

## Simulation and Experimental Evaluation of DC Motor Control Strategies Using MATLAB and Arduino Mega



Marwan J. Hussein<sup>1\*</sup>, Omar Talib Khazraji<sup>2</sup>, Ahmed M. Almawla<sup>1</sup>

<sup>1</sup> Construction and Projects Department, Mustansiriyah University, Baghdad 14022, Iraq

<sup>2</sup> Electrical Engineering Department, College of Engineering, Mustansiriyah University, Baghdad 14022, Iraq

Corresponding Author Email: [marwan.aljanabi@uomustansiriyah.edu.iq](mailto:marwan.aljanabi@uomustansiriyah.edu.iq)

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.580107>

### ABSTRACT

**Received:** 11 November 2024

**Revised:** 8 January 2025

**Accepted:** 19 January 2025

**Available online:** 31 January 2025

#### Keywords:

*proportional-integral-derivative, fuzzy logic control, slide mode control, Arduino Mega, DC motor*

The paper presents a comprehensive analysis of three advanced control strategies: Proportional-Integral-Derivative (PID) controllers, Fuzzy Logic Controllers (FLC), and Sliding Mode Controllers (SMC) to achieve accurate speed control of a DC motor. The proposed study is conducted both theoretically and practically, utilizing MATLAB and AVR microcontrollers for real-time experiments. A modified SMC control law is introduced to enhance system performance, reduce the inherent chattering effect, and maintain robustness against parameter variations. The performance of each control strategy is evaluated based on key specifications, including system stability, response time, and adaptability to external disturbances. The findings highlight the strengths and limitations of each control approach and provide valuable insights for selecting the most suitable controller for specific applications. Additionally, the paper explores the integration of artificial intelligence techniques to optimize controller performance in dynamic and uncertain environments, contributing to the advancement of intelligent control systems.

## 1. INTRODUCTION

The increasing adoption of electric transportation systems is driven by the urgent need to reduce CO<sub>2</sub> emissions and improve air quality. Among various types of electric motors, DC motors are essential for applications ranging from transportation to robotics and industrial automation due to their simplicity, efficiency, and precise control. However, achieving optimal performance in these systems requires effective and reliable control strategies to address challenges such as non-linearity, disturbances, and parameter uncertainties. The development and implementation of robust control techniques—such as Proportional-Integral-Derivative (PID) control, Fuzzy Logic Control (FLC), and Sliding Mode Control (SMC)—are crucial to meet the dynamic demands of DC motor control [1].

Accurate speed regulation of DC motors is crucial because it directly affects the efficiency, reliability, and performance of various practical applications. In electric vehicles, precise speed control ensures smooth acceleration, energy efficiency, and an enhanced driving experience. In industrial automation, DC motors are fundamental to systems such as conveyors, robotic arms, and precision production processes, where even minor speed variations can compromise product quality and operational safety. In the healthcare sector, medical technologies like surgical robots and infusion pumps rely on DC motors for their accuracy and responsiveness, which is vital for ensuring patient safety. Furthermore, DC motors play a key role in renewable energy systems, where precise control

improves energy collection and conversion efficiency in devices like solar trackers and wind turbines. These diverse applications underscore the need for advanced control techniques that can meet the complex demands of modern systems. However, DC motors often operate in environments characterized by significant non-linearities, external disturbances, and unpredictable parameter fluctuations. Such conditions necessitate control solutions that are both precise and robust, capable of adapting to changing dynamics [2].

PID controllers are commonly used for DC motor control due to their simplicity and effectiveness in linear systems. These controllers maintain stable operation by adjusting proportional, integral, and derivative gains. However, tuning PID controllers can be challenging, especially in non-linear systems, potentially resulting in issues such as overshoot, oscillations, and slow response times [3].

FLC offers an alternative by simulating human reasoning through linguistic rules and fuzzy sets. Unlike PID controllers, FLC does not require precise mathematical models, making it suitable for non-linear and uncertain systems. FLC operates through fuzzification, rule-based inference, and defuzzification, allowing it to adapt to changing operating conditions [4]. While it effectively handles uncertainties, FLC's performance relies heavily on expert knowledge for rule design and may be sensitive to transient conditions.

SMC is a robust non-linear technique that directs system states along a predefined sliding surface to achieve desired dynamics. It operates in two phases: the reaching phase, where system states converge to the sliding surface, and the sliding

phase, which ensures robustness against disturbances [5]. SMC provides exceptional precision and resilience but can suffer from chattering, which limits its practical implementation.

Each approach—PID, FLC, and SMC—has advantages and limitations. PID controllers are popular due to their straightforwardness, yet their performance can decline under non-linear conditions and disturbances. In contrast, FLC and SMC provide viable solutions for non-linear systems, with FLC employing adaptive rule-based reasoning and SMC leveraging robustness and precision on sliding surfaces. However, FLC relies on expert knowledge to construct rules, while SMC requires careful design to mitigate its chattering issue and ensure practical applicability. This research compares the performance of these advanced control strategies—PID, FLC, and SMC—for precise DC motor speed control [6-8].

The study evaluates their adaptability, precision, and robustness using MATLAB Simulink simulations and hardware implementation on an Arduino Mega platform. This work bridges the gap between theoretical models and real-world applications by addressing challenges like tuning complexity and chattering. The findings offer valuable insights for developing cost-effective, high-performance control systems for embedded motor control solutions.

## 2. LITERATURE REVIEW

PID controllers have a rich history that dates to the early 20th century. James Clerk Maxwell developed the foundational concepts in 1868, analyzed governors for controlling steam engines, and introduced the idea of proportional control. Later, in 1922, Nicolas Minorsky formalized the PID framework while studying automatic ship steering systems [9]. His research emphasized the benefits of combining proportional, integral, and derivative actions to enhance system stability and eliminate steady-state errors.

The PID controller gained widespread adoption during the 1930s and 1940s, particularly in industrial automation. Its simplicity and adaptability made it the preferred choice for controlling processes such as temperature, pressure, and flow [10].

Fuzzy logic, introduced by Lotfi A. Zadeh in 1965, offers a framework for handling ambiguity and subjectivity in control systems. Unlike binary logic, which operates strictly between 0 and 1, fuzzy logic works on a continuum. Early applications, such as Mamdani's fuzzy logic controller (FLC) developed in 1974 for steam engine regulation, showcased its effectiveness in managing non-linear systems without the need for precise mathematical models. In 1985, the Takagi-Sugeno (T-S) model further enhanced computational efficiency by integrating linear equations into output fuzzy sets, making it applicable to more complex and dynamic systems [11-13].

SMC emerged in the 1950s as a robust technique for variable-structure systems. Early implementations faced challenges such as chattering and complexity. However, advancements have since introduced higher-order sliding modes, effectively addressing these issues. Modern SMC designs now improve dynamic performance, account for unstructured system dynamics, and handle external disturbances, making them suitable for real-world non-linear applications [14, 15].

## 3. MATHEMATICAL MODELLING

### 3.1 Modelling of the DC motor

Diagram of a DC motor, as seen in Figure 1, armature resistance ( $R$ ), armature inductance ( $L$ ), friction torque, inertia ( $J$ ), angular speed ( $\omega$ ), motor torque ( $T$ ), armature current ( $i$ ), DC power supply, and  $V_{Emf}$  electromotive force voltage.

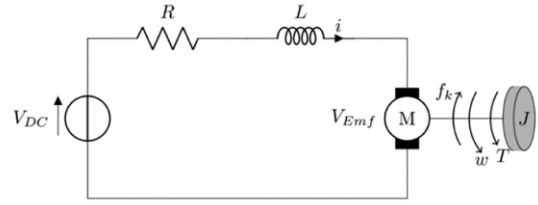


Figure 1. Diagram of the DC motor [16]

The equations derived from the analysis of Kirchhoff's voltage law are as follows:

$$Ri + L \frac{di}{dt} = V - K_e W \quad (1)$$

where,  $K_e$  is the constant for electromotive force.

Newton's second law of rotation can be expressed as follows:

$$J\dot{\omega} + b\omega = K_t i \quad (2)$$

where,  $K_t$  is the motor's torque constant, and  $b$  is the friction constant of the motor.

Upon executing the mathematical computations with Eqs. (1) and (2), we derived:

$$L \frac{di}{dt} = -Ri - K_e W + V \quad (3)$$

$$J\dot{\omega} = -b\omega + K_t i \quad (4)$$

Define the control signal variable and the state space variables as shown below:

$$X_1 = W \quad (5)$$

$$X_2 = i \quad (6)$$

$$u = V \quad (7)$$

Next, the state space variable is shown below:

$$\dot{X}_1 = \dot{W} = \frac{d\theta}{dt} = -\frac{b}{J}W + \frac{K_t}{J}i \quad (8)$$

$$\dot{X}_2 = \dot{i} = \frac{di}{dt} = -\frac{K_e}{L}W - \frac{R}{L}i + \frac{1}{L}u \quad (9)$$

Next, use the substitution method to find:

$$\dot{X}_1 = -\frac{b}{J}X_1 + \frac{K_t}{J}X_2 \quad (10)$$

$$\dot{X}_2 = -\frac{K_e}{L}X_1 - \frac{R}{L}X_2 + \frac{1}{L}u \quad (11)$$

The state-space model is as follows:

$$\dot{X} = AX + Bu \quad (12)$$

$$y = CX \quad (13)$$

where,

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, A = \begin{bmatrix} -\frac{b}{J} & \frac{K_t}{J} \\ -\frac{K_e}{L} & -\frac{R}{L} \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}, C = [1 \ 0]$$

Table 1 presents the DC motor's parameter values used to build the Simulink and design the controller.

**Table 1.** Value of the DC motor [17]

Parameters	Value
R	0.5 $\Omega$
L	3.3 mH
J	0.0005 kg.m <sup>2</sup>
$K_b$	0.0027 V.s/rad
$K_t$	0.018 N.m/A
$K_e$	0.06 V/rad/s

The state space model, defined in Eqs. (12) and (13), is created by applying the parameters listed in Table 1, as seen below:

$$\dot{x} = \begin{bmatrix} -5.4 & 36 \\ -0.0181 & -0.1515 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0.3030 \end{bmatrix} u \quad (14)$$

$$y = [1 \ 0] \quad (15)$$

Then, convert the state model to a transfer function using MATLAB.

$$\omega(s) = \frac{10.91}{s^2 + 5.552s + 1.47} V(s) \quad (16)$$

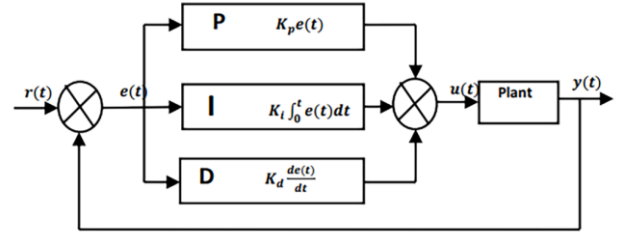
### 3.2 PID controller model

Due to its simplicity and efficacy, the proportional-integral-derivative (PID) controller is a regularly used feedback control mechanism for managing the speed of DC motors [18]. It operates by continuously calculating the error between the desired speed and the motor's actual speed. The controller then applies a correction based on three components: proportional, integral, and derivative. The output of the controller adjusts the motor's speed accordingly [19, 20].

where,  $e(t)$  is the error between the desired and actual speed at the time,  $K_p$  is the proportional gain, which determines the response to the current error,  $K_i$  is the integral gain, which addresses accumulated past errors, and  $K_d$  is the derivative gain, which predicts future errors based on the rate of change of the error.

Popular tuning techniques include manual tuning, Ziegler-Nichols, and Cohen-Coon. During manual tuning, the proportional gain ( $K_p$ ) is first increased until the system starts oscillation. Subsequently, the integral gain ( $K_i$ ) is calibrated to eliminate any steady-state inaccuracy, while the derivative gain ( $K_d$ ) is adjusted to reduce oscillations [21]. The Ziegler-Nichols method provides an empirical technique that uses the system's ultimate gain and oscillation period to recommend starting parameter settings. Upon establishing these values,

precise modifications are implemented to equilibrate trade-offs across performance indicators, including rise time, overshoot, and settling time. Figure 2 shows the PID configuration model.

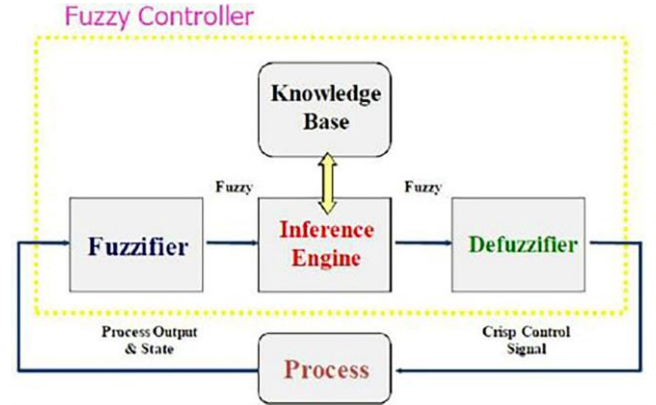


**Figure 2.** The configuring model of a PID controller

### 3.3 Fuzzy logic controller model

FLC consists of four main components: the fuzzifier, knowledge base, inference mechanism, and defuzzifier (refer to Figure 3).

- **Fuzzifier:** This component converts classical inputs into fuzzy sets using membership functions, which can be triangular, trapezoidal, or Gaussian, depending on the specific application.
- **Knowledge Base:** The knowledge base contains IF-THEN rules that define the control logic for the system.
- **Inference Mechanism:** This mechanism processes the rules from the knowledge base to generate a fuzzy output.
- **Defuzzifier:** Finally, the defuzzifier converts the fuzzy output into a crisp control signal.



**Figure 3.** Schematic of fuzzy logic controller

The Mamdani method is widely credited for its intuitive design and remains a popular choice in literature and applications. However, other methods, such as Sugeno, offer computational advantages and may be preferred in certain scenarios [22-25].

$$U(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt} \quad (17)$$

This research adopts the Mamdani method to develop an FLC for regulating the speed of a DC motor. The Mamdani method is noted for its intuitive and interpretable approach, leveraging expert knowledge to build a rule-based system. Fuzzy rules are formulated as "if-then" statements, connecting input variables such as Error (E) and Sum of Error ( $\Sigma E$ ) to corresponding control actions [26]. Inputs are fuzzified using

Gaussian membership functions, which enable smooth transitions and improve sensitivity to changes. The output employs triangular membership functions to simplify the defuzzification process and enhance the system's responsiveness.

The FLC is designed with four variables: two inputs and two outputs, enhancing precision and stability in controlling the motor's nonlinear dynamics. This configuration minimizes oscillations and ensures responsive, reliable control [27]. A total of 25 fuzzy rules defines the system's operational logic, addressing various errors and dynamic conditions. The inference mechanism utilizes these rules to generate suitable control actions, while the defuzzification process produces precise output signals optimized for real-time applications.

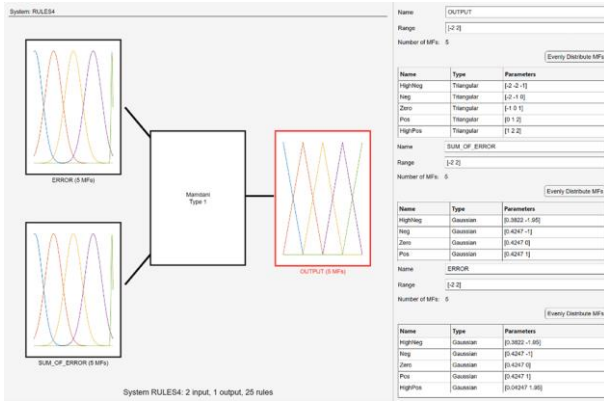


Figure 4. Membership function of error

Table 2. FLC rules conditions

Rule	Condition (If)	Output	Weight
1	E: HighNeg, SE: HighNeg	HighNeg	1
2	E: HighNeg, SE: Neg	HighNeg	1
3	E: HighNeg, SE: Zero	Neg	1
4	E: HighNeg, SE: Pos	Neg	1
5	E: HighNeg, SE: HighPos	Zero	1
6	E: Neg, SE: HighNeg	HighNeg	1
7	E: Neg, SE: Neg	Neg	1
8	E: Neg, SE: Zero	Neg	1
9	E: Neg, SE: Pos	Zero	1
10	E: Neg, SE: HighPos	Pos	1
11	E: Zero, SE: HighNeg	Neg	1
12	E: Zero, SE: Neg	Neg	1
13	E: Zero, SE: Zero	Zero	1
14	E: Zero, SE: Pos	Pos	1
15	E: Zero, SE: HighPos	Pos	1
16	E: Pos, SE: HighNeg	Zero	1
17	E: Pos, SE: Neg	Zero	1
18	E: Pos, SE: Zero	Pos	1
19	E: Pos, SE: Pos	Pos	1
20	E: Pos, SE: HighPos	HighPos	1
21	E: HighPos, SE: HighNeg	Zero	1
22	E: HighPos, SE: Neg	Pos	1
23	E: HighPos, SE: Zero	Pos	1
24	E: HighPos, SE: Pos	HighPos	1
25	E: HighPos, SE: HighPos	HighPos	1

Note: E refers to Error, and SE refers to Sum of Errors

Figure 4 and Table 2 illustrate the membership functions and the interactions of fuzzy rules, respectively, emphasizing the system's capability to adapt effectively to changing operating conditions. This approach demonstrates how the FLC balances simplicity and robustness, making it well-suited for applications demanding high precision and adaptability.

Figure 5 demonstrates how fuzzy logic rules influence the output signal, showcasing the impact of different combinations of Error (E) and Cumulative Error ( $\Sigma$  Error) on the control response. These rules define the relationships between input and output variables, shaping the system's dynamic behavior. Figure 6 provides a more detailed perspective through 3D surface plots, illustrating how changes in both error terms result in smooth, continuous transitions in the control signal. This visual representation emphasizes the adaptability of the fuzzy logic controller in managing system dynamics and ensuring stable performance under varying conditions.

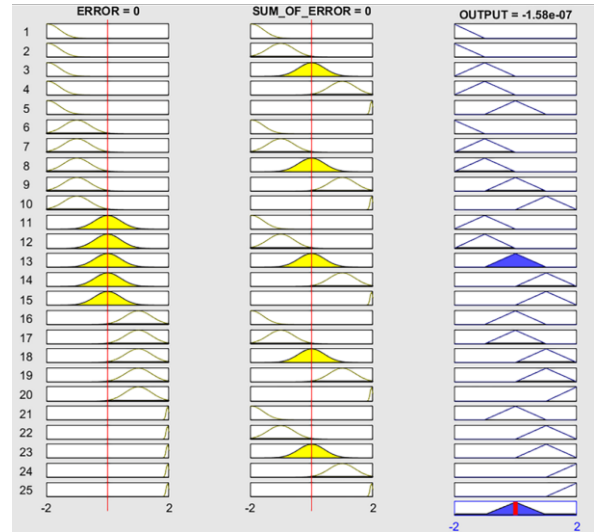


Figure 5. FLC rules membership function

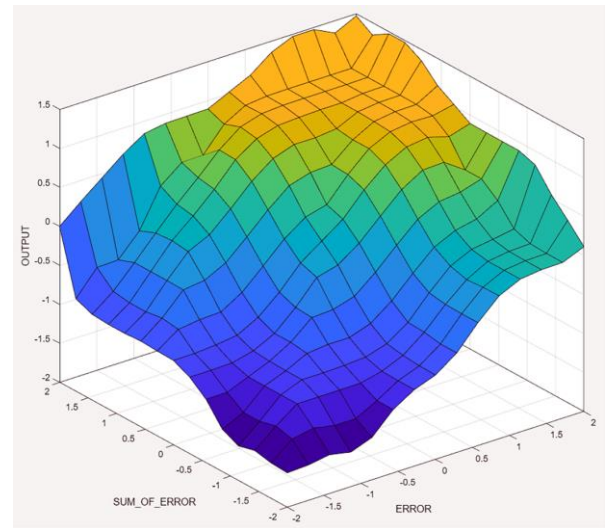


Figure 6. FLC controls the surface

### 3.3 Slide mode controller model

The controller for the motor must be written in state space; thus, we can write Eq. (16) in the time domain as shown below:

$$\ddot{\omega} + 5.552\dot{\omega} + 1.47\omega = 10.91V \quad (18)$$

Define the state space variables as:

$$X_1 = \omega \quad (19)$$



$$X_2 = \dot{\omega} \quad (20)$$

$$u = V \quad (21)$$

Next, we can express the system in its controllable canonical form as follows:

$$\dot{X}_1 = X_2 \quad (22)$$

$$\dot{X}_2 = -5.552X_2 - 1.47X_1 + 10.91V \quad (23)$$

Next, we will proceed with the design of the sliding mode function as follows:

$$S = Ce + \dot{e} \quad (24)$$

The variable ( $e$ ) represents the tracking error value, while the value of the variable ( $C$ ) must adhere to the Hurwitz rule ( $C > 0$ ) [28]. The values of the error and its derivative are as follows:

$$e = \omega_d - \omega \quad (25)$$

$$\dot{e} = \dot{\omega}_d - \dot{\omega} \quad (26)$$

$$\ddot{e} = \ddot{\omega}_d - \ddot{\omega} \quad (27)$$

The variable ( $\omega_d$ ) represents the reference signal, while ( $\omega$ ) represents the actual angular velocity. Now, let's define the Lyapunov function as follows:

$$V = \frac{1}{2}S^2 \quad (28)$$

To guarantee stability, the Lyapunov function in Eq. (28) must meet the condition ( $\dot{V} < 0$ ) [29].

$$S\dot{S} < 0 \quad (29)$$

The equation for the sliding mode function is:

$$\dot{S} = C\dot{e} + \ddot{e} = C\dot{e} + \ddot{\omega}_d - \ddot{\omega} \quad (30)$$

Therefore, the derivative of the Lyapunov function  $\dot{V}$  is:

$$S\dot{S} = S(C\dot{e} + \ddot{\omega}_d + 5.552\dot{\omega} + 1.47\omega - 10.91U) \quad (31)$$

To satisfy the condition ( $S\dot{S} < 0$ ) the sliding mode controller is designed as follows:

$$U = \frac{1}{10.91}(1.47\omega + 5.552\dot{\omega} + \ddot{\omega}_d + C\dot{e} + Ksgn(s)) \quad (32)$$

where,

$$sgn(s) = \begin{cases} 1. & s > 0 \\ 0. & s = 0 \\ -1. & s < 0 \end{cases}$$

By substituting Eq. (32) to Eq. (28), then we get:

$$S\dot{S} = -Ksgn(s) < 0 \quad (33)$$

## 4. RESULT AND DISCUSSION

The chapter will be organized into two parts: the first will address numerical simulation with MATLAB Simulink (2023b), and the second will concentrate on hardware setup utilizing Arduino Mega.

### 4.1 Overall MATLAB simulation PID, Fuzzy, and SMC

Figures 7-9 present detailed simulations of the system developed in MATLAB Simulink. Each figure highlights specific aspects of the comparative analysis of various control strategies for DC motor applications, offering a comprehensive overview of their performance under different conditions. The simulation evaluates the dynamic performance of PID, FLC, and SMC controllers based on key performance metrics such as response time, stability, and accuracy. By systematically analyzing these criteria, the study determines the most effective control strategy for enhancing motor efficiency and precision. This comprehensive evaluation provides valuable insights into how each controller adapts to the specific requirements of DC motor systems, facilitating the selection of optimal control solutions for practical applications.

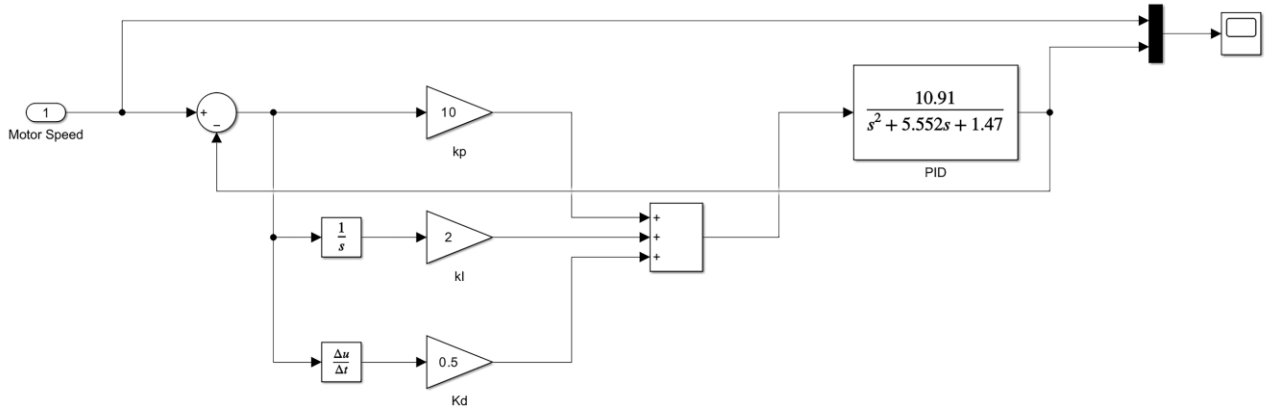


Figure 7. PID simulation model

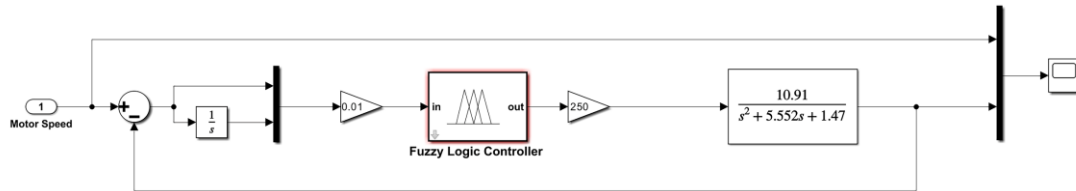


Figure 8. FLC simulation model

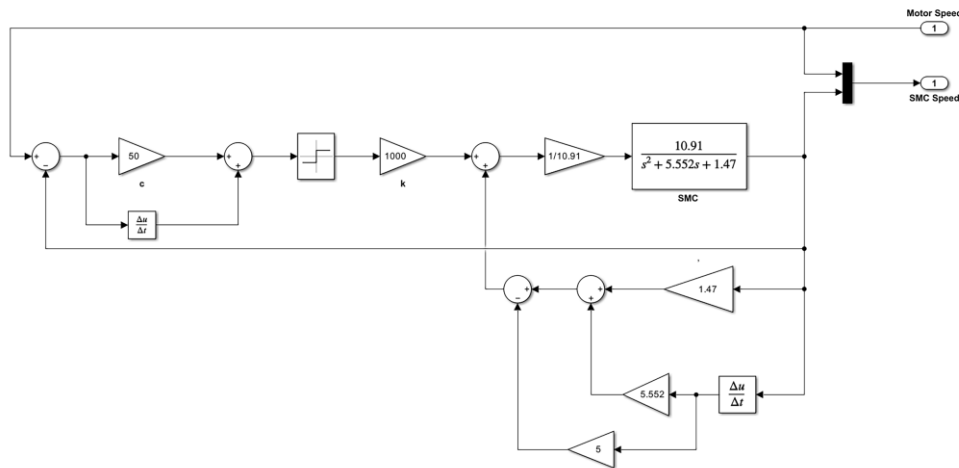


Figure 9. SMC simulation model

The results in Figure 10 show that the SMC outperforms the FLC and PID controllers in achieving an optimal system response. The SMC delivers moderate rise time, enabling the system to reach the desired setpoint quickly. Additionally, it minimizes the settling time, allowing the system to stabilize more rapidly. Notably, the SMC maintains zero overshoot, exhibiting precise control and a smooth transition to steady-state operation.

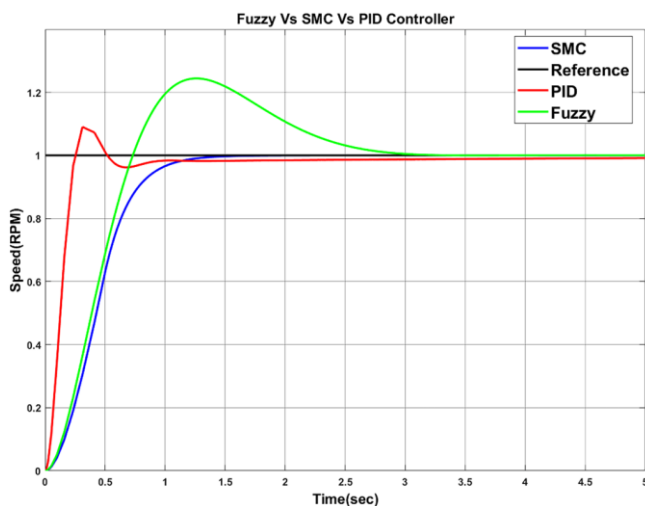


Figure 10. Overall system response

The simulation results present a comprehensive comparative analysis of the performance of PID, FLC, and SMC in effectively controlling the speed of a DC motor, as shown in Table 3.

Table 3. Simulation metrics of PID, FLC, and SMC

Controller	Rise Time	Settling Time	Overshoot (%)
PID	0.216	0.95	9
FLC	0.730	2.99	24.3
SMC	0.780	1.50	0

The PID controller achieved a rapid rise time of 0.216 seconds and a settling time of (0.95) seconds, with a moderate overshoot of (9%). These results highlight PID's quick response capability, although it tends to produce some overshoot due to its reliance on fixed gain parameters that are sensitive to system dynamics.

In contrast, the FLC exhibited a slower rise time of (0.73) seconds and a longer settling time of (2.99) seconds, along with a higher overshoot of (24.3%). This performance can be attributed to FLC's adaptive and heuristic nature, which emphasizes robustness and flexibility over speed. As a result, it responds less aggressively to changes in setpoints. Despite this slower response, FLC demonstrates effectiveness in handling system uncertainties and non-linearities that may challenge traditional controllers like PID.

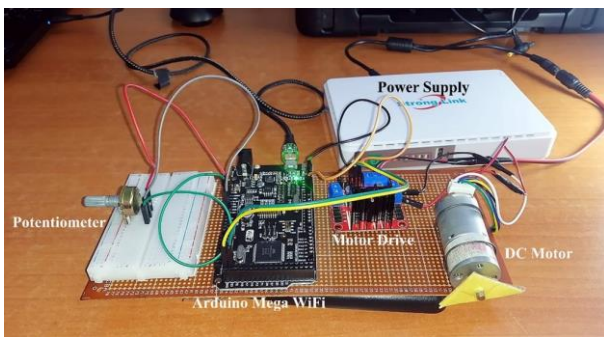
The SMC recorded a rise time of (0.780) seconds, a settling time of (1.50) seconds, and no overshoot, showcasing its precision and robustness. The absence of overshoot underscores SMC's ability to maintain stability and accuracy under non-linear conditions. However, the simulation also revealed a chattering effect inherent to SMC due to the control law's discontinuous nature, such as its high-frequency switching behavior. While this chattering does not affect rise and settling times, it poses a challenge for real-world implementation, potentially causing wear and inefficiencies.

## 4.2 The design and implementation of Arduino hardware

The hardware setup for the DC motor control experiments, outlined in Table 4, is designed to facilitate the implementation and comparison of PID, FLC, and SMC strategies. At the heart of the system is a (JGA25-371) DC motor equipped with an encoder sensor, which delivers precise speed and position measurements for feedback control. The control algorithms run on an Arduino Mega WiFi, leveraging its enhanced memory and computational capabilities, making it suitable for advanced control applications. An (L298) motor driver connects the microcontroller to the DC motor, ensuring efficient power delivery and reliable actuation. Real-time current monitoring is achieved using an (INA260) current sensor, which ensures robust system protection and feedback for current control. A stable (12V) DC power supply powers the entire system, while a (10 kΩ) potentiometer serves as a manual input device, allowing for on-the-fly adjustments during testing. Figure 11 depicts the experimental setup.

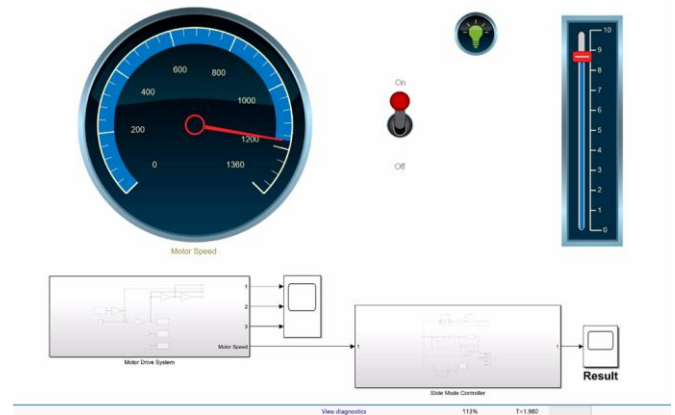
**Table 4.** Summary of system design elements

Elements	Specifications
DC motor with an encoder sensor	JGA25-371
Microcontroller	Arduino Mega
Motor drive	L298
Current sensor	INA260
DC power supply	12V
Potentiometer	10 KΩ

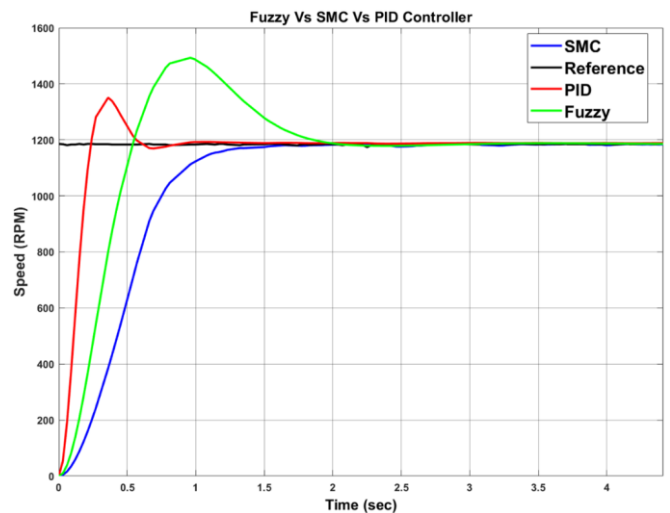


**Figure 11.** Overall system hardware setup

The MATLAB Simulink interface designed for this practical experiment provides an interactive environment for testing and analyzing various DC motor control strategies, including PID, FLC, and SMC. The Motor Speed gauge, prominently displayed on the left, enables accurate monitoring of the motor's speed in RPM, with a maximum speed of (1360 RPM) and an operating voltage of (12V). This feature provides immediate feedback on system performance. A toggle switch facilitates manual control, allowing users to activate or deactivate the motor drive system, simulating different operating conditions. Additionally, an adjustable slider on the right allows for the control of the reference speed or input variable, facilitating the testing of the control system's responsiveness to changes in setpoint values. The indicator light enhances the interface by visually representing the system's active state. Using MATLAB, this setup simulates advanced real-time prediction and data control, allowing for a detailed and realistic evaluation of control methodologies in a practical setting. As illustrated in Figure 12, this interactive Simulink interface effectively bridges simulation and practical testing.



**Figure 12.** Real-time hardware MATLAB simulation model



**Figure 13.** Overall system response

**Table 5.** Practical metrics of PID, FLC, and SMC

Controller	Rise Time	Settling Time	Overshoot (%)
PID	0.236	0.570	14.5
FLC	0.540	2.000	27.08
SMC	1.430	1.420	0

Figure 13 and Table 5 present the results from the experimental evaluation of PID, FLC, and SMC controllers. The figure shows the response curves for each controller, highlighting their dynamic behavior.

Meanwhile, the table provides a detailed numerical comparison of their performance metrics, which include rise time, settling time, and overshoot percentage.

The comparative analysis of PID, FLC, and SMC controllers identifies both expected trends and discrepancies between theoretical simulations and practical experiments, emphasizing the challenges of real-world implementation such as:

- **Modeling Accuracy:** Simulations depend on mathematical models that may not perfectly represent a system's real-world behavior. Any simplifications or assumptions made during the modeling process can result in discrepancies.

- **Parameter Variations:** The parameters used in simulations (such as resistance, inductance, and friction) may differ from those in the actual system due to manufacturing tolerances, wear and tear, or varying environmental conditions.

- **Nonlinearities:** DC motors display nonlinear behaviors, including friction, saturation, and variations in load. Simulations may not fully account for these nonlinearities, leading to differences in performance.

- **Sensor Noise and Calibration:** Sensors used to measure speed and other variables can introduce noise and errors in experiments. Furthermore, issues with sensor calibration can affect the accuracy of the measurements.

- **Control Algorithm Implementation:** The implementation of control algorithms in software and hardware can vary, resulting in differences in performance. Delays, discretization errors, and computational limitations may also impact the results.

- **Disturbances and Load Changes:** Real-world systems are frequently subjected to disturbances and changes in load that are difficult to predict and simulate accurately. These factors can cause the system to behave differently than expected.

PID Controller simulations predicted a rise time of (0.216) seconds, a settling time of (0.95) seconds, and a (9%) overshoot. In practice, the rise time increased to (0.236) seconds, the settling time improved to (0.57) seconds, and the overshoot rose to (14.5%). These differences likely stem from unmodeled dynamics, such as friction and sensor noise. The improved settling time suggests enhanced damping due to real-world non-linearities, while the increased overshoot indicates the presence of practical disturbances.

FLC simulation estimates indicated a slower response, with a rise time of (0.73) seconds, a settling time of (2.99) seconds, and an overshoot of (24.3%). In practice, both the rise time and settling time improved significantly to (0.54) seconds and (2.00) seconds, respectively, due to the FLC's robustness in handling system uncertainties. However, the overshoot increased to (27.08%), highlighting the difficulties in managing unmodeled disturbances in hardware environments. Unmodeled effects, such as sensor delays or environmental vibrations, can amplify transient behavior, resulting in greater overshoot.

SMC Simulations projected a rise time of (0.78) seconds, a settling time of (1.50) seconds, and no overshoot. Practical results showed a longer rise time of (1.43) seconds but a slightly quicker settling time of (1.42) seconds, with the overshoot remaining at (0%). These outcomes illustrate SMC's resilience to disturbances and highlight practical challenges related to actuator dynamics and sensor noise, which can produce chattering effects. For example, high-frequency noise in the sensors can amplify chattering effects in the control signals, which may affect the system's smoothness and rise time.

The discrepancies between simulation and practical results stem from real-world factors such as friction, sensor delays, and environmental disturbances, which are often not included in idealized models. Friction in mechanical joints introduces nonlinear damping effects that impact both transient and steady-state behaviors. Additionally, sensor noise, including quantization and thermal noise, distorts feedback signals and leads to variations in controller performance. The practical improvements in settling times for PID and FLC are attributed to additional damping from system inertia. In contrast, the increased overshoot and delays reflect the influence of unmodeled dynamics and hardware constraints. These findings underscore the importance of bridging the gap between simulation and practical implementation to achieve robust performance in real-world control systems.

Figure 14 shows the initial response curve of the SMC,

which reveals a significant chattering effect due to the high-frequency switching characteristic of this control technique. To mitigate this issue and improve system performance, the control law was adjusted using the following equation:

$$U = \frac{1}{10.91} (1.47\omega + 5.552\dot{\omega} + \ddot{\omega}_d + C\dot{e} + K \frac{s}{|s| + \delta}) \quad (34)$$

The introduction of the factor ( $\delta$ ) in the denominator smooths the switching behavior by reducing the gain near the sliding surface. This modification significantly reduces the chattering effect, as evidenced by the improved response curve shown in Figure 15. However, despite this enhancement, the modified curve still exhibits minor oscillations. These can be attributed to residual switching dynamics and unmodeled system nonlinearities. While the adjustments have notably reduced the severity of chattering, further optimization of parameters such as ( $\delta$ ) and ( $K$ ) may be necessary to achieve an even smoother response. These results demonstrate the balance between maintaining the robustness of SMC and enhancing its practical performance.

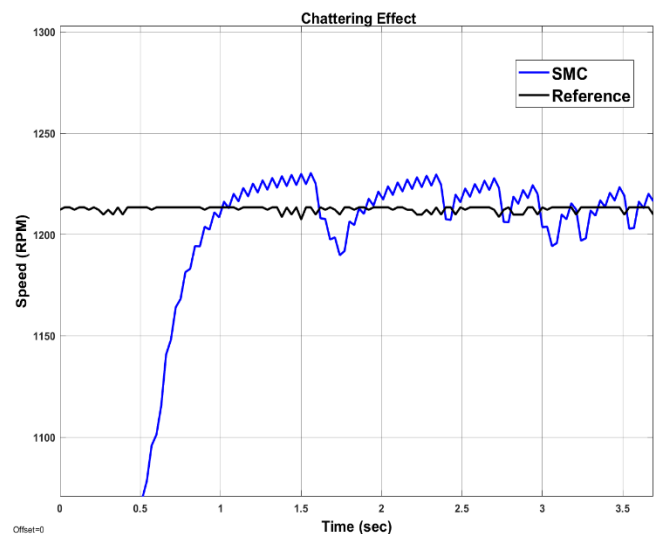


Figure 14. SMC<sub>1</sub> chattering effect

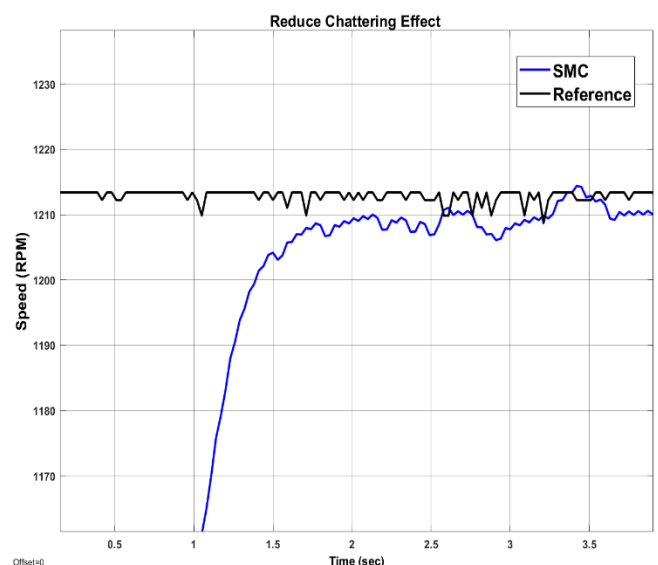
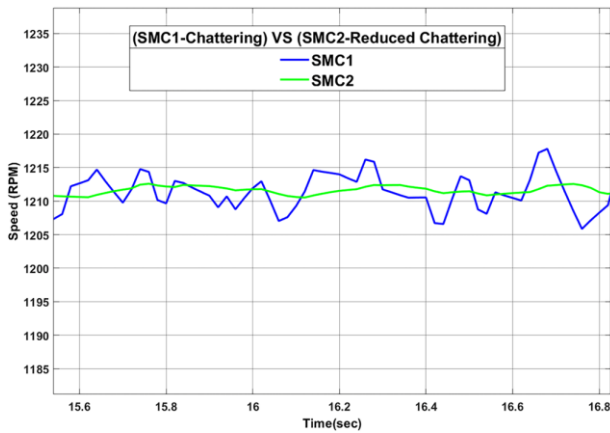


Figure 15. SMC<sub>2</sub> reduced the chattering effect





**Figure 16.** SMC<sub>1</sub> chattering VS SMC<sub>2</sub> reduced chattering

Figure 16 illustrates the performance of the SMC system both before and after the introduction of the smoothing factor ( $\delta$ ) to reduce chattering. The curve representing the SMC with chattering shows significant high-frequency oscillations resulting from the control law's inherent switching nature. In contrast, the modified SMC, which includes the smoothing factor, exhibits a much smoother response with minimal residual oscillations.

## 5. CONCLUSION

This study provides a detailed analysis and practical implementation of three advanced control strategies: Proportional-Integral-Derivative (PID), Fuzzy Logic Control (FLC), and Sliding Mode Control (SMC) for accurate speed regulation of DC motors. By combining theoretical simulations in MATLAB Simulink (2023b) with practical experiments using an Arduino Mega, the research emphasizes the strengths and weaknesses of each controller in real-world conditions.

The PID controller demonstrated quick response characteristics, achieving a rise time of 0.236 seconds, a settling time of 0.57 seconds, and an overshoot of 14.5%. Its simplicity and efficiency make it highly suitable for stable and predictable environments. In contrast, the FLC demonstrated strong adaptability, with a rise time of 0.54 seconds and a settling time of 2.00 seconds, but with a higher overshoot of 27.08%. This reflects challenges in managing transient behavior caused by environmental disturbances. The SMC showcased exceptional precision and robustness, achieving zero overshoot, a rise time of 1.43 seconds, and a settling time of 1.42 seconds. However, initial experiments indicated significant chattering due to high-frequency switching, which is a common limitation of SMC.

To address this issue, a smoothing factor ( $\delta$ ) was introduced into the SMC control law. This modification significantly reduced chattering and high-frequency oscillations, resulting in a smoother response with minimal residual oscillations while preserving the inherent robustness of SMC. The critical role of  $\delta$  in enhancing the practicality of the SMC underscores its potential for precise control in non-ideal conditions. Based on the findings, the following recommendations are proposed to guide the selection of control strategies:

- PID: Best suited for stable environments, such as conveyor systems and household appliances.
- FLC: Ideal for nonlinear systems, such as robotic arms and

adaptive automotive technologies.

- SMC: Recommended for high-precision applications, including satellite control and surgical robotics, where robustness and zero overshoot are essential.

This study bridges the gap between theoretical modeling and practical implementation, addressing challenges such as unmodeled dynamics and sensor noise. Future research could build on these findings by incorporating artificial intelligence techniques, such as machine learning, to dynamically optimize controller parameters and enhance system performance further. By providing actionable insights and quantitative benchmarks, this research contributes to advancing control systems in industrial automation and precision engineering.

## REFERENCES

- [1] Labaran, S.M., Bashir, H.A. (2018). DC motor speed control analysis: An overview. In 3rd National Engineering Conference on Bridging the Gap between Academia and Industry (ACICon2018), Kano, Nigeria, pp. 47-54.
- [2] Chen, J. (2024). The application of PID control in DC motor control systems. *Applied and Computational Engineering*, 81: 164-170. <https://doi.org/10.54254/2755-2721/81/20241083>
- [3] Almwala, A.M., Hussein, M.J., Abdullah, A.T. (2024). A comparative study of DC motor speed control techniques using fuzzy, SMC and PID. *Journal Européen des Systèmes Automatisés*, 57(2): 397-406. <https://doi.org/10.18280/jesa.570209>
- [4] Lu, J., Ma, G., Zhang, G. (2024). Fuzzy machine learning: A comprehensive framework and systematic review. *IEEE Transactions on Fuzzy Systems*, 32(7): 3861-3878. <https://doi.org/10.1109/TFUZZ.2024.3387429>
- [5] Ghalimath, P., Sankeswari, S.S. (2015). Speed control of DC motor using sliding mode control approach. *IOSR Journal of Electrical and Electronics Engineering*, 10(4): 17-22. <https://doi.org/10.9790/1676-10411722>
- [6] Cherif, K., Sahbani, A., Saad, K.B. (2021). Speed control for BLDC motor using fuzzy sliding mode. In *Innovative and Intelligent Technology-Based Services for Smart Environments-Smart Sensing and Artificial Intelligence*, pp. 163-170. <https://doi.org/10.1201/9781003181545-24>
- [7] Jain, S., Israr, M., Samuel, P. (2023). Closed loop control of brushless DC motor by various controllers for lightweight EVs. In *2023 5th International Conference on Power, Control & Embedded Systems (ICPCES)*, Allahabad, India, pp. 1-6. <https://doi.org/10.1109/ICPCES57104.2023.10076051>
- [8] Othman, A.A., Shatla, H.A., Hamdy, M. (2022). A comparative study of auto-tuned PID controller based on different SMC schemes for a DC motor speed control: Practical validation. In *2022 23rd International Middle East Power Systems Conference (MEPCON)*, Cairo, Egypt, pp. 1-6. <https://doi.org/10.1109/MEPCON55441.2022.10021808>
- [9] Rikwan, R., Ma'arif, A. (2023). DC motor rotary speed control with Arduino UNO based PID control. *Control Systems and Optimization Letters*, 1(1): 27-31. <https://doi.org/10.59247/csol.v1i1.6>
- [10] Adel, Z., Hamou, A.A., Abdellatif, S. (2018). Design of real-time PID tracking controller using Arduino Mega 2560 for a permanent magnet DC motor under real

- disturbances. In 2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM), Algiers, Algeria, pp. 1-5. <https://doi.org/10.1109/CISTEM.2018.8613560>
- [11] Khairudin, M., Yatmono, S., Nashir, I.M., Arifin, F., Aulia, W. (2021). Exhaust fan speed controller using fuzzy logic controller. In *Journal of Physics: Conference Series*, 1737(1): 012046. <https://doi.org/10.1088/1742-6596/1737/1/012046>
- [12] Prasannavenkadesan, V., Narayanan, K.B., Raja, S. (2024). Modified fuzzy logic system based predictive model for cortical bone drilling temperature. *Fuzzy Information and Engineering*, 16(3): 207-219. <https://doi.org/10.26599/FIE.2024.9270042>
- [13] Boffa, S. (2022). Extracting concepts from fuzzy relational context families. *IEEE Transactions on Fuzzy Systems*, 31(4): 1202-1213. <https://doi.org/10.1109/TFUZZ.2022.3197826>
- [14] Neda, O.M. (2024). Enhancing performance of BLDC: An improved fast terminal slide mode control for BLDC motor speed control. *Research Square*. <https://doi.org/10.21203/rs.3.rs-4802199/v1>
- [15] Hadi, A.R.S., Alamili, A., Abbas, S. (2023). A tracking control design for a DC motor using robust sliding mode learning control. *International Journal of Power Electronics and Drive System*, 14(4): 1937-1945. <https://doi.org/10.11591/ijpeds.v14.i4.pp1937-1945>
- [16] Why don't switches appear in circuit diagrams of DC motors? *Stack Exchange*. <https://electronics.stackexchange.com/questions/263425/why-dont-switches-appear-in-circuit-diagrams-of-dc-motors>
- [17] JGA25-371 Geared Motor with Encoder. [https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/694/114090047\\_Web.pdf](https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/694/114090047_Web.pdf)
- [18] Hamoodi, A.N., Mohammed, R.A., Hamoodi, S.A. (2023). Automatic speed control of DC series motor by using Arduino. *Przeglad Elektrotechniczny*, 99(3): 137-141. <https://doi.org/10.15199/48.2023.03.24>
- [19] Nayak, S.G., Patil, A., Gadagenavar, G. (2022). Development and analysis of speed control of BLDC motor with Arduino controller. *International Journal of Innovative Technology and Exploring Engineering*, 11(4): 63-66. <https://doi.org/10.35940/ijitee.C9813.0311422>
- [20] Maryadi, T.H.T., Wardhana, A.S.J., Damarwan, E.S., Sukisno, T. (2021). Performance of frequency regulation on power in the development of variable speed drive training kit as a learning media for motor control practices. *Journal of Physics: Conference Series*, 2111(1): 012036. <https://doi.org/10.1088/1742-6596/2111/1/012036>
- [21] Supriyono, H., Alanro, F.F., Supardi, A. (2024). Development of DC motor speed control using PID based on Arduino and MATLAB for laboratory trainer. *Jurnal Nasional Teknik Elektro*, 13(1): 36-41. <https://doi.org/10.25077/jnte.v13n1.1155.2024>
- [22] Abuelenin, S.M., Abdel-Kader, R.F. (2022). Closed-form mathematical representations of interval type-2 fuzzy logic systems with application to inverted pendulum stabilization. *Fuzzy Information and Engineering*, 14(3): 269-290. <https://doi.org/10.1080/16168658.2022.2152922>
- [23] Odry, Á., Tadic, V.L., Odry, P. (2021). A stochastic logic-based fuzzy logic controller: First experimental results of a novel architecture. *IEEE Access*, 9: 29895-29920. <https://doi.org/10.1109/ACCESS.2021.3059482>
- [24] Lakshmi, R., Gomalavalli, R., Raju, N.R., Kumar, K.M. (2023). Comparison of transient responses of two switch forward converter with sliding mode controller and HC-Fuzzy controller. *Journal of Emerging Technologies and Innovative Research*, 10(1): 277-287.
- [25] Prasetyo, H.F., Rohman, A.S., Santabudi, M.R.A.R. (2017). Implementation of model predictive control using Algorithm-3 on Arduino Mega 2560 for speed control of BLDC motor. In 2017 3rd International Conference on Science in Information Technology (ICSITech), Bandung, Indonesia, pp. 642-647. <https://doi.org/10.1109/ICSITech.2017.8257192>
- [26] Jayatileke, H.R., De Mei, W.R., Ratnayake, H.U.W. (2014). Real-time fuzzy logic speed tracking controller for a DC motor using Arduino due. In 7th International Conference on Information and Automation for Sustainability, Colombo, Sri Lanka, pp. 1-6. <https://doi.org/10.1109/ICIAFS.2014.7069560>
- [27] Singh, V.K., Sahu, A., Beg, A., Khan, B., Kumar, S. (2018). Speed & direction control of DC motor through Bluetooth HC-05 using Arduino. In 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), Bhopal, India, pp. 1-3. <https://doi.org/10.1109/ICACAT.2018.8933698>
- [28] Bassetto, M., Mengali, G., Abu Salem, K., Palaia, G., Quarta, A.A. (2023). A sliding mode control-based guidance law for a two-dimensional orbit transfer with bounded disturbances. *Actuators*, 12(12): 444; <https://doi.org/10.3390/act12120444>
- [29] Saputra, D.D., Ma'arif, A., Maghfiroh, H., Baballe, M.A., Tusset, A.M., Sharkawy, A.N., Majdoubi, R. (2023). Performance evaluation of sliding mode control (SMC) for dc motor speed control. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 9(2): 502-510. <https://doi.org/10.26555/jiteki.v9i2.26291>