

Project Report for ECE 351

Lab 9: Fast Fourier Transform

Andrew Gibson

28 March 2023

https://github.com/gibs0630/ECE351_Code

https://github.com/gibs0630/ECE351_Reports

Contents

1	Introduction	1
2	Equations	1
3	Methodology	1
4	Results	2
5	Questions	5
6	Conclusion	6

1 Introduction

With Python, it is possible to apply a filter on a signal that has already been measured and stored. Being able to do post-processing is important for analysing important details in a signal.

2 Equations

Formula's used

Fourier's series

$$x(t) = \sum_{n=-\infty}^{\infty} \left[\frac{1}{T} X \left(n \frac{2\pi}{T} \right) e^{jn \frac{2\pi}{T} t} \right]$$

equations from the lab

$$H(s) = \frac{\frac{1}{RC}s}{s^2 + \frac{1}{RC}s + \frac{1}{LC}}$$

$$f_1(t) = \cos(2\pi * 100t) + \cos(2\pi * 3024t) + \sin(2\pi 850000t)$$

3 Methodology

This lab had us create bode plots of a transfer function and analyze their effect on a signal. We used Python to generate in the ω domain the frequency response and phase shift of a transfer function. Then we compared it with the outputs from a few libraries available.

4 Results

Part 1

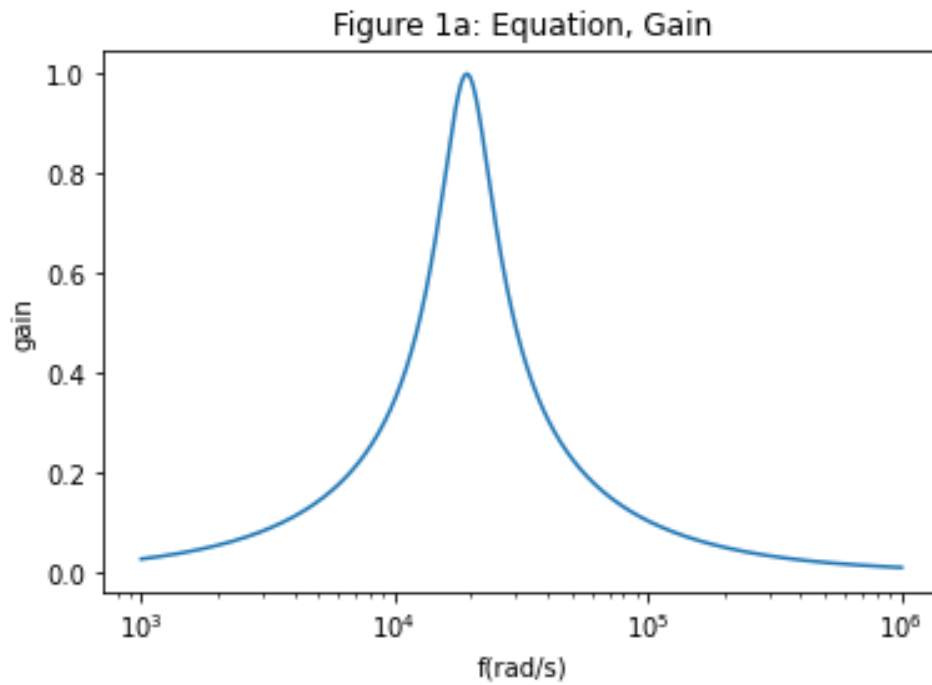


Figure 1a shows the magnitude of the transfer function for a given frequency using hand calculations.

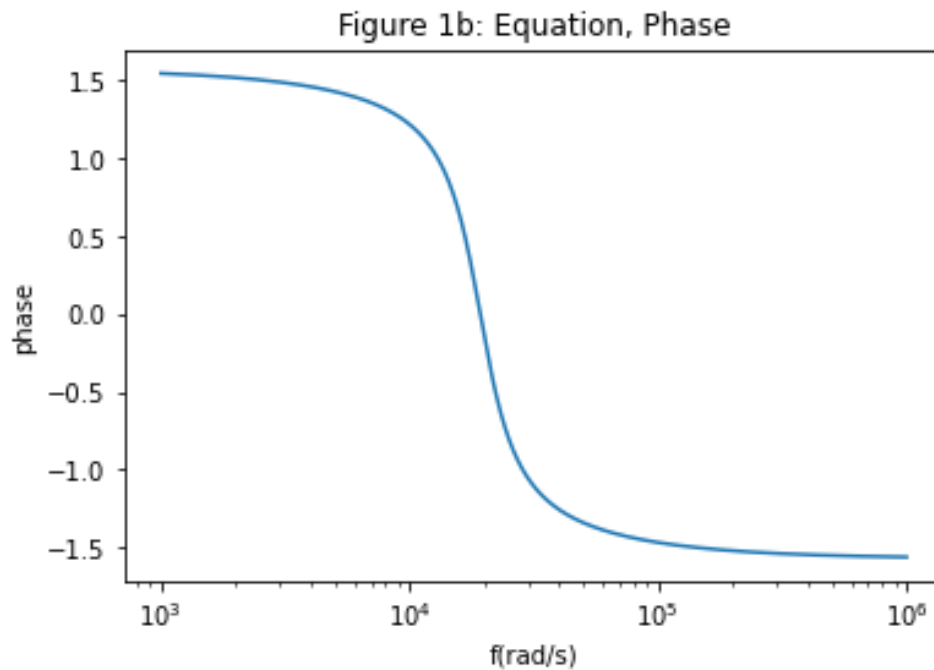


Figure 1b shows the phase of the transfer function for a given frequency using hand calculations.

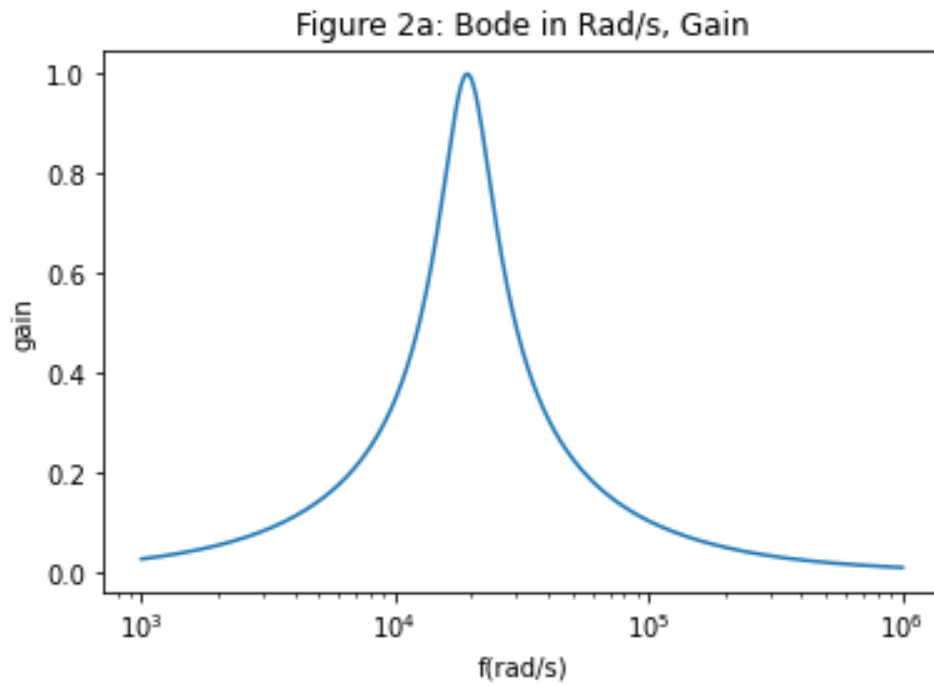


Figure 2a shows the magnitude of the transfer function for a given frequency using the `scipy.signal.bode` function.

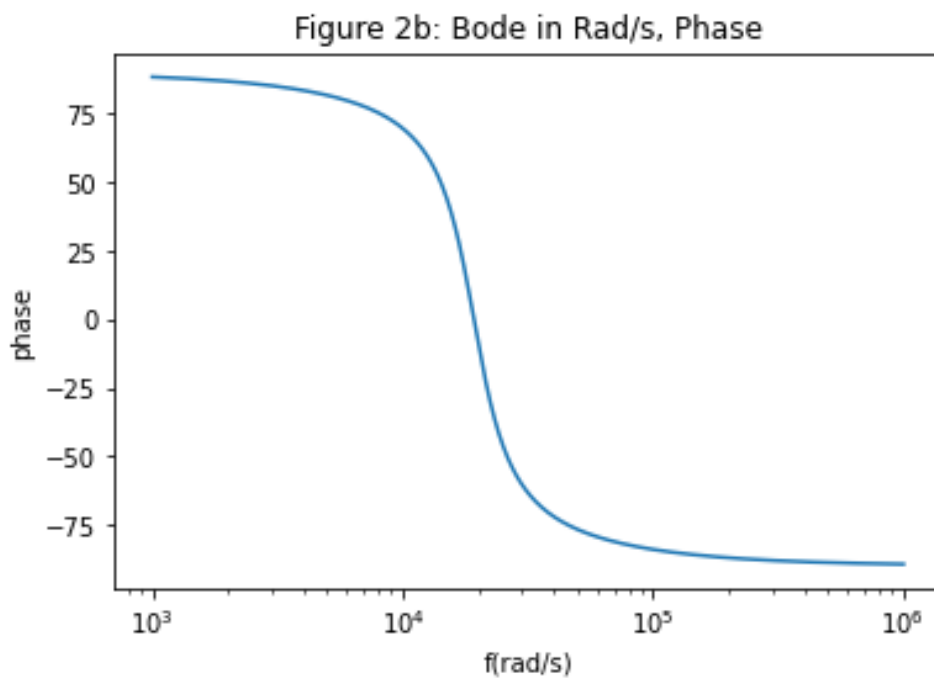


Figure 1b shows the phase of the transfer function for a given frequency using the `scipy.signal.bode` function.

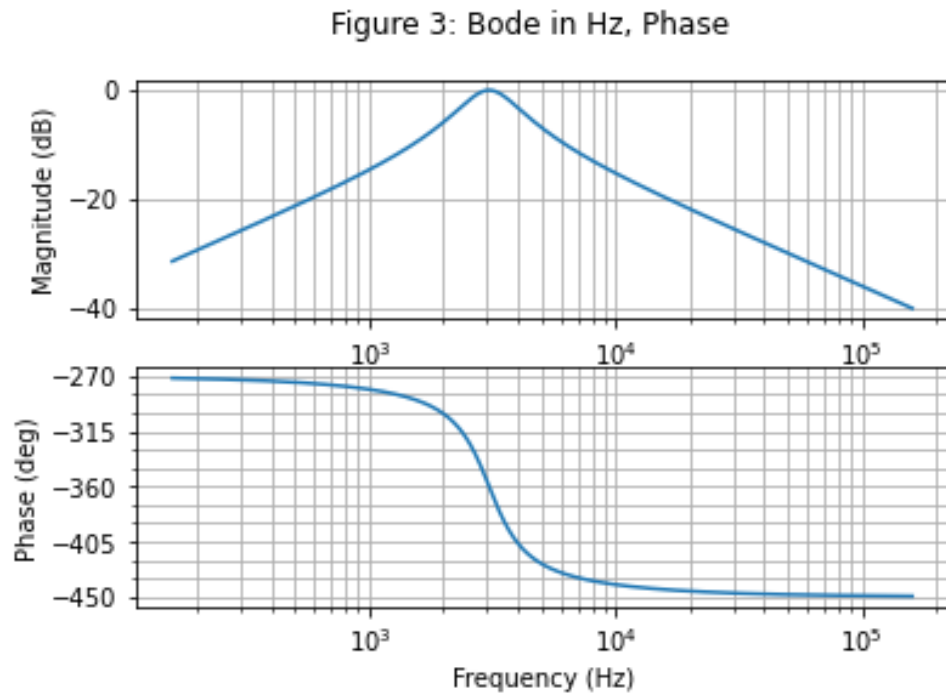


Figure 3 shows the bode plots of the transfer function for a given frequency using the `TransferFunction` function and the `control.bode` function. The magnitude here is in a scale of dB instead of linear.

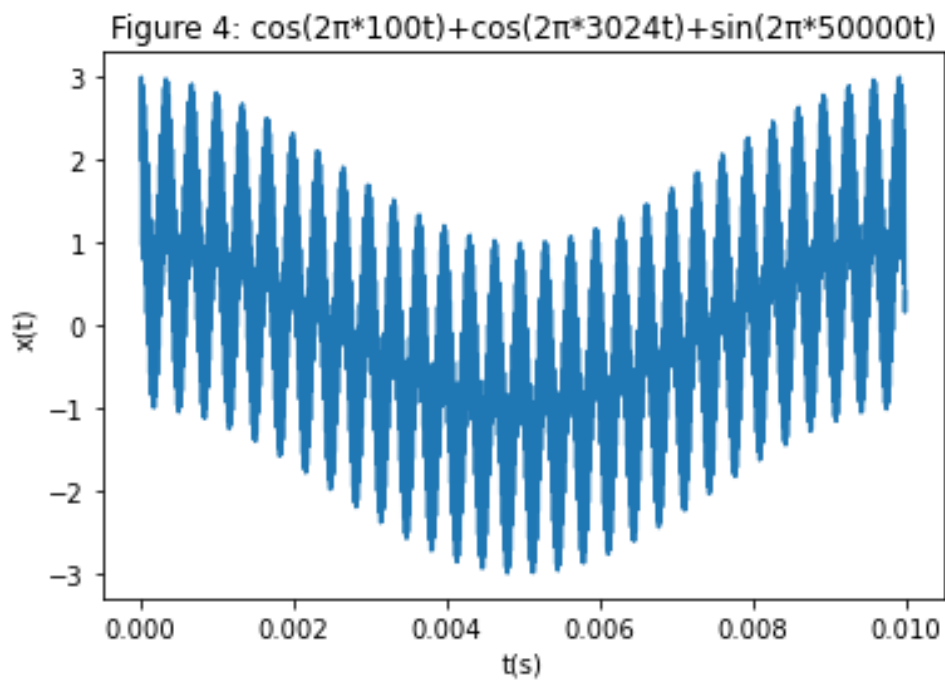


Figure 4 shows the time domain of a signal that will be passed through the transfer function.

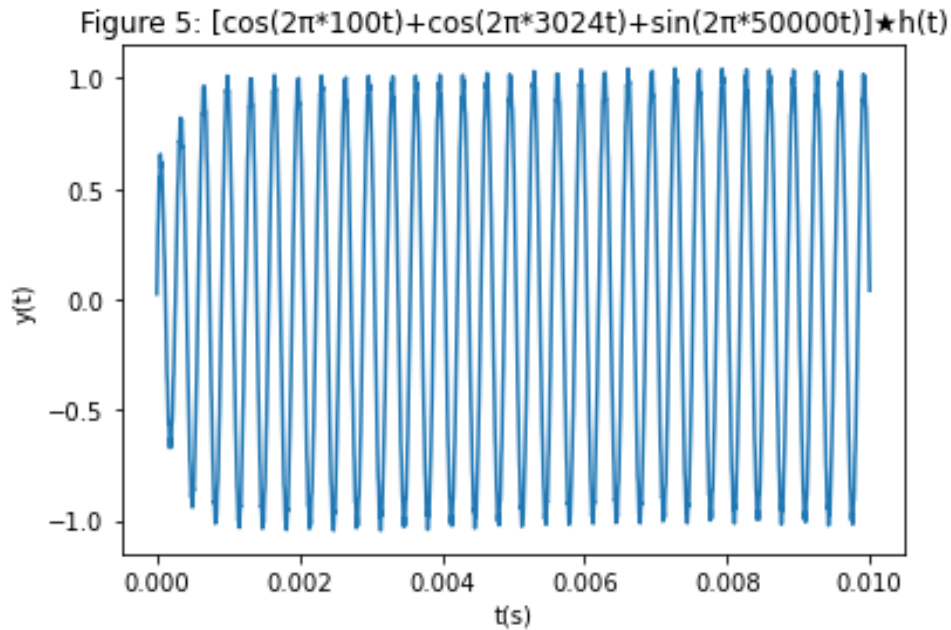


Figure 5 shows the time domain output; notice that the low frequency signal (which causes the largest wave) is not present, and the high frequency (which causes the sinusoidal to appear solid in color) is also not present. The start ramps up in amplitude because of an artifact from not having data-points before $t=0$.

5 Questions

Q: Explain how the filter and filtered output in Part 2 makes sense given the Bode plots from Part 1. Discuss how the filter modifies specific frequency bands, in Hz.

A: The filter makes sense when consulting the bode plot. In the original signal, there were three super-positioned sinusoidal with different frequencies. According to the bode plot for magnitude, both the 100Hz and 50000Hz would be attenuated, reducing them to -30 dB relative to the input signal.

Q: Discuss the purpose and workings of `scipy.signal.bilinear()` and `scipy.signal.lfilter()`

A: The `scipy.signal.bilinear` function is supposed to convert a signal from the s-domain to the z-domain, where the z-domain is composed of discrete timestamps instead of a temporal continuum. The `scipy.signal.lfilter` function is supposed to do discrete convolution but with a time-domain input and a z-domain filter and then output the filtered output in the time-domain.

Q: What happens if you use a different sampling frequency in `scipy.signal.bilinear()` than you used for the time-domain signal?

A: If a different sampling frequency is used, then the function may not attenuate certain signals, or it may temporal stretch a signal, or produce artifacts (such as oscillating amplitude for a certain frequency, a wah-wah like effect). For example if the sampling frequency was 3 times that of the input signal sampling frequency then something like this could occur.

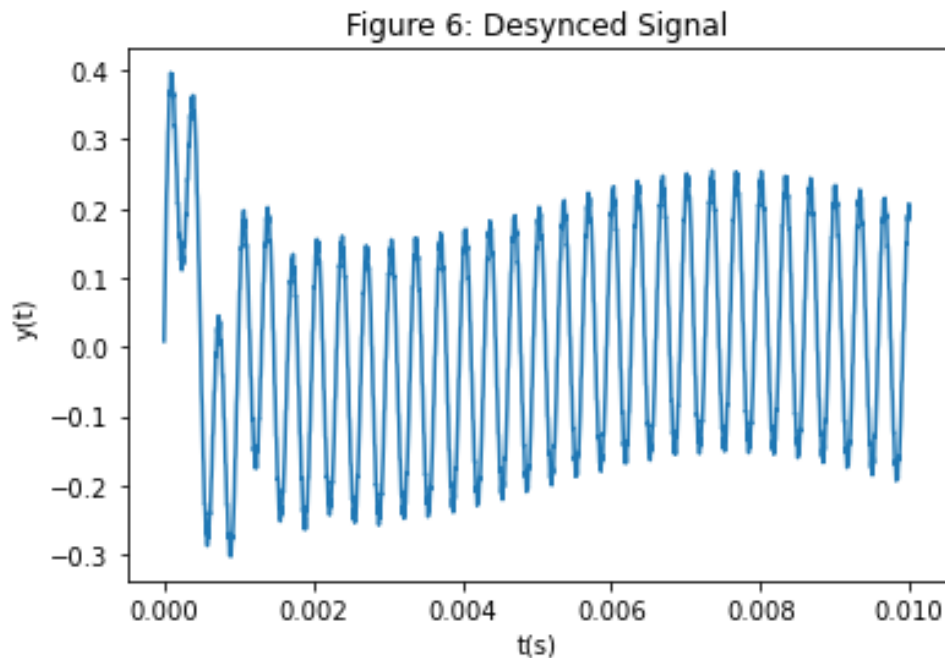


Figure 6 shows that having a miss-match of sampling frequency has undesired effects.

Q: Leave any feedback on the clarity of lab tasks, expectations, and deliverables

A: There was some minor clarity issue with what the variable f_s was.

6 Conclusion

When conducting analysis, it is important to be able to present the findings. sometimes those finding are hidden behind noise because of a too wide of a viewed domain or because of error caused by the inherit nature of the computers.