Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

Development of Prototype Based Closed Loop PID Control for DC

Bhumika Godre[1], Yogita Barapatre[2], Sujal Pakhide[3], Dr. Chandrakant Rathore[4]

[1,2,3] Student, Electrical Engineering Department, S. B. Jain Institute of Technology, Management and Research, Nagpur
[4]Assistant Professor, Electrical Engineering Department, S. B. Jain Institute of Technology, Management and Research, Nagpur

Abstract:
This paper proposes prototype-based closed-loop PID control system and implemented using an Arduino Uno microcontroller for precise speed regulation of a DC motor. Real-time speed feedback is obtained from an encoder, and a PID algorithm dynamically adjusts the PWM signal to control the motor through an L298N driver. The proposed system ensures fast response, minimal steady-state error, and stable operation. Experimental results confirm the effectiveness of the developed control strategy.

Keywords: DC motor, PID controller, closed-loop control, Arduino Uno, speed control, PWM, encoder feedback.

1. INTRODUCTION

DC motors are extensively used in modern engineering and industrial applications due to their simple construction, high efficiency, reliable performance, and excellent speed–torque characteristics. They find widespread application in household appliances, robotics, electric vehicles, conveyor systems, manufacturing industries, and automated production lines. In such applications, accurate and stable speed regulation is crucial to ensure smooth operation, enhanced productivity, energy efficiency, and system reliability. Precise motor control also contributes significantly to improved product quality and operational safety. Despite their widespread usage, conventional open-loop control techniques are insufficient for achieving precise speed control under practical operating conditions. Open-loop systems lack feedback mechanisms and therefore cannot detect real-time speed variations or compensate for external disturbances such as load changes, supply voltage fluctuations, and mechanical wear. These limitations result in poor dynamic response, speed instability, large steady-state errors, and reduced system efficiency**.** Consequently, there is a strong requirement for a closed-loop control strategy that continuously monitors motor performance and applies corrective actions in real time. Among various control methods, the Proportional–Integral–Derivative (PID) controller is widely preferred due to its simple structure, robustness, ease of implementation, and reliable control performance.

The development of a prototype-based closed-loop PID control system for DC motor speed regulation using an Arduino Uno microcontroller is presented. The system employs an incremental encoder to provide continuous real-time speed feedback. The measured speed is compared with the desired reference value, and the resulting error is processed through a PID control algorithm. Based on the controller output, the Pulse Width Modulation (PWM) signal is dynamically adjusted and applied to the motor via an L298N motor driver to achieve accurate speed regulation. Additionally, a user interface comprising potentiometers, push buttons, and an I2C LCD display is incorporated to enable real-time monitoring and manual tuning of PID parameters. Experimental validation demonstrates fast dynamic response, minimal steady-state error, and stable operation under varying load conditions**,** confirming the effectiveness of the proposed control strategy.

2. METHODOLOGY

A structured experimental platform is developed to acquire and analyze the time-domain speed response of a DC motor, enabling the formulation of a comprehensive closed-loop control model. The overall architecture of the proposed system, illustrating the signal flow and control structure, is presented in Figure. 1

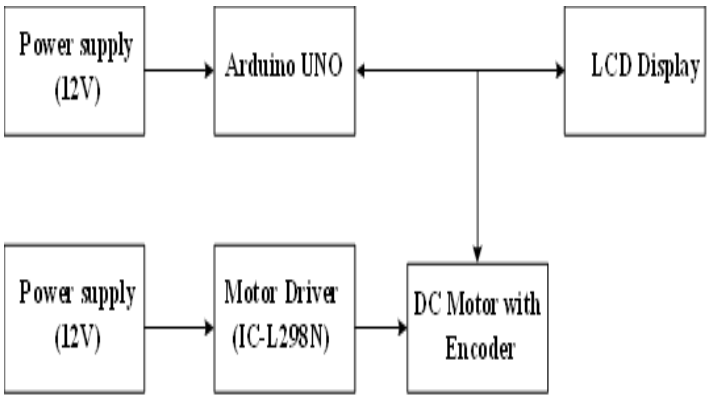


Figure 1. Proposed block diagram

The system is designed to implement a real-time feedback-based PID control strategy for accurate and stable speed regulation.

The system comprises a 12 V power supply, Arduino Uno microcontroller, L298N motor driver, DC motor with incremental encoder, and an LCD display. Separate power supplies are used for the control unit and the motor drive to ensure stable and noise-free operation. The encoder continuously measures the motor speed and provides feedback to the microcontroller. The measured speed is compared with the reference value, and the resulting error is processed using a PID algorithm.

The Arduino generates a PWM signal based on the controller output, which is applied to the motor driver to regulate the motor speed. The LCD display provides real-time monitoring of key parameters, including reference speed, actual speed, and control error. Experimental testing is conducted under varying speed commands and load conditions to evaluate system response, stability, and control performance.

3. PROPORTIONAL INTEGRAL DERIVATIVE (PID) CONTROLLER

The proportional integral derivative (PID) controller is the most used controller in robotics and industrial. It is because the controller is easy to understand, easy to be implemented in simulation, and hardware implementation, and has a simple structure but could give an excellent response. The PID controller signal in time-domain could be written as

$$u = K_p e + \frac{K_p}{T_i}\int_0^t e(t)\,d + K_p T_d \frac{de(t)}{dt} \quad (1)$$

Or it can be written as,

$$u = K_p e(t) + K_i \int_0^t e(t)\,dt + K_p \frac{de(t)}{dt} \quad (2)$$

where

$$K_i = \frac{K_p}{T_i},\ K_d = K_p T_d \quad (3)$$

Where the variable K_p is the proportional constant, the variable K_i is the integral constant, the variable K_d is the derivative constant.

4. WORKING PRINCIPLE

The system works on the closed-loop control principle, where the Arduino Uno continuously compares the setpoint speed with the actual motor speed received from the feedback sensor. The difference between them called the error, is processed by the PID controller.

The PID controller then computes a corrective control output based on:

$$\text{Control Output} = K_p \times e(t) + K_i \times \int e(t)\,dt + K_d \times \frac{di}{dt} \quad (4)$$

The PID algorithm calculates the control signal using Proportional (K_p), Integral (K_i) and Derivative (K_d) terms to minimize the error. This control signal is converted into a PWM output, which drives the DC motor through the L298N motor driver.

5. SYSTEM IMPLEMENTATION

The hardware implementation for this undertaking is based on an Arduino Uno, that acts as the main control hardware. It takes user inputs, runs the PID control algorithm, and emits PWM signals to control the speed of a DC motor.

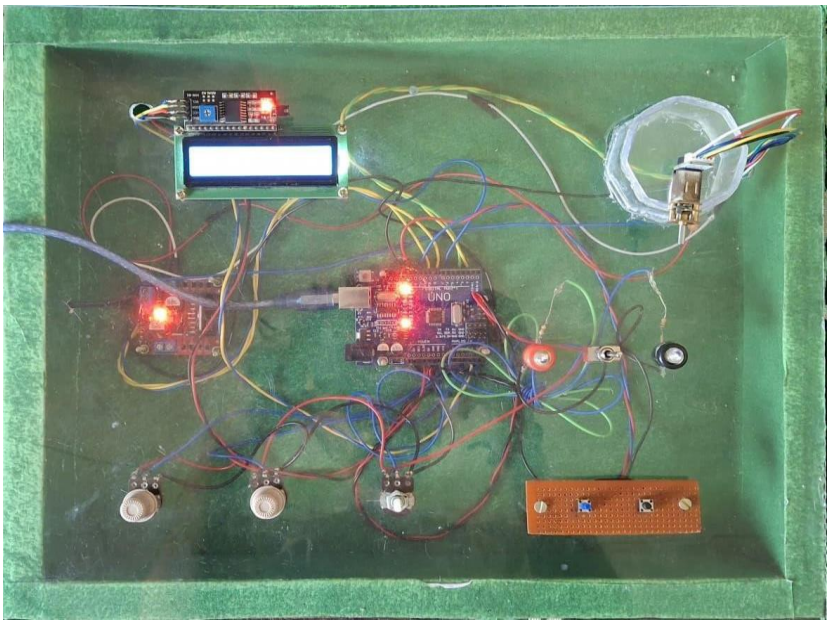
An L298N motor driver is connected between the Arduino and the DC motor. The motor driver receives the PWM signal from the Arduino and provides the power to drive the motor, enabling operation in a safe and efficient manner.

Three potentiometers are attached to the analog pins of the Arduino as an adjustment for the various PID constants, K_p, K_i, K_d as a means for the user to adjust the response of the control system in real time.

Two push buttons are included for increasing and decreasing the setpoint for the speed of the motor.

The setpoint value selected along with the actual speed of the motor and PID constants are displayed on the 16×2 I2C LCD display to show the performance of the system in real time. The prototype hardware setup is shown in Figure 2.

Figure 2. Hardware setup



The system is powered by a 12V DC adaptor where the Arduino is powered at 12V and the motor driver is powered at 12V for the DC motor. All components are mounted to a prototype board with proper wiring to ensure stability, safety and troubleshooting in case of errors or malfunctions.

6. RESULT AND DISCUSSION

The proposed system is tested under different operating points (Set point (rpm)) with changing value of PID controllers. These tests were performed to determine its effectiveness in maintaining the desired speed (actual set point). The results are under the above-mentioned points are recorded and presented below in the form of tables and figures.

Table 1, displayed the change in variation of proportional (K_p) value. It is noted from the table that as the value of K_p is increased the value of the steady-state error is decreased. However, the motor speed not reached to set point (speed).

Table 1: Proportional Control System Response

K_p	Set Point (RPM)	Motor Speed (RPM)	Steady State Error (RPM)
0.2	150	63	87
0.4	150	88	62
0.6	150	108	42
0.8	150	115	35
1	150	123	27

Based on Figure 3 it is observed that increase in proportional value can decrease the rise time value, increase overshoot value, and decrease the steady-state error. The result does not have a settling time because the response system cannot reach the set point

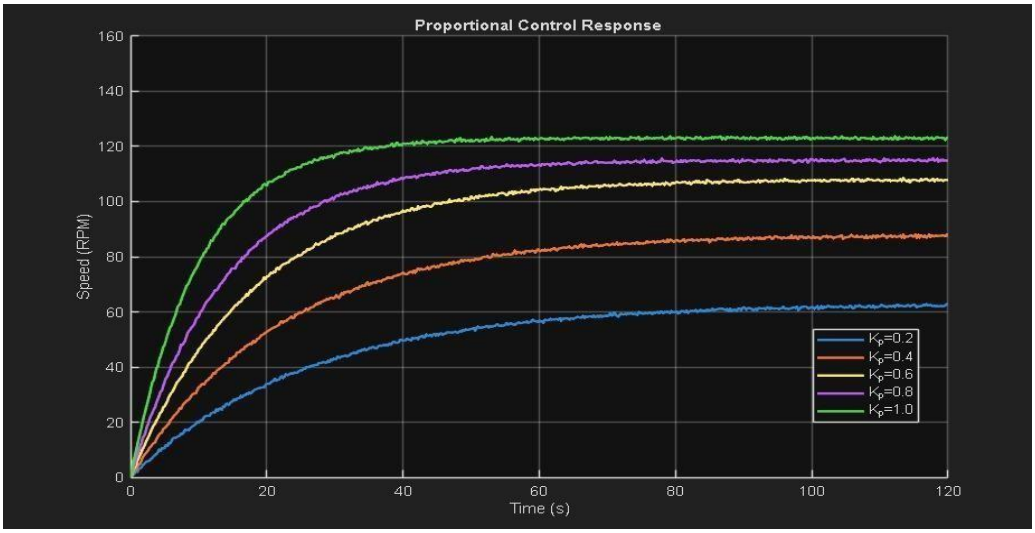


Figure 3. The system's response of Proportional control

The second test is to analyze the effect of increased integral value. Table 2, displayed the change in variation of proportional (K_i) value. It is noted from the table that increased integral value can decrease and eliminate the steady-state error. However, the overshoot is not decreased.

Based on Figure 3 it is observed that increased integral value can decrease the rise time value, increase overshoot value and eliminate the steady-state error.

Table 2: Integral Control System Response

K_i	Set Point (RPM)	Motor Speed (RPM)	Steady State Error (RPM)
0.01	150	108	42
0.02	150	115	35
0.03	150	138	12
0.04	150	148	2
0.05	150	150	0

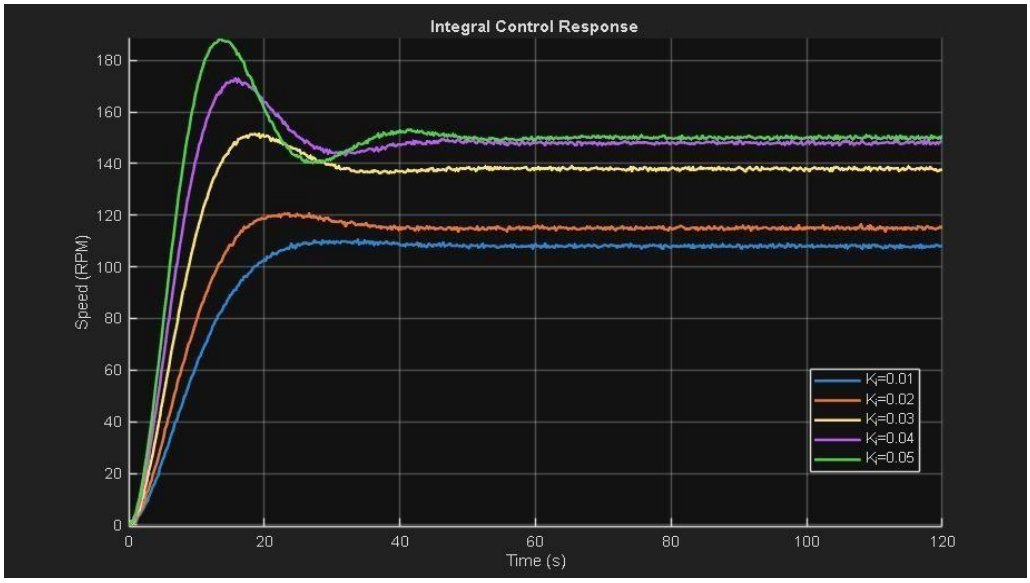


Figure 4. The system's response of Integral control

The third is analyzing the effect of increased derivative value. The result is shown in Figure 5. The system response is in Table 3. Based on Figure 5 and Table 3, it could be known that increased derivative value can decrease the overshoot a little bit. The derivative controller does not affect the response system significantly, probably because of the use of less sample time.

Table 3 Derivative Control System Response

K_p	Set Point (RPM)	Motor Speed (RPM)	Steady State Error (RPM)
0.01	150	150	0
0.02	150	146	4
0.03	150	144	6
0.04	150	145	5
0.05	150	147	3

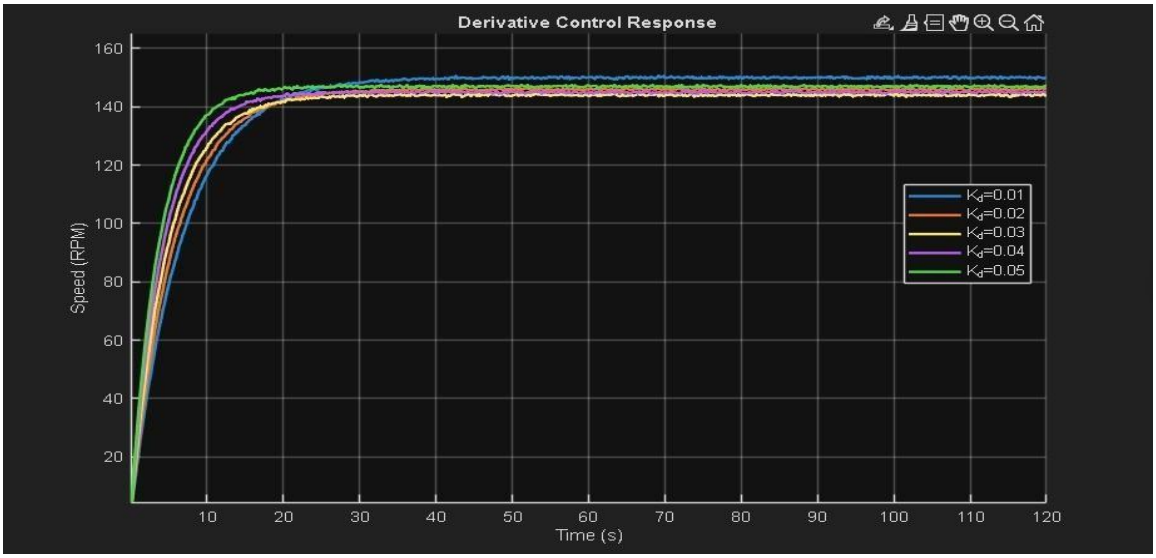


Figure 5. The system's response of Derivative control

The final test was performed with the combination K_p, K_i and K_d. The best proportional integral derivative parameter value is shown in Table 4. The result of the system response is shown in Figure 6.

Table 4. PID Control System Response

Value	Set Point (Motor Speed In RPM)	System Response			
		Risetime	Setting Time	Overshoot	Steady State Error
K_p =1.4 K_i = 0.6 K_d =0.03	150	2.4511 s	11.5 s	2.0 s	2

When all proportional–integral–derivative (PID) terms are active, with parameters set to K_p = 1.4, K_i= 0.6 and K_d = 0.3 the system works best. The motor speed increases smoothly, showing little to no overshoot. The derivative term (K_d) aids in response stabilization which reduces abrupt shifts and lessens oscillations. Because of this, the motor speed gets close to the 150 RPM target and quickly stabilizes around 147–150 RPM. The graph shows how full PID control offers quick response times, better stability, low overshoot and small steady-state error.

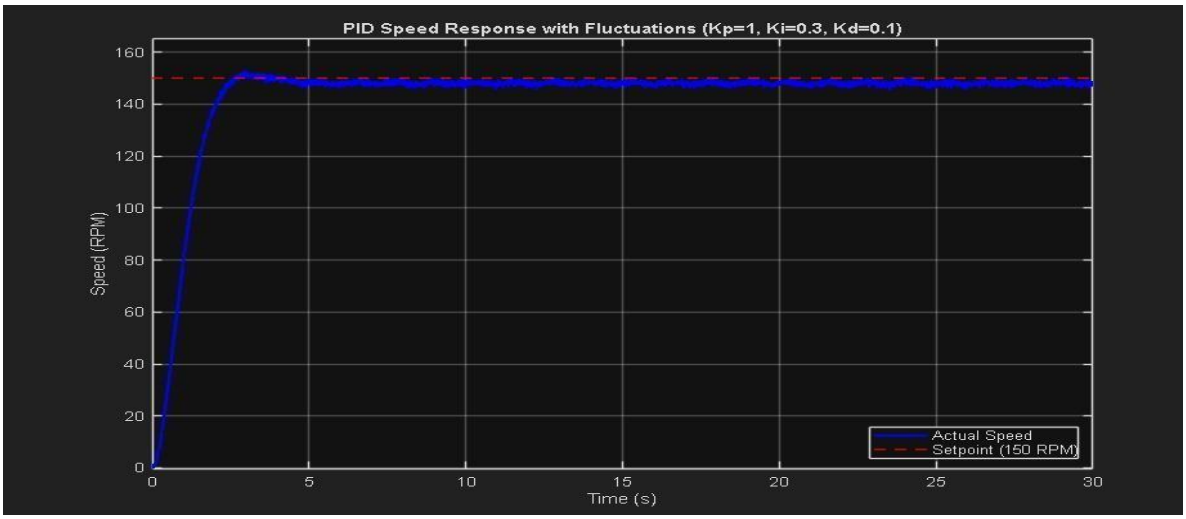


Figure 6. The system's response of PID control

7. DISCUSSION

Speed control of a DC motor was done using closed-loop PID controller with manual tunning of K_p, K_i and K_d parameters. The feedback from the encoder was successfully read by the Arduino Uno, which then updated the PWM output to maintain the desired speed. The parameters of the PID controller (K_p, K_i and K_d) were manually tuned using potentiometers. The results obtained for different value of PID parameters are noted and it was observed that motor speed varies as per the controller values. Through this experimentation it was noted that the system demonstrated accuracy and stability. The set point (speed of motor) was followed closely with minimal steady-state error.

8. CONCLUSION

This work outlines the design and configuration of a closed-loop PID control system to regulate rotational speed in a DC motor using an Arduino Uno and encoder feedback. It quantifies motor speed contrasts it with a specified reference and adjusts the control signal to minimize deviations. Experimental results are presented that show that without any PID control, the motor response was poor-it did not reach the set target speed reliably-and unstable, which therefore reflects upon the use of an open-loop system.

REFERENCES

1. Reis, M. R., Silva, F. S., Araújo, W. R., Calixto, W. P., Araújo, D. S., Mendes, I. N., & Rose, R.M. "Speed control for direct current motor using optimization tuning for PID controller," In 16th IEEE International Conference on Environment and Electrical Engineering (EEEIC), pp. 1-4, June 2016.
2. Ma'arif, A., Raharja, N. M., Rosyady, P. A., Baswara, A. R. C., & Nuryono, A. A., "Control of dc motor using proportional integral derivative (pid): Arduino hardware implementation," In 2nd IEEE International Conference on Industrial Electrical and Electronics (ICIEE), pp. 74-78, October 2020.
3. Achanta, R. K., & Pamula, V. K. "DC motor speed control using PID controller tuned by jaya optimization algorithm," In IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), pp. 983-987, September 2017.

4. Sahputro, S. D., Fadilah, F., Wicaksono, N. A., & Yusivar, F.,“ Design and implementation of adaptive PID controller for speed control of DC motor,” In 15th IEEE International Conference on Quality in Research (QiR): International Symposium on Electrical and Computer Engineering, pp. 179-183, July 2017.
5. Khubalkar, S. W., Chopade, A. S., Junghare, A. S., & Aware, M. V.,“ Design and tuning of fractional order PID controller for speed control of permanent magnet brushless DC motor,” In IEEE First International Conference on control, measurement and Instrumentation (CMI) pp. 326-320, January 2016.
6. Gowthaman, E. and Balaji, C.D,“Self-tuned PID based speed control of PMDC drive,”In IEEE International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), pp. 686-692, March 2013.
7. Cao F, Wang Y. “Study and application of fuzzy PID control-based for FFU motor speed regulation control system,” In IEEE International Electric Machines & Drives Conference (IEMDC), pp. 160-163, May 2011.
8. Yu G, Li Y, Yao Y, Wang Q, Fu L, “Design on PID Control System of DC Motor Based on MC9S08AW32,” In IEEE International Conference on Computer Technology, Electronics and Communication (ICCTEC), pp. 983-987, December 2017.