ARTICLE
JOURNAL`S UNIVERSITY OF BABYLON FOR ENGINEERING SCIENCES (JUBES)
مـــجلــة جـــامعة بـــابــل للعلـــــوم الهندسية
Vol. 33, No.6. \ 2025 ISSN: 2616 - 9916

Hybrid PID–Fuzzy Control for Precise DC Servomotor Speed Regulation

Sarkar Jawhar Mohammed Shareef [1] Salar Ahmed Qadir [2] Arfan M.Salih Hassan[3]

[1] *Department of Electrical Engineering, College of Engineering, Salahaddin University-Erbil, Kurdistan Region, Iraq.*

[2,3] *Department of Electrical and Computer Engineering, College of Kalar Technical, Garmian Polytechnic University, Kurdistan Region, Iraq.*

sarkar.mohammed@su.edu.krd salar.ahmed@gpu.edu.iq erfan.m.salih@gpu.edu.iq

Received:	31/7/2025	Accepted:	24/12//2025	Published:	31/12/2025

Abstract

Servo motors have been adopted in various industry applications, as a result of which there has been a trend toward improving their speed accuracy. DC servomotors with feedback encoders established themselves as participants for various motion control applications in industrial automation and consumer electronics. This article presents speed control of DC servomotors with different control methods (traditional PID, fuzzy logic, and hybrid PID–fuzzy logic controller). Although of simplicity, there are constraints to the PID controllers for nonlinearities and uncertainties; hence, a significant step has been made toward intelligent and hybrid control strategies. In MATLAB Simulink the control performance of these controllers is simulated, and it is noted that the hybrid PID–fuzzy logic controller has less settling time, very little overshoot, and near-zero steady-state error. This hybrid approach offers the accuracy of the PID control system and the flexibility of the fuzzy logic controller and introduces a robust and effective method to enhance the motor speed response for the servo motor application. applications,

Keywords: Servo Motor; PID Controller; Fuzzy Logic Control; Hybrid PID and Fuzzy Logic Control.

1 Introduction

A DC servomotor is a device that transforms electrical energy into mechanical motion. The DC servomotor differs from a standard DC motor in that it is equipped with an encoder to provide speed feedback [1]. Servomotors are essential components in a wide range of applications, particularly in industrial automation and various indoor devices. They are commonly found in systems that require precise control of angular or linear position, velocity, and acceleration. Notable examples of indoor applications include CD/DVD players and computer hard disk drives, where servomotors control the positioning of optical or magnetic heads with high accuracy and speed [2].

A typical servomotor system is designed to provide accurate position control. These systems are employing a DC or AC electric motor as the primary source of actuation to produce controlled mechanical movement. The operation of a servomotor is fundamentally based on the principle of negative feedback, a core concept in control theory. In a feedback control system, a command signal (input) representing the desired position or speed is continuously compared to the actual output of the system. The actual output is measured by a transducer or sensor, such as an encoder or potentiometer, which converts mechanical motion into an electrical signal. The difference between the desired and actual values, known as the error signal, is then processed by a controller. This controller amplifies the error and generates a corrective signal to drive the motor in such a way that the error is minimized or eliminated, thus aligning the output with the input command [3].

There are various control techniques used to regulate the speed and position of servomotors, including PID (Proportional-Integral-Derivative) control, Sliding Mode Control, Fuzzy Logic, Finite-Time Control, Model Predictive Control (MPC), and others [4].

Recent advancements in intelligent control strategies have significantly enhanced the performance of servo systems across various industrial applications. Traditional PID controllers have been the good choice mainly because of their simplicity; however, they are unsuitable for many practical applications due to their inability to adapt and lack of robustness.

Dakheel et al. (2022) compared conventional PID controllers with artificial neural networks (ANN) for the control of a direct current servomotor based on MATLAB simulations. Their findings demonstrated that ANN controllers offered better performance and more realistic true-time simulation of the motor action, thus drawing the advantage of this type of learning-based control instead of fixed-gain PID for the intelligent control [5].

Similarly, Guo et al. (2022) introduced a hybrid positioning control strategy that combines genetic algorithm (GA) with a PID controller on the hydraulic servo systems. The GA was used to optimize PID, increasing response speed and control precision. To suppress the amplitude oscillations caused by the GA algorithms and enhance robustness against external interferences, a Kalman filter was adopted. Simulation results proved that this Kalman-GA-PID controller is better in accuracy and stability compared to the traditional PID controller [6].

Building upon nature-inspired optimization methods, Ji et al. (2023) refined the beetle antennae optimization (BAO) algorithm by incorporating features from particle swarm optimization (PSO), beetle antennae search (BAS), chaos mapping, and adaptive weighting. This enhanced BAO algorithm was applied to industrial robot speed control and showed a 60% improvement in optimization performance over traditional PID, BAS, and PSO controllers. Extensive Simulink/MATLAB simulations demonstrated its effectiveness in handling nonlinearities and external disturbances, making it well-suited for complex industrial environments [7].

Shi et al. (2023) also explored nature-inspired techniques by using an improved artificial bee colony (ABC) algorithm to optimize a fractional PID controller for AC servo systems. Enhancing the algorithm's local search capabilities and guiding the search process resulted in faster response times, reduced overshoot, and improved system robustness compared to conventional control methods [8].

In the context of electro-hydraulic servo systems with parameter uncertainties and external disturbances, Coşkun, Mustafa Yavuz & Itik (2023) developed a model-free, self-learning PID control strategy. By estimating time-varying parameters and nonlinearities using gradient and time-difference methods, the approach dynamically adapts to changing system conditions. The use of intelligent PID (i-PID) compensation and learning control laws ensured accurate trajectory tracking and strong disturbance rejection, as verified through theoretical analysis and simulation experiments [9].

Fuzzy logic and intelligent optimization techniques have gained increasing attention for improving the control performance of electric motors across various platforms and applications.

Torres-Salinas et al. (2022) developed a fuzzy logic-based position controller optimized with genetic algorithms, implemented on a real-time 3D printer platform. The controller was tested under varying loads and trajectory conditions. Results showed improved response times and consistent accuracy, even with changes in initial system conditions, outperforming traditional control techniques previously reported in the literature [10].

In the context of electric vehicles and robotics, Pavithra et al. (2022) focused on enhancing the performance of BLDC and servo motors using fuzzy logic integrated with LabVIEW and Arduino Uno. The system enabled automatic speed control and effective real-time performance visualization. Their findings support the use of fuzzy-based control as a reliable and efficient method for improving motor operation in industrial and modern technological applications [11].

Abdelghany et al. (2023) proposed an optimized PID controller using fuzzy self-tuning combined with the Ant Colony System (ACS) algorithm for DC servo motor control. The controller used a two-stage process, first identifying optimal PID gains with ACS and then applying fuzzy logic for dynamic gain adjustment. MATLAB-Simulink simulations and experimental results demonstrated strong performance in both steady-state and transient conditions, with the controller maintaining optimal gains and effectively handling system disturbances [12].

The primary distinction between the current study and previous research lies in the enhanced control strategy applied to servo motor speed regulation. While earlier works have explored various traditional methods, such as standalone PID controllers or basic fuzzy logic systems, this study introduces a hybrid PID–Fuzzy Logic control system designed to significantly improve overall performance. Specifically, the proposed approach aims to achieve faster settling times, minimize overshoot, and reduce steady-state error to a negligible value. By combining the precise tuning capability of PID control with the adaptability and nonlinearity-handling strengths of fuzzy logic, the system offers a more responsive solution for servo motor speed control.

The proposed controller involves several tuning parameters whose interactions can make the adjustment process more time-consuming than standard methods. Additionally, the hybrid structure increases computational load, which may limit implementation on low-cost hardware. These practical considerations highlight the need for careful tuning and hardware-aware optimization in real applications.

The proposed series-structure hybrid controller combines a conventional feedback loop with an added compensating stage arranged in series, allowing improved tracking accuracy and disturbance rejection compared to standard single-loop designs. Unlike existing hybrid or parallel structures, the series configuration enables independent shaping of the transient response

ARTICLE
JOURNAL`S UNIVERSITY OF BABYLON FOR ENGINEERING SCIENCES (JUBES)
مـــجلـة جـــامعة بـــــابـل للعلــــوم الهندسية
Vol. 33, No.6. \ 2025 ISSN: 2616 - 9916

and robustness characteristics, providing a flexible and effective alternative for servo control applications.

2 Methodology

2.1 Mathematical Model of Servomotor System

A DC servomotor is a type of motor that takes electrical energy and uses it to create precise mechanical movement. The circuit depicted in Figure 1 represents a DC servomotor model. When the armature control voltage Va is applied, it generates an armature current Ia, that flows through a series combination of the armature resistance □□, armature inductance □□, and the motor's rotor.

The rotor shaft, typically illustrated extending to the right, is associated with torque Tm and angular displacement θm. The DC servomotor's transfer function represents the relationship between angular displacement and armature voltage.

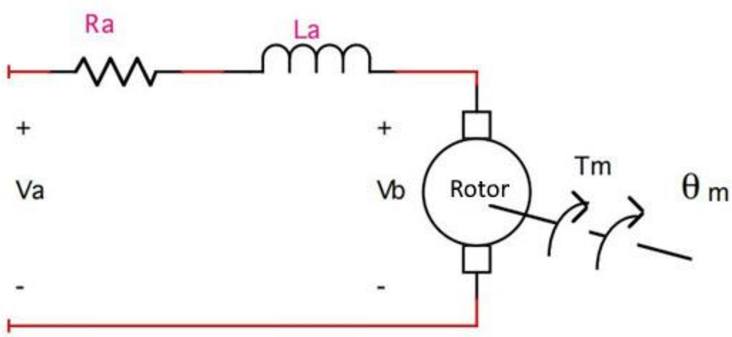

Figure 1: DC Servomotor System

Applying Kirchhoff's Voltage Law (KVL) helps to understand how the electrical system works, as shown in the following equations:

$$Va(s) = Ia(s)Ra + Vb(s) \tag{1}$$

$$Ia(s) = \frac{Va(s) + Vb(s)}{Ra} \tag{2}$$

$$\frac{\theta m(s)}{Va(s)} = \frac{Kt}{JmRas^2 + s(BmRa + KtKe)} \tag{3}$$

Here, Kt and Ke represent the torque and electromotive force constants, respectively, while Jm denotes the rotor's moment of inertia and Bm is the viscous friction constant [13]. The servomotor's parameters are listed in Table1 [14].

Table 1: Parameters of Servomotor

Parameters	Value	Unit
Torque constant Kt	1.1895	N.m/A
Electromotive force constant Ke	1.1895	V/(rad/s)
Armature resistance Ra	3.2645	Ω
Armature inductance La	0.0123	H
Viscous friction constant Bm	0.019	N.m/(rad/s)
Moment of inertia of the rotor Jm	0.0183	Kg.m^2

The DC servomotor state equation will be as follows after putting the parameters in Table 1:

$$\frac{\theta m(s)}{Va(s)} = \frac{1.1895}{0.01829 \times 3.2645 s^2 + s(0.019 \times 3.2645 + 1.1895 \times 1.1895)} \quad (4)$$

The coefficients in the denominator of equation 4 could be simplified as follows:

$$\frac{\theta m(s)}{Va(s)} = \frac{1.1895}{0.05975 s^2 + 1.4769 s} \quad (5)$$

Figure 2 shows a MATLAB Simulink modeling system of the DC servomotor used in this study, including sensors and position control devices for rotor and speed, based on equation 4 and the servomotor's parameters.

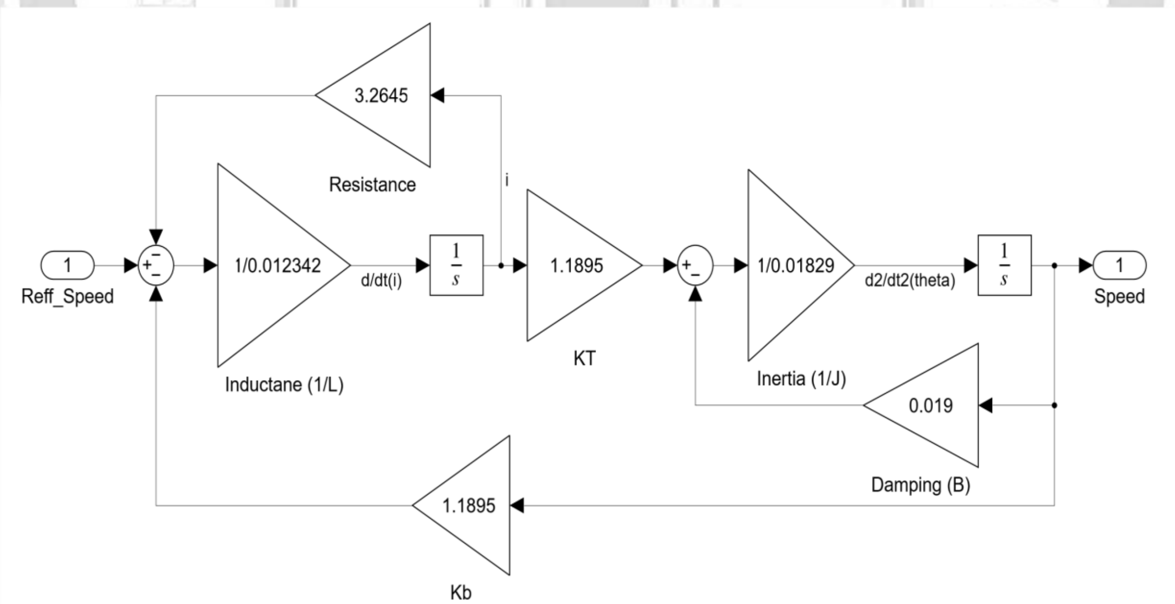

Figure 2: MATLAB Simulink DC servomotor modeling system

2.2 Controller Types for Speed Control

This section provides a comprehensive overview of various control strategies used in this study for speed control of the Dc servo motor system, focusing on Proportional-Integral-Derivative (PID) controllers, Fuzzy Logic Controllers (FLC), and Hybrid PID-Fuzzy Logic Controllers. The discussion presents an overview of each approach, highlighting their structure and implementation considerations.

2.2.1 Proportional-Integral-Derivative (PID) Controller

PID controllers have been widely recognized for their simplicity and effectiveness in industrial control systems since the 1930s [13]. They operate by continuously calculating an error value as the difference between a desired set point and a measured process variable. The controller then attempts to minimize this error by adjusting the process control outputs based on three components: The Proportional (P) term, which responds immediately to current error; the Integral (I) term, which accumulates past errors to eliminate steady-state deviations; and the Derivative (D) term, which anticipates future errors by considering the rate of error change. Together, these components work to improve the accuracy and stability of control systems. The mathematical representation of a PID controller is given by [15]:

$$u(t) = Kp \times e(t) + Ki \times \int e(t)dt + Kd \times de(t)/dt \qquad (6)$$

Where u(t) is the control output, e(t) is the error signal, Kp, Ki, and Kd are the proportional, integral, and derivative gains respectively. The block diagram of speed control of a DC servomotor with a PID controller closed-loop based on Equation (4) and Equation (6) is shown in Figure 3.

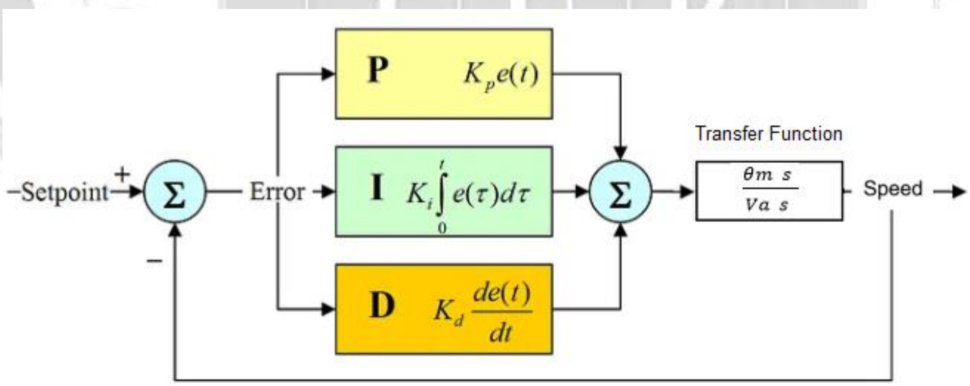

Figure 3: PID closed-loop system [16]

Figure 3 illustrates that the PID controller adjusts the control signal based on the error between the reference input (desired position or speed) and the actual output θm(s), aiming to minimize this error by tuning the proportional (Kp), integral (Ki), and derivative (Kd) gains. Tuning these parameters ensures optimal performance, where Kp addresses the present error, Ki eliminates steady-state error by accumulating past errors, and Kd predicts future errors to

Vol. 33, No.6. \ 2025 ISSN: 2616 - 9916

improve stability. Proper tuning leads to improved transient response, reduced overshoot, and minimal steady-state error, allowing the servomotor to precisely follow the desired input trajectory.

2.2.2 Fuzzy Logic Controllers (FLC)

Fuzzy logic control (FLC) are intelligent controllers that diverge from conventional controllers by utilizing human reasoning and rule-based decision-making. FLC operates with linguistic variables and employs membership functions such as "High," "Medium," and "Low," along with fuzzy rules to determine outputs. This approach allows for more flexible and intuitive control mechanisms, making it applicable across various fields, including control theory and artificial intelligence, as introduced by Zadeh's fuzzy set theory in 1965 [17]. Based on Equation (4), a closed-loop fuzzy logic-based block diagram of a speed control system for a DC servomotor is shown in Figure 4.

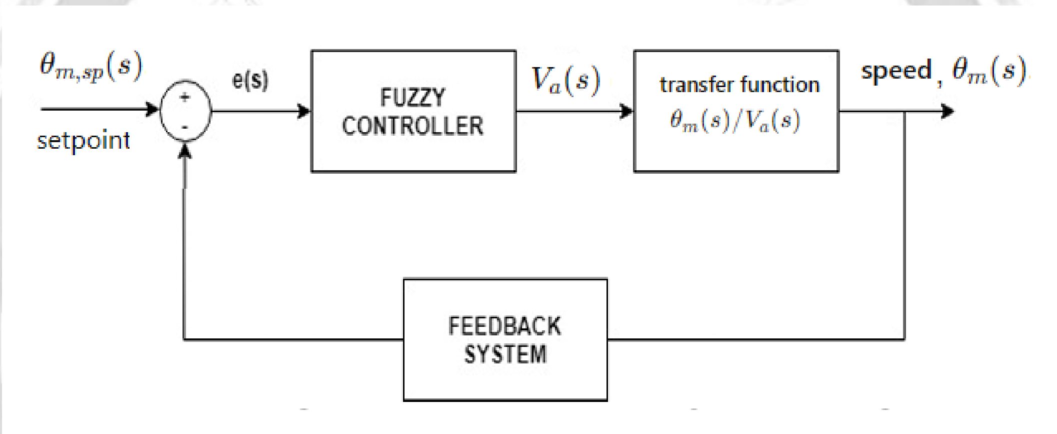

Figure 4: FLC closed-loop system

The Fuzzy Inference System (FIS) Structure for DC Servomotor Speed Control used in MATLAB Simulink is shown in Figure 5.

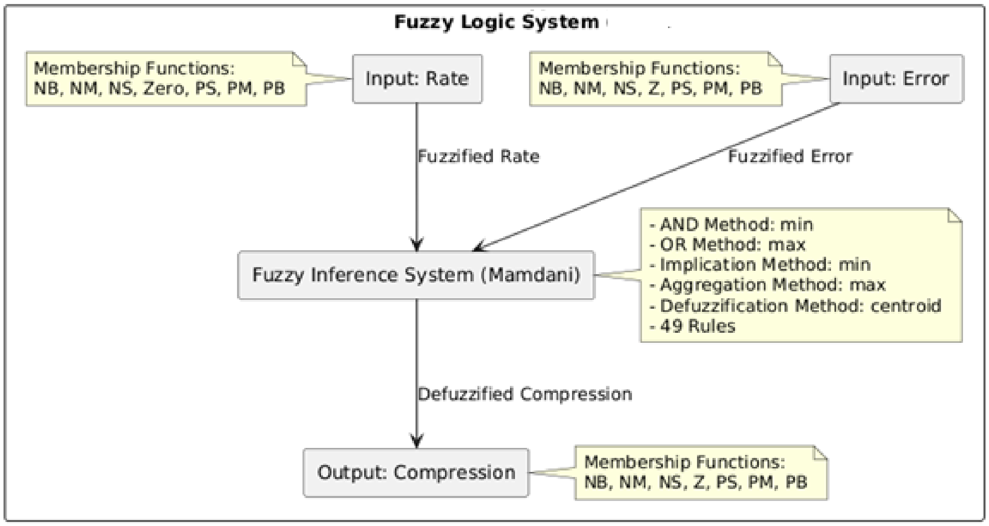

Figure 5: Structure of FLC for DC Servomotor Speed Control

Figure 5 shows a fuzzy logic system designed for DC servomotor speed regulation, where the two input variables are error (E) and the rate of change of error (ΔE), and the output variable is the controller output (speed). Each input and output variable are defined using triangular membership functions, with seven linguistic terms: NB, NM, NS, Zero (Z), PS, PM, and PB. This results in a total of 49 fuzzy rules used in the Mamdani-type fuzzy inference system, where the fuzzified inputs are processed using methods like min (AND), max (OR and aggregation), min (implication), and centroid (defuzzification). The system transforms crisp input values into fuzzy sets, applies the rule base, and then converts the fuzzy output back into a crisp value representing speed control (compression in this diagram). The universe of discourse for the inputs and output are defined as $[-1, 1]$ for both error and rate of error and $[0, 1.5]$ for the controller output. The specific rule and linguistic variables for error, rate of error, and controller output are tabulated in Table 2.

Table 2: (a) Rule Base. (b) Linguistic variables.

(a)

$E/\Delta E$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NM	NS	ZE	ZE
NM	NB	NB	NM	NS	ZE	PS	PS
NS	NB	NM	NS	ZE	PS	PM	PM
ZE	NM	NS	ZE	ZE	ZE	PS	PM
PS	NS	ZE	PS	PS	ZM	PB	PB
PM	ZE	PS	PM	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB
PB	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**

ARTICLE
JOURNAL`S UNIVERSITY OF BABYLON FOR ENGINEERING SCIENCES (JUBES)
مـــجلـة جـــامعة بـــابـل للعلــــوم الهندسية
Vol. 33, No.6. \ 2025 ISSN: 2616 - 9916

(b)

Positive Big	Negative Big	Negative Medium	Negative Small	Zero error	Positive Small	Positive Medium	Positive Big

The choice of seven membership functions was based on a balance between control precision and computational complexity. Preliminary experiments with fewer membership functions (e.g., five or three) resulted in reduced accuracy and slower response during high dynamic changes. The selected configuration allows for finer control resolution and better handling of nonlinearities.

2.2.3 Hybrid PID–FLC controller

A PID-fuzzy controller, also called a hybrid PID-fuzzy controller, is the combination of PID control and fuzzy control. Unlike traditional PID, the hybrid fuzzy PID dynamically adjusts the PID gains using fuzzy inference based on system error and its rate of change. The most common structures used for hybrid controllers are FLCs paralleled with classical PID [18], [19].

In this paper, we propose the hybrid connection of a PID and fuzzy logic controller (PID-FLC) for controlling the speed of DC servo motors; the novelty of this configuration is its series structure. In this system, the PID controller comes first, and its output is then passed into the fuzzy logic controller. This series connection (one after the other) is not the most common setup; many systems combine them differently, like in parallel or switching [20]. This series arrangement provides better handling of control signals. The hybrid controller based on PID-PLC is illustrated in Figure 6. This diagram shows how the fuzzy logic controller and a traditional PID controller can be integrated to achieve precise speed control.

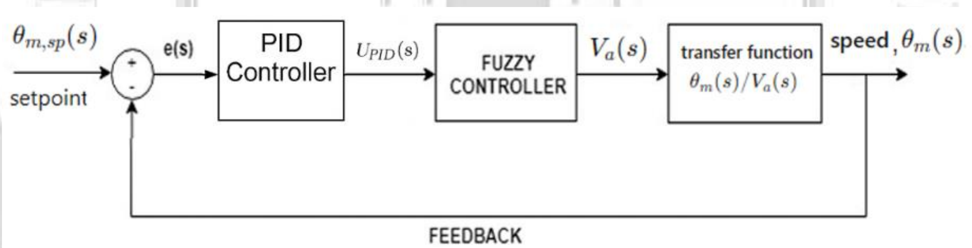

Figure 6: Hybrid PID–Fuzzy controller block integrated with DC servo motor transfer function

The diagram in Figure 6 illustrates a hybrid PID-fuzzy logic system for controlling the speed of a DC motor. The system works by comparing a desired speed setpoint (from the step input) with the actual measured speed of the motor, generating an error signal (e). This error is then processed by a PID controller. The output of the PID controller, along with its rate of change (Δe), serves as input to the fuzzy logic controller to enhance the controller's adaptability to dynamic system behavior. The fuzzy controller acts as a supervisory layer, intelligently tuning the control signal based on its rule set—implementing 49 fuzzy rules using 7×7 membership functions to process the linguistic variables and generate an intelligent voltage command for the DC motor. This hybrid approach aims to combine the straightforward implementation of a PID controller with the adaptive and robust performance of a fuzzy logic system, resulting in a more precise and stable motor speed output.

3 Simulation Model of DC Servomotor Speed Control

To evaluate the performance of the DC servomotor with different controllers, a simulation model was created using MATLAB Simulink, as shown in Figure 7. The figure shows a Simulink model that compares how a DC motor behaves under different control methods. There are four setups: one where the motor runs without any controller, one with a classic PID controller, one with a fuzzy logic controller, and one that combines both PID and fuzzy logic. Each of these setups gets the same reference speed input, and the system measures how well the motor speed matches that input, as well as how much error there is over time.

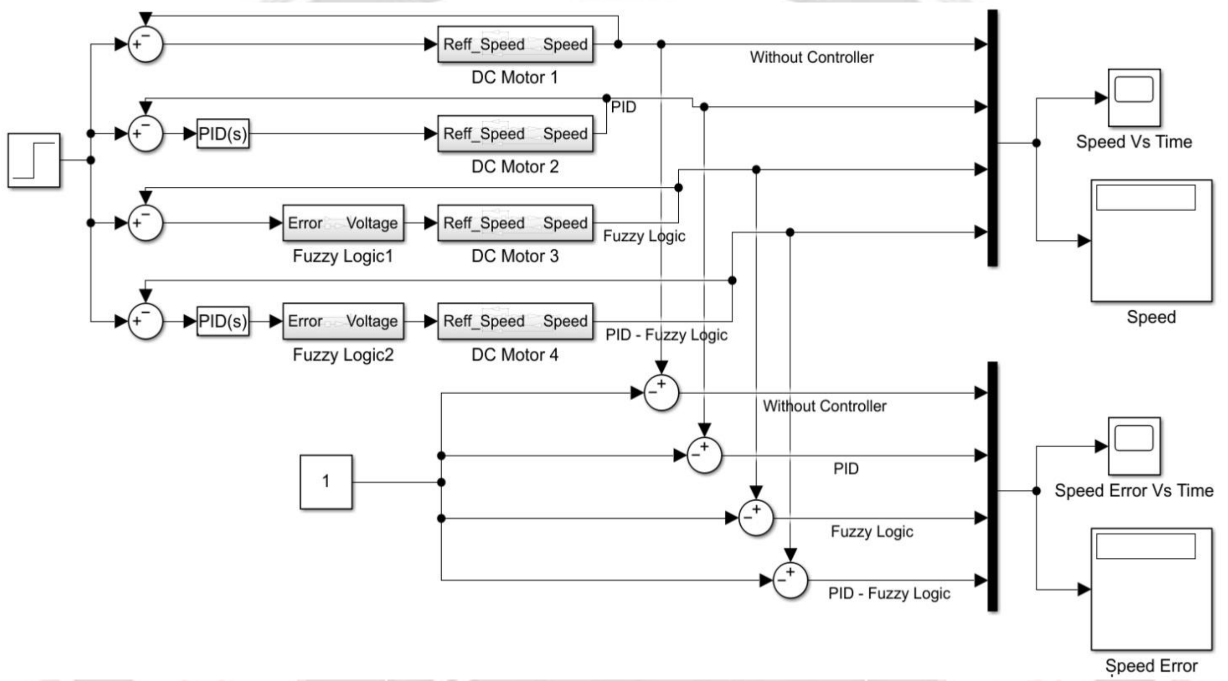

Figure 7: Simulation model of Servomotor

The PID controller parameters were automatically tuned using a transfer function-based tuning method. The resulting values are as follows: the proportional (P) gain is 1.94, the integral (I) gain is 83.74, the derivative (D) gain is 0.003, and the filter coefficient (F) is 5400. For the fuzzy logic controller, the input gain is set to 0.001, while the output gain is 55.

In the first setup, the motor runs without any control, so it just reacts naturally to the input speed. The second uses a PID controller, which tries to keep the motor speed as close as possible to the target by adjusting the input based on the difference between the actual and desired speeds. The third setup uses fuzzy logic, which is a bit smarter in handling uncertainties by using rules to decide how to adjust the motor speed. The last one combines both PID and fuzzy logic to try and get the best of both worlds: the accuracy of PID with the adaptability of fuzzy logic.

Figure 8 shows how a servo motor responds over time using four different control strategies: a standard PID controller, a fuzzy logic controller, a hybrid approach that combines both (PID-fuzzy logic), and a system running with no controller at all. The vertical axis represents the motor speed in per unit (pu), and the horizontal axis shows time in seconds. From what we can see, the hybrid PID-fuzzy logic controller delivers the best performance overall. It reaches the desired speed of 1Pu very quickly and stays steady with almost no fluctuation. The PID controller also does a good job but has a small overshoot and takes a little longer to settle. The fuzzy logic controller takes even longer to stabilize and doesn't quite reach the target speed, although it's still much better than not having a controller at all. The motor, without a controller, clearly demonstrates how poorly the system performs in that scenario. Instead of reaching the 1 pu target, the speed levels are around 0.445 pu, which is less than half the desired value.

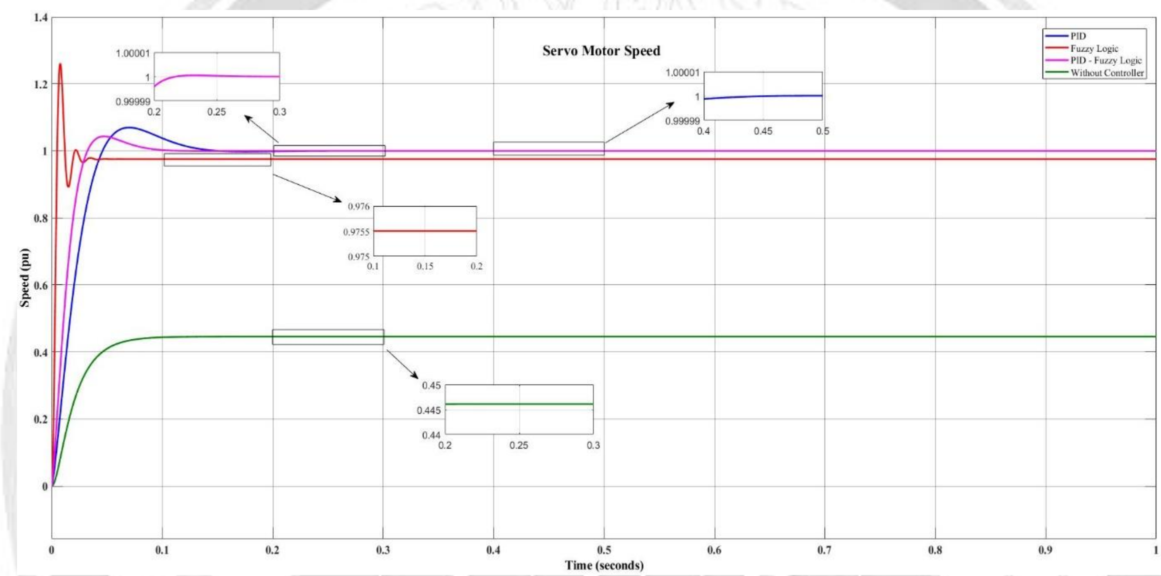

Figure 8: Servomotor Speed for Different Controllers

The graph clearly shows how each control method handles overshoot differently. The Fuzzy Logic controller has the highest overshoot, peaking at around 1.29 pu, which is a 29.221% overshoot above the target speed of 1 pu. The PID controller performs better, with a more moderate overshoot of 6.989%, reaching up to about 1.07 pu before settling. The hybrid PID-fuzzy logic controller stands out as the most precise, with the least overshoot; it approaches the target speed smoothly, with an overshoot of 4.737%. Meanwhile, the uncontrolled system doesn't overshoot at all, but that's because it never even reaches the target speed; it levels off far below at 0.445 pu, showing a major steady-state error of over 55.5%. Overall, the hybrid controller offers the best overshoot performance, combining accuracy with a fast, stable response.

In terms of settling time, the hybrid PID-fuzzy logic controller performs the best, achieving the fastest settling time of 0.23 seconds. The conventional PID controller follows with a slightly slower settling time of 0.26 seconds. The Fuzzy Logic controller alone does not settle at the desired value of 1 pu, instead stabilizing at 0.9755 pu at 0.05 seconds. The uncontrolled

JOURNAL`S UNIVERSITY OF BABYLON FOR ENGINEERING SCIENCES (JUBES)
مـــجلــة جـــامعة بـــــابـل للعلـــــوم الهندسية
Vol. 33, No.6. \ 2025 ISSN: 2616 - 9916

system does not reach the target at all, settling instead at 0.4461 pu at 0.1 seconds, making its settling time negligible.

The zoomed-in sections of the figure show how each controller behaves once the system gets close to the target speed. Both the PID and hybrid controllers stabilize extremely close to 1 pu, showing excellent accuracy. The Fuzzy Logic controller falls a bit short, levelling off just under 0.976 pu. While it's not bad, it doesn't offer the same level of precision. All in all, the hybrid PID-fuzzy logic controller stands out as the most reliable option; it reacts quickly, stays stable, and hits the target speed with almost perfect accuracy.

Figure 9 shows how the speed error of a servo motor changes under different control methods: no controller, a PID controller, a fuzzy logic controller, and a combination of PID with fuzzy logic. Without any controller, the speed error slowly decreases but eventually settles around -0.5539 pu, which means the system struggles to correct the error and responds quite slowly. When using the fuzzy logic controller, the performance improves noticeably—the error settles much closer to zero at about -0.0245 pu. The PID controller does even better, quickly reducing the speed error with a fast recovery. It does have a bit of overshoot, but it eventually settles very close to zero, around 1.221×10^{-14} pu, showing a sharp and effective correction.

The best results come from combining PID with fuzzy logic. This approach almost eliminates any overshoot and reaches near-zero error faster than the others, settling at about -5.662×10^{-15} pu. This reflects superior accuracy and stability in controlling the motor's speed. Overall, the figure clearly shows how integrating fuzzy logic with PID control significantly enhances speed error correction, leading to a system that is both highly precise and fast to stabilize with minimal residual error.

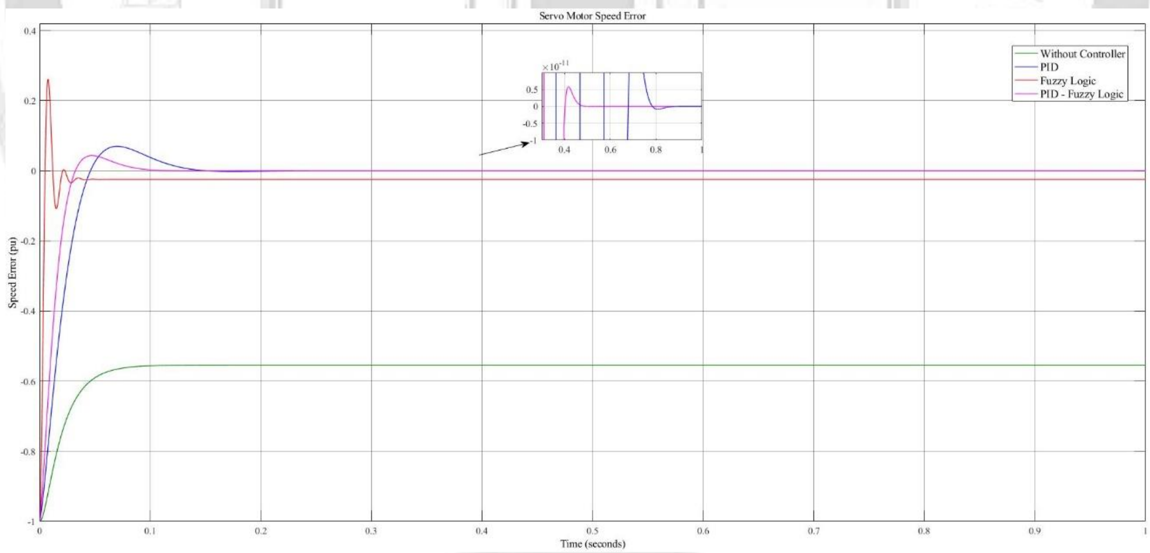

Figure 9: Servomotor Speed Error for Different Controllers

To better understand the differences in settling time, overshoot, and steady-state speed error among all the controllers used in this work, Table 3 presents a comparative summary.

Table 3: Comparison of the proposed method with conventional PID control and the fuzzy logic controller

Control Type	Settling Time (seconds)	Overshoot%	Speed error Pu
Uncontrolled system	0.05	55.5	-0.5539
Conventional PID	0.26	6.989	0.0245
Fuzzy Logic	0.1	29.221	1.221×10^{-14}
Hybrid PID–Fuzzy Logic controller	0.23	4.737	-5.662×10^{-15}

This table compares different control strategies—Uncontrolled, Conventional PID, Fuzzy Logic, and Hybrid PID–Fuzzy Logic—for a DC servomotor. The uncontrolled system shows very poor performance, with a high overshoot (55.5%) and large negative speed error (–0.5539 pu). The conventional PID controller significantly reduces overshoot to 6.989% and improves speed accuracy (0.0245 pu) but requires a longer settling time (0.26 s). The Fuzzy Logic Controller achieves the fastest settling time (0.1 s) but with a relatively higher overshoot (29.221%) and minimal speed error (~0). In contrast, the Hybrid PID–Fuzzy Logic controller offers the most balanced performance: low overshoot (4.737%), improved speed accuracy (~0), and a relatively fast settling time (0.23 s), demonstrating its superior effectiveness in precision speed control.

4 Conclusions

In this work, the performance of the different control techniques (conventional PID control, fuzzy logic, and a hybrid PID with fuzzy logic) was analyzed for regulation of the speed of a DC servomotor. The basis for the analysis procedure was the simple modelling of the DC servo electric motor's electrical and mechanical dynamics upon which the simulation models and performance evaluation were based.

The four control schemes were simulated by MATLAB Simulink. The motor without a controller performed poorly, approximately 0.445 pu, and has experienced a significant steady-state error of more than 55%. The PID controller brought significant improvements (it returned to its setpoint faster, with smaller error) but reduced the overshoot by only a little surplus of about 7%. While the fuzzy logic controller was more adaptable to nonlinear conditions, it generated a larger overshoot and slightly less accurate target speed tracking.

The hybrid PID with a fuzzy logic controller obtained the best performance in all the cases. It arrived at its set-point speed of 1 pu very quickly and exhibited very good stability with low overshoot values. The hybrid approach achieved a speed error steady-state of about -5.662×10^{-15} pu (practically zero), which is less compared to the fuzzy logic controller with a steady-state error of -0.0245 pu and without any controller having a steady-state error of -0.5539 pu.

Thus, the PID-fuzzy logic controller combines the precision control of PID with the flexible and non-linear control of fuzzy logic. Such a hybrid approach is well suited for tasks that need to combine fast, accurate, and smooth motor actions. Further improvements could benefit from adaptive approaches like neural networks or learning methods, including optimization algorithms, which could help to have encouraging results in more complex or time-varying environments. Future work may include real-time and Hardware-in-the-Loop (HIL) experiments to evaluate the controller's performance under realistic operating conditions and to verify its practical feasibility beyond simulation.

Acknowledgements

The authors would like to extend their sincere thanks and gratitude to Salahaddin University-Erbil for the great assistance and support to achieve this research.

References

[1] A. Hughes, and B. Drury, "Electric Motors and Drives: Fundamentals, Types and Applications," 4th ed., Elsevier, 2013.

[2] A. Subramanian, A. Saxena, and J. Coble. "Servomotor Dataset: Modeling Health in Mechanisms with Typically Intermittent Operation," Vol. 15 No.1: Proceedings of the Annual Conference of the PHM Society 2023 , DOI: https://doi.org/10.36001/phmconf.2023.v15i1.3580 (2023).

[3] S. Sahdev. "Electrical Machines," First Edition ed., Vol. 1, (Cambridge University Press, United Kingdom 2018).

[4] S. Autsou, K. Kudelina, T. Vaimann, A. Rassõlkin, and A. Kallaste, "Principles and Methods of Servomotor Control: Comparative Analysis and Applications," Applied Sciences, DOI: https://doi.org/10.3390/app14062579 (2024).

[5] H. Dakheel, Z. Abdullah, N. Shyaa, and S. Shneen, "Simulation model of ANN and PID controller for direct current servo motor by using Matlab/Simulink," Telkomnika (Telecommunication Computing Electronics and Control). DOI: http://doi.org/10.12928/telkomnika.v20i4.23248 (2022).

[6] Y. Guo, X. Zha, Y. Shen, Y. Wang, and G. Chen, "Research on PID Position Control of a Hydraulic Servo System Based on Kalman Genetic Optimization," Vol. 11, Actuators journal. DOI: https://doi.org/10.3390/act11060162 (2022).

[7] Tian Ji, H. Wei, J. Wang, S. Tian, Y. Yao, Yi and S. Hu, "Research into the Beetle Antennae Optimization-Based PID Servo System Control of an Industrial Robot," Vol. 11, Mathematics journal, DOI: https://doi.org/10.3390/math11194066 (2023).

[8] J. Shi, H. Gu, C. Shi, and T. Ma, "Fractional order PID AC servo control system based on artificial bee colony algorithm," 3rd International Conference on Consumer Electronics and Computer Engineering (ICCECE),page (838-841) , DOI: https://doi.org/10.1109/ICCECE58074.2023.10135251(2023).

[9] M. Coşkun, I. Yavuz and S. Mehmet. "Intelligent PID control of an industrial electro-hydraulic system," ISA Transactions journal, DOI: https://doi.org/10.1016/j.isatra.2023.04.005 (2023).

[10] H. Torres-Salinas, J. Rodriguez, E. Cruz-Miguel, L. Ángeles-Hurtado, "Fuzzy Logic and Genetic-Based Algorithm for a Servo Control System," Vol. 13, Micromachines journal, DOI: https://doi.org/10.3390/mi13040586 (2022).

[11] C. Pavithra, W. Melvin, S. Prakash, R. Chinnaa, R. Krishna, "Automated speed control and performance analysis of BLDC and servo motors using fuzzy logic," AIP Conference Proceedings on Innovations and Research in Marine Electrical and Electronics Engineering, DOI: https://doi.org/10.1063/5.0101183(2022).

[12] M. Ghany Mohamed, S. Ossama, and A. Elnady. "Optimum PID Controller with Fuzzy Self-Tuning for DC Servo Motor," Journal of Robotics and Control (JRC), DOI: https://doi.org/10.18196/jrc.v4i4.18676 (2023).

[13] A. Senapati, "Speed Performance Analysis and Control of DC Servomotor Using Linear & Nonlinear Controller," International Journal for Research in Applied Science and Engineering Technology, Vol. 6, Pages 1338-1347, DOI: https://doi.org/10.22214/ijraset.2018.1206 (2018).

[14] L. Mezher, "Characteristics of Servo DC Motor with PID Controller. Journal of Mechanical Engineering Research and Developments," Vol. 44, page (392-400), (2021).

[15] A. Halim and I. Ismail, "Tree physiology optimization on SISO and MIMO PID control tuning. Neural Computing and Applications," Vol. 31, Issue 11, Pages 7571 - 7581, https://doi.org/10.1007/s00521-018-3588-9(2019).

[16] A. Hassan, K. Al-Shamaa, and K. Abdalla, "Comparative Study for DC Motor Speed Control Using PID Controller," International Journal of Engineering and Technology, DOI: https://doi.org/10.21817/ijet/2017/v9i6/170906069 (2017).

[17] J. Liu, "Fuzzy Logic Control. In: Intelligent Control Design and MATLAB Simulation," Fuzzy Logic Control Chapter, pages (33–56), (Springer, Singapore 2018), https://doi.org/10.1007/978-981-10-5263-7_4.

[18] A. Adnan, and M. Hussain, "Optimal Hybrid Fuzzy PID for Pitch Angle Controller in Horizontal Axis Wind Turbines," Iraqi Journal for Electrical and Electronic Engineering. Vol. 21. pages (32-42), DOI: https://doi.org/10.37917/ijeee.21.2.4(2025).

[19] Ha. Zhou, and X. Wang, "PID and Fuzzy Logic Optimization of the Pitch Control of Wind Turbines," Journal of Emerging Investigators. DOI: https://doi.org/10.59720/24-178(2025).

[20] B. Linh and T. Yen, "Design of Hybrid Fuzzy Pid Controller for Dc Servo Motor Part 1: Design of hybrid fuzzy PID controller," Journal of Emerging Investigators. International Journal of Advances in Engineering and Management (IJAEM). Vol. 3, pages (1545-1548) (2021).

ARTICLE
JOURNAL`S UNIVERSITY OF BABYLON FOR ENGINEERING SCIENCES (JUBES)
مـــجلــة جـــــامعة بـــــــابــل للعلــــــوم الهندسية
Vol. 33, No.6. \ 2025 ISSN: 2616 - 9916

التحكم الهجين باستخدام PID والمنطق الضبابي لتحقيق تنظيم دقيق لسرعة محركات السيرفو التيار المستمر

سركار جوهر محمد شريف [1] سالار أحمد قادر [2] عرفان محمد صالح حسن [3]

[1]جامعة صلاح الدين – أربيل ، قسم الهندسة الكهربائية، إقليم كوردستان ، العراق

[3،2] جامعة بوليتكنيك كرميان ، قسم الهندسة الكهرباء و الحاسبات، إقليم كوردستان ، العراق

sarkar.mohammed@su.edu.krd salar.ahmed@gpu.edu.iq erfan.m.salih@gpu.edu.iq

الخلاصة

اعتمدت محركات السيرفو في العديد من التطبيقات الصناعية، مما أدى إلى توجه متزايد نحو تحسين دقة التحكم في سرعتها. وقد أثبتت محركات السيرفو ذات التيار المستمر (DC) المزودة بمشفّرات تغذية راجعة كفاءتها في مختلف تطبيقات التحكم في الحركة ضمن أنظمة الأتمتة الصناعية والإلكترونيات الاستهلاكية. يقدم هذا البحث دراسة للتحكم في سرعة محركات السيرفو ذات التيار المستمر باستخدام طرق تحكم مختلفة، تشمل المتحكم التقليديPID ، والمنطق الضبابي، والمتحكم الهجين–PID المنطق الضبابي. وعلى الرغم من بساطة متحكماتPID ، إلا أنها تعاني من قيود عند التعامل مع اللاخطية وعدم اليقين، مما دفع إلى تطوير استراتيجيات تحكم ذكية وهجينة. تم محاكاة أداء هذه المتحكمات باستخدام برنامج MATLAB/Simulink، حيث أظهرت النتائج أن المتحكم الهجين–PID المنطق الضبابي يتميز بزمن استقرار أقل، وتجاوز بسيط جدًا، وخطأ حالة مستقرة قريب من الصفر. يوفر هذا الأسلوب الهجين دقة نظام التحكم PID ومرونة متحكم المنطق الضبابي، ويقدم طريقة قوية وفعّالة لتحسين استجابة سرعة المحرك في تطبيقات محركات السيرفو.

الكلمات الدالة :محرك السيرفو؛ متحكمPID ؛ التحكم بالمنطق الضبابي؛ التحكم الهجين PID والمنطق الضبابي.