Full Name: _____

EEL 4750 / EEE 5502 (Fall 2018) – Code #07 **Due Date:** **Dec. 5, 2018**

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

Question #2: *(20 pts)* *Spectral Subtraction*

A classic, effective approach for noise reduction in speech signal processing is known as spectral subtraction. The spectral subtraction process is as follows:

- **Step 1:** Obtain your noisy signal $x[n]$. We assume the speech contains both noise-free audio $y[n]$ and noisy audio $n[n]$ such that $x[n] = y[n] + n[n]$.
- **Step 2:** Compute the STFT of $x[n]$ to get $X[k, m]$, where k corresponds to frequency and m corresponds to time / frames. Again, we assume the speech contains both noise-free audio $Y[k, m]$ and noisy audio $N[k, m]$ such that $X[k, m] = Y[k, m] + N[k, m]$.
- **Step 3:** Extract the phase response of the STFT to get $P[k, m] = \angle X[k, m]$.
- **Step 4:** Estimate the **magnitude** of the noise signal

  $$|\widehat{N}[k, m]| = \lambda |X[k, m]| + (1 - \lambda)|\widehat{N}[k, m - 1]| \, ,$$

  where $0 < \lambda < 1$. This assumes that speakers do not use the same frequencies over an extended period of time. That is, when you speak, the frequencies you use constantly change. Hence the above difference equation smooths each individual frequency. A high λ retains more of the previous estimate, producing a greater smoothing effect. A low λ uses more of the current measurement, producing a weaker smoothing effect.
- **Step 5:** Estimate the magnitude of the speech signal by subtracting the estimated noise STFT magnitude from the original STFT magnitude according to

  $$|\widehat{Y}[k, m]| = \begin{cases} |X[k, m]| - |\widehat{N}[k, m]| & \text{if} \quad |X[k, m]| - |\widehat{N}[k, m]| > 0 \\ 0 & \text{if} \quad |X[k, m]| - |\widehat{N}[k, m]| \leq 0 \end{cases}$$

- **Step 6:** Recombine the phase response of the STFT with your estimate to obtain the complete processed speech STFT

  $$\widehat{Y}[k, m] = |\widehat{Y}[k, m]| \exp(jP[k, m])$$

- **Step 7:** Compute the inverse STFT of $\widehat{Y}[k, m]$ to get $y[k, m]$.

Perform spectral subtraction on the audio file `noisy_speech.wav`, a segment of speech from the 1992 computer game *Star Control 2* with noise. Choose your own window length and λ parameter. You may want a large window (e.g., W ¿ 1000) and/or large λ (e.g., $\lambda > 0.9$) to robustly characterize the noise. I encourage you to experiment.

Similar to the last two coding assignments, submit an image of the STFT before and after the spectral subtraction processing. Submit plots of the time-domain signals before and after spectral subtraction. Submit a .wav audio file of the speech signal after spectral subtraction (type `help audiowrite` to learn how to make this file). Finally, submit your modified `stft_func.m` file (see next page for details about the function).

To help you with this process, we included with this assignment a new `stft_func` that computes the STFT (with 50% overlap). This part of the function is:

```
z = x((W*(m-1)/2+1):(W*(m-1)/2+W));          % Get data segment
xSTFT(:,m) = fft(z);                          % Fourier Transform
```

The function then processes the signal (currently, it does nothing) in the section:

```
% ***** PERFORM PROCESSING HERE AND ASSIGN ySTFT *****
ySTFT(:,m) = xSTFT(:,m);                      % REPLACE THIS
% ************************************************
```

This is the only part of the code that you need to change in this assignment (note: your code will be more than one line). The code then performs the inverse STFT with overlap-add on the processed data. This part of the function is:

```
y((W*(m-1)/2+1):(W*(m-1)/2+W)) = ...          % Inv. Fourier Transform
        y((W*(m-1)/2+1):(W*(m-1)/2+W)) + ...
        real(ifft(ySTFT(:,m))).*hann(W);
```

Note that $X[k,m]$ is `xSTFT` in the function and $\widehat{Y}[k,m]$ is `ySTFT`. Again, for this assignment, modify the middle process part of this function to perform spectral subtraction. Your result should be able to remove most of the noise from the audio file.