Advanced and Sustainable Technologies
ASET
E-ISSN 2976-2294
Volume 3, No 2, December 2024 [133-143]

Simulation and Experimental Study of Arduino DC Motor Speed Control with PID Controller

S. I. Samsudin[1*], K. Osman[1], S. F. Sulaiman[1], S. I. M. Salim[1] and Nur Izzati Mohd Aliashak[1]

[1]Faculty of Technology and Engineering Electronics and Computer, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal, Melaka, 76100, Malaysia.

Received 28 September 2024, Revised 7 November 2024, Accepted 11 November 2024

ABSTRACT

DC motors have wide applications in industrial machinery, robotics and power systems. However, a DC motor without a controller may run in unstable speed and leading to the failure of the system operation. In fact, proportional-integral-derivative (PID) controllers are commonly used to control the speed of DC motors due to simple control structure but effective control performance. This project aims to regulate the speed of Arduino DC motor with PID controller in MATLAB Simulation platform. The project focuses on the development of Arduino DC motor, integration of the Arduino DC motor to MATLAB continuing with the validation of the PID control performances. It was proved that the speed of the DC motor was successfully controlled by the PID controller with more than 85% improvement of the mean error for both simulation and experimental works. The study demonstrates the effectiveness of PID in regulating motor speed and its potential for advanced control strategies of various DC motor applications, across educational and industrial areas.

Keywords: Arduino DC Motor, PID Controller.

1. INTRODUCTION

The DC motor is a device that converts electrical energy to mechanical energy. It has been used in various applications such as in transportation and robotic control systems [1]. The position and speed of the DC motor are both controllable components of the DC motor. The DC motor is implemented in most industry sectors due to its low cost and ease of use. Even though DC motors provide good torque at low speeds, their efficiency decreases at higher speeds, especially in brushed motors due to friction and heat generation in the brushes and commutator [2]. Brushed DC motors have limited speed due to the commutator and brushes. High speed leads to excessive wear on these components and leads to failures [3]. Designing a controller is proposed to solve the DC motor problem efficiently. In fact, a proportional-integral-derivative (PID) controller is the easiest and the most popular controller used in control applications [4].

The application of Arduino-based PID controllers for DC motor control has seen significant growth in both educational and industrial environments, providing an accessible, cost-effective solution for achieving precise motor control. Ma'arif et al. [4] developed PID to control angular speed using the Arduino Uno for data processing while the encoder sensor to calculate the angular speed with a motor driver. Peerzada et al. [5] illustrated the real-time stability of an Arduino-based PID setup with an L298N motor driver, where the PID controller ensures that the motor consistently meets speed setpoints under changing load conditions. Shaharudin et al. [6] introduced a modular Arduino PID control setup designed for speed and directional control in DC motors. This modularity is advantageous for robotics and manufacturing settings, where

*Corresponding author: sharatul@utem.edu.my

flexibility and quick adaptability to varying conditions are essential. The works on DC motors are also discussed in [7] where the design of a real-time experimental PID tracking controller using Arduino Mega 2560 for a permanent magnet DC motor under real disturbances was presented. A MATLAB/Simulink was used to design and simulate the controller and then implemented it on the Arduino board. The results showed that the proposed controller can effectively track the desired setpoint and reject disturbances.

With respect to the educational environment, numerous scholars [8-10] focused on integrating Arduino with PID control in educational labs to enhance the learning experience. Ma'arif [8] showed that using Arduino-based PID controllers allows students to experiment with feedback loops for motor speed and position, reinforcing concepts in control system stability and error correction. Ngwe and Tun [9] highlighted the effectiveness of using Arduino-based PID control in student projects, where the PID algorithm is combined with the accessibility of Arduino hardware. Latif et al. [10] created a laboratory trainer system that allows students to manipulate PID parameters in MATLAB and observe the motor response on Arduino in real-time. DC motor speed and position control methods with Arduino-based PID were extended studied in [11] that highlight the versatility of Arduino for various control for both academic and industrial applications.

Collectively, these studies discuss the flexibility and benefits of Arduino-based PID controllers for DC motor control. The advantages span across cost-efficiency, accessibility, and practical applicability, making these systems valuable for education, research, and industry. Although numerous studies exist on DC motor control, most focus primarily on simulation or experimental aspects. The proposed study explains in detail the development of the Arduino DC motor, the integration of the Arduino DC motor into the MATLAB/Simulink platform, and the simulation and experimental work of the PID control performances. This contributes to the extension of advanced control strategies of DC motor control that benefit education and industrial areas.

The paper is divided into 4 parts. Section 1 describes the introduction and background of the proposed work. A brief introduction to the DC motor and PID controller was also explained in this section. Section 2 discusses the material and methods of the study including the main hardware selection and the development of PID DC motor in simulation and experimental studies. The results and discussion are next explained in Section 3. Finally, the conclusion of the research work has been summarized in Section 4.

1.1 DC Motor System

A DC motor circuit is shown in Figure 1. Briefly, the modeling part of a DC motor consists of two parts: electrical and mechanical. The electrical component consists of input voltage, resistor, and inductor while the mechanical part focuses on rotor application.

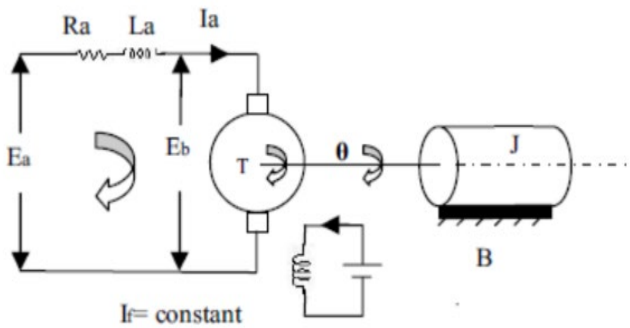


Figure 1: DC motor circuit.

By Kirchoff Law technique, the electrical equation of the DC motor can be described as Equations (1) and (2)

$$E_a = R_a I_a + L_a \frac{d_{ia}}{dt} + E_b \quad (1)$$

where $E_b = K_b \frac{d\theta}{dt}$ (2)

Here Ea is the voltage source, Ia is the armature current, Eb is the back emf voltage and Kb is a constant of back emf.

Meanwhile, the mechanical part of the DC motor can be explained as

$$J_m \frac{d^2\theta}{dt} + B_m \frac{d\theta}{dt} = T_m \quad (3)$$

where $T_m = K_t I_a$ (4)

Here, Jm is the rotor moment of inertia, Bm is the frictional coefficient, Tm is the motor torque while Kt is a torque constant.

Substituting Equation (2) into (1) and Equation (4) into (3) and solving the equations hence resulting in the general transfer function of the DC motor as in Equation (5). A detailed explanation of DC motor modeling can be referred to [1].

$$\frac{\theta(s)}{E_a(s)} = \frac{K_t}{J_m L_a s^3 + (J_m R_a + B_m L_a)s^2 + K_t K_b s} \quad (5)$$

1.2 PID Controller

Proportiona-integral-derivative (PID) controller is one of the best-known controllers used in industrial control due to its simple structure and effective performance in a wide range of operating conditions [12]. Besides, the PID provides good closed-loop response characteristics and simple tuning rules. The three constant parameters of the PID are called proportional (P), integral (I), and derivative (D) which are interpreted in terms of time. For the PID, the proportional gain, K_p reduces immediate errors, the integral gain, K_i corrects for accumulated offset while the derivative gain, K_d anticipates future changes, preventing overshoot and oscillations.

The PID control algorithm can be described as in Equation (6) where *e(t)* is the error signal.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\, dt + K_d \frac{de}{dt} \quad (6)$$

By taking the Laplace transformation, Equation (6) can be further expressed as in Equation (7).

$$G(s) = \frac{U(t)}{e(t)} = K_P + \frac{K_i}{s} + K_d s \quad (7)$$

Figure 2 shows the block diagram of a PID control to DC motor system. The output of the PID controller gives the input signal to the motor driver. The motor driver will send a signal to rotate the DC motor. The output speed of the DC motor is read by the encoder motor and feedback to the PID controller. The error value between the desired and measured speed will then be computed by the PID controller hence allowing the motor to quickly reach and maintain the desired speed despite disturbances, load changes, or inherent system nonlinearities, making the system more responsive and stable.

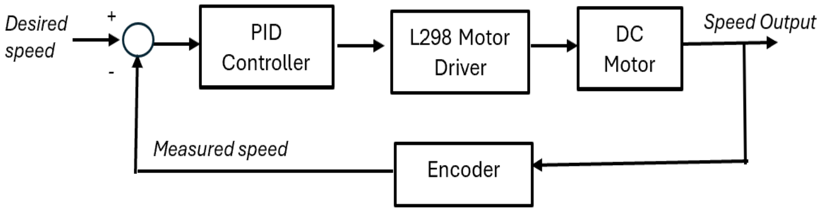


Figure 2: Block Diagram of DC Motor Controlled System.

2. MATERIAL AND METHODS

This section explains the main hardware applied and the development of the DC motor system.

2.1 Hardware

2.1.1 Microcontroller

In a DC motor control system, an Arduino microcontroller acts as the central controller, managing and adjusting the motor's speed and direction. It reads sensor inputs (encoder data for speed) and uses the PID control algorithm to calculate the required adjustments. The Arduino then sends signals to the motor driver circuit to adjust the motor's voltage and current, ensuring accurate speed control. Its simplicity, real-time processing, and flexibility make Arduino a popular choice for DC motor applications in both educational and prototyping applications (Figure 3).

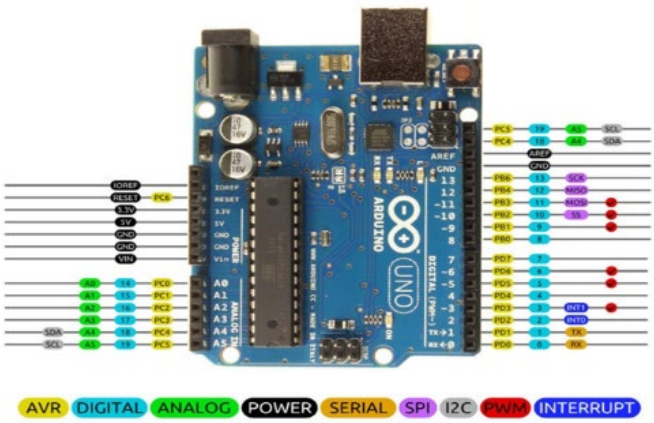


Figure 3: Arduino Board.

2.1.2 L298 Motor Driver

The L298 is used to control the DC motor (Figure 4). L298 is a high voltage, high current dual full-bridge motor driver that controls the rotational speed of a DC motor using the Pulse Width Modulation (PWM) technique. In PWM, the average value of the input voltage is modified by sending a series of ON-OFF pulses. When the duty cycle is high, more voltage is applied to the DC motor and, consequently, a faster motor speed will be resulted whereas when the duty cycle is low, less voltage is applied for a lower motor speed. This method provides smooth, efficient speed control without excessive power loss, making it effective for DC motor applications.

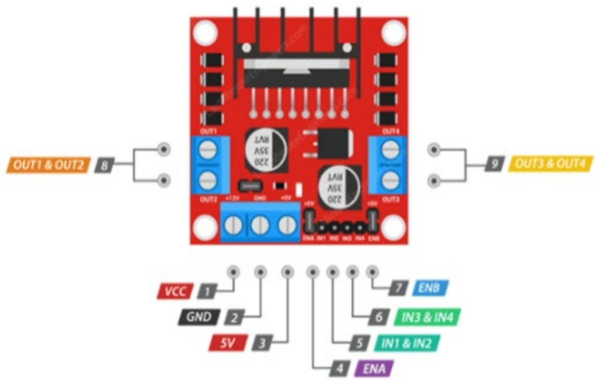


Figure 4: L298 Motor Driver.

2.1.3 12V DC Brushed DC Geared Motor

This project uses a brushed DC (BDC) geared motor (Figure 5). The motor offers speed regulation and can be powered by direct current power. A BDC motor provides stable and continuous current by utilizing rings to power a magnetic drive that conducts the armature of the motor.

Figure 5: 12VDC Brushed DC Geared Motor.

2.2 Development of Arduino DC Motor System

Figure 6 shows the Proteus schematic diagram of the DC motor. Here, IN1 and IN2 of the motor driver are connected to pins 6 and 7 of the Arduino board. The ENA pin is connected to the PWM pin 5 of the Arduino. In addition, the CHA and CHB pins of the DC motor are connected to pins 2 and 3 of the Arduino. The positive terminal (M+) and negative terminal (M-) of a DC motor are connected to the OUT1 and OUT2 output pins of the motor control. Meanwhile, 12V DC was supplied to the motor driver to provide power to the DC motor.

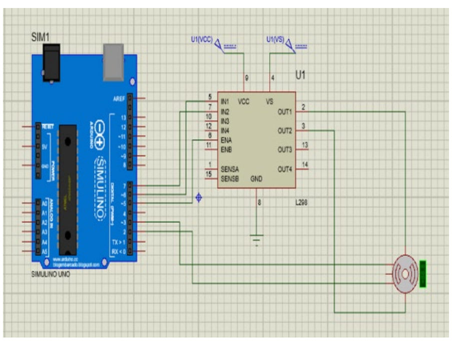
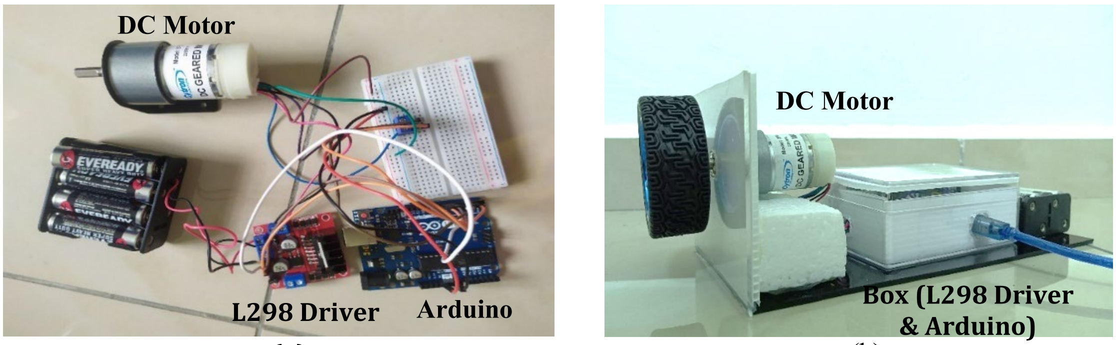


Figure 6: DC Motor Circuit.

Figure 7 shows the hardware implementation and prototype development of the Arduino DC motor system.



(a) (b)

Figure 7: (a) Hardware implementation and (b) Prototype development of the Arduino DC Motor.

In the experimental of DC motor speed control, the output pulses read by the encoder are converted into the revolutions per minute (rpm) according to Equation (8) as

$$Pulses\ to\ rpm = \frac{\Delta\, pulses}{time} \times \frac{1000ms}{1s} \times \frac{1rev}{IMP\ pulses} \times \frac{60s}{1min} \tag{8}$$

2.3 Integration of Arduino DC motor to MATLAB/ Simulink

The Arduino DC motor system was next integrated into MATLAB R2022b by the Simulink Support Package for Arduino Hardware. The DC motor system was first constructed in an open loop system as Figure 8. The system consists of a desired speed which is indicated by the RPM slider and DC motor system.

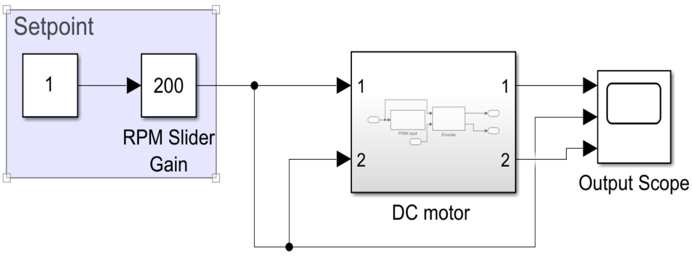


Figure 8: DC Motor.

To extend, the DC motor plant consists of a PWM block and encoder subsystem as shown in Figure 9. The PWM block will transmit the setpoint speed signal to an experimental DC motor system.

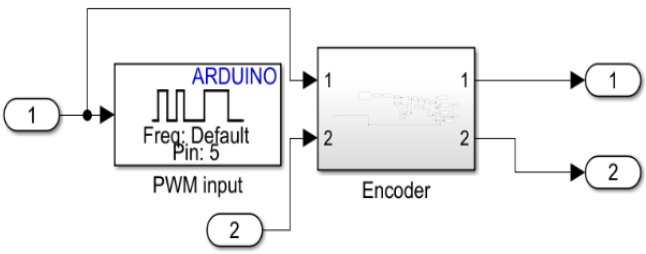


Figure 9: Extension of DC Motor (PWM and Encoder).

Equation 8, which is the calculation of pulses to rpm was used for Encoder implementation. The output pulses are multiplied by 1000 to give the result pulses per second. Then, it is divided by encoder per revolution, which is 80, and gives a result of revolution per second. Lastly, it is multiplied by 60 to obtain the output speed in rpm. The motor encoder subsystem is shown in Figure 10.

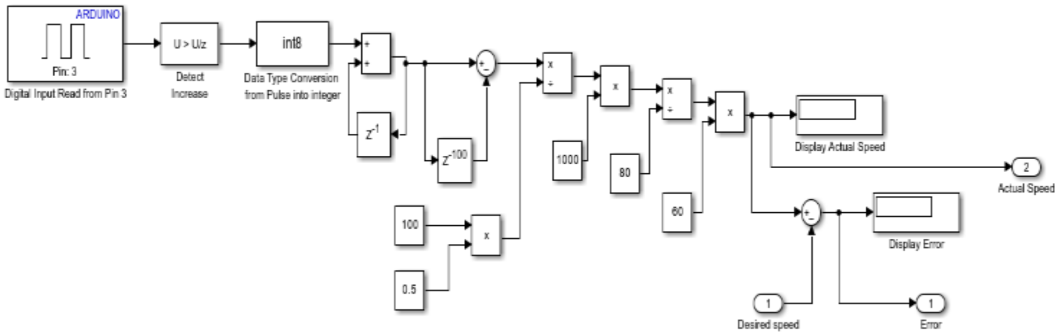


Figure 10: Encoder Subsystem.

Figure 9 was then extended to Figure 11 for a closed loop PID control system. Here, the speed measured by the encoder will be feedback to the sum block, where it is compared to the desired speed to produce an error. The PID controller then computes this error, allowing the motor to quickly regulate to the desired speed. This process keeps the DC motor system responsive and stable.

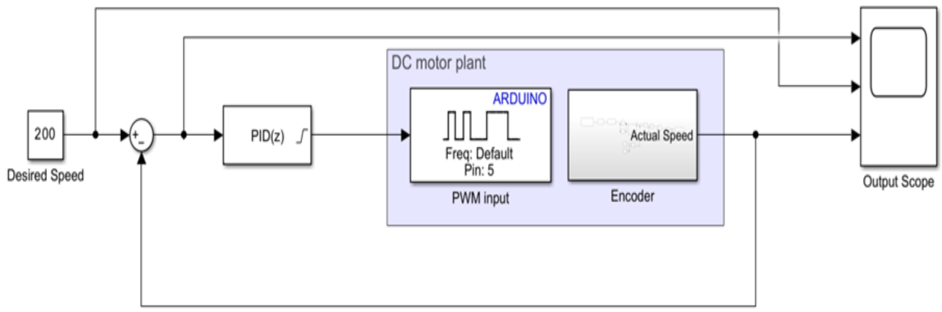


Figure 11: Closed-loop DC Motor System with PID Controller.

3. RESULTS AND DISCUSSION

3.1 Model Identification

In brief, model identification is a process to obtain a model of a dynamic system using input and output data. The model will then be used for the corresponding control application. Here, a System Identification Toolbox was used to obtain a transfer function of the DC motor system. There are a few general steps for modeling the DC motor using the system identification technique. This process includes collecting data, selecting an appropriate model structure, estimating parameters to fit the model to the data, and validating the model's accuracy. Equation 9 represents the transfer function identified by the MATLAB System Identification Toolbox with 85.07% best fit; which is used in the simulation of a DC motor control system.

$$G(z) = \frac{0.001048z+0.002346}{z^2-1.792z+0.795} \tag{9}$$

3.2 Simulation of DC Motor

Figure 12 shows the speed response of the simulation DC motor control. The desired speed was set to 200 rpm while the K_p, K_i, and K_d were set to 0.97, 27.23, and 0.001, respectively. It can be seen that the speed of the DC motor exceeds the desired speed of around 219 rpm, without a PID controller. On the other hand, the motor speed was successfully regulated to 200 rpm with the PID controller starting at 0.2 seconds.

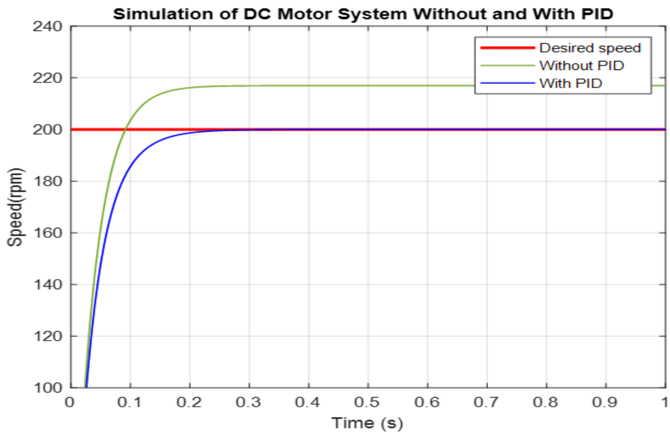


Figure 12: Control response of the simulation DC Motor.

Figure 13 illustrates the error responses of the simulated DC motor. Without a PID controller, the system exhibits a steady-state error of about 19 rpm but the steady-state error was significantly reduced to zero with the PID controller. With a PID controller, the speed measured by the encoder was feedback to the sum block while the error measured was then computed by the PID. The zero error response proves the effective control performance of the PID in eliminating the steady-state errors for a DC motor control system.

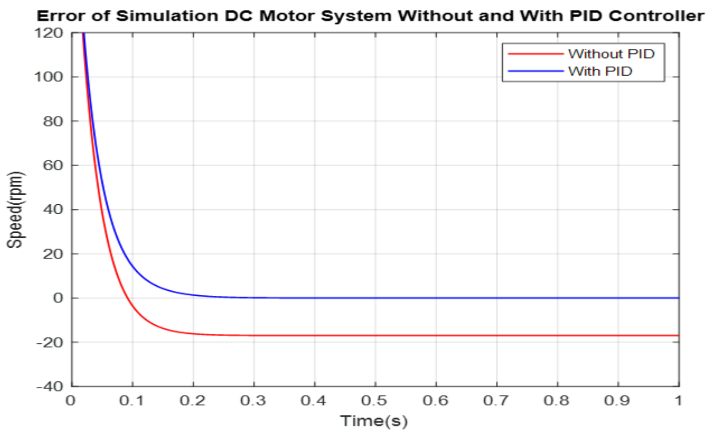


Figure 13: Error response of the simulation DC Motor.

3.3 Experimental DC Motor System

The experimental DC motor control was next explored. Initially, the desired speed value will be sent to pin 5 of the Arduino. The encoder will measure the actual speed of the DC motor and send the data to pin 3. On the contrary with the PID controller, the output of the PID is connected to the PWM pin 5 of the Arduino. The PID gain parameters; K_p, K_i, and K_d are maintained as previously applied in the simulation work.

Figure 14 shows the speed response of the experimental DC motor. It was observed that the speed varies between 210 to 230 rpm without the PID. However, the DC motor was successfully regularly tracking the desired speed with a PID, even with an initial overshoot.

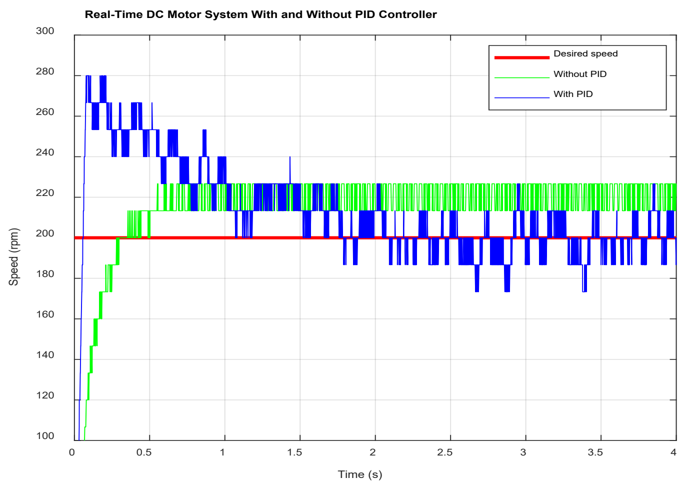


Figure 14: Control response of the experimental DC Motor.

To further extend, the variation in error performance by the experimental DC motor is presented in Figure 15. Without PID, deviations from the desired speed are declared as errors whenever the variation of the system output to the desired speed. In contrast, introducing a PID controller results in a distinct pattern. Over time, the error steadily diminishes, ultimately converging to zero. This reduction in error signifies the PID controller's effectiveness in compensating for the steady-state error, ensuring that the system attains and maintains at the desired speed.

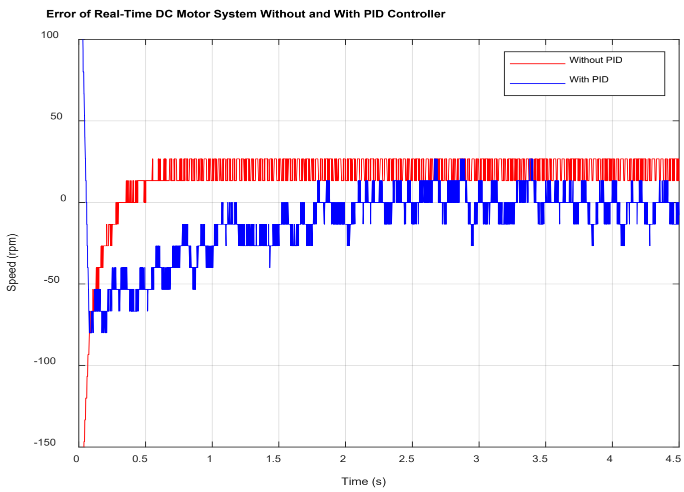


Figure 15: Error response of the experimental DC Motor.

3.4 Discussion on Simulation and Experimental Studies

Table 1 indicates the comparative mean error of the DC motor speed control with and without the PID controller for both simulation and experimental studies. It was demonstrated that the error was significantly reduced by the PID controller hence demonstrating the effective control performance of the PID controller in regulating the speed. With PID control, the motor maintains stability by constantly adjusting the error to match the desired DC speed and adapting effectively to disturbances for good control performance.

Table 1: Mean error for DC motor control.

	Simulation Study		Experiment Study	
	Without PID	With PID	Without PID	With PID
Mean error	15.904	0.993	22.044	2.312

To be extended, about 93.76% and 89.51% improvement of the mean error measured for both simulation and experimental study, respectively as illustrated in Figure 16.

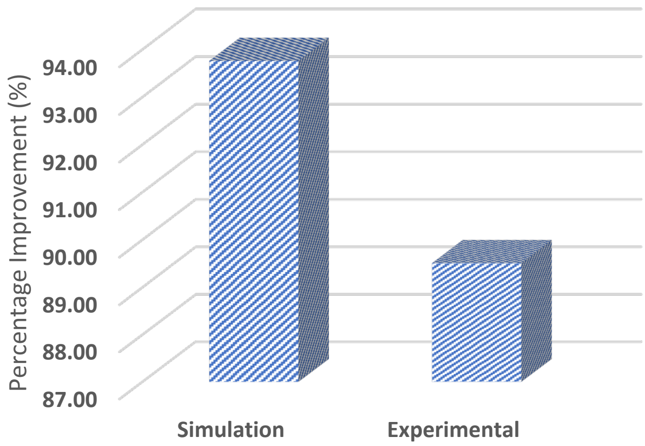


Figure 16: Comparative improvement of mean error by PID controller.

In both simulation and experimental studies, the PID controller plays a crucial role in minimizing error, which represents the difference between the desired and actual speed performance over time. In simulations, a well-tuned PID controller quickly reduces mean error by dynamically adjusting control inputs to follow the setpoint accurately, leading to smoother performance and reduced steady-state error. Experimental results validate these simulations, showing that the PID controller adapts to real-world disturbances, such as friction or load changes, significantly reducing mean error.

4. CONCLUSION

The project aims to investigate the proportional-integral-derivative (PID) control performances for speed DC motor systems. The project starts with model identification of the DC motor, continuing with the development of the Arduino DC motor and validation of the PID control performances in the MATLAB/Simulink platform. It was proved that the speed of the DC motor was successfully controlled by the PID controller with more than 85% improvement of the mean error for both simulation and experimental study. This proves the application of the PID controller substantially minimizes the error within the system, signifying its enhanced control capabilities. The work explores the initial study of Arduino-based DC motor speed control using a PID controller, advancing toward the implementation of an advanced controller for optimal speed regulation which offers valuable benefits for both educational and industrial applications.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge to Universiti Teknikal Malaysia Melaka (UTeM), Centre for Research and Innovation Management of Universiti Teknikal Malaysia Melaka (CRIM-UTeM), and Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer (FTKEK) UTeM.

REFERENCES

[1] Rahman, N. N. A., & Yahya, N. M. Simulation study of P, PI, and PID controllers for DC motor position. Journal of Engineering and Technology, vol 14, issue 1 (2023) pp. 159–173.
[2] Park, J., Kim, S., & Lee, J. Brushless DC Motor Design and Control: A Review. Journal Electrical Engineering & Technology, vol 16, issue 5 (2021) pp. 345-355.
[3] Kim, J., & Lee, H. Performance Challenges of Brushed DC Motors in High-Speed Applications. Journal of Power Electronics and Systems, vol 49, issue 2 (2021) pp. 95-102.
[4] Ma'arif, A., Iswanto, & Raharja, N. M. Control of DC motor using Proportional Integral Derivative (PID): Arduino hardware implementation. In Proceedings of the 2nd International Conference on Industrial Electrical and Electronics, (2020) pp. 74–78.
[5] Peerzada, P., Larik, W. H., & Mahar, A. A. DC motor speed control through Arduino and L298N motor driver using PID controller. International Journal of Electrical Engineering & Emerging Technology, vol 4, issue 2 (2021) pp. 21–24.
[6] Shaharudin, N., Hasan, M. Z., & Noor, S. M. Direct current (DC) motor speed and direction controller with Arduino. Journal of Physics: Conference Series, vol 2129, issue 1 (2021) p. 012035.
[7] Adel, Z., Hamou, A. A., & Abdellatif, S. Design of Real-time PID tracking controller using Arduino Mega 2560for a permanent magnet DC motor under real disturbances. In 2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM) (2018) pp. 1-5.
[8] Ma'arif, A. Control of DC motor using Arduino-based PID control in educational laboratories. Journal of Robotics and Control (JRC), vol 1, issue 6 (2021) pp. 459–465.
[9] Ngwe, T., & Tun, K. Arduino-based PID controllers for DC motor speed control: An educational perspective. Journal of Advanced Control Systems, vol 15, issue 3 (2020) pp. 45–53.
[10] Latif, A., Arfianto, A. Z., Widodo, H. A., Rahim, R., & Helmy, E. T. Development of DC motor speed control using PID based on Arduino and MATLAB for laboratory trainer. Jurnal Nasional Teknik Elektro, vol 3, issue 1 (2023) pp. 50–57.
[11] Divakar, A., Joseph, J., George, J. T., Prabhu, N. N., & Nabi, A. A study on DC motor operations and speed control using Arduino-based PID. International Journal of Advanced Research in Electrical, Electronics, and Instrumentation Engineering, vol 13, issue 2 (2024) pp. 112–118.
[12] Alwardat, M. Y., & Balabanov, P. V. Speed control of DC motor using PID controller based on MATLAB. Vestnik Tambovskogo Gosudarstvennogo Tehnicheskogo Universiteta, vol 27, issue 2 (2021) pp. 195-202.

Conflict of interest statement: The authors declare no conflict of interest.

Author contributions statement: Conceptualization, S. I. Samsudin & K. Osman; Methodology, S. I. Samsudin, Nur Izzati Mohd Aliashak & S. I. M. Salim; Result and Analysis, Nur Izzati Mohd Aliashak; Writing & Editing, S. I. Samsudin, S. F. Sulaiman & Nur Izzati Mohd Aliashak.