

ECE 351 - Lab 10 - Frequency Response

Ethan Reeder

7 April 2021

Contents

1	Purpose	3
2	Procedure	3
3	Results	4
4	Questions	6
5	GitHub Link	6
6	Conclusion	7

1 Purpose

The purpose of this lab was to gain familiarity using frequency response and Bode plot tools in Python.

2 Procedure

This lab began with a prelab, in which I calculated the magnitude and phase equations of the transfer function of the circuit shown in Figure 1. These equations are given in Equations 1 through 3.

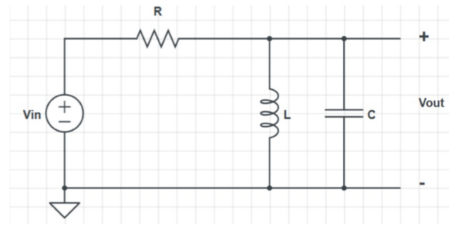


Figure 1: Circuit Diagram for the RLC Filter

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

$$|H(j\omega)| = \frac{\omega L}{\sqrt{(R^2 L^2 C^2 \omega^4 + (L^2 - 2R^2 LC)\omega^2 + R^2)}} \quad (2)$$

$$\angle H(j\omega) = \frac{\pi}{2} - \arctan\left(\frac{\omega L}{R - \omega^2 RLC}\right) \quad (3)$$

Then, using Python, these were graphed versus frequency to create a bode plot, as seen in Figure 2 in the Results section.

Next, the `scipy.signal.bode` command was used to generate a bode plot given the transfer function. This can be seen in Figure 3 in the Results section.

Then, the `control.bode` signal was used to generate a bode plot in Hertz, instead of radians, as well as to demonstrate the abilities and parameters of that function. This final bode plot can be seen in Figure 4 in the Results section.

For Part 2 of the experiment, a signal was given, as seen in Equation 4:

$$x(t) = \cos(2\pi * 100t) + \cos(2\pi * 3024t) + \sin(2\pi * 50000t) \quad (4)$$

This equation was then plotted from $0 \leq t \leq 0.01$ s with a sampling frequency of $2\pi * 50000$ in order to capture all the signals. This plot is shown in Figure 5 of the Results section.

Then, the functions `scipy.signal.bilinear()` and `scipy.signal.lfilter()` were used to create a z-domain equivalent of the transfer function and to pass the input signal through the filter, respectively. The output of this was then plotted over the same time period, as seen in Figure 6 in the Results section.

3 Results

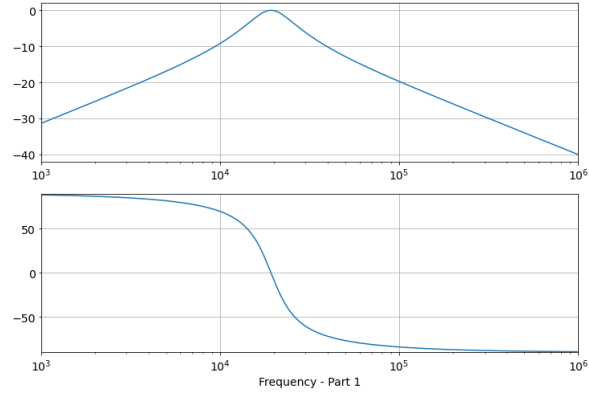


Figure 2: Bode Plot from Prelab Functions

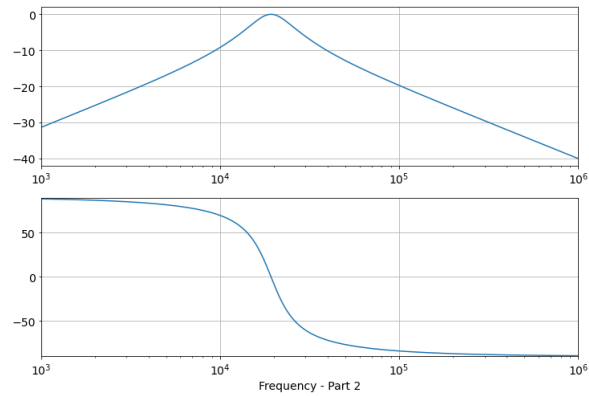


Figure 3: Bode Plot using `scipy.signal` functions

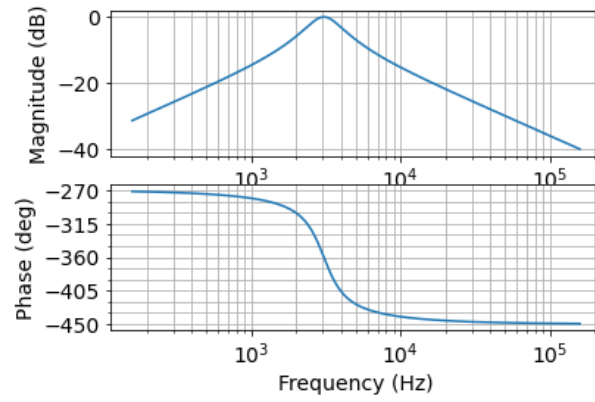


Figure 4: Bode Plot using control functions

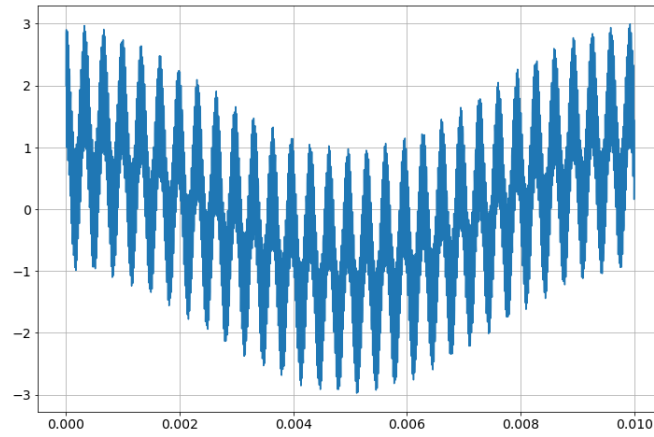


Figure 5: Input Signal

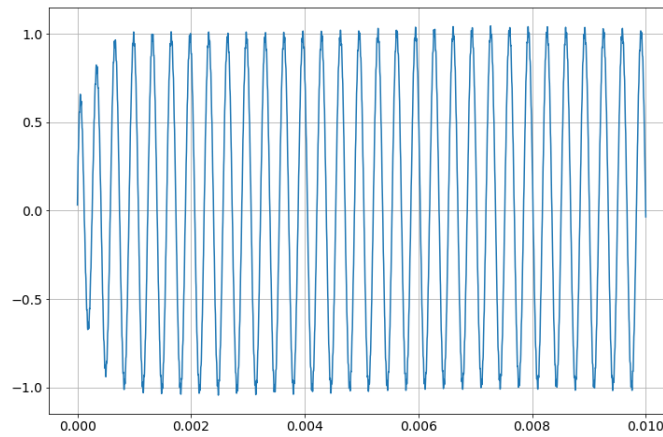


Figure 6: Output Signal

4 Questions

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

The input is a combination of three sinusoidal signals at three different frequencies: 100 Hz, 3024 Hz, and 50000 Hz. According to the Bode Plot shown in Figure 4, which is using Hz as its frequency unit, the output magnitude is at 0 dB at around 3 kHz. This implies an attenuation of 1, or basically no attenuation. At the other two frequencies of the input signal, 100 Hz and 50 kHz, the magnitude is well below -20 dB, which would result in an attenuation of 0.1 or less.

Therefore, the output of the filter, seen in Figure 6, is basically just the graph of $\cos(2\pi * 3024t)$, since its frequency passes through the filter with little-to-no affect.

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

The `scipy.signal.bilinear()` function creates a z-transform equivalent model of a transfer function. The outputs of this function are then used as inputs to the `scipy.signal.lfilter()` function, along with an input signal, to determine the filtered output of that signal, given the transfer function.

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

Similar to how if the step size is too big, the plot will not be an accurate representation of the data given a fast frequency, the sampling frequency for the bilinear function will determine the frequency of the z-transform of the transfer function. Thus, if it's too low, then the output of the `lfilter` function will be imprecise. In general, it's best to have the sampling frequency be high enough to "see" all the inputs.

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

The lab was clear and the expectations were understood.

5 GitHub Link

<https://github.com/EReeder35>

6 Conclusion

In conclusion, this was another educational lab for learning more Python skills and cementing my understanding of concepts from the lecture, especially the magnitudes and phases of the Fourier transforms of various functions.