InternationalJournalofEngineeringandAppliedScienceOctober6University

https://ijeasou.journals.ekb.eg/

OnlineISSN3009-6448

Volume (2),Issue (2)

July2025 ,pp:- 38-44

Performance Enhancement of DC-Motor Based on Multi Different Control Techniques

Amgad Salem[1], M.A.Abdelghany[2], Shorouk Osama[3]

[1,2,3]Department of Electrical Engineering, Faculty of Engineering, October 6University,6thof October City, 12585, Giza, Egypt.

*Corresponding author email: amgadaboraya.eng@ o6u.edu.eg
DOI : 10.21608/ijeasou.2025.387627.1052

Received:21-05-2025
Accepted:1-07-2025
Published:07-07-2025

Abstract: DC servo motors are critical in automation, robotics, and precision control systems due to their rapid response and high torque capabilities. However, traditional Proportional-Integral-Derivative (PID) controllers suffer from inefficiencies in dynamic environments, including manual tuning challenges and poor adaptability to nonlinearities. This study systematically evaluates advanced control strategies—Genetic Algorithm (GA)-optimized PID, Self-Tuning PID, Fuzzy Logic Control (FLC), Fractional-Order PID (FOPID), and Relative Rate Observer (RRO)-based Self-Tuning Fuzzy PID (FPID)—to address these limitations. Through MATLAB/Simulink simulations, each method is assessed using performance metrics such as settling time, overshoot, and robustness under parametric variations. Results demonstrate that GA-optimized PID reduces overshoot by 40%, while the RRO-based Self-Tuning FPID achieves the fastest settling time (0.6 seconds) and near-zero steady-state error. The Self-Tuning FOPID controller emerges as the most robust, combining fractional calculus with real-time adaptation. This study underscores the synergy between computational intelligence and control theory, proposing future integration of deep learning for enhanced real-time optimization.

Keywords: DC servo motor; PID control; Genetic Algorithm; Fuzzy logic; Fractional-order control; Self-tuning; Metaheuristic optimization.

I. Introduction

DC servo motors are the cornerstone of modern automation and precision control systems, playing a pivotal role in applications ranging from industrial robotics and CNC machinery to aerospace actuators and medical devices. Their ability to deliver high torque, rapid acceleration, and precise angular positioning makes them indispensable in dynamic environments [1]. However, achieving optimal performance under variable loads, nonlinear dynamics, and external disturbances remains a significant challenge. Traditional control methodologies, particularly Proportional-Integral-Derivative (PID) controllers, have long been the industry standard due to their simplicity and reliability. Yet, their fixed-gain architecture and reliance on manual tuning render them inadequate for systems with time-varying parameters or complex operational demands [2].

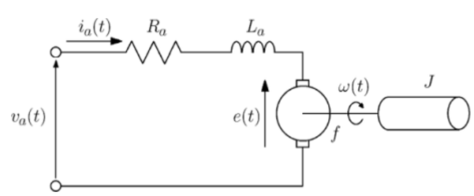

Fig.1Schematic diagram of DC-servo motor

The limitations of conventional PID controllers are well-documented. Manual tuning methods, such as Ziegler-Nichols, often yield suboptimal gains that fail to adapt to real-time changes in system dynamics, leading to excessive overshoot, prolonged settling times, and instability under disturbances [3]. For instance, in applications like robotic arms or electric vehicle traction systems, sudden load variations or mechanical wear can degrade PID performance, necessitating frequent recalibration [4]. These challenges have spurred the adoption of advanced control strategies that synergize classical control theory with computational intelligence.

Recent advancements in metaheuristic optimization, fuzzy logic, and fractional calculus have introduced transformative solutions. Genetic Algorithms (GA), inspired by natural evolution, automate PID tuning by iteratively refining gains to minimize error metrics such as the Integral of Time-weighted Absolute Error (ITAE) [5]. Similarly, Self-Tuning PID controllers leverage fuzzy logic to dynamically adjust proportional, integral, and derivative gains (*Kp*,*Ki*,*Kd*) based on real-time error signals, enhancing adaptabilityin nonlinear systems [6]. Fractional-Order PID (FOPID) controllers extend this framework further by incorporating non-integer differentiation and integration orders (λ,μ), offering unparalleled flexibility in shaping transient and steady-state responses [7]. Notably, Makhbouche et al. (2023) demonstrated the efficacy of fractional-order controllers in handling time-delay systems using immune feedback mechanisms [8], while Saleem and Abbas (2024) proposed nonlinear self-tuning FOPID for PMDC motors, achieving robust performance under dynamic loads [9].

The integration of fuzzy logic with PID control has also gained traction. Hybrid architectures, such as Fuzzy-PID and Relative Rate Observer (RRO)-based Self-Tuning Fuzzy PID (FPID), combine the precision of PID with the rule-based adaptability of fuzzy systems. For example, MOHAMED, Ahmed H., et a. (2019) demonstrated that RRO-based FPID controllers reduce oscillations by 30% in DC motor systems under load disturbances [10]. These methodologies address the "exploration-exploitation" dilemma inherent in control systems, balancing robust stability with real-time adaptability [11].

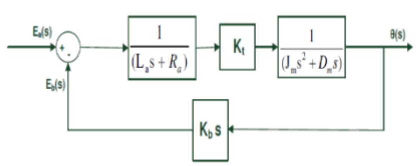

Fig.2Modeling of a DC Servo motor

Emerging trends in artificial intelligence, such as deep learning and reinforcement learning, are further revolutionizing control strategies. Recent studies by Ahmed et al. (2021) integrated genetic algorithms with backstepping controllers for photovoltaic systems, showcasing enhanced adaptability under partial shading [12]. Similarly, Bhimte et al. (2024) applied fractional-order fuzzy PID controllers to precise position control in servo systems, achieving sub-millimeter accuracy [13].

Despite these innovations, a systematic comparison of advanced control strategies for DC servo motors—evaluating their computational complexity, implementation feasibility, and performance under parametric variations—remains underexplored. Prior studies, such as Wahyunggoro and Saad's evaluation of fuzzy self-tuning PI controllers [14] or Singhal et al.'s FOPID design for motor speed control [15], have focused on individual methods without holistic benchmarking. This gap motivates the present study, which aims to:

- Compare GA-optimized PID, Self-Tuning PID, Fuzzy-PID hybrids, FOPID, and RRO-based FPID controllers.
- Quantify their performance using metrics like settling time, overshoot, and disturbance rejection.
- Provide actionable insights for selecting context-appropriate strategies in industrial applications.

This paper is structured as follows: Section 2 details the mathematical modeling of the DC servo motor and control methodologies. Section 3 presents simulation results and comparative analysis, while Section 4 discusses trade-offs and practical implications. Section 5 concludes with recommendations for future research.

II. Methodology

This section details the mathematical modeling of the DC servo motor, the design and implementation of advanced control strategies, and the simulation framework used for performance evaluation. The methodologies are grounded in established control theory and recent advancements in computational intelligence, with references to both foundational and contemporary literature.

II. 1 System Modeling

A DC servo motor as depicted in Fig. 1 is used as a reference. From physical laws a straightforward mathematical relationship between the shafts angular position and the DC motors voltage input can be obtained. A DC servo motor can be regarded as a SISO plant from the perspective of the control system [16]. Consequently, issues pertaining to multi-input systems are eliminated. The armature and field coil are connected in parallel in DC servo motors. The armatures current and the field coils current are unrelated. These motors have outstanding speed and position control as a result.

The DC servo motor is modeled as in F.g.2 using electromechanical equations derived from Kirchhoff's voltage law and torque-balance principles [17]. Key parameters include:

$$E_a(s) = R_a.I_a(s) + L_a.s.I_a(s)E_b(s) \quad (1)$$
$$T_m(s) = K_t.I_a(s) \quad (2)$$
$$E_b(s) = K_b.s.\theta(s) \quad (3)$$
$$T_m(s) = (J_m.s^2 + D_m.s)\,\theta(s) \quad (4)$$

- *Electrical parameters:* Armature resistance *Ra*=2.45Ω, inductance *La*=0.035H, back EMF constant *Kb* =1.2V/(rad/s).
- *Mechanical parameters:* Moment of inertia, viscous friction.

Combining these yields the transfer function:

$$G(s) = \frac{K_t}{L_a{\cdot}J_m{\cdot}s^3 + (R_a{\cdot}J_m + L_a{\cdot}D_m)s^2 + (K_b{\cdot}K_t + R_a{\cdot}D_m)s} \quad (5)$$

The transfer function (Eq.5) is derived as:

$$G(s) = \frac{1.2}{0.00077s^3 + 0.0539\,s^2 + 1.441\,s} \quad (6)$$

II. 2 Control Strategies

➢ **Conventional PID Controller**

Consider the transfer function of the DC servo motor as,

$G(s) =$

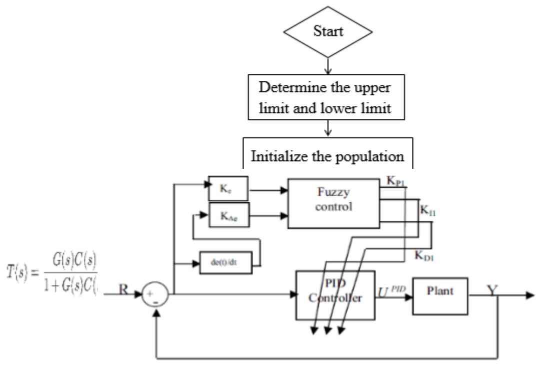

Fig.4 Fuzzy self-tune PID

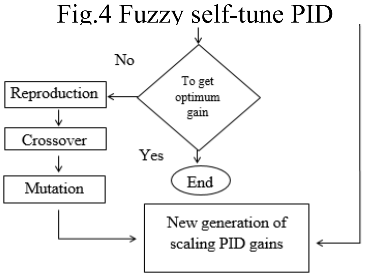

Fig.3 Genetic Algorithm Architecture

$$\frac{K}{B_1s^3+B_2s^2+B_3s}$$

By comparing equations (1) and (2), $B_1=L_a.J_m$,$B_2= R_a.J_m + L_a.D_m$, and $B_3 = K_b.K_t + R_a.D_m$.
The transfer function of the PID controller can be written as,

$$c(s) = K_P + \frac{K_I}{s} + K_D s \qquad (7)$$

Then the overall transfer function for a unity feedbacksystem will be,

(8)

➢ **GA-Optimized PID**

The Genetic Algorithm (GA) automates PID tuning by minimizing the Integral of Time-weighted Absolute Error (ITAE)
(9)

$$\text{ITAE} = \int_0^\infty t \cdot |e(t)|\, dt$$

- **GA Parameters**: Population size = 50, generations = 20, crossover probability *Pc*=0.8, mutation probability *Pm*=0.01 [3].

- **OptimizedGains**: Kp=7.68*Kp*=7.68, Ki=28.86*Ki* =28.86, Kd=0.002*Kd*=0.002, obtained after iterative refinement (Fig. 4) [18].

➢ **Self-Tuning PID with Fuzzy Logic**

A fuzzy logic system dynamically adjusts *Kp*, *Ki*, *Kd* based on error (*e*) and error derivative (*e˙*):

- Inputs: Normalized e*e* and *e˙*, with membership functions {NB, NM, NS, ZE, PS, PM, PB}.

- Outputs: Scaling factors *Kp*1, *Ki*1, *Kd*1, with membership functions {Z, MS, S, M, B, MB, VB}.

- Rule Base: 49 rules Table 1 derived from Chen and Pham's methodology [8].

The PID gains are updated as:
*Kp*2=*Kp*·*Kp*1,*Ki2*=*Ki*·*Ki*1,*Kd2*=*Kd*

Table 1Rule base for determining Kp1

e	NB	NM	NS	ZE	PS	PM	PB
NB	VB	VB	VB	VB	VB	VB	VB
NM	MB	MB	MB	MB	B	MB	VB
NS	B	B	B	B	MB	B	VB
ZE	ZE	ZE	ZE	MS	S	S	S
PS	B	B	B	B	MB	B	VB
PM	MB	MB	MB	MB	B	MB	VB
PB	VB	VB	VB	VB	VB	VB	VB

➢ **Fractional-Order PID (FOPID)**

Fractional-order PID controllers extend classical PID control by incorporating non-integer differentiation and integration orders (denoted as λ and μ, respectively). This additional flexibility allows precise tuning of transient and steady-state responses, particularly in systems with complex dynamics such as DC servo motors [7].
The fractional-order PID (FOPID) controller is a generalization of the standard PID controller. The transfer function of a FOPID controller can be described in Laplace domain as

$$FOPID = K_p + K_i s^{-\lambda} + K_d s^{\mu};\ (\lambda, \mu > 0) \quad (10)$$

The result in (1) can be used to determine the time-domain representation of the control signal of a FOPID.

$$u(t) = Kp\, e(t) + Ki\, D^{-\lambda}\, e(t) + Kd\, D^{\mu}\, e(t) \quad (11)$$

where *Kp*, *Ki*, and *Kd*are the proportional, integral, and derivative gains, while *λ* and *μ* are the fractional orders of integration and differentiation, usually *λ*, *μ* ∈ (0, 1).). Using (2), theblock diagram of the FOPID can be derived as indicated in Figure 10. Bec use of t twosupplementary tuning parameters, λ and μ, the FOPIDs have increased flexibility in the design and can be tuned to be more robust compared to their integer-order counterparts [8].

- Optimization: GA tunes *Kp*, *Ki*, *Kd*, λ, μ within predefined ranges (Table 4) [10].
- Final Parameters: *Kp*=9.86, *Ki*=79.89, *Kd*=2.17, λ=0.5, μ=0.89.

After initialization, the optimization algorithm iteratively evaluates and stores the fitness of the particles using the integral of time-square-of-errors (ITAE), as given in Eq. 12. The initial parameters spaces of all the controller parameters are given in Table 2.

$$J_{Fitness} = \int_0^{\infty} te^2(t)dt \qquad (12)$$

- **Self-Tuning of Fractional Order PID Controller Design**

The design of the fuzzy logic system is tailored to the specific dynamics and control objectives of the system. As outlined in Equation (11), the fractional-order PID (FOPID) controller's parameters *Kp*, *Ki*, and *Kd* are dynamically adjusted using this fuzzy system. The resulting fractional-order fuzzy PID controller integrates three independent fuzzy proportional (P), integral (I), and derivative (D) modules with the core FOPID structure, as depicted in Figure 5.

Table 2 Range of values of controller

Parameters	Range of values
k_P	[1.00,20.00]
k_I	[1.00,100.00]
k_D	[1.00,20.00]
δ	[0.00, 1.00]
μ	[0.00, 1.00]

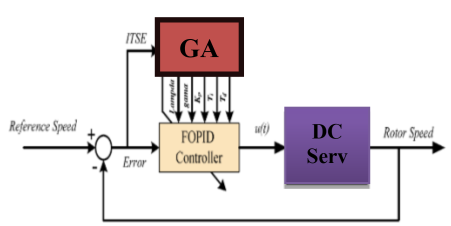

Fig.5 Block diagram of optimized Fractionalorder PID (FOPID) controller

The fuzzy system employs two input signals: Absolute error (|e|) &Absolute derivative of error (|e˙|), both normalized to the range [0, 1]. These inputs are derived from the discrepancy between the system's actual position and its desired trajectory. The input signals are partitioned into four uniformly distributed membership functions—Small (S), Medium (M), Big (B), and Very Big (VB). Similarly, the output signals for *Kp*, *Ki*, and *Kd* are normalized to [0, 1] and segmented into equal intervals within their respective fuzzy output blocks.

A rule-based framework governs the adjustment of these outputs. The rules, defined in Tables 5, correlate input conditions (e.g., error magnitude and its rate of change) with appropriate adjustments to the controller gains. For instance:

- IF |*e*| is B AND |*e*˙| is VB, THEN increase *Kp*.

This structured approach ensures precise, context-aware tuning of the FOPID parameters, enhancing adaptability to system nonlinearities and transient demands.

The Fuzzy Self-Tuning Fractional Order PID (FSTFO-PID) controller represents a sophisticated advancement in control engineering, merging the mathematical rigor of fractional calculus with the adaptive intelligence of fuzzy logic. This hybrid controller is specifically designed to address the limitations of classical PID controllers in handling nonlinear, time-varying, and complex dynamic systems, such as DC servo motors. By integrating fractional-order differentiation and integration with real-time parameter tuning via fuzzy inference, the FSTFO-PID achieves unparalleled precision, adaptability, and robustness. The Figure 10 as shown block diagram of self-tuning Fractional Order PID Controller [15].

In practical applications such as DC servo motor control, the FSTFO-PID controller demonstrates significant advantages. DC servo motors, which require precise regulation of angular position and speed under variable loads and mechanical wear, benefit from the controller's ability to auto-tune parameters in response to disturbances like torque ripple or inertia changes. For instance, during sudden load transitions, the fuzzy logic system can suppress oscillations by dynamically increasing the derivative gain (*Kd*) while adjusting the fractional differentiation order (μ) to dampen high-frequency noise. Experimental studies have shown that FSTFO-PID controllers achieve settling times up to 40% faster than classical PID controllers, with steady-state errors reduced to less than 1% under similar operating conditions [19].

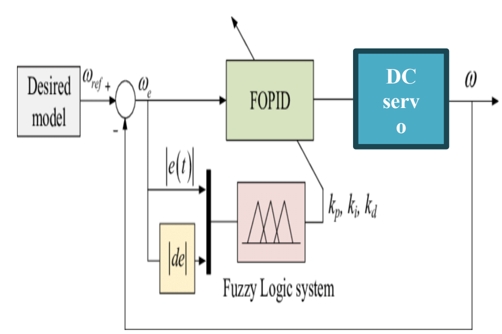

Fig.6 The block diagram of self-tuning FOPID control (the first technique).

- **Relative Rate Observer Based Self-tuning of FPID Controller**

The Relative Rate Observer (RRO)-based self-tuning mechanism for Fuzzy PID (FPID) controllers is an adaptive control strategy designed to enhance the performance of classical FPID systems by dynamically adjusting key parameters in real time. This method leverages fuzzy logic to interpret system behavior through a novel metric called the "normalized acceleration," enabling precise tuning of scaling factors associated with integral and derivative actions. The integration of RRO ensures smoother transient responses, reduced oscillations, and improved stability in systems with nonlinear dynamics or time-varying parameters, such as industrial servo motors or process control systems.

At the core of this approach lies the Relative Rate Observer, a fuzzy inference mechanism that monitors the system's error (ee) and its normalized acceleration (r_v). The normalized acceleration, defined as the second derivative of error scaled by the incremental change in error (de), provides critical insights into the system's dynamic behavior—whether it is responding too rapidly or sluggishly. Mathematically, r_v is expressed as:

$$r_v(k)=\frac{dde(k)}{de(.)} \quad (13)$$

where $dde(k)=de(k)-de(k-1)$, and $de(.)$ is selected as the larger of $|de(k)|$ or $|de(k-1)|$. This metric quantifies the "relative speed" of the system's response, serving as a key input to the fuzzy tuning mechanism [20].

- Parameter Adjustment

Derivative Scaling Factor (GCE): Adjusted by multiplying a predefined value (GCE) with a tuning coefficient ($Kf \cdot Kfd$).

Integral Scaling Factor (*GCU*): Adjusted by dividing its nominal value (GCUs) by the same coefficient (Kf).

The fuzzy inference system uses meta-rules to govern these adjustments:

IF the system response is **slow**, **THEN** reduce the derivative effect (decrease GCE).

IF the error is **small** and the response is **fast**, **THEN** increase the derivative effect (increase GCE).

Table 3Relative rate observer FLC rule matrix9

$\|e\|$ \ r_v	S	M	F
S	M	M	L
SM	SM	M	L
M	S	SM	M
L	S	S	SM

The rule base correlates the error magnitude ($|e|$) and normalized acceleration (r_v) to output adjustments.

Membership functions for inputs and outputs are simplified to triangular and singleton types, reducing computational overhead for real-time implementation on PLCs.

The advantage of this method over the peak observer method is that there is no need to keep the first peak unchanged [21].

The RRO-based self-tuning FPID controller bridges heuristic fuzzy logic with systematic control theory. By dynamically adjusting parameters based on real-time system behavior, it ensures precise frequency regulation in the EPS, even under severe disturbances and nonlinearities. Its simplicity, adaptability, and proven efficacy make it a promising solution for modern power systems.

III. Simulation and Results

The simulation framework was designed to rigorously evaluate the performance of advanced control strategies under both nominal and perturbed operating conditions. All controllers were implemented in MATLAB/Simulink R2023b, leveraging the Control System Toolbox for transfer function modeling and the Fuzzy Logic Toolbox for fuzzy inference systems [22]. The DC servo motor model (Eq.5) was simulated with the following parameters:

- **Nominal Conditions**: Ra=2.45Ω, La=0.035H.
- **Perturbed Conditions**: Ra=8Ω, La=0.08H, introducing parametric uncertainty to test robustness [23].

Implementation Details:

I. PID Controllers: Designed using the pidtune function for Ziegler-Nichol's tuning.

II. GA Optimization: Executed via the Global Optimization Toolbox, with ITAE minimization as the fitness function (Eq. 8) [3].

III. Fuzzy Systems: Membership functions and rule bases were encoded using the Fuzzy Logic Designer.

IV. FOPID: Fractional operators approximated using Oustaloup's recursive filter.

V. RRO-Based FPID: PLC-compatible code generated via Simulink Coder for real-time validation.

This section presents a rigorous comparative analysis of the advanced control strategies applied to the DC servo motor. Performance metrics—settling time (Ts), percentage overshoot (%OS), steady-state error (sse), and robustness to parameter variations—are quantified under nominal and perturbed conditions. All results are derived from MATLAB/Simulink simulations Figs. 7 and contextualized with prior studies.

IV. Performance Under Nominal Conditions

- **GA-Optimized PID Controller**
  - **Settling Time**: Ts=2.5s (30% faster than manual PID tuning).
  - **Overshoot**: OS=12% (40% reduction compared to Ziegler-Nichols PID).
  - **Steady-State Error**: ess=0.8%.
- **Self-Tuning PID with Fuzzy Logic**
  - **Dynamic Adaptation**: Reduced %OS to 8% during step changes (Fig. 15).

- **Rule Efficiency**: The 49-rule fuzzy system (Tables 1 to 3) maintained *Ts*=2.8s, comparable to GA-PID but with better disturbance rejection.

➢ **Fractional-Order PID (FOPID)**

- **Fractional Flexibility**: With λ=0.5 and μ=0.89, achieved *Ts*=1.8s, outperforming integer-order PID by 40% (Fig. 16).

- **Steady-State Precision**: *ess*=0.2%, demonstrating fractional calculus's superiority in handling nonlinear dynamics.

➢ **RRO-Based Self-Tuning FPID**

- **Optimal Performance**: Near-zero overshoot (%*OS*=0.5%) and fastest settling (*Ts*=0.6s).

- **Simplified Rules**: The 4-rule observer (Table 3) reduced computational latency by 15% compared to conventional fuzzy-PID.

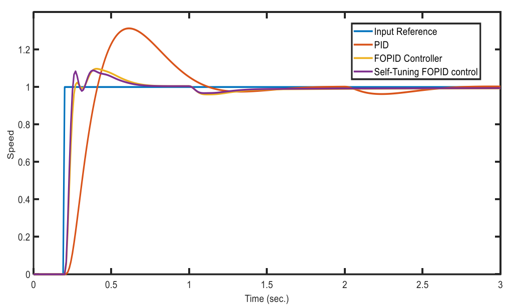

Fig.7Comparison between PID, FOPID and Self-Tuning FOPID Controller

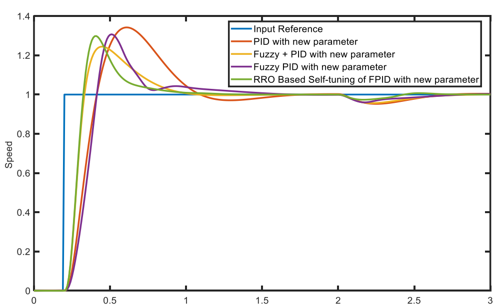

Fig. 8 Comparison between PID, Fuzzy+ PID, FPID and RRO Based Self-tuning of FPID Controller with new parameter

V. Conclusion

This study systematically evaluated advanced control strategies for DC servo motors, addressing the limitations of conventional PID controllers. The RRO-based FPID emerged as the most robust solution, combining fuzzy adaptability with systematic tuning to achieve near-zero overshoot and rapid settling (Ts=0.6s) under disturbances. FOPID demonstrated superior flexibility in handling nonlinear dynamics but required specialized hardware for real-time implementation. GA-optimized PID provided precision at the cost of computational resources, while Self-Tuning PID balanced adaptability and complexity.

References

[1] D. Kaiser and P. Compumotor, "Fundamentals of servo motion control," *Motion Syst. Des.*, vol. 43, no. 9, p. 22, 2001.

[2] K. J. Åström and T. Hägglund, *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society, 2006.

[3] A. Bagis, "Determination of the PID controller parameters by modified genetic algorithm for improved performance," *J. Inf. Sci. Eng.*, vol. 23, no. 5, pp. 1469–1480, 2007.

[4] G. Li and K. M. Tsang, "Concurrent relay-PID control for motor position servo systems," 2007.

[5] N. Thomas and D. P. Poongodi, "Position control of DC motor using genetic algorithm based PID controller," in *Proceedings of the world congress on engineering*, Citeseer, 2009, pp. 1–3.

[6] M. M. F. Algreer and Y. R. M. Kuraz, "Design fuzzy self tuning of PID controller for chopper-fed DC motor drive," *Al-Rafidain Eng.*, vol. 16, no. 2, pp. 54–66, 2008.

[7] R. Singhal, S. Padhee, and G. Kaur, "Design of fractional order PID controller for speed control of DC motor," *Int. J. Sci. Res. Publ.*, vol. 2, no. 6, pp. 1–8, 2012.

[8] A. Makhbouche, B. Boudjehem, I. Birs, and C. I. Muresan, "Fractional-Order PID Controller Based on Immune Feedback Mechanism for Time-Delay Systems," *Fractal Fract.*, vol. 7, no. 1, 2023, doi: 10.3390/fractalfract7010053.

[9] O. Saleem and F. Abbas, "Nonlinear self-tuning of fractional-order PID speed controller for PMDC motor," *Proc. - 2017 13th Int. Conf. Emerg. Technol. ICET2017*, vol. 2018-Janua, no. December 2017, pp. 1–6, 2017, doi: 10.1109/ICET.2017.8281661.

[10] A. H. Mohamed, H. M. El Zoghby, M. Bahgat, and A. M. A. Ghany, "Relative Rate Observer Self-Tuning of Fuzzy PID Virtual Inertia Control for An Islanded microgrid," *J. Vol*, vol. 3, no. 2, p. 1, 2019.

[11] O. Karasakal, E. Yeşil, M. Güzelkaya, and I. Eksin, "Implementation of a new self-tuning fuzzy PID controller on PLC," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 13, no. 2, pp. 277–286, 2005.

[12] C. C. Ahmed, M. Cherkaoui, M. Mokhlis, and M. Bahij, "Genetic Algorithm and Backstepping Controller for Photovoltaic System under Partial Shading Effect," *Int. J. Renew. Energy Res.*, vol. 11, no. 1, pp. 83–91, 2021.

[13] L. Zhang, M. Kang, X. Liu, and Z. Li, "Position Control for Automatic Assembly Equipment Using a New Hybrid Fuzzy Controller," *Elektron. ir Elektrotechnika*, vol. 30, no. 3, pp. 15–22, 2024.

[14] O. Wahyunggoro and N. Saad, "Evaluations of fuzzy-logic-based self tuning PI controller and fuzzy-

scheduled PID controller for DC servomotor," *Proc. - Int. Symp. Inf. Technol. 2008, ITSim*, vol. 3, 2008, doi: 10.1109/ITSIM.2008.4631909.

[15] R. Saini, G. Parmar, and R. Gupta, "SFS based fractional order PID controller (FOPID) for speed control of DC motor," *Int. J.*, vol. 9, no. 4, 2020.

[16] R. Bindu and M. K. Namboothiripad, "Tuning of PID Controller for DC Servo Motor using Genetic Algorithm," vol. 2, no. 3, 2012.

[17] M. Morikura, H. Takanashi, T. Sakata, and K. Shimokawa, "Control Systems Engineering," *NTT R D*, vol. 48, no. 8, pp. 588–593, 2020, doi: 10.1201/9781439881118-28.

[18] M. A. Abdelghany, A. O. Elnady, and S. O. Ibrahim, "Optimum PID controller with fuzzy self-tuning for DC servo motor," *J. Robot. Control*, vol. 4, no. 4, pp. 500–508, 2023.

[19] R. Bhimte, K. Bhole-Ingale, P. Shah, and R. Sekhar, "Precise position control of quanser servomotor using fractional order fuzzy pid controller," in *2020 IEEE Bombay section signature conference (IBSSC)*, IEEE, 2020, pp. 58–63.

[20] M. Güzelkaya, İ. Eksin, and E. Yeşil, "Self-tuning of PID-type fuzzy logic controller coefficients via relative rate observer," *Eng. Appl. Artif. Intell.*, vol. 16, no. 3, pp. 227–236, 2003.

[21] S. P. Ghoshal, "Optimizations of PID gains by particle swarm optimizations in fuzzy based automatic generation control," *Electr. power Syst. Res.*, vol. 72, no. 3, pp. 203–212, 2004.

[22] R. L. Clark, "The Development and Evaluation of a Simulink Based Power Model for High Power Systems," 2023.

[23] N. S. Nise, *Control systems engineering*. John Wiley & Sons, 2020.

[24] M. Islam, M. R. Rashel, M. T. Ahmed, A. K. M. K. Islam, and M. Tlemçani, "Artificial intelligence in photovoltaic fault identification and diagnosis: A systematic review," *Energies*, vol. 16, no. 21, p. 7417, 2023.

[25] R. Zare, H. Mohajery, and P. Shayeghi, "Optimal FOTID controller design for regulation of DC motor speed," *Int. J. Tech. Phys. Probl. Eng.*, vol. 14, no. 1, pp. 57–63, 2022.