lec1_exercises

February 15, 2019

1 An Introduction to the Discrete Fourier Transform

2 Lecture 1: Overview

3 This is a pdf version of this interactive programming assignment. It is recommended that you download the zip file and use Jupyter Notebook to write and run the code.

3.1 The following is the exercises for the first lecture on the Discrete Fourier Transform. See this lecture video here. For more supplemental resources, see also the course website.

3.1.1 The two text files "piano.txt" and "trumpet.txt" are files from the book "Computational Physics" by Mark Newman. More infomation here.

3.1.2 The "piano.wav" file is synthesized by one of my students Lukas Maldonadowerk.

3.2 Exercises

```
In [ ]: import numpy as np
        from IPython.display import Audio
        import matplotlib.pyplot as plt
        from scipy.io import wavfile
        %matplotlib notebook
```

Use `np.loadtxt(filename)` to import `trumpet.txt` and load into an array `ys`. Create an Audio object with the data from the text file at the sampling rate `44100` Hz. Play the audio.

Create and initialized the variables `N`, the number of samples, `fs`, the sampling rate and `L` the length of the audio.

Use `np.linspace(start, stop, num samples)` to create array of N time samples `ts` for the time interval $[0, L]$.

Use `matplotlib` to plot the waveform of the trumpet recording. Use `ax.set_xlim(x1, x2)` to zoom in.

```
fig, ax = plt.subplots()
ax.plot(ts, ys)
```

Now use `np.fft.fft(samples)` to convert the samples to Fourier coefficients. Then create the array of frequencies(harmonics) f_k that the DFT will detect. $f_k = k/L$ where $k = 0, 1, 2..., N-1$.

Now plot the frequency domain representation (frequencies vs magnitude of Fourier coefficients). What note did the trumpet play? What are the harmonics? You may use the interactive feature of the matplotlib graph to estimate these values.

As in the lecture, collect your code above to write the function `plot_signal_time` to plot the waveform as a function of time. We will use this function frequently in the next few Jupyter notebook problem sets.

Call the function to plot the waveform of the trumpet. Zoom in.

```
In [ ]: def plot_signal_time(ys, t1, t2, fs = 44100):
            """ plots the signal ys on the time domain [t2, t2]
            at the sampling rate fs.
            """
```

As in the lecture, collect your code above to write the function `plot_signal_frequency` to plot the spectrum of the signal. We will use this function frequently in the next few Jupyter notebook problem sets.

```
In [ ]: def plot_signal_frequency(ys, f1, f2, fs = 44100):
            """ plots the signal ys on the frequency domain [f1, f2]
            at the sampling rate fs.
            """
```

Import `piano.txt` and use `plot_signal_frequency` above to plot the magnitudes of the Fourier coefficients. Compare this frequency domain of the trumpet to that of the piano. Answer:

We used the interactive feature of matplotlib to estimate the fundamental frequency and its harmonics. Now use Python code to find the fundamental frequency exactly of the piano recording. Hint: Use `np.argsort(array)` to sort an array. This function returns the indices that would sort the array. The formula $f_k = k/L$ will be useful.

Find the three most dominant(largest Fourier coefficient magnitude) frequencies in the trumpet recording. Slicing will be helpful: Given an array arr, `arr[start:stop]` returns the subsequence from index start to index stop(excluding).

Use the `read(filename)` function from the `wavfile` module to read in "piano.wav". Play the audio file and plot the frequencies and determine chord of the piano recording.

```
fs, ys = wavfile.read("piano.wav")
```

The following website of notes/frequencies will be helpful. http://pages.mtu.edu/~suits/notefreqs.html