Optimum PID Controller with Fuzzy Self-Tuning for DC Servo Motor

M. A. Abdelghany [1], Abdelrady Okasha Elnady [2*], Shorouk Ossama Ibrahim [3]
[1, 3] Electrical Department, Faculty of Engineering, October 6 University, Egypt
[2] Mechatronics Department, Faculty of Engineering, October 6 University, Egypt
Email: [1] m.a.abdelghany.eng@o6u.edu.eg, [2] rady_nady.eng@o6u.edu.eg, [3] Shorouk.eng@o6u.edu.eg
*Corresponding Author

***Abstract*—DC motors are simple and controllable, making them a popular choice for various applications. However, the speed and load characteristics of DC motors can change, making it difficult to control them effectively. This paper proposes an optimum PID controller with fuzzy self-tuning for DC servo motors. The controller uses two steps to adjust the PID gains: The ACS algorithm is employed to identify the optimal PID gains in the first step. A fuzzy logic (FLC) controller is employed in the second stage to further fine-tune the gains. The FLC considers two cost functions: the first function is the sum of the squares of the error between the controlled output and reference input. The second function is a mathematical expression that specifies the required characteristics of the system response. The fuzzy self-tune then uses a set of rules to adjust the PID gains in response to changes in the system. The rules are based on the two cost functions designed to maintain the optimum PID gains for various operating settings. The outcomes of the two functions are: Kp = 5.2381, Ki = 7.0427, and Kd = 0.49468, with rising time = 0.2503, overshoot = 2.5079, and settling time = 10.4824 in the first cost function. The second cost function outcomes are Kp = 8.1381; Ki = 8.6427; and Kd = 0.49468. The FST-PID controller's performance is evaluated using Matlab-Simulink. The proposed controller was tested on a DC servo motor, and the results showed good performance in both steady-state and transient responses. The controller also maintained the optimum PID gains in the event of changes or disturbances. So, the motor's speed can effectively control under a variety of conditions.**

Keywords—*DC Servo Motor; PID Controller; Ant ColonySystem; Fuzzy Self Tuning.*

I. Introduction

Due to its simplicity and controllability, the DC motor is a desirable for various applications that demand changing speed and load characteristics. It is a typical actuator used in various mechanical systems and commercial products, including rolling mills, traction devices, industrial robots, and educational robots [1-5]. These advantages motivate researchers to create various algorithms and strategies for control of DC motor speed and position [6-11].

A tracking and regulating constrained optimal PID parameters (K_P, K_I, K_D) for DC motor speed controller design is presented [12-15]. A great-performance motor drive system often needs to have a great dynamic response to implement angle and speed tracking. Furthermore, the motor drive should respond to real-time load changes [16, 17].

Scientists have developed multiple control system mechanisms to acquire the optimum output signals for position and velocity control. The control system can either be a traditional PID control or fuzzy-logic control (FLC) with set-point tracking performance that may be self-tuned [18].

Gain adjustments in PID controllers are fast and simple, resulting in a stable response. However, PID control system operates inefficiently when some variables are altered, or the torque burden is changed, resulting in an insufficient response [22-26].

FLC mimics the human foresight method by employing linguistic ideology rather than numerical calculations. Through their linguistic expression, fuzzy principles generate an output value. FLC offers an optimum response when torque loads change; however, the system architecture causes a sluggish response [27-31].

The benefits of the PID control system (fast tuning) and FLC (a better-executed approach to complex systems) are combined to produce a "fuzzy self-tuning PID controller" that produces a best response. FLC functions as the system's supervisor and configures the PID controller's coefficients [32-35].

The PID gains; K_P, K_I, and K_D can be self-adjusted online after controlling the system's output [36-39]. In PID control, fine-tuning a PID controller's parameters is crucial. Ziegler and Nichols introduced the well-known Ziegler-Nichols approach to fine-adjust a PID controller's coefficients. Despite being straightforward, this tuning technique can sometimes be relied upon to work. In many industrial facilities, it is typically difficult to obtain optimum or near-optimum PID parameters using the Ziegler-Nichols formula [40-41].

Using fuzzy logic control (FLC), the gains (K_{P1}, K_{I1}, and K_{D1}) were outputs from the controller design, while the error and change of error were inputs to the self-adjusted [42-44]. The FLC is an addition to the traditional PID controller that allows the PID controller's settings to be adjusted online in response to changes in the signals' errors [45-47]. To satisfy the operating ranges and make them more generic, the controller also includes scaling gain inputs for the error and change of error (K_e, $K_{\Delta e}$) [48-50].

Analyzing, classifying, or enhancing existing systems or data necessitates the utilization of an algorithm for optimization to choose the optimal solution from all possible solutions. The most efficient optimization algorithm is the Ant colony system (ACS). Using this algorithm, the controller will produce the best and optimum outcome for the specified control system while being evaluated repeatedly when Ant Colony System (ACS) is used for tweaking [51-55]. In all optimization problems, the aim is to get at least one fitness function or more than one fitness function to obtain the optimum key that achieve the complementary statuses [56-60].

As seen from the previous presentation, many methodologies and algorithms exist to improve the PID gains. This article is a contribution for improving the PID gains. It proposes a fuzzy self-adjusted optimal PID controller for a DC servo motor. The optimal gains; K_P, K_I, and K_D are obtained using the ACS algorithm. Using FLC, the best gains (K_{P1}, K_{I1}, and K_{D1}) considering two cost functions is obtained. The first function to be minimized is the squared error sum between the reference input and the controlled output. The "rise time", the "maximum overshoot", the "settling time" and "steady-state error" with the desired ones are formulated and considered as the second function to be minimized. Then the Fuzzy Self Tune was used to overcome any change or neutrality and thus maintain Optimum PID gains.

The effectiveness of the suggested controllers is evaluated through 4 tests. In the first test, the PID parameters were obtained using the two cost functions and then compared to the parameters obtained by trial and error.

In test 2, the PID parameters were obtained from the two cost functions while decreasing the motor speed. These parameters were then compared to each other to determine which set of parameters produced the best results. In this test, the velocity increased in the tenth second. In test 3, the best PID parameters were obtained using the first cost function. These parameters were then self-tuned using the center-of-gravity defuzzification technique. In test 4, the PID parameters were obtained by considering the uncertainty caused by a 30% change in Ra or La. Simulations showed that the PID parameters obtained using two cost functions outperformed those obtained by trial and error. The first cost function produced better results than the second cost function. The proposed algorithm can achieve optimal tuning parameter values when the ACS is used for tuning.

II. DC Motor Modeling

As a reference, consider the DC-servo motor shown in Fig. 1. A simple mathematical link between the shaft position and the voltage supplied to the DC motor can be drawn using physical rules. DC servo motors can be viewed as a single input, single output (SISO) drives from the perspective of the control system. The field coil and armature are parallel in DC servo motors. The armature's current is unrelated to the current in the field coil. These motors have exceptional control of velocity and position as a result.

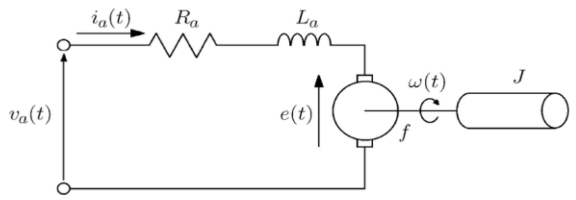

Fig. 1. "DC-Servo Motor" schematic diagram

The electrical, mechanical, and electro-mechanical equations are the three main equations that make up a DC motor's modeling [52]. Apply Kirchhoff's Voltage Law (KVL) to derive the electrical equation for a DC motor system, which is as (1).

$$v_a(t) - v_b(t) = L\frac{\partial i_a(t)}{\partial t} + R_a i_a(t) \tag{1}$$

Using Newton's 2nd law as a foundation, the mechanical equation is produced as (2).

$$T_m(t) = J\frac{\partial^2\theta_m(t)}{\partial t^2} + B\frac{\partial\theta_m(t)}{\partial t} \tag{2}$$

As the input voltage is applied to the armature, current passes through resistance and inductance to create magnetic flux, which in turn causes the rotor to move under the motor torque as shown equation (3).

$$T_m(t) = K_t i_a(t) \tag{3}$$

The motor shaft speed produced the following "back" electromagnetic force (EMF) as shown in (4).

$$v_b(t) = K_b\omega_m(t) = K_b\frac{\partial\theta_m(t)}{\partial t} \tag{4}$$

Eq. (4) is substituted in Eq. (1) and Eq (3) into Eq. (2) the result is shown (5) and (6).

$$V_a(t) = L\frac{\partial i_a(t)}{\partial(t)} + R_a i_a(t) + K_b\frac{\partial\theta_m(t)}{\partial t} \tag{5}$$

and,

$$K_t i_a(t) = J\frac{\partial^2\theta_m(t)}{\partial t^2} + B\frac{\partial\theta_m(t)}{\partial t} \tag{6}$$

The two equations above are transformed using the Laplace transform to produce the two equation (7) and (8).

$$V_a(s) = sLI_a(s) + R_aI_a(s) + sK_b\theta_m(s) \tag{7}$$

$$K_tI_a(s) = s^2J\theta_m(s) + sB\theta_m(s) \tag{8}$$

Substituting (7) and (8) gives the motor TF as in the equation (9),

$$\frac{\omega_m(s)}{V_a(s)} = \frac{K_t}{JLs^2 + (JR_a + BL)s + BR_a + K_tK_b} \tag{9}$$

The transfer function (9) is multiplied by the term 1/s to yield equation (10).

$$\frac{\omega_m(s)}{V_a(s)} = \frac{K_t}{s[JLs^2 + (JR + BL)s + BR + K_tK_b]} \tag{10}$$

where V_a is the DC input voltage, Rais the stator's resistance, L is the stator's inductance, B is the damping coefficient, J is the evaluated rotor moment of inertia, ω is the instant rotor speed, K_b is the phase back-EMF coefficient, and K_t is the constant of line phase torque. Table I presents the parameter values of the DC-servo motor.

The block diagram in Fig. 2 depicts the DC motor's dynamic behavior, as indicated by (9).

TABLE I. DC SERVO MOTOR PARAMETER VALUES

Parameter	Label	Value
R_a	"armature resistance"	1 Ω
L_a	"armature inductance"	0.08 H
J	"motor moment of inertia"	1 kg/m^2
B	"damping coefficient"	0.15 N.m. s
K_b	"motor EMF to speed proportional constant"	1.2 v/rad/sec.
K_t	"motor torque to current proportional constant"	1.2N.m/amb

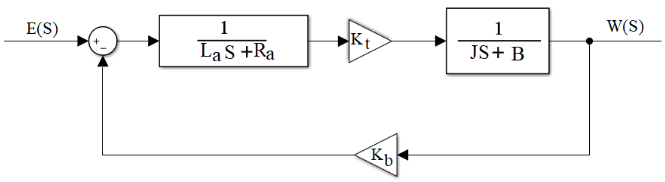

Fig. 2. Equivalent block diagram of DC servo motor

III. METHODOLOGY

This study proposes a method to control the velocity of a DC servo motor. The method uses a fuzzy self-tuning (FST) technique to update the PID gains. The procedure involves two steps:

1. Determining the PID gains using an antcolony optimization strategy (ACS).
2. Self-tuning the gains using FST.

Simulations in MATLAB-SIMULINK were used to evaluate the efficacy of the FST-PID controller, taken into consideration both nominal system parameters and uncertain parameters under various disturbances. The result demonstrates that the proposed algorithm effectively determines the optimal PID gains for the DC servo motor. Three steps can be used to summarize the proposed method:

1. Determine the system's needs, which is the speed DC servo motor.
2. Tune the PID gains using the ACS algorithm.
3. Use FST to self-tune the optimal PID gains online.

This design will develope the transient performance of the DC servo motor when subjected to changing loads, uncertain parameters, and demand conditions. The flowchart in Fig. 3 provides a clear and concise overview of the methodology.

IV. CONTROL TECHNIQUES

Two parts are proposed as the primary construction of the fuzzy self-tuning PID controller. The ant colony optimization procedure is the first control strategy that determines the optimal PID control parameters [51]. This strategy is done by creating an artificial ant colony that explores the parameter space, looking for the combination of parameters that obtains the optimum performance. The second control technique uses a fuzzy self-adjusted PID controller to improve the performance of the controller outputs. This controller uses fuzzy logic to tune the PID parameters based on the system's current state in real-time.

The DC servo- motor is used as a practical study case to determine the performance of the suggested controller. Different simulations are carried out, including random load and parameter variations [52-53]. The result indicates that the suggested controller can achieve good control performance under various conditions.

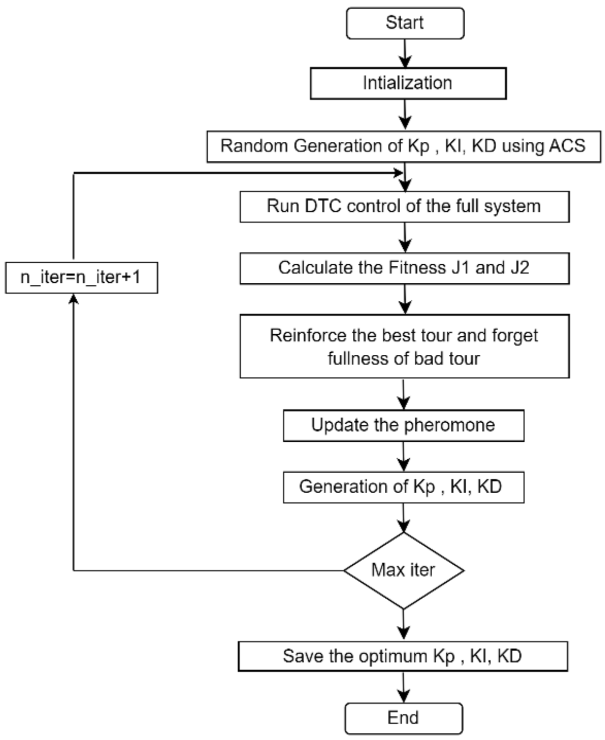

Fig. 3. Flowchart of PID controller optimization by ACS

A. Ant Colony System Based PID Controller

The Ant Colony Optimization algorithm (ACS) was initially suggested by Colorni, Dorigo, and Maniezzo [54]. It uses a group of artificial ants that communicate indirectly with each other to obtain the shortest path from their nest to a predefined objective [55-57]. ACS is often used to optimize control systems. In this context, the goal is to obtain the controller gains for a closed-loop system with an unknown plant and an ant-based PID controller that would minimize or maximize a defined cost function. To search controller settings as efficient as possible, ACS approaches employ the following two performance index criteria [58-60]:

- The number of iterations: This is the number of times the ACS algorithm runs through its entire search process.
- Cost function: This measures how well the controller parameters perform in terms of minimizing or maximizing the defined cost function.

ACS aims to find controller parameters that minimize the number of iterations while also minimizing the cost function.

First Cost Function: The cost function, J1, given in (11), measures the error between the predicted and the factual values. It is calculated by squaring the difference between each predicted value and its corresponding actual value, and then summing the squared differences. The goal is to minimize this cost function by finding the values of the parameters that make the squared differences as small as possible.

$$J_1 = \int_0^{\infty} e(t)^2 \, dt \tag{11}$$

Second Cost Function: The cost function, J2, given in (12) is a mathematical expression that specifies the required characteristics of the system response. It includes four terms: the required "rise time" (trd), the required "maximum overshoot" (Mpd), the required "settling time" (tsd), and the required "steady state error" (essd). The values of the constants c1, c2, c3, and c4 are positive weighting factors that are selected based on the relative importance of each term. If the required "rise time" is more important than the required maximum overshoot, then c1 will be larger than c2. Similarly, if the desired "steady state error" is more important than the desired "settling time", then c4 will be larger than c3. The second cost function can evaluate different control designs and select the one that best meets the desired system response characteristics.

$$J_2 = \frac{1}{[C_1(t_r - t_{rd}) + C_2(M_p - M_{pd}) + C_3(t_s - t_{sd}) + C_4(e_{ss} - e_{ssd})} \quad (12)$$

The suggested design process consists of two steps:

i. Determine the system's optimal PID gains.
ii. Design a fuzzy logic controller (FLC) component to self-adjust based on the discovered PID gains.

The transfer function of the PID controller is represented by equation (13).

$$K(s) = K_P + \frac{K_I}{s} + K_D\, s \quad (13)$$

Where, K_P, K_I, and K_D stand for PID gains.

The optimal PID controller used in this work is based on the "Ant Colony System (ACS)" optimization technique. The "ACS" technique was described in [52]. Table II presents the results of the ideal PID gains for the DC servo motor.

TABLE II. ANT COLONY SYSTEM (ACS) PID CONTROLLER GAINS

	K_P	K_I	K_D
J1-Ant	5.2381	7.0427	0.49468
J2-Ant	8.1381	8.6427	0.49468

B. *Design Procedure of "PID Fuzzy Self Tuning"*

The four primary components of a fuzzy logic system are depicted in Fig. 4 [61].

1. Input devices, formerly known Fuzzifier
2. Baserule
3. Inference procedure
4. Output units that have been referred as Defuzzifier

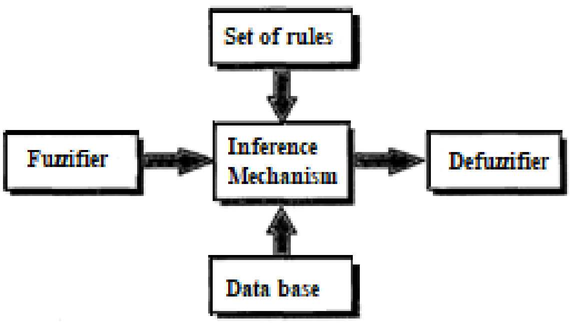

Fig. 4. Block diagram of the fuzzy-logic controller

The PID gains; K_P, K_I, and K_D can be adjusted online after the controlling the system's output. This process is called self-tuning or self-adjusted and is depicted in Fig. 5.

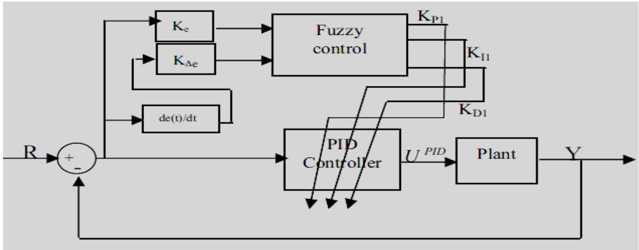

Fig. 5. Fuzzy self-tuned pid controller

Here are the steps involved in fuzzy self-tuning design:

5. Define the input and output variables of the fuzzy controller

The PID controller is a control system that uses three parameters to control a system. The following equation can represent the PID controller as shown in (14).

$$U = K_P + K_I \int edt + K_D \frac{de(t)}{dt} \quad (14)$$

The PID controller can be modified using fuzzy logic to improve its performance. The fuzzy logic controller outputs new gains, K_{P1}, K_{I1}, and K_{D1}, which are then used to calculate the gains after fuzzy effect, K_{P2}, K_{I2}, and K_{D2}.

$$K_{P2} = K_P \times K_{P1}, K_{I2} = K_I \times K_{I1}\;, K_{D2} = K_D \times K_{D1}$$

In other words, the PID controller's gains are multiplied by the fuzzy logic controller's output gains. This results in a new PID controller with improved performance which is then used to calculate the controller output, U. The controller output U after the fuzzy effectis given by equation (15).

$$U = K_{P2} + K_{I2} \int edt + K_{D2} \frac{de(t)}{dt} \quad (15)$$

6. Determine the "membership functions" for the input/output variables

The rule databases of 49 fuzzy rules, input/output variables in the simulation are represented by symmetric triangular fuzzy sets (shown in Fig. 6 and Fig. 7). Table III, Table IV, and Table V simplify the "rule bases." The linguistic labels for a self-tuning PID controller are given as follows: N is for "negative," P is for "positive," Z is for "zero," L is for "small," A is for "medium," and G is for "large." For example, NA denotes "negative-medium," PG denotes "positive-big," and so on.

7. Create the fuzzy rules

The "normalizing gain" for e and Δe can be evaluated using the formulas in (16) and (17).

$$K_e = \frac{(e_{max} - e_{min})}{(e_{max} + e_{min})} \quad (16)$$

$$K_{\Delta e} = \frac{(\Delta e_{max} - \Delta e_{min})}{(\Delta e_{max} + \Delta e_{min})} \quad (17)$$

Where, K_e is the normalizing error input gain. $K_{\Delta e}$ is the Gaining error input normalization.

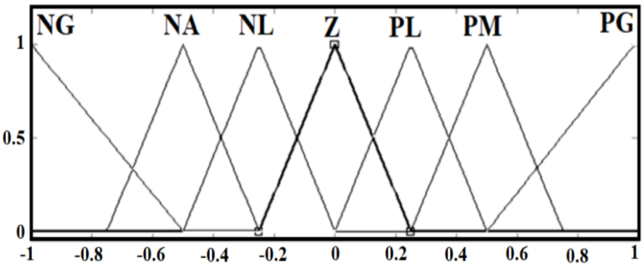

Fig. 6. Input e and Δe membership functions

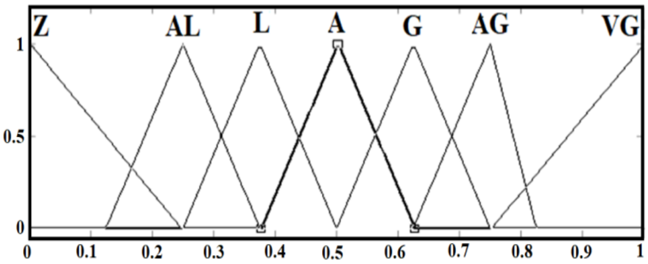

Fig. 7. Output membership functions

The rule bases are instructions that can be used to determine the values of K_{P1}, K_{I1}, and K_{D1}. These values are used to control the system's response due to changes in the error signal.

8. Defuzzify the output of the fuzzy controller

The defuzzification method is a process that takes fuzzy sets and converts them into single, precise values. In this case, the center of gravity method is used for "defuzzification." This method finds the gravity center of the fuzzy set and uses that value as the output of the defuzzification process, which is formulated as (18).

$$U = \frac{\sum_{i=1}^{r} u(u_i)u_i}{\sum_{i=1}^{r} u(u_i)} \quad (18)$$

Where, $u(u_i)$ is the element's membership weight. u_i is the output of the rule i.

9. Calculate the K_{P1}, K_{I1}, and K_{D1} gains based on the defuzzified output

Determine K_{P1}, K_{I1}, and K_{D1} using Table III through Table V, ased on the defuzzified output.

10. Implement the gains in the PID controller
11. Repeat steps 4-6 until the system's output is stable

TABLE III. RULE BASE FOR DETERMINING KP1

e \ Δe	NG	NA	NL	Z	PL	PA	PG
NG	VG	VG	VG	VG	VG	VG	VG
NA	AG	AG	AG	AG	G	AG	VG
NL	G	G	G	G	AG	G	VG
Z	Z	Z	Z	AL	L	L	L
PL	G	G	G	G	AG	G	VG
PA	AG	AG	AG	AG	AG	AG	VG
PG	VG	VG	VG	VG	VG	VG	VG

TABLE IV. RULE BASE FOR DETERMINING KI1

e \ Δe	NG	NA	NL	Z	PL	PA	PG
NG	A	A	A	A	A	A	A
NA	A	A	A	A	A	A	A
NL	L	L	L	L	L	L	L
Z	AL	AL	AL	AL	AL	AL	AL
PL	L	L	L	L	L	L	L
PA	A	A	A	A	A	A	A
PG	A	A	A	A	A	A	A

TABLE V. RULE BASE FOR DETERMINING KD1

e \ Δe	NG	NA	NL	Z	PL	PA	PG
NG	Z	AL	L	A	AG	G	VG
NA	AL	L	A	G	G	G	VG
NL	L	A	G	AG	VG	VG	VG
Z	A	G	AG	AG	VG	VG	VG
PL	AG	AG	VG	VG	VG	VG	VG
PA	G	AG	VG	VG	VG	VG	VG
PG	VG	VG	VG	VG	VG	VG	VG

V. SIMULATION

This section illustrates how the proposed control techniques were implemented using the Matlab and Simulink Toolbox. The first method is an Ant Colony Algorithm-based optimal PID controller. The second method is a self-adjusted fuzzy PID controller. Fig. 8 and Fig. 9 depict the block diagrams of PID controller and fuzzy self-adjusted PID controller, respectively.

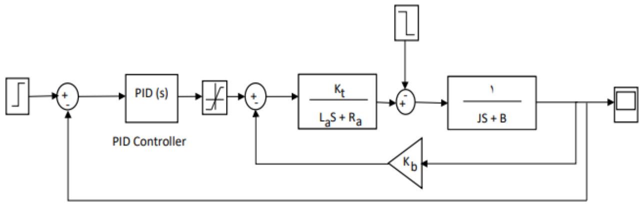

Fig. 8. PID controller for DC servo motor model

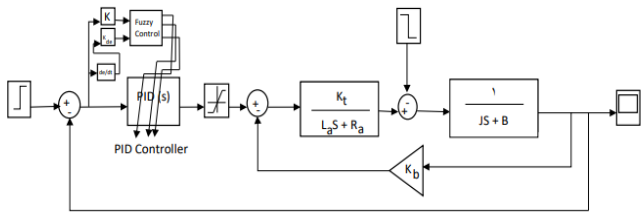

Fig. 9. Fuzzy self tune PID controller for DC-Servo motor model

A. Simulation Results

The simulation results of the suggested techniques are presented in this section. Optimal PID parameters and Fuzzy Self-adjusted PID gains are derived through a series of tests, which include parameter uncertainty and load torque variations.

In test 1, the optimal PID parameters are obtained using the two cost functions after increasing the motor speed. These parameters are then compared to those obtained using trial and error.

In test 2, the optimal PID parameters are obtained using the two cost functions after decreasing the motor speed. The results of these two cost functions are then compared to each other.

In test 3, the best optimal parameters obtained from test 2 (calculated using the first cost function) are self-tuned.

In test 4, obtains the system response due to uncertainty in Ra and La.

Test 1: Step Disruption Speed (r.p.m.) Equals 1v and Under Load Torque

A step disturbance of about 1V was applied to the DC-servo motor to compare the optimal ACS-PID controller with the trial-and-error-PID controller. Fig. 10 and Fig. 11 show the response of the rotor speed and the control input.

The ACS-PID controllers with the two cost functions performed better than the trial-and-error-PID controllers in terms of having quicker settling times and no steady-state error. The trial-and-error-PID controller still performed worse than the ACS-PID controller, especially when the results were based on the two cost functions. The results are presented in Table VI.

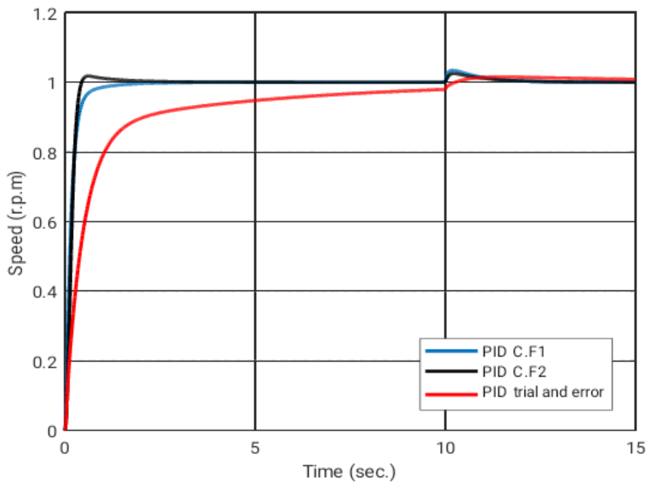

Fig. 10. DC servo motor system response with PID C.F1, PID C.F2 and trial and error-PID controllers

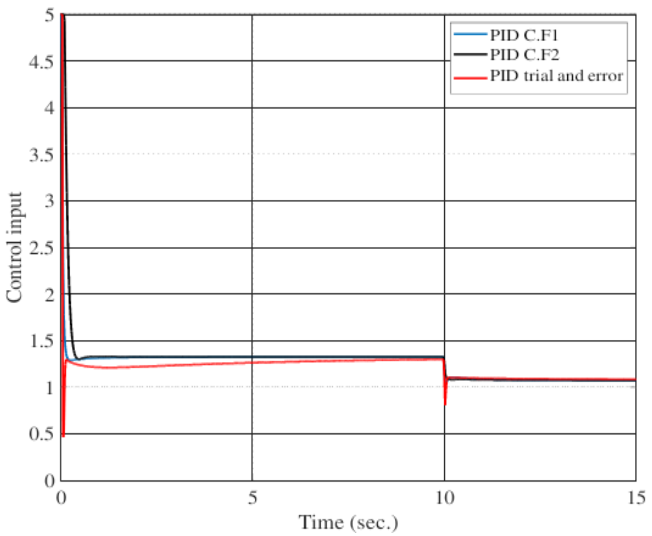

Fig. 11. Controller output for PID C.F1, PID C.F2 and trial and error-PID controllers

TABLE VI. PARAMETERS COMPARISON OF TWO COST FUNCTIONS WITH TRIAL AND ERROR PARAMETERS

	PID 1	PID 2	PID (Trial and Error)
Kp	5.2381	8.1381	5.6059
Ki	7.0427	8.6427	1.1895
Kd	0.49468	0.49468	2.1993
Rise Time	0.3208	0.2503	2.2586
Settling Time	10.7258	10.4824	10.0453
Settling Min	0.9111	0.9165	0.9078
Settling Max	1.0345	1.0252	1.0151
Overshoot	3.4393	2.5079	0.6452
Undershoot	0	0	0
Peak	1.0345	1.0252	1.0151
Peak Time	10.200	10.220	11.5000

Test 2: Step Disruption Speed (r.p.m.) Equals 1v and Over Load Torque

At time t = 10 seconds, the speed was abruptly slowed by 0.3v to evaluate the effectiveness of the proposed controllers. This decrease in speed caused a rise in load torque. Fig. 12 and Fig. 13 show the rotor speed response and the control input against time.

The system responses of both the PID C.F1 (cost function 1) controller and the PID C.F2 (cost function 2) controller clearly show that they were able to overcome these changes and provide a good response with a small "settling time." However, the PID C.F1 controller was more effective than the PID C.F2 controller, as it achieved a smaller settling time

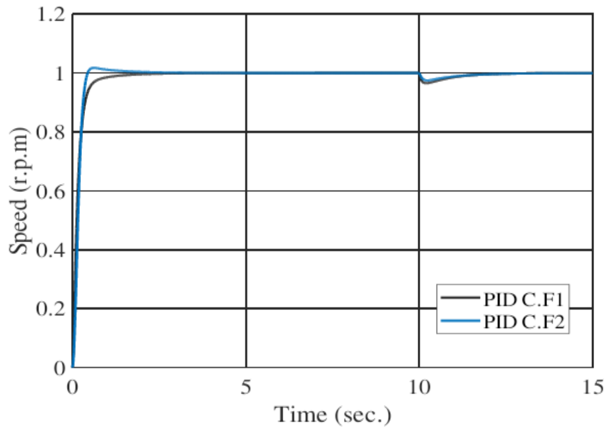

Fig. 12. DC servo motor system response with PID C.F1 and PID C.F2

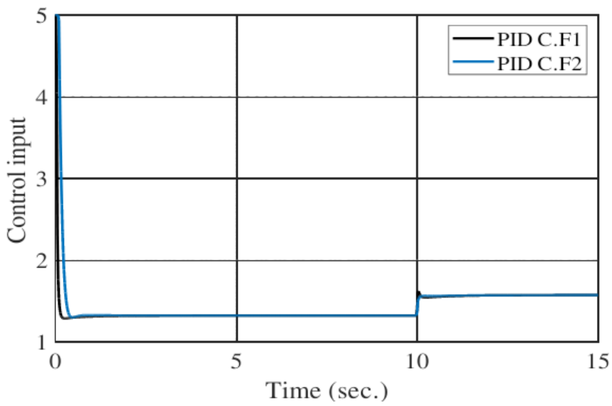

Fig. 13. Controller output for PID C.F1 and PID C.F2

Test 3: Step Disruption Speed (r.p.m.) = 1v and Under Load Torque

The effectiveness of the suggested algorithm was determined by abruptly increasing the speed by 0.3 volts at t = 10 seconds. This increase in speed led to a decrease in burden torque. Fig. 14 illustrates that using a self-tuned PID controller resulted in less overshoot and a shorter settling time than using an ideal PID. Fig. 15 illustrates the output controller's time response. In the practical simulation, FST-PID controller output has a slight overshoot, but it responds more rapidly to speed adjustments at this point.

In conclusion, the self-tuned PID controller was demonstrated to be more effective than the ideal PID controller in terms of minimizing "overshoot and settling time". Despite a modest overshoot, the FST-PID controller performed well in the practical simulation.

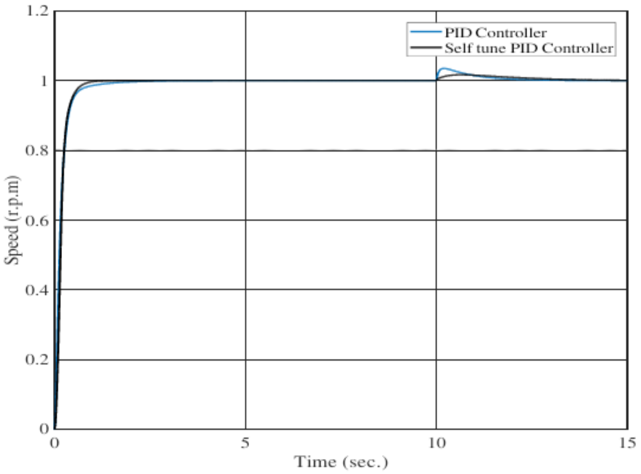

Fig. 14. DC servo motor system response with PID and self-tuning of PID

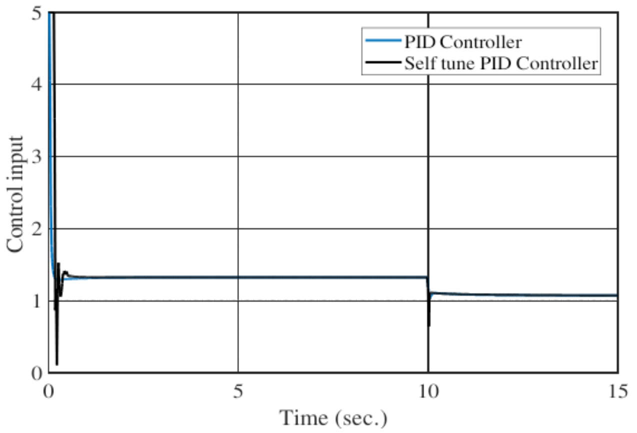

Fig. 15. Controller output for PID and fuzzy self-tuning of PID

Test 4: Uncertain System Parameters

With variation 30%, the Ra and La parameters of the DC Servo Motor are regarded to be uncertain. The parameters are modified as follow: from 0 to 5 (a value of -30%), from 5 to 10 (a value of 0%), and from 10 to 15 (a value of +30%).

The uncertain system parameters are presented in Table VII. The proposed "self-tuning PID controller" is shown to respond robustly to these uncertain parameters.

The results of the self-tuning PID controller are extremely encouraging, particularly in the presence of system linearity and indeterminate parameters. This is shown in Fig. 16, which compares the results of the "self-tuning PID controller" to a regular PID controller. The time response of the controller inputs is displayed in Fig. 17.

In summary, the "self-tuning PID controller" is a robust and effective method of controlling DC Servo Motors, even in ambiguous parameters.

TABLE VII. UNCERTAIN PARAMETERS OF THE SYSTEM

Parameters	0% Value	30% Increase	30 % decrease
Ra	1	1.3	0.7
La	0.08	0.104	0.056

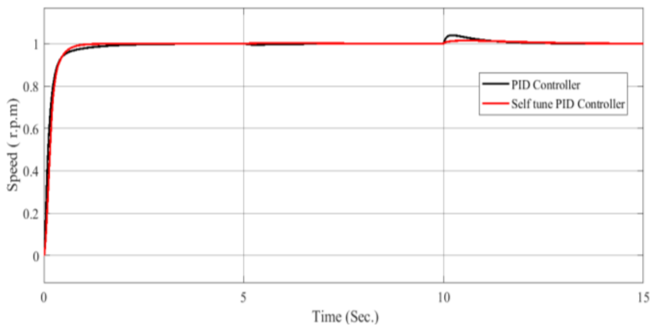

Fig. 16. DC servo motor system response with PID and self-tuning of PID

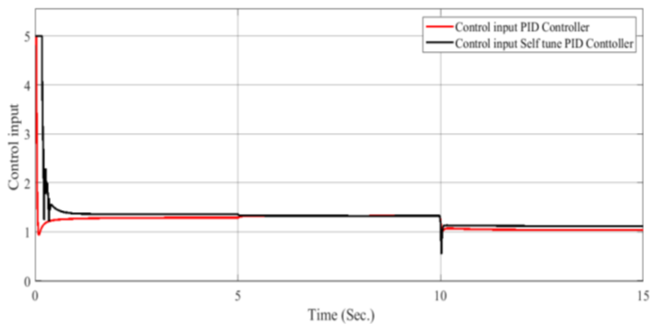

Fig. 17. Controller output for PID and "fuzzy self-tuning PID"

VI. CONCLUSION

This study demonstrates the design method of PID controller with fuzzy self-adjusted applied to DC Servo Motor to enhance performance, as reduced "transient time" and immunity to external perturbations. To investigate the proposed technique, a comparative study was carried out between fuzzy self-adjusted PID, and PID controllers based on various tests, such as parameter uncertainty and load torque variations. The results indicate that the fuzzy self-adjusted PID and PID controllers can dampen system oscillations adequately. In comparison to the "PID controller," the self-adjusted-PID controller significantly enhances the performance of the DC servo motor's speed control. PID Controllers are less capable of handling parameter uncertainty than self-adjusted-PID Controllers. In conclusion, the results confirmed the effectiveness of the suggested self-adjusted-PID controller, and the system performance was adequate in both normal and abnormal cases (due to parameters uncertainties). Also, fuzzy logic is unable to handle uncertainty accurately, so in future work, interval type-2 fuzzy logic could be used to improve the accuracy of uncertainty calculations.

REFERENCES

[1] A. K. Mishra, V. K. Tiwari, R. Kumar, and T. Verma, "Speed control of dc motor using artificial bee colony optimization technique," *2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE)*, pp. 1-6, 2013, doi: 10.1109/CARE.2013.6733772.

[2] A. Rodríguez-Molina, M. G. Villarreal-Cervantes, and M. Aldape-Pérez, "An adaptive control study for a DC motor using metaheuristic algorithms," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13114–13120, 2017.

[3] A. Rodríguez-Molina, M. G. Villarreal-Cervantes, J. Álvarez-Gallegos, and M. Aldape-Pérez, "Bio-inspired adaptive control strategy for the highly efficient speed regulation of the DC motor under parametric uncertainty," *Appl. Soft Comput. J.*, vol. 75, pp. 29–45, 2019.

[4] D. Potnuru, K. Alice Mary, and C. Sai Babu, "Experimental implementation of flower pollination algorithm for speed controller of a BLDC motor," *Ain Shams Eng. J.*, vol. 10, no. 2, pp. 287–295, 2019.

[5] A. T. El-Deen, A. A. H. Mahmoud, and A. R. El-Sawi, "Optimal PID tuning for DC motor speed controller based on genetic algorithm," *Int. Rev. Autom. Control*, vol. 8, no. 1, pp. 80–85, 2015.

[6] Z. -L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," in *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 384-391, 2004, doi: 10.1109/TEC.2003.821821.

[7] L. Syafaah, Widianto, I. Pakaya, D. Suhardi, and M. Irfan, "PID designs using DE and PSO algorithms for damping oscillations in a DC motor speed," *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 1-5, 2017, doi: 10.1109/EECSI.2017.8239138.

[8] B. Hekimoğlu, "Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm," *IEEE Access*, vol. 7, pp. 38100–38114, 2019.

[9] X. Lin, Y. Liu, and Y. Wang, "Design and Research of DC Motor Speed Control System Based on Improved BAS," *2018 Chinese Automation Congress (CAC)*, pp. 3701-3705, 2018, doi: 10.1109/CAC.2018.8623171.

[10] S. Duman, D. Maden, and U. Güvenç, "Determination of the PID controller parameters for speed and position control of DC motor using Gravitational Search Algorithm," *2011 7th International Conference on Electrical and Electronics Engineering (ELECO)*, pp. I-225-I-229, 2011.

[11] J. Agarwal, G. Parmar, R. Gupta, and A. Sikander, "Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor," *Microsyst. Technol.*, vol. 24, no. 12, pp. 4997–5006, 2018.

[12] U. Bhatnagar and A. Gupta, "Application of grey wolf optimization in optimal control of DC motor and robustness analysis," *SKIT Research Journal*, vol. 8, no. 1, pp. 19–25, 2018.

[13] M. H. Jasim, "Tuning of a PID controller by bacterial foraging algorithm for position control of DC servo motor," *Engineering and Technology Journal*, vol. 36, no. 3A, pp. 287-294, 2018.

[14] L. J. Qiu, M. F. Qu, and X. Y. Liu, "Position Sensorless Control of Brushless DC Motor Based on Sliding Mode Observer," In *International Conference on Advanced Hybrid Information Processing*, pp. 342-355, 2021.

[15] S. A. Fayad, M. Shaban, M. Attia, and S. A. M. Abdelwahab, "Performance Enhancement of Speed and Position Control for DC Servo Motor Using Artificial Intelligence Technique," *International Journal on Electrical Engineering and Informatics*, vol. 14, no. 3, pp. 552-566, 2022.

[16] A. A. Emhemed and R. B. Mamat, "Modelling and simulation for industrial DC motor using intelligent control," *Procedia Engineering*, vol. 41, pp. 420-425, 2012.

[17] S. Simon and P. Rajalakshmy, "Speed Control of DC Motor using PSO Based Fuzzy Logic Controller," *International Journal of Research & Technology (IJERT)*, vol. 3, no. 2, pp. 877-880, 2014.

[18] I. Petrás, "Fractional-order feedback control of DC motor," *Journal of Electrical Engineering*, vol. 60, no. 3, pp. 117–128, 2009.

[19] R. Morales, J. A. Somolinos, and H. S. Ramírez, "Control of a DC motor using algebraic derivative estimation with real time experiments," *Measurement*, Vol. 47, pp. 401–17, 2014.

[20] R. Bai, "Neural network control-based adaptive design for a class of DC motor systems with the full state constraints," *Neurocomputing*, vol. 168, pp. 65–9, 2015.

[21] S. Saxena and Y. V. Hote, "Design and validation of fractional-order control scheme for DC servomotor via internal model control approach," *IETE Technical Review*, vol. 36, no. 1, pp. 49-60, 2019.

[22] H. Yang, D. Dong, Y. Ren, J. Xu, and S. Yang, "Study on Fuzzy PID Control in Double Closed-Loop DC Speed Regulation System," *2011 Third International Conference on Measuring Technology and Mechatronics Automation*, pp. 465-469, 2011, doi: 10.1109/ICMTMA.2011.687.

[23] T. Ravichandran and F. Karray, "Knowledge based approach for online self-tuning of PID-control," *Proceedings of the 2001 American Control Conference*, vol. 4, pp. 2846-2851, 2001, doi: 10.1109/ACC.2001.946328.

[24] G. Y. Long, L. H. Long, and F. J. Bo, "Design and Simulation of Cascade Fuzzy Self - Adaptive PID Speed Control of Thyristor-Driven DC Motor", *International Conference on Machine Learning and Cybernetics*, pp. 655-660, 2006.

[25] P. Thepsatorn, A. Numsomran, V. Tipsuwanporn, and T. Teanthong, "DC Motor Speed Control using Fuzzy Logic based on LabVIEW," *2006 SICE-ICASE International Joint Conference*, pp. 3617-3620, 2006, doi: 10.1109/SICE.2006.314890.

[26] B. B. Alagoz, A. Ates, and C. Yeroglu, "Auto-tuning of PID controller according to fractional-order reference model approximation for DC rotor control," *Mechatronics*, vol. 23, pp. 789–97, 2013.

[27] K. -Y. Cheng and Y. -Y. Tzou, "Fuzzy optimization techniques applied to the design of a digital PMSM servo drive," in *IEEE Transactions on Power Electronics*, vol. 19, no. 4, pp. 1085-1099, July 2004, doi: 10.1109/TPEL.2004.830036.

[28] C. Xia, P. Guo, T. Shi, and M. Wang, "Speed control of brushless DC motor using genetic algorithm based fuzzy controller," In *Proceeding of the 2004 International Conference on Intelligent Mechatronics and Automation, Chengdu, China, 3rd edn. A Treatise on Electricity and Magnetism*, vol. 2, pp. 68-73, 2004.

[29] L. Wang, H. Li, and Y. Sun, "Fuzzy Control Based on Self-adjusted Parameter for Brushless DC Motor," *2007 International Conference on Mechatronics and Automation*, pp. 2352-2356, 2007, doi: 10.1109/ICMA.2007.4303921.

[30] S. Wang, S. Yu, and Z. Feng, " A method for controlling a loading system based on a fuzzy PID controller," *Mechanical Science and Technology for Aerospace Engineering*, vol. 30, no. 1, pp. 166-172, 2011.

[31] Y. Lv, H. Luo, and Y. Cai, "Research on tuning method for fuzzy PID," *The Fourth International Workshop on Advanced Computational Intelligence*, pp. 334-337, 2011, doi: 10.1109/IWACI.2011.6160027.

[32] I. H. Altas and A. M. Sharaf, "A generalized direct approach for designing fuzzy logic controllers in Matlab/Simulink GUI environment," *International journal of information technology and intelligent computing*, vol. 1, no. 4, pp. 1-27, 2007.

[33] A. Z. Alassar, I. M. Abuhadrous, and H. A. Elaydi, "Modeling and control of 5 DOF robot arm using supervisory control," *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, pp. 351-355, 2010, doi: 10.1109/ICCAE.2010.5451398.

[34] M. S. A. Omar, T. Y. Khedr, and B. A. A. Zalam, "Particle swarm optimization of fuzzy supervisory controller for nonlinear position control system," *2013 8th International Conference on Computer Engineering & Systems (ICCES)*, pp. 138-145, 2013, doi: 10.1109/ICCES.2013.6707189.

[35] Z. Bingül and O. Karahan, "A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control," *Expert Systems with Applications*, vol. 38, no. 1, pp. 1017-1031, 2011.

[36] S. Tiwari, A. Bhatt, A. C. Unni, J. G. Singh, and W. Ongsakul, "Control of DC Motor Using Genetic Algorithm Based PID Controller," *2018 International Conference and Utility Exhibition on Green Energy for Sustainable Development (ICUE)*, pp. 1-6, 2018, doi: 10.23919/ICUE-GESD.2018.8635662.

[37] J. Agarwal, G. Parmar, R. Gupta, and A. Sikander, "Analysis of grey wolf optimizer based fractional order PID controller in speed control

of DC motor," *Microsystem Technologies*, vol. 24, pp. 4997-5006, 2018.

[38] V. Yadav and V. K. Tayal, "Optimal Controller Design for a DC Motor using PID Tuner," *2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, pp. 442-445, 2018, doi: 10.1109/PEEIC.2018.8665658.

[39] 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*, vol. 2, pp. 1-3, 2009.

[40] L. S. Mezher, "Characteristics of servo DC motor with PID controller," *Journal of Mechanical Engineering Research and Developments*, vol. 44, no. 2, pp. 392-400, 2021.

[41] P. K. Meena and B. Bhushan, "Simulation for Position Control of DC Motor using Fuzzy Logic," *International Journal of Electronics, Electrical and Computation System (IJEECS)*, vol. 6, no. 6, 2017.

[42] B. Li, X. Wang, L. Wang, W. Yang, and J. Xia, "Adaptive Event-triggered Fuzzy Control for DC Motor Servo Systems," *2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 1101-1105, 2021, doi: 10.1109/DDCLS52934.2021.9455634.

[43] H. Wang, P. X. Liu, and B. Niu, "Robust fuzzy adaptive tracking control for nonaffine stochastic nonlinear switching systems", *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2462-2471, 2018.

[44] J. W. Xia, J. Zhang, J. E. Feng, Z. Wang, and G. M. Zhuang, "Command filter-based adaptive fuzzy control for nonlinear systems with unknown control directions", *IEEE Transactions on Systems Man and Cybernetics: Systems*, vol. 51, no. 3, pp. 1945-1953, 2021.

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

[46] H. X. Li and S. K. Tso, "Quantitative design and analysis of fuzzy proportional-integralderivative control a step towards autotuning," *International Journal of Systems Science*, vol. 31, no. 5, pp. 545-553, 2000.

[47] L. Reznik, O. Ghanayem, and A. Bourmistrov, "PID plus fuzzy controller structures as a design base for industrial applications," *Engineering applications of artificial intelligence*, vol. 13, no. 4, pp. 419-430, 2000.

[48] H. Alzarok, "Towards achieving an optimum speed performance for DC servo motors via Fuzzy logic controllers," *2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, pp. 1-8, 2022, doi: 10.1109/MI-STA54861.2022.9837780.

[49] N. L. Ismail, K. A. Zakaria, N. S. Nazar, M. Syaripuddin, A. S. N. Mokhtar, and S. Thanakodi, "DC motor speed control using fuzzy logic controller," In *AIP conference proceedings*, vol. 1930, no. 1, 2018.

[50] Y. A. Almatheel and A. Abdelrahman, "Speed control of DC motor using Fuzzy Logic Controller," *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, pp. 1-8, 2017, doi: 10.1109/ICCCCEE.2017.7867673.

[51] Q. Zeng and G. Tan, "Optimal Design of PID Controller Using Modified Ant Colony System Algorithm," *Third International Conference on Natural Computation (ICNC 2007)*, pp. 436-440, 2007, doi: 10.1109/ICNC.2007.518.

[52] W. Zhang, Y. Fang, R. Ye, and Z. Wang, "Analysis and design of a double fuzzy PI controller of a voltage outer loop in a reversible three-phase PWM converter," *Energies*, vol. 13, no. 15, p. 3778, 2020.

[53] E. W. Suseno and A. Ma'arif, "Tuning of PID controller parameters with genetic algorithm method on DC motor," *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 41-53, 2021.

[54] D. Somwanshi, M. Bundele, G. Kumar, and G. Parashar, "Comparison of fuzzy-PID and PID controller for speed control of DC motor using LabVIEW," *Procedia Computer Science*, vol. 152, pp. 252-260, 2019.

[55] M. Omar, M. Soliman, A. A. Ghany, F. Bendary, "Optimal tuning of PID controllers for hydrothermal load frequency control using ant colony optimization," *International journal on electrical engineering and informatics*, vol. 5, no. 3, p. 348, 2013.

[56] M. G. Abdolrasol, "Artificial neural networks based optimization techniques: A review," *Electronics*, vol. 10, no. 21, p. 2689, 2021.

[57] H. Albalawi, S. A. Zaid, M. E. El-Shimy, and A. M. Kassem, "Ant Colony Optimized Controller for Fast Direct Torque Control of Induction Motor," *Sustainability*, vol. 15, no. 4, p. 3740, 2023.

[58] M. A. George, D. V. Kamat, and C. P. Kurian, "Electronically Tunable ACO Based Fuzzy FOPID Controller for Effective Speed Control of Electric Vehicle," in *IEEE Access*, vol. 9, pp. 73392-73412, 2021, doi: 10.1109/ACCESS.2021.3080086.

[59] M. Omar, M. Soliman, A. M. A. Ghany, and F. Bendary, "Ant Colony Optimization based PID for single area load frequency control," *2013 5th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 119-123, 2013.

[60] S. Prainetr, T. Phurahong, K. Janprom, and N. Prainetr, "Design Tuning PID Controller for Temperature Control Using Ant Colony Optimization," *2019 IEEE 2nd International Conference on Power and Energy Applications (ICPEA)*, pp. 124-127, 2019, doi: 10.1109/ICPEA.2019.8818517.

[61] M. Akar and I. Temiz, "Motion controller design for the speed control of dc servo motor," *International Journal of Applied Mathematics and Informatics*, vol. 1, no. 4, pp. 131-137, 2007.