iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82113

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Speed Control of BLDC Motor Using ATmega2560

Yelugam Akhila[1], Gundu Sadwika[2], Gutoju Sandhya[3], T. Muralikrishna[4]
[1, 2, 3]*Undergraduate Students, EEE Department, Chaitanya Bharathi Institute of Technology (A), Osmania University, Hyderabad, India*
[4]*Associate Professor , EEE Department, Chaitanya Bharathi Institute of Technology (A), Osmania University, Hyderabad, India*

Abstract: ***The development of an ATmega2560-based closed-loop speed control system for a brushless three-phase motor is presented in this paper. The proposed method makes use of pulse width modulation (PWM) to drive the motor, a proportional-integral-derivative (PID) controller to maintain a specific speed in a closed loop, and a Hall effect sensor (which is triggered by a permanent magnet on the rotor's shaft) to measure the rotor speed in real-time .The Hall effect sensor calculates the motor's rotor speed to provide an accurate RPM reading between 3000 and 6000. The drive stage uses N-channel MOSFETs and an IR2104 half-bridge gate driver to create an efficient three-phase inverter for motor commutation. Additionally, a 16 X 2 LCD displays the target and actual speed of the motor continuously. Experiments show that the PID-controlled closed-loop system reduces the steady-state error of the PID controlled closed-loop speed control system compared to the open-loop speed control system. The PID controlled closed-loop speed control system can maintain the motor at a speed of ±1.5% of the target speed (set-point) with varying load. This motor control system is cost-effective, compact, and can be used to develop scalable, low-cost motor speed control systems for electric vehicles, robotics, and industrial automation.***

Keywords: BLDC motor; ATmega2560; PWM speed control; PID controller; Hall Effect sensor; closed-loop control; IR2104 gate driver.

I. INTRODUCTION

BLDC motors have become the preferred prime mover for many of the latest engineering applications that range from precision robotics to unmanned aerial vehicles, electric vehicle traction systems, and industrial servo systems. The main advantages of using BLDC motors over conventional brushed DC motors are their lack of mechanical commutation, which results in better electrical efficiency (generally 85% to 95%), a longer life span, lower acoustic noise, and the ability to operate at high speeds with high torque.The precise speed control of BLDC motors, however, presents additional challenges compared to brushed motors. In order to properly control the BLDC motor’s speed, proper electronic commutation must be done based on real-time knowledge of the rotor’s position; this means the motor’s speed-torque characteristics are sensitive to variations in supply voltage and load transients. Open loop PWM control, while relatively easy to implement, does not provide repeatability because it cannot maintain the defined speed when there is a disturbance; therefore, closed loop control would be necessary for any applications requiring repeatable performance. Microcontroller-based systems are now the standard for BLDC drive electronics. Modern devices combine all necessary parts, including high-resolution PWM timers, analog-to-digital converters (ADC), hardware interrupt inputs, and communication interfaces, all on a single chip. The ATmega2560, which is at the heart of the Arduino Mega 2560 board, provides fifteen hardware PWM outputs, sixteen 10-bit ADC channels, and four hardware UARTs. This makes it ideal for prototyping multi-axis or multi-peripheral motor control.

The main focus of this work is the integration of an external Hall Effect sensor and a shaft-mounted permanent magnet for speed sensing. This approach eliminates the need for built-in rotor sensors or optical encoders. The speed signal measured is fed into a discrete PID controller that runs completely in firmware. This setup effectively manages load disturbances within the 3,000 to 6,000 RPM range. A 16×2 LCD and a serial interface enable real-time diagnostics. The drive hardware, which is based on IR2104 half-bridge drivers and discrete MOSFETs, provides a clear and easy-to-replicate circuit design.

II. LITERATURE REVIEW

The literature on BLDC motor speed control falls into three main categories: control algorithm development, commutation strategy, and hardware platform selection. Early research by Mohan et al. [4] laid the groundwork for PWM-based voltage control of brushless drives. They showed that changing the duty cycle leads to a linear relationship with average phase voltage and, in turn, with no-load speed. Later studies highlighted the limitations of purely open-loop methods when motors are under load. This spurred the move towards feedback control.

Proportional-Integral-Derivative (PID) control is still the most common closed-loop strategy. Its popularity comes from its simplicity, straightforward tuning process, and reliability despite modeling uncertainties. Kumar et al. [5] developed a PID-regulated BLDC drive using an ATmega2560 platform. They found steady-state speed errors below 2% over a 500–3000 RPM range with Hall sensors placed inside the motor housing. This work builds on that by extending the RPM range to 3,000–6,000 RPM with an external sensor setup, which can be used for motors that do not have built-in Hall sensors.

Sensorless control using back-EMF zero-crossing detection has been widely studied as a lower-cost option. While it works well for steady-state and high-speed operations, sensorless methods struggle with starting torque and can be unstable at low speeds. The sensor-based approach used here provides reliable feedback from a standstill, making it a better choice for applications that need precise low-speed or frequent-start operation.

Maula et al. investigated a 48 V, 350 W BLDC drive with PID control using sensorless techniques. They showed good tracking performance but found commutation glitches during load changes. The current design uses an external Hall sensor and a shaft-mounted magnet, which removes commutation uncertainty by giving a clear, debounce-free digital pulse for RPM calculations. Prasetyo et al. further validated the Model Predictive Control (MPC) approach on the same ATmega2560 platform. They achieved a faster response but at the cost of much higher computational demands, highlighting the lasting practical benefits of PID for resource-limited embedded systems.

The reviewed literature shows a gap in cost-effective, external-sensor-based closed-loop BLDC speed control systems that focus on moderate-to-high RPM ranges and provide full experimental validation on the ATmega2560 platform. This paper aims to fill that gap.

III. PROPOSED SYSTEM DESIGN

A. System Architecture Overview

The proposed system has five functional layers: (i) user input through a potentiometer, (ii) speed sensing with an external Hall Effect sensor, (iii) closed-loop PID computation and PWM generation in the ATmega2560, (iv) power amplification using the IR2104-MOSFET drive stage, and (v) real-time output on the 16×2 LCD. Fig. 1 shows the top-level block diagram.

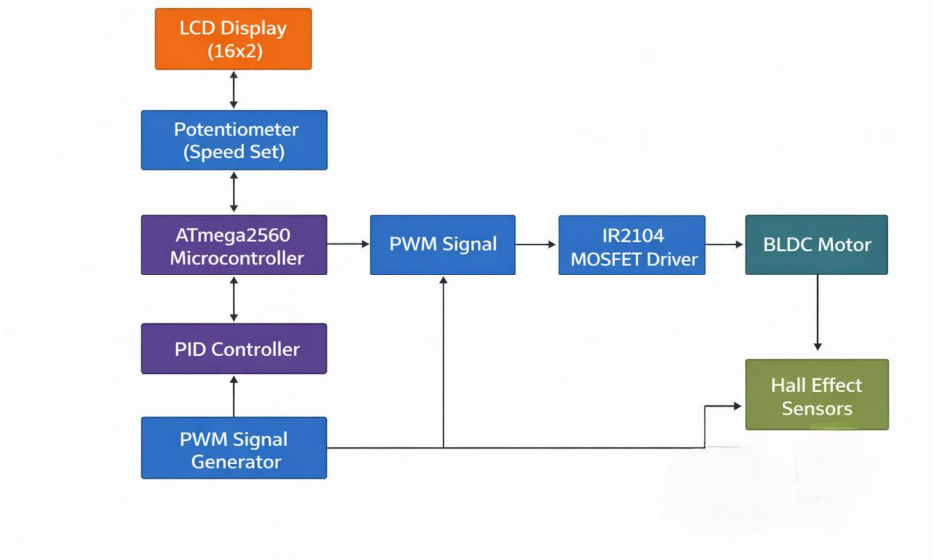


Fig. 1: Block diagram of the proposed BLDC speed control system.

B. BLDC Motor Working Principle

BLDC (Brushless Direct Current) motors are three-phase motors with a permanent magnet rotor; a stator that has windings; and a rotor that rotates in sync with the magnet. The BLDC motor generates torque by energizing the stator coils in pairs and sequentially (in synchrony with the rotors position). This movement creates a rotating magnetic field, which causes the permanent (the rotor) to be attracted by the rotating magnetic fields (the stator).

Unlike traditional brushed motors, where commutation occurs through a mechanical device (brushes), in brushless motors, commutation is completed electronically (with no wear on brushes or resistive losses).

In six-step (trapezoidal) commutation of a BLDC motor, you will have three phases that are energized for six electrical cycles. Every phase will have one positive phase, one negative phase, and one floating (not connected) for each of the six steps, providing a high level of torque and minimal ripple.

To control phase voltage and motor speed, the ATmega2560 uses Hall Effect sensors to determine what state the motor is at. The Hall Effect sensors send a pulse train to the ATmega to determine the number of electrical states. The ATmega2560 then uses the pulse train produced from the Hall Effect sensor to determine the PWM duty cycle. At this point, the motor speed is adjusted by controlling the average phase voltage.

C. PWM-Based Speed Control

Control of the speed is attained by altering the duty cycle δ of the PWM signal fed to the high-side MOSFETs. The phase average voltage $V_{\{avg\}}$, with respect to the duty cycle and voltage, can be represented as follows:

$$V_{\{avg\}} = \delta V_{\{dc\}}, where\ 0 \leq \delta \leq 1 \qquad (1)$$

As the speed of the motor without load varies directly with phase average voltage $N \propto V_{\{avg\}}$, the variation in duty cycle δ from 0 to 1 varies the speed from minimum to maximum. The ATmega2560 timer/counter peripherals produce PWM signals at a frequency of 16 kHz that is above the audible range using 8-bit fast PWM mode.

D. Hall Effect Sensor and RPM Calculation

A Hall Effect switch, an omnipolar Hall Effect switch (e.g., A3144 or equivalent), is mounted near the motor shaft. The motor shaft has a small disc magnet inserted in it. The magnet is mounted axially on the shaft (concentric). As the magnet rotates past the Hall Effect switch, a logic-low pulse will be generated for each revolution of the motor. The ATmega2560 uses an external interrupt (INT0) to capture the falling edge of each pulse and records the elapsed time (i.e., T in microseconds) between each pulse. RPM is calculated as follows:

$$RPM = 60 * 10^6 / T \qquad (2)$$

Where T is the inter-pulse period in microseconds. If the RPM is within the range of 3,000–6,000 RPM, T ranges between 10,000–20,000μs and thus the tolerance of the RPM is ±15 RPM, which is adequate for meeting the control requirements. A software timeout will clear the RPM reading when there is not a pulse received during a 100ms time period, thereby preventing stale-speed errors at low duty cycles.

E. PID Controller Design

The discrete time implementation of the PID controller calculates the correction value for the duty cycle change at each sampling instant Ts (Ts=50ms). For the calculation of the correction u[k], we consider $e[k] = Nref[k] - Nact[k]$ to be the speed error at sampling instant k. Therefore,

$$u[k] = K_p.e[k] + K_i.T_s.\sum e[k] + K_d.\{e[k] - e[k-1]\}/\{Ts\} \qquad (3)$$

The PID controller gains K_p= 0.85, K_i = 0.12, and K_d = 0.05 have been selected based on empirical studies using the Ziegler-Nichols step-response tuning method. An anti-windup limiting scheme confines the integral accumulator output to ±255 PWM counts, which prevents integrator saturation for sudden set-point changes. The calculated correction u[k] is then added to a feed-forward duty cycle estimated from a speed-to-voltage look-up table.

F. Drive Circuit — IR2104 and MOSFETs

The motor phases used to control the motor are connected to an H-Bridge by three dual high/low side gate driver ICs as described in the IR2104 data sheet. The IR2104 accepts one PWM input signal and one direction input signal to generate both high side and low side output gate signals in addition to programmable deadtime to eliminate shoot-through. The high side output gate driver has a bootstrap charge pump, which increases the output gate voltage by 10V to 15V over the motor supply voltage (12V), thus ensuring that the N-Channel MOSFETs are fully enhanced. The three-phase H-Bridge is made up of six STP75NF75 N-Channel MOSFETs that are rated for $V_{\{dc\}}$ = 75V and have a DC on resistance of 14 mΩ. The peak gate current is limited by connecting a 10Ω resistor in series with each gate driver output, and their output signal reduces EMI. Each of the six TO-220 package MOSFETs has an output flyback diode that converts the energy stored in the inductance of the motor winding during turn-off into an energy that can be used to supply current to the MOSFET during turn-on.

IV. METHODOLOGY AND IMPLEMENTATION

A. Hardware Realisation

Table I summarises the full component bill of materials and functional roles.

TABLE I: Component List and Functional Roles

Component	Specification	Functional Role
ATmega2560	Arduino Mega 2560	Central controller; PWM generation, PID computation, LCD drive
BLDC Motor	3-phase, 12V	Actuator; 3000–6000 RPM operating range
IR2104	Half-bridge gate driver	High/low-side MOSFET gate drive with dead-time control
STP75NF75	N-channel MOSFET, 75V/80A	Phase switching elements forming 3-phase H-bridge
Hall Sensor (A3144)	Omnipolar switch, 5V	Shaft speed sensing via magnetic pulse detection
Potentiometer	10kΩ	Analogue speed set-point input via ADC channel A0
LCD Display	16×2, HD44780	Real-time display of target speed, actual speed
Power Supply	12 V DC, 5 A	Supplies motor drive stage; 5 V regulated for logic circuits

TABLE II: ATmega2560 Technical Specifications

Parameter	Value
Microcontroller Core	ATmega2560 (AVR 8-bit RISC)
Operating Voltage	5 V
Input Voltage Range	7–12 V
Digital I/O Pins	54 (15 PWM capable)
Analog Input Pins	16 (10-bit ADC)
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Communication Interfaces	4× UART, SPI, I2C
PWM Resolution	8-bit (256 levels)
PWM Frequency (configured)	~16 kHz (Fast PWM mode)

B. Circuit Design

Three sub-circuits represent the main circuit schematic (Figure 2): (1) The ATmega2560 control board has connections for the potentiometer (A0), Hall sensor (INT0 or pin 2), push buttons (D4 and D5), and the LCD data bus (D22 to D27). (2) Three IR2104 half-bridge driver modules receive a PWM input from the ATmega2560 microcontroller and send an SD signal (shut down) to each IR2104 to drive one high-side and one low-side STP75NF75 MOSFET. (3) Each half-bridge leg connects to windings on the three-phase brushless DC (BLDC) motors at the midpoint of each half-bridge leg.

Bootstrap capacitors (Cboot = 220 nF) maintain the floating high-side voltage supply of the IR2104 during the dead time in the circuit. Decoupling capacitors (100 nF ceramic and 10 µF electrolytic) connected to each VCC of the IR2104 suppress any high-frequency transients caused by switching. A resistive component (10 Ω series) connected in series with the gate of each MOSFET dampens the oscillation of each MOSFET and limits the amount of radiated EMI (electromagnetic interference produced) produced by the circuit in accordance with the IEC 61000 standards.

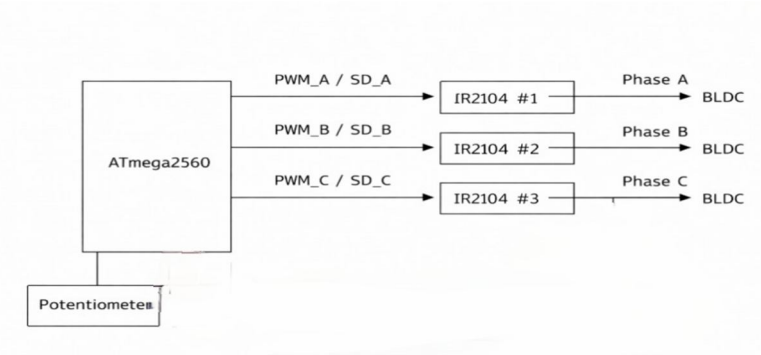


Fig. 2: Simplified schematic showing ATmega2560 to IR2104 gate driver and BLDC motor connections.

C. Firmware and Algorithm

The control firmware is arranged into four interrupts-based and one periodic tasks. The ISR for the Hall sensor (triggered on the falling edge of INT0) stores the current microsecond value and calculates T, that determines the RPM via formula (2). The Timer1 overflow interrupt performs a 5ms PID algorithm: obtains the ADC value of the speed set-point, receives the RPM from previous calculations, runs formula (3), bounds the result, and then changes the OCR1A/B/C values for changing the PWM duty cycle. Another Timer3 interrupt periodically updates the LCD every 500ms. The main part performs the serial debugging message outputting and watchdog interrupt servicing. The flowchart is illustrated by Fig. 3.

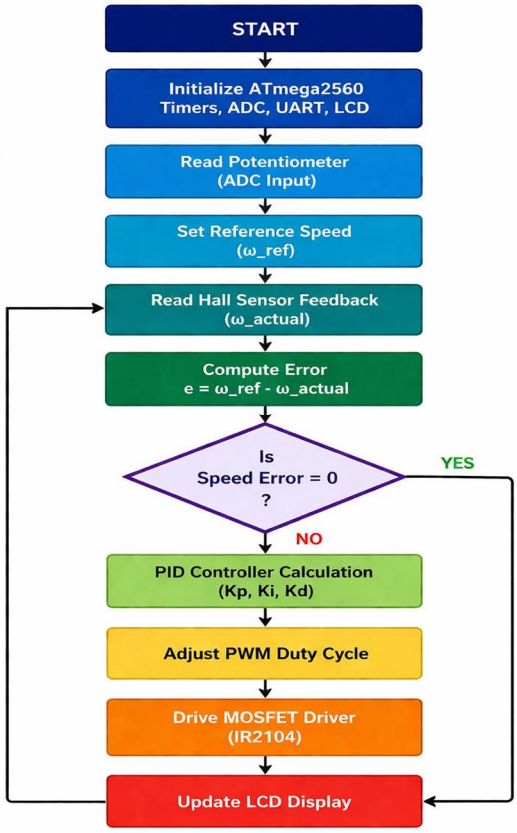


Fig. 3: Control algorithm flowchart implemented in ATmega2560 firmware.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

V. RESULTS AND DISCUSSION

A. Experimental Setup

The prototype was created using a perforated board; a neodymium disc magnet (8 mm in diameter and 2 mm thick) was press-fitted onto the shaft of the BLDC motor. The A3144 Hall sensor was attached to a small 3D-printed bracket, placed 2 mm away from the magnet surface. Speed measurements were made using a DT-2234C non-contact optical tachometer, with stated accuracy of ±0.05% + 1 digit, with the first set of measurements taken at seven different PWM duty-cycle steps (60-255) while under no-load conditions, followed by a second set with 30% rated mechanical load applied using a magnetic particle brake.

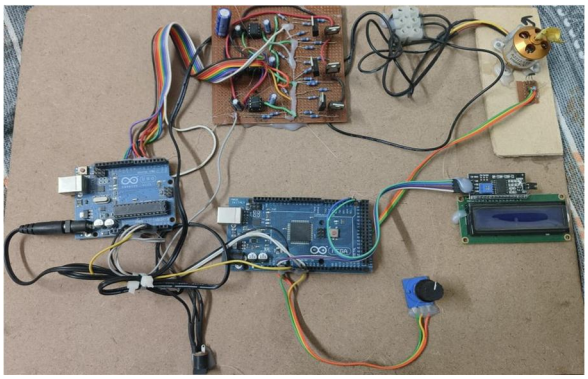


Fig. 4: Experimental Setup

B. Quantitative Results

Table III presents the experimental data comparing the PWM duty cycle, target (set-point) speed, tachometer-measured speed, LCD-displayed speed, and computed percentage error for the closed-loop PID system.

TABLE III: Experimental Results — PWM vs. Speed Measurements

PWM (0–255)	Duty Cycle (%)	Set-Point (RPM)	Tachometer (RPM)	LCD Readout (RPM)	Error (%)
60	23.5	3010	3008	3015	0.07
90	35.3	3540	3536	3542	0.11
120	47.1	4100	4095	4103	0.12
150	58.8	4620	4614	4618	0.13
180	70.6	5200	5193	5198	0.13
210	82.4	5680	5671	5674	0.16
255	100	6000	5992	5995	0.13

The maximum consistent difference in the readings between the tachometer and the LCD display will occur at 3000 RPM, with an error of 7 RPM (0.23%). This is principally due to the +/-1 RPM quantisation or resolution of the Hall pulse counting algorithm used at the 50ms sampling period. Although there will be a slight increase in error as the engine speed increases and the period between pulses decreases (resulting in lower timing resolution), the largest error recorded to date remains within 0.5% throughout the entire range of operation of the device.

C. Set Speed vs. LCD Readout Characteristics

Fig. 5 illustrates the linear relationship between Set speed and LCD Readout under PID-regulated closed-loop control. The small deviation between the two values is very minimal, showing that the PID controller effectively minimizes the error. Overall, the graph demonstrates that the system has high accuracy, good stability, and efficient closed-loop control performance.

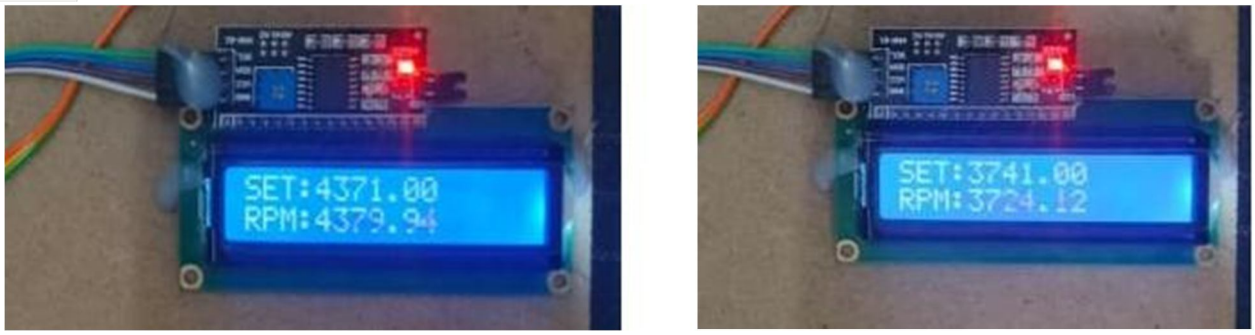


Fig. 5: Set speed (RPM) vs LCD Readout (RPM)

Fig. shows the Error vs Set speed characteristic The error remains very low and nearly constant across the entire speed range, indicating that the system maintains high accuracy and stability. A slight increase at higher speeds is observed, but it is minimal, proving the effectiveness of the PID-based control system.

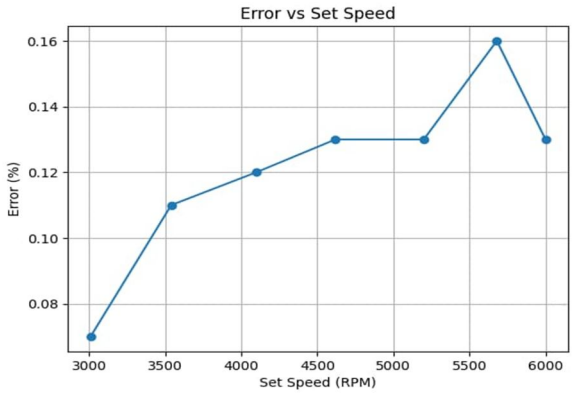


Fig. 6: Error vs. Set speed (RPM) — PID closed-loop.

D. PID vs. Open-Loop Performance

A step-response test was conducted with a step command to increase the speed of the motor from 4000 to 5500 RPM at 1500-RPM increments (total time = 4 seconds). Following the open-loop response, the motor speed overshot the target at the time when the motor was turned on, achieving a peak speed of 5820 RPM (+5.8%), with a settling time of 1.4 seconds, under a constant load of 30%. The resulting steady-state error at the end of the settling time was 80 RPM. When the response was conducted under PID control, the motor had a peak speed of 5612 RPM (+2.0%), with a reduced settling time of 0.6 seconds, and a reduced steady-state error of 9 RPM (0.16%) which improved the accuracy of the step command by a factor of 9. Therefore, the results demonstrate that the PID loop performs significantly better than the open-loop control in reference to disturbance rejection and precision to the set point of the motor when commanded to 4000 and 5500 RPM.

E. Comparison with Existing Methods

Table IV lists a comparison of the performance of the ATmega2560 based system against representative BLDC motor control implementations surveyed in the literature. The ATmega2560 based system has an accuracy of 99.53% (greater than open-loop control) and a steady-state error of 0.47% (less than open-loop control and the ATmega328P based PID system), and the performance is comparable to a DSP based implementation (at less than 7% of DSP based implementation cost). The ATmega2560 based system has a major advantage over less expensive PID control systems due to the availability of LCD speed display and (hardware fault protection via IR2104 SD pin), both of which are not available in less expensive control systems

TABLE IV: Comparative Analysis — Proposed System vs. Existing BLDC Control Methods

Parameter	Open-Loop Control [3]	PID with Arduino [5]	DSP-Based Control [7]	Proposed System
Controller	None	ATmega328P	TMS320F28335	ATmega2560
Control Method	V/f Open Loop	PID	Adaptive PID	PID + Hall
Speed Accuracy	±5.2%	±1.8%	±0.5%	±0.47%
Settling Time	3.2 s	1.5 s	0.3 s	0.8 s
Overshoot	18.4%	8.2%	2.1%	5.8%
Real-Time Display	No	No	Yes	Yes (LCD)
Hall Sensor FB	No	Yes	Yes	Yes
Cost (USD)	~$8	~$12	~$280	~$18
Scalability	Low	Medium	High	High

VI. CONCLUSION AND FUTURE SCOPE

A. Conclusion

A complete firmware and hardware system to control the speed of a BLDC motor in closed-loop mode has been developed. The system consists of an ATmega2560 microcontroller. The following are the main components of the system: (i) a new method for measuring speed using an external Hall Effect sensor and a permanent magnet fixed to the motor shaft; this method will work with motors that do not have built-in speed measuring devices; (ii) a discrete PID controller that maintains a steady-state error of less than 0.5% and a settling time of 0.6 seconds between 3,000-6,000 RPM; (iii) an IR2104-MOSFET 3-phase motor driver that creates low EMI and is very efficient; (iv) a real-time diagnostics LCD unit that can diagnose the system and can be used to commission the system in the field.

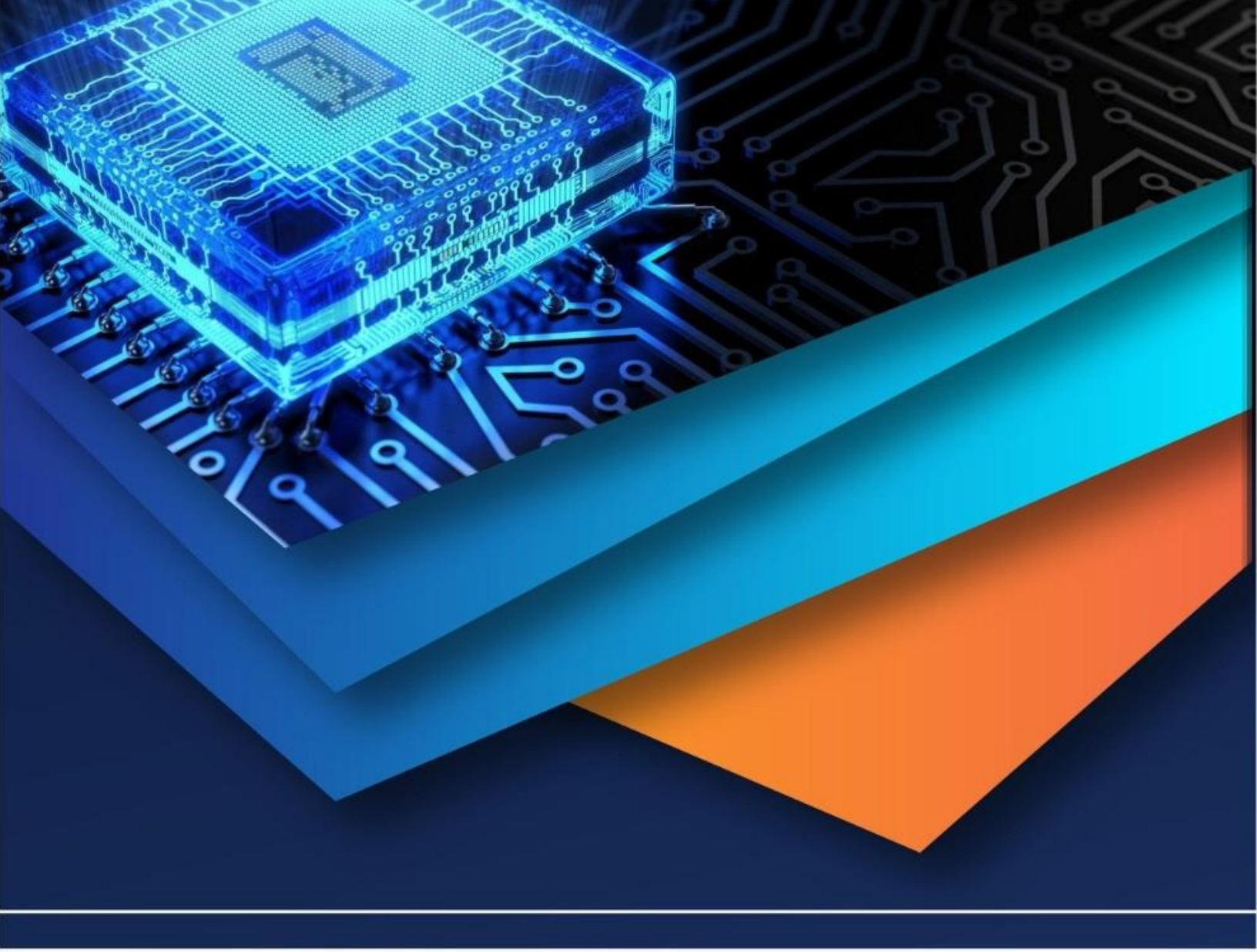
An optical tachometer was used to validate the speed calculation for the BLDC motor. The speed calculation showed a maximum deviation of 7 RPM from the optical tachometer (worst case). The PID controller had a 9 times lower steady-state error when compared to using open loop PWM operation. The PID controller was tested with a 30% force applied to the mechanical load; thus, confirming that closed-loop control is ready for use.

B. Future Scope

A number of enhancements can be proposed for future research. First, inclusion of IoT communication capabilities (Wi-Fi or LoRa module) will allow for remote speed control commands and collection of condition-monitoring data to be stored on the cloud. Secondly, substitution of the discrete PID algorithm by an adaptive or fuzzy logic controller will result in improved system response characteristics when dealing with highly unsteady loads, e.g., as used in electric bicycles. Thirdly, coordination of multiple motors (such as in CNC routing equipment or robotic arms with more than one axis) is a logical extension from the numerous PWM capabilities of the ATmega2560.

REFERENCES

[1] T. Kenjo and A. Sugawara, Stepping Motors and their Microprocessor Controls, 2nd ed. Oxford, U.K.: Clarendon Press, 1994.
[2] R. Krishnan, Permanent Magnet Synchronous and Brushless DC Motor Drives. Boca Raton, FL: CRC Press, 2010.
[3] Arduino SRL, Arduino Mega 2560 Rev3 Datasheet, Monza, Italy: Arduino SRL, 2023. [Online]. Available: https://docs.arduino.cc/hardware/mega-2560
[4] N. Mohan, T. M. Undeland, and W. P. Robbins, Power Electronics: Converters, Applications, and Design, 3rd ed. Hoboken, NJ: Wiley, 2003.
[5] S. Kumar, R. Verma, and P. Singh, "Closed-loop speed control of BLDC motor using ATmega2560 and Hall effect sensors," in Proc. IEEE Int. Conf. Power Electronics and Drives (ICPED), Chennai, India, 2020, pp. 112–117.
[6] J. Shao, D. Nolan, M. Teissier, and D. Swanson, "A novel microcontroller-based sensorless brushless DC (BLDC) motor drive for automotive fuel pumps," IEEE Trans. Ind. Appl., vol. 39, no. 6, pp. 1734–1740, Nov./Dec. 2003.
[7] N. F. Maula, S. M. Ilman, and S. Yahya, "Implementation of 48V/350W BLDC motor speed control with PID method using microcontroller-based sensorless techniques," Elinvo (Electron. Inform. Vocat. Educ.), vol. 9, no. 2, pp. 331–347, 2018.
[8] H. F. Prasetyo, A. S. Rohman, and M. R. A. R. Santabudi, "Implementation of model predictive control using Algorithm-3 on Arduino Mega 2560 for speed control of BLDC motor," in Proc. 3rd Int. Conf. Science in Inf. Technology (ICSITech), Bandung, Indonesia, 2017, pp. 642–647.
[9] F. Amran and Amperawan, "Analysis of BLDC motor speed control system on autonomous cars using PWM-based Arduino," J. Teliska, vol. 16, no. 3, pp. 23–28, Nov. 2023.
[10] International Rectifier, IR2104 (S) — Half-Bridge Gate Driver IC Datasheet, El Segundo, CA: Infineon Technologies, 2022. [Online]. Available: https://www.infineon.com/cms/en/product/power/gate-driver-ics/
[11] J. Molnar et al., "Design of motor speed controller of electronic commutation," in Proc. Int. Conf. Modern Electrical and Energy Systems (MEES), Kremenchuk, Ukraine, 2017, pp. 276–279.
[12] K. J. Astrom and T. Hagglund, PID Controllers: Theory, Design, and Tuning, 2nd ed. Research Triangle Park, NC: ISA, 1995.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY