ECE 4750/6750 + CS 4501/6501: Digital Signal Processing **Due Th 2/29**

Homework 2: Discrete Fourier Transform

Instructions: Submit a single Jupyter notebook (.ipynb) of your work to Canvas by 11:59pm on the due date. All code should be written in Python. **Be sure to show all the work involved in deriving your answers! If you just give a final answer without explanation, you may not receive credit for that question.**

You may discuss the concepts with your classmates, but write up the answers entirely on your own. Do not look at another student's answers, do not use answers from the internet or other sources, and do not show your answers to anyone. **Cite any sources you used outside of the class material (webpages, etc.), and list any fellow students with whom you discussed the homework concepts.**

DFT Audio Signal Analysis

You will be using the fast Fourier transform (FFT) to analyze two audio signals. These are a clip of a person speaking (`winniethepooh.wav`)[1] and a clip of piano music (`bach.wav`)[2]. Download these files from the class webpage.

Note: To compute the DFT you can use the `rfft` and `irfft` functions from the `numpy.fft` library, which only compute $L/2$ length Fourier transforms because of the symmetry of the DFT of real signals ($X[L-k] = \bar{X}[k]$).

1. The frequency band for voice used in telephones is approximately 300 to 3400 Hz.[3]. To simulate this, take the FFT of the voice audio signal and apply a bandpass filter in the voice frequency range. That is, set the $Y[k] = w[k]X[k]$, where $w[k]$ is set to one for frequencies in the 300–3400 Hz range and set to zero elsewhere. **Note:** you will need to first calculate the conversion of the frequency parameter k into Hz. Take the inverse FFT and listen to the resulting audio.

2. According to the same Wikipedia page, the fundamental frequency of a male speaking voice is between 85 to 155 Hz. Apply the same bandpass filter procedure as above, but with the frequency range of 85–310 Hz. (We have to double the upper limit of the frequency range to avoid aliasing. This is due to what is called the Nyquist-Shannon theorem, which we'll cover later in class!) Again, listen to the resulting audio. Is the speech still understandable when keeping only the fundamental frequencies, or were the harmonic overtones important contributions?

3. Now compute an FFT of the piano music signal. Plot the magnitude of the FFT, $|X[k]|$, as a function of frequency in Hz. What are the 3 most prominent frequencies in Hz (a.k.a., what is the position on the frequency axis of the three highest peaks in the plot)? Refer to the following table of musical note frequencies:

[1] Chapter 5 of A. A. Milne's classic, *Winnie the Pooh*, read by Phil Chenevert, audio from LibriVox: https://librivox.org/winnie-the-pooh-by-a-a-milne/

[2] J. S. Bach's *Two Part Invention Number 13 in A Minor*

[3] https://en.wikipedia.org/wiki/Voice_frequency

https://en.wikipedia.org/wiki/Scientific_pitch_notation#Table_of_note_frequencies
What musical notes do these frequencies represent? Give the note letter and octave number, for example, 440 Hz is the note A4. **Hint: They may not be the notes you are expecting for a piece in the key of A minor!**

DFT Exercises

4. Consider the following periodic signal, with period L:

$$x[n] = \frac{1}{2}(\delta[n-1] - \delta[n-L+1]), \qquad \text{for } 0 \leq n \leq L-1.$$

   (a) Using the definition of the DFT analysis equation, compute $X[k]$. Simplify your equation until it is clear that $X[k]$ is imaginary (in the form of i times a real number). **Show your steps.**

   (b) Now compute the DFT $X[k]$ by starting with a DFT pair we discussed in class and applying properties of the DFT. Confirm that it is equal to the answer you got in part (a). **Again, show your steps.**

   (c) In Python, construct the signal $x[n]$ for $L = 10$. Compute the DFT using the `numpy.fft.fft` function. **Note: Use the parameter `norm = "ortho"` in this function to get the $\frac{1}{\sqrt{L}}$ scale factor in the DFT.** Plot the resulting $X[k]$ and compare it to a plot of your equation in part (a).

5. Consider the following periodic signal, with period L being an **even number**:

$$x[n] = \begin{cases} 1 & \text{for } n \text{ even,} \\ -1 & \text{for } n \text{ odd.} \end{cases}$$

   (a) Write an equation for $X[k] = \mathcal{DFT}\{x[n]\}$. Simplify your equation until it is clear that $X[k]$ is real-valued. **Show your steps.**

   (b) In Python, construct the signal for $L = 10$. Compute the DFT using `numpy.fft.fft` function. Plot the resulting $X[k]$ and compare it to the plot of your equation in part (a).

For Grads Only (or Extra Credit for Undergrads)

6. Consider the following periodic signal, with period L being an **even number**:

$$x[n] = \begin{cases} \cos\left(\frac{2\pi n}{L}\right) & \text{for } n \text{ even,} \\ -\cos\left(\frac{2\pi n}{L}\right) & \text{for } n \text{ odd.} \end{cases}$$

   (a) Write an equation for $X[k] = \mathcal{DFT}\{x[n]\}$. Simplify your equation until it is clear that $X[k]$ is real-valued. **Show your steps.**

   (b) In Python, construct the signal $x[n]$ for $L = 40$. Compute the DFT using the `numpy.fft.fft` function. Plot the resulting $X[k]$ and compare it to a plot of your equation in part (a).