Experiment No. 5

Experiment Name: Root Locus and Time-Response Analysis of a DC Motor with PID, PI, and PD Controllers

Objectives:

The objective of this experiment is to generate and analyze the root locus of an armature-controlled DC motor, design suitable **PI**, **PD**, and **PID controllers** to meet desired transient-response specifications and compare their **closed-loop step responses**—including rise time, settling time, overshoot, and steady-state error—using MATLAB/Simulink simulations.

Theory:

A **standard armature-controlled DC motor** exhibits both electrical and mechanical dynamics that can be described by coupled differential equations. The **electrical subsystem** governs the armature circuit, while the **mechanical subsystem** represents the rotor's motion.

- **Electrical equation:**

$$L\frac{di_a(t)}{dt} + Ri_a(t) + K_e\omega(t) = v_a(t)$$

where $L$is the armature inductance, $R$is the armature resistance, $K_e$is the back EMF constant, $\omega(t)$is the angular velocity, and $v_a(t)$is the armature voltage input.

- **Mechanical equation:**

$$J\frac{d\omega(t)}{dt} + B\omega(t) = K_t i_a(t)$$

where $J$is the rotor inertia, $B$is the viscous friction coefficient, and $K_t$is the torque constant.

By taking the **Laplace transform** and eliminating the armature current $i_a(t)$, the **open-loop transfer function** (speed plant) of the DC motor can be derived as:

$$G(s) = \frac{\Omega(s)}{V_a(s)} = \frac{K_t}{(Js + B)(Ls + R) + K_e K_t}$$

This transfer function describes how the motor's angular speed $\Omega(s)$responds to an input voltage $V_a(s)$.

When a **unity feedback configuration** is applied with a controller $C(s)$, the **closed-loop transfer function** of the system becomes:

$$T(s) = \frac{C(s)G(s)}{1 + C(s)G(s)}$$

and the **characteristic equation** is given by:

$$1 + C(s)G(s) = 0$$

This equation determines the system's stability and transient behavior, which can be analyzed using the **root locus** method.

Depending on the control objective, different controller structures can be implemented:

- **PI Controller:**

$$C_{PI}(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s}$$

Provides zero steady-state error for step inputs and moderate transient response.

- **PD Controller:**

$$C_{PD}(s) = K_p + K_d s$$

Improves damping and transient response but does not eliminate steady-state error.

- **PID Controller:**

$$C_{PID}(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

Combines the benefits of P, I, and D control—fast response, zero steady-state error, and improved stability.

The **root locus method** is employed to study how the closed-loop poles of the system move as the loop gain varies. By introducing zeros through PI, PD, or PID controllers, the shape of the root locus is modified, allowing improved **damping ratio (ζ)** and **natural frequency (ω_n)** to meet desired transient specifications.

For a **dominant underdamped system**, transient response characteristics are estimated using the following standard relationships:

$$M_p \approx 100e^{-\pi\zeta/\sqrt{1-\zeta^2}} \text{and} T_s(2\%) \approx \frac{4}{\zeta\omega_n}$$

where M_p is the **percent overshoot** and T_s is the **settling time**.

By adjusting the controller parameters K_p, K_i, and K_d, the root locus can be shaped to achieve the desired damping ratio and natural frequency, thus optimizing system performance in terms of **speed, stability, and accuracy**.

Required Software:

- MATLAB
- Simulink

Plant Model and Controller Design

The simulation of the DC motor speed control system was carried out using the following motor parameters:

$$J = 0.01,\ B = 0.1,\ K_e = K_t = 0.01,\ R = 1,\ L = 0.5.$$

The controller gains were tuned as follows:

- **PI Controller:** $K_p = 23.665,\ K_i = 43.66192,\ K_d = 0$
- **PD Controller:** $K_p = 530.43375,\ K_i = 0,\ K_d = 91.85$
- **PID Controller:** $K_p = 594.048,\ K_i = 1971.592,\ K_d = 47.6$

Design Overview:
The **PI controller** introduces a pole at the origin, making the system type-1 and thus eliminating steady-state error for step inputs. Its zero placement enhances transient performance. The **PD controller** introduces a lead zero that increases the phase margin, thereby reducing overshoot and improving response speed, though some steady-state error persists due to the absence of integral action. The **PID controller** combines the advantages of both—its integral term removes steady-state error, the derivative term adds damping for improved stability, and the proportional term determines overall responsiveness.

Code:

```
clc;
clear all;
close all;

% --- DC Motor Parameters ---
J = 0.01;
B = 0.1;
Ke = 0.01;
Kt = 0.01;
```

```
R = 1;
L = 0.5;

% --- Transfer Function of the DC Motor ---
s = tf('s');
TF = Kt / ((J*s + B)*(L*s + R) + Ke*Kt);

%% =============================
%        PID Controller
% =============================
Kp = 346.4;
Ki = 1627;
Kd = 25.572;

C1 = pid(Kp, Ki, Kd);
CL1 = feedback(C1*TF, 1);
S1 = stepinfo(CL1);
disp('PID Parameters:');
disp(S1);

figure(1);
rlocus(C1*TF);
grid on;
title('Root Locus with PID');

figure(2);
step(CL1, 5);
grid on;
title('Step Response with Designed PID');


%% =============================
%         PI Controller
% =============================
Kp = 21.84;
Ki = 31.4277;
Kd = 0;

C2 = pid(Kp, Ki, Kd);
CL2 = feedback(C2*TF, 1);
S2 = stepinfo(CL2);
disp('PI Parameters:');
disp(S2);

figure(3);
```

```matlab
rlocus(C2*TF);
grid on;
title('Root Locus with PI');

figure(4);
step(CL2, 5);
grid on;
title('Step Response with Designed PI');


%% ============================
%         PD Controller
% ============================
Kp = 513.83;
Ki = 0;
Kd = 86.67;

C3 = pid(Kp, Ki, Kd);
CL3 = feedback(C3*TF, 1);
S3 = stepinfo(CL3);
disp('PD Parameters:');
disp(S3);

figure(5);
rlocus(C3*TF);
grid on;
title('Root Locus with PD');

figure(6);
step(CL3, 5);
grid on;
title('Step Response with Designed PD');
```

Output:

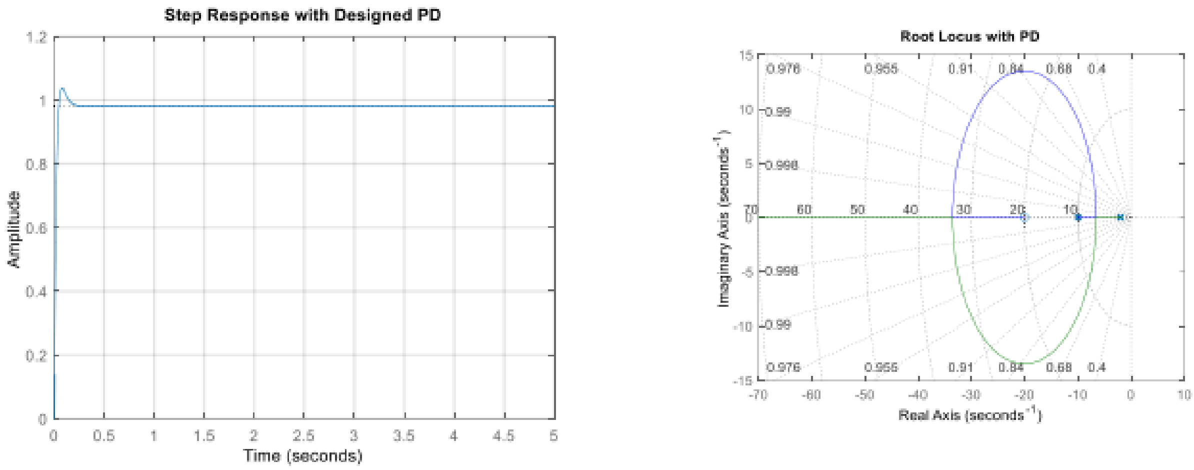

Figure 1: PD Controller: Step Response and Root Locus

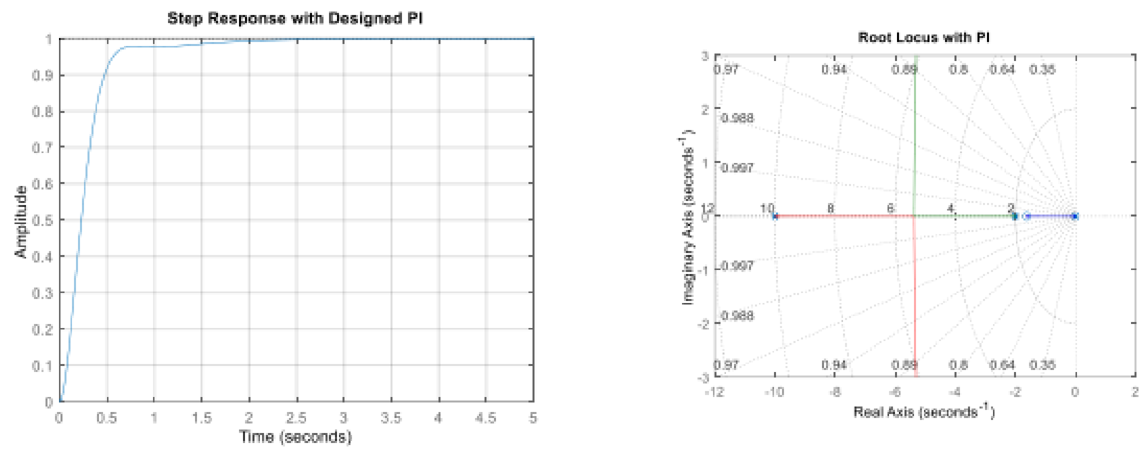

Figure 2: PI Controller: Step Response and Root Locus

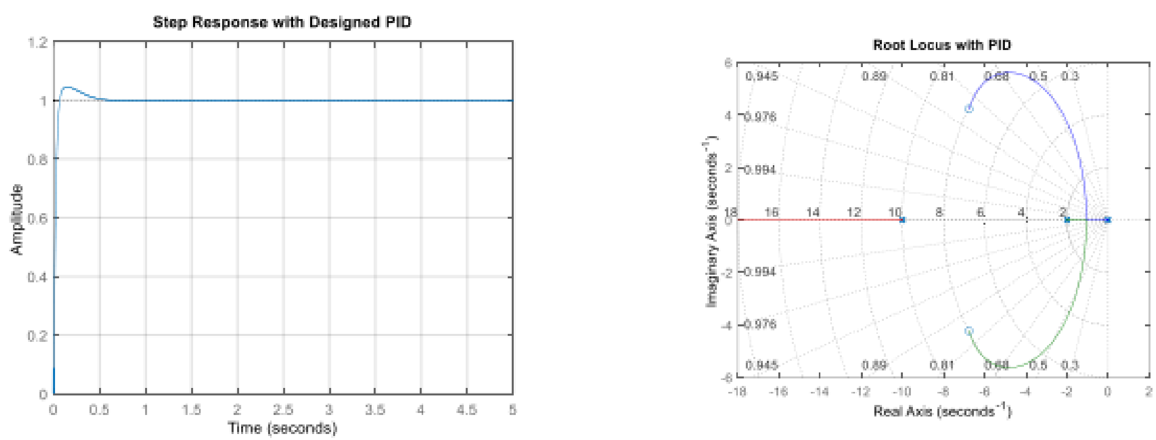

Figure 3: PID Controller: Step Response and Root Locus

Result:

The simulation results demonstrate that all three controllers effectively stabilize the DC motor speed but exhibit distinct performance characteristics. The **PI controller** provides **type-1 system behavior**, eliminating steady-state error for step inputs while maintaining moderate overshoot and settling time. The **PD controller** achieves the **fastest transient response** with minimal overshoot, though it retains a small steady-state error due to the absence of integral action. The **PID controller** delivers the most **balanced performance**, combining the rapid response of the PD controller with the zero steady-state error of the PI controller. Overall, the PID configuration achieves the best trade-off between **speed, stability, and accuracy**, making it the most effective for precise DC motor speed control.

Discussion:

The **root-locus analysis** clearly explains the observed controller behaviors. The **PI controller** introduces a zero that shifts the dominant poles leftward, enhancing response speed while its pole at the origin ensures zero steady-state error. The **PD controller** provides a phase lead that increases damping and minimizes overshoot, though it leaves a small steady-state offset. The **PID controller** successfully integrates both effects, positioning the poles in a region that meets desired performance goals such as minimal overshoot and a settling time below 0.05 seconds. In practical applications, it is advisable to include a **low-pass filter** on the derivative term to suppress noise amplification and to perform **robustness checks**, such as parameter sensitivity analysis, to ensure stable performance under system variations.

Conclusion:

The **root-locus-based controller design** effectively guided the selection of PI, PD, and PID configurations for DC motor speed control. The **PI controller** eliminated steady-state error with a moderate response speed, the **PD controller** achieved high speed and strong damping at the cost of steady-state accuracy, and the **PID controller** provided the best overall performance with **low overshoot, fast settling, and negligible steady-state error**. The optimized gain values successfully met standard transient specifications, demonstrating the inherent trade-offs between **speed, damping, and accuracy** in practical control system design.

References:

1. K. Ogata, Modern Control Engineering, 5th ed., Prentice Hall, 2010.
2. G. F. Franklin, J. D. Powell, and A. Emami-Naeini, Feedback Control of Dynamic Systems, 7th ed., Pearson, 2014.
3. N. S. Nise, Control Systems Engineering, 7th ed., Wiley, 2015.
4. K. J. Åström and T. Hägglund, PID Controllers: Theory, Design, and Tuning, 2nd ed., ISA, 1995.

Heaven's light is our guide

RAJSHAHI UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department Of Electrical & Computer Engineering

Course title:

Control System and Robotics Sessional

(MTE 4118)

Lab Report

Submission Date:

Submitted To:	Submitted By:
Md. Sakib Hassan Chowdhury	Syed Mahmudul Imran
Lecturer	Roll: 2010058
Dept of MTE, RUET	