

Frequency Response

Cameron Williams

ECE 351-51

Lab Report 10

14 April 2020

1 Introduction

The objective of this lab was to become familiar with frequency response tools and Bode plots using Python.

2 Methodology

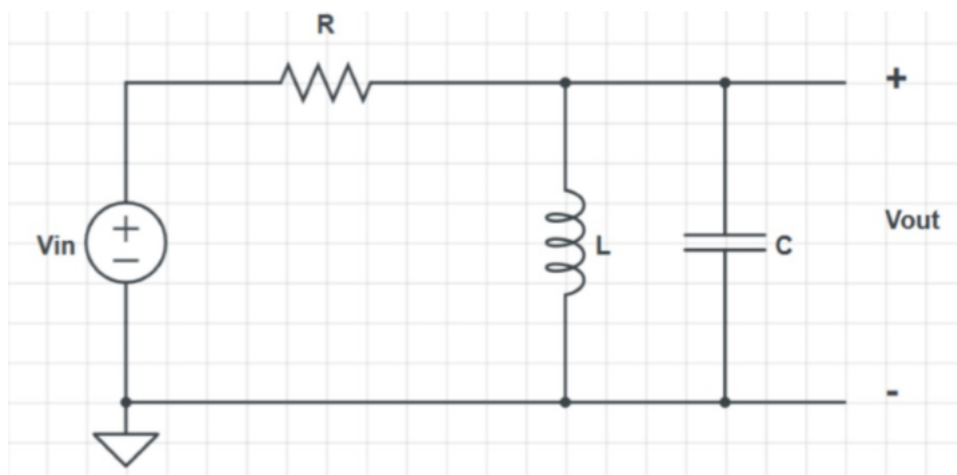


Figure 1: $R = 1\text{k}\Omega$, $L = 27\text{ mH}$, $C = 100\text{ nF}$

Consider the RLC circuit in figure 1. Which has the transfer function,

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

I started this lab by creating a Python function called `H_out` for determining the magnitude (in dB) and phase shift (in degrees) of an input frequency for the transfer function provided. I used the `matplotlib.pyplot.semilogx()` function to create a bode plot of the output values. The Bode plot may be seen in Figure 1 of the Results section. Next, I used the `scipy.signal.bode()` function to verify the results of my manually created bode plot. This plot may be seen in Figure 2 of the Results section. This plot showed that there was a problem with the phase shift displayed by the plot in Task 1. In particular, there were values that were phase shifted by 90 degrees or more that I corrected by shifting them down to match the plot in Task 2. Note that the included plot for task 1 is the one that has been fixed. For the final task of Part 1, I used the control package to recreate the plot from Task 2 with the frequency displayed in Hertz. This new plot may be seen

in Figure 3 of the Results section.

For the next part, I started by plotting $x(t) = \cos(2\pi 100t) + \cos(2\pi 3024t) + \sin(2\pi 50000t)$ between 0 and 0.01 seconds. I set the sampling frequency to 100000 in order to accurately sample all frequencies that make up $x(t)$. A plot of $x(t)$ may be seen in Figure 4 of the Results section. Finally, I used the function `scipy.signal.bilinear()` to convert the $x(t)$ signal to the z-domain and passed this new function, $X(z)$, through the `scipy.signal.lfilter()` function to create the output $y(t)$. A plot of $y(t)$ may be seen in Figure 5 of the Results section.

Results

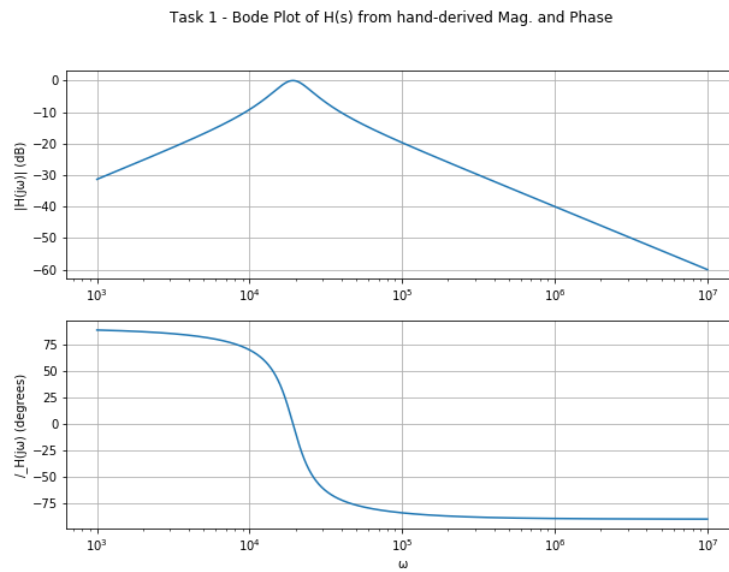


Figure 1: Part 1 Task 1 Plot.

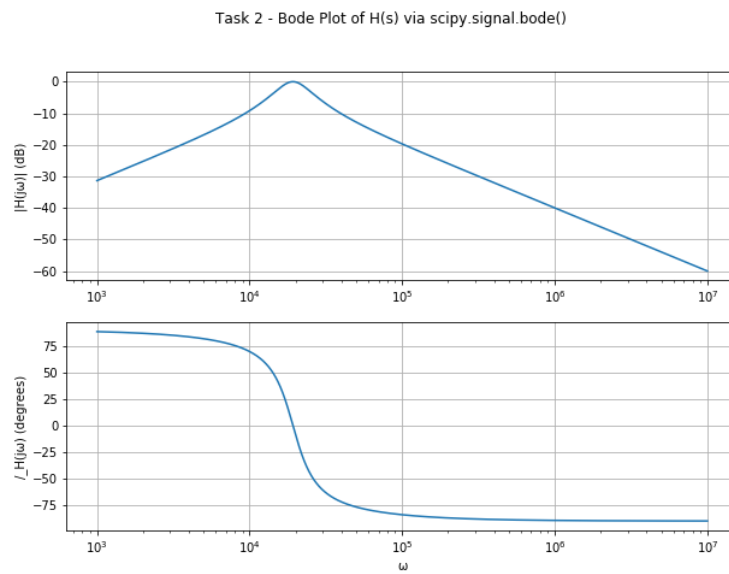


Figure 2: Part 1 Task 2 Plot.

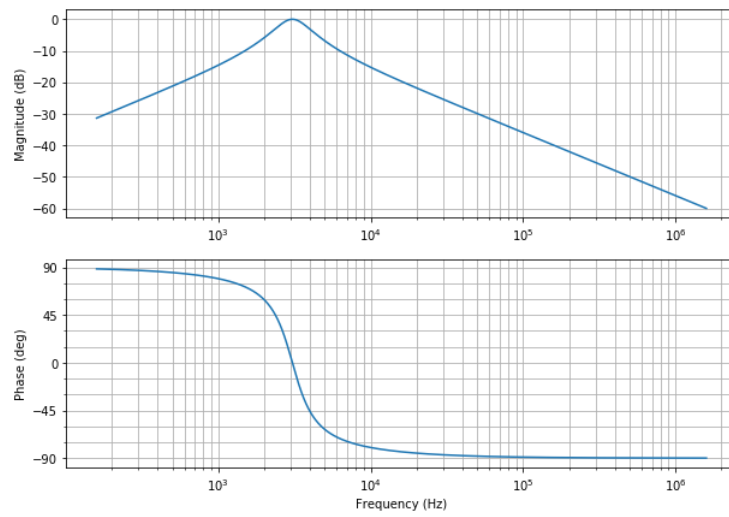


Figure 3: Part 1 Task 3 Plot.

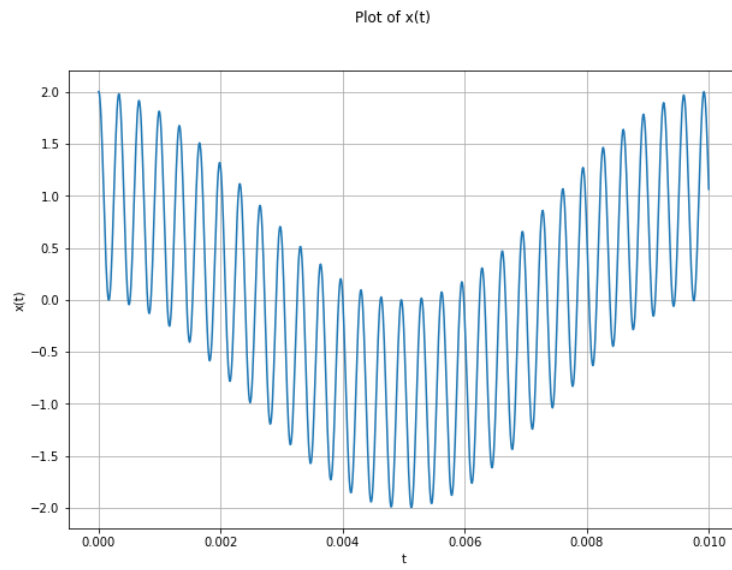


Figure 4: Part 2 Task 1 Plot.

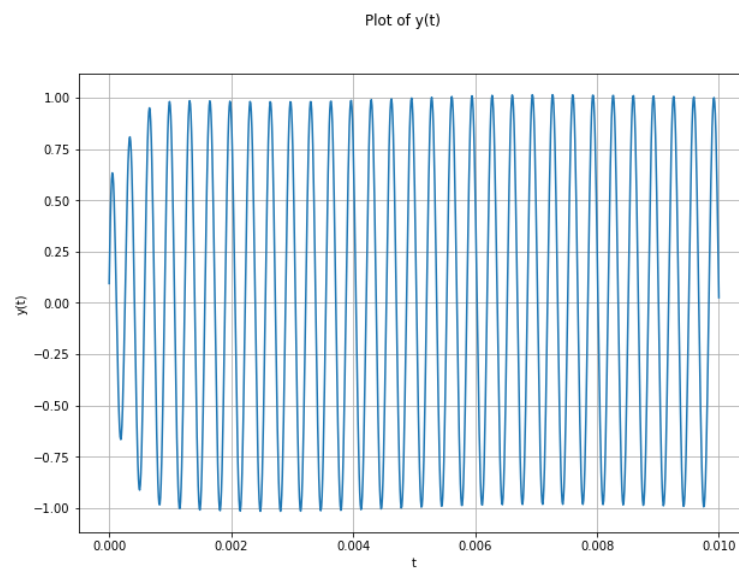


Figure 5: Part 2 Task 2 Plot.

Questions

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

There are three frequencies present in $x(t)$: 100 rad/s, 3024 rad/s, and 50000 rad/s. In the Bode plots in Part 1, 100 rad/s and 50000 rad/s are both attenuated by more than 90% while 100 rad/s is attenuated by less than 30%, so the 100 rad/s component is clearly visible in the filtered output $y(t)$.

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

As the lab handout implies, the bilinear function is used to transform a time-domain signal into a z-domain signal. Like many of the other functions we've used from the `scipy.signal` library, the inputs are the numerator and denominator arrays of coefficients of the s terms of the transfer function, though the sampling frequency is also passed as an argument. The `scipy.signal.lfilter()` function takes a pair of z-domain numerator and denominator arrays representing the transfer function of the filter in question as well as an input signal in order to produce the numerator and denominator for the output signal.

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

If a smaller sampling frequency is used in the bilinear function call than what was used on the time domain signal, the outputted magnitudes are all smaller than normal. With a larger sampling frequency is used, the outputted magnitudes are larger than normal.

4. Leave any feedback on the clarity/usefulness of the purpose, deliverables, and expectations for this lab.

The expectations for this lab were expressed clearly in the lab handout.