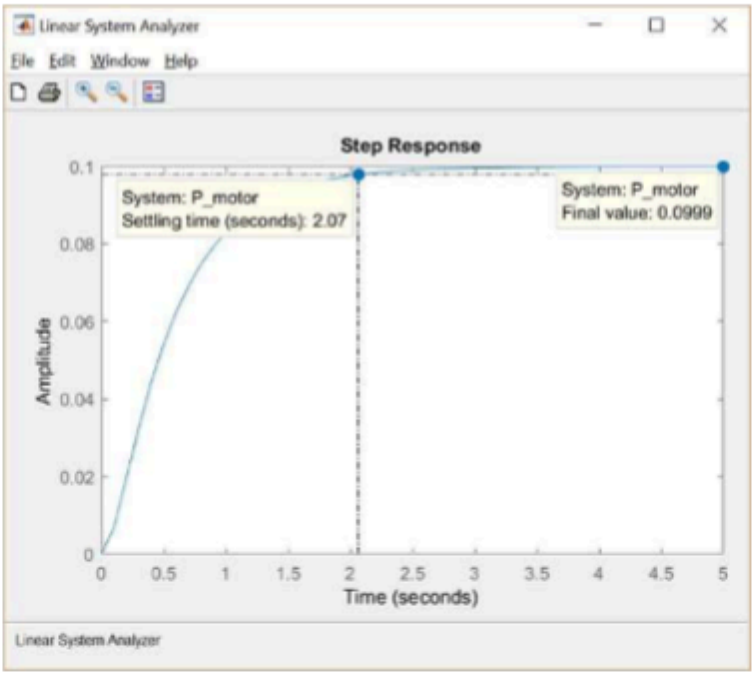
Lab Task

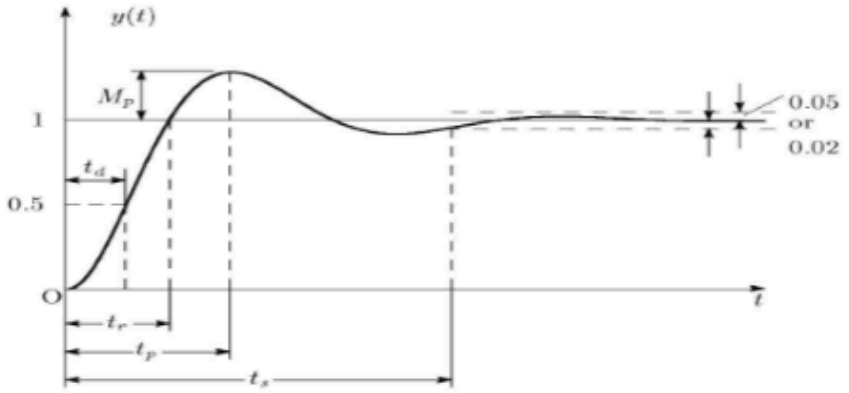
Controller Design Requirements

First consider that our uncompensated motor rotates at 0.1 rad/sec in steady state for an input voltage of 1 Volt. The figure below was simulated using the command:

```
J = 0.01;

b = 0.1;

K = 0.01;

R = 1;

L = 0.5;

s = tf('s');
P_motor = K/((J*s+b)*(L*s+R)+K^2);
linearSystemAnalyzer('step', P_motor, 0:0.1:5);
```



While we are considering the design requirements it is important to revisit the idea of what these system properties mean. The figure below is an example of what rise time, settling time, peak overshoot etc. represent, when we look at the transient response of a dynamic system.

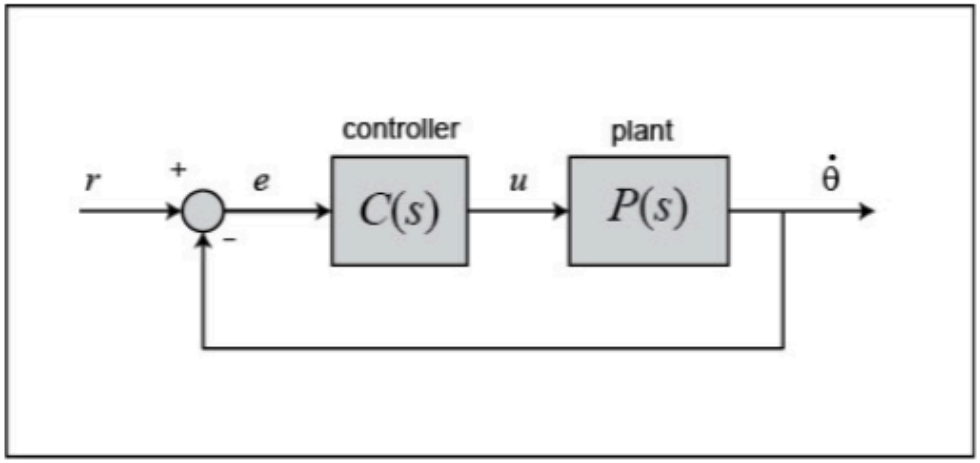


Since the most basic requirement of a motor is that it should rotate at the desired speed, we will require that the steady-state error of the motor speed be less than 1%. Another performance requirement for our motor is that it must accelerate to its steady-state speed as soon as it turns on. In this case, we want it to have a settling time less than 2 seconds. Also, since a speed faster than the reference may damage the equipment, we want to have a step response with overshoot of less than 5%. In summary, for a unit step command in motor speed, the control system's output should meet the following requirements.

- Settling time less than 2 seconds
- Overshoot less than 5%
- Steady-state error less than 1%

Task 1

Adhering to the above system design requirements, design a controller with a lag compensator for the servo motor velocity control using root locus controller design theory. The figure below is an overview of what the closed-loop system would look like.



You may use the Control System Designer in MATLAB to aid with the design process. Once the controller design is completed fill the table below with the closed-loop system properties.

Poles	Damping	Frequency (rad/s)	Time Constant (s)

Controller Design Steps:

- Use the Control System Design Toolbox in MATLAB to first ascertain the Gain (K_c) that help you obtain the desired settling time and overshoot criterion. Is there steady state error after the gain was introduced?
- If there is steady state error, then you'll have to design a lag controller to reduce the steady state error. A lag compensator is one type of controller known to be able to reduce steady-state error. However, we must be careful in our design to not increase the settling time too much. The transfer function for the lag compensator is written as:

$$C(s) = K_c \frac{s - z}{s - p}$$

For our simulation we can add a lag compensator with a zero at -1 and a pole at -0.01. The controller gain K_c can then be adjusted, using the Control System Design Toolbox, to complement the compensator transfer function to achieve the design stipulations.

Task 2

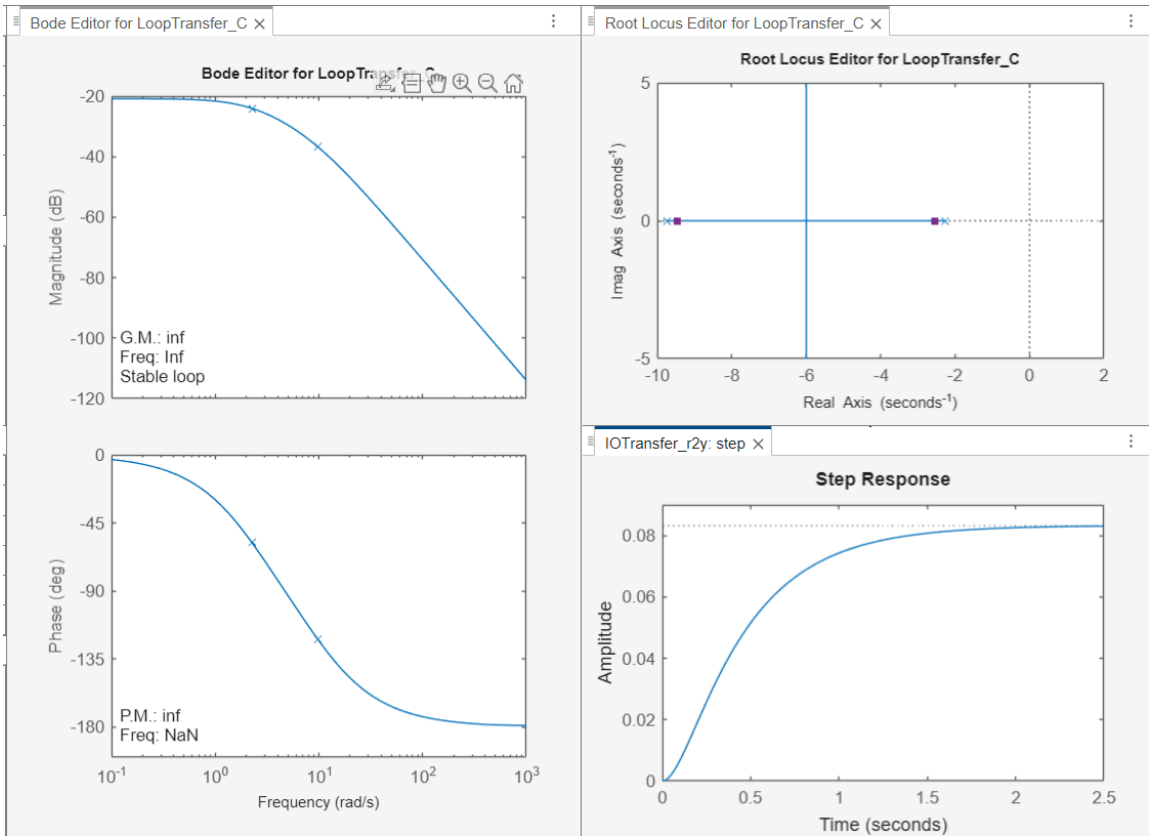
Once the controller has been successfully designed, simulate the system with the controller and the plant by writing a .m file in MATLAB with r(t) =1 rad/s and unity feedback for t=0:0.1:5.

Task 3

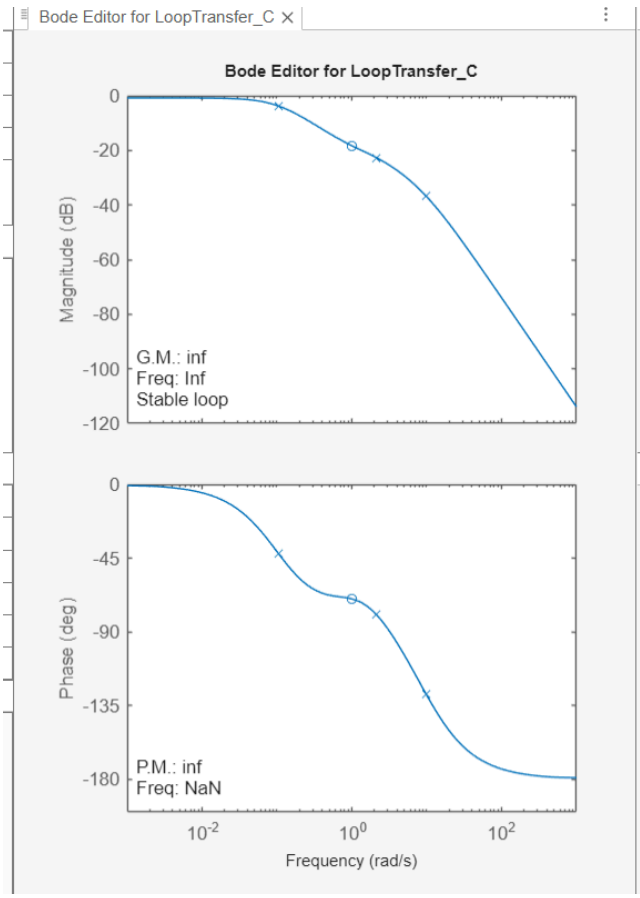
Repeat Task 2 in Simulink.

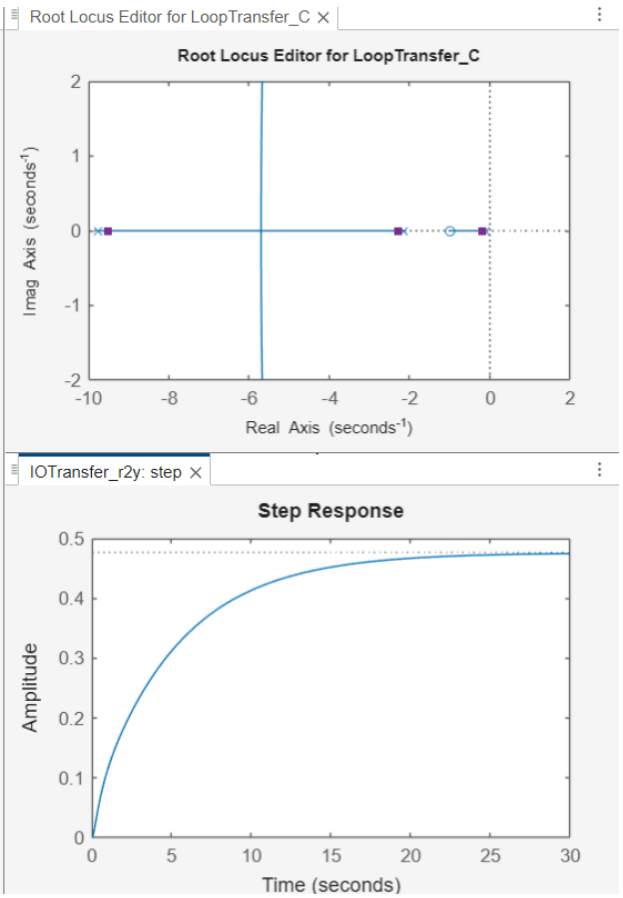
Task 1

```
clear;
clc;
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;

s = tf('s');
P_motor = K/((J*s+b)*(L*s+R)+K^2);

system1 = P_motor
system2 = 1;

sys = feedback(system1, system2);

controlSystemDesigner(sys)
```



```
clear;
clc;
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;
C= 1;

s = tf('s');
P_motor = K/((J*s+b)*(L*s+R)+K^2);
systemC = C*(s+1)/(s+0.01);

system1 = P_motor*systemC
system2 = 1;

sys = feedback(system1, system2);

controlSystemDesigner(sys)

```





Task 2

```
clear;
clc;
J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;
C= 75;

s = tf('s');
P_motor = K/((J*s+b)*(L*s+R)+K^2);
systemC = C*(s+1)/(s+0.01);

system1 = P_motor*systemC
system2 = 1;

sys = feedback(system1, system2);

a = 0:0.1:5;
x = ones(size(t));

[y,a] = lsim(sys, x, a);

figure;
plot(t,y);
xlabel('Time (s)');
ylabel('Output (rad/s)');
title('Control System');
grid on;
```

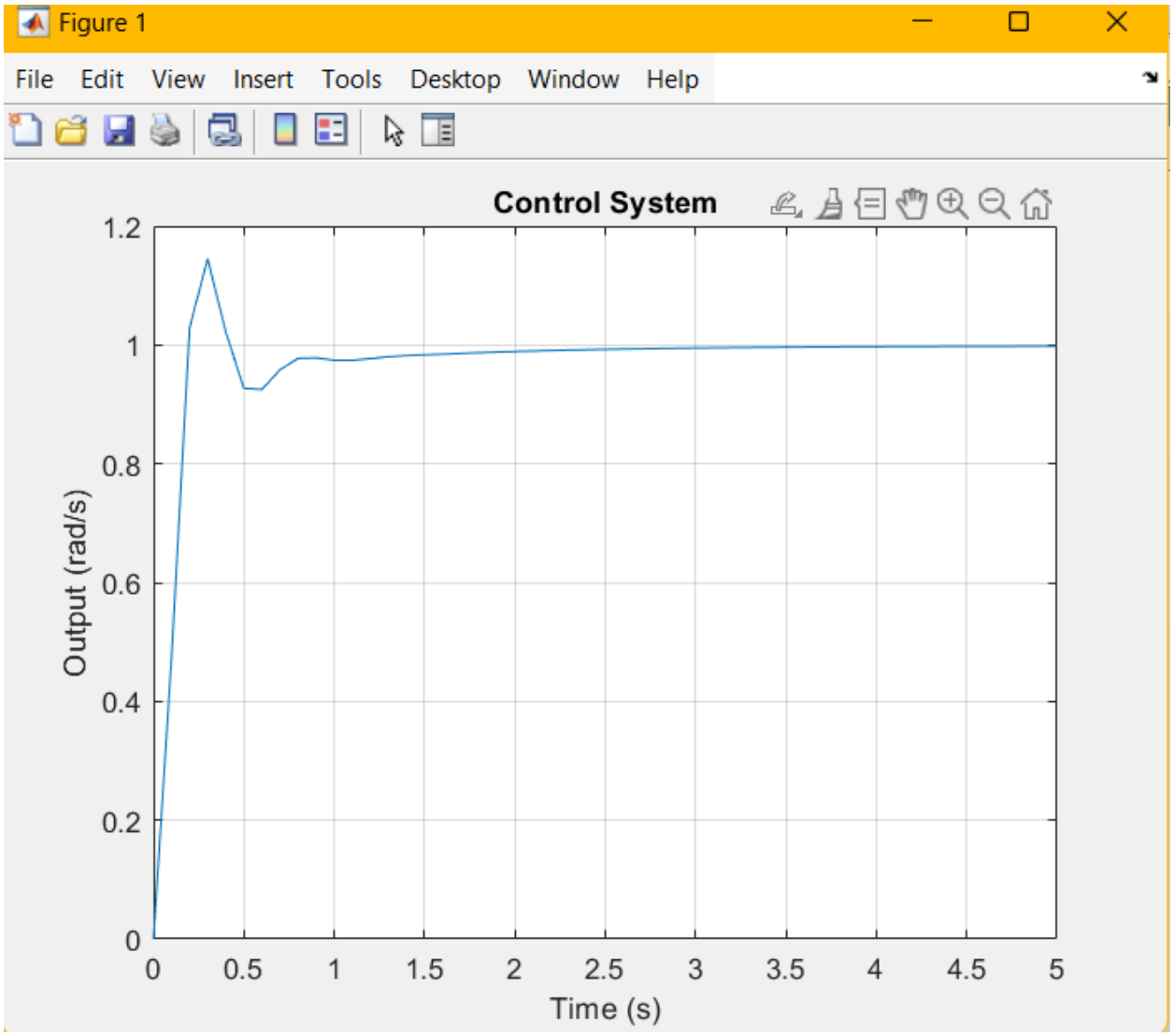
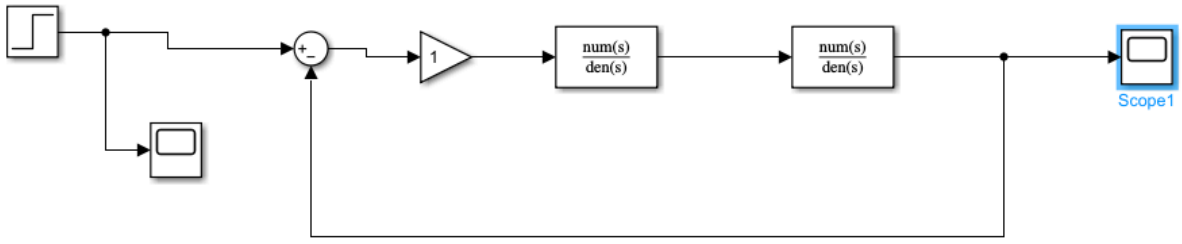
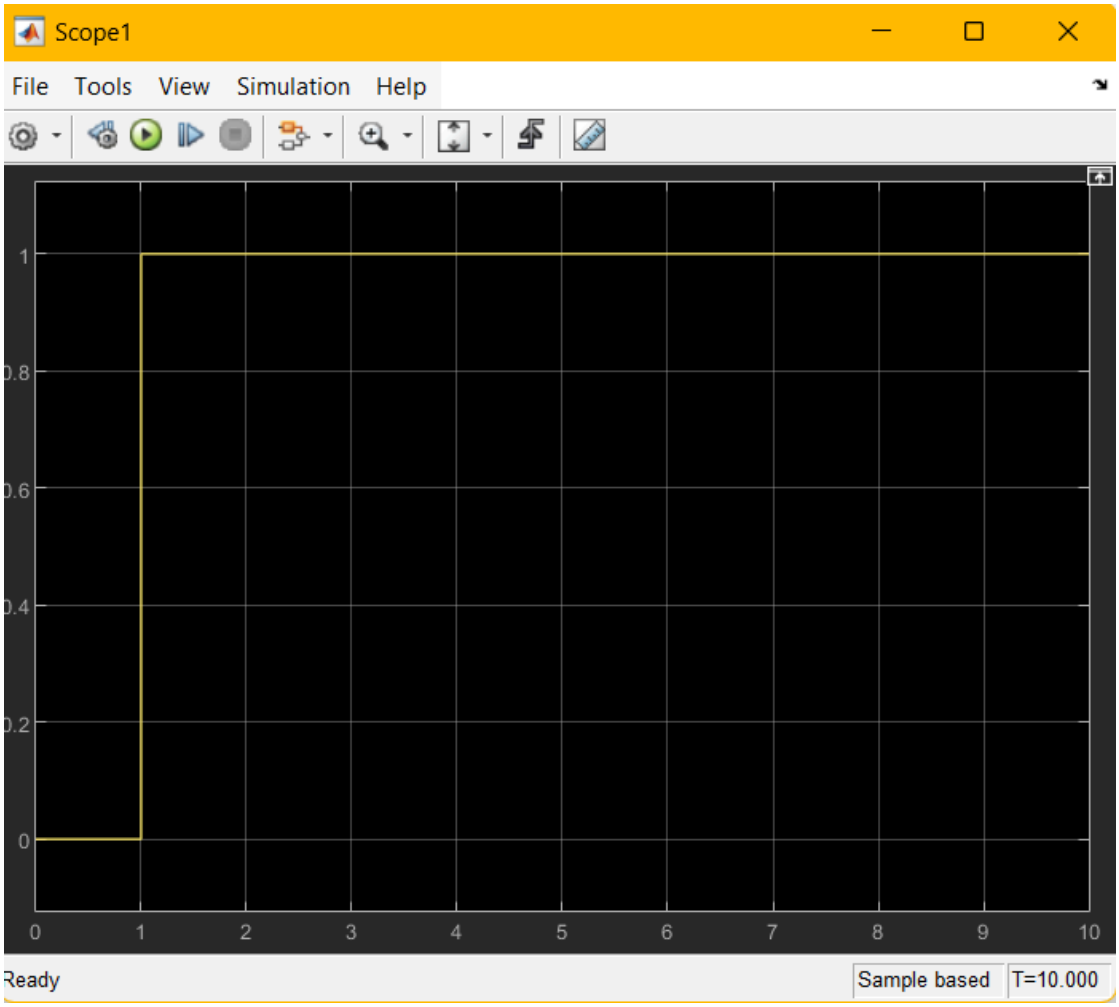


Figure 1
File Edit View Insert Tools Desktop Window Help
Control System
Output (rad/s)
Time (s)
0
0.2
0.4
0.6
0.8
1
1.2
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

Task 3


num(s)
den(s)
num(s)
den(s)
Scope1


Scope1
File Tools View Simulation Help
1
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000