## **DC MOTOR SPEED CONTROL(PID SIMULATION)**

PROJECT

Name:S.Meharaj

Year:3rd year

Branch:Electrical and Electronics Engineering

Institute:Annamacharya University

AIM: The primary aim of DC motor speed control code, particularly when using PID simulation is to precisely regulate a DC motor's speed according to a desired setpoint, while minimizing errors and disturbances.

DISCRIPTION: A DC motor speed control simulation using a PID (Proportional-Integral-Derivative) controller typically involves modeling the motor's behavior, defining the PID control algorithm, and then simulating the interaction between the controller and the motor to achieve a desired speed.

SOURCE CODE:

import matplotlib.pyplot as plt

import numpy as np

# Simulation parameters

setpoint = 100  # Desired motor speed (RPM)

Kp = 0.1      # Proportional gain

Ki = 0.01     # Integral gain

Kd = 0.05     # Derivative gain

dt = 0.1      # Time step for simulation (seconds)

sim\_time = 150 # Total simulation time (seconds)

# Controller initialization

integral = 0

previous\_error = 0

motor\_speed = 0  # Initial motor speed (RPM)

# Store data for plotting

time\_data = []

speed\_data = []

# Simulation loop

for t in np.arange(0, sim\_time, dt):

    # Calculate error

    error = setpoint - motor\_speed

    # PID calculations

    integral = integral + error \* dt

    derivative = (error - previous\_error) / dt

    output = Kp \* error + Ki \* integral + Kd \* derivative

    # Update motor speed (simplified model)

    # In a real system, you'd have a motor model here

    motor\_speed = motor\_speed + output \* dt

    # Store data for plotting

time\_data.append(t)

    speed\_data.append(motor\_speed)

    # Update previous error

    previous\_error = error

# Plot the results

plt.plot(time\_data, speed\_data)

plt.xlabel("Time (s)")

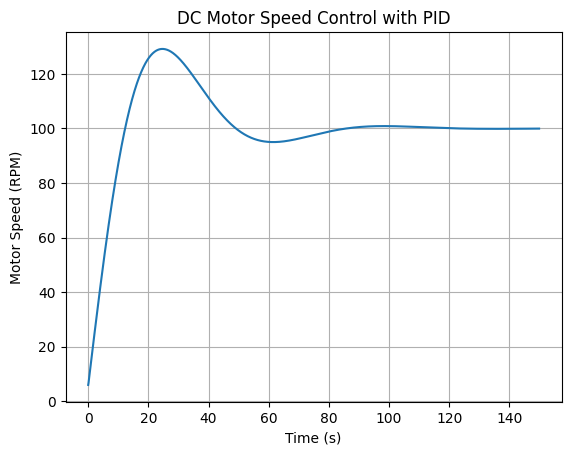
plt.ylabel("Motor Speed (RPM)")

plt.title("DC Motor Speed Control with PID")

plt.grid(True)

plt.show()

OUTPUT:



CONCLUSION: The simulations demonstrate that PID controllers can effectively regulate motor speed, reducing rise time, steady-state error, and overshoot, leading to improved system stability and performance.