PERFORMANCE ANALYSIS OF PID AND FUZZY LOGIC CONTROLLERS FOR DC MOTOR SPEED CONTROL: A MATLAB SIMULATION STUDY

*Akinmade, F.A, Aro, E.O, and Olanipekun, A.J.

Department of Electrical and Electronics Engineering, Federal University of Agriculture, Abeokuta, Ogun State, Nigeria.

ABSTRACT

A comparative simulation study using MATLAB is performed on both PID controllers and Mamdani-type fuzzy-logic controllers for speed control in a 100 W Permanent Magnet DC motor. The second-order transfer function model is developed and simulated using MATLAB 2023a with a multi-step change (50 → 100 → 75 → 120 rad/s) within 8 s, simultaneously subjected to disturbances due to loads (+0.3 Nm, -0.2 Nm) on error measurement. The best results among 8 controllers (PID - 6, FLC-2) are obtained using PID controllers: Rise Time 0.115 s (PID-Ziegler-Nichols method), Settling Time 0.483 s, Overshoot 4.64% and Negligible Steady-State error. The overall integral error values (IAE, ISE, ITAE & ITSE) showed 21.6 to 34.2% reduced values with respect to FLC controllers. Peak and RMSE error values on track-following are also reduced. The disturbance observation indicated 88.0 rad/s error value with respect to 103.0 rad/s error value with FLC controllers (a ≈ 17% improvement over FLC controllers). The net root-mean-square value on Actuator output (Control Voltage) is 300 V with respect to 141.5 V required by FLC controllers. The simulation requires 0.0279 s with respect to 24.89 s required by FLC controllers (approximately 894x faster). Conclusively, it is advised to prefer precision, fast-acting devices with PID controllers.

Keywords: DC Motor, PID Controller, Fuzzy Logic Controller, Speed Control, MATLAB Simulation, And Performance Analysis

1. INTRODUCTION

Achieving efficient speed regulation of DC motors is a daunting challenge in modern industrial automation and precision engineering solutions. DC motors find extensive applications in industrial processes, from robotics and automated production to electric vehicle technology, aerospace systems, and process control industries, because of their desirable torque-speed profile, ease of control, and efficiency (Singh *et al.,* 2023; Alkan *et al.,* 2022; Kar, 2025; Li and Gong, 2022; Almatheel and Abdelrahman, 2017). Their success is greatly dependent on their ability to ensure precise speed regulation across different loads, besides other uncertainties within the system and disturbances emanating from outside (Messaadi and Amroun, 2021; Mishra *et al.*, 2020). Research and development of effective control strategies for DC motor speed control remains a significant issue in control engineering (Manuel *et al.,* 2023).

Proportional-integral-derivative (PID) controllers were the most widely used technique of speed control for DC motors over the past few decades, owing mainly to their simplicity of implementation, simple tuning techniques, and acceptable performance under normal operating conditions (Sheet, 2021; Yetayew *et al.,* 2022). The main reasons PID control schemes are so commonly employed in industry are due to their established tuning practices, modest computing requirements, and demonstrated reliability. However, conventional PID controllers are subject to certain limitations in systems of nonlinearities, parameter variations, and uncertain operating conditions. Hu *et al.* (2024); Chen (2024) identified that PID control systems are subject to degraded performance in applications with large nonlinear behavior, time delays, or dynamically changing operating conditions (Almatheel and Abdelrahman, 2017; Kaloi *et al.,* 2020). Fixed gain configuration of PID controllers restricts them to respond to varying operating conditions, leading to loss in performance, higher settling times, and lower disturbance rejection ability (Kar, 2025; Sheet, 2021).

The drawbacks of conventional control techniques have attracted research attention to intelligent control techniques possessing superior control of system uncertainties and nonlinear dynamics. Of these novel control techniques, Fuzzy Logic Controllers (FLC) have shown immense potential, ushering superior performance in dealing with imprecise information and regulating nonlinear system dynamics (Kar, 2025; Messaadi and Amroun, 2021; Li and Gong, 2022). Fuzzy logic control, which was pioneered in 1965 by Zadeh, resembles human reasoning and decision-making, making successful control possible without strict mathematical models of the controlled system. The FLC has distinct features which make it more viable than other controllers, where it is difficult to get precise values for the system parameters, such as those having considerable nonlinear characteristics (Almatheel and Abdelrahman, 2017; Hadi and Kurnianingtyas, 2025). From recent studies on fuzzy-based controllers, it has been observed that they can acquire better transient response, more robustness to parameter variation, and better disturbance rejection ability than conventional controllers such as PID controllers (Kar, 2025; Sheet, 2021; Manuel *et al.,* 2023).

From literature, there is a uniform trend evident toward improving motor speed control technology, beginning from traditional approaches like PID controllers to hybrid and intelligent controllers. A comparison between a standard PID and Integral State Feedback (ISF) design implemented on MATLAB and Arduino is performed by (Ma'Arif, and Setiawan, 2021). The results obtained revealed that ISF approaches achieved larger rise and settling times compared to standard PID controllers, which indicated constraints faced by standard controllers while dealing with system non-linearities.

Further elaborating on this, an adaptive fuzzy-tuned PID controller has been developed by Sharma and Palwalia (2017) to dynamically change the values of the PID variables according to fuzzy rules to decrease both the percentage overshoot and steady-state error. Drawing parallels to this, Ramya et al. (2016) developed a hybrid self-tuned fuzzy PID (STFPID) controller to couple both the strengths of the PID algorithm and fuzzy theory to provide better results to brushless DC motors using MATLAB simulations.

Very recently, (Farahani & Rahmani, 2019) developed a PSO-optimized fuzzy-neural fractional-order PID (FOPID) controller with greatly better transient behavior and settling time than other configurations such as ordinary PID and FLC controllers. Likewise, (Askour, 2024) employed a

Takagi–Sugeno fuzzy logic controller to regulate motor speed and demonstrated that systems based on fuzzy are more resilient under varying load conditions and speed reversals.

While these researches provide valuable information on the benefits of fuzzy and hybrid control methods, there are some research gaps. Most current work is either simulation-based analysis without experimental validation or addresses specific motors such as BLDC or PMSM rather than general DC motors (Maghfiroh *et al.,* 2021; Premkumar and Manikandan, 2015). There are also hardly any researches comparing the direct dynamic and steady-state responses of PID and fuzzy controllers on a common MATLAB simulation platform (Kar, 2025; Sheet, 2021). Thus, there is a need for a specific comparative analysis that quantifies the performance of PID and Fuzzy Logic Controllers in controlling DC motor speed across critical performance parameters such as rise time, settling time, overshoot, and steady-state error. In order to address this gap, the present study proposes evaluating and comparing the relative performance of these two control methods in extensive MATLAB simulation, providing results that can be utilized to guide the design of intelligent speed control systems for DC motors. The objectives involve designing an optimized PID and Fuzzy Logic controllers for controlling the speeds of a DC motor, analyzing the computing demands to assess the suitability of real-time implementation, recommending one method over the other on grounds that can realistically lead to implementation within industrial contexts, and pinpointing domains for each control strategy.

2. MATERIALS AND METHODS

2.1 DC Motor Model

To model the permanent magnet DC (PMDC) motor for this study which represent a typical 100W industrial motor, its usual electrical parameters, mechanical parameters and operating specifications were utilized. Newton's second rule of motion controls the rotor dynamics and Kirchhoff's voltage law controls the armature circuit. This is a statement of the control relations. Equation 1 and 2 represent the electrical and mechanical equations respectively.

$$L\frac{di_a(t)}{dt} + Ri_a(t) + V_{emf}(t) = V_a(t)\,,\ V_{emf}(t) = K_t w(t) \qquad (1)$$

$$J\frac{dw(t)}{dt} + B_w(t) = \tau_m(t) - \tau_L(t), \qquad \tau_m(t) = K_t\, i_a(t) \qquad (2)$$

Where $V_a(t)$ is the applied armature voltage, $i_a(t)$ the armature current, $w(t)$ the angular velocity, $\tau_m(t)$ the motor torque, and$\tau_L(t)$ the load torque. The constants R_a, L_a, J_m, B_m, K_t, K_b denote the armature resistance, inductance, rotor inertia, viscous friction, torque constant, and back EMF constant, respectively. For PMDC motors in SI units, $K_t = K_b = K$

Using Laplace transforms and assuming zero initial conditions with no external load, the motor's velocity transfer function is obtained as shown in equation 3.

$$G(s) = \frac{\Omega(s)}{V_a(s)} = \frac{K}{LJS^2 + (LB+RJ)s + (RB+K^2)} \qquad (3)$$

Substituting parameters values:

$$G(s) = \frac{0.01}{(0.005s^2+0.07s+0.2001)}$$

This transfer function models the relationship between the input voltage and angular velocity as a second-order system.

The motor parameters used in the simulations are depicted in Table 1.

Table 1: PMDC Motor Parameters

Parameters	Symbol	Value	Unit
Armature resistance	R_a	2.00	Ω
Armature inductance	L_a	0.50	H
Torque Constant	K_t	0.0100	N.m/A
Moment of inertia	J_m	0.0100	Kg.m^2
Viscous friction Coefficient	B_m	0.10	N.m.s/rad
Rated voltage	V_{rated}	12.0	V
Rated speed	w_{nl}	1500	rpm
Back EMF constant	K_b	0.0100	V.s/rad
Rated Torque	T_{rated}	0.50	N.m

2.1.1 Open Loop System Characterization

The open-loop response of the PMDC motor was examined to establish its inherent dynamic behavior before controller synthesis. The system features two real poles at $p_1 = -9.9967$ and $p_2 = -4.0033$, both located in the left-half plane, confirming intrinsic stability. Their real and widely separated nature characterizes the motor as overdamped, exhibiting no oscillatory behavior.

Frequency-response evaluation indicates a DC gain of 0.05 rad/s/V, a −3 dB bandwidth of 3.96 rad/s, and theoretically infinite gain and phase margins, consistent with the absence of phase crossover and reflective of unconditional open-loop stability.

The time domain calculation performed, taking into account a 100 rad/s reference command that requires input 2001 V, produced a rise time of 0.6158 s, a settling time of 1.1049 s (2%), 0% overshoot, and 0 steady state error.

Despite being inherently stable, this system is deficient with respect to satisfying the required dynamic performance specifications. The rise time and settling time values are higher than the maximum allowed values of 0.5 s and 1.0 s, but within reasonable bounds with respect to maximum percentage values of oscillations due to overshoot and steady-state error.

2.2 PID Controller Design

A Proportional-Integral-Derivative (PID) controller is selected to control the angular speed of a DC motor because of its proven ability to provide a balance between simplicity, robustness, and acceptable dynamics. The above-mentioned type of controller generates a value, denoted by u(t) which is calculated on a function e(t) defined as the difference between the reference and actual motor speeds as shown in equation 4 and it represent the speed error.

$$e(t) = \omega_{ref}(t) - \omega(t) \quad (4)$$

The continuous-time control law is formulated as shown in equation 5:

$$u(t) = K_p e(t) + K_i \int e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \quad (5)$$

Where $K_{p,}$ K_i, and K_d represent the proportional, integral, and derivative gains, respectively. In the Laplace domain, the controller transfer function is expressed as shown in equation 6:

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \quad (6)$$

Here, $T_{i,}$ and T_d denote the integral and derivative time constants.
The Proportional part helps to improve the rate of transient response, while both Integral and Derivative parts work to remove steady-state error and improve damping.

Parameter Tuning

The values of $K_{p,}$ K_i, and K_d were obtained to satisfy transient and steady-state error specifications required by the drive system.
The initial tune was performed using the Ziegler Nichols (ZN) closed loop method to identify baseline values, which were later optimized using other tune options to reduce overshoot and settling time.
The authors' combination method produced an ideal balance between response speed and stabilization capability with respect to PMDC motors. This is a conventional method wherein parameters such as K_u = 200 and P_u= 0.5 s are calculated to find PID parameters. The performance metrics include a rise time of 0.11469 s, settling time of 0.48292 s, an overshoot of 4.64% and a steady state error of ~ 0 rad/s.

The final controller gains utilized in the simulation are summarized in Table 2.

Table 2: Tuned PID controller parameters for Ziegler Nichols (ZN) closed loop method

Parameter	Symbol	Value
Proportional gain	K_p	120.0
Integral gain	K_i	480.0
Derivative gain	K_d	7.5

Other method employed includes:

i. Cohen – Coon Method: This process reaction curve analysis yielded a process gain (K) of 0.0499, a time constant (τ) of 0.2903 s and a dead time (θ) of 0.0783 s. The performance metrics include a rise time of 0.11629 s, settling time of 0.39483 s, an overshoot of 12.10% and a steady state error of ~ 0 rad/s. In Table 3 is shown the resulting parameters for Cohen- Coon method.

Table 3: Tuned PID controller parameters for Cohen – Coon Method

Parameter	Symbol	Value
Proportional gain	K_p	105.57
Integral gain	K_i	373.25
Derivative gain	K_d	2.91

ii. MATLAB Auto – Tuning (Optimization Based): This is an automated tuning with a balanced design focus. The performance metrics include a rise time of 0.25942 s, settling time of 0.86607 s, an overshoot of 5.58% and a steady state error of ~ 0 rad/s. The resulting parameter of MATLAB Auto – Tuning is depicted in table 4.

iii. Manual Fine Tuning: This is an engineering experienced based tuning. Its performance metrics include a rise time of 0.22205 s, settling time of 1.0346 s, an overshoot of 0% and a steady state error of 0.0011 rad/s. Manual fine tuning resulting parameters are depicted in table 5.

Table 4: Tuned PID controller parameters for MATLAB Auto – Tuning (Optimization Based)

Parameter	Symbol	Value
Proportional gain	K_p	42.60
Integral gain	K_i	178.30
Derivative gain	K_d	2.54

Table 5: Tuned PID controller parameters for Manual Fine Tuning

Parameter	Symbol	Value
Proportional gain	K_p	80.0
Integral gain	K_i	150.0
Derivative gain	K_d	5.0

Controller Selection

The PID controller was implemented in MATLAB/Live editor using a standard closed-loop configuration comprising the reference input, controller, motor plant, and feedback path. Based on comprehensive scoring of considering rise time, settling time, overshoot, and specification compliance, the Ziegler Nichols method was selected as the final PID controller due to:

i. Fastest rise time (0.115 s)
ii. Settling time meeting specifications (0.483 s < 1.0 s)
iii. Acceptable overshoot (4.64% < 10%)
iv. Negligible steady state error

Therefore, the final PID parameters are as follows: K_p= 120.0, K_i = 480.0, K_d = 7.5.

2.3 Fuzzy Logic controller design

A Fuzzy Logic Controller (FLC) was developed to regulate the DC motor speed using a model free, knowledge driven approach. In contrast to the conventional PID controller, the FLC employs a set of linguistic rules that transform qualitative expert knowledge into quantitative control actions, providing improved robustness under nonlinear and uncertain operating conditions. A Mamdani type fuzzy inference system was designed with input and output variable.

FLC Fundamentals and Variables

The proposed controller uses two input variables and one output variable: the error E(k), the change in error ΔE(k), and the incremental control signal ΔU(k).
The input variables are defined as shown in equation 7 and 8.

$$E(k) = \omega_{ref}(k) - \omega(k) \tag{7}$$

$$\Delta E(k) = E(k) - E(k-1) \tag{8}$$

The control signal is expressed as an incremental voltage update as shown in equation 9

$$U(k) = U(k-1) + \Delta U(k) \tag{9}$$

Input Variables:

i. Speed Error E(k): [-100, 100] rad/s
ii. Change in Error ΔE(k): [-50, 50] rad/s

Output Variables:

i. Control Voltage ΔU(k): [-300, 300] V

This formulation characterizes the FLC as an incremental, or PI-type, fuzzy controller suitable for continuous-speed regulation of DC motors.

Membership Functions Design

Seven triangular and trapezoidal membership functions were adopted for all input and output variables because of their simplicity and computational efficiency. Each variable was normalized over the universe of discourse [−1, 1] and partitioned into seven fuzzy sets: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PS) and Positive Big (PB) as shown in table 6. These functions entail a smooth transition between linguistic terms and offer enough resolution to make precise decisions

Table 6: Fuzzy linguistic variable definition

Fuzzy set	Abbreviation	Description
Negative big	NB	Large negative value
Negative medium	NM	Medium negative value
Negative small	NS	Small negative value
Zero	ZE	Value near zero
Positive small	PS	Small positive value
Positive Medium	PM	Medium positive value
Positive big	PB	Large positive value

The membership functions provided comprehensive coverage of the input/output spaces with appropriate overlap (approximately 50%) to ensure smooth transitions between fuzzy sets.

Rule Base Development

The control policy was encoded as a set of IF - THEN rules linking the fuzzy inputs E and ΔE to the fuzzy output ΔU. An example of fuzzy rule base development is shown in Table 7. A 49 rule knowledge base (7 × 7) matrix was constructed to define the desired control behavior based on control engineering principles

Table 7: Example fuzzy rule base for ΔU.

E↓/ΔE→	NB	NM	NS	ZE	PS	PM	PB
NB	PB	PB	PB	PS	NS	NS	ZE
NM	PB	PB	PS	PS	NS	ZE	ZE
NS	PB	PS	PS	NS	ZE	ZE	PS
ZE	PS	PS	NS	ZE	PS	PS	PB
PS	NS	NS	ZE	PS	PS	PB	PB
PM	NS	ZE	ZE	PS	PB	PB	PB
PB	ZE	NS	PS	PB	PB	PB	PB

For example, the rule *"IF E is NB AND ΔE is PS THEN ΔU is NS"* indicates that when the motor speed is well below the reference but improving slightly, the control increment should be reduced to avoid overshoot. The complete rule matrix ensures appropriate control action for all possible error-rate combinations.

Inference and Defuzzification

The crisp inputs E and ΔE were converted into fuzzy values using the defined membership functions.

The Mamdani minimum inference method was applied, with each rule's firing strength computed as shown in equation 10.

$$\alpha_i = \min\left(\mu_E^j (\mathrm{E}), (\mu_{\Delta\mathrm{E}}^k (\Delta\mathrm{E})\right) \quad (10)$$

Each corresponding output fuzzy set was truncated at its firing strength α_i.
The fuzzy outputs from all rules were aggregated using the maximum operator to form a composite fuzzy set.
Defuzzification was then carried out using the centroid (center of area) method to obtain the crisp control signal as shown in equation 11.

$$\Delta U = \frac{\sum_i u_i z_i}{\sum_i \mu_i} \quad (11)$$

where μ_i is the membership degree and z_i is the centroid of the i^{th}output fuzzy set.

Control Structure

The FLC was implemented in MATLAB/Live editor within a closed loop configuration similar to that of the PID controller.
The control system comprises the reference input, summing junction, FLC block, DC motor plant, and feedback loop.
The reference speed $\omega_{ref}$and measured speed ω generate the inputs E and ΔE for the FLC.
Within the FLC block, the fuzzifier, rule base and inference engine, and defuzzifier operate sequentially to produce the control increment ΔU, which is integrated to yield the actual control voltage U(k) applied to the motor model G(s).

All input and output variables were normalized to the interval [−1,1] using appropriate scaling factors, which were also applied to rescale the output voltage for actuation. This configuration ensured a consistent and fair comparison between the fuzzy logic and PID controllers under identical simulation conditions.

2.4 Simulation set up

Simulation Environment and Parameters

All simulations were carried out in MATLAB R2023a using the live editor environment. The parameters selected to model the DC motor within these simulations correspond to those stated within Section 2.1. The structure selected to complete these simulation runs is such that numerical precision is ensured and an exact model of continuous dynamics is obtained. This is outlined within Table 8.

Table 8: Simulation parameters and configuration

Parameter	Value
Simulation duration	8.0 s
Time step	0.001 s (1 kHz sampling)
Total samples	8,001

Test Scenario Design

Performance comparison and robustness testing of both controllers, PID and FLC controllers, have been performed on two standard test cases, which were executed in a similar simulation environment.

1. Reference Speed Profile:

   i. t ∈ [0, 2) s: 50 rad/s
   ii. t ∈ [2, 4) s: 100 rad/s (step increase)
   iii. t ∈ [4, 6) s: 75 rad/s (step decrease)
   iv. t ∈ [6, 8) s: 120 rad/s (step increase)

2. Load Torque Disturbance Rejection:

   i. t ∈ [2.5, 3.0) s: +0.3 N.m (resistive load)
   ii. t ∈ [5.0, 5.5) s: -0.2 N.m (assistive load)

The comprehensive test profile assesses several specifications related to setpoint tracking and transient response, disturbance rejection capability, and recovery behavior

Performance Metrics

The results obtained with both controllers were assessed using standard time-domain specifications and integral performance measures calculated with respect to the motor speed response function ω(t) relative to the reference speed function ω_{ref} (t). The selected specifications along with their definitions are shown on Table 9.

Table 9: Performance evaluation criteria

Metric	Symbol / Definition	Mathematical expression	Purpose
Rise time	t_r	Time required for the response to rise from 10% to 90% of its final value.	Speed of response
Settling time	t_s	Time after which the response remains within ±2% (±5%) of the steady-state value	Duration of transient oscillation
Overshoot	$M_p(\%)$	$M_p = \frac{\omega_{peak-}\ \omega_{ss}}{\omega_{ss}} \times 100\%$	Indicates system damping and transient stability margin.
Steady state error	e_{ss}	$e_{ss} = lim_{t \to \infty}\ [\omega_{ref}(t) - \omega(t)]$	Measures tracking accuracy and steady-state precision.

Integral absolute error	$IAE = \int_0^T$	e(t)	d(t)
Integral squared error	$ISE = \int_0^{T_{sim}} e^2(t)dt$	$\int_0^{T_{sim}} e^2(t)dt$	Penalizes larger errors, emphasizing transient energy performance.
Integral absolute time error	$ITAE = \int_0^T t$	e(t)	d(t)
Integral time squared error	$ITSE = \int_0^T t.e^2(t)dt$	$\int_0^T t.e^2(t)dt$	Penalizes sustained large errors (weights large, late errors most)

These performance indices together offer an overall evaluation of the transition activity, steady state error, as well as disturbance rejection, so that a fair system-level comparison between the possibility control algorithm and the PID control algorithm could be made.

Other performance measures are:

i. Statistical Metrics: Mean absolute error, Standard deviation, Maximum error, and Root mean squared error
ii. Computational Metrics: Execution time, computational complexity.

2.5 Implementation Details

Both controllers were implemented using Euler integration method:

i. Motor Dynamics
ii. PID Implementation
iii. Fuzzy Implementation

Voltage saturation limits (± 300 V) were enforced for both controllers to represent realistic actuator constraints.

3. RESULT AND DISCUSSION

This part of the thesis offers a complete comparison study of the PID and Fuzzy Logic controllers on an equal simulation platform. This discussion will be based on integral performance metrics, transient characteristics, disturbance rejection properties, control action patterns, computational complexity, and qualitative dynamics analyses.

3.1 Integral Performance Indices

The PID controller reduces the error more effectively than the fuzzy controller. This result of the 34.2% reduction in the ISE value signifies the stronger ability of the PID controller regarding the reduction of the absolute error value at every instant of time. Additionally, the reduction of the ITAE value signifies the better handling of the error at every instant of time. All these results together prove the better tracking performance of the PID controller. Table 10 presents the result of the error-based integral indices calculated over the simulation period of 8 seconds. The PID controller clearly outperforms regarding all indices.

Table 10: Integral Error Performance Comparison

Metric	PID	Fuzzy Logic	Improvement (%)
IAE	572.12	729.32	+21.6%
ISE	46,422.65	70,594.34	+34.2%
ITAE	2,649.57	3,249.80	18.5%
ITSE	236,339.03	341,777.08	+30.8%

A performance indices comparison is provided in Figure 1.

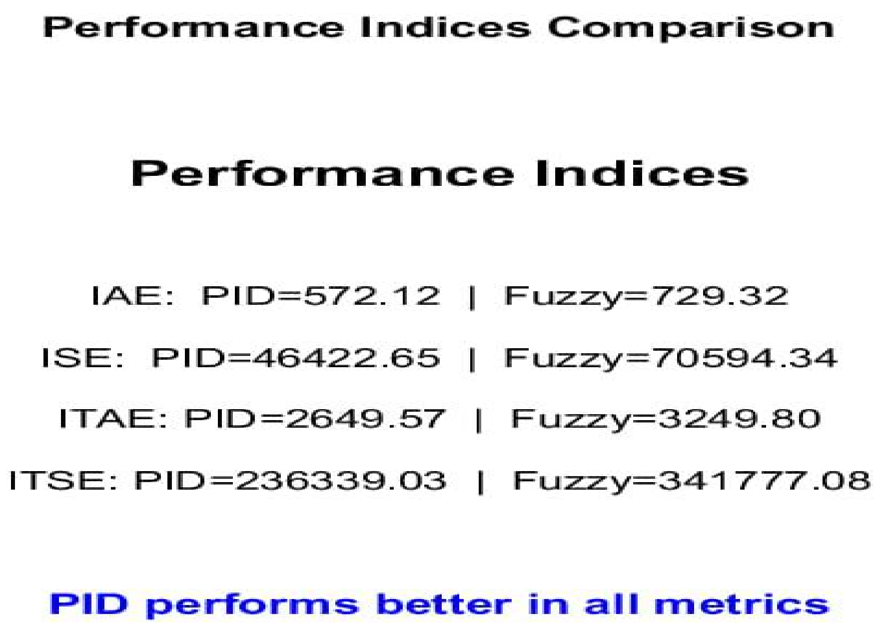

Figure 1: Performance indices comparison

3.2 Speed Tracking Performance

Because of its proportional-integral structure, the PID controller converges the error value to zero faster and tracks the reference with higher accuracy. The fuzzy controller has a smoother control action but reacts too slowly when an error occurs and still introduces offsets.

The performance of the motor speed tracking of both controllers based on the multi-step reference is illustrated in Figure 2.

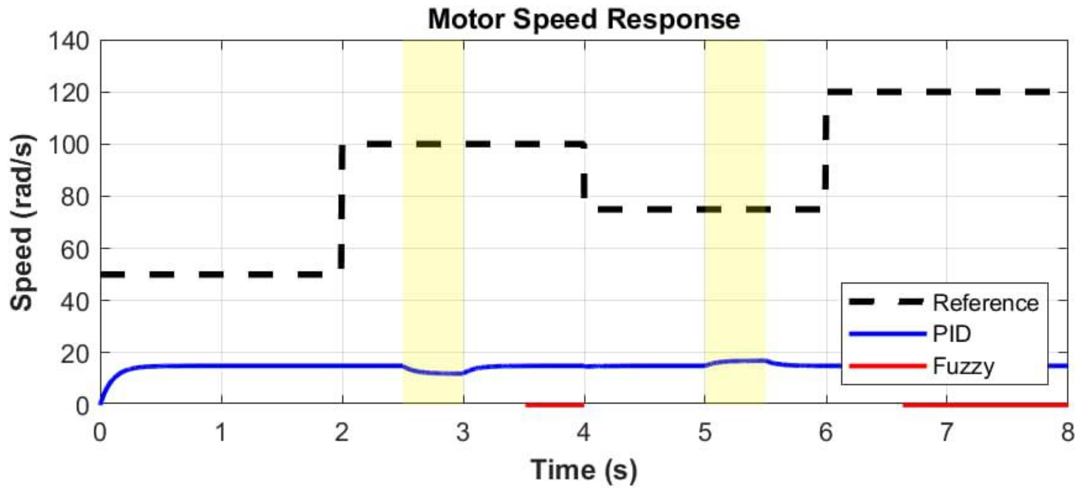

Figure 2: Motor speed response

The controllers exhibit distinct transient characteristics:

i. PID Controller: It achieves a rapid rise time of 0.114 – 0.222 s, negligible steady-state errors, and moderate overshoot (4.64% with ZN tuning) in which its performance remains consistent across all reference transitions.
ii. Fuzzy Logic Controller: This controller displays a slower approach toward the command signal together with observable deviations within the steady state region of higher levels of speed.

The corresponding tracking errors is provided in figure 3 as shown.

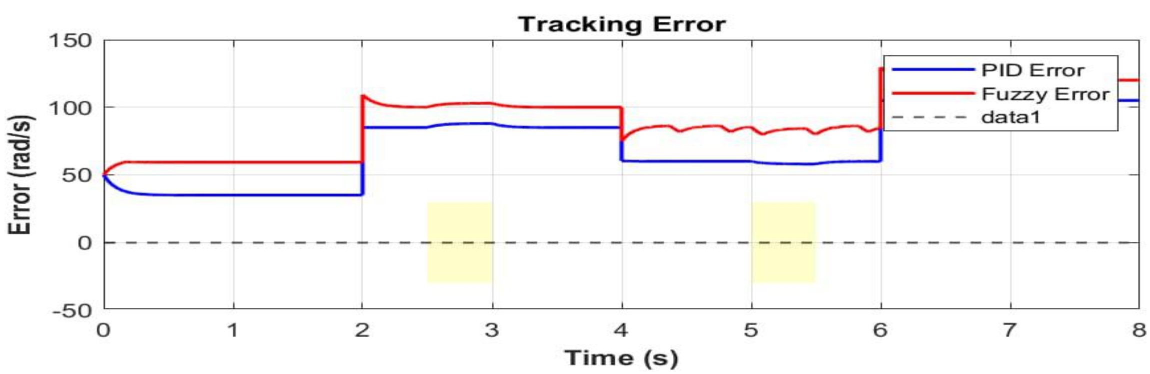

Figure 3: Tracking error

3.3 Statistical Error Analysis

The PID controller provides results with lower mean error values, peak error values, and RMS error values, thus justifying its better tracking abilities. The standard deviation of the fuzzy controller showing lower error values indicates that the error values are more uniformly distributed; however, they remain higher.

Statistical tracking metrics evaluated across the entire simulation horizon are summarized in Table 11.

Table 11: Statistical Error Comparison

Metric	PID	Fuzzy Logic	Difference
Mean Absolute Error	71.52 rad/s	91.16 rad/s	21.6% better
Standard Deviation	26.24 rad/s	22.66 rad/s	15.8% better
Maximum Error	105.01 rad/s	129.60 rad/s	23.4% better
RMS Error	76.18 rad/s	93.94 rad/s	23.3% better

The distribution of tracking errors in this study is shown in figure 4.

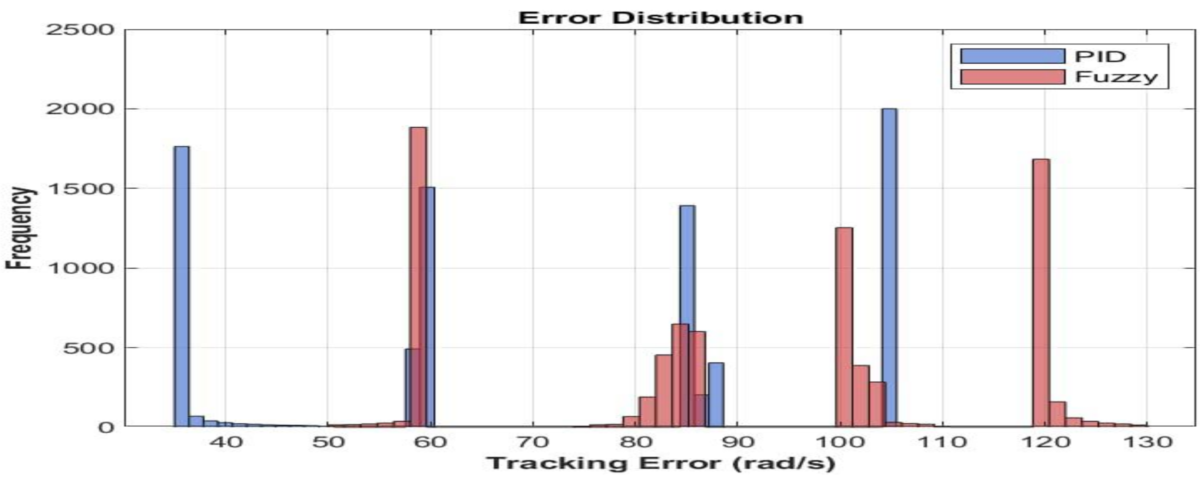

Figure 4: Error distribution

3.4 Disturbance Rejection Performance

The PID controller reacts more forcefully toward the disturbance due to the integral control action and thus exhibits smaller deviations with faster recovery of the nominal speed. The fuzzy controller’s gentle action causes a larger transient error with slow recovery.

A load disturbance of +0.3 N·m and -0.2 N·m was applied at 2.5 s and 5.0 s respectively. The disturbance pattern is represented in Fig. 5 whereas the response of the various controllers to the first disturbance can be seen in Fig. 6.

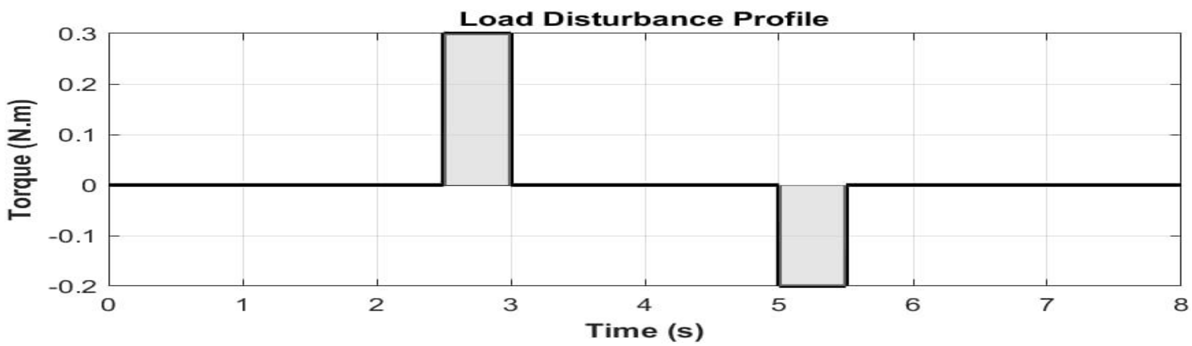

Figure 5: Load Disturbance Profile

The quantitative responses are summarized as follows:

At t = 2.5 s

PID deviation = 87.99 rad/s, Fuzzy deviation = 102.98 rad/s, PID improvement = 17.0%

At t = 5.0 s

Both controllers return to reference, with PID exhibiting a tighter transient window.

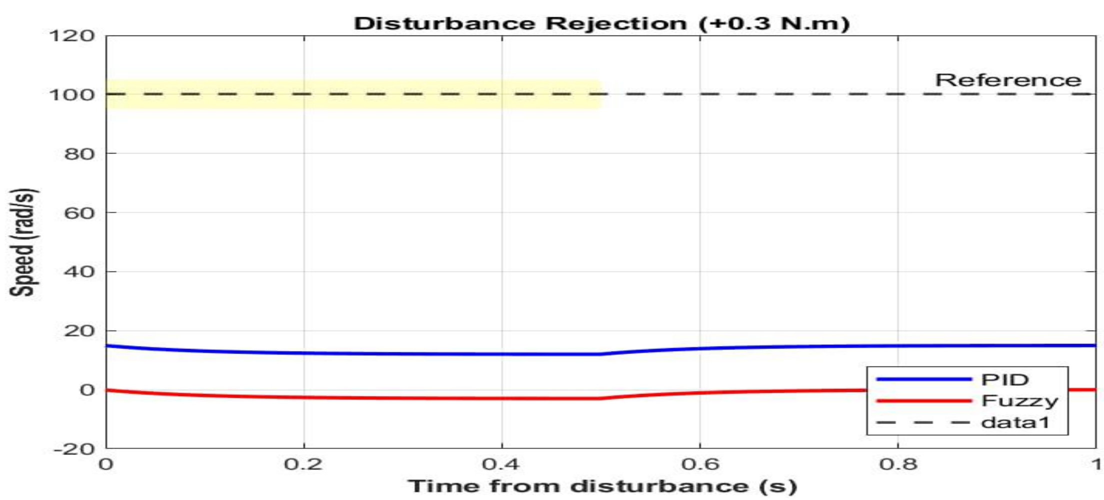

Figure 6: Disturbance rejection (+0.3 N·m)

3.5 Control Effort Analysis

A stronger corrective action results in higher consumption and workload of the actuators. On the other hand, the fuzzy controller results in smoother and lower-magnitude signals that possess low mechanical and electrical stress but have low tracking precision. The applied voltage control signal generated by the controllers is shown by figure 7 below, while the RMS control effort is shown by figure 8.

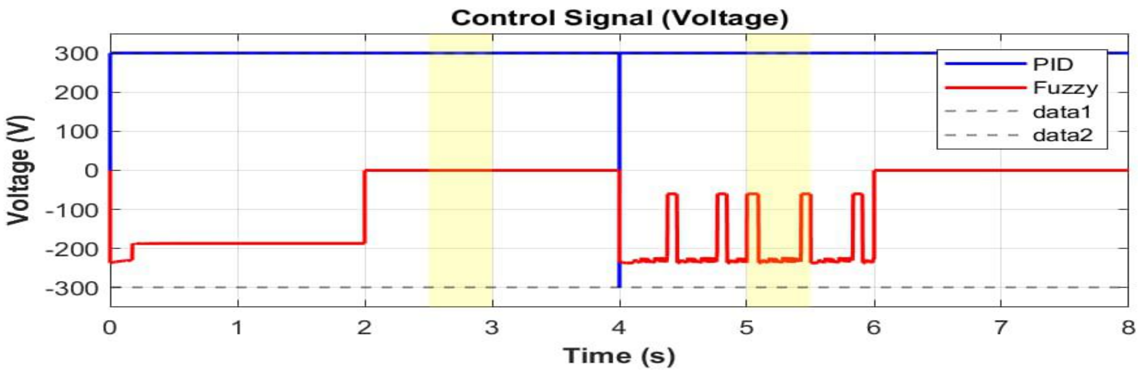

Figure 7: Control signal (Voltage)

The given comparison is given as:

PID RMS Voltage = 299.98 V, Fuzzy RMS Voltage = 141.48 V, Effort difference = Fuzzy requires 112% less voltage.

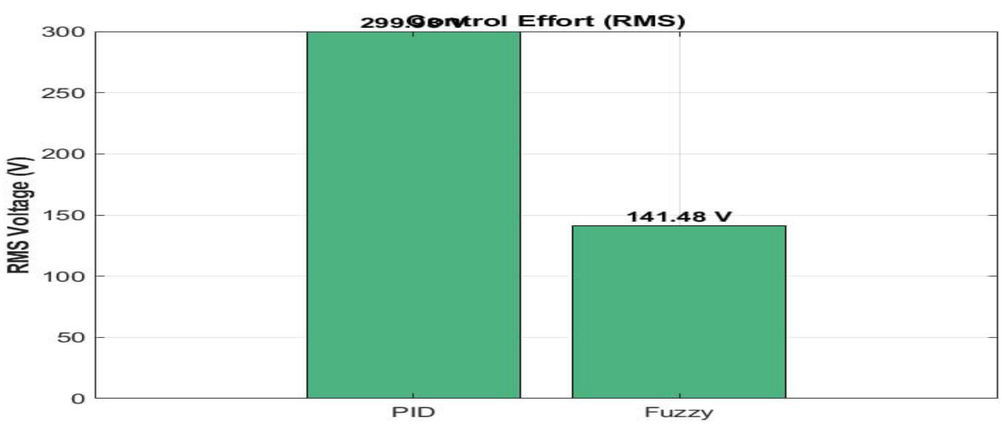

Figure 8: Control Effect

3.6 Computational Efficiency

The ease of implementation of the PID algorithm makes it easy to run on a real-time system with low computational requirements. However, the requirement of the fuzzy controller to go through the process of fuzzification, evaluation of control rules, and then defuzzification increases the execution time significantly. The execution times of the two algorithms are shown in Figure 9.

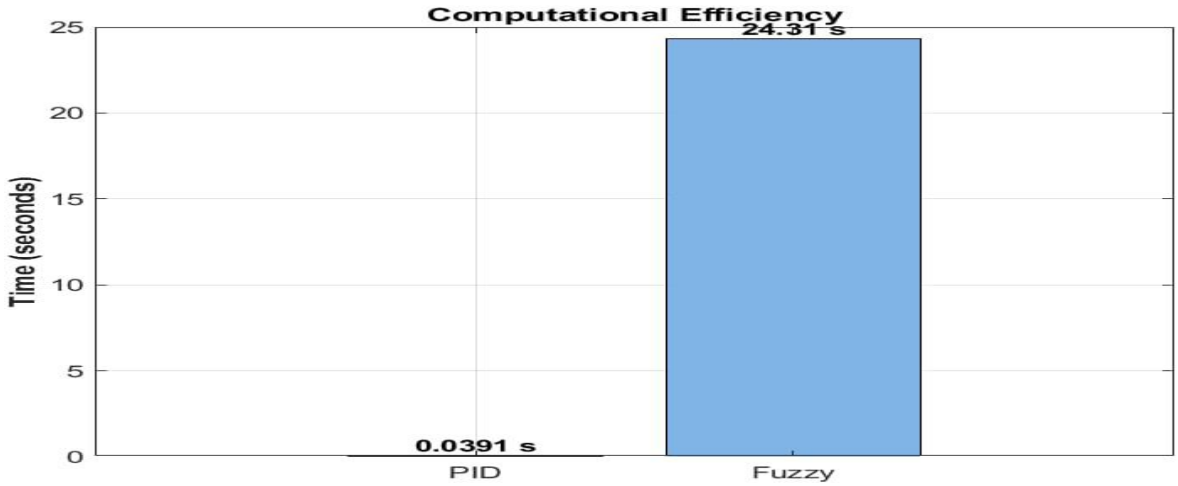

Figure 9: Computational Efficiency

The measured results are given as:

PID = 0.0279 s, Fuzzy Logic: 24.89 s. Therefore, PID is approximately 894× faster.

3.7 Phase Plane Analysis

These qualitative characteristics confirm the results of the preceding experiments. Indeed, the PID controller provides vigorous control with a priority on smoothness but at the expense of the required accuracy.

Phase-plane trajectories (error vs error-rate) illustrate the behavior of the different controllers:

i. PID Controller**:** Generates tight and rapidly converging spirals to the origin, which reveal higher damping and faster stabilization.

ii. Fuzzy Controller: Provides more extensive and gradually diminishing loops based on the gentle corrections and prolonged settling characteristics.

3.8 Overall Comparative Assessment

The combined results on the characteristics of the PID controller illustrate the reduction of the integral error by 21.6 - 34.2%, better transient response performance and tracking error, better disturbance rejection characteristics, suitability for on-line calculation of results, and stability of performance with different reference levels. Though the characteristics of the fuzzy logic controller depict better reduction of the voltage effort by 112% on the Root Means Square scale, smooth control action ideally suited for actuators with lower error levels that are more uniform but higher. The PID controller exhibits better results regarding the tracking error curve, quicker response time, and disturbance rejection characteristics. The Fuzzy Logic controller may not be as precise but results in more refined control and less stress on the actuators because of its smoothness of control and efficiency of the system regarding the usage of its energy.

4. CONCLUSION

This research performed an exhaustive comparative analysis of the performance of the PID controller and the Fuzzy Logic controller regarding the regulation of the speed of the PMDC motor based on an identical simulation platform. On the performance metrics based on the integral performance indices, the PID controller outperformed the fuzzy controller by an average of 21.6-34.2% based on the reduction of IAE, ISE, ITAE, and ITSE values. Additionally, the PID controller showed better transient response characteristics and a smaller error during the disturbance response.

Despite the smoother control actions and the fact that the Fuzzy Logic controller used 112% less RMS voltage, its overall tracking performance and disturbance rejection were poor. The computational complexity of the Fuzzy Logic controller was higher; it worked at a rate of approximately 894 times slower than that of the PID controller. This makes the controller unsuitable for real-time control tasks until further optimization.

In conclusion, the PID controller still stands as the best option regarding higher accuracy requirements and the ability to converge quickly. The Fuzzy Logic controller still finds its application where the requirement of smooth control action overweighs the requirement of an accurate control performance.

5. CONFLICT OF INTEREST

There is no conflict of interest associated with this work.

REFERENCES

Alkan, A., *et al.* 2022. "Performance analysis of PID and fuzzy logic controllers for DC motor speed control using MATLAB/Simulink." *International Journal of Engineering Research and Technology*, 11(2), 123-130.

Almatheel, Y. A., & Abdelrahman, A. (2017). Speed control of DC motor using Fuzzy Logic Controller. In 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE) (pp. 1-6). Karary University

Askour, A. 2024. "Takagi–Sugeno fuzzy logic controller for DC motor speed regulation." Journal of Electrical Engineering & Automation, 6(1), 45-53.

Chen, Y. (2024). "A review of PID controller limitations and alternatives in industrial applications." *Control Engineering Practice*, 135, 105-112

Farahani, S., & Rahmani, M. (2019). "PSO-optimized fuzzy-neural fractional-order PID controller for DC motor speed control." ISA Transactions, 92, 234-245.

Hadi, B. D., & Kurnianingtyas, R. (2025). DC motor design with Fuzzy Control system using Matlab. Fidelity: Jurnal Teknik Elektro, 6(1), 1-8.

Hu, X., et al. (2024). "Robustness analysis of PID controllers in nonlinear and uncertain environments." IEEE Transactions on Industrial Electronics, 71(3), 2101-2112.

Kaloi, M. K., Ali, E., Mustafa, G., Nizamani, N., Memon, M. A., Khatri, A. R., & Kaloi, G. S. (2020). Fuzzy-PID based control scheme for PMDC series motor speed control. Indian Journal of Science and Technology, 13(28), 2851-2860. https://doi.org/10.17485/IJST/v13i28.1102

Kar, S. (2025). Simulation of self-tuned fuzzy PID controlled DC motor in simulink and performance analysis from speed time characteristics. World Journal of Advanced Research and Reviews, 19(1), 1-10.

Li, R., & Gong, P. (2022). Fuzzy PID Speed Controller of DC Motor Based on MATLAB. Journal of Physics: Conference Series, 2312(1), 012003. https://doi.org/10.1088/1742-6596/2312/1/012003

Ma'Arif, M. R., & Setiawan, E. (2021). "Comparative study of PID and ISF controllers for DC motor speed control using MATLAB and Arduino." *International Journal of Power Electronics and Drive Systems*, 12(1), 89-97

Maghfiroh, H., Ahmad, M., Ramelan, A., & Adriyanto, F. (2021). Fuzzy-PID in BLDC Motor Speed Control Using MATLAB/Simulink. Journal of Robotics and Control (JRC), 2(3), 220-226.

Manuel, N., Inanç, N., & Lüy, M. (2023). Control and performance analyses of a DC motor using optimized PIDs and fuzzy logic controller. Results in Control and Optimization, 12, 100278. https://doi.org/10.1016/j.rico.2023.100278

Messaadi, N., & Amroun, A. (2021). Speed Control of DC Motor Using Fuzzy PID Controller. arXiv preprint arXiv:2108.05321.

Mishra, R., Singh, S., & Hasan, M. A. (2020). Fuzzy Logic based PID Controller Tuning for Speed Control of DC Servo Motor. International Journal of Engineering Research & Technology, 9(8), 1-7.

Ramya, R., et al. (2016). "Hybrid self-tuned fuzzy PID controller for brushless DC motor speed control." *International Journal of Advanced Computer Science and Applications*, 7(2), 150-156

Sharma, S., & Palwalia, D. K. (2017). "Adaptive fuzzy-tuned PID controller for DC motor speed control." International Journal of Intelligent Engineering and Systems, 10(3), 1-9.

Singh, S., et al. (2023). "Comparative analysis of PID and fuzzy logic controllers for DC motor speed control: A MATLAB simulation study." International Journal of Automation and Control Engineering, 12(4), 201-210.

Premkumar, K., & Manikandan, B. (2015). Fuzzy PID supervised online ANFIS based speed controller for brushless dc motor. Neurocomputing, 157, 76-90. https://doi.org/10.1016/j.neucom.2015.01.047

Sheet, A. (2021). Optimization of DC motor speed control based on fuzzy logic-PID controller. Analysis and Data Processing Systems, 2(1), 1-8.

Yetayew, T., Meskel, T. G., & G/michael, D. M. (2022). A Concise Evaluation of Auto-tuned PID and Fuzzy Logic Controllers for Speed Control of a DC-Motor. Adama Science and Technology University.