
Project Report for ECE 351

Lab 10 - Fast Fourier Transform

Skyler Corrigan

November 8, 2022

ECE351 Code Repository:

[https : //github.com/ElfinPeach/ECE351_Code.git](https://github.com/ElfinPeach/ECE351_Code.git)

ECE351 Report Repository:

[https : //github.com/ElfinPeach/ECE351_Report.git](https://github.com/ElfinPeach/ECE351_Report.git)

Contents

1	Objective	1
2	Deliverables	1
2.1	Figures	1
2.2	Discussion	3
3	Questions	3
3.1	Question 1	3
3.2	Question 2	4
3.3	Question 3	4

1 Objective

The objective is to become familiar with frequency response tools and Bode plots using Python. To do so, we are analyzing the following filter:

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

For the second part, the following signal was passed through the filter:

$$x(t) = \cos(2\pi \cdot 100t) + \cos(2\pi \cdot 3024t) + \sin(2\pi \cdot 50000t)$$

2 Deliverables

2.1 Figures

The following graphs are results from the equations.

Figure 1: Part 1 Task 1

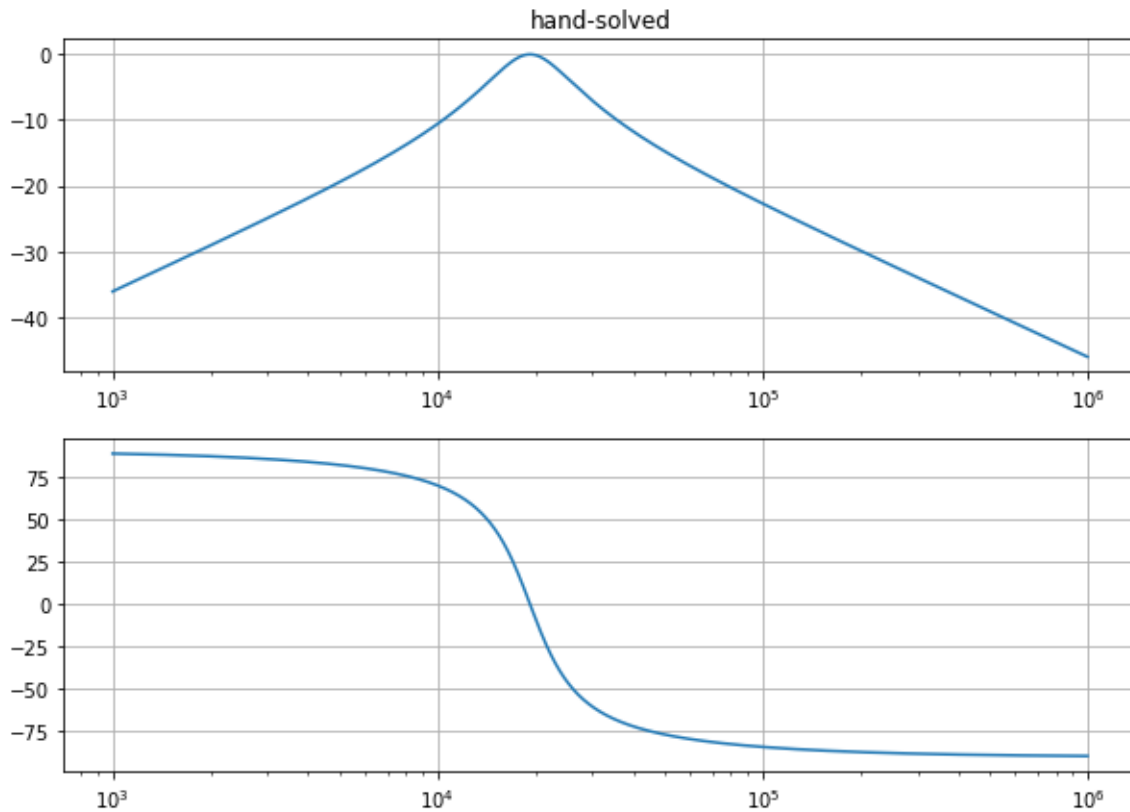


Figure 2: Part 1 Task 2

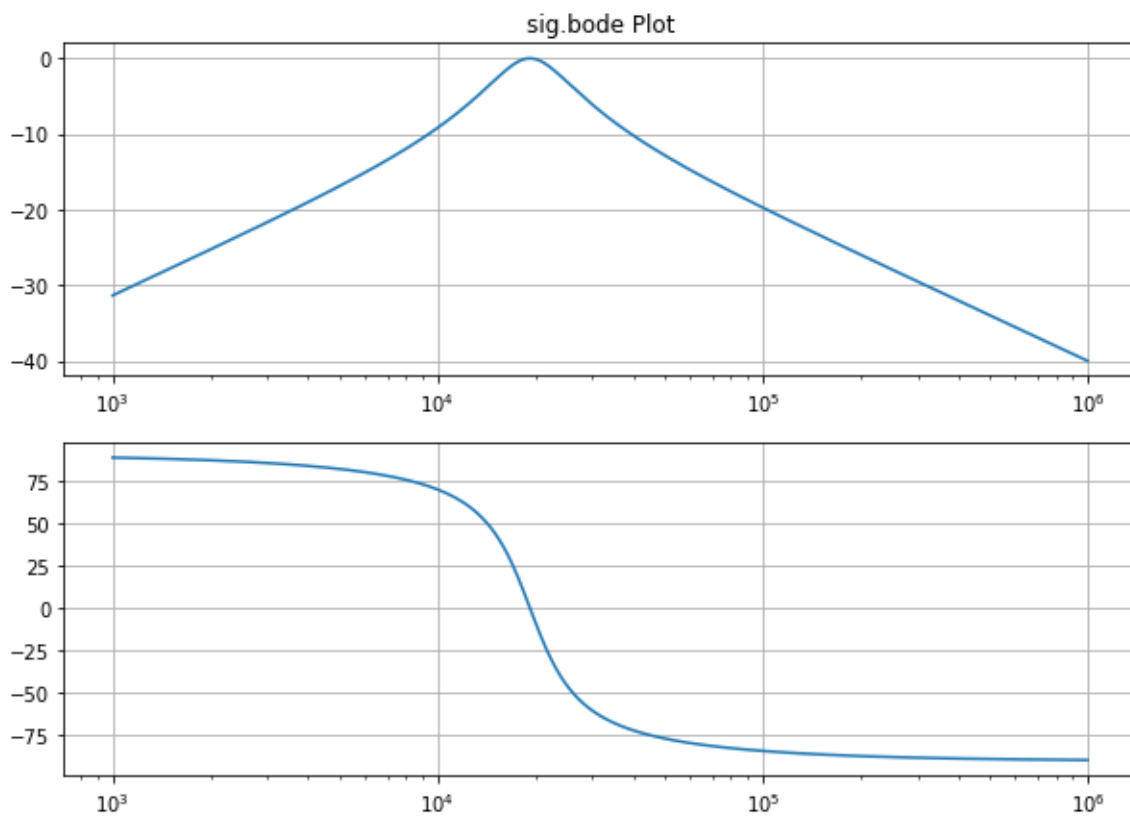


Figure 3: Part 1 Task 3

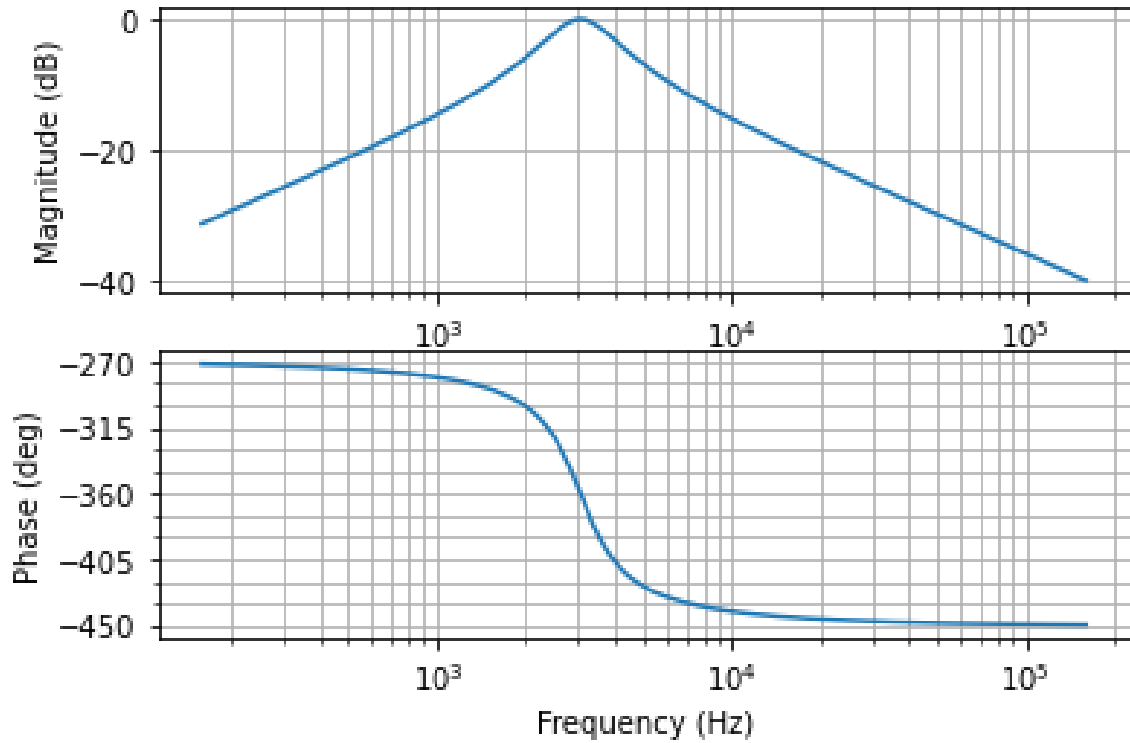
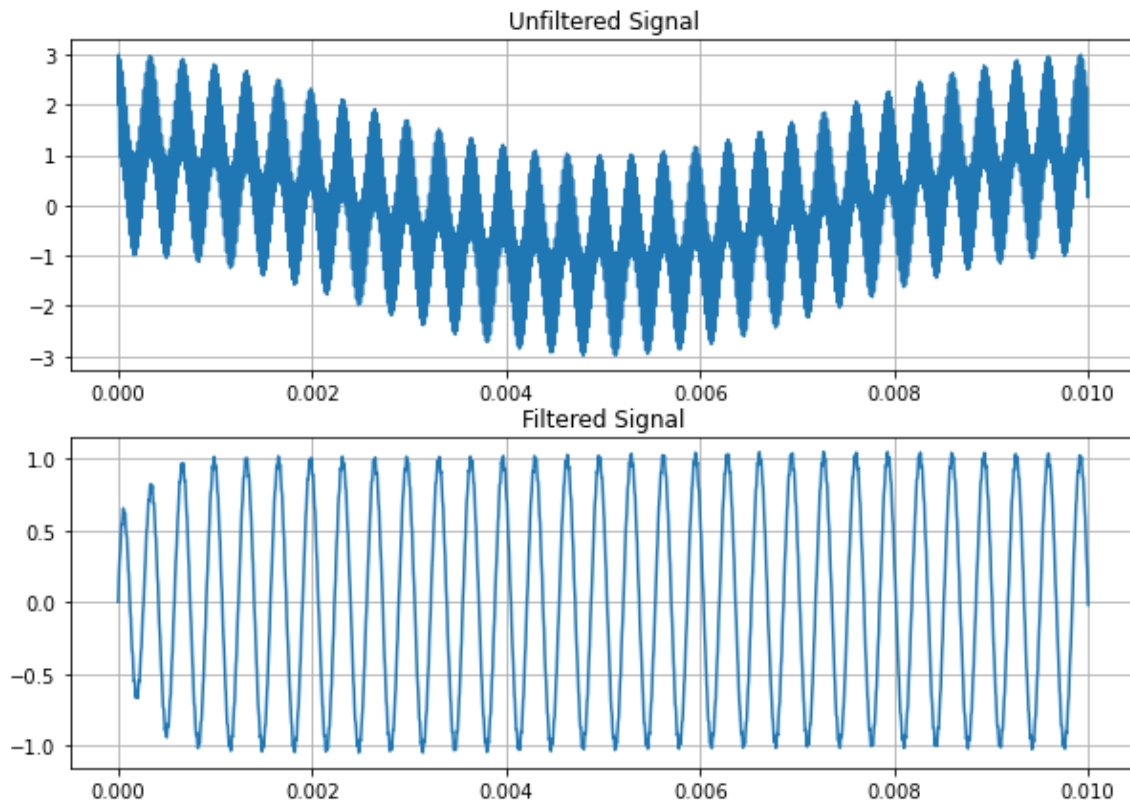


Figure 4: Part 2



2.2 Discussion

My hand-solved plot has slight variation in the magnitude compared to the computer doing the same plot. Otherwise, the three different plots for Part 1 appear to be identical.

In part 2, the filtered signal is a lot nicer and easier to read compared to the unfiltered signal.

3 Questions

3.1 Question 1

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.

This filter is a band-pass filter, so the Bode plot makes sense. Before/after the frequency that results in a zero of the filter, the Bode plot shows that the magnitude reduces dramatically. This shows that the filter's response in the Bode plot makes sense and is consistent with what is expected.

The first few oscillations of the filtered signal in Part 2 show a dampened signal. This is the cosine parts of the input signal being filtered out, leaving the sine signals only.

3.2 Question 2

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

`scipy.signal.bilinear()` is used to convert a transfer function in the Laplace domain to a z-domain function.

`scipy.signal.lfilter()` is used to filter a signal using a z-domain transfer function, allowing the user to take a function in the Laplace domain and use it as a filter in Python.

3.3 Question 3

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

If a different sampling frequency was used, it could modify the resolution of the filtered function. For example, if the sampling frequency was less than the frequency of the function, it could result in inaccurate filtered signals. Increasing the sampling frequency will increase the resolution of the filtered signal.