



University of Idaho

ECE 351 - SECTION #53

Lab #9 Report

FAST FOURIER TRANSFORM

Submitted To:

Kate Antonov
University of Idaho
kantonov@uidaho.edu

Submitted By :

Macallyster S. Edmondson
University of Idaho
edmo7033@vandals.uidaho.edu
github.com/mac-edmondson

29th March, 2022

Contents

1	Introduction	2
2	Equations	2
3	Methodology	2
	3.1 Lab: Part 1	2
4	Results	3
5	Error Analysis	11
6	Questions	11
7	Conclusion	11
8	Attachments	12

1 Introduction

The goal of this weeks lab was to become famaliar with the Fast Fourier Transform by implementing it in Python. This lab was extremely helpful for my understanding of the related class topics. This lab was completed using *Python* through the *Spyder-IDE*. The packages used in the completion of this lab were **numpy** for definitions of mathematical functions, **matplotlib.pyplot** to plot outputs of functions, and **scipy.fftpack** to perform the Fast Fourier Transform calculations within the lab.

All code for this lab, including this report, can be found on my Github.

2 Equations

The equations used within this lab are shown in this section. The equations will be referenced by number throughout the rest of the report.

General Fourier Series Equations/Formulas:

$$x(t) = \cos(2\pi t) \quad (1)$$

$$x(t) = 5 \sin(2\pi t) \quad (2)$$

$$x(t) = 2 \cos((2\pi \cdot 2t) - 2) + \sin^2((2\pi \cdot 6t) + 3) \quad (3)$$

$$x(t) = \sum_{k=1}^N N \frac{2}{k\pi} (1 - \cos(k\pi)) \sin(k\omega_0 t) \quad (4)$$

Fourier Series found from preliminary:

$$x(t) = \sum_{k=1}^N \frac{2}{k\pi} (1 - \cos(k\pi)) \sin(k\omega_0 t) \quad (5)$$

3 Methodology

3.1 Lab: Part 1

In Part 1 of this lab, we created a Fast Fourier Transform (FFT) function of our own to perform the FFT of all functions given in Equations (1) - (4). The FFT function I initially made can be seen below.

```

1 def myFFT(x, fs):
2     N = len(x)
3     X_fft = spfft.fft(x)
4     X_fft_shifted = spfft.fftshift(X_fft)
5 
```

```
6 freq = np.arange(-N/2, N/2)*fs/N
7 X_mag = np.abs(X_fft_shifted)/N
8 X_phi = np.angle(X_fft_shifted)
9
10 return freq, X_mag, X_phi
```

The FFT function was later modified to filter out elements with a magnitude `X_mag` $< 1e-10$, and set the corresponding angle `X_phi` = 0. This made the plots for each function readable. Examples are seen in the Results Section of this report. The modified function is seen below.

```
1 def myFFT(x, fs, clean=False):
2     N = len(x)
3     X_fft = spfft.fft(x)
4     X_fft_shifted = spfft.fftshift(X_fft)
5
6     freq = np.arange(-N/2, N/2)*fs/N
7     X_mag = np.abs(X_fft_shifted)/N
8     X_phi = np.angle(X_fft_shifted)
9     if(clean):
10         for i in range(0, len(X_phi)):
11             if (X_mag[i] < (1e-10)) :
12                 X_phi[i] = 0
13
14     return freq, X_mag, X_phi
```

4 Results

The results of this lab are very straightforward. The implementation of all functions worked as expected and the results are as expected. More analysis of theory and results is discussed in the Questions Section of this report.

The deliverables for Part 1 of this lab are seen in all figures given below. There are 2 figures for Equations (1) - (3), showing the difference between filtered and unfiltered results. The plot for Equation (4) is filtered. Each plot shows the function at the top, a zoomed out view of magnitude and angle of the FFT results, and a zoomed in view of the magnitude and angle of the FFT results.

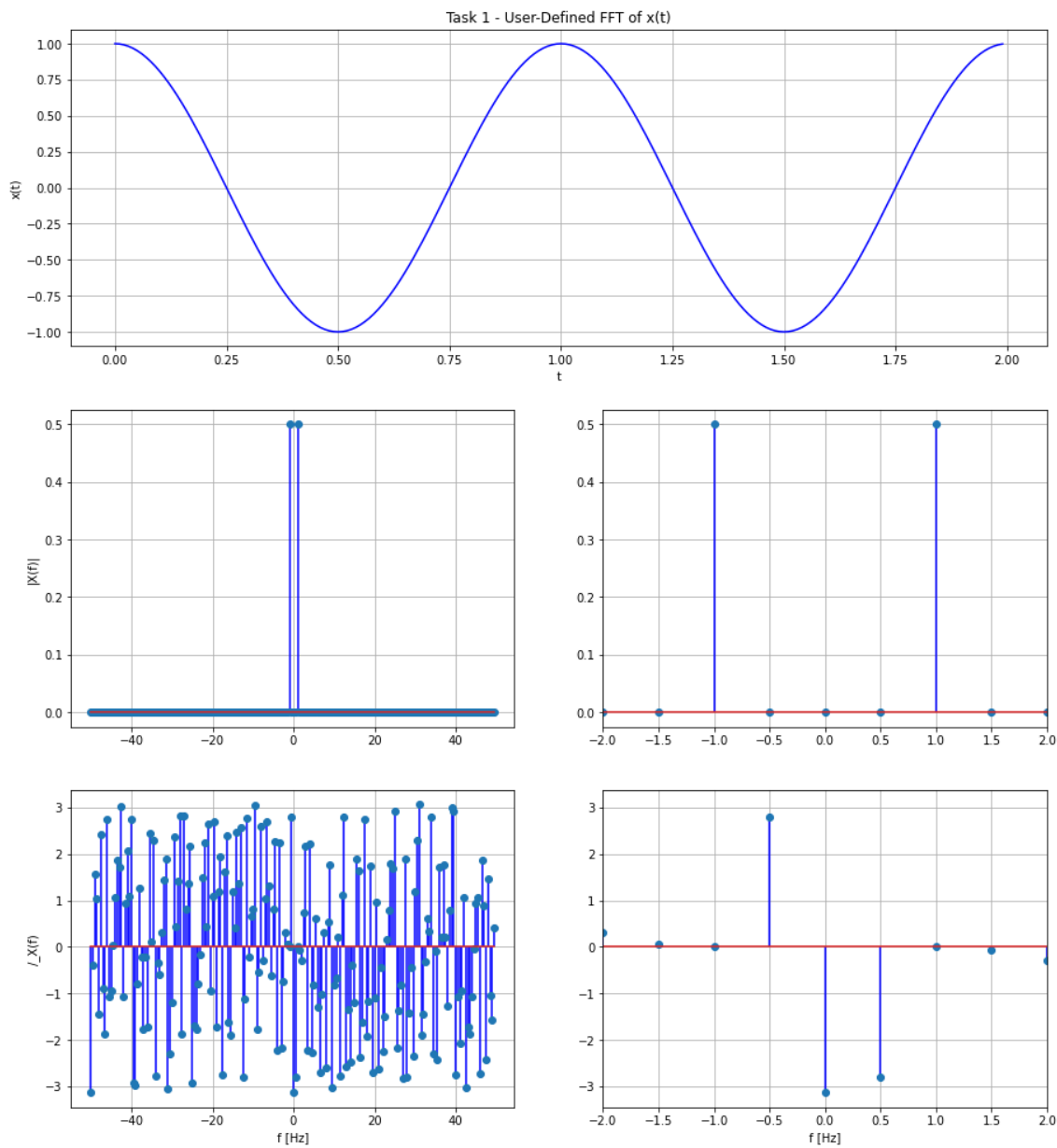


Figure 1: Function and FFT Results of Equation (1)

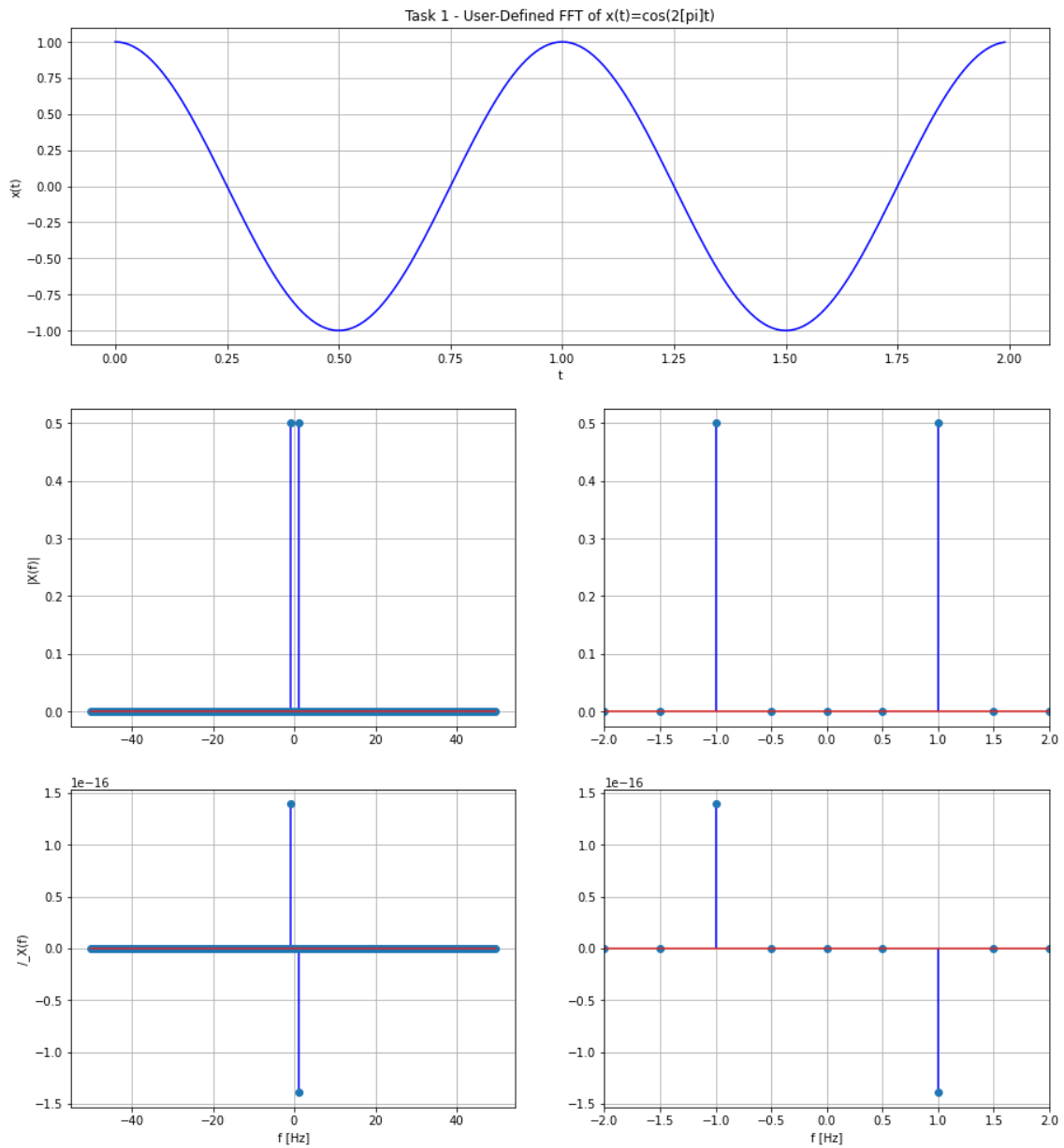


Figure 2: Function and FFT Results of Equation (1) (Filtered)

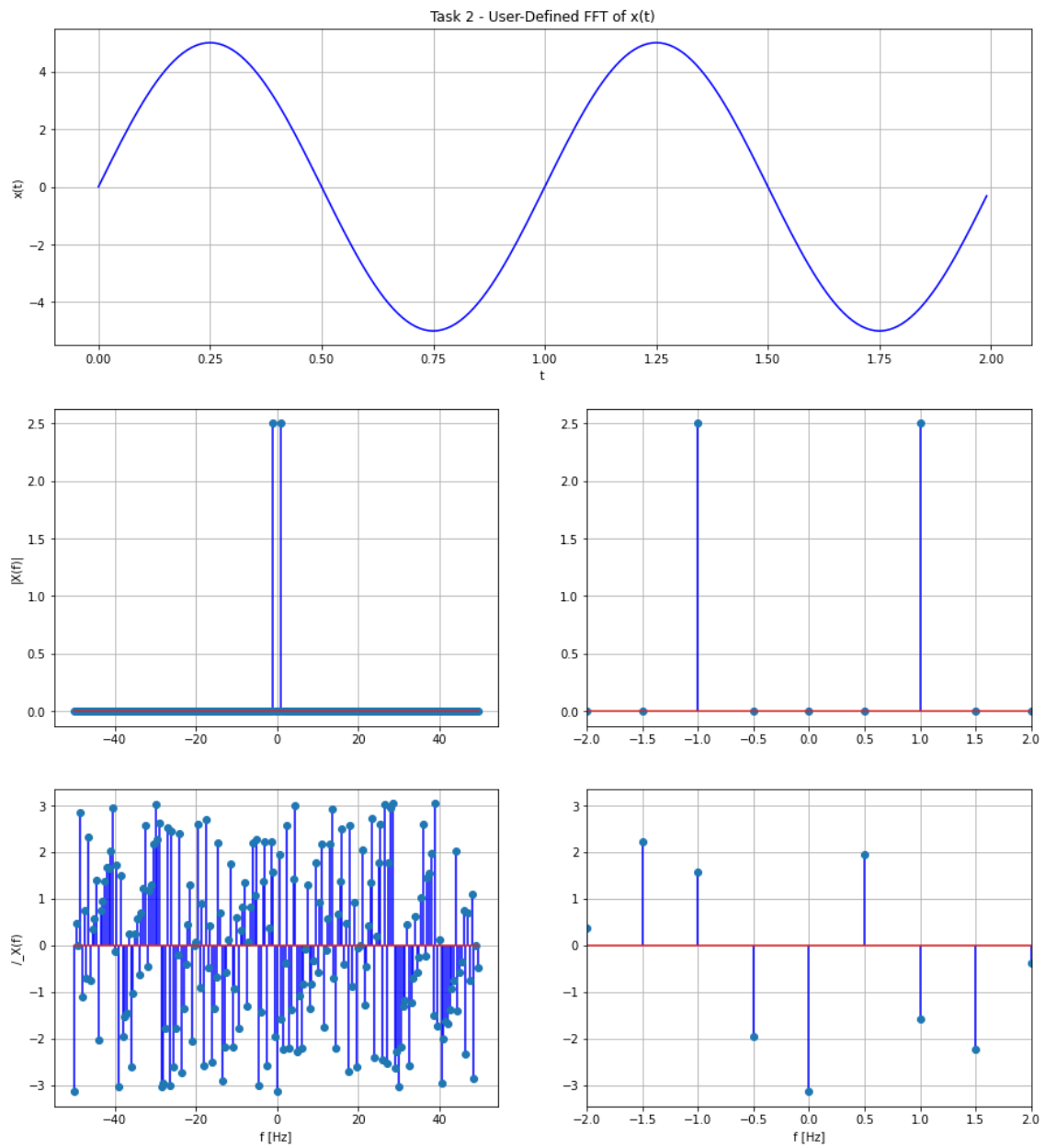


Figure 3: Function and FFT Results of Equation (2)

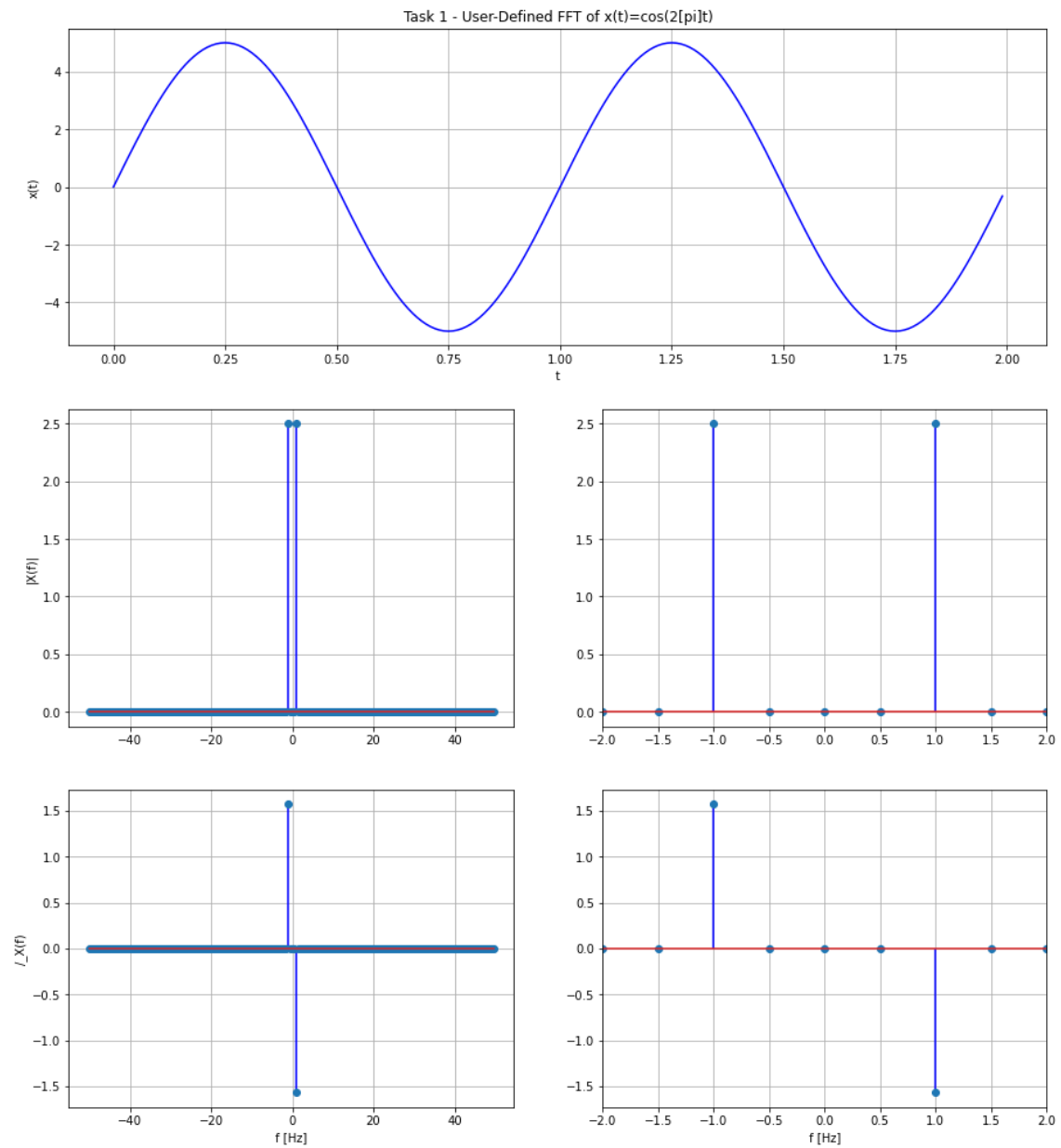


Figure 4: Function and FFT Results of Equation (2) (Filtered)

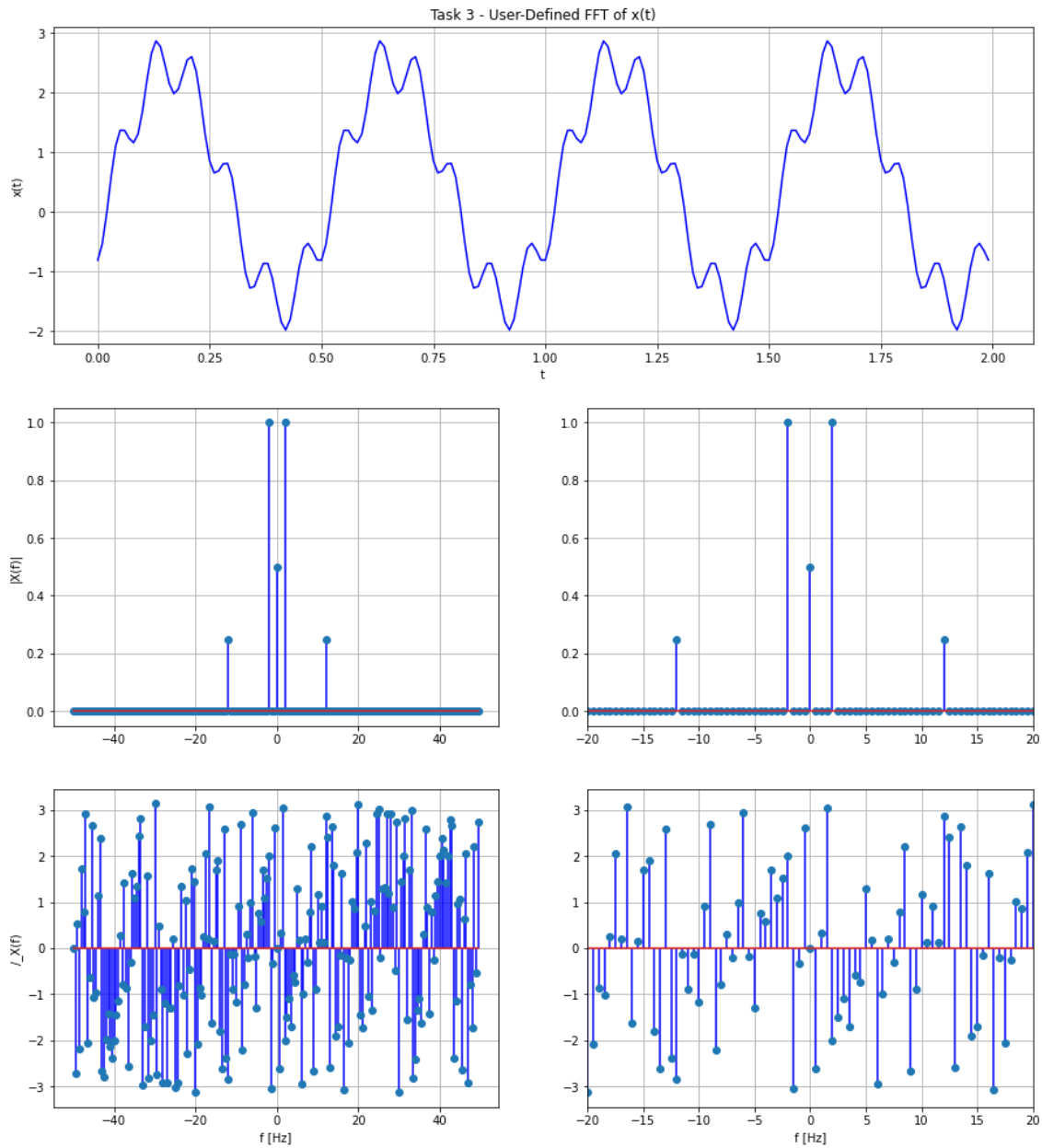


Figure 5: Function and FFT Results of Equation (3)

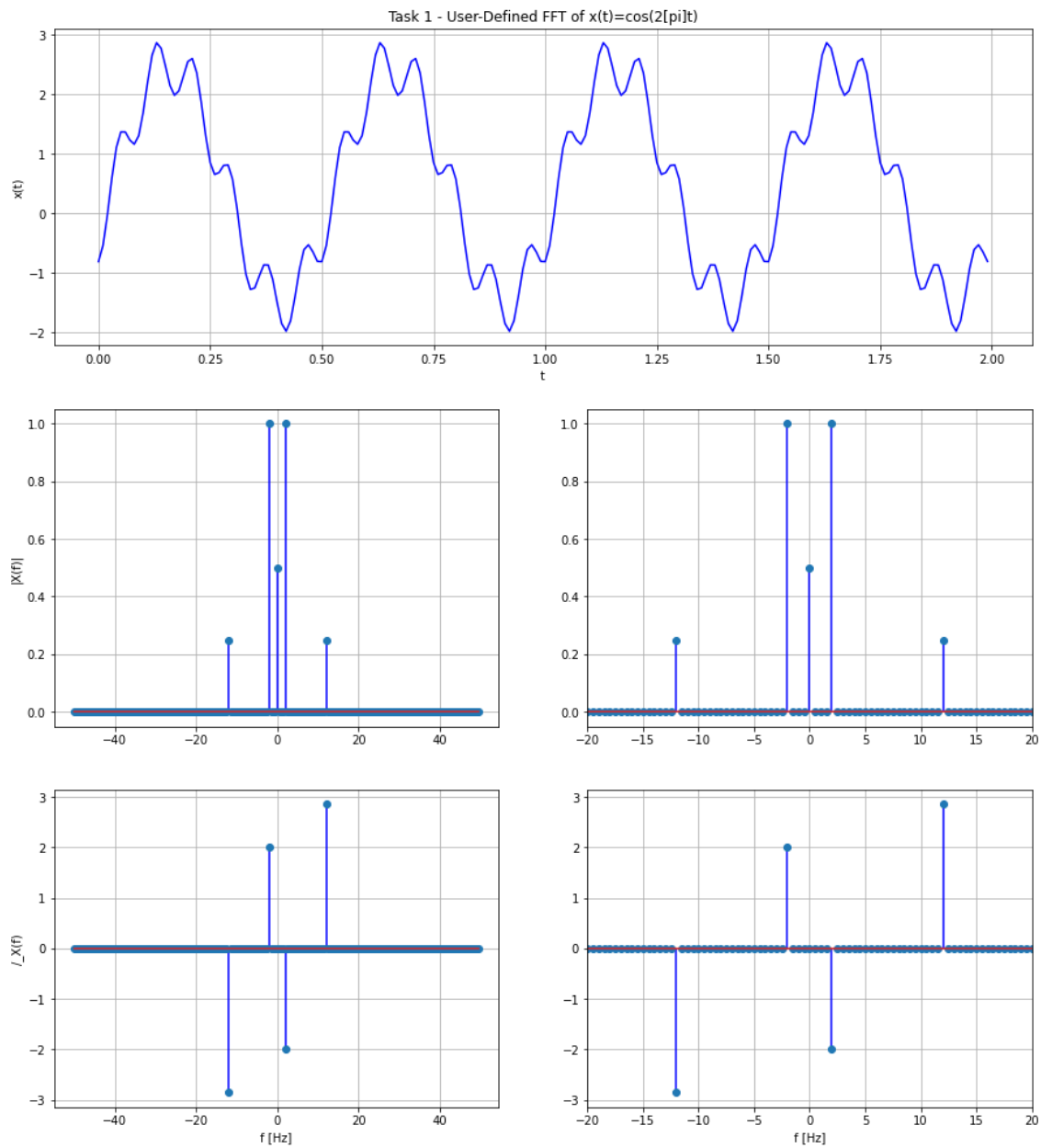


Figure 6: Function and FFT Results of Equation (3) (Filtered)

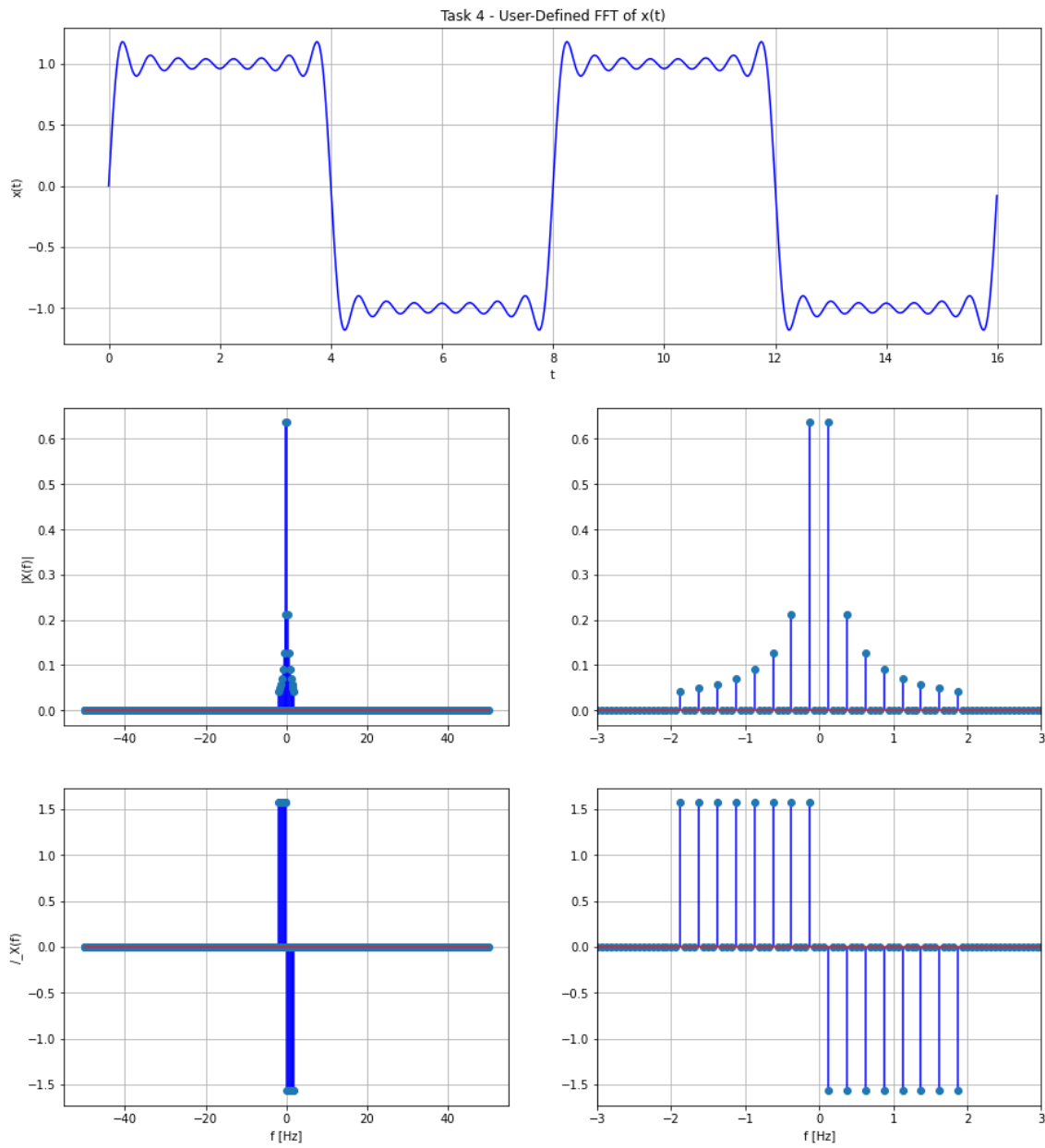


Figure 7: Function and FFT Results of Equation (4) (Filtered)

5 Error Analysis

One consideration in sources of error in this lab is the Fast Fourier Transform itself. As the name might imply, this is an approximation algorithm to find the fourier transform of a function and is not exact as compared to the Theoretical Fourier Transform of a function. See Questions Section for further discussion (Question 3).1

6 Questions

1. What happens if **fs** is lower? If it is higher?
 - As **fs** is the the sampling frequency, the greater it is the more resolution there will be in the output of the FFT. Thus, if it is lower the resolution would be lower. The more samples you have the more accurate your results should be, so a very complicated function may benefit from a higher **fs** value.
2. What difference does eliminating the small phase magnitude make?
 - When the small phase magnitude is eliminated (**x_mag** < 10e-10), it is like filtering the results to only view what is important. As the angle for the very small magnitude terms exists, it is still shown on the angle plots. But, as the magnitude of of majority of the terms is insignificant the angle of the final function will have an insignificant amount of attenuation from terms with such a small magnitude.
3. Verify your results from **Tasks 1 and 2** using the Fourier transforms of cosine and sine. Explain why your results are correct.
 - Fourier Transform in Hz of **Cosine**: $\frac{1}{2}[\delta(f+f_0)+\delta(f-f_0)]$;**Sine**: $\frac{j}{2}[\delta(f+f_0)-\delta(f-f_0)]$. We can see the magnitude and angle will match Plots 2 & 4 when substituting in Equations (1) & (2).
4. Leave any feedback on the clarity of lab tasks, expectations, and deliverables.
 - This lab was very clear for all instructions, expectations and deliverables.

7 Conclusion

In conclusion, I feel this lab was very successful. It would have been nice if the lab talked further about how FFT theory, but this is something students can always look into themselves. All in all, I am very satisfied with what this lab has taught me and feel it was an excellent use of time.

8 Attachments

No attachments for this lab.