CO326: Industrial Networks
Lab 04 - PID Control

Introduction

The goal of this lab exercise is to learn about the PID controller and a few basic tuning rules of it. After completing this lab, you will be able to

1) explain what PID control is
2) tune a PID controller for a desired system output and
3) relate PID controller parameters to step response characteristics of the controlled system.

For this lab exercise, consider the application of a PID controller in speed control of a DC motor.

The equivalent circuit diagram of the DC motor is shown in Fig.1.

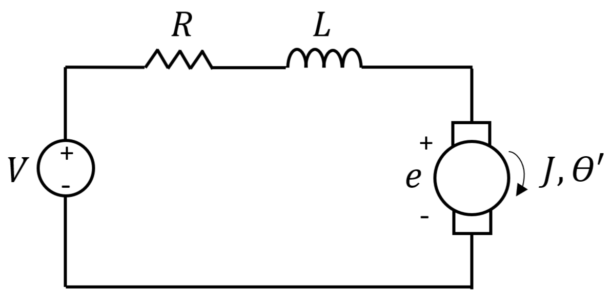

Fig.1 Circuit diagram of the DC motor

The input of the system is the voltage applied to the armature (V), while the output is the rotational speed of the shaft (θ').

From Fig.1, the governing equations of the DC motor can be derived as follows.

$$J\theta'' + b\theta' = Ki$$

$$L\frac{di}{dt} + Ri = V - K\theta'$$

Where,
J- Moment of inertia of the rotor
b- Motor viscous friction coefficient
K- Motor torque constant (K_t)/ Electromotive force constant (K_e)
R- Resistance
L- Inductance
Applying the Laplace transform, the above equations can be expressed in terms of the Laplace variable s.

$$s(Js + b)\theta(s) = KI(s)$$

$$(Ls + R)I(s) = V(s) - Ks\theta(s)$$

By eliminating $I(s)$, the transfer function of the system can be obtained as,

$$p(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{K}{(Js+b)(Ls+R)+K^2}$$

PID Controller:

The transfer function of the most basic form of PID controller, as we use in this lab exercise, is

$$C(s) = K_p + \frac{K_i}{s} + K_d s$$

where K_p = Proportional gain, K_i = Integral gain and K_d = Derivative gain.

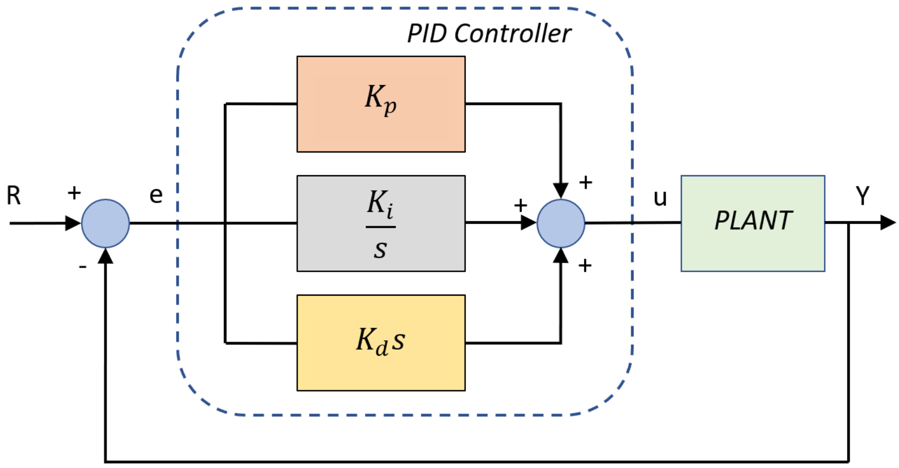

Fig.2 Block diagram of the system with PID controller

We assume that the controller is used in a closed loop unity feedback system. The input **R** is the reference signal and **Y** is the system output. The variable **e** denotes the tracking error, which is sent to the PID controller. The control signal **u** from the controller to the plant is equal to the proportional gain (K_p) times the magnitude of the error plus the integral gain (K_i) times the integral of the error plus the derivative gain (K_d) times the derivative of the error.

In this lab exercise, we are most interested in the following characteristics of the closed-loop step response.

1. Rise Time: the time it takes for the plant output **y** to rise beyond 90% of the desired level for the first time.
2. Overshoot: how much the peak level is higher than the steady state, normalized against the steady state.
3. Settling Time: the time it takes for the system to converge to its steady state.

4. Steady-state Error: the difference between the steady-state output and the desired output.

Lab Exercises

The parameters of the DC motor that we are going to use in this lab exercise are as follows.

Table 1: Parameters of the DC motor

Parameter	Value
J- Moment of inertia of the rotor	0.01 $kg.m^2$
b- Motor viscous friction coefficient	0.1 $N.m.s$
K_t- Motor torque constant	0.01 $V/rad/s$
K_e- Electromotive force constant	0.01 $N.m/A$
R- Resistance	1 Ω
L- Inductance	0.5 H

You will need MATLAB for these lab exercises. If you don't have MATLAB installed on your computer, please download the trial version from [Get MATLAB - MATLAB & Simulink] and install it on your computer.

PART I: Open-Loop Step Response

Open MATLAB. Create a new *m-file* and run the following code.

```
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);

t = 0:0.01:5;
step(P_motor,t)
grid
title('Step Response')
```

The graph shows the system response when a step input is applied. Record the value of input, output, rise time, overshoot, settling time and steady-state error.

Table 2: Open-loop step response

Input	Output	Rise Time	Overshoot	Settling	Steady-state

(Amplitude)	(Amplitude)	(s)	(Amplitude)	Time (s)	Error (Amplitude)

State your observations comparing the output to the input (reference).

PART II: Closed-Loop Step Response

Now, consider the closed-loop system with a PID controller as shown in Fig.2.

In order to see how each of the terms, K_p, K_i and K_d contributes to the changes of the closed-loop step response, consider the scenarios; Proportional Control, Proportional-Integral Control, Proportional-Derivative Control and Proportional-Integral-Derivative Control.

Proportional Control

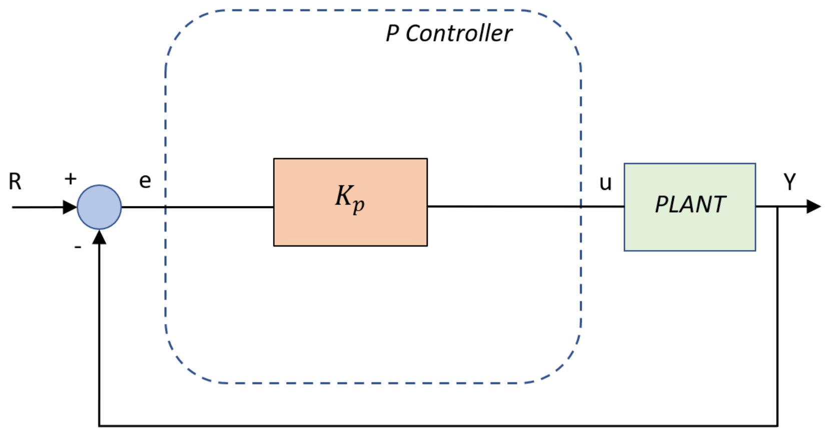

Fig.3 Block diagram of the system with P controller

Create a new *m-file* and type in the following commands.

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);*

Kp = 200;
C = pid(Kp);
*sys_cl = feedback(C*P_motor,1);*

t = 0:0.01:5;
step(sys_cl,t)

grid
title('Step Response with Proportional Control')

Run the *m-file*. Change the value of K_p as given in Table 3 and observe how the system response changes. Record the value of rise time, overshoot, settling time and steady-state error for each K_p in the following table.

Table 3: Step response with Proportional control

Gain	Rise Time	Overshoot	Settling Time	Steady-state Error
$K_p = 50$				
$K_p = 100$				
$K_p = 200$				

State your observations and conclusions.

Proportional-Integral Control

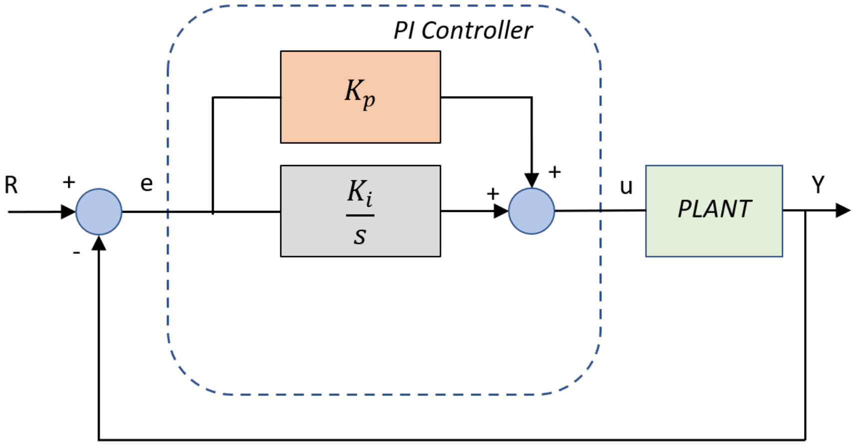

Fig.4 Block diagram of the system with PI controller

Enter the following commands into a new *m-file* and run it.

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);*

Kp = 50;
Ki= 50;

```
C = pid(Kp,Ki);
sys_cl = feedback(C*P_motor,1);

t = 0:0.01:5;
step(sys_cl,t)
grid
title('Step Response with Proportional Integral Control')
```

Keep the value of $K_p = 50$ and change the value of K_i as given in Table 2. Observe how the system response changes.
Record the value of rise time, overshoot, settling time and steady-state error for each K_i in the following table.

Table 4: Step response with Proportional-Integral control

Gain	Rise Time	Overshoot	Settling Time	Steady-state Error
$K_p = 50$ $K_i = 50$				
$K_p = 50$ $K_i = 100$				
$K_p = 50$ $K_i = 200$				

State your observations and conclusions.

Proportional-Derivative Control

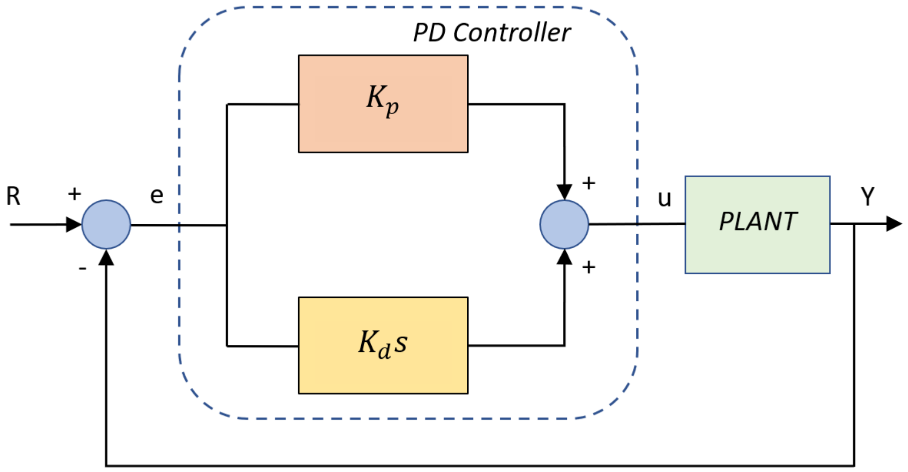

Fig.5 Block diagram of the system with PD controller

Create a new *m-file* and type in the following commands.

```
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);

Kp = 50;
Kd=10;
C = pid(Kp,0,Kd);
sys_cl = feedback(C*P_motor,1);

t = 0:0.01:5;
step(sys_cl,t)
grid
title('Step Response with Proportional Derivative Control')
```

Keep the value of $K_p = 50$ and change the value of K_d as given in Table 3. Observe how the system response changes.
Record the value of rise time, overshoot, settling time and steady-state error for each K_d in the following table.

Table 5: Step response with Proportional-Derivative control

Gain	Rise Time	Overshoot	Settling Time	Steady-state Error
$K_p = 50$ $K_d = 0.1$				
$K_p = 50$ $K_d = 1$				
$K_p = 50$ $K_d = 10$				

Proportional-Integral-Derivative Control

Create a new *m-file* and run the following commands. Give values of K_p,K_i and K_d as preferred and observe how the system response changes.

```
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);

Kp = <value>
Ki = <value>
Kd = <value>
C = pid(Kp,Ki,Kd);
sys_cl = feedback(C*P_motor,1);

t = 0:0.01:5;
step(sys_cl,t)
grid
title('Step Response with Proportional Integral Derivative Control')
```

Summarize the effects of increasing each of the controller parameters K_p,K_i and K_d in Table 4 (*increase/ decrease/ no change*).

Table 6: Step response with Proportional-Integral-Derivative control

Gain	Rise Time	Overshoot	Settling Time	Steady-state Error
K_p				
K_i				
K_d				

PID Controller Design

Design a PID controller to satisfy the below criteria.

For a 1-rad/s step reference,

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

Record the results in the table below.

Table 7: Proportional-Integral-Derivative control

Input	
Output	
K_p	

K_i	
K_d	
Rise Time	
Overshoot	
Settling Time	
Steady-state Error	

Submission

Create a report named **GrouoXX_Lab04Report.pdf** where XX is your Grouup number including the necessary explanations, screenshots and answers for the Lab Exercises.

Submit a folder named **Lab04_GroupXX** including the report and the matlab project file.