Introduction to MATLAB and Simulink Lab

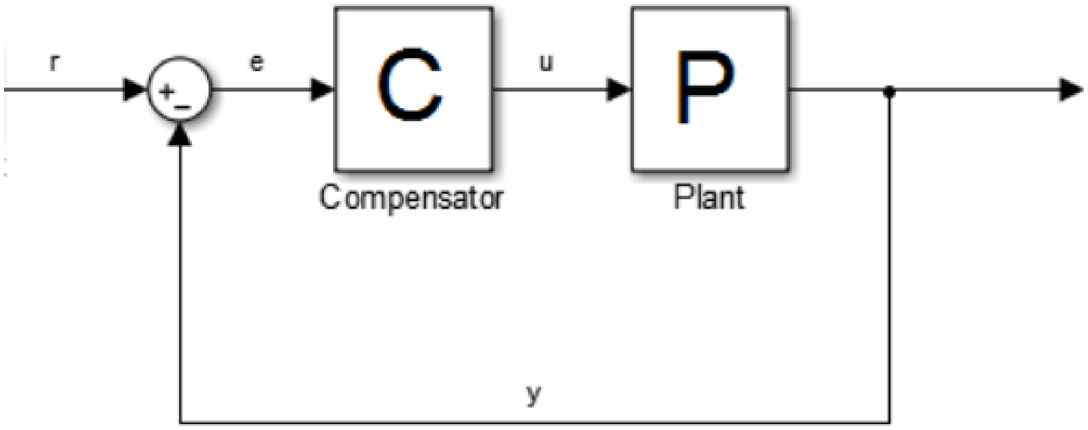
Submitters:

Or Shaul

Saray Sokolovsky

Pre Lab

For the following system:



$$P(s) = \frac{1}{s^2(s+3)}$$

a) The transfer function has no zeros. It has double pole at Z=0, and a single pole at -3. From the Roth criterion for stability, the system is unstable because there is a pole at 0. In addition, by RL sentence, one of the conditions is if there are coefficients in the numerator that are negative or zero the system is unstable, on our plant- the coefficient of S^1 here is 0.

Therefore, the plant is marginally stable.

b) By definition, D is equals to the limit of the transfer function at infinity.

In this question the limit at infinity of the transfer function is 0, Therefore

$D = 0.$

The state space representation matrices are:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -3 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

The matrix A we got is an upper triangular matrix and therefore, the eigenvalues are 0 and -3. The zeros of the given transfer function are the eigenvalues of matrix A we found.

c) given P(s) and $C(s) = \frac{1.15s+1}{0.15s+1}$, the closed-loop transfer function is given by:

$$T(s)=\frac{P(s)C(s)}{1+P(s)C(s)}=\frac{\frac{1}{s^2(s+3)}\cdot\frac{1.15s+1}{0.15s+1}}{1+\frac{1}{s^2(s+3)}\cdot\frac{1.15s+1}{0.15s+1}}=\frac{\frac{1.15s+1}{s^2(s+3)(0.15s+1)}}{1+\frac{1.15s+1}{s^2(s+3)(0.15s+1)}}$$

$$=\frac{\frac{1.15s+1}{s^2(s+3)(0.15s+1)}}{\frac{(s^2(s+3)(0.15s+1)+1.15s+1)}{s^2(s+3)(0.15s+1)}}$$

$$=\frac{1.15s+1}{s^2(s+3)(0.15s+1)+1.15s+1}=\frac{1.15s+1}{0.15s^4+1.45s^3+3s^2+1.15s+1}$$

The zero of T(s) is the root of the numerator: $z_1=-0.86$

The poles of T are the roots of the denominator: $p_1=-2.5195, p_2=-6.9143, p_{3,4}=-0.11642\pm 0.6075i$

A system is stable if all its poles have negative real parts. Therefore, we can say that the closed loop transfer function is indeed stable.

d)

Root- Locus plot:

graphical approach used to determine the stability of a control system as a system parameter varies. The root locus plot shows the paths of the poles of the closed-loop transfer function as the gain K varies from 0 to ∞.

We are starting by marking the zeros and poles of the loop gain (PC) then, plotting the RL to show how the closed-loop poles move as a function of the loop gain (k). After that, we should check if any branch of the RL plot crosses into the right half plain of the s-plane. If not, the closed loop system is stable for any gain. If it does crosses, this means that for certain values of K, the poles of the Sensitivity are on the right half plane, thus making the feedback loop unstable.

Nyquist plot:

Nyquist plot helps us determine the number of right-side poles of the Sensitivity (S).

Meaning $Z=P-N$

Let N be the winding number of the Nyquist curve around -1.

Let P be the number of poles of the loop gain in C^+ .

The number of poles of S in C^+ is $P-N$. For S to be stable we need to require $N=P$.

e) The critical gain (K_{cr}) in a RL plot is the value of the system gain K at which the system transitions from stable to unstable. Meaning that the pole moves from the left-hand side of the s-plane (the stable region) to the right-hand side of the s plane. K_{cr} represents the gain at which a system becomes marginally stable. For gain less than the critical gain, the system is stable. And for gain larger than the critical gain, the system becomes unstable.

In Lab

Introduction

In this experiment, we used MATLAB and Simulink to implement a system and get its step responses. We used functions such as "Bode" to get the bode plot, "Nyqlog"- a given function for plotting the Nyquist plot), "feedback" to get the transfer function of the closed loop and other functions. In Simulink we built block diagrams of the system as demanded. We implemented the demanded systems based on the prelab report and simulated the responses to a step function input.

3.1. Open Loop

For the following plant $P(s) = \frac{1}{s^2(s+3)}$ we plotted the open loop response for a step input:

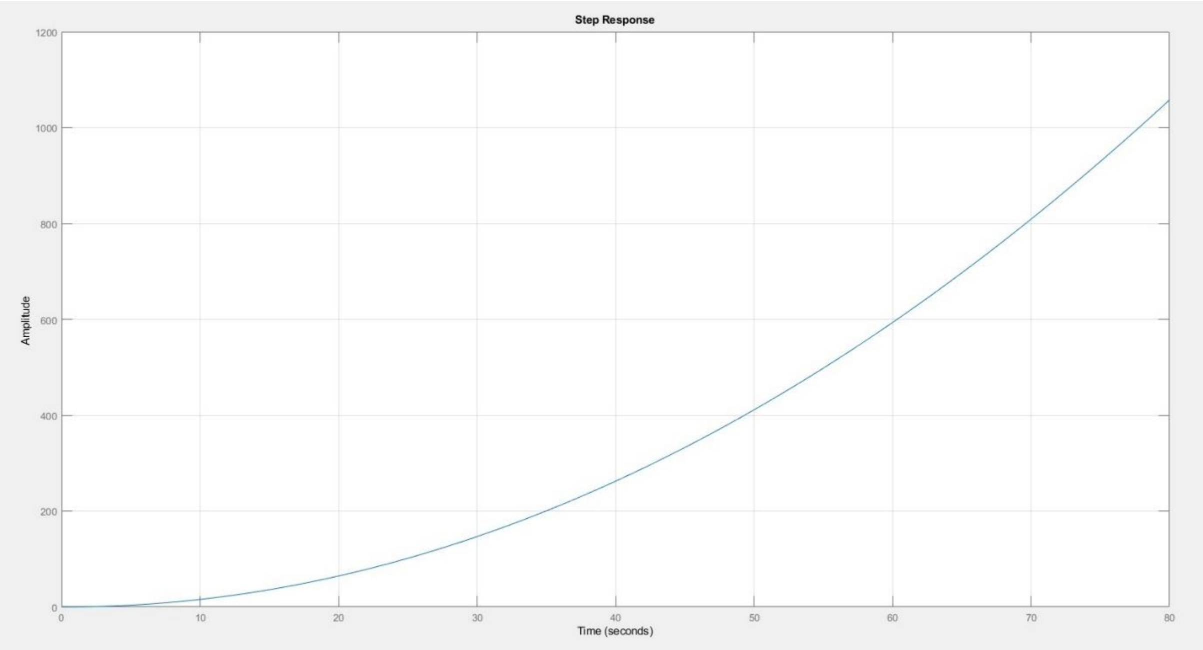


Figure 1 - step response of the open loop

As we can see from the graph, the response goes to infinity (meaning its not not bounded) therefore it is not stable. We predicted that from the prelab part – we have two poles in zero.

Converting the transfer function into a state-space representation the outcome from the MATLAB:

```
A matrix:
    -3     0     0
     1     0     0
     0     1     0

B matrix:
     1
     0
     0

C matrix:
     0     0     1

D matrix:
     0
```

Figure 2 - State space representation

The results aren't the same as in the prelab but that there is number of ways to (infinite) to realize a given transfer function and we chose a specific representation of the matrix. Also, we opened the equations and did get the same results.

Finding the eigenvalues of the A matrix:

```
Eigenvalues:
     0
     0
    -3
```

Figure 3 - Eigenvalues of the matrix A

And indeed, the eigenvalues obtained in MATLAB are the same as those in the Prelab.

Completing the state-space representation's necessary matrices, and convert the plant back to a transfer-function form got us:

```
den =

     1     3     0     0


A =

    -3     0     0
     1     0     0
     0     1     0


B =

     1
     0
     0


C =

     0     0     1


D =

     0


num1 =

     0     0     0     1


den1 =

     1     3     0     0
```

Figure 4 - convert the plant back to a transfer-function form

We can see that the result matches the original transfer function.

3.2. Closed Loop

For the following compensator $C(s) = \frac{1.15s+1}{0.15s+1}$ the closed loop of transfer-function $\frac{y(s)}{r(s)}$ with a step input:

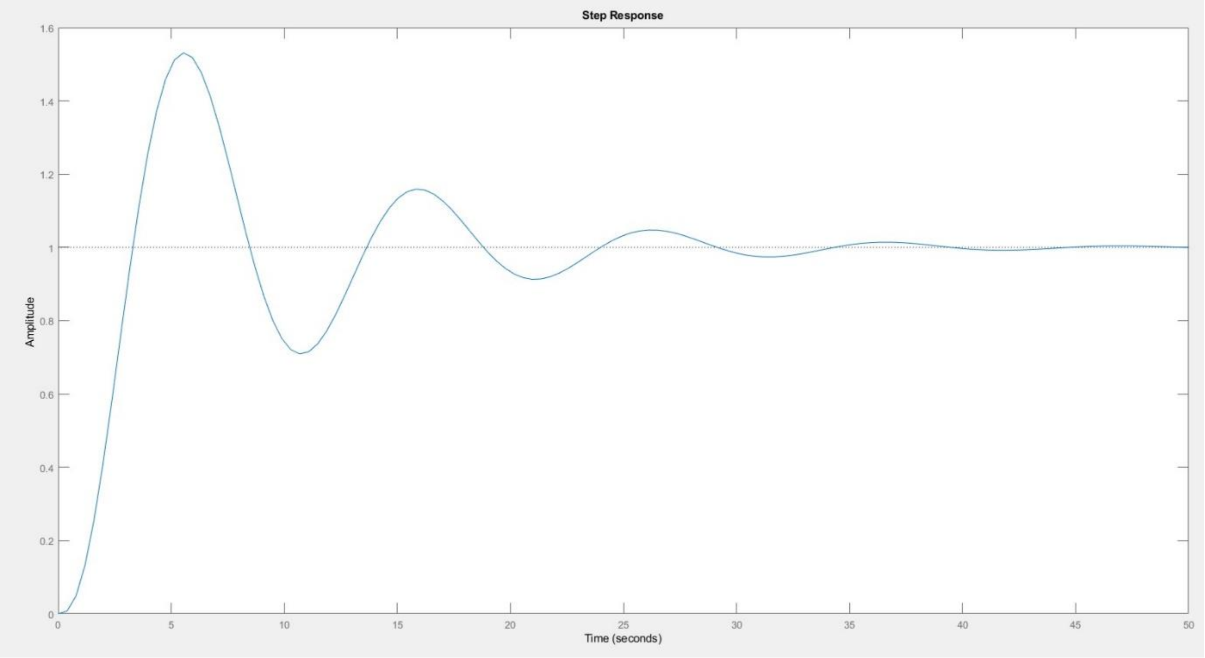


Figure 5 - step response of the closed loop

We can see that the response settles on a finite value after a finite amount of time meaning that the system is indeed stable.

3.3. Plots

A Bode plot of the open-loop system $C \cdot P$:

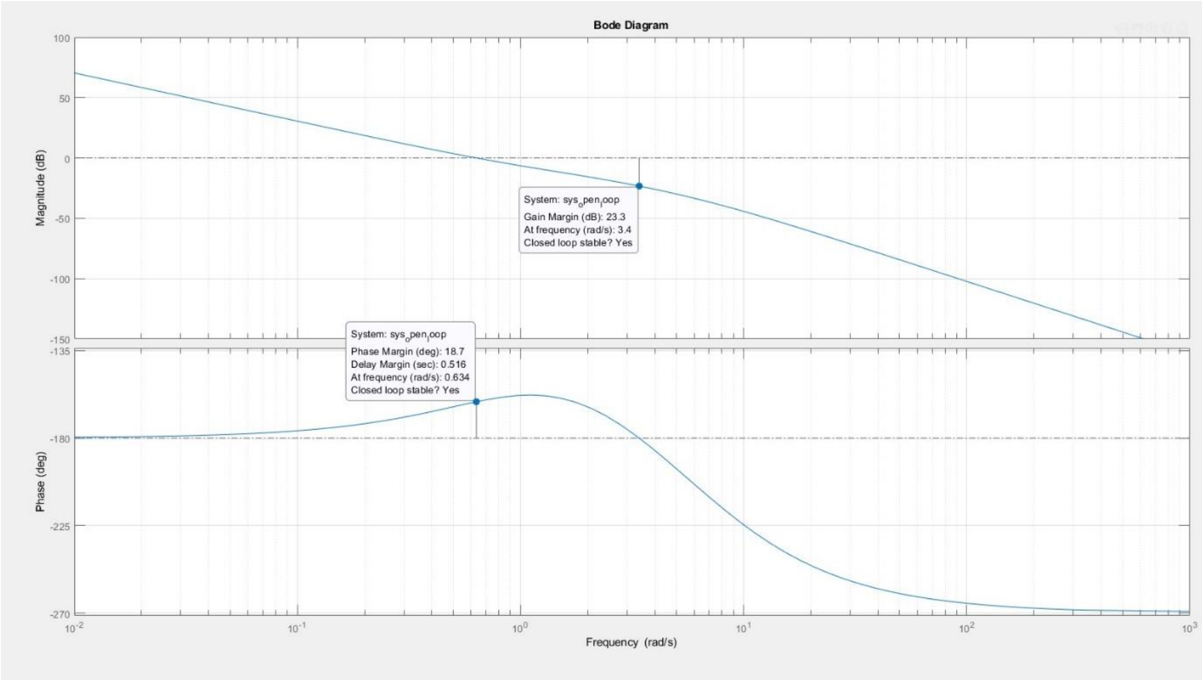


Figure 6 - Bode plot of the open loop CP

Root-locus plot for the system, together with the compensator:

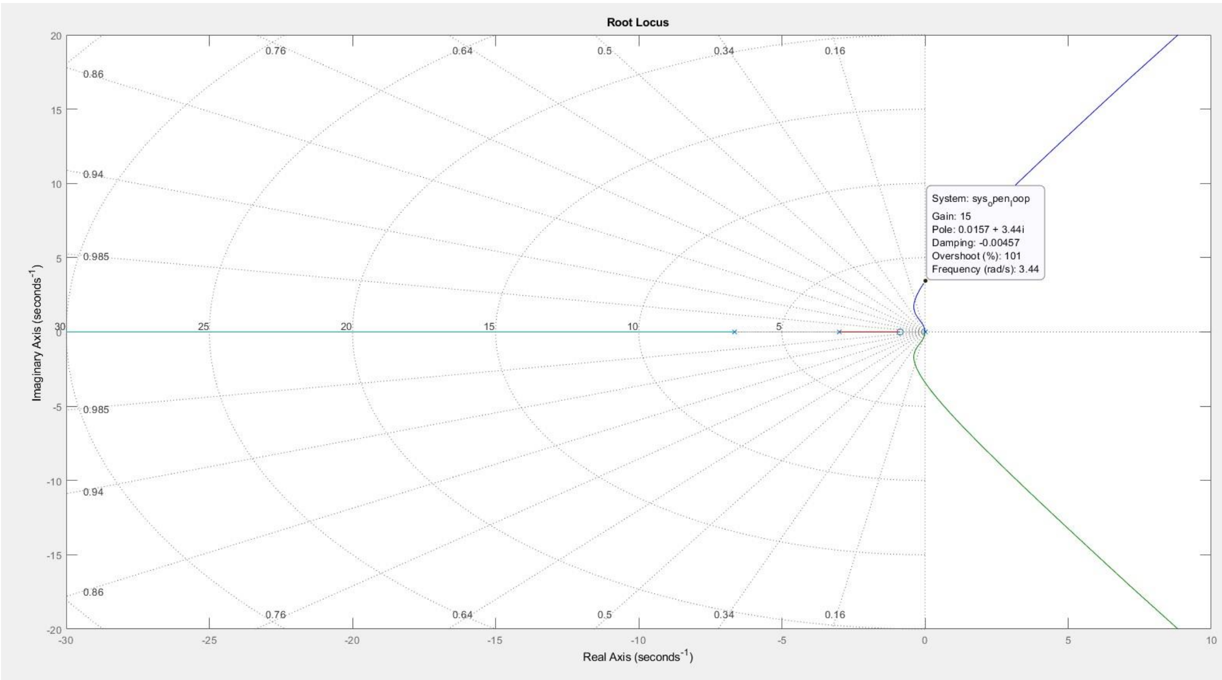


Figure 7 - RL of the system

Critical is the gain at which the system becomes unstable. From the root locus plot, it is the gain at which any one of the poles go throw the positive side of the real axis. The critical gain is marked on the root locus plot, K_{cr}=15.

We plotted the system response for a step input with $K_1 = \frac{K_{cr}}{2} = 7.5,\ K_2 = K_{cr} + 1 = 16$:

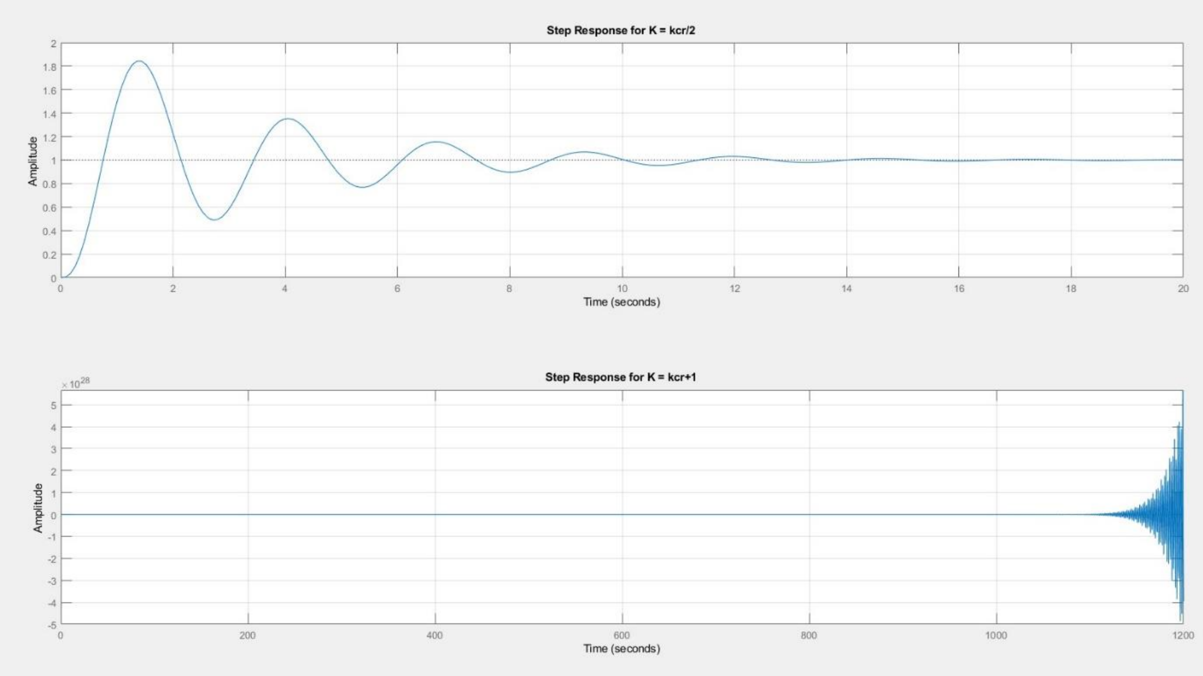


Figure 8 - step response of K_1 and K_2

When $K > K_{cr}$, the system is unstable, so the step response is not bounded.

When $K < K_{cr}$ we get a stable response as expected.

Nyquist plot of the system $C \cdot P$:

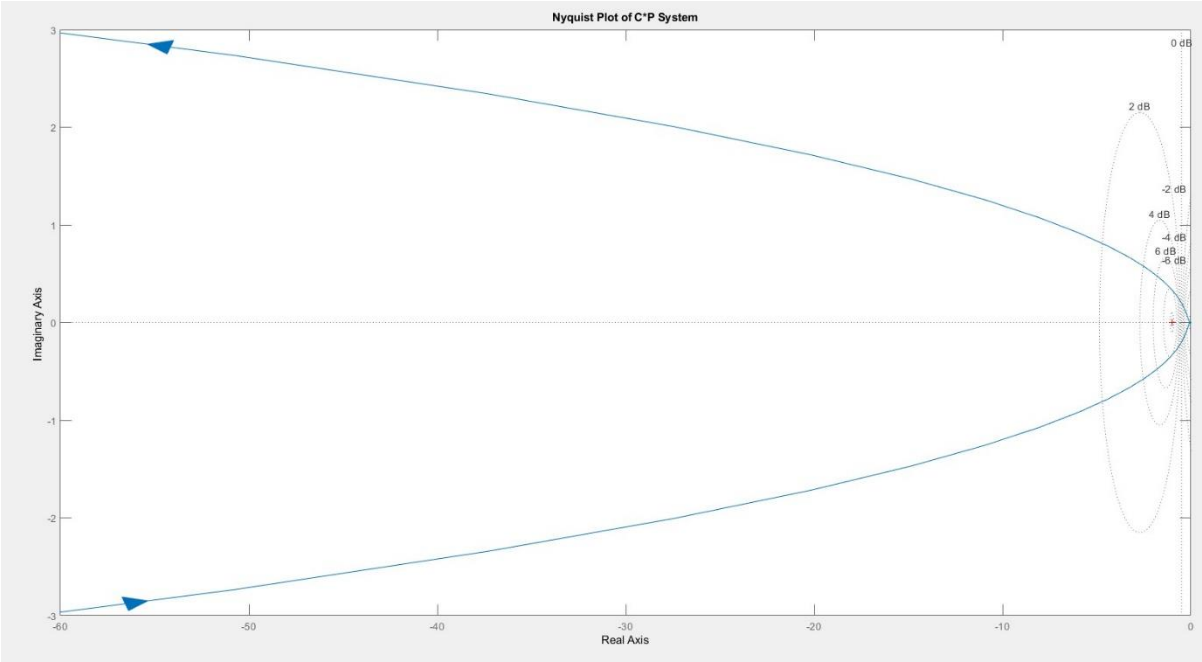


Figure 9 -Nyquist plot of the CP

The difficulty in assessing stability directly from the diagram is since the diagram is not in logarithmic form and therefore, we cannot tell if there is a closed loop around -1. Hence, because of the above difficulty, we use the nyqlog() function:

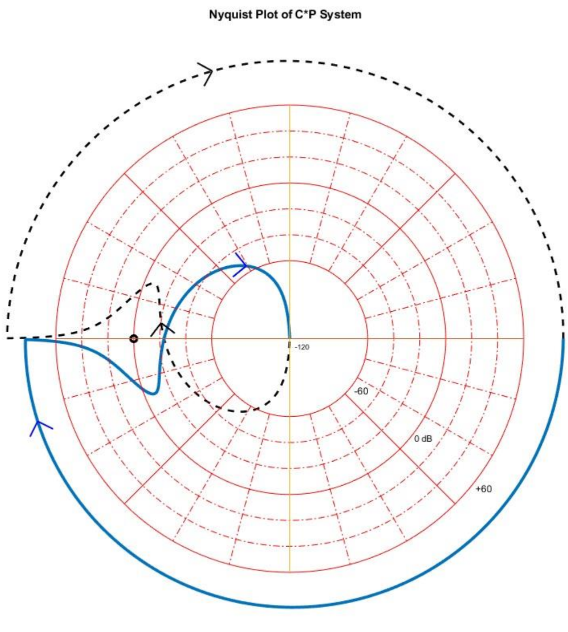


Figure 10 - nyquist using the nyqlog function for CP

There are 2 closed loops around -1 and in C*P there are 2 unstable poles, therefore it does meet nyquist condition for stability.

Nyqlog() function for $K = K_{cr} + 1 = 16$:

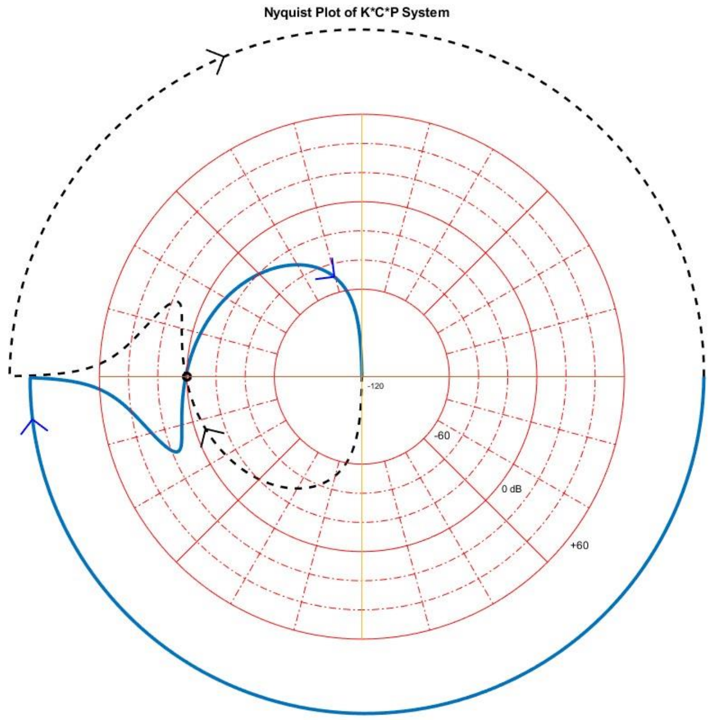
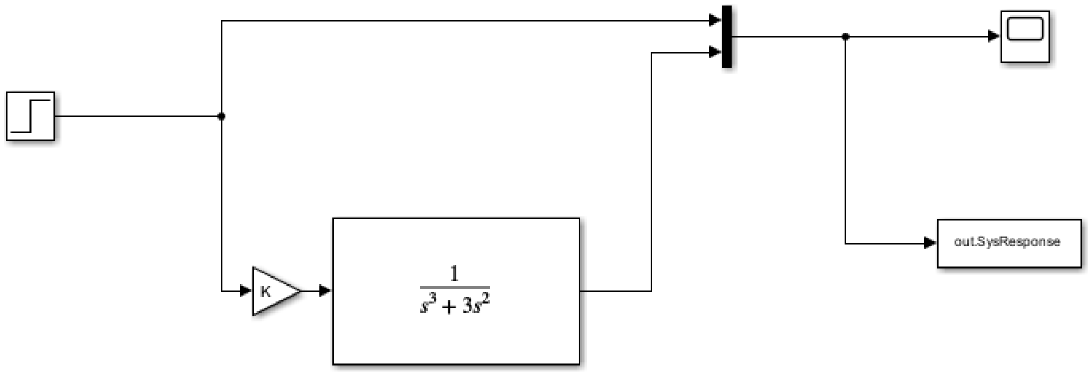


Figure 11 - nyquist using the nyqlog function for CPK

We see that the plot passes (or at least passes very close) through -1. This indicates that the system is unstable. This is true since $K > K_{cr}$, which implies that the system is unstable. Also, we can see that due to the fact we add K the loop became bigger (the gain is bigger).

3.4. Simulink

The open loop scheme:



For $K = 1$ and for a time of 50s, we get the following plot:

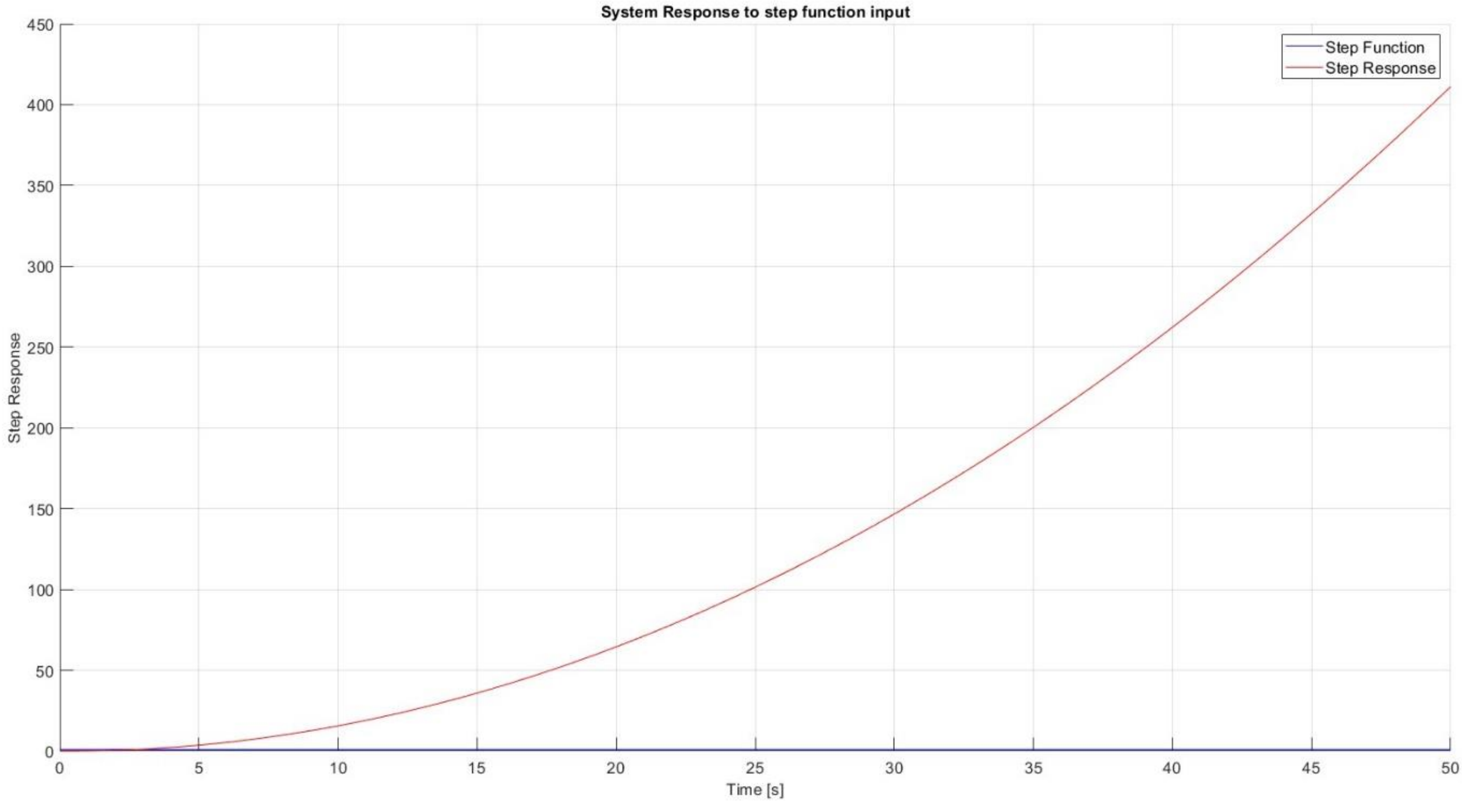
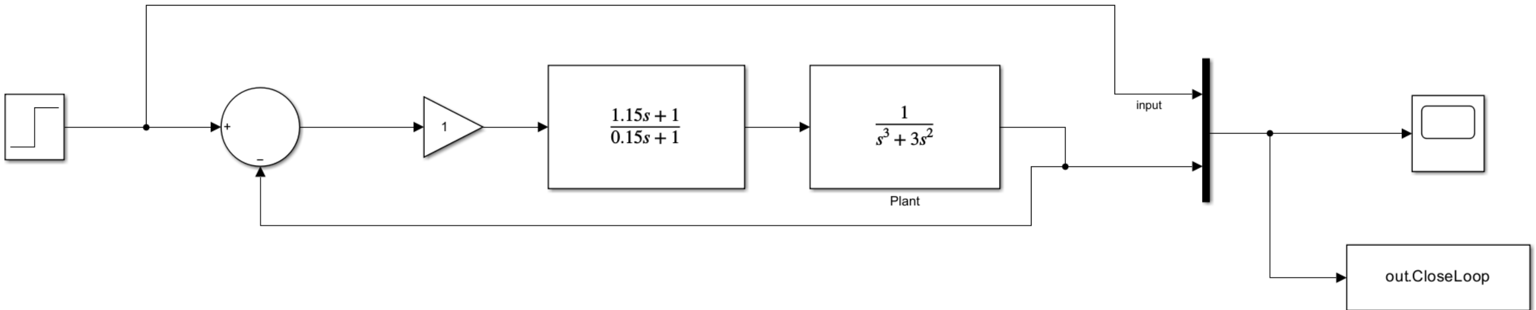


Figure 12 - system response for K=1

The closed loop diagram:



For $K = 1$ and for a time of 50s, we get the following plot:

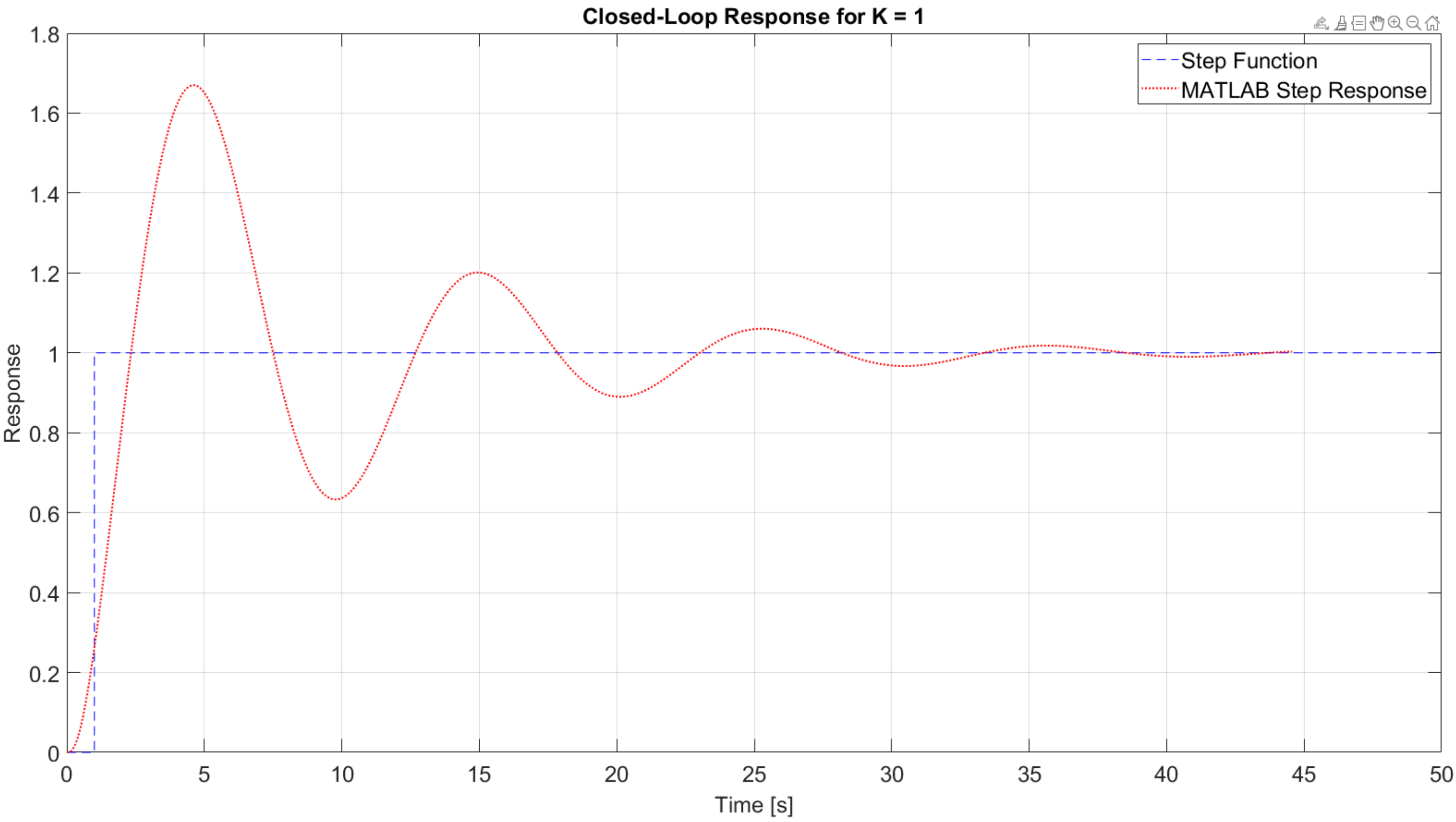


Figure 14 - closed loop response for K=1

Now we got a stable system response to a step function input.

For $K = \frac{K_{cr}}{2} = 7.5$ and for a time of 50s, we get the following plot:

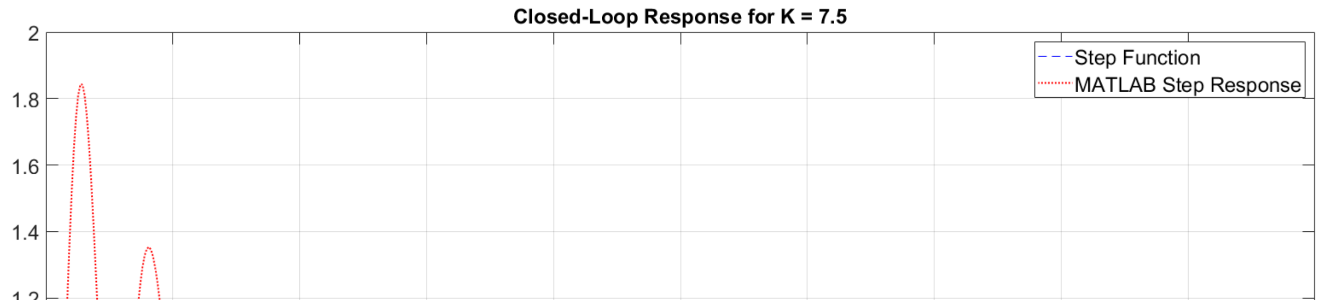


Figure 15 - closed loo[for K=7.5

For $K = K_{cr} + 1 = 16$ and for a time of 50s, we get the following plot:

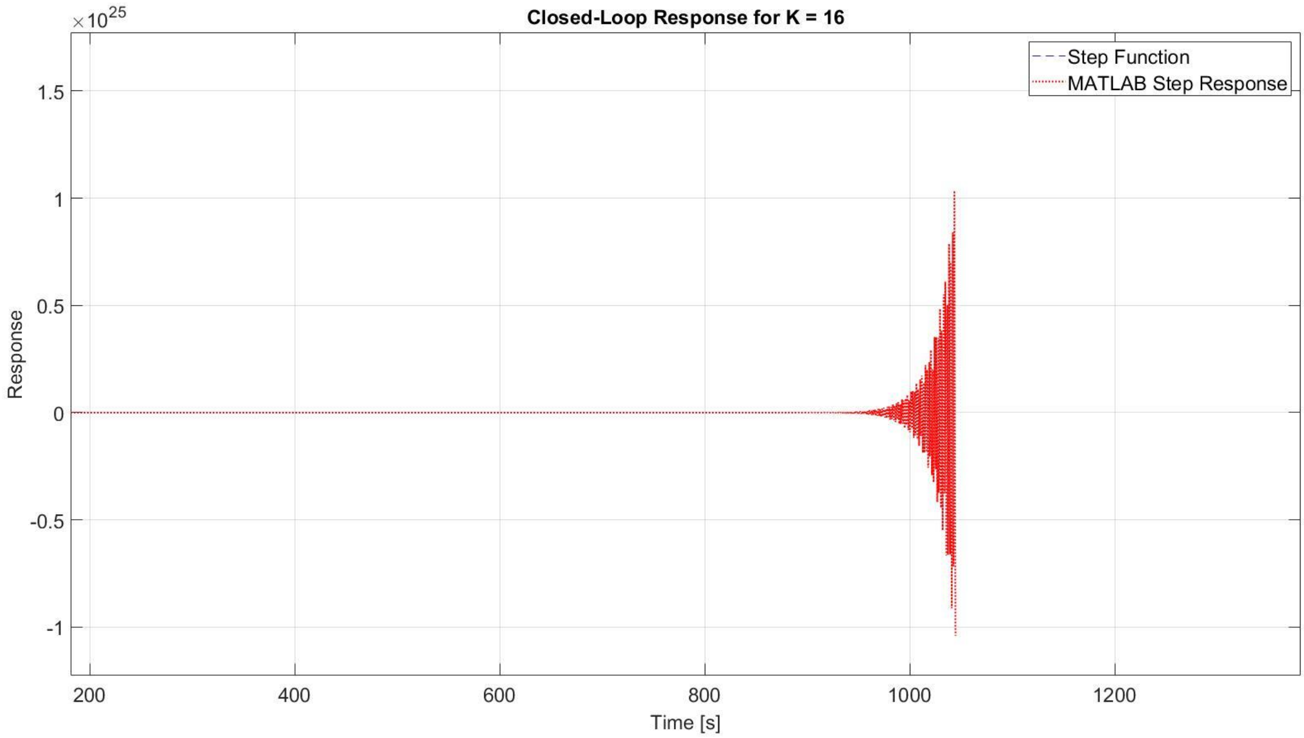


Figure 16 - closed lop for K=16

3.5. Code following

We added comments to each row with an explanations:

```
x = 0:1:4; % Create a vector 'x' ranging from 0 to 4 with a step size of 1
```

```
num = x(2); % Extract the second element of 'x' and assign it to 'num'
den = [x(3:5), x(1)]; % Create a denominator vector by concatenating elements 3 to 5 of 'x' with the
first element of 'x'

sys = tf(num, den); % Create a transfer function 'sys' using 'num' as the numerator and 'den' as the
denominator

K = 1; % Set the variable 'K' to 1
mdl = "close_loop"; % Define the Simulink model name as a string
open_system(mdl); % Open the Simulink model specified by 'mdl'

for i = 1:3 % Start a loop that iterates three times
   figure(i); % Create a new figure window with the figure number specified by the loop index 'i'
   K = i * 1.5; % Update the value of 'K' within the loop
   [y, t] = step(feedback(K * sys, 1)); % Simulate the step response of the closed-loop system with the
updated value of 'K'
   set_param(mdl, "SimulationCommand", "start"); % Start the simulation of the Simulink model
specified by 'mdl'
   pause(1); % Pause the execution for 1 second to wait for the simulation to finish
   set_param(mdl, "SimulationCommand", "writedatalogs"); % Write logs from the simulation to the
desktop
   plot(out.CloseLoop.time, out.CloseLoop.signals.values(:, 2), '--r'); % Plot the time versus the
second column of the signals data from the simulation output 'out.CloseLoop' with a dashed red line
   hold on; % Keep the current plot active for adding additional elements
   plot(t, y, ':b', 'LineWidth', 1.5); % Plot the time response 't' versus the output response 'y' with a blue
dotted line and set the line width to 1.5
   grid on; % Add a grid to the current plot
   hold off; % Release the current plot for the next iteration of the loop
   pause(1); % Pause the execution for 1 second to wait for the plot to update

   if i==1

    legend('simulation step response', 'matlab step response')
   end
end
```

Legend was added to figure 1 by adding the if statement into the MATLAB code:

```
if i==1

    legend('simulation step response', 'matlab step response')
   end
```

Figure 1:

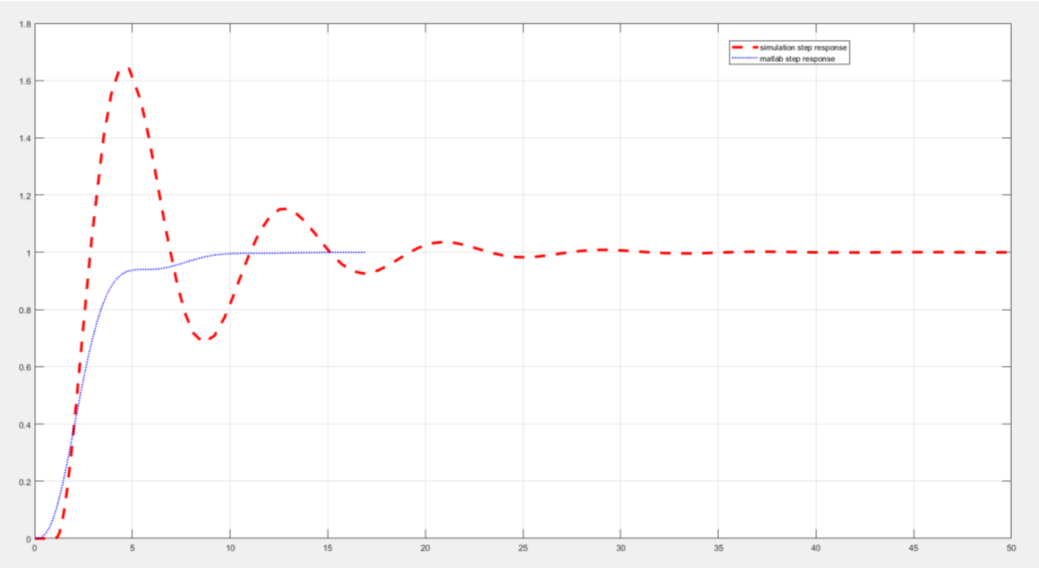


Figure 17 - Figure 1

Figure 2:

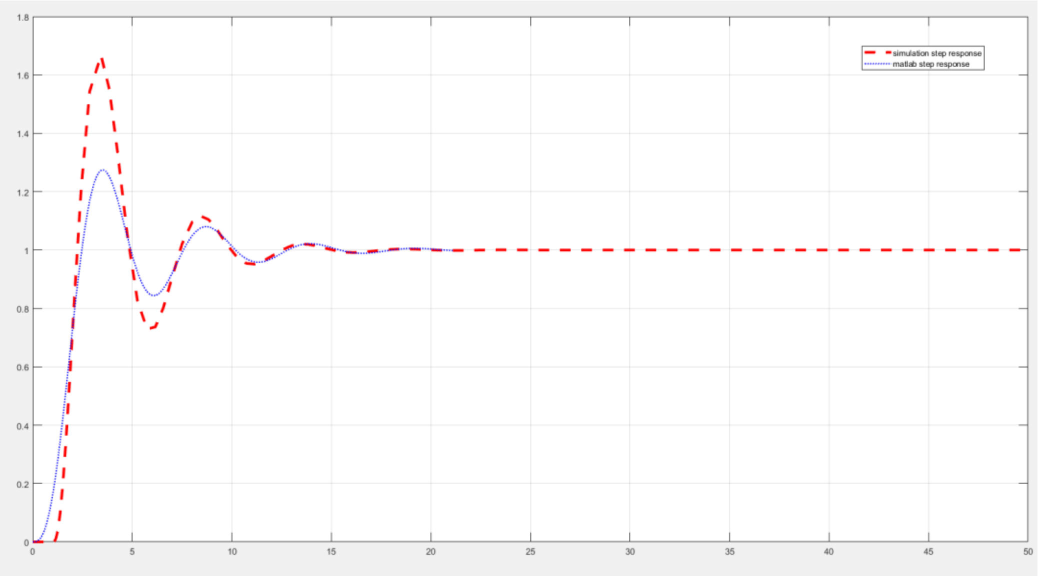


Figure 18 - Figure 2

The main reason to the difference between figure 1 and 2 is the gain K.

- In the first picture, for k=1.5, we can tell that the blue line response follows the step input without oscillations.
- In the second picture, for k=3, we see oscillations. Moreover, it takes more time to the outputs to stabilize around the step input.

We except the Simulink simulation and the MATLAB's transfer function to create the same graphs but we obtain a different result. A reason for the difference can be that when we use Simulink, we also use a scope that can maybe affect the measurements. (it has its own capacitance and resistance).

Difficulties that raised during the experiment and suggestions for solutions

No difficulties in this experiment

Conclusions

In lab, in this experiment we verified the results we got in the prelab report. we practiced MATLAB and Simulink and learnt new techniques and functions. we verified the phase margin and gain margin of the system using the Nyquist plot and. Moreover, we implemented the given systems, simulated their step responses and got the graphs we expected.