

PHSX 444: Lab 02; Fourier Transforms

William Jardee

November 8, 2021

1 Introduction

One of the neatest characteristics of orthogonal bases is how they can be used to recreate a wide range of functions. Probably the most notable of all these bases is the Fourier transform. As can be shown through a relatively simple, but clever, derivation, all functions can be written as:

$$Y(t) = \int_{-\infty}^{\infty} y(\omega) \frac{e^{-i\omega t}}{\sqrt{2\pi}} dt$$
$$y(t) = \int_{-\infty}^{\infty} Y(\omega) \frac{e^{-i\omega t}}{\sqrt{2\pi}} dt$$

This isn't all too useful in application, however. There are two issues here. The first requiring an exact knowledge of the function to begin with, either in the frequency or time domain. The second being the computation time of finding such a solution to the equation (or the discrete version that we will see).

To solve the first issue, the Fourier Transform can be rewritten as the Discrete Fourier Transform:

$$Y(t) = \sum_{k=0}^{N-1} \Delta t \cdot \frac{e^{i\omega_n t_k}}{\sqrt{2\pi}} y_k$$
$$y(\omega) = \sum_{n=0}^{N-1} \frac{2\pi}{N\Delta t} \cdot \frac{e^{i\omega_n t_k}}{\sqrt{2\pi}} Y_n$$

Where Δt is the time step collected by the data. Now it is possible to translate the time domain of a function to the frequency domain with discrete description of the data. Doing this calculation, especially on the fly, is computationally intensive, requiring $\Omega(N^2)$ time. To beat this, the Fast Fourier Transform (FFT) can be used. A great explanation has been given by Reducible on youtube: [?], which goes into depth on the various steps in implementing the FFT. In order to use this method, there is one stipulation that must be followed: the time must be gathered as a multiple of 2 (i.e. 2, 4, 8, 16, ...).

While the theory behind FFTs is awe-inspiring, it is not at all required for the physicists to implement it. Numpy, the ever-present Python module, offers a weldable version of the algorithm for the layman.

It should be noted that when data is being collected to analyze the frequency domain of a function, there are a couple practical limits that must be worried about. If the measured time step is Δt , and the total duration of measurements is t_{\max} , then the frequency step that can be analyzed, $\Delta f = \frac{1}{t_{\max}}$ and the max frequency that information can be resolved for is $f_{\max} = \frac{1}{2\Delta t}$. This max frequency is also called the Niquist Frequency. Here the trade offs of quick measurements and quality data have to be analyzed. This is a common trade off that shouldn't at all be surprising to see.

2 Experimental Methods

The purpose of this lab was to get hands on experience with Fourier Analysis. The first part of the lab was just to collect data outputted from a function generator and in real time analyze it with a spectrum analyzer. The same output was then also collected by an oscilloscope to be analyzed with an FFT method outside of the lab time. In theory, the data collect by both methods should be equivalent. The output of the function generator can also be compared for different settings as well, seeing how a live analysis compares to one after the fact. On such setting is the window function. If a function being analyzed doesn't look periodic over the realm being measured, it can be manipulate to look pseudo-period. One such method is to use the Hanning Window, which is multiplying by a \sin^2 that has a period equal to the measurement window.

To test the ability to find a buried frequency in data, a module fittingly called "Buried Treasure" was provided. There were four different settings that offered a weak sine wave buried in noise. This offered a challenge to underline how to run the interface of the equipment effectively and show the importance of averaging data; a lesson that has been learned.

To test practical understanding of the Fourier Methods, an LRC circuit will be analyzed. Measurement of the response function will be carried out with four different methods, and they should all give the same response. We can state this because all four are measuring the same physical system and the only deviation should be because of faulted data collection or mathematical methods. For a general LRC circuit, as can be seen in Figure ??, the expected response is

$$X_C = \frac{1}{\omega C} \quad , \quad X_L = \omega L \quad , \quad \omega = \frac{1}{\sqrt{LC}}$$

$$Z = \sqrt{R^2 + (X_L - X_C)^2} \tag{1}$$

$$\text{Phase} = \phi = \tan^{-1} \left[\frac{X_L - X_C}{R} \right] \tag{2}$$

Where R is the resistor's resistance, C is the capacitor's capacitance, and L is the inductor's inductance. Z is the effect impedance of the three elements together, and ϕ is the phase shift that oscillatory inputs experience from the system.

The four methods that will be used to measure will be:

1. Using a single-frequency drive as an input and an oscilloscope and lock-in amplifier to measure the output at several frequencies
2. Using a noise drive as the input and the spectrum analyzer to measure the output.

3. Using an Arduino-based network analyzer to sweep over a wide range of frequencies with fine steps to get the amplitude and phase response.
4. Measuring the transient response using a square wave and applying a FFT to get the transfer function.

For the first method, a constant Sine wave was put into the LRC circuit, ranging from 4.1kHz to 5.5kHz with a step size of 0.2kHz. The Sine wave was provided by the function generator and the Sine and Cosine amplitudes were measured using the Lock-In Amplifier. The amplitude of the total response was also measured by an oscilloscope at the same time to ensure the data read by the lock-in amplifier was reliable (all the data can be see in Table ?? in Appendix A).

For the second method, a noise drive was taken from the Teach Spin Fourier Methods Electronic Modules system provided in lab. The FFT was then observed in the spectrum analyzer.

For the third method, a wide range of frequencies were provided by an Arduino system, and the response was measured by the same Arduino. This system was provided as a “black box” that we trust worked, as the data we observed seemed to be reasonable. The Arduino sampled the phase response, both in the Sine and Cosine orientation, and the phase shift. For our trials, the system swept from 1kHz to 10kHz with 100 steps.

The fourth and final method fed the LRC circuit a square wave from the function generator and measured the amplitude and phase of the response by first snapshot-ing the the output with an oscilloscope and applying an FFT to the data post-lab. The square wave, when focused on one half of a period, looks like a step function, thus giving a good understanding of the transient response.

The same four methods applied to measure the response function of the LRC circuit were also applied to an acoustical cavity. The acoustical cavity was set up so that a speaker pushed the sound into the bottom of the cylindrical container, and a microphone picked up the volume of the output on the top of the container. The microphone was plugged into the Teach Spin Fourier Methods Electronic Modules system and then fed into the appropriate analysis device; oscilloscope, lock-in amplifier, spectrum analyzer, or arduino system. The acoustical cavity had four possible heights: 3cm, 4cm, 5cm, and 6cm. Each of the four were analyzed.

Tubes have resonance frequencies where the nodes of the sound waves fit well in the tube. If the tubes don’t have a wavelength that allows a node to find purchase on the opposite wall, some of the returning waves will destructively interfere with those coming from the speaker. However, when a resonance frequency is achieved, a sharp jump in the surviving amplitude can be observed. Similarly, when nearly all the reflected waves destructively interfere, the amplitude should drop to nearly zero.

For the first measurement method, measurements were taken from 1.3kHz to 4.3kHz with a step size of 0.3kHz. The noise drive was again used for the second method. The third method swept over a frequency of 0kHz (this was an option available for the Arduino, however it is hard to decipher what this data point actually means) to 10kHz with a step size of 500Hz and a drive amplitude of 0.95V. There were two sweeps taken to average the results. The same method to measure the transient response was used.



Figure 1: The LRC circuit used in the second part of this lab (<https://www.circuit-diagram.org/editor/>)

To save some time, only the 6cm cavity was used to measure the first method. The second and fourth methods used all four cavity sizes. Data was taken for the third method in all three cavity sizes; however, the data for the 3cm cavity was somehow lost in the process and will be excluded from the analysis.

3 Results and Analysis

This lab has been split up into the three different stages as explained above. The first outlines the first experiences with the spectrum analyzer and manually applying the FFT to the time domain in order to achieve the expected results. The second part goes over the LRC response function. The third goes over the acoustical cavity response function.

3.1 General Fourier Introduction

It can be seen that the difference between the uniform window and hanning window in Figure ?? and Figure ?? isn't very noticeable. However, when applied to a frequency that wasn't as ideal, such as in Figure ?? and Figure ??, the hanning correction made the FFT of the 1.03kHz frequency (Figure ?? look comparable to that of the 1kHz frequency (Figure ??.

It seems that using the hanning window on anything, other than the ideal Sine Wave, returned a more reasonable analysis; for both the spectrum analyzer and the oscilloscope. It

can be seen that the FFT of the less ideal frequency has a wider spread of possible frequency peak. This should serve as the practical average and standard deviation of the result; taking the highest peak as the average and the HMF_W to be the standard deviation.

To test the impact of different windows on the FFT, five different windows were tested on a square wave (Figure ??). While this doesn't give any enlightenment as to where each shines, the largest take away is that the four window functions that are not the uniform window effectively grounded the noise introduced to the FFT by a non-periodic function.

The final test to see if we understood Fourier Analysis was to find the treasure buried in the noise provided. The tests were labeled A, B, C, and D. The first three were attempted, and only the first two could be found within the time restraint. On setting A, once the data was averaged over a large number of passes, about 1000, the spectrum analyzer seemed to give a definitive spike (Figure ??). The oscilloscope collected data with an acquire time about the same as that used for the spectrum analyzer, 32ms, and averaged over 128 passes. Even with the large amount of data collection, the peak is barely visible. On setting B (Figure ??), with even more passes averaged and an acquire time of 8ms, the spectrum analyzer could barely pick out the peak and nothing was found from the FFT done post lab. On setting C (Figure ??), the noise was so dominant nothing was able to be picked out.

3.2 LRC Analysis

3.3 Acoustical Cavity Analysis

4 Discussion and Conclusions

Appendix A

| Freq | X | Y | Amp |
|------|--------|--------|------|
| 4.1 | 183.69 | 29.24 | 560 |
| 4.3 | 197.54 | 37.87 | 600 |
| 4.5 | 213.8 | 49.68 | 650 |
| 4.7 | 232.8 | 66.59 | 720 |
| 4.9 | 254.3 | 90.76 | 790 |
| 5.1 | 277.2 | 126.8 | 880 |
| 5.3 | 297.2 | 180.24 | 1010 |
| 5.5 | 303.3 | 257.2 | 1140 |
| 5.5 | 303.7 | 256.4 | 1140 |

Table 1: sin lock-in/osci LRC

../images/lab02/1 kHz Sine Wave (uniform).png

(a)

../images/lab02/1 kHz Square Wave (uniform).png

(a)

../images/lab02/1 kHz Triangle Wave (uniform).png

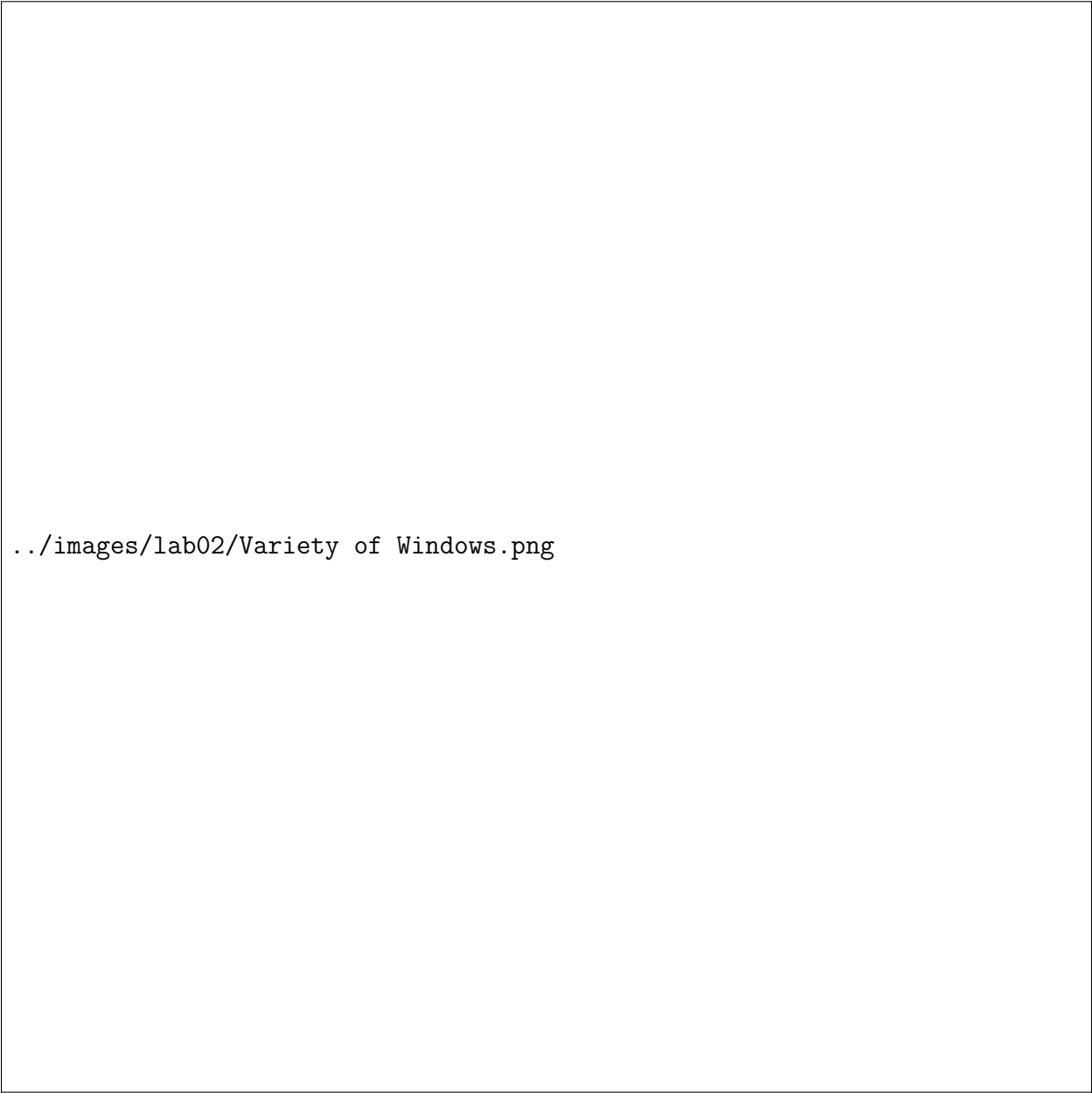
(a)

../images/lab02/1 kHz and 1_03 kHz Sine Wave Sum (uniform).png

(a)

../images/lab02/1 kHz and 1_03 kHz Sine Wave Product (uniform).png

(a)




`../images/lab02/Variety of Windows.png`

Figure 7

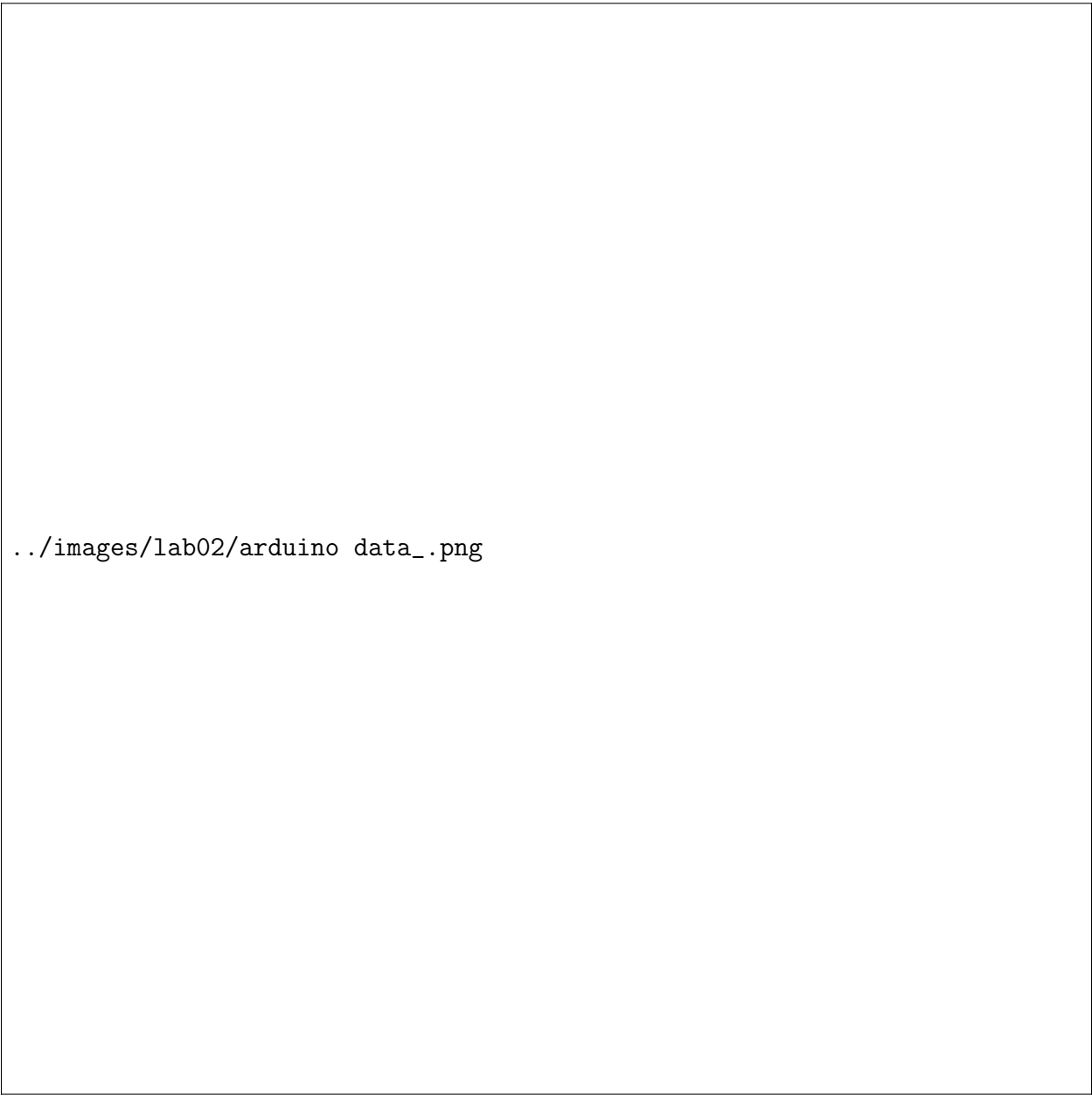
../images/lab02/Buried Treasure Setting A (uniform).png

(a)



../images/lab02/Noise Drive into LRC Filter (uniform).png

Figure 9



`../images/lab02/arduino data_.png`

Figure 10