MECHATRONICS CONTROL AND AUTOMATION LAB (MCTA 3104)

EXPERIMENT 2:

PID CONTROLLER DESIGN FOR DC MOTORS SPEED CONTROL

INDIVIDUAL REPORT

PROGRAMME: MECHATRONICS ENGINEERING

NUR AYU NAZIERA BINTI ROSLI	2119202
DATE OF EXPERIMENT	21 OCTOBER 2024
DATE OF SUBMISSION	28 OCTOBER 2024

Contents

Introduction

This experiment focuses on designing and tuning a PID controller to achieve optimal speed control for a DC motor. DC motor control is essential in mechatronic systems, where precise speed regulation is often required. By tuning the proportional (P), integral (I), and derivative (D) gains, the controller's performance can be modified to meet desired specifications such as a quick response time, minimal overshoot, and low steady-state error. Using MATLAB and Simulink, this experiment simulates different controller configurations and analyzes the motor's step response, which helps in identifying the best controller settings for improved performance.

Objectives

The objectives of this experiment are:

- To tune a PID controller for a system/plant using MATLAB.
- To investigate the features of each proportional (P), integral (I) and derivatives (D) controller to the plant.

In this experiment, DC motor speed control is chosen as the system/plant.

Background

The plant of DC motor is a good example of a dynamic system that includes both mechanical dynamic due to inertia and viscous damping and the electrical dynamic due conductance and resistance of the motor. The resulting model is a second order. The control objectives are to have a fast response to a step input (i.e. speed change or position change) with small overshoot and small steady-state error.

Physical Model of Differential Equations

A common actuator in control system is DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables can provide transitional motion. A sketch and a schematic diagram of an armature-controlled DC servo motor are shown in Figures 1 and 2 respectively.

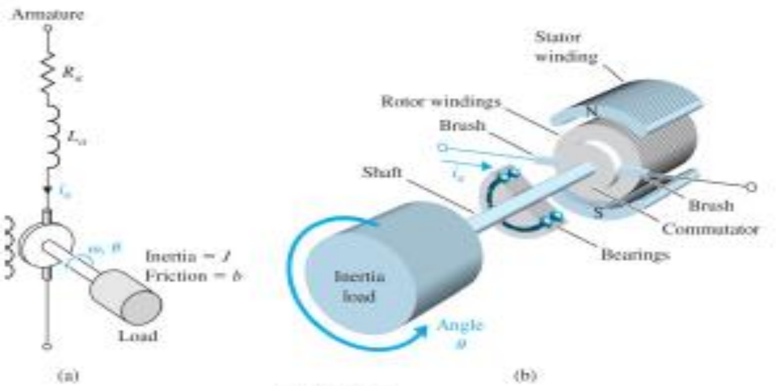


Figure: 02-16ab

Figure 1: (a) Schematic diagram and (b) sketch of a DC motor [1]

The system variables of a DC motor include:

• e_a: armature drive potential (V)

•e_b: back EMF or counter-electromotive-force (V)

• i_a: armature current (A)

• *T*: torque produced by servomotor (Nm)

• θ: angular position of rotor (rad)

• θ·: angular speed of rotor (rad/s)

The physical parameters of the system include:

• *Ra*: armature electric resistance (W) = 1.0 W

• *La* : armature electric inductance (H) = 0.5 H

• *J*: rotor’s moment of inertia (Nms2/rad) = 0.01 Nms2/ rad

• *b*: damping ratio (Nms) = 0.1 Nms

• *KT*: torque constant (Nm/A) = 0.01 Nm/A

• *Ke*: back EMF constant (Nm/A) = 0.01 Nm/A

In this experiment, these given physical parameter values will be used in for the MATLAB simulation studies.

The motor torque, T, is related to the armature current, i_a, by a constant K_t, and the back EMF, e_b, is related to the rotational velocity by a constant K_e as shown in the following equations:

$$T(s) = K_T i_a(s) \tag{1}$$

$$e_b(t) = K_e\dot{\theta}(t) \tag{2}$$

In SI units, K_t, is equal to K_e, ($K_T = K_e = K$). From the above figures, the following equations based on Newton's law combined with Kirchhoff's law can be derived:

$$J\ddot{\theta} + b\dot{\theta} = Ki_a \tag{3}$$

$$L\frac{di}{dt} + Ri_a = e_a - K\dot{\theta} \tag{4}$$

2.2. System's Transfer Function

Using Laplace transforms, the above modeling equations, can be expressed in term of s as:

$$s(Js + b)\theta(s) = KI_a(s) \tag{5}$$

$$(Ls + R)I_a(s) = E_a(s) - Ks\theta(s) \tag{6}$$

By eliminating $I(s)$, the open-loop transfer function where the rotational speed ($\dot{\theta}$) is the output and the voltage (E_a) is the input can be obtained as:

$$G_v(s) = \frac{s\theta(s)}{E_a(s)} = \frac{K}{(Js + b)(L_a s + R_a) + K^2} \tag{7}$$

$$= \frac{0.01}{0.005s^2 + 0.06s + 0.1001} \tag{8}$$

Note that, we are going to design PID controller for this transfer function in this lab.

3. PID Controller

PID controller is the most popular controller in many applications. The block diagram of a closed-loop control system is illustrated as:

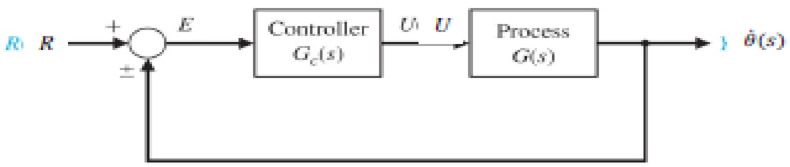


Figure 2: Block diagram of closed-loop control system

Recall that the transfer function for a PID controller is:

$$PID = K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_D s + K_I}{s} \tag{9}$$

In some cases only combination of PI or PD controller is sufficient to achieve the required specification.

$$PI = K_p + \frac{K_I}{s} = \frac{K_I\left(s + \frac{K_P}{K_I}\right)}{s} \tag{10}$$

$$PD = K_P + K_D s = K_D\left(s + \frac{K_P}{K_D}\right) \tag{11}$$

The value of the K_P, K_I and K_D can be theoretically calculated or tuned to compensate the plant in order to achieve the desired performance of the system, in terms of setting time, rise time, steady state error and percentage of overshoot.

Simulation Procedure

Basic requirement of a motor is that it should rotate at the desired speed and the steady-state error of the motor speed should be small. Other performance requirement is that the motor must accelerate to its steady-state speed as soon as it turned on. Since any speed faster than the reference speed may damage the motor, the percentage overshoot is also need to be limited.

For this lab, the motor speed output should have the following desired specifications:

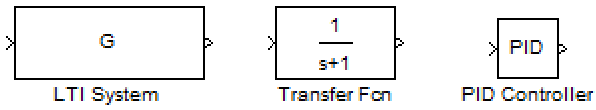
- Settling time (Ts) ≤ 2 seconds
- Percentage overshoot (PO) ≤ 5 %
- Steady-state error (Ess) ≤ 1 %

Note:
- For step no. 1 to 6, please include in your answers the relevant MATLAB script, Simulink block diagrams and plot of the step responses.

- An example of MATLAB script may be used to define a transfer function:

```
>> G = tf([a1],[b1 b2])
```

- In Simulink, the following block diagrams may be used to represent transfer function:



PID Controller Tuning

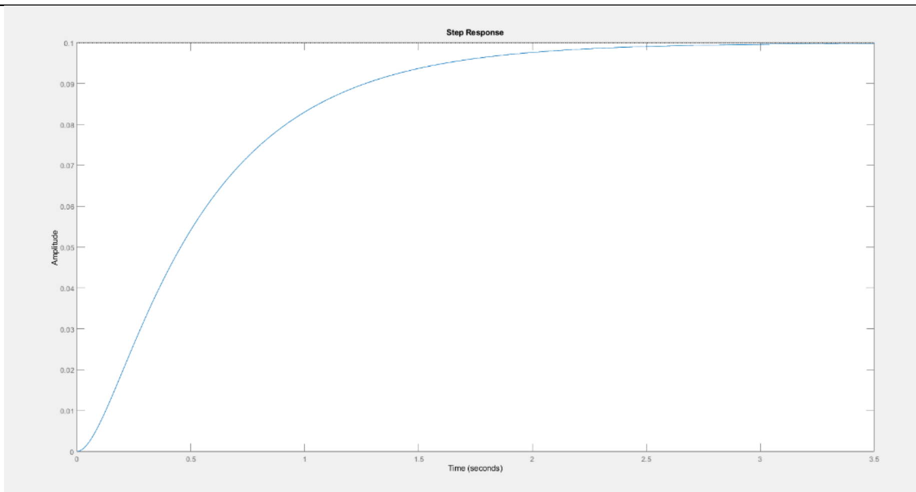
1) By using MATLAB/Simulink, obtain the open-loop unit step response of Gv(s). Calculate the:
   - Settling time (Ts)
   - Rise time (Tr)
   - Percentage overshoot (PO)
   - Steady-state error (Ess)

   and give comments.

<table>
<tr><td>MATLAB script / Simulink block diagrams</td><td colspan="2">t=0:3:100;
dt= 0.1;

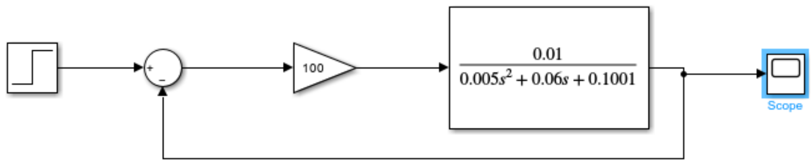
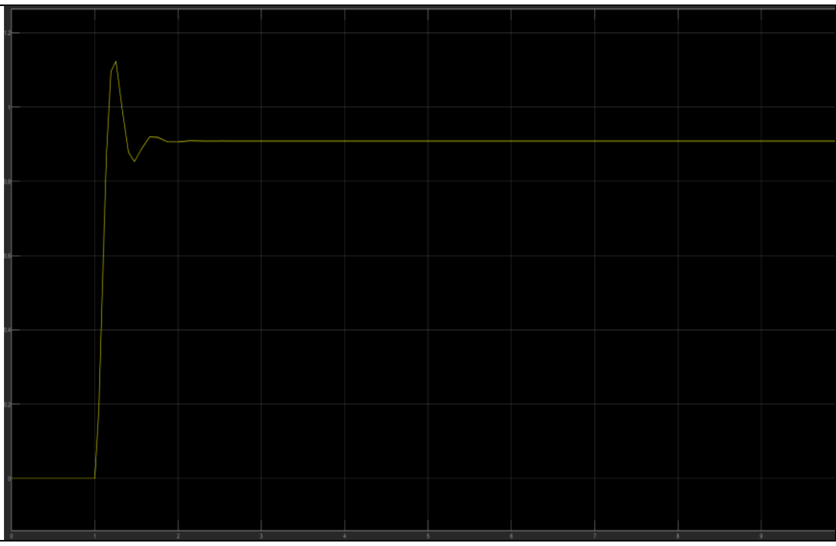
num = [0.01];
den = [0.005 0.06 0.1001];

G=tf(num,den);
step(G);</td></tr>
<tr><td rowspan="2">Open-loop unit response</td><td rowspan="2"></td><td>TS= 2.07S</td></tr>
<tr><td>Tr= 1.14S</td></tr>
</table>

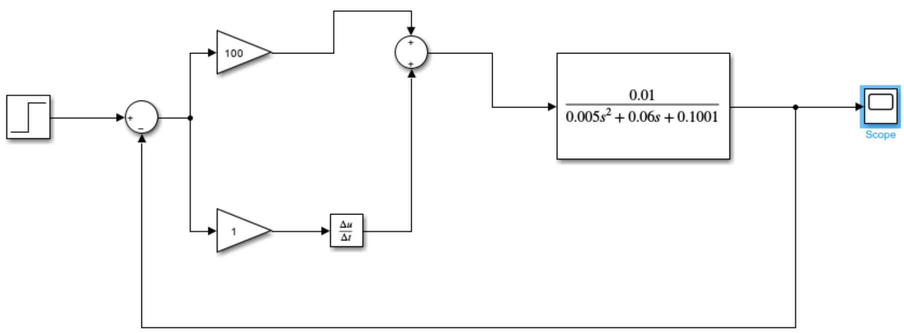
		PO= na Ess= 0.0999
Comments	The system has a relatively quick response time with a low steady-state error, but the percent overshoot is not provided, making it difficult to assess potential transient overshoot behavior.	

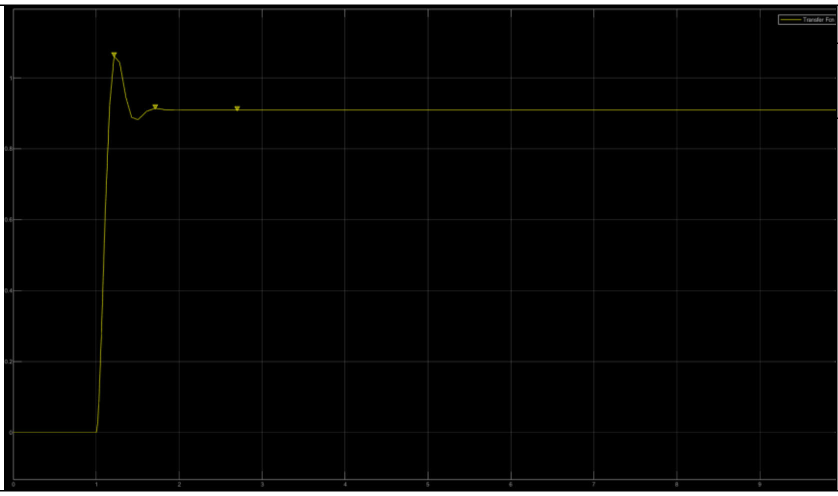
2) Modify the open-loop block diagram to include feedback system with a proportional controller P with Kp = 100. Obtain the closed-loop step response and give comments on the performance (Ts, PO and Ess).

MATLAB script / Simulink block diagrams	t=0:3:100;
	dt= 0.1;
	Kp = 100;
	Ki = 0;
	Kd= 0;
	h = [1];
	
	num = [0.01];
	den = [0.005 0.06 0.1001];
	G=tf(num,den);
	C= pid(Kp,Ki,Kd);
	
	Mc = feedback(G * C,h);
	step(Mc);

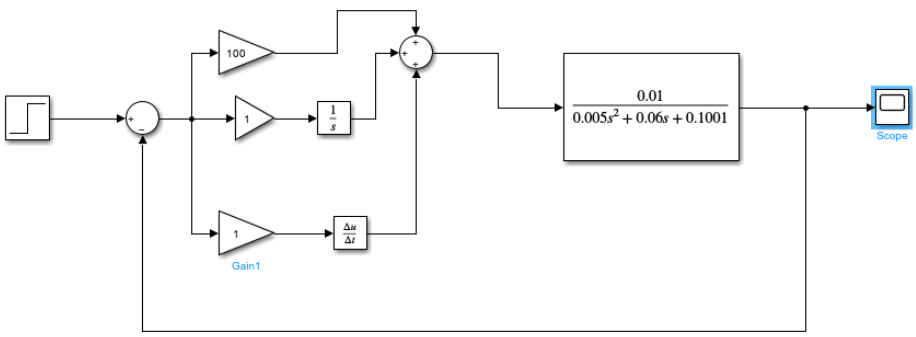
		
Open-loop unit response		TS= na Tr= 107.75s PO=24.375 % Ess= 0.909
comments	The system has a slow response with significant overshoot, and the steady-state error of 0.909 indicates a substantial deviation from the desired final value, pointing to possible instability or inaccuracy in steady-state performance.	

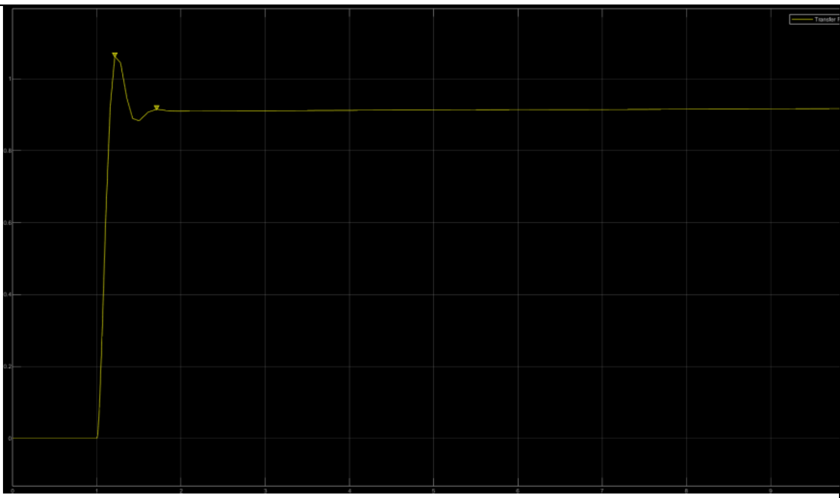
3) Replace the proportional P controller with a proportional-derivative PD controller with KP = 100 and KD= 1 (small KD). Obtain the closed-loop step response and give comments on the PO and Ess as compared to the P controlled in step no. 2.

MATLAB script / Simulink block diagrams		
		TS= na

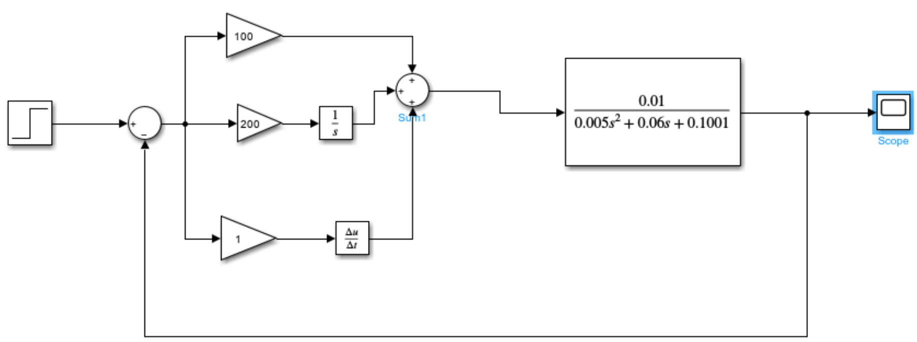
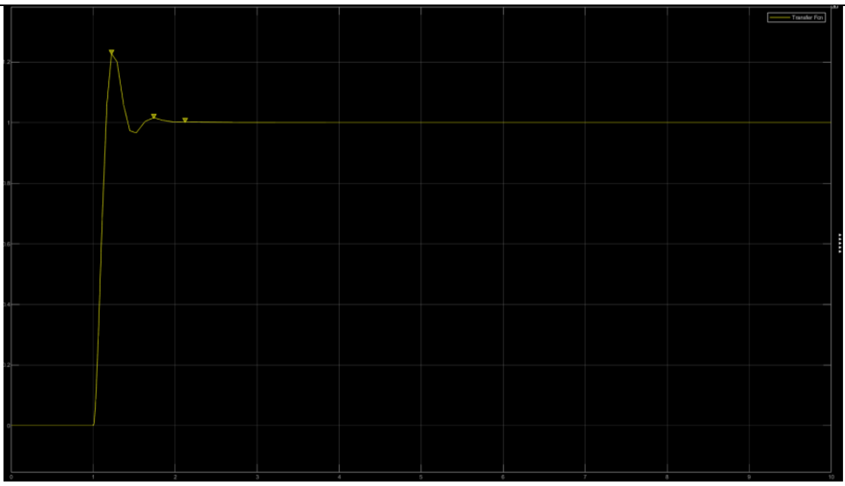
Open-loop unit response		Tr= 109.00s PO=17.059 % Ess= 0.9966
comments	The system response is slow with moderate overshoot, and a steady-state error of 0.9966 shows a high deviation from the desired final value. This suggests that the system may not be achieving adequate accuracy in its steady-state performance.	

4) Replace the proportional-derivative controller PD with a proportional-integral-derivative PID controller with KP =100, KI = 1 and KD = 1 (small KI and KD). Obtain the closed-loop step response. Determine whether all the desired specifications have been satisfied and give comments.

MATLAB script / Simulink block diagrams		
Open-loop unit response		TS= na Tr= 110.511s PO=15.698 %

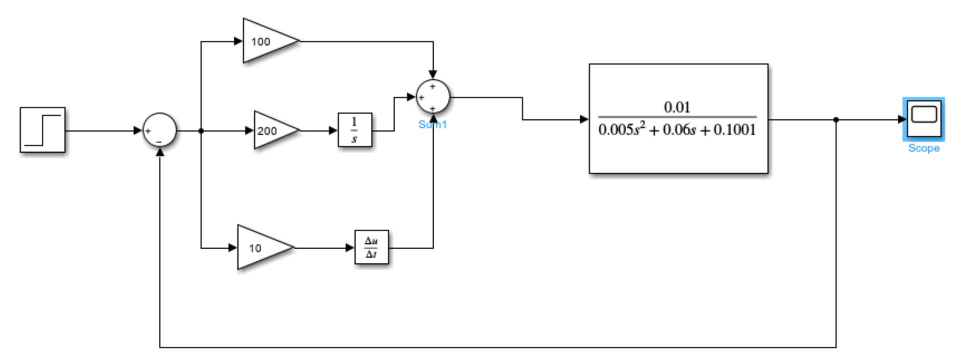
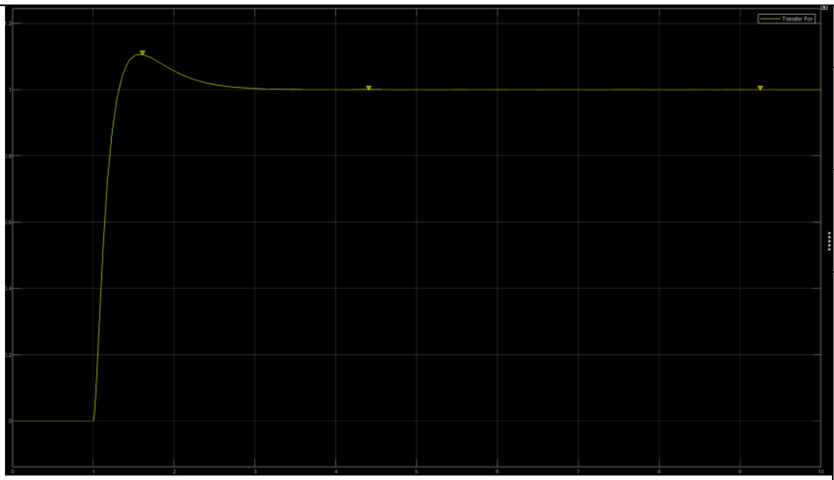
		Ess= 0.9151
comments	With a high-rise time and a lower percent overshoot than previous results, this configuration suggests some improvement in stability. However, the steady-state error of 0.9151 indicates significant deviation from the desired final value, reflecting limited accuracy in steady-state performance.	

5) Tune the PID controller by increasing KI = 200 (KP =100, KD = 1). Obtain the closed-loop step response. Determine whether all the desired specifications have been satisfied and give comments.

MATLAB script / Simulink block diagrams		
Open-loop unit response		TS= na Tr= 108.570s PO=22.84 % Ess= 0.9987

comments	This system setup has a moderate rise time with a high percent overshoot, indicating some level of instability. The steady-state error of 0.9987 reflects a very high deviation from the target value, suggesting poor accuracy in achieving the desired steady-state.

6) Retune the PID controller by increasing KD = 10 (KP =100, KI = 200). Obtain the closed-loop step response. Determine whether all the desired specifications have been satisfied and give comments.

<table>
<tr><td>MATLAB script / Simulink block diagrams</td><td colspan="2">
</td></tr>
<tr><td rowspan="4">Open-loop unit response</td><td rowspan="4"></td><td>TS= na</td></tr>
<tr><td>Tr= 214.255s</td></tr>
<tr><td>PO=10.556 %</td></tr>
<tr><td>Ess= 1</td></tr>
<tr><td>comments</td><td colspan="2">This configuration shows the longest rise time and the lowest percent overshoot, which indicates increased stability. However, the steady-state error of 1 reveal that the system does not reach the desired final value, suggesting that it may be unable to achieve accurate steady-state performance.</td></tr>
</table>

Conclusion

The experiment successfully demonstrates the impact of tuning PID parameters on the performance of a DC motor's speed control system. Through incremental adjustments in the proportional, integral, and derivative gains, it was observed that each parameter has a unique effect on settling time, rise time, overshoot, and steady-state error. Although stability and speed were achievable, challenges in balancing these with accuracy emerged, particularly at high values of integral or derivative gains. The insights gained from the analysis reinforce the importance of careful PID tuning for achieving a desired response in control systems.