Maxim Shinskiy

CE269 – Control theory and practice

Table of Contents

[Part A 2](#_Toc35805299)

[Resources 2](#_Toc35805300)

[Picutes 2](#_Toc35805301)

[Calculations 4](#_Toc35805302)

[Discussion 4](#_Toc35805303)

[Steady-State Error 4](#_Toc35805304)

[Relative Stability 5](#_Toc35805305)

[Part B 5](#_Toc35805306)

[Resources 5](#_Toc35805307)

[Discussion 6](#_Toc35805308)

[Steady-State Error 6](#_Toc35805309)

[Relative Stability 6](#_Toc35805310)

[Part C 7](#_Toc35805311)

[Resources 7](#_Toc35805312)

[Calculations 10](#_Toc35805313)

[Discussion 10](#_Toc35805314)

[Appendix 12](#_Toc35805315)

# Part A

## Resources

### Picutes

Model of a closed loop control system with description

A screenshot of a cell phone

Description automatically generated

Scope output

A picture containing wall

Description automatically generated

Stability:

Change of response from R(s):

Gain = 0.5 Gain = 1

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Gain = 1.5 Together

A close up of a map

Description automatically generatedA close up of text on a white background

Description automatically generated

Change of response from D(s):

Gain = 102 Gain = 127

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Gain = 152 Together

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

### Calculations

Command to find the polynomial of a denominator of Gp(S) function

>> poly([0 -1 -2 -5 -5 -10])

ans = 1 23 187 665 1000 500 0

## Discussion

### Steady-State Error

In Part A of the assignment we are asked to model a system with disturbance applied at time t = 150 and with input of d(t) = 0.25r(t).

To calculate steady-state error values due to reference input and due to disturbance, we use formula.

, where is a position error constant and . To find error constant we need to inspect transfer functions and . For reference step input, , the error is caused by transfer function , for disturbance, , by transfer function . In both cases the denominator is multiplied by s. Therefore, as s approaches 0, approaches infinity. This makes and hence the .

### Relative Stability

Scope output shows damped sinusoid that exponentially decays into a reference input signal that is a line. As system has two inputs applied at different points (reference input at t = 0, disturbance at t=150), we can split it into two and change the gain to see the difference of stability of the system. From this we can judge that the system is underdamped in both cases.

|  |  |  |  |
| --- | --- | --- | --- |
| Gain | Settling Time (s) | OS (%) | Stability Comments |
| 0.5 | 773.02 | 89.6 | %OS highest, settling time highest |
| 1 | 64.94 | 73.9 | Original value |
| 1.5 | 39.12 | 76.8 | %OS lowest, settling time lowest |

With the gain of 0.5 the settling time and percentage overshoot increased dramatically. It took 773 seconds to settle to 98% of its final value and 89.6 overshoot. The final value is the same as initial because .

:

|  |  |  |  |
| --- | --- | --- | --- |
| Gain | Settling Time (s) | OS (%) | Stability Comments |
| 102 | 123.81 | 82.87 | %OS highest, settling time lowest |
| 127 | 65.63 | 91.48 | Original Value |
| 152 | 45.75 | 99.25 | %OS lowest, settling time highest |

I changed the value by 25 to see the behaviour of the system. Lower gain has lower settling time but higher %OS and opposite for bigger than normal gain.

# Part B

## Resources

Model of a closed loop control system with description

A screenshot of a cell phone

Description automatically generated

Scope output

A picture containing small, sitting, green, lot

Description automatically generated

## Discussion

### Steady-State Error

In part B the input was changed from unit-step to unit-ramp applied at t = 0 and disturbance applied at t = 50.

In this part to calculate steady-state error values due to reference and due to disturbance, we use formula.

, where is a velocity error constant and . To find error constant we need to inspect transfer functions and . For reference ramp input, , the error is caused by transfer function , for disturbance, , by transfer function . In both cases the denominator is multiplied by s, just. Therefore, as s approaches 0, approaches infinity. This makes and hence the . Which is the same as in part A.

### Relative Stability

In the same way as in part A I changed the gain to see the difference in settling time and percentage overshoot, I also took the same values for gain as before.

Gc(s):

|  |  |  |  |
| --- | --- | --- | --- |
| Gain | Settling Time (s) | OS (%) | Stability Comments |
| 0.5 | 773.2 | 89.6 | %OS highest, settling time highest |
| 1 | 64.94 | 73.6 | Original value |
| 1.5 | 39.12 | 76.8 | %OS lowest, settling time lowest |

Gp(s):

|  |  |  |  |
| --- | --- | --- | --- |
| Gain | Settling Time (s) | OS (%) | Stability Comments |
| 102 | 123.81 | 82.87 | %OS highest, settling time highest |
| 127 | 65.63 | 91.48 | Original value |
| 152 | 45.75 | 99.25 | %OS lowest, settling time lowest |

In part B the result occurred to be the same as before. The change in input from step to ramp didn’t change the response of the system.

# Part C

## Resources

Root Locus; K = 1

A close up of a map

Description automatically generated

Root Locus; K = 7.5376; lines represent damping ratio of 0.504, pos = 16%

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generatedA screenshot of a cell phone

Description automatically generated

### Calculations

Finding the numerator and denominator

>> poly([-0.071+6.25j -0.071-6.25j])

ans = 1.0000 0.1420 39.0675

>> ans \* 0.117

ans = 0.1170 0.0164 4.5709

>> poly([-0.047 -0.262+5.1j -0.262-5.1j ])

ans = 1.0000 0.5710 26.1033 1.2257

Plotting root locus and displaying the transfer function

K = 1;

Gc = tf([0 8], [1 2]);

Gsm = tf([0.117 0.0164 4.5709], [1 0.571 26.1033 1.2257]);

G = K\*Gc\*Gsm;

display(G);

rlocus(G);

0.936 s^2 + 0.1312 s + 36.57

G = ---------------------------------------------

s^4 + 2.571 s^3 + 27.25 s^2 + 53.43 s + 2.451

## Discussion

To plot root locus in MatLab I used code to define Gc and Gsm transfer functions. Then I plot them using the line rlocus(G);. This will draw the graph. For gain K = 1 the zeros are and the poles are , , 0.047.

When system is marginally stable it’s not stable nor unstable, so the output doesn’t come to the steady state value neither goes away from it. To find the K I used the Routh-Hurwitz criterion.

My characteristic equation is: , after inserting G, the equation becomes:

Plotting the Routh table:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 27.25+0.936K | 2.451+36.57K |
|  | 2.571 | 53.43+0.1312K | 0 |
|  |  |  | 0 |
|  |  | 0 | 0 |
|  |  | 0 | 0 |

First Column of a table:

|  |  |
| --- | --- |
|  | 1 |
|  | 2.571 |
|  |  |
|  |  |
|  |  |

From the table it is clear, that for all terms except for , if K > 0 the value is greater than 0 too. Therefore, for system to be marginally stable, has to be equal to 0.

The lowest value for gain is 7.5376, that means that this is the minimum value for which the system becomes marginally stable and maximum will be 387.6815.

Therefore, gain K = 7.5376.

From the root locus we can find the gain for the certain percentage overshoot. To plot root locus I used MatLab commands:

rlocus(G);

pos = 16;

z = -log(pos/100)/sqrt(pi^2+[log(pos/100)]^2);

sgrid(z, 0);

for percentage overshoot

From the pictures its seen that gain

The coordinates are

In the Simulink I built a system with corresponding gain of 2.87. Then using linear analysis tool I found the actual response for the step input. After I used MatLab to analyse the response using stepinfo(linsys1), the result is:

ans =

RiseTime: 0.6930

SettlingTime: 5.1888

SettlingMin: 0.9042

SettlingMax: 1.1360

Overshoot: 16.2558

Undershoot: 0

Peak: 1.1360

PeakTime: 1.4431

To find final steady state value we need to find the steady state error first. For this we analyse the transfer function

We can see that the input is r(t) = u(t) so when, , position error is what G is when S approaches 0. From the function we can see that . To find the error we use formula , insert and we get = 0.0228. So the final value = initial – error = 1 – 0.0228 = 0.9772.

# Appendix