Engr325 Lab #4 Winter, 2020

Data Acquisition and the Fast Fourier Transform

Name ________________________________

Partner (s)______________________________ **Grade** _______**/10**

Introduction

The goal of this lab is to explore the usefulness of the FFT as it applies to acquired signals.

Objectives

- Further your learning about data acquisition using A/D converters;
- Learn proper technique for sampling an analog signal;
- Apply the Fourier transform to find the frequency content of various composite signals.

Equipment Provided

- Oscilloscope;
- Signal generator;
- Computer with ADC and connection boards.

References

- Instrumentation text book;
- Course web page
  - Lecture notes;
  - Matlab scripts.

Procedure

Part 1:

1) Using the **Arb** button on the signal generator, choose *Select Waveform → Built-in waveforms → Cardiac* option to output an approximation for an EKG signal.

2) Go to the course web page and use the Matlab script under Lab #3 to acquire the signal from the function generator. You shouldn't have to modify this script. Then, using the Matlab script under Lab #4 (modify ***carefully***), use the **fft** function to display the frequency components present in the waveform. Set the frequency to 500Hz and the amplitude to $2V_{pp}$ with no DC offset. Print both a time-domain and a frequency-domain plot.

Part 2:

Voice analysis uses ratios of frequencies throughout different parts of a word to figure out what word was said. In this part of the lab you will be designing a Matlab program that will recognize whether a person says "yes" or "no." When saying the word "yes", the frequencies in the letter "s" are much higher than those in the letters "y" and "e." However, frequencies of "n" and "o" in the word "no" are relatively the same frequency. Because of this, you can take advantage of the frequency content of the two words to distinguish between them.

1) Import an audio file of your own voice into Matlab from either your smart device or your laptop. For processing in Matlab, the best audio file to create is a short one (1 second or so) with just the one spoken word. Make two recordings – one with a spoken 'yes' and one with a spoken 'no'. Try and keep the volume equal between the two, and speak the 'no' with a low, steady voice and the 'yes' with a higher voice, emphasizing the 's' sound.

2) Matlab can read a number of different file formats. Use a command like:

```
audiofile = '/filepath/audio.mp3';
```

3) Convert the audio file into a magnitude vector (**s0**) and the sampling frequency (**f0**):

```
[s0,f0] = audioread(audiofile);
```

4) First, print a time domain plot of both the 'yes' and 'no' recordings.

5) Determine the frequency content by displaying a Fast Fourier Transform (**fft)** of **s0**, for both a spoken 'yes' and 'no'. You should graphically display the fft of each including the frequency range between 0 – 10kHz. Print your plots.

6) Visually determine a strategy on how you can differentiate between a 'yes' and 'no' in Matlab. Implement your strategy as best you can and show the lab instructor correct operation.

To Turn In

1) Each ***person*** should staple your plots to the back of this handout and turn them in.
2) Each ***person*** should place a copy of your Matlab code (both parts I and II) into the lab#4 drop box in D2L.