Note that the code above is plotting time in seconds since we know the sampling rate.

(b) Use slightly different MATLAB code

```
imagesc((0:(M-1))*W/Fs, 2*pi/W*(0:(W-1)), 10*log10(abs(STFT)./
    max(max(abs(STFT)))), [-20 -5])
xlabel('Time [seconds]')
ylabel('Normalized Frequency [rad/s]');
```

to plot the short-time Fourier transform magnitude values for the same W. This normalizes the data so that the maximum value is 1 (0 dB) and plots the values in the decibel range of -20 dB to -5 dB to make the data is easier to visualize. Again, use the `axis` command from Question #2 to focus only on the lower frequencies from 0 to $\pi/20$.

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

$$h[n] = \frac{1}{10000} \sum_{k=0}^{10000-1} \delta[n-k] \ .$$

Use `conv` or `fft` to filter $x[n]$ and get an output $y[n]$. Then compute the short-time Fourier transform of the filtered signal. Plot the short-time Fourier transform magnitude values for the same W you chose in part (b).

(d) How does the filter affect the short-time Fourier transform? How does the filter affect the music?