1 Introduction

The goal of this lab is to become familiar with Laplace-domain block diagrams and use the factored form of the transfer function to judge system stability.

2 Equations

For this lab, we are given the following functions:

$$G(s) = \frac{s+9}{(s^2-6s-16)(s+4)} \tag{1}$$

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

$$B(s) = s^2+26s+168 \tag{3}$$

3 Methodology and Results

This lab consists of two parts:

- Part 1:
- Task 1: Type G(s), A(s), and B(s) in factored form, isolating the poles and zeros. Identify the poles and zeros of each function:

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

$$G(s) = \frac{s+9}{(s-8)(s+2)(s+4)} \tag{5}$$

Zeros:
s= -9
Poles:
s=-2 , s=8 , s=-4

$$A(s) = \frac{s+4}{s^2+4s+3} \tag{6}$$

$$A(s) = \frac{s+4}{(s+1)(s+3)} \tag{7}$$

Zeros:
s= -4
Poles:

s= -1 , s=-3

$$B(s) = s^2 + 26s + 168 \tag{8}$$
$$B(s) = (s + 12)(s + 14) \tag{9}$$

Roots:
s=-12 , s=-14

- Task 2: Use the sig.tf2zpk() function to check results from task 1:

```
	#%%  part 1 task 2

	numG = [1, 9]
	denG = [1 , -2 , -40, -64]
	z1, p1, k1 = sig.tf2zpk(numG, denG)
	print("\n\nFor G(s):")
	print("Zeros:\n", z1)
	print("Poles:\n", p1)
	print("Gain:\n", k1)

	numA = [1, 4]
	denA = [1 , 4 , 3]
	z2, p2, k2 = sig.tf2zpk(numA, denA)
	print("\n\nFor A(s):")
	print("Zeros:\n", z2)
	print("Poles:\n", p2)
	print("Gain:\n", k2)

	B = [1, 26, 168]
	rootsB = np.roots(B)
	print("\n\nFor B(s):")
	print("Roots:\n", rootsB)


```

- The output of this task is:

```
	For G(s):
	Zeros:
	[-9.]
	Poles:
	[ 8. -4. -2.]
	Gain:
	1.0

	For A(s):
	Zeros:
	[-4.]
	Poles:
        [-3.  -1.]
        Gain:
        1.0

        For B(s):
        Roots:
        [-14.  -12.]

```

- Task 3:

- Type and properly format the open-loop transfer function in factored form:

$$H(s) = G(s)A(s) \tag{10}$$

$$H(s) = \frac{s+9}{(s-8)(s+2)(s+4)}\frac{s+4}{(s+3)(s+1)} \tag{11}$$

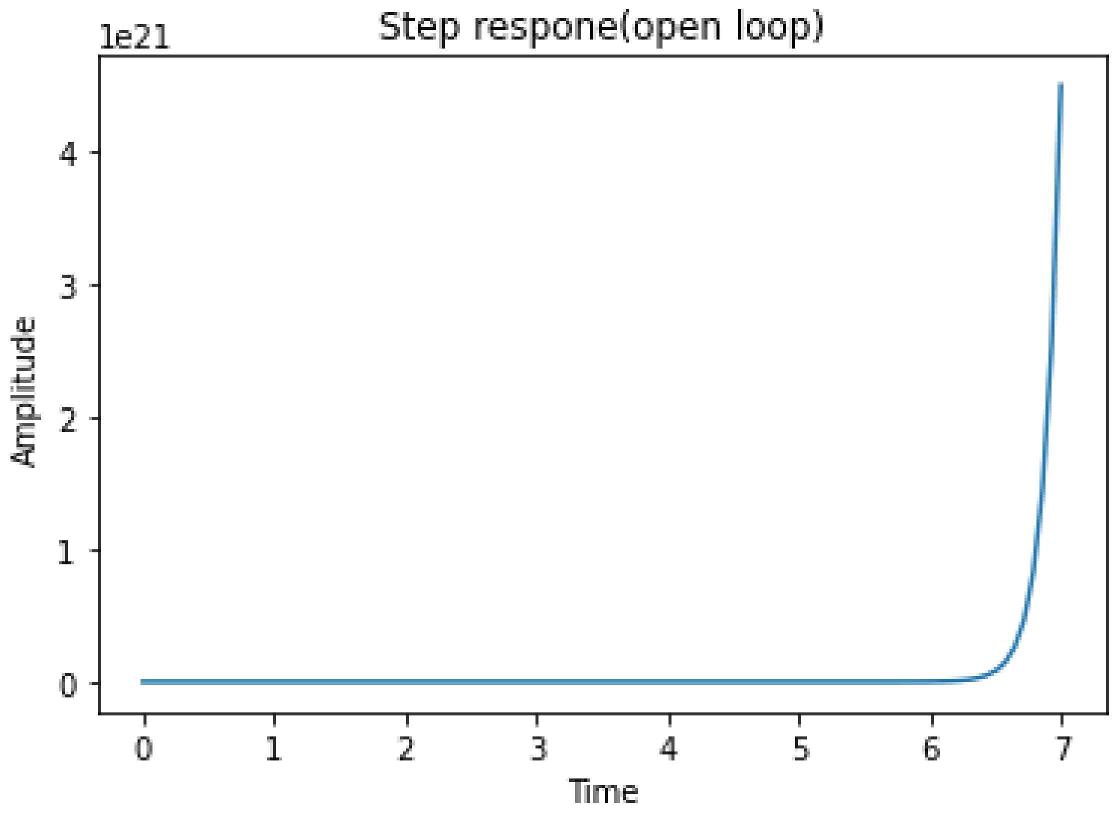
$$H(s) = \frac{s+9}{(s-8)(s+2)(s+3)(s+1)} \tag{12}$$

- Task 4: Is the open-loop response stable?:
  The open-loop response will unstable because the pole at s=8 of the system will be in the positive half of the plane.

- Task 5: Plot the step response of the open-loop transfer function:

```
        #%% part 1 task 5

        #OLTF: open-loop transfer function
        #CLTF: closed-loop transfer function

        OLTFNum = sig.convolve(numG, numA)
        OLTFDen = sig.convolve(denG, denA)

        tout, yout = sig.step((OLTFNum, OLTFDen))

        plt.plot(tout, yout)
        plt.xlabel("Time")
        plt.ylabel("Amplitude")
        plt.title("Step respone(open loop)")

```

The output of this code is:



- Task 5: Does your result from Task 5 support your answer from Task 4?
  Yes, it does. As it can be seen from the graph, the amplitude is going to infinity which means the system is unstable.

- Part 2:

- Task 1: Type the closed-loop transfer function for the given block diagram symbolically in terms of each block's numerator and denominator (i.e. numG, denG):

$$H(s) = \frac{G(s)A(s)}{1 + G(s)B(s)} \tag{13}$$

$$H(s) = \frac{\frac{numA}{denA}\frac{numG}{denG}}{1 + \frac{numG}{denG}(B)} \tag{14}$$

$$H(s) = \frac{(numA)(numG)}{denA(denG + (numG)(B))} \tag{15}$$

- Task 2: Using the scipy.signal.convolve() and scipy.signal.tf2zpk() functions to perform all arithmetic, nd numerical values for the total numerator and denominator, then type the resulting transfer function:

```
#%% part 2 task 2

print("\n\nPART 2")

CLTFNum = sig.convolve(numG, numA)
print("\nNumerator")
print(CLTFNum)

CLTFDen = sig.convolve(denG + sig.convolve(numG, B), denA)
print("\nDenominator")
print(CLTFDen)

z3, p3, k3 = sig.tf2zpk(CLTFNum, CLTFDen)
print("\nFor Close loop transfer function:\n")
print("Zeros:\n", z3)
print("Poles:\n", p3)
print("Gain:\n", k3)

```

- The output of this code is:

```
Numerator
[ 1 13 36]

Denominator
[   2   41   500 2995 6878 4344]

For Close loop transfer function:

Zeros:
[-9. -4.]
Poles:
[-5.16237064+9.51798197j -5.16237064-9.51798197j -6.17525872+0.j
-3.         +0.j          -1.         +0.j         ]
Gain:
0.5

```

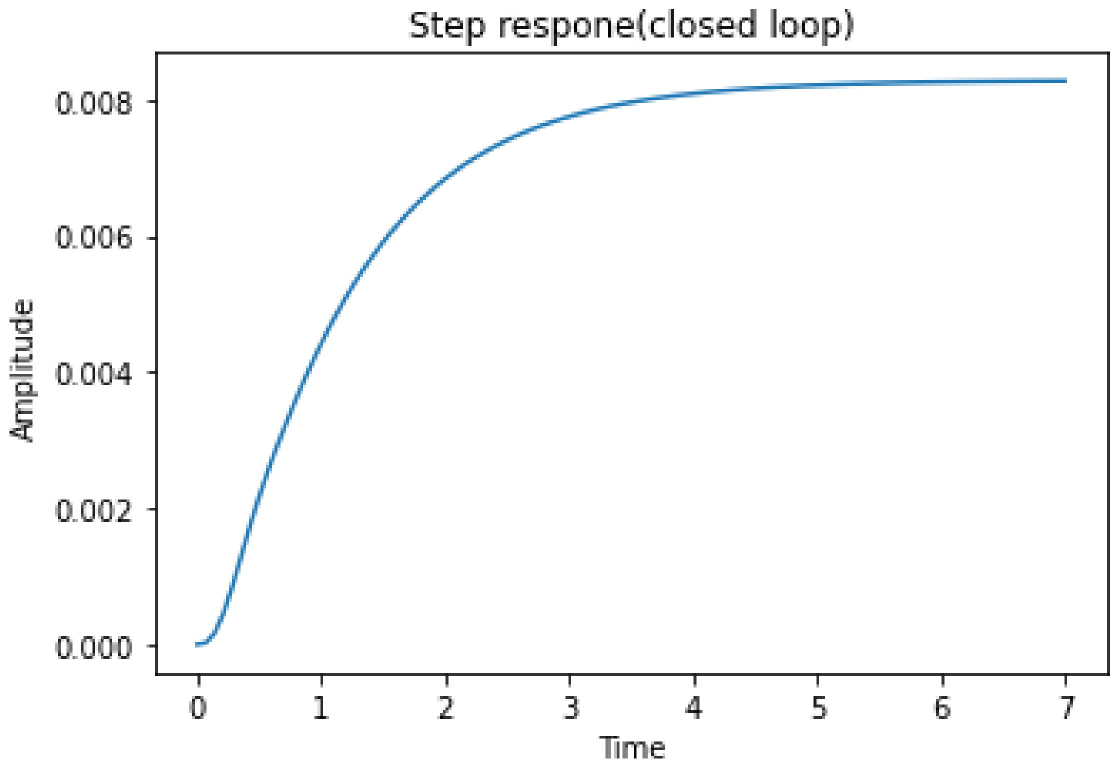
- Task 3: Using the closed-loop transfer function from Task 1, is the closed-loop response stable?
  Yes, the closed system is stable because all the poles are in the negative plane of the graph.

- Task 4:Plot the step response of the closed-loop transfer function using scipy.signal.step():

```
#%% part 2 task 4

tout, yout = sig.step((CLTFNum, CLTFDen))

```

```
    plt.figure()
    plt.plot(tout, yout)
    plt.xlabel("Time")
    plt.ylabel("Amplitude")
    plt.title("Step respone(closed loop)")
    plt.show()

```

The output of this code is:



- Task 5: Does your result from Task 4 support your answer from Task 3?
  Yes, it does. As can be seen from the graph, as time approaches infinity the signal will reach a stable state.

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?

   Since we have numerator and denominator, scipy.signal.convolve() would convolve the function even though it might be undefined. However, in lab three we did not have denominator, so our function can be undefined which means it does not work with the user-defined function from lab 3.

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?
   The first difference is in stability, as shown in figure 1, the system is rapidly increasing (unstable), but in figure 2, the system increased until a specific number and then becomes a straight line alone t-axis (stable). Also, all the poles in the closed-loop are in the left half plane, but we have poles in the open-loop are in the right half plan. Another difference is in the open loop, there is no output-feedback, but in the closed loop, there is an output-feedback.

3. What is the difference between scipy.signal.residue() used in Lab 6 and scipy.signal.tf2zpk() used in this lab?

   There is not difference. Both are used to find zeros, poles and gain.

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, it is possible for both. For the closed-loop system, if not all poles are in the left half plan it will be unstable. For the open-loop system, if all poles are in the left half plan it will be stable.

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

5 Conclusion

In this lab, we became familiar with Laplace-domain block diagrams and use the factored form of the transfer function to judge system stability.