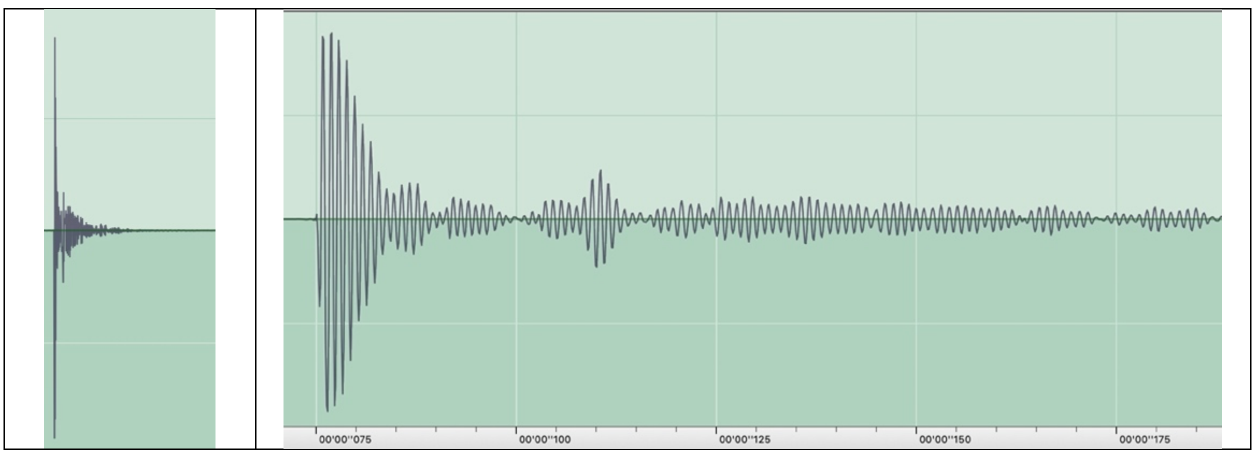
CPE 367 – Experiment 7 v7b (60 points)

Spectral-Based Acoustic Measurements and the DFT – Dr F DePiero

Overview

This experiment provides an opportunity to better understand the DFT. We will use the DFT to measure the size of an air gap enclosed within solid material. Measurement systems similar to this are used for inspection purposes, for example to find cracks inside railroad tracks or in the walls of a pipeline. Different measurement principles are employed on these kinds of inspection systems (e.g., time of flight). In our case we will find the frequency of an acoustic signal, to determine the size of a void.

Figures 1a and 1b show time-domain plots of our acoustic signal. The right plot is zoomed horizontally, revealing a sinusoidal component (~1kHz). This frequency is inversely proportional to the size of an air gap underneath a 12” x 20” ceramic floor tile and recording the resulting sound with a standard microphone. The acoustic signal was generated by dropping a golf ball on the tile floor. The impulse generated by the golf ball (theoretically) excites all possible modes of oscillation of the air within the gap; however, the bulk of the energy from the impulse goes into the fundamental frequency. This is evident in the sinusoidal component seen in Figure 1b and in the spectrum of Figure 2.



Figures 1a and 1b. The acoustic signal is a brief impulse of sound, under 1/10 second in duration. Within the impulse, a sinusoidal component at ~1kHz is visible.

Learning Objectives

- Implement the DFT in software, manipulating complex-valued signals
- Verify properties of Fourier transforms and convolution
- Predict the spectrum of a signal based on properties of the Fourier transform
- Create a measurement system based on the spectrum of an acoustic signal
- Reliably determine the location of a peak using a weighted average

Prerequisite Learning Objectives

- Knowledge of the mathematical definition of the DFT
- Python programming with WAV file I/O

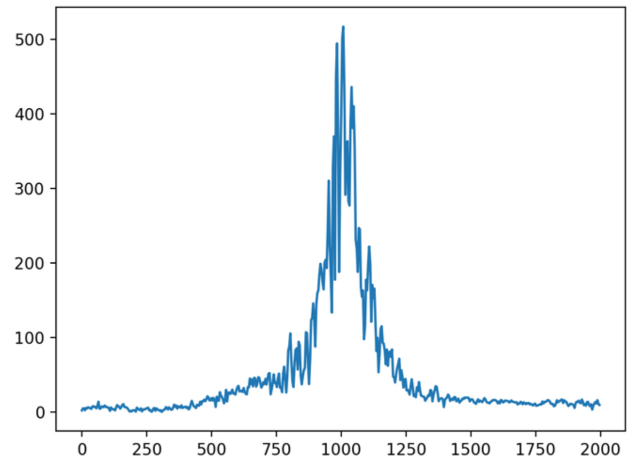


Figure 2. Spectrum of the acoustic signal from the standing wave in the cavity. Resonant frequency ~1kHz.

Background – Measurement An air gap (or void) within solid material can form a resonant cavity for sound. The resonant frequency of the cavity depends on the dimension of the enclosed space. As sound waves propagate through the cavity and reflect off each wall, they form a standing wave pattern. Larger cavities have lower pitched sounds because more time is required for the sound waves to traverse the cavity and reflect back into it. Further information and animations of standing waves are available at WikiPedia.

The air molecules in the cavity have little motion near the walls (nodes) and maximum motion (antinode) in the middle of the void. The node-antinode-node waveshape corresponds to half a wavelength of the standing wave. (A full wavelength would include node-antinode-node-antinode-node). The relationship between wavelength and frequency is given by $\lambda f = v$, where f is frequency in Hz, v is the velocity of propagation in air (343 meters/sec) and lambda is the wavelength in meters. Hence for f_0 in Hz, the dimension, L, of the air gap is

$$L = \frac{\lambda}{2} = \frac{v}{2f_0} = \frac{343}{2f_0} \text{ (meters)}$$

Note that depending on the complexity of the shape of a cavity, it is possible for multiple standing waves to occur simultaneously. (Consider a rectangle with standing waves along both the longer and shorter dimensions). With our measurements, a single standing wave pattern and single mode dominate the signals.

Background - Spectrum The overall shape of the spectrum seen in Figure 2 can be appreciated by first modeling the time domain signal. The sound impulse can be approximated as the product of a sinusoid with an exponentially decaying pulse. Because we multiply in time, the resulting spectrum is a convolution between the impulses of a sinusoid (at +/- f_0) the spectrum of the exponential signal, E(f). See Figure 3.

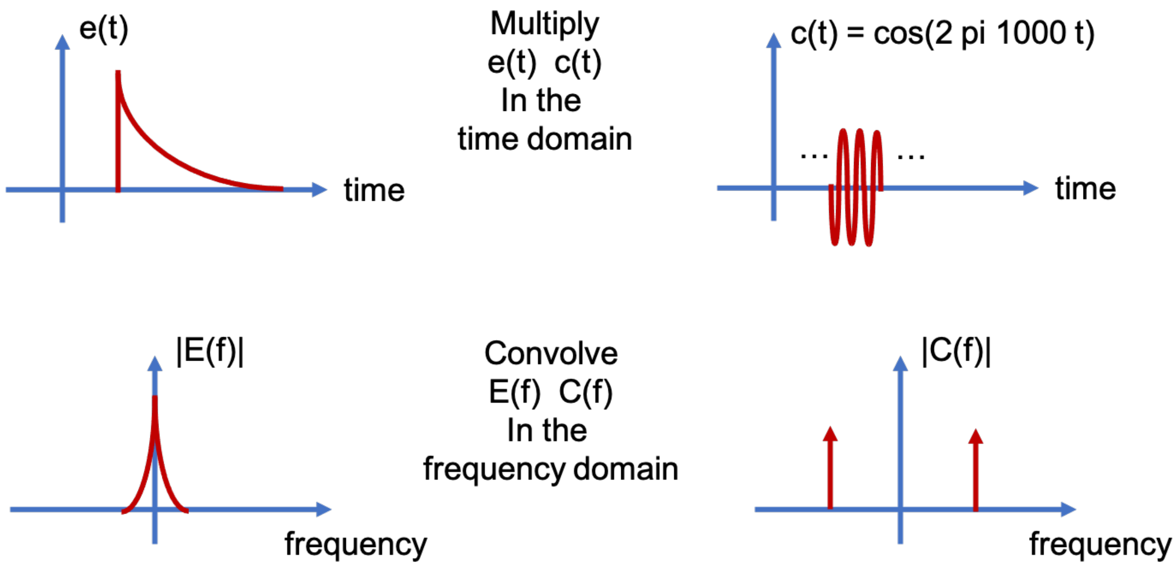


Figure 3. Multiplication of a sinusoid by an exponential envelope in time results in two copies of the spectrum of E(f) in the frequency domain.

This results in two copies of E(f), one centered at $+f_0$ and one at $-f_0$. See the plot in Figure 4, which shows the spectrum of $E(f - f_0)$, over the range of positive frequencies.

$$E(f) * \left[\, \delta(f_0) + \delta(-f_0) \,\right] = E(f + f_0) + E(f - f_0)$$

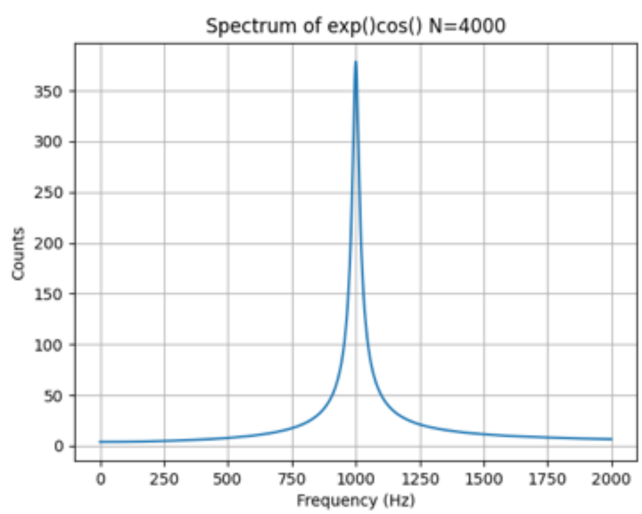


Figure 4. Spectrum corresponding to the product of the exponential with a 1kHz sinusoid. This spectrum is associated with mathematical models of the waveforms – not an acquired signal.

Background - DFT The DFT can be defined as follows, including the 1/N factor

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j\,2\pi\,kn/N} \qquad 0 \le k\ < N$$

To implement this transform in a software or hardware we must address the matter of evaluating complex arithmetic. This can be readily accomplished by employing Eular's identity

$$e^{j\theta} = \cos(\theta) + j\,\sin\,(\theta)$$

Eular's identity permits the real and imaginary components of the DFT to be evaluated separately and stored separately. For example, in Python these components could be stored using a list associated with the index k and using a dictionary for the real and imaginary parts. FYI, here is an example of the syntax that might be used.

X[k][‘real’] = 0   X[k][‘imag’] = 0

You may wish to include additional items in your spectrum such as [‘fk_hz’] or [‘magnitude’].

Background – Peak Finder When making a measurement it is often necessary to determine some characteristic from acquired data. In some cases, this is done by fitting the raw data to a curve, for example. For this measurement, we need to find the frequency of oscillation of the air in the air gap. As shown in Figure 2, the spectrum is not an ideal impulse that would be associated with a single sinusoid. Rather the spectrum is impacted by the very brief exponential envelope (Fig 4), as well as noise (Fig 2) and likely the effect of less dominant modes of oscillation within the cavity. Despite the non-ideal aspects of the spectrum, the peak is reasonably evident as shown in Figure 2. We need to extract the frequency of the central peak in a reliable manner.

We will use the following method to determine the center frequency. First, find the maximum magnitude of all the samples of the spectrum. This can be described as a "raw" peak value. Define the magnitude of this raw peak as a parameter T. As suggested by Figure 2 the frequency of this raw peak isn't necessarily the best value to use for the measurement. We can establish a more accurate and reliable value for the center frequency (f_0 in Hz) using a weighted average calculation

$$f_0 = \frac{\sum_{k_1} f_{k_1}\ |\,X[k_1]\,|}{\sum_{k_1}\ |\,X[k_1]\,|}$$ (in Hz), for k_1 samples having $|\ X[k_1]\ | > (T/2)$

Here, $|\ X[k_1]\ |$ is the magnitude at sample k_1 and f_{k1} is the frequency in Hz of that sample. Note that only a limited number of samples are used; these are the samples with magnitudes above a threshold of T/2. As you will find, the end result of this approach will provide a consistent measurement of the center frequency of the peak. (You are free to experiment with variations on the threshold or on the weighted average calculation, but this approach seems to work reliably).

Procedures, Questions and Deliverables

1) Implement a DFT Write a Python program that loads the first N samples from a WAV file and then computes the DFT. Use N = 4000 and note that the signals used in this experiment are sampled at 8kHz. Then compute the magnitude of the spectrum. Plot the spectrum magnitude for the positive frequencies and label the horizontal axis in Hz. When plotting the magnitude, skip DC and plot up through 2000 Hz. The Y axis has the same units as the original signal. For our purposes we can assign the units as "Counts" (referring to the 2^16 integer values in the intensity of the WAV file). Consider creating functions that allow you to 1) acquire a list of signal values from a WAV file, to 2) compute the DFT with magnitude, and to 3) plot a spectrum. These functions will be used repeatedly. Reuse software from prior labs as a starting point.

Use your software to display the spectrum of the cos_1khz_pulse_20msec.wav file. This signal has a 1kHz sinusoid that has been time limited via multiplication by a 20 mSec pulse.

- 1a) Include a plot of the magnitude spectrum for the cos_1khz_pulse_20msec signal (2)
- 1b) What is the width of each sidelobe observed in the spectrum? How does it compare to the theoretical value of 1/(pulse duration) = 1/(20mSec)? (2)
- 1c) Documentation: The Python code that implements the DFT (8)

2) Simulate a Measured Signal Write a Python program that generates a WAV file. The signal you generate should approximate the impulse of sound in one of the measurement recordings (e.g, "tile1a.wav"). See Figures 1a and 1b. Use a sinusoid at 1kHz and an appropriate the envelope with an exponential function. Create a signal that is 1 sec long, with a sample rate of 8kHz. Note that the exponential pulse has an effective duration of around 0.1 Sec. Use your DFT program to plot the magnitude spectrum (as before N = 4000, plot up through 2kHz, skip DC).

- 2a) Include a plot of the magnitude spectrum for simulated measurement signal (4)
- 2b) Include a plot of the magnitude spectrum for a version of the simulated signal that is delayed by 0.1 Sec (4)

3) Measure the Size of the Air Gap Use your DFT program, plotting routine and peak finder to determine the resonant frequency of the air gap, and compute the dimension of the cavity. Process each of the given signals (five for each of two tiles).

tile1a.wav	tile1b.wav	tile1c.wav	tile1d.wav	tile1e.wav
tile2a.wav	tile2b.wav	tile2c.wav	tile2d.wav	tile2e.wav

For each of these 10 cases, document the following.

- 3a) Plot of the magnitude spectrum (1)
- 3b) Frequency of the "raw" peak in Hz (1)
- 3c) Frequency of the peak found via the weighted averaging method (1)
- 3d) Measured air gap (using weighted average) in inches (1)

Note the ceramic floor tiles are 12" x 20". (These dimensions limit the size of the air gap!)

Appendix – Example Code for Plotting with MatPlotLib and Numpy

```
# pip3 install matplotlib
import matplotlib.pyplot as plt

# pip3 install numpy
import numpy as np

# …
    # matplotlib mimics the plotting environment of MatLab. Plots can be displayed like this
    #    given two lists of data for x and y:   xpoint_list and ypoint_list

    fig, ax = plt.subplots()
    ax.plot(np.array(xpoint_list), np.array(ypoint_list))

    ax.set(xlabel='Frequency (Hz)', ylabel='Counts', title='Cool Plot! ')
    ax.grid()

    fig.savefig('image_file.png')
    plt.show()
```