Lab 2: Frequency Analysis

CSL607 Multimedia Systems

Due at 11:59 PM, Aug 14, 2019

Instructions

- You need to be present in the lab for at least 2 hour to submit for the lab to be graded. The TA will be available during the lab hours to help you with the lab exercise.
- You have to work individually for this lab.
- You are not allowed to share code with your classmates. You are encouraged engage in high level discussions with your classmates; however ensure to include their names in the report. If you refer to any source on the Internet, include the corresponding citation in the report. If we find that you have copied code from your classmate or from the Internet without proper reference, you will get a straight F grade in the course.
- The submission must be a zip file with the following naming convention - rollnumber.zip.
- Include appropriate comments to document the code. Include a `read me` file containing the instructions on for executing the code. The code should run on institute linux machines.
- Upload your submission to moodle by the due date and time. Do not email the submission to the instructor or the TA.

1 Introduction

The goal of this lab it to familiarize you with frequency characteristics of a signal. Frequency is very closely related to semantics. For example, male speech has a different frequency characteristic than female speech. A perfectly periodic and infinite time signal can be represented using a set of discrete sine waves. However, in real life we generally deal with limited time signal which may not be periodic. It is believed that such signals consist of an infinite number of frequencies, i.e., it can no longer be represented using discrete frequencies but needs a continuous frequency axis.

As you know, computers cannot deal with the continuous numbers. Fourier Transform is a powerful technique that provides a sampled version of the frequency axis, i.e., an N point Fourier Transform results in magnitude and phase of N frequencies that are uniformly spaced between 0 and $\frac{f_s}{2}$. Recall that according to the sampling theorem, $f_m \leq \frac{f_s}{2}$ for faithful reconstruction of the original signal (f_m is the input signal bandwidth, i.e. highest frequency).

If x is input signal and X is its Fourier Transform, then the k^{th} frequency component is calculated as follows:

$$X_k = \sum_{n=0}^{N-1} x_n.e^{\frac{-i2\pi kn}{N}} \tag{1}$$

and from a given Fourier Transform, n^{th} sample is recovered as follows:

$$x_n = \sum_{k=0}^{N-1} X_k.e^{\frac{i2\pi kn}{N}}. \tag{2}$$

A computationally efficient implementation of Fourier Transform is called FFT (Fast Fourier Transform). In Python, you should look for FFT and iFFT methods to work with Fourier Transform. Remember that actual FFT consists of both imaginary and real components, but we are only interested in the magnitude of the frequency response. The size of FFT (N) is generally taken the same as the number of input samples. IMPORTANT: Run a Google search to understand the meaning of each of the concepts discussed in this exercise. You may need fftpack module from scipy package, for details please see this: `https://docs.scipy.org/doc/scipy/reference/tutorial/fftpack.html#one-dimensional-discrete-fourier-transforms`. You can also use numpy modules to calculate fft and ifft, see this `https://docs.scipy.org/doc/numpy/reference/routines.fft.html`.

2 Fourier Transform of a Since Wave

Simulate a scenario that you are sampling a one second sinewave, see the code below. It divides 1 second in Fs equispaced points. This means you are taking Fs samples per second, and hence Fs is the sampling frequency. You can control f to change the frequency.

```
# Import matplotlib.pyplot and numpy for plotting and numerical operations
import matplotlib.pyplot as plt
import numpy as np
# Set the sampling rate to 100
Fs = 100
# Set frequency to 1, i.e., 1 cycle per second or 1 cycle per Fs samples
f = 1
# Create an array of 100 items in increasing order from 1 to 100
# This is the x-axis for plotting
x = np.arange(Fs)
# compute the value (amplitude) of the sin wave at the for each sample
y = np.sin((2*np.pi*f * x)/Fs )
# Plot the just created sine wave
plt.stem(x, y, use_line_collection=True)
plt.plot(x,y)
plt.show()
```

Listing 1: Sine wave generation

Take Fs to be 100 and vary the frequency from 1 Hz to 100 Hz and observe the signal as well as it Fourier transform with fft() function. Reconstruct the original signal with ifft function and plot. Write your observations particularly in the context of sampling frequency.

3 Digital Vs Analog Frequency

Read the audio file (killbill.mp3) in a variable. The audio may be dual channel, which will result in two vectors. In most cases these are copies of each other. Take any one vector for analysis. Obtain Fourier transform of the audio signal (read without native option) with fft() function. Plot double sided magnitude response using abs() and plot() functions. Mark the y-axis as the magnitude and x-axis as the analog frequency of the signal. Make sure that the DC component (zero frequency) is in the center of the frequency response. How will you do that?

4 Male and Female Voice Analysis

Take around 5s audio files of a male and a female student (ask your colleague to volunteer for it). Obtain the Fourier transform using the code implemented in the previous section. Write your observations about magnitude of the frequency response. You can also take the audio file killbill.mp3 as an example which has female voice and male voice both. You need to check the sampling rate and time to get the exact sample number where male voice starts.

5 Music and Speech Analysis

Repeat the previous experiment with speech and music files. Observe difference in frequency components of music and speech. You can take killbill.mp3 as example which has speech followed by music.
Now manipulate different frequency coefficients (by zeroing particular frequency coefficients), use ifft to get back the original signal, play it, and observe the effect. Write your observations. Can you make a male speech sound like a female voice?

6 Submission

Submit all the Python codes written and a report (Max four pages) with: all observations. Zip all files and name as lab2_entryNumber.zip. There will be a viva of the submission where you will have to explain your observations.