**Full Name:** _____

**EEL 4750 / EEE 5502 (Fall 2019) – Code #06** **Due Date:** **Nov. 18, 2019**

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

**Question #2:** *(6 pts)* *Overlap-Add Theory*

(a) Consider the rectangular window function

$$w[n] = u[n] - u[n - W] \ .$$

Show that

$$\sum_{k=-\infty}^{\infty} w\left[n - Wk\right] = c_1 \ .$$

Prove this for any $W$. Determine $c_1$.

(b) **(EEE 5502 Only)** Consider the rectangular window function

$$w[n] = u[n] - u[n - W] \ .$$

Show that

$$\sum_{k=-\infty}^{\infty} w\left[n - \left(\frac{W}{2}\right)k\right] = c_2$$

for when $W$ is even. Prove this for any even $W$. Determine $c_2$.

(c) **(EEE 5502 Only)** Consider the Hann window function

$$w[n] = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{W-1}\right)\right]\left[u[n] - u[n - W]\right] \ .$$

Show that

$$\sum_{k=-\infty}^{\infty} w\left[n - \left(\frac{W-1}{2}\right)k\right] = c_3$$

for when $W$ is odd. Prove this for any odd $W$. Determine $c_3$.

**Question #3:** *(6 pts) Failures of STFT*

Included with this assignment is a `[y, xSTFT, ySTFT] = stft_denoise(x, W)` function to compute the STFT of `x` with window size `W`. As in the last assignment, the function outputs of a modified signal `y` and two STFTs, `xSTFT` and `ySTFT`. Use a window of length $W = 2000$.

(a) The function `[y, xSTFT, ySTFT] = stft_denoise(x, W)` adds three lines into the code. Explain what this first line accomplishes:

```
mSTFT(:,m) = lambda*abs(mSTFT(:,m-1)) + (1-lambda)*abs(xSTFT(:,m));
```

(b) Explain what the next two lines of code accomplish:

```
ySTFT(:,m) = xSTFT(:,m);
for k = 1:W, if ySTFT(k,m) < mSTFT(k,m)*2, ySTFT(k,m) = 0;  end; end
```

(c) Load `noisy_speech.wav`. With the provided STFT function, process the noisy speech. How does the processing change the audio? Plot the STFT of the sound with and without the processing. Plot the time axis in second. Plot the frequency axis in Hz.

**Question #4:** *(8 pts) Overlap-Add*

You may have noticed choppiness in the reconstructed time signal. This is because there is no smooth transition from one frame to another. In this question, we will fix this problem. To remove the choppiness, change the `STFT_func` and/or `STFT_denoise` function (we will use the modified `STFT_denoise` function later) such that:

- Each time you compute the FFT, shift the frame by $W/2$ instead of $W$. Hence, each frame will have a 50% overlap with an adjacent frame.
- After computing each IFFT, multiply the length-$W$ time-domain signal with a Hann window
$$w[n] = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{W-1}\right)\right] .$$
This creates a smooth transition between frames.
- After computing each IFFT and multiplying each frame by a Hann window, sum the overlapping components of the adjacent frames.

This process is known as overlap-add.

**For each part below, plot the magnitude of the STFT for the original music signal *and* the modified music signal. Use `axis` to zoom-in on same relevant (i.e., non-zero) information for each part.** For each part, use a window of size $W = 2000$.

(a) Submit the new function with your changes.

(b) Load `noisy_speech.wav`. With your new overlap-add STFT, implement the `stft_denoise` function from Question #2 across plot the STFT.

(c) Answer: how and why does the audio change due to your overlapping and adding.