Lab-2 Preliminary Work

Ayhan Okuyan

Electrical and Electronics Engineering Department, Bilkent University, 06800 Ankara, Turkey

1. Introduction

This laboratory assignment consisted of us designing a PI controller for a transfer function for which a first order transfer function defined as

$$G_p(s) = \frac{Ke^{-hs}}{\tau s + 1} \cong \frac{K}{\tau s + 1}\frac{-\frac{hs}{2}+1}{\frac{hs}{2}+1}$$

Here, we use the first order pade approximation. We have covered the K and τ values from Lab 1i in which we have found three different transfer functions to model the plant. Then, we have in MATLAB created the transfer functions in the form

$$G_c(s) = K_c\frac{-\frac{s}{z}+1}{s}$$

Where we have changed the value of z from -100 to -1 with a step size 0.1. Then for different root locus, we have investigated the one which would yield the real result for an optimized settling time, which is what is requested and what will be discussed throughout this lab report.

2. Laboratory Content

We have started the lab by taking the average of the three values for K and τ that we have found on the previous lab assignment.

$$K = \frac{23.76 + 47.42 + 2.99}{3} = 26.723$$

$$\tau = \frac{0.054364 + 0.031993 + 0.132173}{3} = 0.07284$$

Then we have found the transfer function equation for the first order system with the delay. Since we are using a PI controller, we multiply Gp with the Gc apart from the K value and treat the system as a P controller type system. Hence the transfer function becomes for the values that I have obtained from the previous lab and with a given h, which is equal to 0.01.

$$G = \frac{-0.13361655s + 26.7233}{0.0003642065s^2 + 0.778413s + 1}\frac{-\frac{s}{z}+1}{s}$$

Then, we have constructed the MATLAB code that would construct these root loci for separate z. Then what we do is to look for the root values that the rlocus() MATLAB function returns and save the root values that are the maximums in each locus. Then, we take the minimum of this maximum vector, hence we obtain the root value that is furthest form the closest ones. Hence, what I obtain is a vector of these root values. This is the MATLAB code that I have written to find this vector.

```
format long
close all
plnt_poles = [0.0003642065 0.0778413 1];
plnt_zeros = [-0.13361655 26.7233];
plnt = tf(plnt_zeros,plnt_poles);
int = -100:0.1:-1;
d_vals = [];
for z = int
    contr_poles = [1 0];
    contr_zeros = [-1/z 1];
    contr = tf(contr_zeros, contr_poles);

    [roots,ks] = rlocus(plnt*contr);

    max_re = max(real(roots));
    min_max_root = min(max_re);
    d_vals = [d_vals min_max_root];

end
```

Then I have plotted the minimum of maximum root values that I have found with respect to the corresponding z interval. The code and the corresponding graph are given below.

```
% plot the d-values
figure();
plot(int,d_vals)
title("d Values");
xlabel("z");
ylabel("min(max(real(roots)))");
% min=== z=-23.5 d=-57.49
```

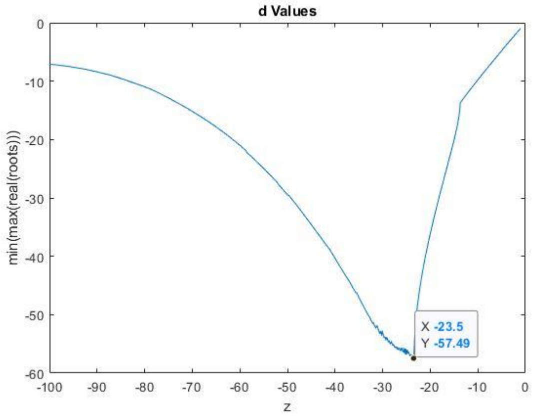

Fig. 1: The minimum of maximum root locus points with respect to their corresponding z values

Hence, from the graph we see that the point z equals -23.5 results with the global minimum. Hence, I picked the optimal value for my design as this value and plot its root locus to find the point that the complex conjugate roots that I recover are closer to the real axis and further from the imaginary axis. The code and the plot are given below.

```
% plot the root locus for z1
[minimum_d,min_index] = min(d_vals);
z1 = int(min_index);
contr_poles = [1 0];
contr_zeros = [-1/z1 1];
contr = tf(contr_zeros, contr_poles);
figure();
```

```
rlocus(contr*plnt);
title("Root Locus for z1");
```

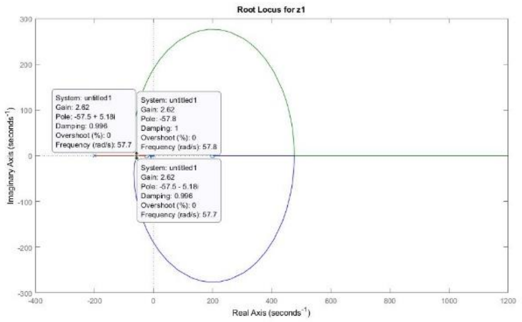

Fig. 2: The Root Locus for z=-23.5

Here we observe the root values that corresponds to the d value that we have found. Here, we observe that the damping is nearly equal to one and the overshoot is 0%. The design criteria suggest that the settling time is minimum. Hence, I use the point presented in the figure that fit the criteria. Here, I note that the gain, which corresponds to the Kc value which is 2.62.

Furthermore, the laboratory work suggests us to define two other z values z2 and z3 for which they would become z1/2 and z1/3 respectively. Hence, these values become as given.

$$z = -23.5$$

$$z_2 = -11.75$$

$$z_3 = -7.8\overline{3}$$

Then, what I do is to look at the Fig1 and find approximate points for these z points and look at their corresponding values. The figure is given below.

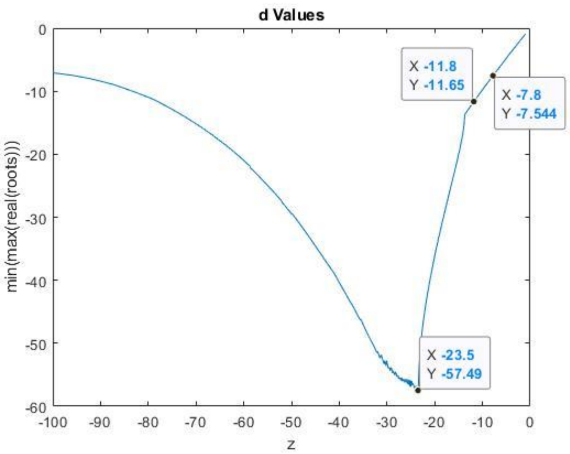

Fig. 3: The minimum of maximum root locus points with respect to their corresponding z values with z1, z2 and z3 marked.

Hence, we draw the root locus for z2 and z3 in order to find the point with the given roots. The MATLAB code written to plot the root locus for z2 and z3 is given below.

```
% plt the root locus for z2 and z3
contr_poles = [1 0];
contr_zeros = [-1/z2 1];
contr = tf(contr_zeros, contr_poles);
figure();
rlocus(contr*plnt);
title("Root Locus for z2")
contr_poles = [1 0];
contr_zeros = [-1/z3 1];
contr = tf(contr_zeros, contr_poles);
figure();
rlocus(contr*plnt);
title("Root Locus for z3");
% gain for z1=2.62, z2=1.07, z3=0.677
```

The root locus for z2 is given below with the approximate pole locations for each branch.

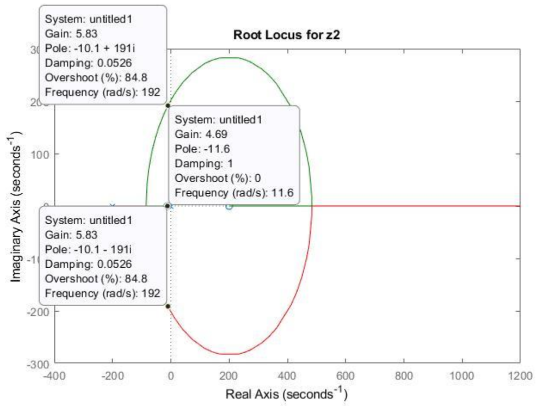

Fig. 4: The Root Locus for z=-11.8 which is nearly -11.75 since we use 0.1 decimal precision.

Here, we see that the gain is given as 4.69 and note this gain. Then, we move onto the root locus for the third system. The plot is below.

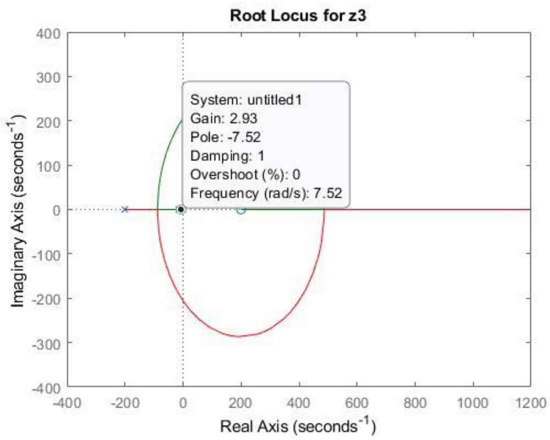

Fig. 5: The Root Locus for z=-11.8 which is nearly -11.75 since we use 0.1 decimal precision.

Here we note that the gain to be nearly equal to 2.93.
The next part requires us to design these circuits on a Simulink model. Then we integrate the K values that we have acquired onto the model and observe the outputs on one graph. The constructed closed loop system for the three distinct z values are given below.

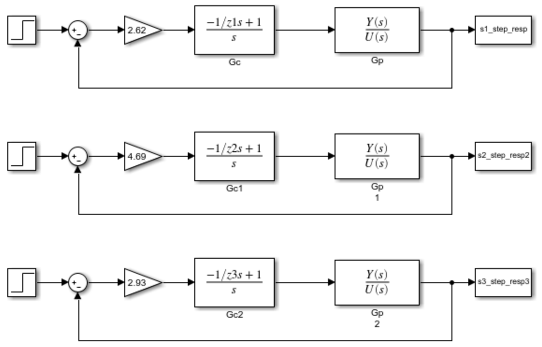

Fig. 6: The Simulink Model

This is a closed loop feedback system with a 10u(t) step input. Here, I have used the variables directly from my MATLAB workspace since I run the script and the simulation at the same time. Then, I have written the MATLAB code below to plot these step responses. The code and plot are given.

```
% plot the step responses
figure();
plot(s1_step_resp);
hold on
plot(s2_step_resp2);
plot(s3_step_resp3);
title("Step responses for z1,z2 and z3");
legend("Step Response for z1","Step Response
for z2","Step Response for z3");
```

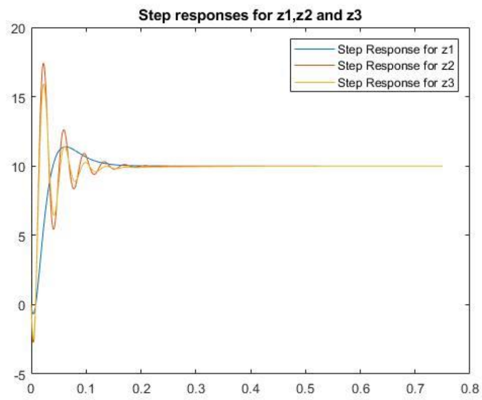

Fig.7: The Step Responses of the Closed Loop Systems in a time interval 0-0.75s

Here we see the responses of the three different closed-loop systems. In all three of our cases, it seems that the settling time is around 0.1~0.2 seconds, which is the indication of a successful design. However, the overshoot has gotten bigger and the number of oscillations increased with the z2 and z3 step responses.

3. Conclusion

This lab consisted of us designing a PI controller for a predefined transfer function which required us to aim for the optimal settling time. Hence, we have looked for the optimal controller zero value to use from a predefined interval -100:0.1:-1. Then, we have designed the controllers for the three distinct z values, one being the one that we have optimized and the others being the half and one third of that value. Furthermore, we have designed these values on Simulink and then discussed on the differences.

REFERENCES

1. Oğuz Yeğin, Lab-2 Preliminary Work Guide, April 2019