Project Report for ECE 351

Lab 7: Block Diagrams and System Stability

Isaias Munoz

October 9, 2022

University of Idaho

Contents

1 Introduction

The purpose of this lab is to use block diagrams and use the transfer functions factor forms to say if a system is stable or not.

2 Methodology

2.1 Part 1

The first task was to solve manually for the poles and zeroes of the given functions and factor them nicely.

$$G(s) = (\frac{s+9}{(s^2-6s-16)(s+4)}$$

$$A(s) = (\frac{s+4}{(s^2+4s+3)}$$

$$B(s) = s^2 + 26s + 168$$

The factorization of them along with their poles and zeroes is below.

$$G(s) = (\frac{s+9}{(s-8)(s+2)(s+4)}$$

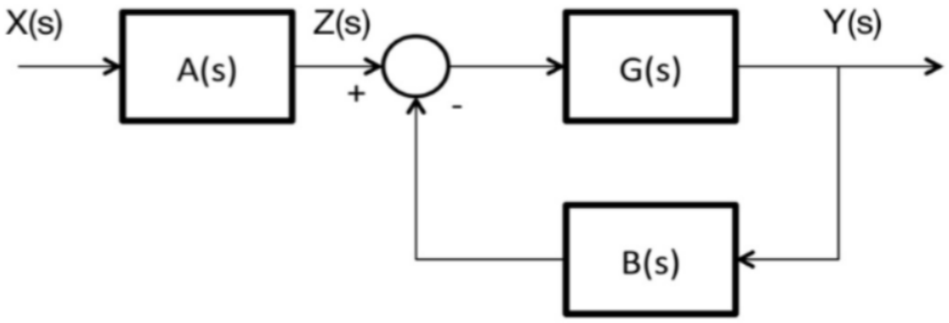
G(s) Poles: 8,-2,-4 and Zeroes: -9

$$A(s) = (\frac{s+4}{(s+1)(s+3}$$

A(s) Poles: -1,-3 and Zeroes: -4

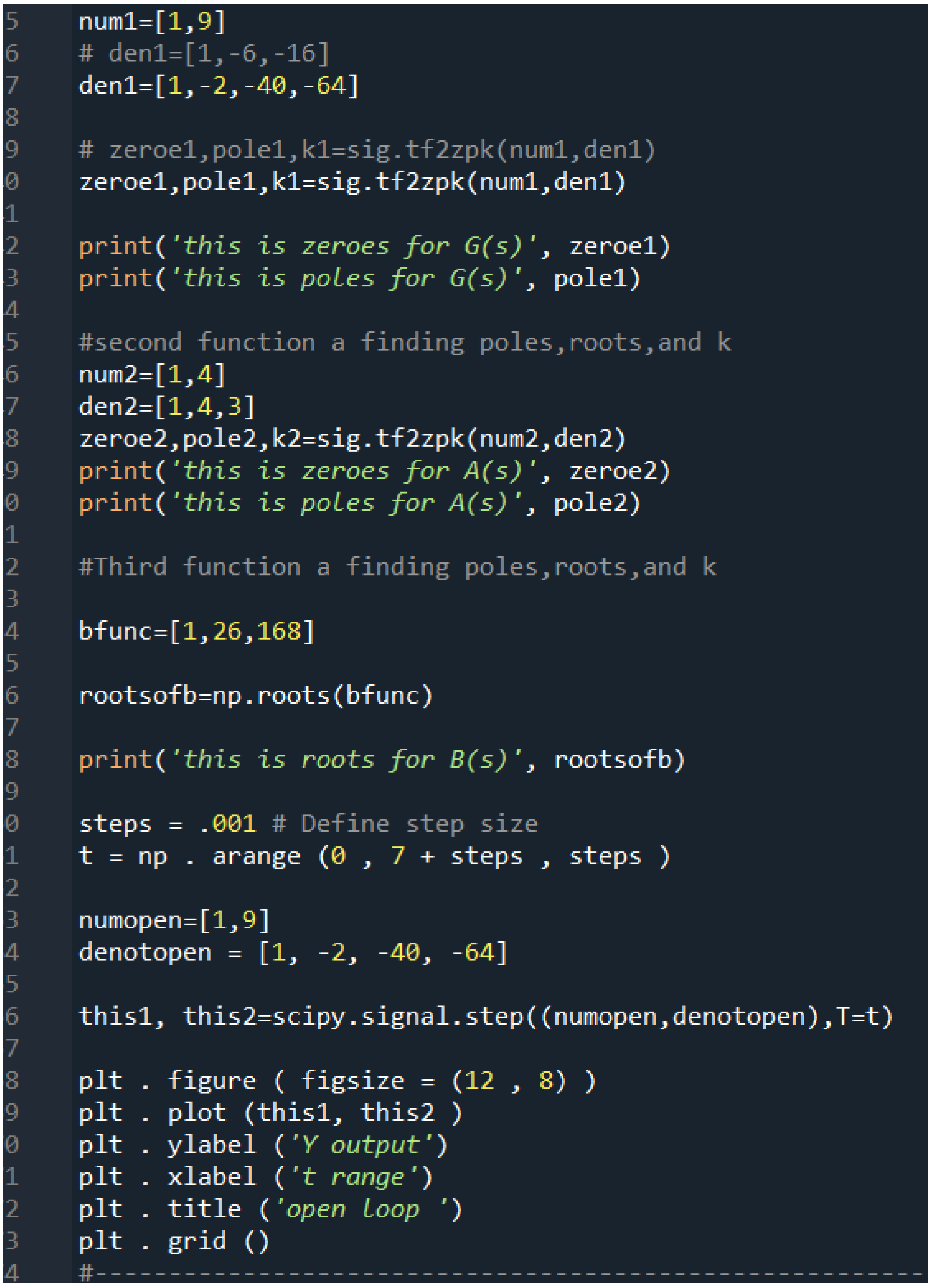
$$B(s) = (s+12)(s+14)$$

B(s) Poles: does not have any and Zeroes: -14,-12.



1

Figure 1: Block Diagram



```python
num1=[1,9]
# den1=[1,-6,-16]
den1=[1,-2,-40,-64]

# zeroe1,pole1,k1=sig.tf2zpk(num1,den1)
zeroe1,pole1,k1=sig.tf2zpk(num1,den1)

print('this is zeroes for G(s)', zeroe1)
print('this is poles for G(s)', pole1)

#second function a finding poles,roots,and k
num2=[1,4]
den2=[1,4,3]
zeroe2,pole2,k2=sig.tf2zpk(num2,den2)
print('this is zeroes for A(s)', zeroe2)
print('this is poles for A(s)', pole2)

#Third function a finding poles,roots,and k

bfunc=[1,26,168]

rootsofb=np.roots(bfunc)

print('this is roots for B(s)', rootsofb)

steps = .001 # Define step size
t = np . arange (0 , 7 + steps , steps )

numopen=[1,9]
denotopen = [1, -2, -40, -64]

this1, this2=scipy.signal.step((numopen,denotopen),T=t)

plt . figure ( figsize = (12 , 8) )
plt . plot (this1, this2 )
plt . ylabel ('Y output')
plt . xlabel ('t range')
plt . title ('open loop ')
plt . grid ()
#--------------------------------------------------------
```

Figure 2: Code for verifying zeroes,poles, and plotting open loop

Figure 2 shows making $num1, num2$ for $G(s)$ as the numerator and denominator array I continued in the same manner for $A(s)$ and $B(s)$. Since $B(s)$ was a little different I had to use *np.roots* to find it's zeroes. I then continued and moved onto finding the open loop transfer of the diagram provided. Below is the open loop Transfer function.

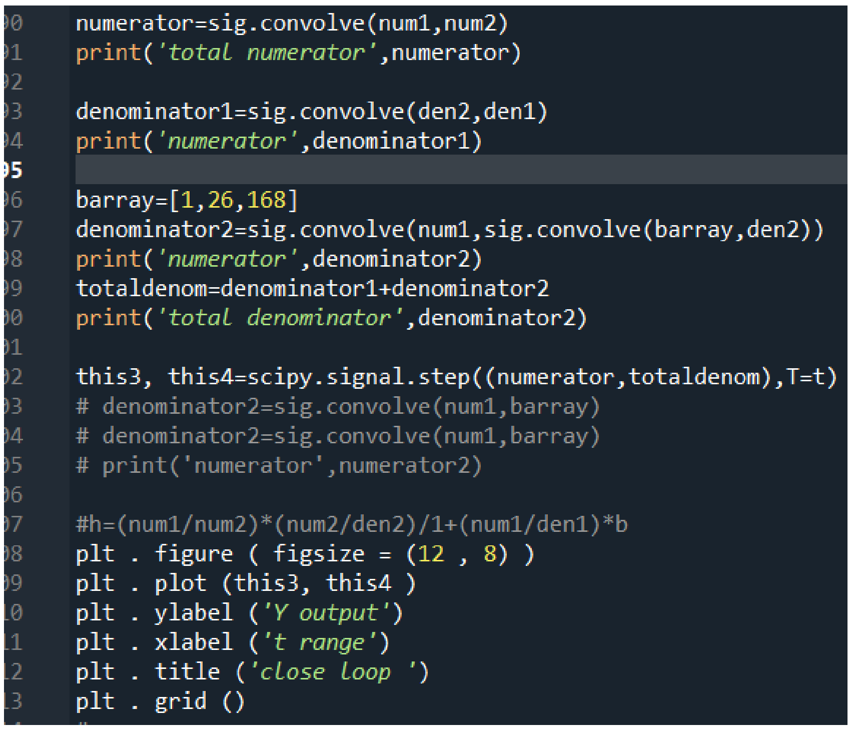
$$\frac{Y(s)}{G(s)} = (\frac{s+9}{(s-8)(s+2)(s+4)} \frac{s+4}{(s^2+4s+3)}$$

Looking at Figure 2 I convoluted it with the step response input and obtain a graph which can be found in the result sections for the open loop transfer function. It is important to notice that my open loop coefficient are simplified on Figure 2 line 3 and 4 because of the s+4 cancels out.

2.2 Part 2

Part 2 was to find the open loop transfer function and then use the same method in part 1 to graph. Below is the symbolic way I found out the open transfer function. The important point in this section was the way you found your numerator and denominator Figure 3 shows the code for implementing these tasks.

$$\frac{Y(s)}{G(s)} = (\frac{(num1)(num2)}{(den2)(den1) + (den2)(num1)(Barray)}$$



```
numerator=sig.convolve(num1,num2)
print('total numerator',numerator)

denominator1=sig.convolve(den2,den1)
print('numerator',denominator1)

barray=[1,26,168]
denominator2=sig.convolve(num1,sig.convolve(barray,den2))
print('numerator',denominator2)
totaldenom=denominator1+denominator2
print('total denominator',denominator2)

this3, this4=scipy.signal.step((numerator,totaldenom),T=t)
# denominator2=sig.convolve(num1,barray)
# denominator2=sig.convolve(num1,barray)
# print('numerator',numerator2)

#h=(num1/num2)*(num2/den2)/1+(num1/den1)*b
plt . figure ( figsize = (12 , 8) )
plt . plot (this3, this4 )
plt . ylabel ('Y output')
plt . xlabel ('t range')
plt . title ('close Loop ')
plt . grid ()
```

Figure 3: Code for open loop

As seen in Figure 3 I called the arrays and using the open loop transfer function I convoluted their respected numerators and denominators. Since I have three terms in the denominator I did a convolution of one and then used its results to convoluted it with the remaining one. Then I graphed it using the same method as Part 1, which the graph can be found under the result sections.

3 Results

3.1 Part 1

1

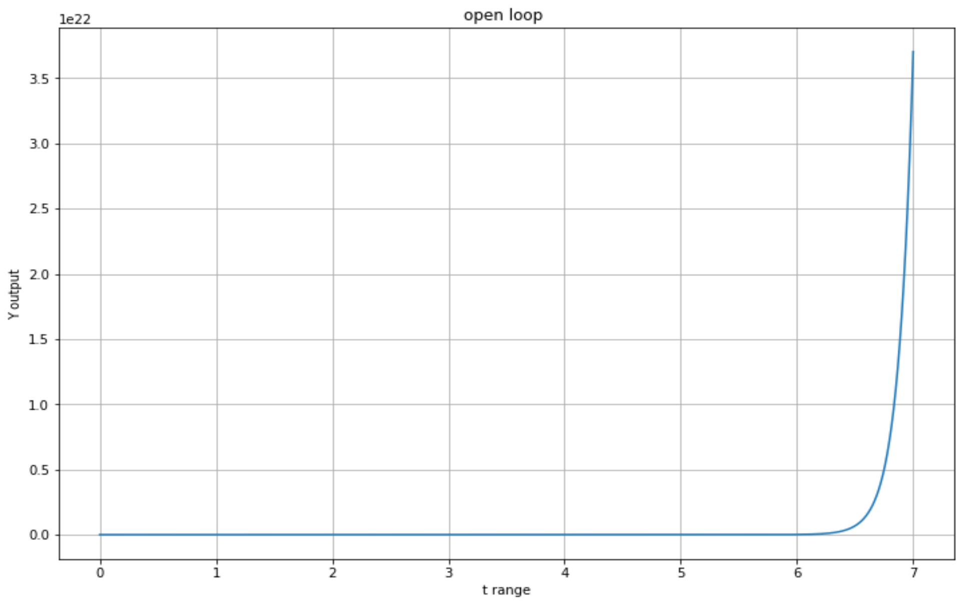
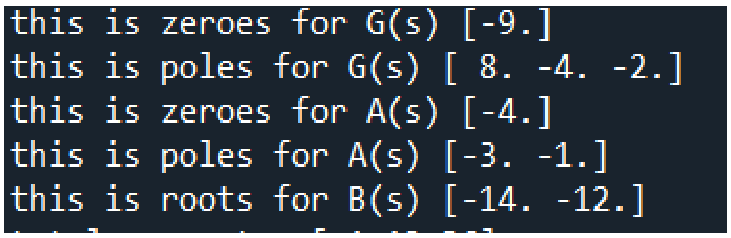


Figure 4: Open Loop Graph

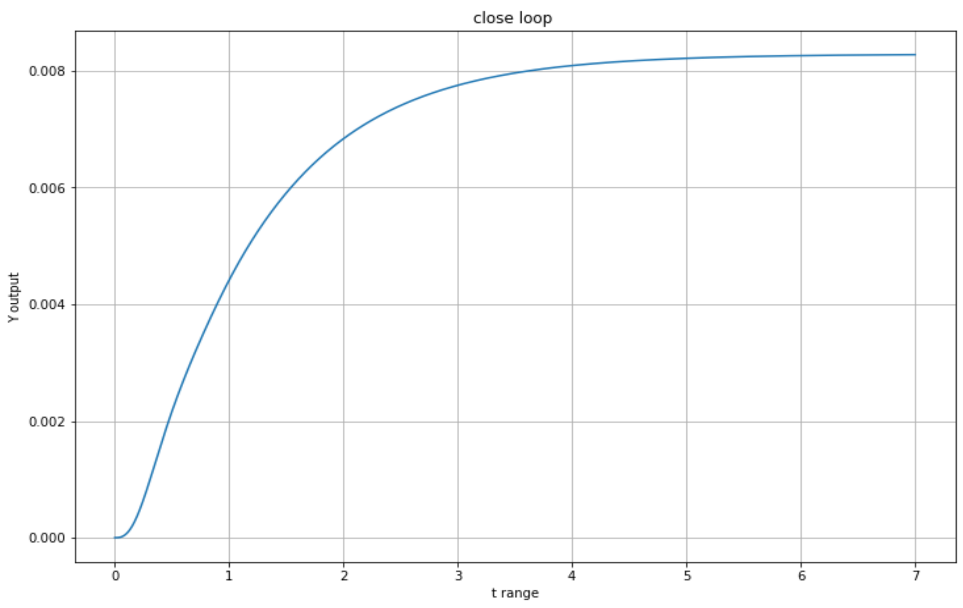
1



```
this is zeroes for G(s) [-9.]
this is poles for G(s) [ 8. -4. -2.]
this is zeroes for A(s) [-4.]
this is poles for A(s) [-3. -1.]
this is roots for B(s) [-14. -12.]
```

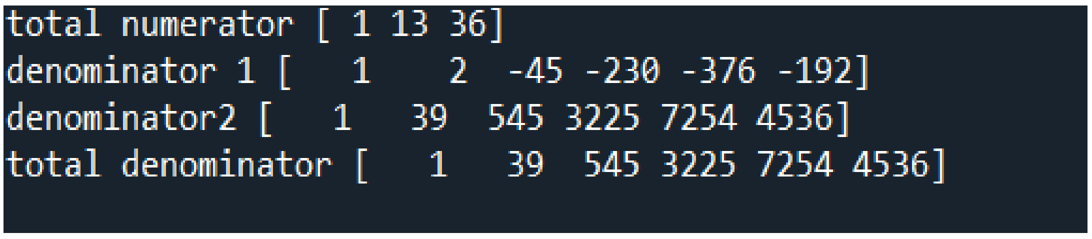
Figure 5: Verifying the poles and zeroes

3.2 Part 2



1

Figure 6: Close Loop Graph



```
total numerator [ 1 13 36]
denominator 1 [    1     2   -45  -230  -376  -192]
denominator2 [    1    39   545  3225  7254  4536]
total denominator [    1    39   545  3225  7254  4536]
```

1

Figure 7: Showing the numerator and denominator using python

4 Questions

1. In Part 1 Task 5, why does convolving the factored terms using scipy.signal.convolve() result in the expanded form of the numerator and denominator? Would this work with your user-defined convolution function from Lab 3? Why or why not?

I think since in Laplace domain convultion is multiplication and so the program just multiplies and prints out the results. I don't think it would work because it is not in the Laplace domain yet.

2. Discuss the difference between the open- and closed-loop systems from Part 1 and Part 2. How does stability differ for each case, and why?
In open loop the negative feedback is gone and a different transfer function is produced with different poles in the denominator meaning different stability then in the closed loop which everything is taken into account and stability could change or not.

3. What is the difference between scipy.signal.residue() used in Lab 6 and scipy.signal.tf2zpk() used in this lab?
Residue would determine only partial fraction expansion and tf2zpk does only factoring polynomials.

4. Is it possible for an open-loop system to be stable? What about for a closed-loop system to be unstable? Explain how or how not for each.
Yes I think so, one way to check is to see the denominator weathered it is positive poles meaning it is stable or just negative poles meaning unstable.

5. Leave any feedback on the clarity/usefulness of the purpose, deliverables, and expectations for this lab.
I wasn't sure were to answer some of the questions that were given throughout the tasks, so I answered them below.

For part 1 deliverable 3 The function was not stable due to a negative root in the denominator and the graph approves since it is an exponential and will continue to accumulate more to the right as time goes on, little by little but still advance to the right forever, making it unstable.

For part 2 deliverable 2 it is stable because looking at the graph it is reaching a steady state at some point.