Design and Evaluation of a PID Controller for Electrical Systems Using MATLAB

Arjun Ray[1], Debabrata Sarkhel[2], Chandan Adhikari[3], Sagar Singha[4], Aritra Bera[5,] *Rahul* Kumbhakar[6]
[1]*Lecturer, Swami Vivekananda School of Diploma, Durgapur, West Bengal- 713212*
[2]*Assistant Professor, Swami Vivekananda School of Diploma, Durgapur, West Bengal- 713212*
[3,4,5,6]*Diploma, Swami Vivekananda School of Diploma, Durgapur, West Bengal- 713212*

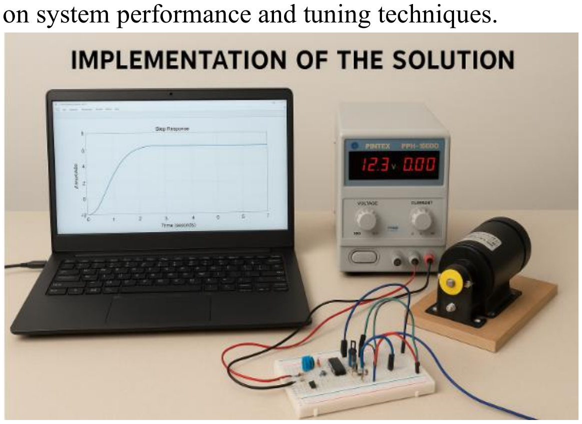
***Abstract*— In this study, a proportional-integral-derivative (PID) controller for electrical system applications is designed, simulated, and its performance assessed using MATLAB/Simulink. The study examines comparative performance measures, system response characteristics, and controller tuning techniques. The findings show that PID-based control is still a reliable and effective way to manage electrical systems, especially in power converter and motor control applications.**

***Index Terms*— Uses of teachable machines, machine learning, artificial intelligence, image recognition, and prediction models**

I. INTRODUCTION

Because of their ease of use, dependability, and efficiency in managing dynamic systems, PID controllers are frequently utilized in electrical engineering. For modeling, analyzing, and implementing PID controllers in a variety of electrical applications, including DC motor speed control, inverter output regulation, and generator voltage management, MATLAB/Simulink offers a robust platform.

The modeling and simulation of a PID controller in MATLAB is examined in this study, with an emphasis on system performance and tuning techniques.



II. OVERVIEW

The significance of PID controllers in power electronics and industrial automation has been demonstrated by earlier research. The Ziegler–Nichols approach, the Cohen–Coon method, and optimization-based tuning with MATLAB's PID Tuner are popular tuning techniques. In order to improve stability under changing loads, modern works concentrate on adaptive and auto-tuned PID schemes.

III. METHODOLOGIES

3.1 MATLAB System Modeling

A typical second-order electrical system or DC motor model is utilized to assess the PID controller. A DC motor transfer function, for instance, can be written as follows:

$G\,(s) = K\,(Ls + R)\,(Js + B)$

G(s) = (Ls+R)(Js+B) K

The MATLAB's built-in transfer function commands and Simulink blocks are used to create this model.

3.2. Design of PID Controllers:

The equation for the PID controller is:

$u\,(t) = Kp$

$e\,(t) + Ki\; e\,(t)dt + Kdde\,(t)dt$

u(t) = K p The

e(t) + K i

e(t)dt+K d The

dt de(t) The

Where:

K p- The Gain in proportion

K i.- Integral gain

K d- The Gain from derivatives

The PID controller can be implemented in MATLAB by using:
PID Controller block in Simulink
The MATLAB script's pid() function

3.3. Techniques for Tuning:
The Ziegler-Nichols Method is based on period and final gain.

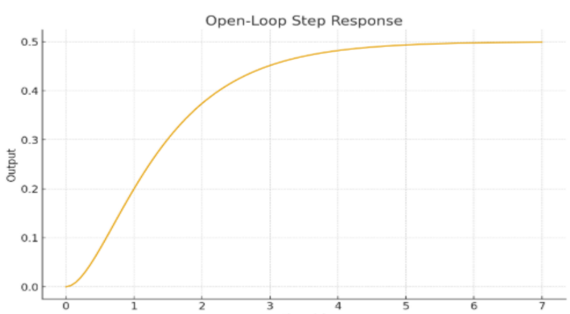
PID Tuner App: Offers automated optimization and tweaking.

Manual tuning: Changing gains to reduce settling time and overshoot.

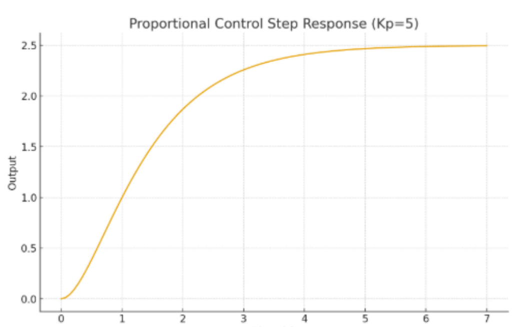
IV. SIMULATION & FINAL RESULTS

MATLAB/Simulink was used to run simulations on a model of an electrical system. Among the important performance indices are:
Rise time
Time spent settling
Overshoot
Error in a steady state

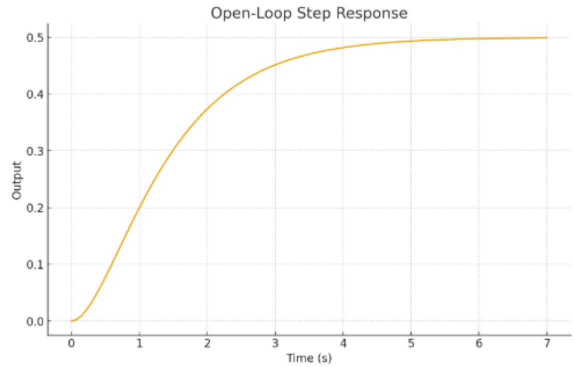
5.1 Without a Controller
The system has a large steady-state error and slow response time.



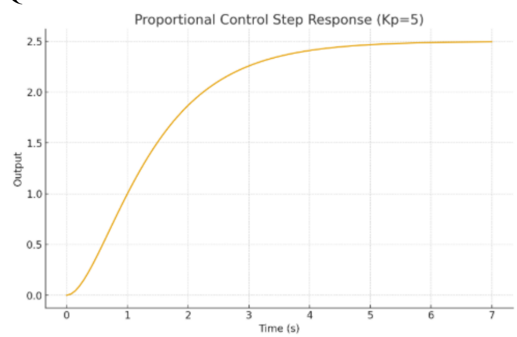
5.2 Using Only Proportional Control increases rise time but causes significant overshoot.



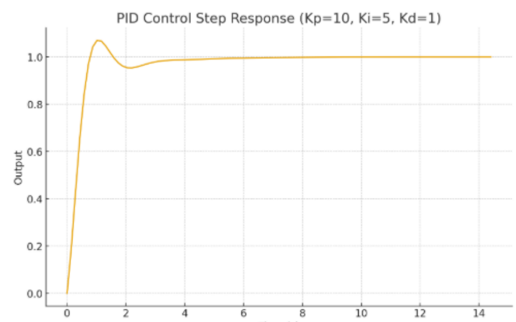
5.3 Complete PID Management
The system accomplishes:
Reduce the overshoot



Quicker time to settle



Minimal inaccuracy in the steady state



Simulink plots demonstrate how the PID controller greatly improves system performance.

VI. IMPLEMENTATIONS OF THE SOLUTION

The electrical plant was first modeled using its transfer function in MATLAB/Simulink in order to apply the solution. The built-in PID block was then used to install a PID controller, and the PID Tuner tool was used to adjust its gains. A closed-loop feedback system was developed, and system response was monitored through simulations. In comparison to the open-loop and proportional-only scenarios, the results demonstrated that the PID controller enhanced rise time, decreased overshoot, and eliminated steady-state error.

ACKNOWLEDGMENT

We are appreciative of our institution, Swami Vivekananda School of Diploma, for giving the resources and assisting us in conducting tests until the project was finished, as well as our sir, Mr. Arjun Ray, for mentoring us during this project.

REFERENCES

[1] MATLAB Documentation – Control System Toolbox
[2] K. Ogata, *Modern Control Engineering*
[3] Ziegler, J. G., & Nichols, N. B. – PID Tuning Methodology