



University of Idaho

ECE 351 - SECTION #53

---

## Lab #10 Report

---

### FREQUENCY RESPONSE

*Submitted To:*

Kate Antonov  
University of Idaho  
kantonov@uidaho.edu

*Submitted By :*

Macallyster S. Edmondson  
University of Idaho  
edmo7033@vandals.uidaho.edu  
[github.com/mac-edmondson](https://github.com/mac-edmondson)

5<sup>th</sup> April, 2022

# Contents

1	Introduction . . . . .	2
2	Equations . . . . .	2
3	Methodology . . . . .	2
	3.1 Lab: Part 1 . . . . .	2
	3.2 Lab: Part 2 . . . . .	4
4	Results . . . . .	4
5	Error Analysis . . . . .	10
6	Questions . . . . .	10
7	Conclusion . . . . .	11
8	Attachments . . . . .	12

# 1 Introduction

The goal of this weeks lab was to become familiar with frequency response tools and Bode plots using Python. This lab was completed using *Python* through the *Spyder-IDE*. The packages used in the completion of this lab were `numpy` for definitions of mathematical functions, `matplotlib.pyplot` to plot outputs of functions, `scipy.signal` to filter an input based on a transfer function, and `control` to create Bode plots with already existing functions.

All code for this lab, including this report, can be found on my Github.

# 2 Equations

The equations used within this lab are shown in this section. The equations will be referenced by number throughout the rest of the report.

Preliminary Transfer Function & derived magnitude and phase:

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

$$|H(j\omega)| = \frac{\frac{1}{RC}\omega}{\sqrt{\omega^4 + [(\frac{1}{RC})^2 - \frac{2}{LC}]\omega^2 + (\frac{1}{LC})^2}} \quad (2)$$

$$\angle H(j\omega) = \frac{\pi}{2} - \tan^{-1}\left(\frac{\frac{1}{RC}\omega}{-\omega^2 + \frac{1}{LC}}\right) \quad (3)$$

Part 2 Input Signal:

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

# 3 Methodology

## 3.1 Lab: Part 1

In Part 1 of this lab, we used the derived magnitude and angle of the pre-lab [Equations (2) & (3)] to plot a Bode plot using the `matplotlib.pyplot.semilogx()` function. This same process was done using the `control.bode()` function by simply plugging in the original transfer function coefficients. One plot was generated with an x-axis unit of Hz which proved quite useful for analysis of Part 2 of this lab.

Code implementing Part 1 of this lab can be seen in the below listing.

```

1 # PART 1
2
3 R = 1e3 # Ohms
4 L = 27e-3 # H
5 C = 100e-9 # F
6
7 # 1
8 def mag_db(omega) :
9     out = (((1/(R*C))*omega)/(np.sqrt((w**4) + (((1/(R*C))**2) - (2/(L
10     *C))) * (omega**2)) + ((1/(L*C))**2))))
11     out_db = 20 * np.log10(out)
12     return out_db
13
14 def ang_deg(omega) :
15     ang = np.pi/2 - np.arctan(((1/(R*C))*omega)/((1/(L*C))-(omega**2)))
16     ang_deg = (180/np.pi) * ang
17     return ang_deg
18
19 w = np.arange(1e3, 1e6 + step_size, step_size)
20 mag = mag_db(w)
21 ang = ang_deg(w)
22
23 for i in range(len(ang)):
24     if(ang[i] > 90):
25         ang[i] -= 180
26
27 plt.figure(figsize = (10, 11))
28 plt.subplot(2, 1, 1)
29 plt.semilogx(w, mag, "b-")
30 plt.grid(True, which='both', ls='--')
31 plt.ylabel(' |H(S)| [dB] ')
32 plt.title('Magnitude and Phase Bode Plot [Manual]')
33
34 plt.subplot(2, 1, 2)
35 plt.grid(True, which='both', ls='--')
36 plt.semilogx(w, ang, "b-")
37 plt.ylabel(' angle(H(S)) [deg] ')
38 plt.xlabel('w [rad/s] ')
39
40 # 2
41 num = [1/(R*C), 0]
42 den = [1, 1/(R*C), 1/(L*C)]
43
44 sys = con.TransferFunction(num, den)
45
46 plt.figure(figsize=(10,11))
47 plt.title('Magnitude and Phase Bode Plot [spsig.bode]')
48 _ = con.bode(sys, omega=None, dB = True, Hz = False, deg = True, Plot =
    True)
49
50 plt.figure(figsize=(10, 11))

```

```
49 plt.title('Magnitude and Phase Bode Plot [spsig.bode] (Hz)')
50 - = con.bode(sys, omega=None, dB = True, Hz = True, deg = True, Plot =
    True)
```

### 3.2 Lab: Part 2

In Part 2 of this lab we used the `scipy.signal.bilinear()` & `scipy.signal.lfilter()` functions to filter the input function seen in Equation (4). Figures 4 & 5 show the input function  $x(t)$  and the output after it was run through the filter. This was an extremely useful exercise which I found crucial for my understanding of concepts discussed in class. Such a simple exercise can be so helpful.

Code implementing Part 2 of this lab can be seen in the below listing.

```
1 # PART 2
2 fs = 2*np.pi*50000
3 step_size = 1/fs
4
5 t = np.arange(0, 0.01 + step_size, step_size)
6 x = np.cos(2*np.pi*100*t) + np.cos(2*np.pi*3024*t) + np.sin(2*np.pi
    *50000*t)
7 plt.figure(figsize=(10, 11))
8 plt.subplot(1, 1, 1)
9 plt.plot(t, x, "b-")
10 plt.grid()
11 plt.ylabel('x(t)')
12 plt.xlabel('t')
13 plt.title('Part 2 Task 1')
14
15 z, p= spsig.bilinear(num, den, fs=fs)
16 y = spsig.lfilter(z, p, x)
17
18 plt.figure(figsize=(10, 11))
19 plt.subplot(1, 1, 1)
20 plt.plot(t, y, "b-")
21 plt.grid()
22 plt.ylabel('y(t)')
23 plt.xlabel('t')
24 plt.title('Part 2 Task 4')
```

## 4 Results

The results of this lab are very straightforward. The implementation of all functions worked as expected and the results are as expected. More analysis of theory and results is discussed in the Questions Section of this report.

The deliverables for Part 1 & 2 of this lab are seen in all figures given below.

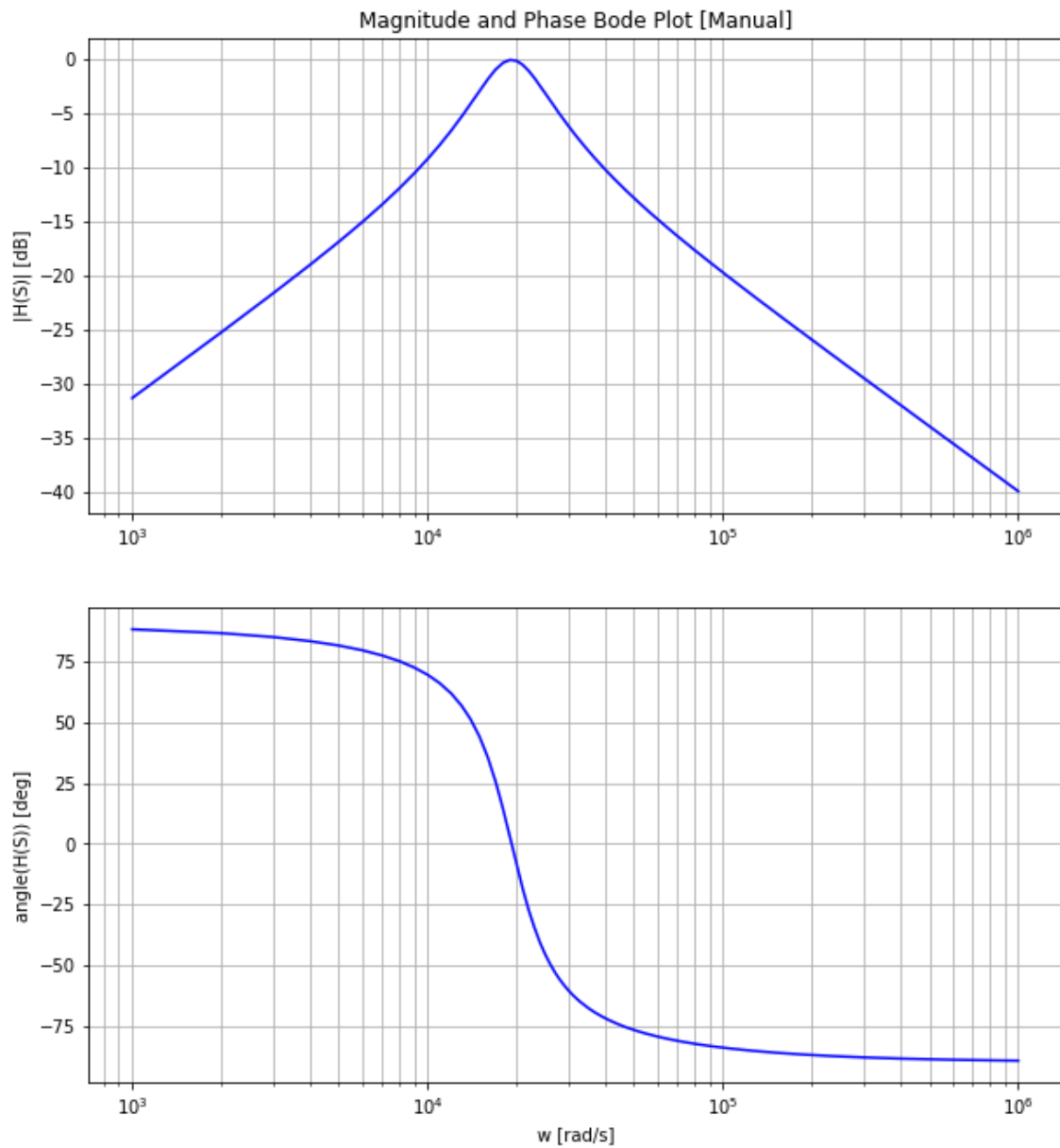


Figure 1: Mag. & Phase Bode Plot, Manual (rad/s) [Equations (2) & (3)]

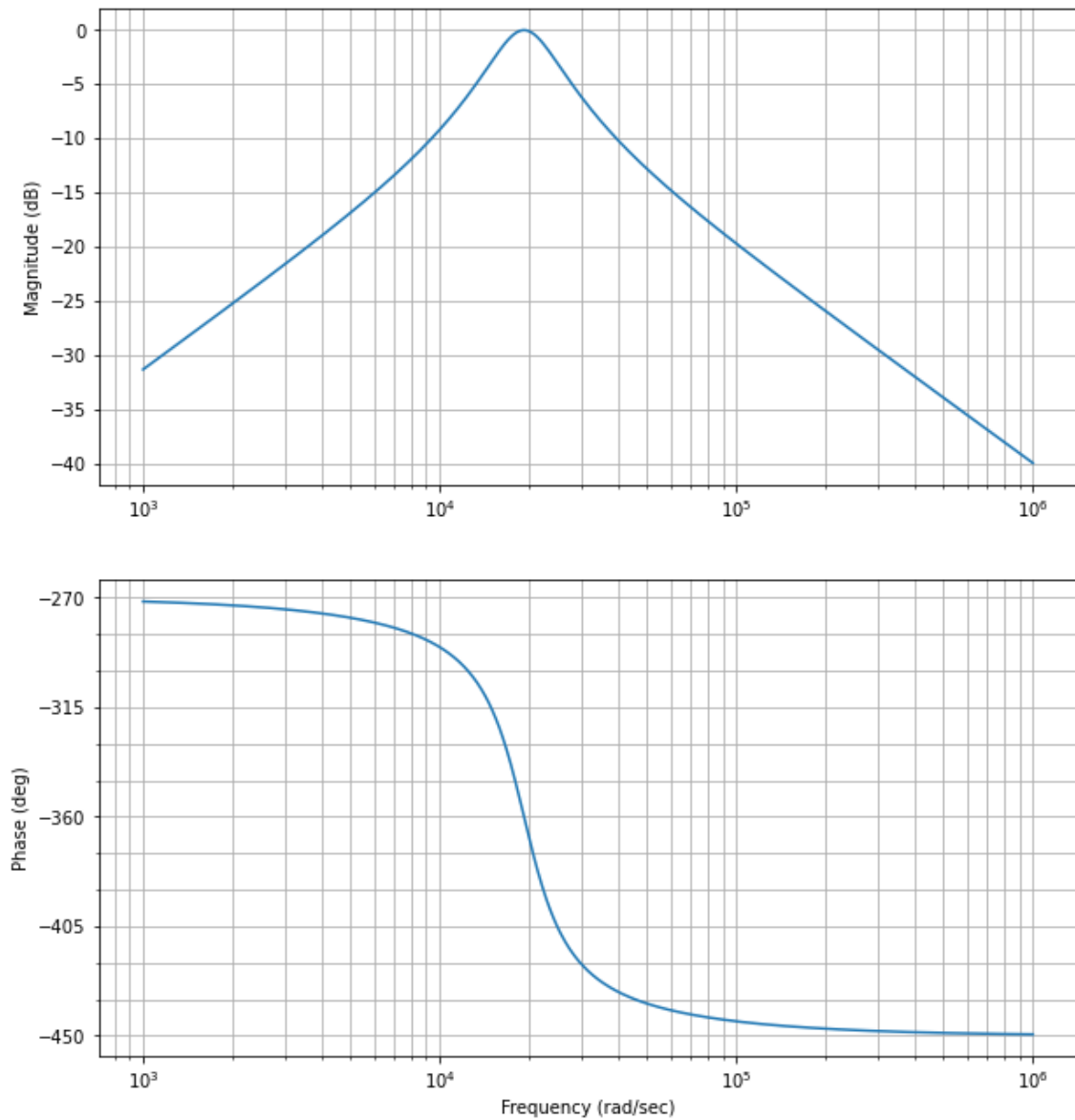


Figure 2: Mag. & Phase Bode Plot, `spsig.bode()` (rad/s) [Equation (1)]

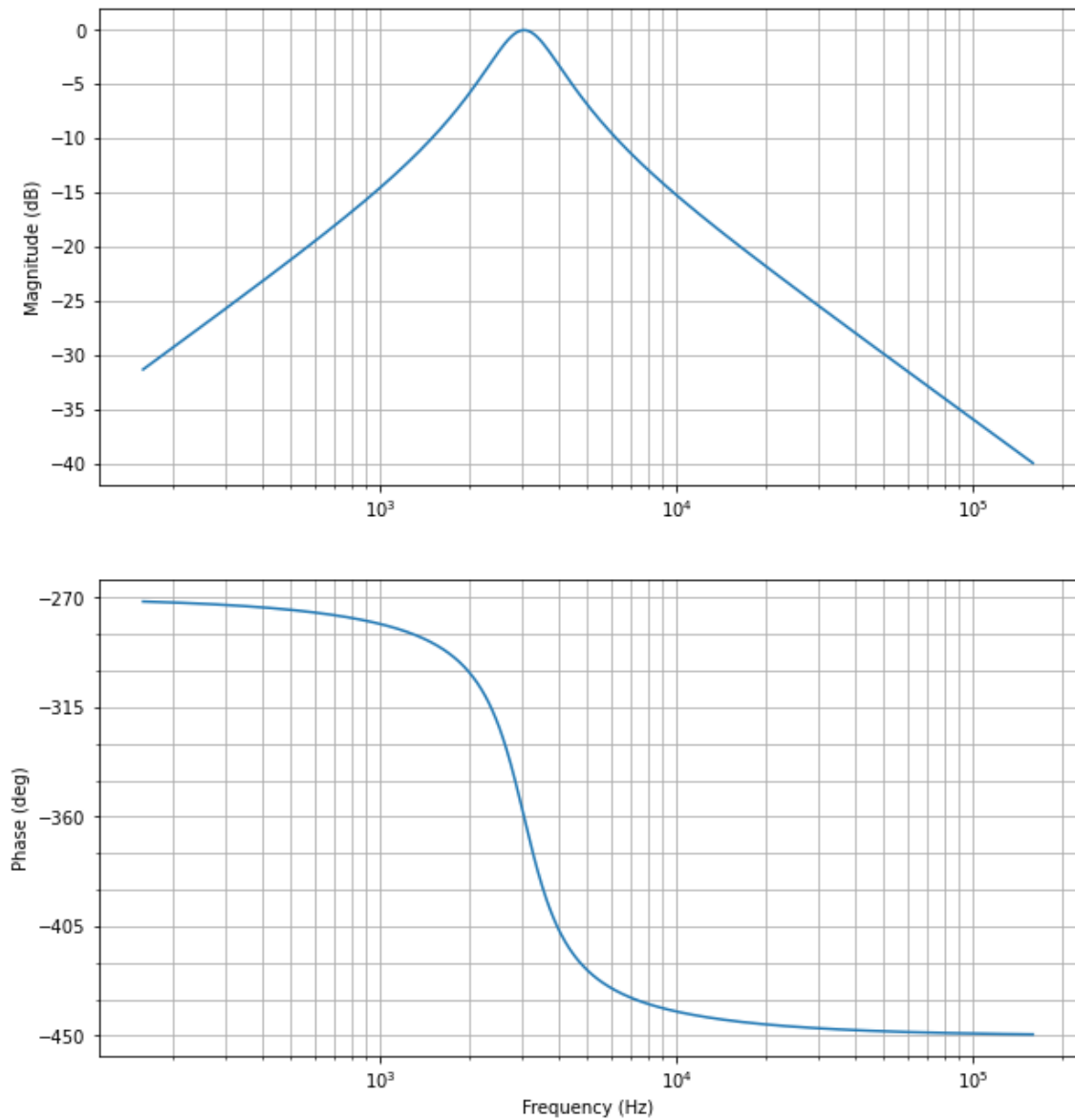
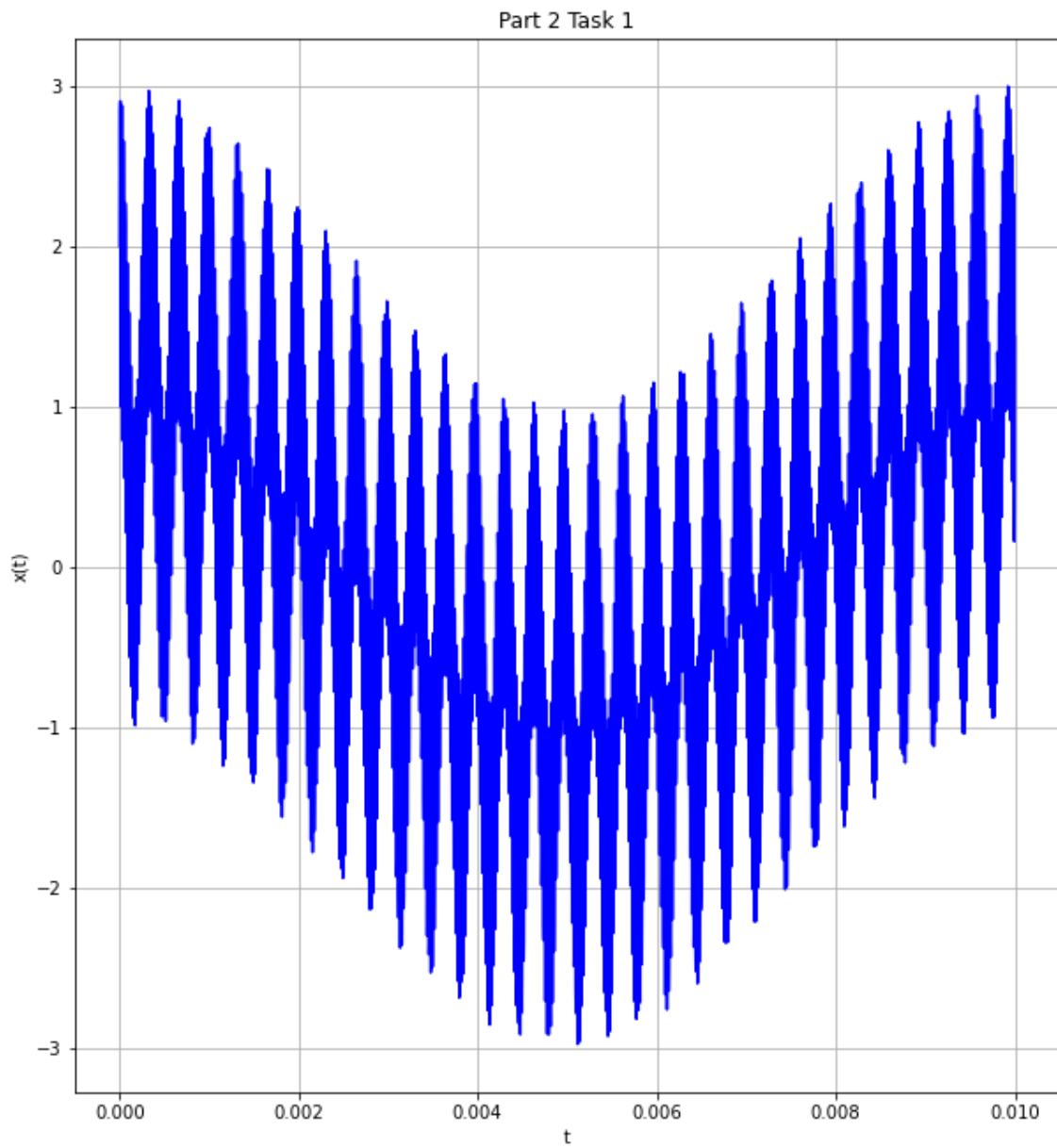


Figure 3: Mag. & Phase Bode Plot, `spsig.bode()` (Hz) [Equation (1)]



Figure 4: Filter Input Function,  $x(t)$  [Equation (4)]

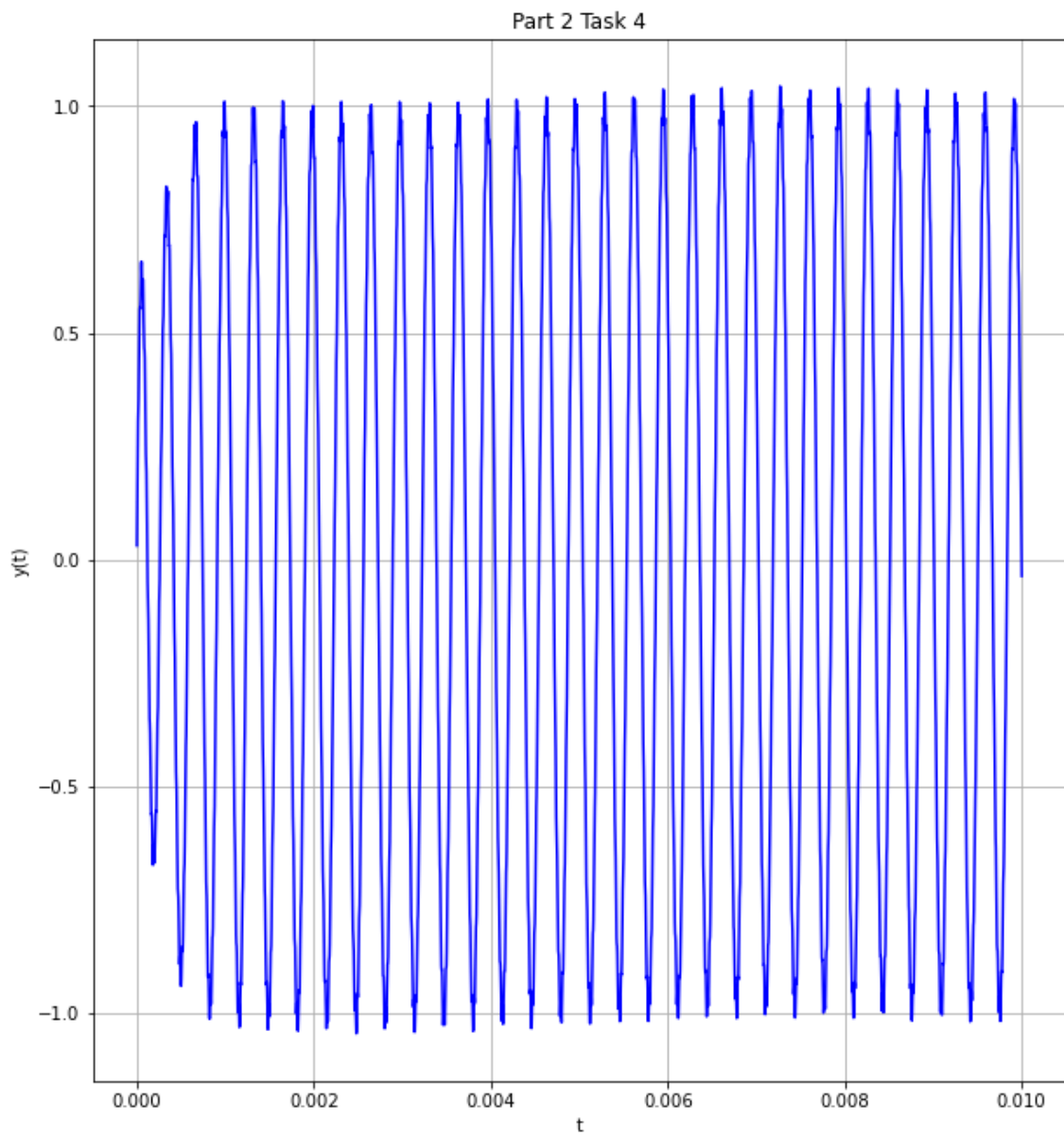


Figure 5: Filter Output

## 5 Error Analysis

No sources of error were seen throughout this lab other than the need for a phase shift of the manually plotted Bode plot (Figure 1) for readability purposes. Other than this no other issues were encountered.

## 6 Questions

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.
  - As seen from the Bode plots in Part 1, the filter described by the transfer function is a band-pass filter only allowing frequencies of  $2\pi \times 3000$  rad/s (3000 Hz) through. Thus the dominant term that should be seen in the output of the filter should be the second term from the input, Equation (4). Looking at the plot in Figure 5, we can see there are approximately 6 periods in 2 milliseconds which gives an approximate frequency of 3000 Hz. As this is the frequency of the second term in the input signal, it appears the filter is working properly. As the frequency is not exactly 3000 Hz, the output of the filter will have a slight magnitude attenuation and phase shift (more noticeable) which is seen in the output of the filter, Figure 5.
2. Discuss the purpose and workings of `scipy.signal.bilinear()` and `scipy.signal.lfilter()`.
  - The `spsig.bilinear()` function performs a bilinear transformation on an s-domain transfer function into the z-domain using Tustin's method. Once the transfer function is in the z-domain, digital filtering is much easier to achieve. The z-domain filter is then plugged into the `spsig.lfilter()` function which uses the z-domain to filter the input function, then is transformed back and returned.
3. What happens if you use a different sampling frequency in `scipy.signal.bilinear()` than you used for the time-domain signal?
  - Like seen in Lab #9, a different frequency sample could have detrimental consequences. If a frequency sample close to the time-domain sampling frequency is used, the output will look close to what it should be but could be time shifted and scaled slightly. If a vastly different sampling frequency is used, the output will be unrecognizable due to the amount of time scaling that takes place. In either case, the results of the output

function will be incorrect.

4. Leave any feedback on the clarity of lab tasks, expectations, and deliverables.
  - This lab was very clear for all instructions, expectations and deliverables.

## 7 Conclusion

In conclusion, I feel this lab was very successful. I really enjoyed the lab questions which helped me recognize some aspects of filters I did not understand before. I feel this has been one of the most helpful labs for my understanding of class topics. All in all, I am very satisfied with what this lab has taught me and feel it was an excellent use of time.

## 8 Attachments

1. Pre-Lab

# Frequency Response

Prelab 10

Spring 2021

## 1 Purpose

Find the analytical expression for magnitude and phase of a transfer function.

## 2 Deliverables Overview

- Typed and properly formatted derivation for both the magnitude and phase of the transfer function. *Note: All steps must be shown.*

## 3 Part 1

### 3.1 Tasks

Consider the RLC circuit in figure 1.

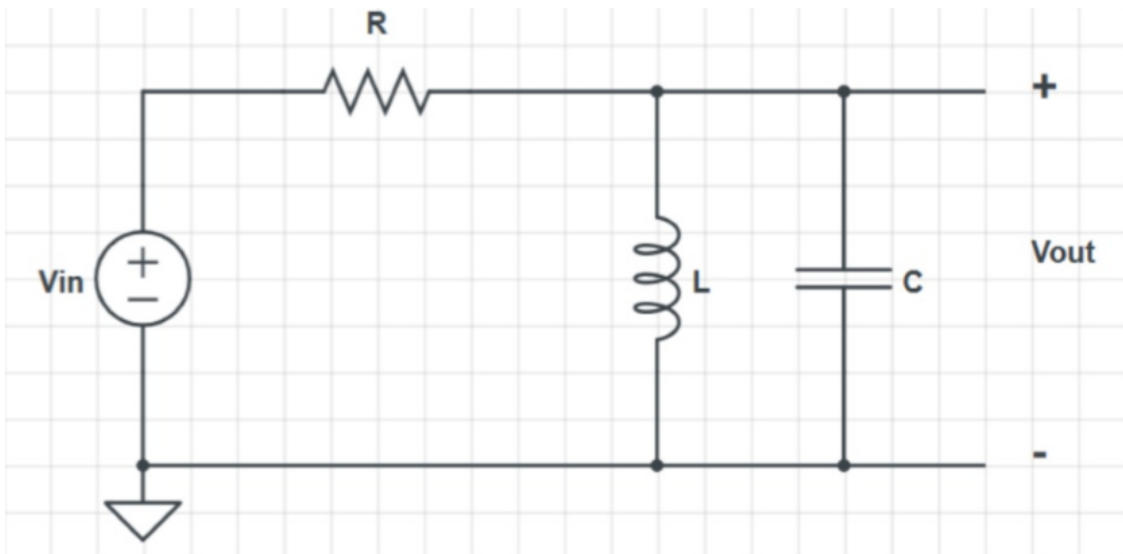


Figure 1:  $R = 1\text{ k}\Omega$ ,  $L = 27\text{ mH}$ ,  $C = 100\text{ nF}$

Which has the transfer function,

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

1. By hand, find the magnitude  $|H(j\omega)|$  and the phase  $\angle H(j\omega)$  for the RLC transfer function  $H(s)$ . Type the analytical derivation symbolically in terms of R, L, and C. Do not use your calculator or Python on this step, show all your work.

$$H(j\omega) = \frac{\frac{1}{RC} j\omega}{(j\omega)^2 + \frac{1}{RC} j\omega + \frac{1}{LC}} = \frac{\frac{1}{RC} j\omega}{\frac{1}{RC} j\omega + \left(\frac{1}{LC} - \omega^2\right)} = \frac{j\omega}{RC(LCj\omega + RC - LC\omega^2)} = \frac{j\omega}{RC^2(Lj\omega + R - LC\omega^2)}$$

$$|H(j\omega)| = \frac{\omega}{RC^2 \sqrt{(R - LC\omega^2)^2 + (L\omega)^2}} = \frac{\omega}{RC^2 \sqrt{R^2 - 2R^2LC\omega^2 + R^2L^2C^2\omega^4}} = \frac{\omega}{R^2C^2 \sqrt{1 - 2LC\omega^2 + L^2C^2\omega^4}}$$

$$|H(j\omega)| = \frac{\omega}{R^2C^2 \sqrt{1 - 2LC\omega^2 + L^2C^2\omega^4}}$$

$$\angle H(j\omega) = \frac{\pi/2}{\tan^{-1}\left[\frac{RC^2}{RC^2} \frac{L\omega}{R - LC\omega^2}\right]}$$

$$\angle H(j\omega) = \frac{\pi}{2 \cdot \tan^{-1}\left(\frac{L\omega}{R(1 - LC\omega^2)}\right)}$$

