ECE 351

Signals and Systems

Lab 10

Frequency Response

Submitted By :
Zachary Pfaff
https://github.com/ZachPfaff

Contents

Frequency Response

Zachary Pfaff

April 5th 2022

1 Introduction

As recently learned in ECE 350, it is possible to see the magnitude and phase response of a system through the use of a bode plot. Bode plots are very important in that they can show us what comes out of a transfer function given a signal. The goal of the following lab is to create our own bode plots based on a transfer function calculated in our lab 10 prelab. The magnitude and phase of the transfer function is shown in the two equations below.

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

$$\angle H(jw) = \frac{\pi}{2} - \arctan(\frac{\frac{1}{RC}w}{-w^2 + \frac{1}{LC}})$$

2 Methodology

The first task of lab asks us to plot the magnitude and phase of the transfer function from 10^3rad/s to 10^6rad/s. This was done by first implementing our hand calculated magnitude and phase equations into python, then using matplotlib.pyplot.semilogx() to plot the x axis on a logarithmic scale, otherwise the scaling for the plot would be off. The code for task 1 is shown below in listing 1. Note that a for loop had to be included to shift the phase plot to appropriately display its phase. This was done by converting my phase equation from radians to degrees.

```
mag = 20*np.log10((w/(r*c))/(np.sqrt((w**4) + ((1/(r*c))**2 - (2/(l
    *c)))*(w**2) + (1/(l*c))**2)))
phase = (np.pi/2) - np.arctan((w/(r*c))/((-(w**2))+(1/(l*c))))

for i in range(len(phase)):
    if  (phase[i]*(180/np.pi)) > 90:
        phase[i] = (phase[i]*(180/np.pi)) - 180
    else:
        phase[i] = phase[i]*(180/np.pi)
```

Listing 1: Part 1 Task 1 Code

The next task of part 1 asked us to plot the frequency of the magnitude and phase again, except this time using scipy.signal.bode. By doing so, we will be able to see if our hand calculated transfer function bode plot matches the bode plot created using python tools. To implement this, a matrix for the numerator and denominator were created. The numerator and denominator were then put into the sig.bode function and set equal to an array that displays the frequency, magnitude, and phase of the sig.bode function output. The code is shown below in listing 2.

```
num = [(1/(r*c)), 0]
den = [1, (1/(r*c)), (1/(l*c))]

h = sig.bode((num, den), w)
```

Listing 2: Part 1 Task 2 Code

The final task for part 1 asks us to plot the frequency response of the same system in Hz instead of rad/s. This code required us to import the control package which I was successfully able to run on mac. The code of this task is shown below in listing 3.

```
import control as con

sys = con.TransferFunction(num, den)
_ = con.bode(sys, w, dB = True, deg = True, Plot = True)
```

Listing 3: Part 1 Task 3 Code

With the bode plot of the transfer function successfully created, we can now start to pass signals into the transfer function to see what comes out. The signal that we will be using for part two is shown in the equation below.

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

The sampling frequency is set high enough to capture each frequency and the step size is set to be 1/f. The signal is also plotted from 0 to 0.01 seconds. The code for this first task is shown below in listing 4.

```
f = 1e6
steps2 = 0.000001
t = np.arange(0, 0.01 + steps2, steps2)
xt = np.cos(2*np.pi*100*t) + np.cos(2*np.pi*3024*t) + np.sin(2*np.
    pi*50000*t)
```

Listing 4: Part 2 Task 1 Code

The next task of part 2 asks us to convert the transfer function into the z-domain. This is done scipy.signal.bilinear() and is impleted into python similarly to the sig.bode command. This only required one line of code which is shown below in listing 5.

```
h1 = sig.bilinear(num, den, f)
```

Listing 5: Part 2 Task 2 Code

With the transfer function converted to the z-domain, we are then able to pass the input signal through the filter using scipy.signal.lfilter(). The output for the

signal before and after being passed through the signal is then graphed. The code for this final task is shown below in listing 6.

```
yt = sig.lfilter(h1[0], h1[1], xt)

plt.figure(figsize = (15, 10))
plt.plot(t, xt)
plt.title('x(t)')
plt.xlabel('t')
plt.ylabel('x')

plt.figure(figsize = (15, 10))
plt.plot(t, yt)
plt.title('x(t)')
plt.xlabel('t')
plt.ylabel('x')
```

Listing 6: Part 2 Task 3 and 4 Code

3 Data

The following section of the lab 10 report displays the plots from each individual task. The first figure shows the bode plots created in task 1 and 2 being compared to ensure the hand calculated transfer function is correct. The following figure shows the frequency response of the same system using hz instead of rad/s. The final figure is a comparison of the output signal before and after the transfer function was converted to the z-domain.

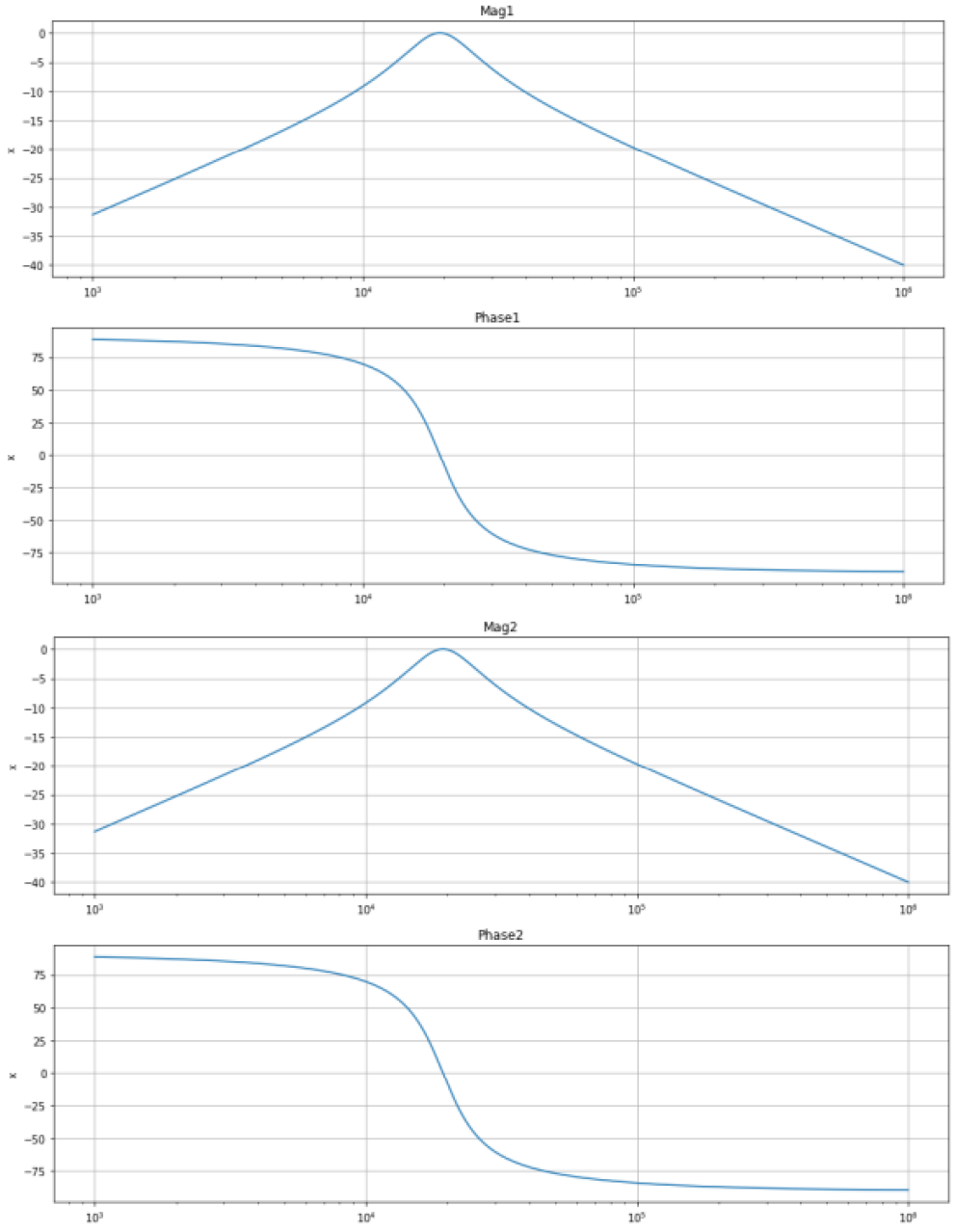

Figure 1: Hand Calculated Transfer Function vs. sig.bode function

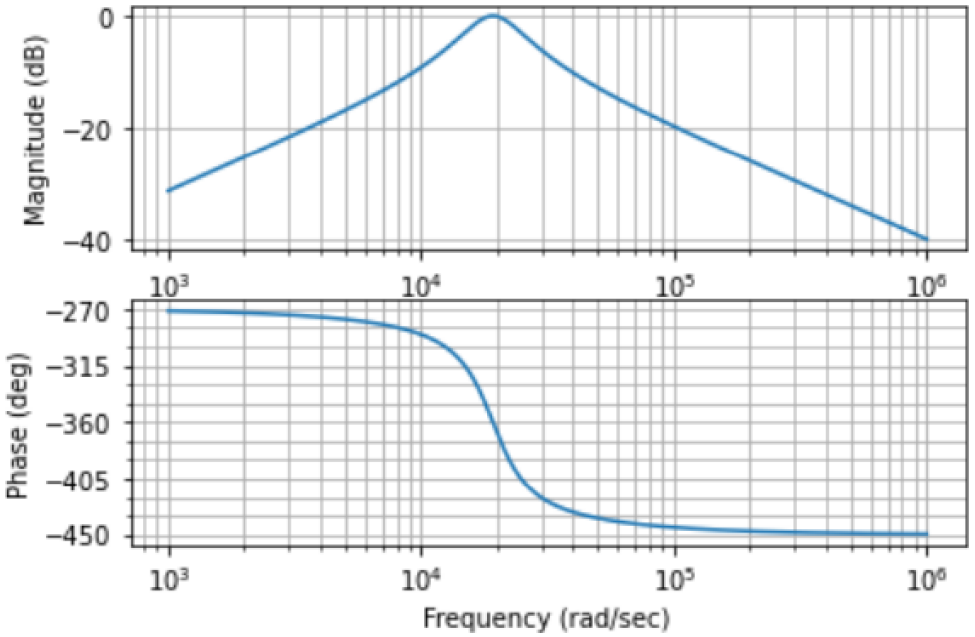

Figure 2: Transfer function in units of Hz

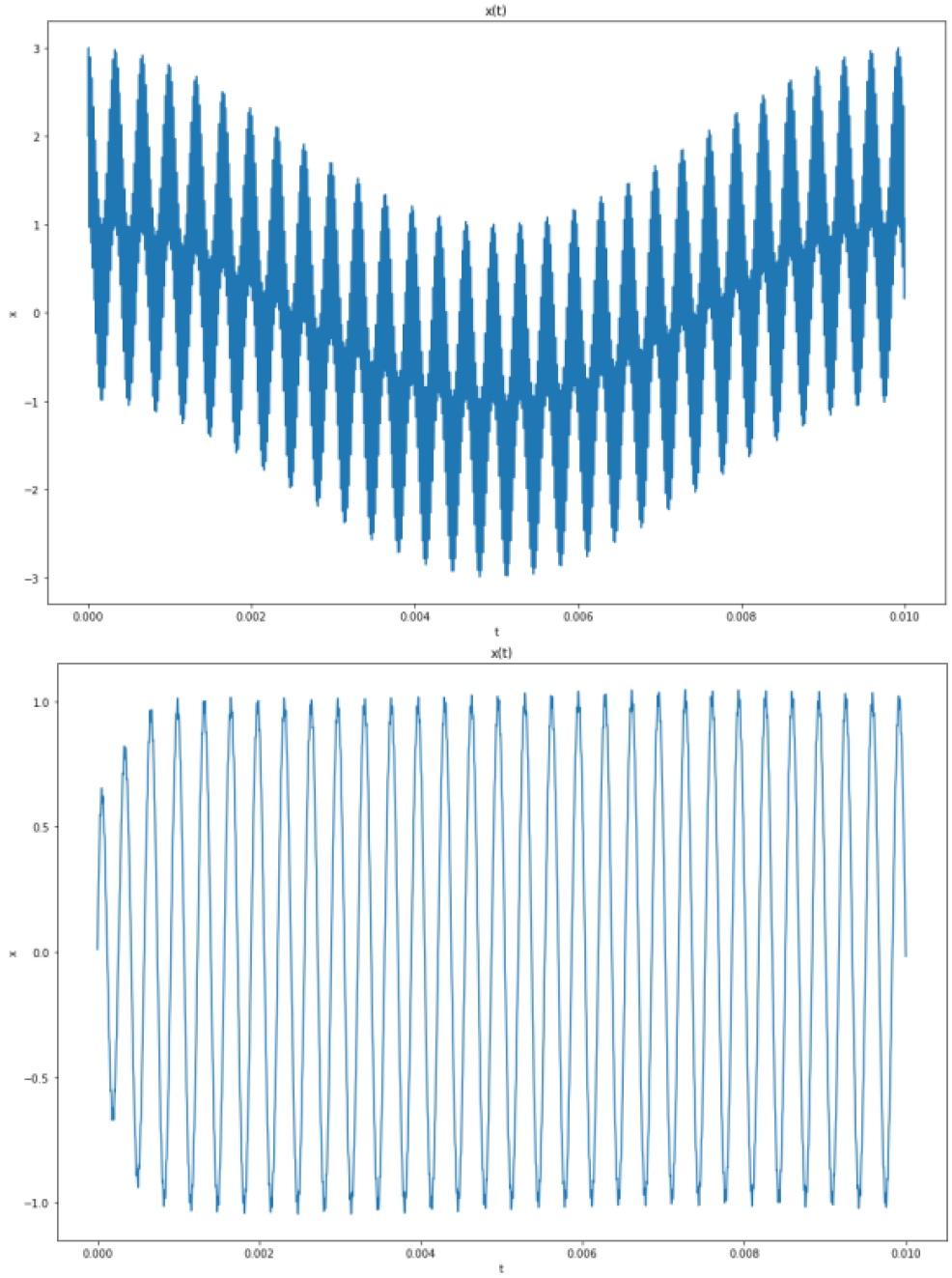

Figure 3: Signal Passed through z-domain transfer function

4 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.

Based on omega, we can see that the middle cosine function has the highest dB on the bode plot which means that part of the signal is having the greatest affect on the output whereas the other two signals do not.

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

scipy.signal.bilinear() is responsible for converting the transfer function to the z-domain. It takes the numerator, denominator, and frequency as inputs and its output is the same transfer function in the z-domain. scipy.signal.lfilter() then calculates what the new output of our signal is when passed through the z-domain transfer function.

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

We wouldnt be able to capture all three frequencies of the signal which would make it difficult to observe.

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

I did not have any problems following lab expectations.