ECE 351-51

MARCH 29, 2022

# Lab 9

*Submitted By :*
Kennedy Beach

# Contents


1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2 Equations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
3 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
4 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
5 Error Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
6 Questions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
7 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

# 1 Introduction

The purpose of this lab was to use Python to plot signals and the magnitude and phases of those signals using a user-defined Fast Fourier Transform. Most of the function was given.

# 2 Equations

The signals to be plotted are shown below:

$$x_1 = cos(2\pi t)$$

$$x_2 = 5sin(2\pi t)$$

$$x_3 = 2cos((2\pi 2t) - 2) + sin^2((2\pi 6t) + 3)$$

The Fourier series developed in the last lab was also plotted where N = 15 and T = 8.

$$a_k = 0$$

$$b_k = \frac{2}{k\pi}(1 - cos(k\pi))$$

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

# 3 Methodology

The first part involved finishing the given Fast Fourier Transform function then plotting the given equations. The magnitude and phase were plotted with the results from the FFT function.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig
import scipy.fftpack

```

```
def fft(x,fs):
    N = len(x) # find the length of the signal
    X_fft = scipy.fftpack.fft(x) # perform the fast Fourier transform (fft)
    X_fft_shifted = scipy.fftpack.fftshift(X_fft) # shift zero frequency components
    # to the center of the spectrum
    freq = np.arange(-N/2, N/2)*fs/N # compute the frequencies for the output
    # signal , (fs is the sampling frequency and
    # needs to be defined previously in your code
    X_mag = np.abs(X_fft_shifted)/N # compute the magnitudes of the signal
    X_phi = np.angle(X_fft_shifted) # compute the phases of the signal

    return freq, X_mag, X_phi

fs = 100
T = 1/fs
t = np.arange(0, 2, T)

x1 = np.cos(2*np.pi*t)
x1_freq, x1_mag, x1_phi = fft(x1, fs)
```

Listing 1: FFT

A cleaner version of the FFT was then defined by making the phase zero for every magnitude that was less than 1e-10. The equations were re-plotted as well as the Fourier series from the last lab.

```
def clean_fft(x,fs):
    N = len(x) # find the length of the signal
    X_fft = scipy.fftpack.fft(x) # perform the fast Fourier transform (fft)
    X_fft_shifted = scipy.fftpack.fftshift(X_fft) # shift zero frequency components
    # to the center of the spectrum
    freq = np.arange(-N/2, N/2)*fs/N # compute the frequencies for the output
    # signal , (fs is the sampling frequency and
    # needs to be defined previously in your code
    X_mag = np.abs(X_fft_shifted)/N # compute the magnitudes of the signal
    X_phi = np.angle(X_fft_shifted) # compute the phases of the signal
    for i in range(len(X_phi)):
        if (np.abs(X_mag[i]) < 1e-10):
            X_phi[i] = 0
```

```python

    return freq, X_mag, X_phi
```

Listing 2: Clean FFT

# 4 Results

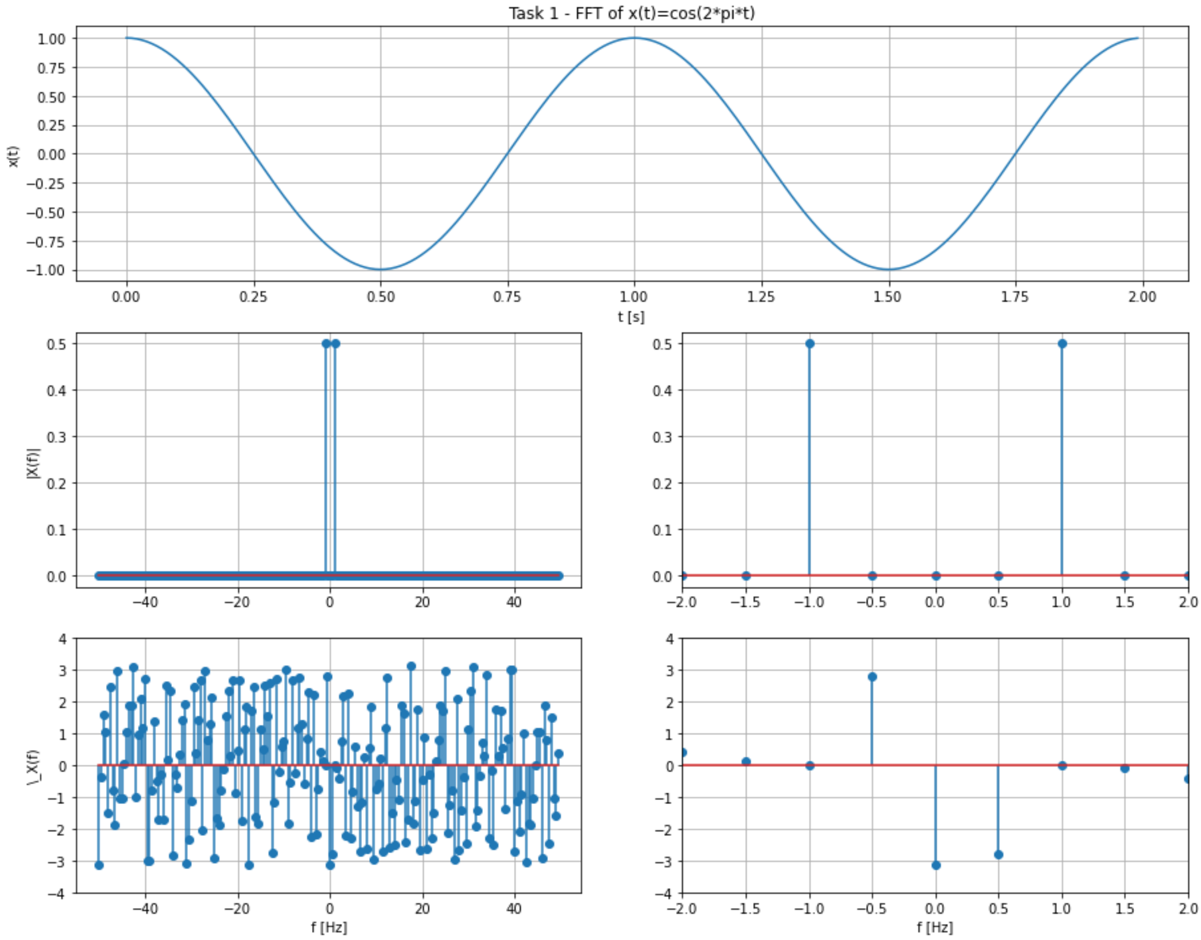


Figure 1: FFT of $x(t) = \cos(2\pi t)$

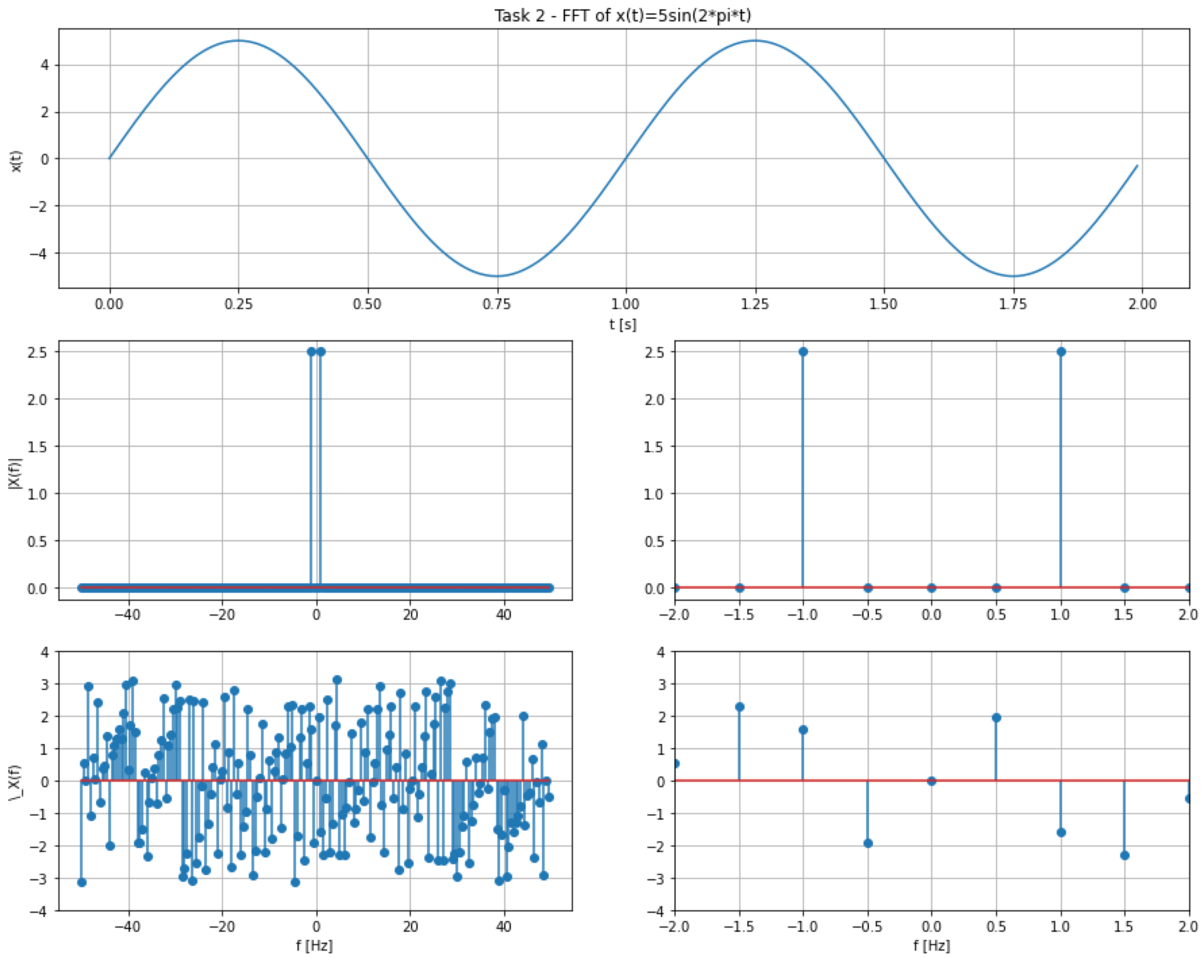


Figure 2: FFT of x(t) = 5sin(2$\pi$t)

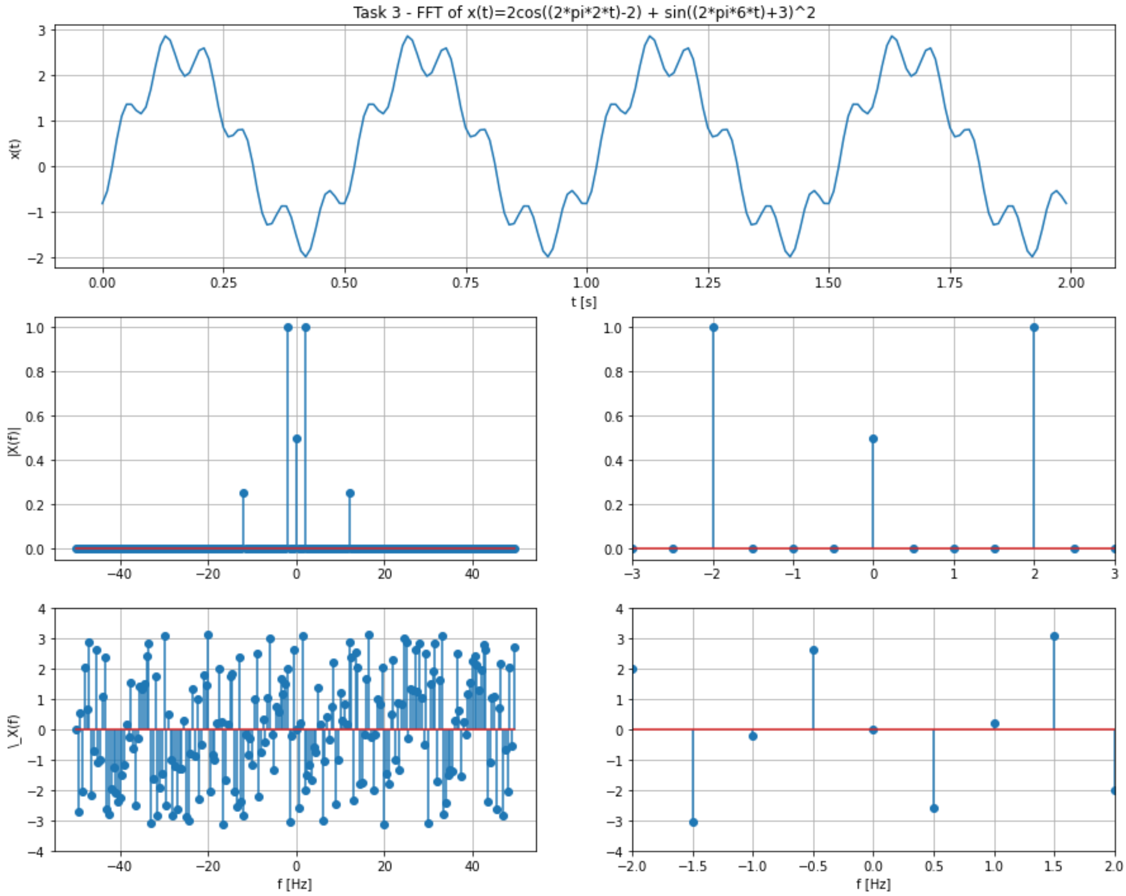


Figure 3: FFT of x(t) = 2cos((2$\pi$ 2t)-2) + sin$^2$((2$\pi$ 6t)+3)

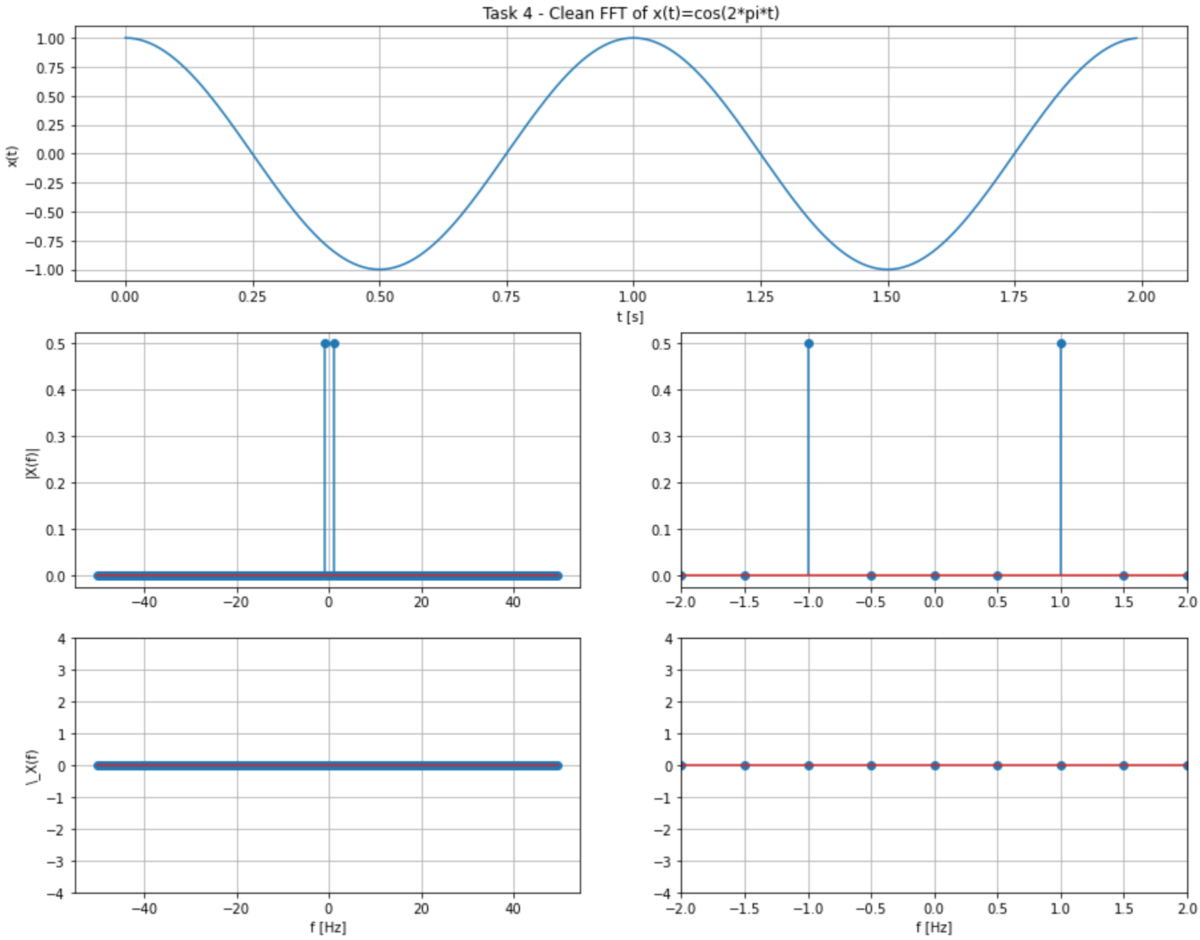


Figure 4: Clean FFT of x(t) = $\cos(2\pi t)$

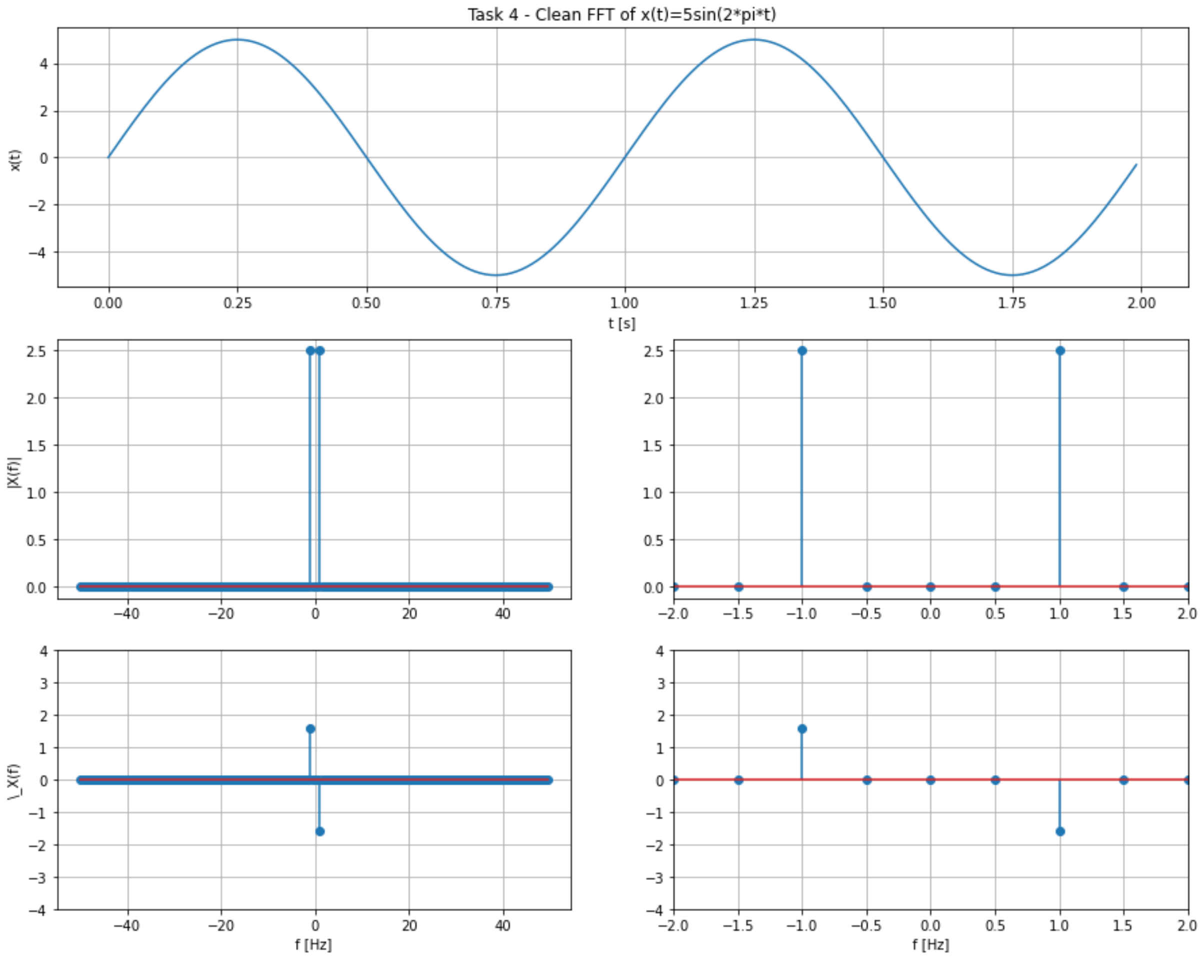


Figure 5: Clean FFT of x(t) = 5sin(2$\pi$t)

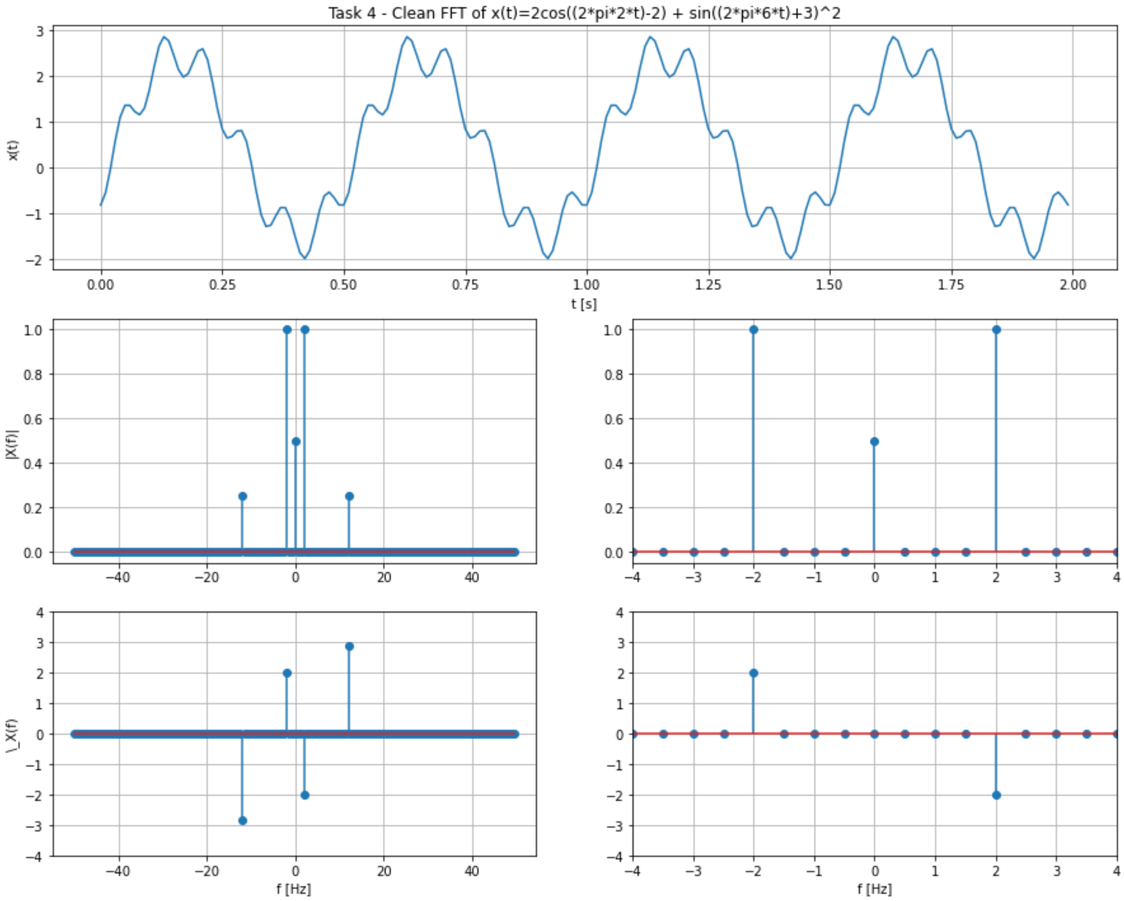


Figure 6: Clean FFT of x(t) = 2cos((2$\pi$ 2t)-2) + sin$^2$((2$\pi$ 6t)+3)

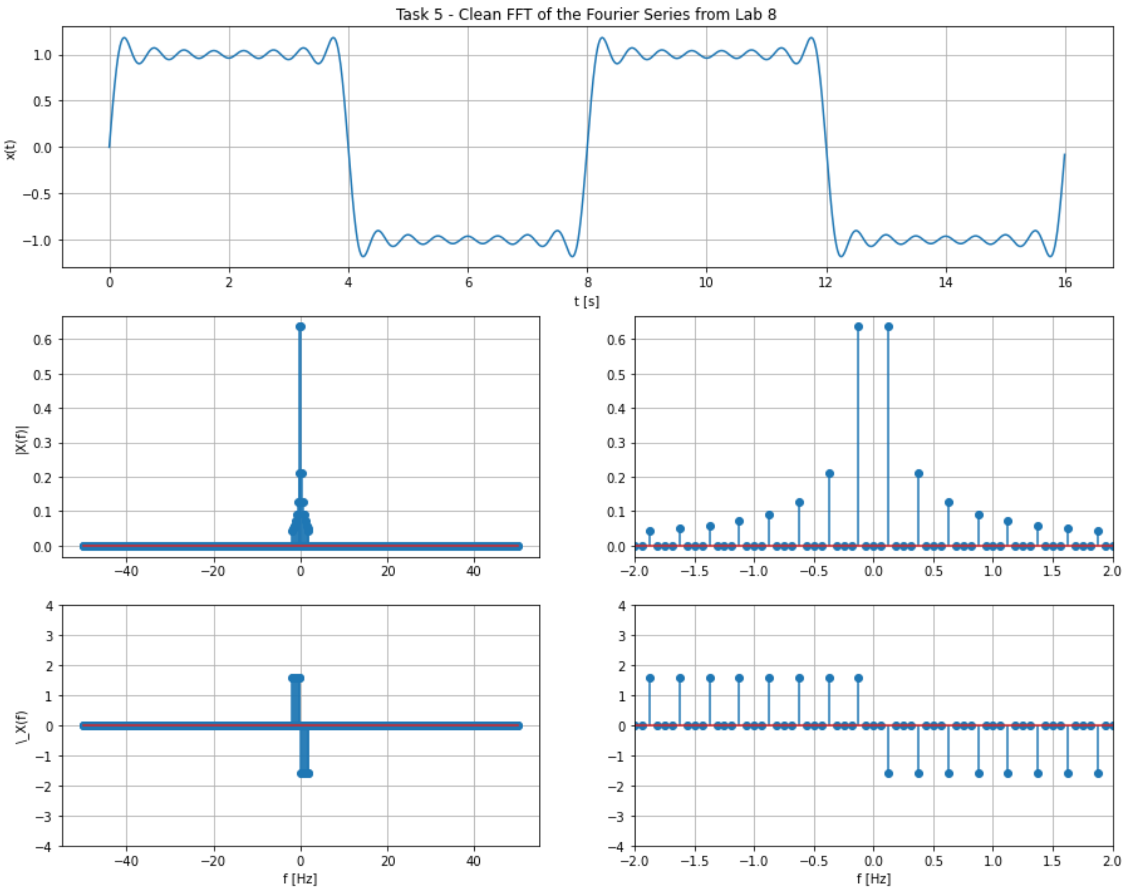


Figure 7: Clean FFT of Fourier series

# 5 Error Analysis

There weren't any errors besides some simple mistypes.

# 6 Questions

**1. What happens if fs is lower? If it is higher? fs in your report must span a few orders of magnitude.**

When fs is lower, there are fewer delta functions in the phase shift portion while there are more when fs is higher. At one point I tried fs = 100000 and the graph looked solid blue. The time to run the code also took longer when fs was higher.

**2. What difference does eliminating the small phase magnitudes make?**

The phase plots look simpler since there are fewer delta functions. Since the magnitudes were so small, the actual Fourier transform results aren't affected as much, despite the amount of delta functions that were eliminated.

**3. Verify your results from Tasks 1 and 2 using the Fourier transforms of cosine and sine. Explain why your results are correct. You will need the transforms in terms of Hz, not rad/s. For example, the Fourier transform of cosine (in Hz) is:**

$$\mathcal{F}\{cos(2\pi f_0 t)\} = \frac{1}{2}[\delta(f - f_0) + \delta(f + f_0)]$$

As seen above, the Fourier transform of the cosine function does not have any imaginary components. This corresponds with the phase shift, or lack of it, in the plots above. The Fourier transform of the sine function has a negative and positive delta imaginary component, which is seen in the phase shifts.

**4. Leave any feedback on the clarity of the expectations, instructions, and deliverables.**

Everything was clear on what needed to be done and turned in.

# 7 Conclusion

This lab showcased the magnitude and phase shifts of various signals using Fast Fourier Transforms. The Python and LaTeX code are seen in `https://github.com/`

Eniac618/ECE351_Code and https://github.com/Eniac618/ECE351_Reports respectively.