project

STUDENT NAME: M.Mounika

BRANCH: Electrical and Electronics Engineering

YEAR: 3rd

INSTITUTION: Annamacharya University

**AIM:** DC Motor Speed Control(PID Simulation).

Description:-

Simulates a DC motor (electrical + mechanical dynamics) with a PID controller.

- Inputs: motor parameters and PID gains (edit at top of file or pass to simulate()).

- Outputs: plots of speed vs time, control (voltage) vs time, and printed performance metrics

- Applications: robotics (actuator/speed control), electric drives, motor control labs

**Program:**

import matplotlib.pyplot as plt

import numpy as np

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)

integral = 0

previous\_error = 0

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

# Store data for plotting

time\_data = []

speed\_data = []

integral = 0

previous\_error = 0

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

# Store data for plotting

time\_data = []

speed\_data = []

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

# Calculate error between setpoint and current motor speed

error = setpoint - motor\_speed

# Proportional term

    P\_out = Kp \* error

  # Integral term (accumulation of past errors)

  integral += error \* dt

   I\_out = Ki \* integral

    # Derivative term (rate of change of error)

  derivative = (error - previous\_error) / dt

    D\_out = Kd \* derivative

 # Calculate the total control output

    control\_output = P\_out + I\_out + D\_out

    # Simulate the motor speed response (simplified linear response)

    motor\_speed += control\_output \* dt

# Update the previous error for the next iteration

    previous\_error = error

    # Save data for plotting

  time\_data.append(t)

    speed\_data.append(motor\_speed)

# Plot the motor speed over time

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

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

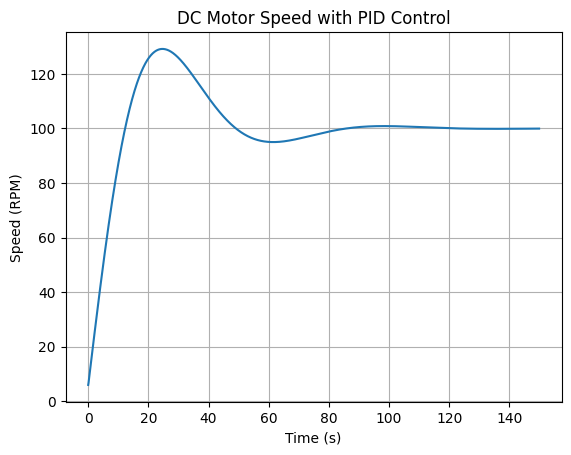
plt.xlabel('Time (s)')

plt.ylabel('Speed (RPM)')

plt.grid(True)

plt.show()

**OUTPUT:**

**Conclusion:**

Obviously the PID controller doesn't get the desired result. A good PID control calculation setting is shown. The positive result of a closed circle is displayed. This paper Demonstrates strategies to plan speed control for DC vehicles with the use of GA and to improve using GA and to resolve over time and fixed condition error. The end result is further shown to make and reflect an increase in GA.