

CSCI 1320 Computer Science I: Engineering Applications – Fall 2018
Instructor: Zagrodzki
Assignment 5
Due Sunday, October 14 by 6pm

Sound Effects and Analysis Tool

Make sure that you complete Lab 5 before starting on this assignment.

For this assignment you will create an audio effects and analysis tool. Your program will need to have a Graphical User Interface (GUI) menu with the following functionality:

1. Choose a File. This menu item will allow the user to enter exact name of audio file to be read into the program. The file has to be in .wav format. The audio vector and sampling frequency (Fs) from this module will be used for the remaining features of this program.
2. Play the audio vector.
3. Echo effect. Create a function that will add an echo effect to the audio signal. The function should take in the sound vector, sampling frequency, delay in seconds, and echo amplitude. The function should return a vector containing the echoed sound.

```
sounVecEcho = soundEcho(soundVec, Fs, delay, echoGain)
```

Steps you should follow when designing your function:

- a. Take in the audio vector and sampling frequency, delay and echo gain.
 - b. Convert the delay in seconds to number of samples
`delaySamples = Fs * delay;`
 - c. Figure out a way to add a delayed version of the signal vector to the original sound vector. `delaySamples` tells you how many elements the offset needs to be.
 - d. Play the new version of your sound vector.
4. Compress. Create a function that takes in the sound vector and compression ratio. The function should compress the file by removing vector elements based on the compression ratio. (i.e. if you the compression ratio is 3, keep only every third element.) The function should return the compressed vector.

```
compSoundVec = compress(soundVec, compression ratio)
```

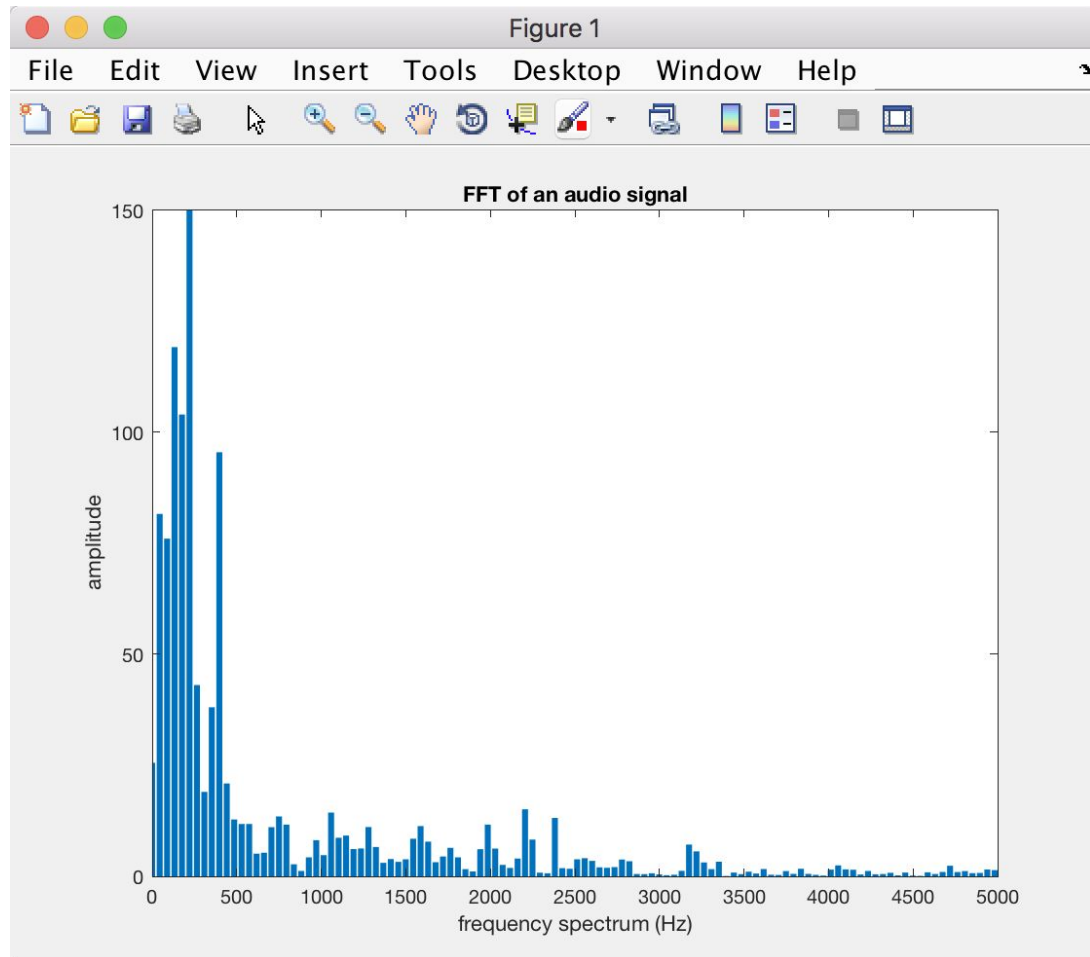
5. Frequency spectrum. While it is perhaps most intuitive to consider signals in the time domain (i.e. plot amplitude vs time), the frequency domain can tell us interesting information about the energy of a given signal. We can transform an audio signal from the time domain to the frequency domain by applying the Discrete Fourier Transform. Write a function that will do the following:

```
fftBar(soundVec, Fs)
```

- a. Take in the audio vector and sampling frequency.
- b. Only use at the first 1000 samples of the audio vector.
- c. Use the fast fourier transform function in MATLAB (fft) to convert the audio signal into the frequency domain.
`Y = fft(audioVec);`
- d. The fft function will return the input signal transformed into the frequency domain. You will see that the vector elements will contain both real and imaginary components. For the purpose of this function, we will go ahead and ignore the imaginary component of the signal. This can be done by taking the absolute value of the entire vector .
`Y = abs(Y);`
- e. The fft function returns both positive and negative frequencies (referred to as a 2-sided plot). We only want to display the positive ones. In order to do so, create a new vector that only contains the first half of the original vec (Y).
- f. Create a frequency vector. We will need this for our bar graph. Use the following expression:

```
f = Fs*(0:(length(Y)-1)/2)/length(Y);
```

- g. Create a bar graph with frequencies on the horizontal axis and amplitudes on the vertical axis. Format your horizontal axis to show frequencies from 0 to 5 kHz. Your vertical axis should be scaled such that the maximum Y value takes up most of the bar graph.
- h. The function does not need to return anything.



Sample FFT plot

Submitting the assignment:

You should have 1 script and at least 3 functions (`soundEcho`, `compress`, `fftBar`) for this program. Make sure all your .m files are well commented and they include a comment block with your name, student ID, course number, assignment number and recitation section. Zip all the files together and submit the resulting .zip file through Moodle as Assignment 5 by due date.