PID Control of a DC Motor
Simulation Study Using MATLAB Simulink

Institute: National Institute of Technology Rourkela
Department: Electrical Engineering
Student Name: Sridhatri Gunupuru

Abstract:
This project presents the modeling, simulation, and performance analysis of a DC motor controlled using a PID (Proportional–Integral–Derivative) controller. The purpose of the study is to design a closed-loop control system capable of achieving fast rise time, reduced steady■state error, and stable operation. Simulink is used to construct the DC motor transfer function and implement the PID controller using a modular block■level approach. The simulation successfully demonstrates system improvement with appropriate tuning of Kp, Ki, and Kd, and provides insights into the sensitivity of each gain parameter.

1. Introduction

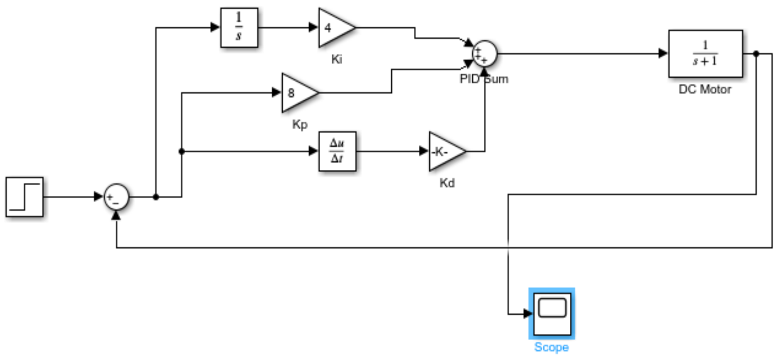
DC motors are widely used in industrial and robotic applications where precise speed and position control are required. A PID controller is one of the most commonly used control strategies because of its simplicity, robustness, and ability to handle a wide range of operating conditions. The aim of this project is to: Model a DC motor using its transfer function. Design a PID controller using basic building blocks. Analyse the closed■loop system response. Observe how tuning of Kp, Ki, and Kd affects stability and performance.

2. DC Motor Modeling

The DC motor is represented using a first■order transfer function of the form:

$G(s) = 1 / (s + 1)$
This simplified transfer function represents a motor with unity gain and a time constant of 1 second. Although simplified, it is sufficient for demonstrating PID control behavior. The motor block in Simulink contains this exact transfer function and takes the control input generated by the PID controller.

Block Diagram Used



3. PID Controller Design

The PID controller is implemented using three separate gain paths: **Proportional (Kp):** Depends on present error. **Integral (Ki):** Eliminates steady■state error using integration. **Derivative (Kd):** Predicts future error and improves stability. The controller output is the weighted sum of the three control actions:

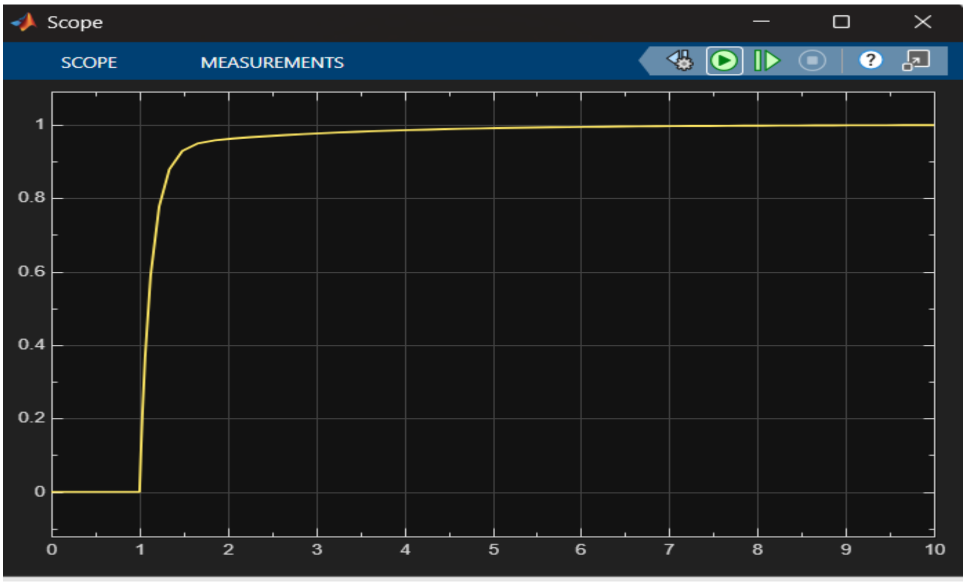
$$\mathbf{u(t) = Kp \cdot e(t) + Ki\int e(t)dt + Kd \cdot de(t)/dt}$$

A summing block is used to combine these terms. The error signal is generated by comparing reference input with motor output.

4. Simulation Setup

The Simulink model includes: Step input (reference speed) Error computation block Kp, Ki, Kd gain blocks Integrator for Ki Derivative block for Kd DC motor transfer function Scope for output visualization The chosen PID values for final response: Kp = 8 Ki = 4 Kd = 1 These values were tuned manually by increasing Kp to improve rise time, adding Ki to eliminate residual steady■state error, and including Kd to smooth oscillations.

Output Response



5. Results & Discussion

The output shows: **Fast rise time:** The motor reaches 90% of the final value within approximately 1.8 seconds. **Smooth settling:** Minimal oscillation after fine tuning of Kd. **Zero steady■state error:** Achieved due to the presence of the integral term Ki. **Stable response:** No drifting or unstable oscillations are observed. As Kp increases, the response becomes faster but with a risk of overshoot. Ki eliminates steady■state error but too much causes oscillations. Kd smooths the system but excessively high values amplify noise. The chosen values achieved a clean and stable response.

6. Conclusion

This project successfully demonstrates the design and simulation of a PID controller for a DC motor using Simulink. The controller effectively improves system performance, minimizing steady■state error and ensuring stable operation. The project also highlights the importance of PID tuning and how each parameter affects system behavior. This simulation framework can be extended for: Real DC motor hardware experiments using Arduino or STM32 Anti■windup PID Advanced control strategies like Fuzzy■PID or Model Predictive Control Overall, the project provides a solid foundation for future control system design and research.