
**Düzce University**
**Journal of Science & Technology**

*Research Article*

# Enhancing DC Motor Speed Control Performance Using Heuristic Optimization and Comparative Analysis of Control Methods

Beytullah BOZALİ [a,*], Nasser Bandar Nasser AL-SAREMİ [a], Ali ÖZTÜRK [b]

[a] *Department of Mechatronics, Faculty of Engineering, Düzce University, Düzce, TURKEY*
[b] *Department of Electric and Electronic Engineering, Faculty of Engineering, Düzce University, Düzce, TURKEY*
** Corresponding author's e-mail address: beytullahbozali@duzce.edu.tr*
DOI: 10.29130/dubited.1505316

## ABSTRACT

Direct Current (DC) motors are an important component that converts electrical energy into mechanical energy, used in a wide range of applications from industrial applications to home appliances. DC motor speed control has an important role in industrial processes to increase efficiency, realize precise movements and optimize energy consumption. In this study, various control methods and parameter optimization techniques for speed control of DC motors, which have a wide range of applications, have been systematically analyzed. The aim of the study is to develop an effective control strategy to ensure that DC motors reach the determined target speed by monitoring them in real time at different speeds and to minimize fluctuations caused by variable loads or external factors. In our study, Proportional-Integral-Derivative (PID), Proportional-Integral (PI), and Proportional-Derivative (PD) control methods were used. The parameters of these controllers were tuned using Matlab Tuned, The Cheetah Optimizer (CO) Algorithm, a new generation heuristic optimization method, and Particle Swarm Optimization (PSO), a widely accepted optimization method. The performances of the controllers were determined using criteria such as Integral of Absolute Error (IAE), Integral Squared Error (ISE), and Integral of Time multiplied by Absolute Error (ITAE). According to the results obtained, it was found that the PID, PI and PD control parameters determined using the CO Algorithm performed better than the controllers created using Matlab Tuned and PSO methods. New optimization methods, such as the CO Algorithm, have been found to have significant potential to improve the performance of control systems. Thanks to this study, it offers a practical approach for optimizing DC motor speed control in industrial processes. As a result, it has been found that the control parameters determined by the CO Algorithm have significant potential in improving the performance of DC motor speed control and control systems compared to other optimization methods.

***Keywords:*** *CO Algorithm, PSO, PID, PI, PD, DC Motor Control*

# Sezgisel Optimizasyon Kullanarak DC Motor Hız Kontrol Performansının Artırılması ve Kontrol Yöntemlerinin Karşılaştırmalı Analizi

## ÖZ

Doğru Akım (DA) motorları, endüstriyel uygulamalardan ev aletlerine kadar geniş bir yelpazede kullanılan, elektrik enerjisini mekanik enerjiye dönüştüren önemli bir bileşendir. DA motor hız kontrolü, endüstriyel süreçlerde verimliliği artırmak, hassas hareketleri gerçekleştirmek ve enerji tüketimini optimize etmek için önemli bir role sahiptir. Bu çalışmada, yaygın kullanım alanlarına sahip DA motorlarının hız kontrolü için çeşitli kontrol yöntemleri ve parametre optimizasyon teknikleri sistematik bir şekilde analiz edilmiştir. Çalışmanın

Received: 26/06/2024, Revised: 25/07/2024, Accepted: 25/07/2024

amacı DA motorların farklı hızlarda gerçek zamanlı olarak izleyerek belirlenen hedef hıza ulaşmasını sağlamak ve değişken yükler veya dış etkenlerden kaynaklanan dalgalanmaları minimize etmek için etkili bir kontrol stratejisi geliştirmektir. Çalışmamızda Oransal-İntegral-Türev (PID), Oransal-İntegral (PI), ve Oransal- Türev (PD) kontrol yöntemleri kullanılmıştır. Bu kontrolörlerin parametreleri, Matlab Tuned, yeni nesil sezgisel optimizasyon yöntemi olan Çita Optimizasyon (CO) Algoritması ve geniş kabul görmüş optimizasyon yöntemi olan Parçacık Sürü Optimizasyonu (PSO) kullanılarak ayarlanmıştır. Kontrolörlerin performanslarını, Hatanın Mutlak Değerinin İntegrali (IAE), Hata Karenin İntegrali (ISE) ve Zaman Mutlak Hatanın İntegrali (ITAE) gibi kriterler kullanılarak belirlenmiştir. Elde edilen sonuçlar göre, CO Algoritması kullanılarak belirlenen PID, PI ve PD kontrol parametrelerinin, Matlab Tuned ve PSO yöntemleri kullanılarak oluşturulan kontrolörlerden daha iyi performans gösterdiği bulunmuştur. CO Algoritması gibi yeni optimizasyon yöntemlerinin, kontrol sistemlerinin performansını artırmak için önemli bir potansiyel taşıdığını bulunmuştur. Bu çalışma sayesinde, endüstriyel süreçlerde DA motor hız kontrolünün optimize edilmesi için pratik bir yaklaşım sunmaktadır. Sonuç olarak, CO Algoritmasıyla belirlenen kontrol parametrelerinin, diğer optimizasyon yöntemlerine göre DA motor hız kontrolünde ve kontrol sistemlerinin performansını iyileştirmede önemli potansiyele sahip olduğu bulunmuştur.

***Anahtar Kelimeler:*** *CO Algoritması, PSO, PID, PI, PD, DA Motor Kontrol*

# I. INTRODUCTION

DC motors are important components that convert electrical energy into mechanical energy and are widely used in a wide range of applications, from industrial applications to household appliances. Speed control of these motors has a critical role in increasing efficiency in industrial processes, performing precise movements and optimizing energy consumption. The main reasons for choosing DC motors are their affordable price, ease of use, flexibility, and durability [1]. Today, many daily used systems include DC motors that convert electrical energy into mechanical energy. Therefore, the performance increase achieved in DC motor control contributes to many innovative application areas such as electric vehicles. Scientific developments in this field influence strategies to increase the performance and efficiency of electric vehicles, providing more sustainable and environmentally friendly transportation solutions. These developments are concentrated in areas such as energy management, data analytics and control systems, making the use of DC motors more efficient and reliable [2], [3]. DC motors are a frequently preferred motor type in a wide range of applications [4], [5]. When the literature is examined, PID controller is used as a DC motor speed control technique. As a result of the literature analysis, it is observed that the PID controller is widely used in DC motor speed control. In addition, it has been determined that brushed and brushless DC motors are frequently preferred in this control technique. For Brushless DC (BLDC) motors to operate stably under various speed and load conditions, advanced performance and robust speed control are required. For this purpose, they preferred adaptive Fractional Order PID (FOPID) controllers to increase the performance of BLDC motors by using the Artificial Bee Colony (ABC) algorithm [6]. The design, implementation and analysis of integer order (IO), Fractional Order (FO) and Artificial Bee Colony (ABC) based PID/FOPID controller for speed regulation in DC motor drive are discussed in detail [7]. DC motors are frequently preferred in controlling their speed due to their simplicity and precision. [8] in their study, motor speed was controlled by using a Fractional Order PID (FOPID) controller instead of the traditional PID controller. Three different DC motor configurations, permanent magnet, externally energized field motor, shunt and series winding motors, were examined. The input voltages required to reach different constant operating speeds of the DC motor have been calculated [9]. They simulated and implemented a speed control strategy to evaluate their operating performance for universal motors in a fully DC electric framework [10]. They used a modified Kalman filter estimator for mechanical sensor-less rotational speed estimation of a brushed DC motor [11]. In their study, two different controllers were proposed for speed control of brushless DC motors: fuzzy online gain adjustable PID controller and fuzzy PID controlled online ANFIS controllers [12]. For a fast and effective speed control, they proposed a hybrid technique in DC motors [13]. In another study, an adaptive fuzzy logic-based speed controller developed for a DC motor is presented. This controller is implemented on field programmable gate array (FPGA) hardware [14]. In another study, experiments were conducted using PI and PID controllers for real-time speed control of the DC motor and the speed (rpm) results were compared. In experiments carried out under the same conditions, controller performance was evaluated based on different criteria such as ISE, ITSE, IAE and ITAE [15]. Studies have been conducted using different algorithms for DC motor control [13], [16]-[18].

The main contribution of this study to the literature is by comprehensively comparing the performances of different control methods and optimization techniques used in DC motor speed control, providing important information for the development of more efficient and effective control strategies in industrial applications. Additionally, by demonstrating the potential of next-generation optimization methods, it directs future research in this field and can help motors determine the most appropriate control methods in DC motor systems facing variable loads and external factors. In this way, valuable contributions are made to increasing energy efficiency in industrial processes, ensuring precise motion control and optimizing overall system performance. Highlighting the potential of the Cheetah Optimization Algorithm to improve the performance of control systems indicates that this algorithm can be used in a wider range of applications and may open new research areas for future studies.

# II. MATERIAL AND METHOD

In this section, a DC motor used in our study is described. Information is presented about PSO and CO, which are heuristic optimization methods used to determine the coefficients of the PID controller used for speed control of the DC motor. Additionally, different criteria used to evaluate the performance of controllers are described.

## A. DC MOTOR

DC motors are basic electromechanical devices that convert electrical energy into mechanical energy and are widely used in systems requiring speed and torque control in a wide range of applications. The operating characteristics of DC motors provide significant advantages in many industries and are therefore preferred due to their advantages such as high performance, easy control, and ability to adjust their speed over a wide range. With the development and widespread use of electrical home appliances, DC motors have also been used in more powerful, safe, and cost-effective control areas. Control of systems is an interdisciplinary subject and is included in all motor branches [19]. DC motors are widely used in various applications. In industrial fields, they are frequently preferred as main drive devices in robotic arms, vehicles, machine tools and many other fields. These motors stand out for their flexibility and powerful performance to suit a wide range of applications [17], [20]-[22].

### A. 1. Finding DC Motor Parameters

Just as the relationship between input and output of physical systems can be expressed in the form of a transfer function, DC motors can also be modeled in a transfer function structure using four equations. They can also be modeled using state space equations, which give the relationship between state variables, derivatives of state variables, inputs, and outputs [23]. The electrical equivalent model of the DC motor used in our study is given in Figure 1 [15].

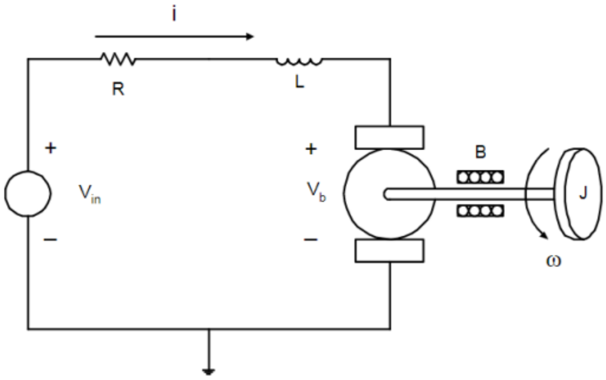



***Figure 1.*** *Electrical model of DC motor [15].*

$$V_{in} = Ri + L\frac{di}{dt} + V_b - Bw \tag{1}$$

The electrical equivalent model of the DC motor is given in Equation 1. Here: $V_{in}$ represents motor supply voltage (volt, V), *R* armature resistance of the motor (ohm, Ω), *L* Inductance of the motor (henry, H), *i* Current of the motor (ampere, A), *di/dt* change of current over time (ampere/second, A/s), $V_b$ back emf voltage (volt, V), *B* damping constant (N.m/A), *w* angular speed of the motor (radians/second, rad/s), *J* moment of inertia (kg.m^2) [15], [19].

The relationship between the Electromotive Force Constant (Ke) and Torque Constant (Kt) of the DC motor is expressed using equation 2.

$$\tau(t) = K_t i(t) \tag{2}$$

Equation 2 expresses the relationship between DC motor current *i* and torque $\tau$ on the motor shaft. Here: $\tau$ represents torque of the motor (newton-meter, Nm), $K_t$ motor torque constant (newton-meter/ampere, Nm/A), *i* motor current (ampere, A).

The armature current and the strength of the magnetic field are proportional to the torque produced by the DC motor. This ratio represents the armature current multiplied by a constant coefficient. These equations are given in equation 3.

$$T = K_t i, \qquad 0.06539664Nm = K_t * 6.5A, \qquad K_t = 0.01Nm/A \tag{3}$$

In this study, the moment of inertia (*J*) value of the DC motor used in the simulation environment is accepted as $J = 0.01(kg - m^2)$.

$$T_{m(t)} = J.\frac{d\omega}{dt} t\ B.\omega(t)\ t\ T_{L(t)} \tag{4}$$

Equation 4 can be used to find the friction coefficient (B) of the DC motor. Equation 4 implies that the differentiated term is zero when the speed is constant. In this case, Equation 5 is obtained. To determine the coefficient of friction, this equation is usually expressed by the following formula:

$$\frac{d\omega(t)}{dt} = 0 \text{ ise } \quad T_{m(t)} = B.\omega(t)\ t\ T_{L(t)} \tag{5}$$

Shaft torque $T_{m(t)}$ and angular velocity ω(t) values were measured by starting the motor under a certain load torque value $T_{L(t)}$. A graph is drawn using these measurements. The graph was analyzed by curve fitting and the friction coefficient (*B)* and load moment function $T_{L(t)}$ values were determined. The friction coefficient of the DC motor used in this study, measured under laboratory conditions by Pololu company, was taken as *B* = 0.1 (Nm/rad/s) [15], [19], [24].

***Table 1.*** *DC motor parameters used in our study.*

| | Parameter | Estimated Value | Unite |
|---|---|---|---|
| $J$ | Moment of Inertia | 0.01 | $kg-m^2$ |
| $B$ | Viscous Friction | 0.1 | $\frac{Nm}{rad/s}$ |
| $K_e$ | Back Emf Constant | 0.01 | $\frac{V}{rad/s}$ |
| $K_t$ | Torque Constant | 0.01 | $Nm/A$ |
| $R$ | Resistance | 2.9 | $\Omega$ |
| $L$ | Inductance | 291e-3 | $H$ |

In our study, the parameters of the DC motor that we modeled in Matlab simulation are given in Table 1. Calculations made with the parameters of the DC motor have been used to estimate the shaft torque and angular speed of the motor under certain load torque values. These parameters are important for understanding and controlling the dynamic behavior of the motor.

## A. 2. Measurement of DC Motor Speed (RPM) in Matlab Simulink Environment

The system model is based on the principle of collecting the torques acting on the inertia of the rotor and integrating the acceleration to gain speed. For this purpose, Kirchhoff's laws have been applied to the armature circuit. First, the integrals of the rotational acceleration and the rate of change of the armature current specified in Equations (6) and (7) are modeled [15], [25].

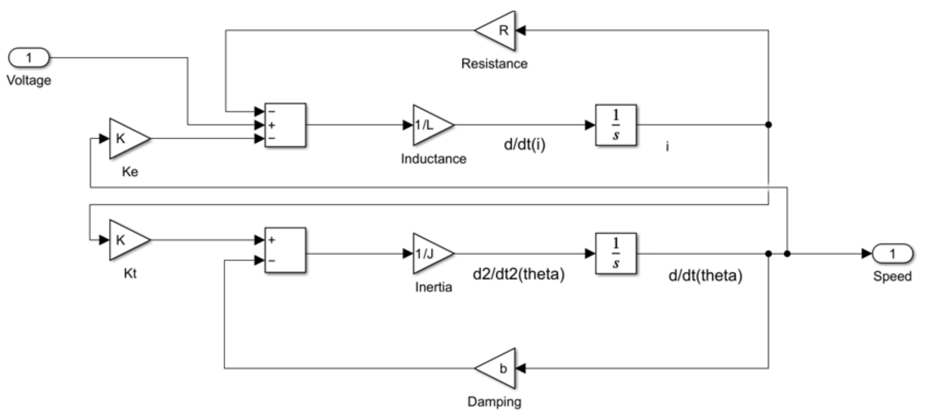
$$\int \frac{d^2\theta}{dt^2} dt = \frac{d\theta}{dt} \tag{6}$$

$$\int \frac{di}{dt} dt = i \tag{7}$$

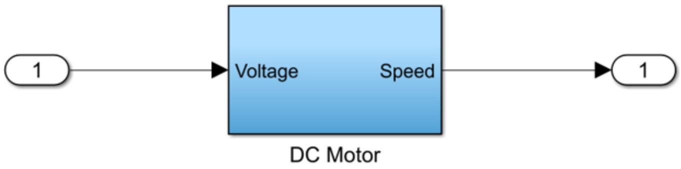
In our study, after using Equations (6) and (7), the formulas in equations (8) and (9) were derived using Newton's and Kirchhoff's laws and applied to the DC motor system [15], [25].

$$J\frac{d^2\theta}{dt^2} = T - b\frac{d\theta}{dt} \quad , \quad \frac{d^2\theta}{dt^2} = \frac{1}{J}(K_t i - b\frac{d\theta}{dt}) \tag{8}$$

$$L\frac{di}{dt} = -Ri + V - e \quad , \quad \frac{di}{dt} = \frac{1}{L}\left(-Ri + V - K_e\frac{d\theta}{dt}\right) \tag{9}$$




***(a)***



*(b)*

***Figure 2. (a), (b)****Matlab Simulink DC motor models used in our study.*

Within the scope of the study, a Matlab Simulink model was created as shown in Figure 2. Then, input and output definitions were made for the system. Voltage was determined as the input signal and speed was determined as the output signal. The Simulink model was run by entering the DC motor parameters into the Matlab command line as given in Table 1.

## A. 3. DC Motor PID Control

In the study, PID controller was used to control the DC motor speed. PID controller is widely used to improve dynamic response and reduce or eliminate steady-state error. The derivative controller adds a finite zero to the open-loop plant transfer function, improving the transient response and stabilizing the system behavior. The integral controller, on the other hand, increases the system type by adding a pole to the origin and minimizes the steady-state error caused by the step function [26]. To determine the coefficients of the PID controller, a new and effective heuristic method (CO) method was used along with a traditional method (PSO). The structure of the PID controller is given in Figure 3.

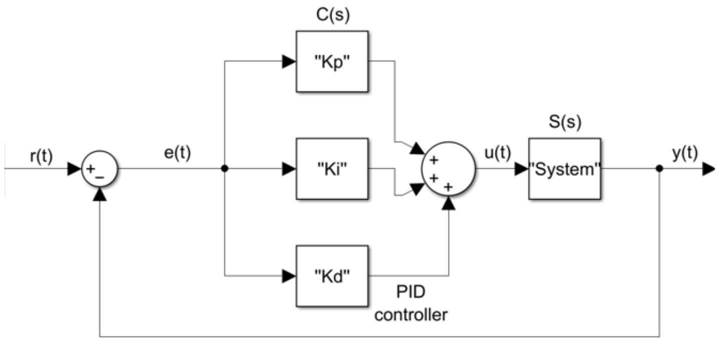



***Figure 3.*** *Structure of PID controller.*

The transfer functions of the PID controller obtained using Figure 3 are obtained as in equations (10) and (11) [15].

$$C(s) = \left(K_P + \frac{K_I}{s} + K_D.S\right)E(s) \tag{10}$$

$$C(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D.s \tag{11}$$

Determining the gains Kp, Ki, and Kd is critical as these coefficients ensure proper control of the system. However, determining these coefficients is quite difficult and it is often not possible to reach a clear conclusion. When selecting the Kp, Ki and Kd gains for the best control performance, performance criteria such as minimum error, minimum overshoot, quick elimination of the error and ensuring system stability should be taken into consideration. These criteria are important to ensure optimal control. In Figure 3, a Matlab simulation model is created using the DC motor parameters in the System section. Then, considering the PID coefficients, the transfer function of the entire system is obtained as in equation (12).

$$T(s) = \frac{Y(s)}{R(s)} = \frac{C(s)*S(s)}{1+C(s)*S(s)} \tag{12}$$

Figure 4 presents the Simulink model designed for measurement of DC motor speed in the MATLAB Simulink environment.

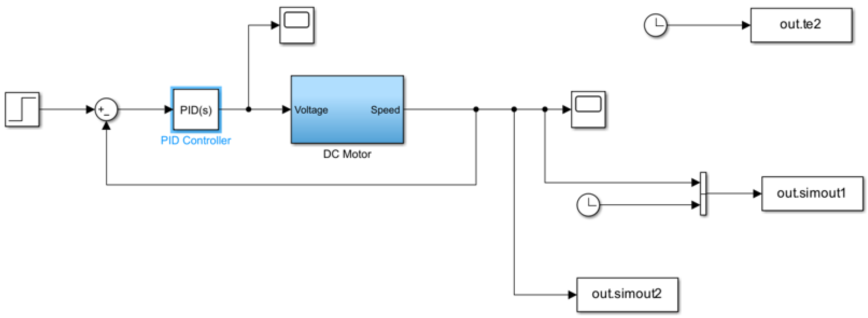



***Figure 4.*** *Simulink model designed for measuring DC motor speed.*

## B. PERFORMANCE INDICES OF CONTROLLERS

DC motor speed control represents an important area for analyzing the efficiency of controllers, focusing on performance indices used to evaluate the effectiveness of control systems. Among the available quantitative measurements of system performance, time integral performance indices occupy an important place among the performance indices commonly used to evaluate the effectiveness of designed control systems. The most used ones are ITAE (Integrated Time Absolute Error), IAE (Integral Absolute Error), ITSE (Integrated Time Squared Error) and ISE (Integral Squared Error) [7], [27]. In the literature, various studies have carried out different analyzes based on commonly used performance indices to evaluate the effectiveness of designed control systems [7], [8], [15], [28], [29]. In our study on DC motor speed control, certain criteria were used to evaluate the performance of the controllers. These criteria are criteria such as ISE, IAE and ITAE. These criteria are expressed by the formulas below [6], [8], [15], [28].

$$ISE = \int_0^\infty e(t)^2\,dt \tag{13}$$

In Equation 13, the *e(t)* function expresses the error function between the output and the desired reference in the control system, while *t* represents the evaluation time. A smaller *ISE* indicates that the control system performs better because it indicates that the difference between actual and desired responses is smaller.

$$IAE = \int_0^\infty |e(t)|\,dt \tag{14}$$

The IAE formula given in Equation 14 is a criterion used to evaluate the performance of a control system. This criterion expresses the time integral of the absolute value difference between the actual and desired responses of the control system. A smaller IAE indicates that the control system performs better because it indicates that the absolute value difference between the actual and desired responses is less.

$$ITAE = \int_0^\infty t\,|e(t)|\,dt \tag{15}$$

The ITAE formula given in Equation 15 is a criterion used to evaluate the performance of a control system. The smaller the ITAE, the better the control system performs because it indicates that the integral of the absolute value difference between the actual and desired responses multiplied by time is lower. This indicates that the control system follows the desired response more quickly and precisely.

## C. METHODS USED IN THE STUDY

In our study, PSO and CO methods are used to determine the parameters of PID controllers. Additionally, the performance of the controllers has been compared using specific benchmarks such as ISE, IAE and ITAE, which are different benchmarks. The results are analyzed by comparing the speed values (rpm) obtained using PSO, CO and PID controllers available in the literature. This study examined the effect of optimized parameters of PID controllers on the performance criteria obtained using different optimization methods.

### C. 1. Particle Swarm Optimization

PSO is an optimization algorithm that mimics natural behavior and is effective in optimizing complex systems and increasing their performance. This algorithm can be used in complex systems such as DC motor speed control to increase the stability of control systems, enable them to follow the desired speed profiles more precisely, and optimize energy efficiency. Many meta-heuristic optimization concepts have proven their effectiveness in solving complex optimization problems, such as PSO, Ant Colony Algorithm, CO, and others. In particular, popular algorithms such as PSO developed by Kennedy and Eberhart (1995) are used in a wide range of areas, and they emphasize that successful results have been achieved in various problems [30], [31]. PSO is an optimization algorithm where particles that make up a population come together to form a swarm. Each particle is assigned a random initial position and velocity and updates its position by converging to the best positioned individual in the swarm. In this way, each particle in the swarm moves in a converging optimization process towards the best solution [32]. In the literature, studies have been conducted on DC motor control using the PSO method [13], [16], [17], [31]. In PSO, the speed of each particle is calculated based on sigmoid function values and the positions of the particles are updated according to these velocities [33]. The speed update is expressed by a specific mathematical operation Equation 16. This process allows the particles to move towards their best positions.

$$v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_1 r_1 \left( pbest_{ij}^{(t)} - x_{ij}^{(t)} \right) + c_2 r_2 \left( gbest^{(t)} - x_{ij}^{(t)} \right) \tag{16}$$

$c_1$ and $c_2$, are acceleration constants, and $r_1$, $r_2$ are numbers that introduce randomness into particle motion. The *w* value in Equation 16 used in the study refers to the inertia weight, and this value must be determined at an appropriate level to increase the performance of the PSO at various stages [33]-[35]. The formula used to calculate the *w* value in Equation 17 is given below:

$$w = (w_1 - w_2) \times \frac{(t_{\max} - t)}{t_{\max}} + w_2 \tag{17}$$

Each particle uses a mechanism based on Equation 18 to continuously update their positions.

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \tag{18}$$

Velocity vectors equation 18 are calculated considering the effects of individual particles and other particles in their environment. This equation expresses changes in speed of particles based on factors such as their positions, speeds, and interactions [35]. Equation 19 defines a velocity update mechanism used in PSO implementation. This equation includes the Sigmoid function to update the decided position vector using the velocity vector of each particle. This update mechanism allows particles to move based on their experience.

$$x_{ij}^{t+1} = \begin{cases} 1 & \text{if} \quad r_{ij} < S(v_{ij}) \\ 0 & \text{in other cases} \end{cases} \tag{19}$$

Equation 20 gives a modified formula for the speed update mechanism.

$$v_i^{(k+1)} = X\left[ v_i^{(k)} + c_1 rand_1 * \left( pbest_i - s_i^{(k)} \right) + c_2 rand_2 * \left( gbest - s_i^{(k)} \right) \right] \tag{20}$$

In Equation 21, $X$ represents the contraction factor.

$$X = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|} \tag{21}$$

PSO is an optimization algorithm that attracts attention with its simple applicability and ease of use. It is widely used in solving various motor problems [36]. PSO is distinctly driven by social response rather than biological evolution like other evolutionary algorithms. The flow chart of the algorithm is designed with a specific order and orientation [37]. The flowchart shown in Figure 5 helps us understand the problem-solving process of PSO step by step and from a scientific perspective [38].

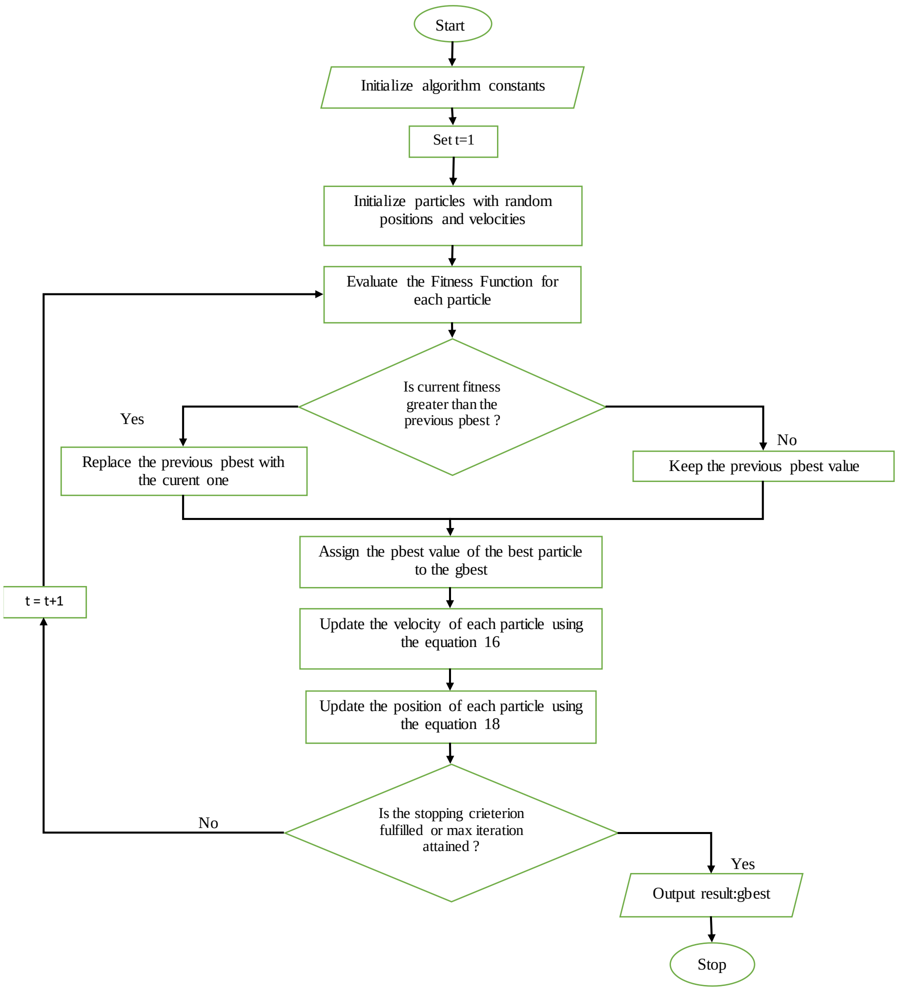



***Figure 5.*** *PSO algorithm flow chart.*

### C. 2. The Cheetah Optimizer Algorithm

CO Algorithm, as an optimization algorithm used in DC motor speed control, is effective in optimizing control parameters. This algorithm can improve the performance of the speed control system, allowing it to better follow the desired speed profiles. Additionally, by increasing energy efficiency, the motor can operate more efficiently. The PSO suffers from biased optimism in regulating its speed and path, which may lead to less accurate results. The algorithm may also suffer from slow convergence in the forward search phase and may become trapped in locally optimal solutions due to its poor local search ability [16], [39], [40]. To overcome these drawbacks and provide better accuracy, a new heuristic method, CO algorithm, was used in our study. CO Algorithm was developed by Akbari et al. [41] as an effective optimization algorithm to mimic the hunting strategies of cheetahs. CO Algorithm uses three main strategies during the hunting process: searching for prey, sitting, and waiting, and attacking. To avoid getting stuck at local optimum points, the algorithm defines leaving the hunt and returning home. Different arrangements of cheetah populations are given in Figure 6 [41], [42].

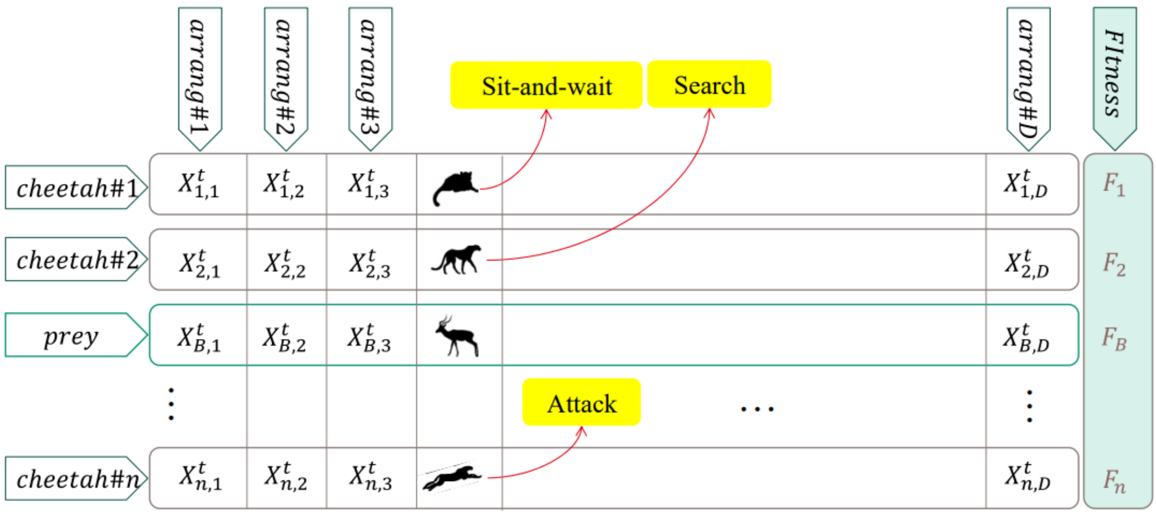



***Figure 6.*** *CO algorithm different editing representation.*

The CO Algorithm stands out as a unique meta-heuristic algorithm, taking direct inspiration from the natural hunting behavior of cheetahs. Below is the mathematical modeling of the CO Algorithm [41]:

**Searching Strategy**

To reach prey, cheetahs observe the environment while sitting or standing. To model this strategy mathematically, the cheetah's current position is defined by the vector $X^t_{i,j}$. The cheetah's position in the next iteration or random search by arbitrary step size for the search is given in Equation 22 [41].

$$X^{t+1}_{i,j} = X^t_{i,j} + \hat{r}^{-1}_{i,j} . \propto^t_{i,j} \tag{22}$$

Here, $X^{t+1}_{i,j}$ and $X^t_{i,j}$ represent the current and next iteration positions of the cheetah at coordinates $i$ and $j$, where $t$ denotes the current hunting time, T is the maximum hunting time, $\hat{r}^{-1}_{i,j}$ is a parameter that randomizes the search direction, and $\propto^t_{i,j}$ specifies the step length for hunting by the cheetahs. Since the process of cheetahs searching for prey in nature is generally slow, the step length $\propto^t_{i,j}$can be adjusted to 0.001x t/T, with the value greater than zero [41].

**Sitting-and-Waiting Strategy**

Cheetahs lie in wait while searching for their prey. This situation is given mathematically in the algorithm in Equation 23.

$$X_{i,j}^{t+1} = X_{i,j}^{t} \tag{23}$$

Here, $X_{i,j}^{t+1}$ and $X_{i,j}^{t}$ represent the updated position of $j$ at each step $i$. This strategy ensures that not every cheetah changes its location at the same time in the search for a solution and can prevent premature convergence [41].

**Attacking Strategy**

Cheetahs use their speed and flexibility when they decide to attack prey. As soon as the potential prey notices the cheetah, it starts to run away, and the cheetah changes its position to catch the prey. This situation is given mathematically in the algorithm in Equation 24.

$$X_{i,j}^{t+1} = X_{B,j}^{t} + \hat{r}_{i,j}^{-1}.B_{i,j}^{t} \tag{24}$$

Here, $X_{B,j}^{t+1}$ defines the current position of prey, $\hat{r}_{i,j}^{-1}$ and $\beta_{i,j}^{t}$ denotes the displacement made by the predator towards the prey in a swift manner. In each iteration, the position of the i-th predator is computed relative to the current position of the prey. The factor $\beta_{i,j}^{t}$ reflects the interaction between the leader and the predators. Thanks to this factor, $X_{k,j}^{t}$ $(k \neq i)$ calculates the positions of the predators in the solution. The term $\hat{r}_{i,j}^{-1}$, which is the return factor, is randomly provided by the equation $\left|r_{i,j}\right|^{\exp(\frac{r_{i,j}}{2})} \sin\left(2\pi r_{i,j}\right)$ to fit the normal distribution. The interaction factor is defined by $\beta_{i,j}^{t}$ in Equation (25) and is expressed by the following formula [41], [42].

$$\beta_{i,j}^{t} = X_{k,j}^{t} - X_{i,j}^{t} \tag{25}$$

**Strategy Selection Mechanism**

During the hunting process, initially a random strategy is applied, but as the cheetah's energy level decreases, the search strategy becomes more preferred over time. In certain cases, despite the initial steps leaning towards a search strategy to achieve better solutions based on high $t$ values, an attack strategy might be preferred. If $r_2$ and $r_3$ are uniformly random numbers chosen from the interval [0, 1], the decision between employing the sit-and-wait strategy or either the searching or attacking strategies hinges on the condition $r_2 \geq r_3$. If $r_2$ is greater than or equal to $r_3$, the sit-and-wait strategy is adopted; otherwise, either the searching or attacking strategy is selected based on a random value derived from Equation 26, using another uniformly random number $r_1$. By adjusting $r_3$, the frequency of switching between the sit-and-wait strategy and the other two strategies can be controlled [41], [42].

$$H = e^{2(1-t/T)}(2r_1 - 1) \tag{26}$$

This scenario emphasizes the cheetah's tendency to adopt a sit-and-wait approach, reducing rapid changes in decision-making variables. Consequently, this strategy boosts the cheetah's success rate in hunting, akin to finding optimal solutions. As the parameter $t$ in function $H$ increases, the likelihood of the cheetah choosing an attacking strategy diminishes due to energy constraints. Nevertheless, there remains a non-zero probability of attacking, mimicking the cheetah's natural behavior. Specifically, when $H \geq r_4$, the cheetah opts for attack mode; otherwise, it switches to search mode. The value of $r_4$, ranging from 0 to 3, plays a crucial role. Higher values of $r_4$ accentuate exploitation, while lower values promote exploration [41].

### C. 3. Solution of DC Motor Speed Control Problem with CO Algorithm

CO Algorithm is a natural optimization method, and a meta-heuristic algorithm used to solve various optimization problems. PID controllers are a control mechanism widely used in control systems. The CO Algorithm has been used to optimize the parameters of PID controllers.

First, during the definition of the problem, it is aimed to optimize the PID controller parameters (Kp, Ki, Kd) for the DC motor speed control system. The purpose of this optimization is to ensure that the system responds quickly, overshoot is minimized, it reaches steady state quickly and oscillations are reduced. The PID equation is created using the error signal and these three parameters to calculate the control signal.

During the parameter setting phase of the CO Algorithm, basic parameters such as population size (N) and maximum number of iterations (T) are determined. Additionally, the unique parameters of the algorithm (speed, acceleration, jump factor, etc.) are also adjusted. While creating the initial population, random initial values are determined for the PID parameters. These values represent the starting points of the optimization process.

In the determination of the fitness function, an error function is selected based on time domain criteria. Commonly used error functions include IAE, ISE, and ITAE. These functions are used to evaluate system performance.

During the operation of the CO Algorithm, a movement mechanism is used that simulates the speed, agility and energy optimization that cheetahs show to catch their prey. Everyone (PID parameter set) updates its speed and position depending on its environment. The algorithm is run up to the maximum number of iterations determined and the positions of individuals are optimized in each iteration.

In the best solution selection phase, the PID parameters of the individual with the best fitness value are selected at the end of the iterations. These optimal PID parameters are integrated into the DC motor speed control system. During the evaluation of system performance, the performance of the system after optimization is evaluated in line with the determined criteria and re-optimization is performed if necessary.

The steps of how the CO Algorithm solves the DC motor speed control problem are given.

| **Step** | **Description** |
|---|---|
| 1. Initial Parameters | Determine cheetah population, search space, and maximum iteration count. |
| 2. Initial Positions | Initialize positions of cheetahs randomly. |
| 3. Fitness Calculation | Calculate fitness values for each cheetah. |
| 4. Search Strategy | Cheetahs update their positions with random step lengths. |
| 5. Waiting Strategy | Cheetahs maintain their positions. |
| 6. Attack Strategy | Cheetahs rapidly change their positions upon spotting prey. |
| 7. Strategy Selection Mechanism | Select strategy randomly with specified values. |
| 8. Iterations | Repeat steps up to the maximum iteration count. |
| 9. Best Solution | Determine the best PID controller gains at the end of maximum iterations. |

# III. RESULTS

## A. MODELING OF DC MOTOR AND COMPARISON OF RESULTS

In this study, DC motor speed control was examined using different control methods, various performance indices and heuristic optimization techniques. PID, PI and PD controllers are used for DC motor speed control. The parameters of these controllers were determined using heuristic optimization methods such as CO and PSO and a non-intuitive method such as MATLAB Tuned. The aim of this study is to determine the most appropriate control method and parameter settings to monitor the desired speed of the DC motor in real time, ensure that it reaches the target speed, and correct fluctuations caused by changing loads or external factors. Criteria such as ISE, IAE and ITAE were used to evaluate the performance of control methods. These criteria made it possible to evaluate how successfully the controllers tracked the desired speed and provided the desired performance. The findings showed how effective each control method and optimization technique were in DC motor speed control and determined which methods provided superior performance under certain conditions. For DC motor speed control, reference speeds of 280 rpm and 560 rpm were used in the Simulink model. Figure 7 shows the Simulink model designed to measure DC motor speed and performance indexes of controllers.

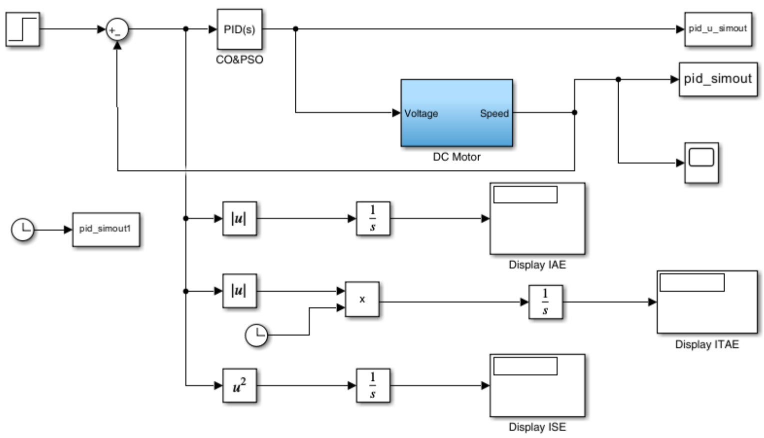



***Figure 7.*** *Simulink model designed for measuring DC motor speed and performance indices of controllers.*

### A. 1. Optimization of PID Parameters, ISE, IAE, and ITAE Performance Indices Using Various Algorithms for a DC Motor Operating at 280 RPM

Table 2 compares the PID parameters, time responses and overshoot values of a DC motor operating at 280 rpm using Matlab Tuned, CO and PSO methods. The CO and PSO algorithms provide better performance in terms of overshoot and settling time compared to Matlab Tuned tuning, but the rise and peak times are slightly longer. Algorithm selection can be made depending on which performance criteria are prioritized for the operation of the DC motor.

***Table 2.*** *Table of PID parameters, time responses and overshoot values of the DC motor at 280 rpm.*

| **Algorithm** | **Kp** | **Ki** | **Kd** | **Time Responses** | | | **Overshoot *(%)*** |
|---|---|---|---|---|---|---|---|
| | | | | Rise Time $t_r$ *(ms)* | Peak Time $t_p$ (*s*) | Settling Time $t_s$ (*s*) | |
| Tuned | 101.3383 | 531.2728 | 2.0227 | 87.703 | 0.183 | 4.8 | 9.989 |
| CO | 90.7824 | 528.1342 | 2.9348 | 98.771 | 0.221 | 3.4 | 3.646 |
| PSO | 86.4183 | 525.9055 | 2.9055 | 103.531 | 0.234 | 3.6 | 3.646 |

The controller optimized with Table 3 CO Algorithm has the lowest error indexes with 17.86 IAE, 2993 ISE and 1.037 ITAE values. This shows that the CO Algorithm best minimizes errors and improves overall performance. In general, the CO Algorithm exhibits the best performance, reaching the lowest IAE, ISE and ITAE values for the DC motor operating at 280 rpm. The PSO Algorithm also gives better results compared to Matlab Tuned, but it is not as effective as the CO Algorithm. These findings show that the CO Algorithm is more successful than other methods in minimizing errors and improving control performance.

***Table 3.*** *IAE, ISE, and ITAE performance indices at 280 RPM for the DC motor.*

| Controller Type | Performance Indices | | |
|---|---|---|---|
| | IAE | ISE | ITAE |
| Tuned | 19.12 | 3133 | 1.432 |
| CO | 17.86 | 2993 | 1.037 |
| PSO | 18.68 | 3070 | 1.188 |

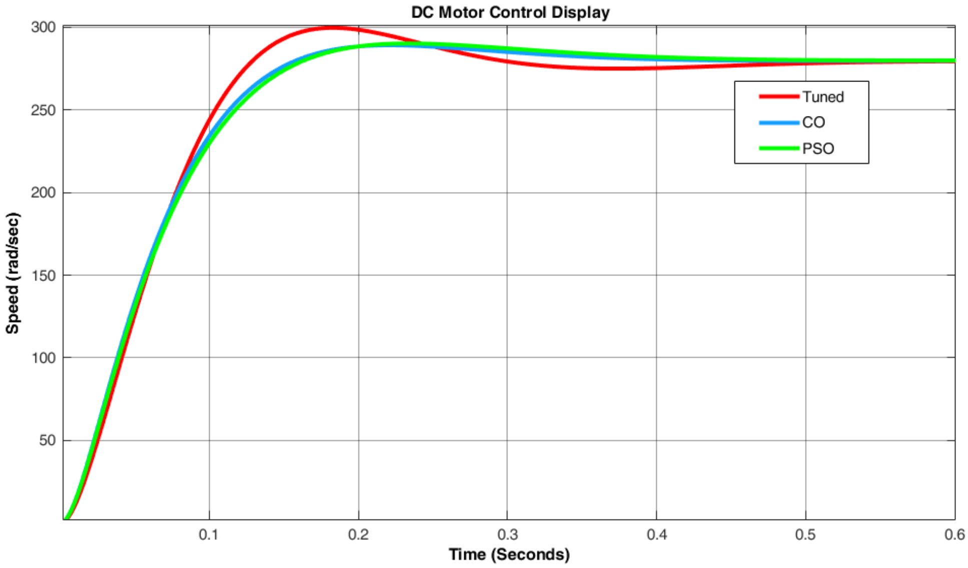



***Figure 8.*** *Control graph of a DC motor running at 280 rpm.*

Figure 8 evaluates the control performance of a DC motor operating at 280 rpm. Matlab Tuned, CO and PSO control method were compared. Determining the most appropriate method for controlling this DC motor operating at 280 rpm depends on the requirements of the application. If faster response is required, the Matlab Tuned method can be preferred; However, if stability and minimum overshoot are more important, the PSO and CO method is more advantageous. For more stable operation of the DC motor, it is more appropriate to use the CO Algorithm with the lowest settling time.

## A. 2. Optimization of PID Parameters, ISE, IAE, and ITAE Performance Indices Using Various Algorithms for a DC Motor Operating at 560 RPM

Table 4 compares the PID parameters, time responses and overshoot values of a DC motor operating at 560 rpm using Matlab Tuned, CO and PSO methods. According to Table 4, the CO method exhibits superior performance in terms of overshoot and settling time compared to Matlab Tuned and PSO algorithm. Although the rise and peak times are slightly longer, the significantly lower overshoot and faster settling time make CO preferable for applications where stability and rapid stabilization are critical.

***Table 4.*** *Table of PID parameters, time responses and overshoot values of the DC motor at 560 rpm.*

| **Algorithm** | **Kp** | **Ki** | **Kd** | **Time Responses** | | | **Overshoot *(%)*** |
|---|---|---|---|---|---|---|---|
| | | | | Rise time $t_r$ *(ms)* | Peak time $t_p$ (ms) | Settling time $t_s$ (s) | |
| Tuned | 887.4 | 490.8 | 137.6 | 5.517 | 14 | 26 | 44.203 |
| CO | 906.9844 | 683.3826 | 92.4342 | 6.903 | 17 | 16 | 38.194 |
| PSO | 965.0637 | 477.8979 | 108.0117 | 6.335 | 16 | 29 | 40.141 |

Table 5 gives the IAE, ISE and ITEA performance indices of the DC motor at 560 rpm. According to Table 5, the CO method exhibits superior performance compared to other algorithms in terms of IAE, ISE and ITAE performance indices. The CO method has the lowest values in terms of absolute value of errors, squared errors and the sum of time-weighted errors, allowing the system to operate more stable, faster and more efficiently. Therefore, the CO method stands out as the best option for DC motor control.

***Table 5.*** *IAE, ISE, and ITAE performance indices at 560 RPM for the DC motor.*

| **Controller Type** | **Performance Indices** | | |
|---|---|---|---|
| | **IAE** | **ISE** | **ITAE** |
| Tuned | 17.09 | 1832 | 4.473 |
| CO | 16.14 | 1798 | 3.802 |
| PSO | 16.55 | 1916 | 4.14 |

Figure 9 evaluates the control performance of a DC motor operating at 560 rpm. Matlab Tuned, CO and PSO control method were compared. The CO method did not perform better than the Tuned and PSO methods in DC motor control. The CO method can ensure more stable and efficient operation of the motor with lower overshoot value, shorter settling time and less fluctuation. Therefore, the CO method stands out as the most suitable option for DC motors operating at 560 rpm.

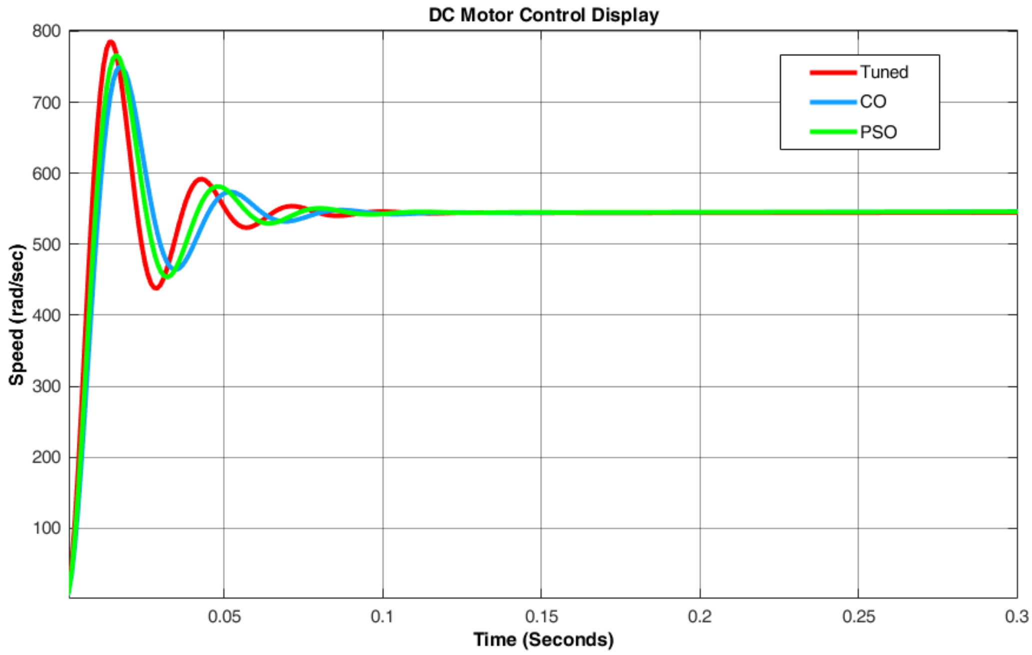



***Figure 9.*** *Control graph of a DC motor running at 560 rpm.*

### A. 3. Optimization of PI Parameters, ISE, IAE, and ITAE Performance Indices Using Various Algorithms for a DC Motor Operating at 280 RPM

Table 6 compares the PI parameters, time responses and overshoot values of a DC motor operating at 280 rpm using Matlab Tuned, CO and PSO methods. In line with these data, it can be said that the CO Algorithm performs quite well in DC motor control with certain PI parameters. It optimizes the speed and stability of the motor, especially with its fast rise and settling time. Therefore, a more effective and efficient motor control can be achieved by choosing the CO Algorithm in certain applications. Considering the settling times, it shows that the CO Algorithm has the shortest settling time, that is, the motor stabilizes to the target speed value the fastest. The CO Algorithm seems to have an advantage over the other two algorithms in providing faster and more effective control.

***Table 6.*** *Table of PI parameters, time responses and overshoot values of the DC motor at 280 rpm.*

| Algorithm | Kp | Ki | Time Responses | | | Overshoot *(%)* |
|---|---|---|---|---|---|---|
| | | | Rise time $t_r$ *(ms)* | Peak time $t_p$ (ms) | Settling time $t_s$ (s) | |
| Tuned | 226.2371 | 1076.69 | 45.929 | 115 | 7.2 | 38.194 |
| CO | 255.0585 | 795.2805 | 44.007 | 108 | 5 | 34.459 |
| PSO | 233.0323 | 966.0349 | 45.693 | 114 | 7.4 | 36.301 |

Table 7 gives the IAE, ISE and ITAE values for the PI control performance indexes of the DC motor at 280 rpm. According to Table 7, the CO Algorithm has the lowest values in all performance indexes. This reveals that the CO Algorithm shows the highest performance and controls the motor speed with minimum error. Therefore, the CO Algorithm for PI control of a DC motor operating at 280 rpm can be considered the most suitable option as it minimizes errors and provides a fast, stable response.

***Table 7.*** *IAE, ISE, and ITAE performance indices at 280 RPM for the DC motor.*

| Controller Type | Performance Indices | | |
|---|---|---|---|
| | IAE | ISE | ITAE |
| Tuned | 24.63 | 3207 | 2.953 |
| CO | 22.73 | 2919 | 2.74 |
| PSO | 23.72 | 3101 | 2.765 |

According to Table 7 and Figure 10, when the performance of the CO Algorithm for PI control of the DC motor at 280 rpm is evaluated, the CO Algorithm is superior to other methods. According to Figure 10, the CO Algorithm exhibits superior performance in terms of both time response and performance indices. These findings show that the CO Algorithm is the most suitable option for controlling the DC motor with the PI control method at a speed of 280 rpm.

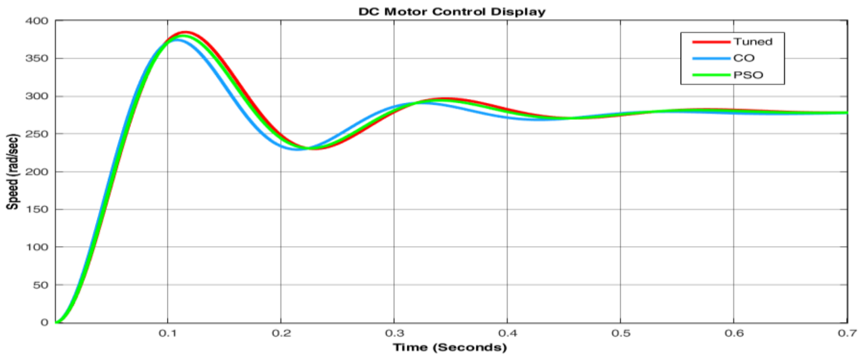



***Figure 10.*** *Control graph of a DC motor running at 280 rpm.*

## A. 4. Optimization of PI Parameters, ISE, IAE, and ITAE Performance Indices Using Various Algorithms for a DC Motor Operating at 560 RPM

Table 8 compares the PI parameters, time responses and overshoot values of a DC motor operating at 560 rpm using Matlab Tuned, CO and PSO methods. When Table 8 and time responses are evaluated, it is seen that although the CO Algorithm has certain disadvantages, it is superior to other algorithms in terms of settling time and overshoot rate. According to Table 8, the CO Algorithm can be considered as the most suitable method for controlling the DC motor at 560 rpm using the PI control method, in terms of minimizing errors and providing a more stable and oscillation-free speed control, compared to Matlab Tuned and PSO methods. Therefore, using the CO Algorithm offers both an effective and efficient solution in DC motor control.

***Table 8.*** *Table of PI parameters, time responses and overshoot values of the DC motor at 560 rpm.*

| **Algorithm** | **Kp** | **Ki** | **Time Responses** Rise time $t_r$ *(ms)* | Peak time $t_p$ (ms) | Settling time $t_s$ (s) | **Overshoot *(%)*** |
|---|---|---|---|---|---|---|
| Tuned | 739.194 | 916.413 | 15.522 | 63 | 7 | 99.764 |
| CO | 421.0687 | 805.4795 | 32.169 | 83 | 5 | 44.203 |
| PSO | 650.4682 | 692.7841 | 15.774 | 67 | 8.2 | 101.428 |

Table 9 gives the IAE, ISE and ITAE values for the PI control performance indexes of the DC motor at 560 rpm. According to Table 9, the Tuned method generally outperforms the CO and PSO methods in terms of IAE, ISE and ITAE indices. While the CO method performs slightly worse than the Tuned method, it generally produces better results than the PSO method. The PSO method generally gives slightly lower performance than the Tuned and CO methods in terms of IAE, ISE and ITAE.

***Table 9.*** *IAE, ISE, and ITAE performance indices at 560 RPM for the DC motor.*

| **Controller Type** | **Performance Indices** IAE | ISE | ITAE |
|---|---|---|---|
| Tuned | 41.25 | 9165 | 5.54 |
| CO | 43.23 | 10130 | 5.778 |
| PSO | 42.18 | 9242 | 6.12 |

Figure 11 evaluates the control performance of a DC motor operating at 560 rpm. In Figure 11, the CO Algorithm provides lower peaks and more controlled initial acceleration, while remaining slightly slower in terms of settling time. This performance of the CO Algorithm can be considered an important tool in the optimization of control systems. However, in certain applications Tuned or PSO methods may also be appropriate.

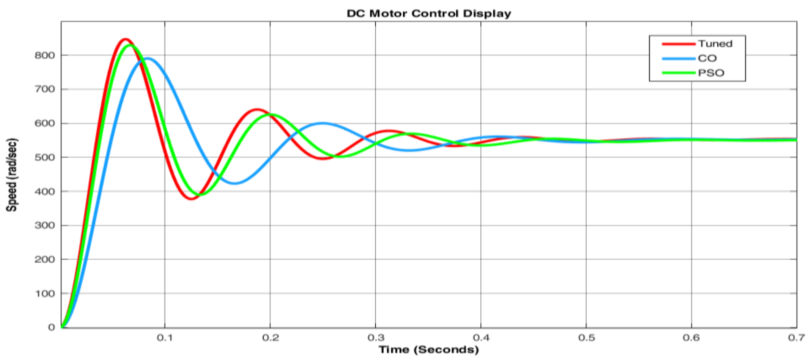



***Figure 11.*** *Control graph of a DC motor running at 560 rpm*

### A. 5. Optimization of PD Parameters, ISE, IAE, and ITAE Performance Indices Using Various Algorithms for a DC Motor Operating at 280 RPM

Table 10 compares the PD parameters, time responses and overshoot values of a DC motor operating at 280 rpm using Matlab Tuned, CO and PSO methods. Table 10 shows that the CO Algorithm exhibits a settling time of 0.25 seconds, which is shorter compared to other methods. This characteristic makes the CO control method preferable for specific applications, particularly where stability and mitigating overshoot are crucial, or when rapid response times are required in the system.

***Table 10.*** *Table of PD parameters, time responses and overshoot values of the DC motor at 280 rpm.*

| **Algorithm** | **Kp** | **Kd** | **Time Responses** Rise time $t_r$ *(ms)* | Peak time $t_p$ (ms) | Settling time $t_s$ (s) | **Overshoot *(%)*** |
|---|---|---|---|---|---|---|
| Tuned | 26458.8891 | 171.3373 | 1.069 | 8 | 0.6 | 163.866 |
| CO | 10079.7118 | 553.0150 | 1.319 | 7 | 0.25 | 131.282 |
| PSO | 20689.1197 | 559.3766 | 0.836 | 6 | 0.3 | 132.289 |

Table 11 gives the IAE, ISE and ITAE values for the PD control performance indexes of the DC motor at 280 rpm. Since the PSO control method generally has the lowest values according to the IAE, ISE and ITAE performance indices, it can offer a better performance for the speed control of the DC motor. The CO method generally exhibits good performance, but compared to the PSO method, it appears to have higher IAE and ISE values in some cases. Matlab Tuned, on the other hand, performs lower than others in terms of performance indices.

***Table 11.*** *IAE, ISE, and ITAE performance indices at 280 RPM for the DC motor.*

| **Controller Type** | **Performance Indices** IAE | **ISE** | **ITAE** |
|---|---|---|---|
| Tuned | 6.173 | 676.3 | 0.2868 |
| CO | 3.868 | 377.6 | 0.3115 |
| PSO | 3.918 | 413 | 0.1947 |

Figure 12 evaluates the control performance of a DC motor operating at 280 rpm. Figure 12 reveals that the CO Algorithm has significant potential for energy efficiency and precise motion control in industrial applications by providing lower peak values, faster settling times, and more stable speed control. This superior performance of the CO Algorithm shows that it is a powerful tool for the optimization of control systems.

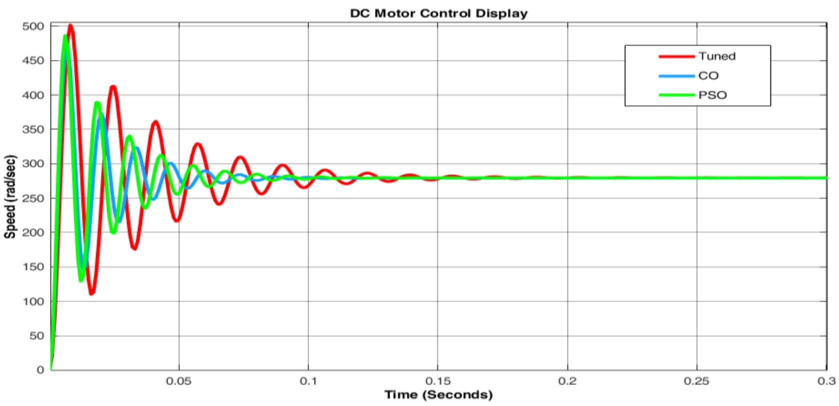



***Figure 12.*** *Control graph of a DC motor running at 280 rpm.*

## A. 6. Optimization of PD Parameters, ISE, IAE, and ITAE Performance Indices Using Various Algorithms for a DC Motor Operating at 560 RPM

Table 12 compares the PD parameters, time responses and overshoot values of a DC motor operating at 560 rpm using Matlab Tuned, CO and PSO methods. According to Table 12, the CO Algorithm provides stable control with low Peak time and Kp, Kd parameters and has the lowest settling time (0.2 seconds). The PSO algorithm shows a higher peak time and a longer settling time of 0.28 seconds compared to the others, while the Tuned algorithm exhibits medium performance. In line with these results, it was found that the CO Algorithm was the most appropriate choice for PD control of a DC motor operating at 560 rpm.

***Table 12.*** *Table of PD parameters, time responses and overshoot values of the DC motor at 560 rpm.*

| **Algorithm** | **Kp** | **Kd** | **Time Responses** | | | **Overshoot** *(%)* |
|---|---|---|---|---|---|---|
| | | | Rise time $t_r$ *(ms)* | Peak time $t_p$ (ms) | Settling time $t_s$ (s) | |
| Tuned | 1818.543 | 28.1982 | 10.283 | 27 | 0.3 | 42.143 |
| CO | 1084.1003 | 55.8350 | 8.906 | 22 | 0.2 | 32.667 |
| PSO | 1107.0990 | 18.8365 | 13.706 | 35 | 0.28 | 34.459 |

Table 13 gives the IAE, ISE and ITAE values for the PD control performance indexes of the DC motor at 560 rpm. According to Table 13, the Tuned algorithm generally shows better performance in terms of IAE and ITAE performance indexes, while the CO Algorithm shows better performance in terms of ISE performance index. The PSO Algorithm generally showed the highest performance indices.

***Table 13.*** *IAE, ISE, and ITAE performance indices at 560 RPM for the DC motor.*

| **Controller Type** | **Performance Indices** | | |
|---|---|---|---|
| | **IAE** | **ISE** | **ITAE** |
| Tuned | 16.93 | 3009 | 3.018 |
| CO | 18.18 | 2341 | 4.742 |
| PSO | 21.65 | 3521 | 4.772 |

Figure 13 evaluates the control performance of a DC motor operating at 560 rpm. Figure 13 clearly shows that the CO Algorithm exhibits superior performance in DC motor speed control compared to the Tuned and PSO methods. The CO Algorithm has significant potential for energy efficiency and precise motion control in industrial applications by providing lower peak values, faster settling times and more stable speed control.

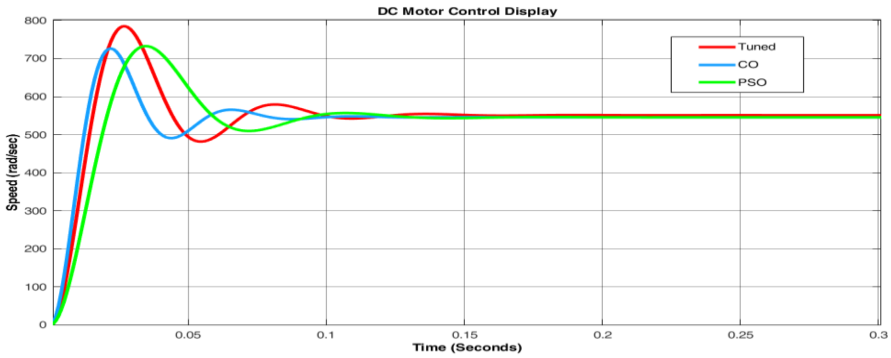



***Figure 13.*** *Control graph of a DC motor running at 560 rpm.*

## A. 7. Performance Evaluation of DC Motor Using PID, PI, and PD Control Methods at Different Speeds

In Figures 14, 15 and 16, DC motor performance evaluation graphs of PID, PI and PD control methods at 280 rpm and 560 rpm speeds are given.

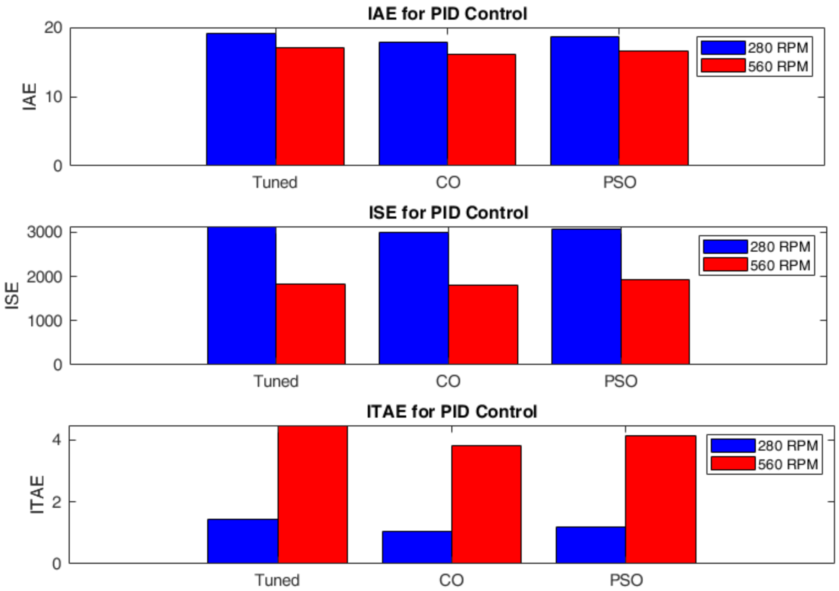



*Figure 14. Performance comparison graph of the PID controller at different speeds.*

When we evaluate the performance of the PID controller at different speeds in Figure 14, the CO method generally shows the best performance at both 280 rpm and 560 rpm speeds. Therefore, it is recommended to use a PID controller optimized with the CO method in DC motor speed control applications.

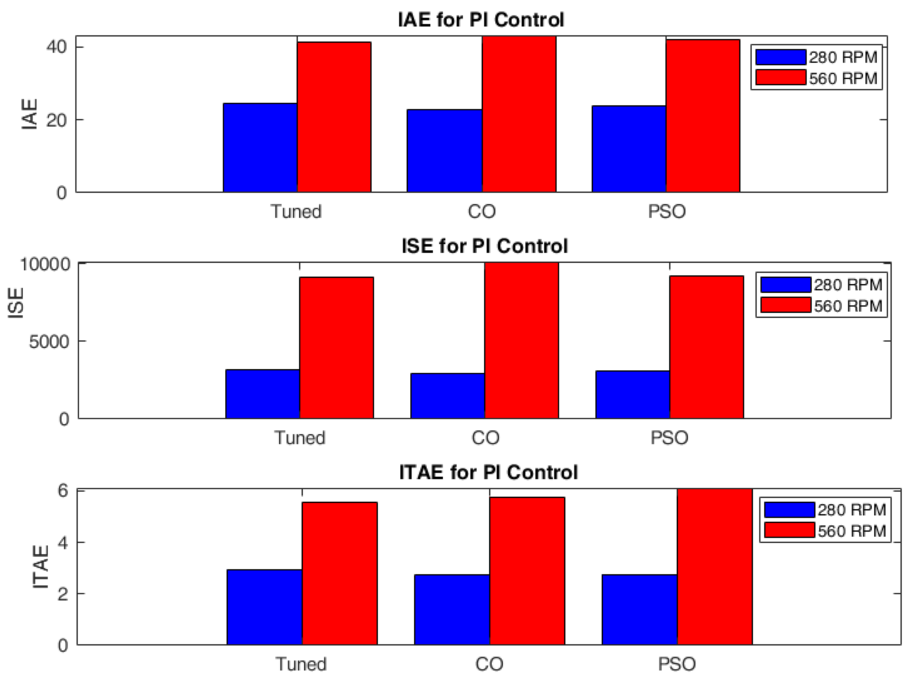



*Figure 15. Performance comparison graph of the PI controller at different speeds.*

When the performance of the PI controller was evaluated according to Figure 15, the CO method at 280 rpm was found to have the best results in IAE, ISE and ITAE metrics. This shows that the CO method provides a more effective PI control performance at low speeds. At 560 rpm, the Tuned method gives the best results in IAE, ISE and ITAE metrics. This means that we can say that the Tuned method provides superior performance at high speeds.

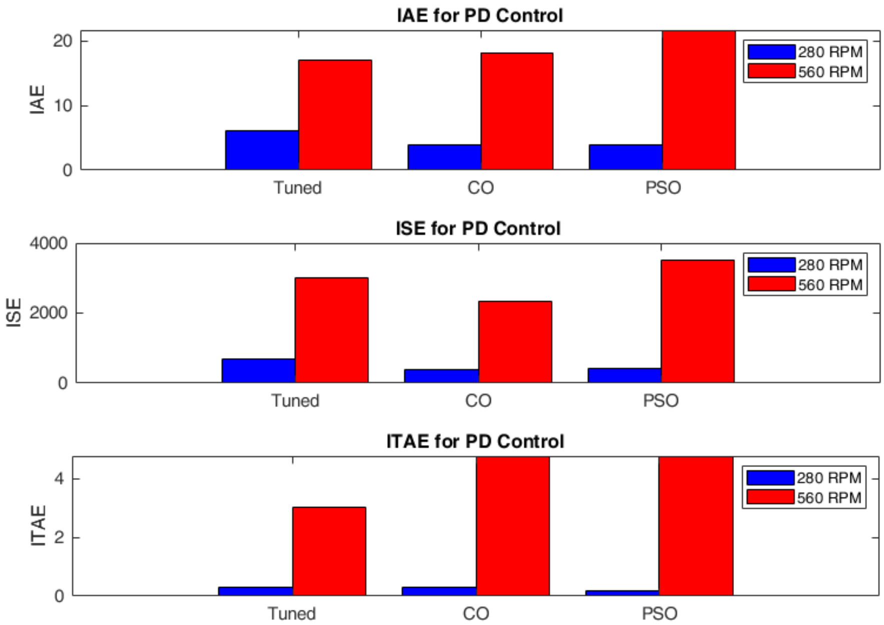



***Figure 16.*** *Performance comparison graph of the PD controller at different speeds.*

According to Figure 16, it has been determined that the CO controller is particularly strong in terms of disturbance rejection. The CO controller's "best performance in terms of disturbance rejection" indicates that the CO controller better maintains or provides the desired output more stably by minimizing external influences. The CO controller has the lowest ISE values at both speeds, meaning it provides the best disturbance rejection. However, the Tuned controller shows the best performance in terms of ITAE at 280 rpm and in terms of IAE and ITAE at 560 rpm. Although the PSO controller generally shows low values in terms of IAE, it has been found to have some disadvantages in terms of other performance measures.

# IV. CONCLUSION

In this study, various control methods and parameter optimization techniques for DC motor speed control were examined. Using PID, PI and PD control methods, the performances of these methods were evaluated with Matlab Tuned, CO Algorithm and PSO techniques. The aim of the study is to monitor the desired speed of the DC motor in real time, ensure that it reaches the target speed, and determine the optimal control method and parameter settings to correct fluctuations caused by changing loads or external factors.

The findings showed that the CO Algorithm showed superior performance compared to other methods in adjusting the parameters of PID, PI and PD controllers. It has been determined that controllers optimized with the CO Algorithm have the lowest error values in terms of performance criteria such as IAE, ISE and ITAE. Especially in the tests performed at 280 rpm and 560 rpm speeds, it was seen that the CO Algorithm exhibited the best performance against variable loads and external factors. It was found that the control parameters determined by the CO Algorithm had lower fluctuation and settling times compared to Matlab Tuned and PSO methods.

This study offers a practical approach for optimizing DC motor speed control in industrial processes. Next-generation and effective optimization methods, such as the CO Algorithm, have been shown to have the potential to significantly improve the performance of control systems. The CO Algorithm stands out especially in applications requiring low error values and high performance and offers new opportunities for future research in this field.

As a result, by comprehensively comparing the performances of different control methods and optimization techniques used in DC motor speed control, it provides important information for the development of more efficient and effective control strategies in industrial applications. Additionally, by demonstrating the potential of next-generation optimization methods such as the CO Algorithm, it directs future research in this field and helps motors determine the most appropriate control methods in DC motor systems facing variable loads and external factors. In this way, valuable contributions are made to increasing energy efficiency in industrial processes, ensuring precise motion control and optimizing overall system performance.

# V. REFERENCES


[1] N. Baćac, V. Slukić, M. Puskaric, B. Štih, E. Kamenar, and S. Zelenika, "Comparison of different DC motor positioning control algorithms," 2014 *37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2014.

[2] S. Tufenkci, B. Baykant Alagoz, G. Kavuran, C. Yeroglu, N. Herencsar, and S. Mahata, "A theoretical demonstration for reinforcement learning of PI control dynamics for optimal speed control of DC motors by using Twin Delay Deep Deterministic Policy Gradient Algorithm," *Expert Syst. Appl.*, vol. 213, pp. 1–16, 2023.

[3] H. X. Wu, S. K. Cheng, and S. M. Cui, "A controller of brushless DC Motor for electric vehicle," In 2004 *12th Symposium on Electromagnetic Launch Technology*, Snowbird, UT, USA. 2004.

[4] N. B. Berahim, "Development of PID Voltage Control for DC Motor Using Arduino," M.S. thesis, Department of Electrical and Electronic Engineering, Tun Hussein Onn University, Malaysia, 2014.

[5] W. Cui, Y. Gong, and M. H. Xu, "A permanent magnet brushless DC motor with bifilar winding for automotive engine cooling application," *IEEE Trans. Magn.*, vol. 48, no. 11, pp. 3348–3351, 2012.

[6] K. Vanchinathan and N. Selvaganesan, "Adaptive fractional order PID controller tuning for brushless DC motor using Artificial Bee Colony algorithm," *Results in Control and Optimization.*, vol. 4, no. 100032, pp. 1–18, 2021.

[7] A. Rajasekhar, R. Kumar Jatoth, and A. Abraham, "Design of intelligent PID/PIλDμ speed controller for chopper fed DC motor drive using opposition based artificial bee colony algorithm," *Eng. Appl. Artif. Intell.*, vol. 29, pp. 13–32, 2014.

[8] S. Khubalkar, A. Junghare, M. Aware, and S. Das, "Modeling and control of a permanent-magnet brushless DC motor drive using a fractional order proportional-integral-derivative controller," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 25, no. 5, pp. 4223–4241, 2017.

[9] A. Bisoi, A. K. Samantaray, and R. Bhattacharyya, "Control strategies for DC motors driving rotor dynamic systems through resonance," *Journal of Sound and Vibration*, vol. 411, pp. 304–327, 2017.

[10] H. Ben Abdeljawed and L. El Amraoui, "Simulation and rapid control prototyping of DC powered universal motors speed control: Towards an efficient operation in future DC homes," *Engineering Science and Technology, an International Journal*, vol. 34, pp. 1–7, 2022.

[11] Q. Zhang, B. Wen, and Y. He, "Rotational speed monitoring of brushed DC motor via current signal," *Measurement*, vol. 184, no.109890, pp. 1–11, 2021.

[12] K. Premkumar, and B. V. Manikandan, "Fuzzy PID supervised online ANFIS based speed controller for brushless dc motor," *Neurocomputing*, vol. 157, pp. 76–90, 2015.

[13] R. Arivalahan, S. Venkatesh, and T. Vinoth, "An effective speed regulation of brushless DC motor using hybrid approach," *Advances in Engineering Software.*, vol. 174, no. 103321, pp. 1–15, 2022.

[14] E. A. Ramadan, M. El-Bardini, and M. A. Fkirin, "Design and FPGA-implementation of an improved adaptive fuzzy logic controller for DC motor speed control," *Ain Shams Engineering Journal*, vol. 5, no. 3, pp. 803–816, 2014.

[15] E. Batan, "Matlab simulink ortamında kullanılabilen arduino temelli kontrol deney seti tasarımı," Yüksek lisans tezi, İmalat Mühendisliği Ana Bilim Dalı, Tarsus Üniversitesi, Mersin, Türkiye, 2019.

[16] K. Premkumar, and B. V. Manikandan, "Speed control of Brushless DC motor using bat algorithm optimized Adaptive Neuro-Fuzzy Inference System," *Applied Soft Computing Journal*, vol. 32, pp. 403–419, 2015.

[17] 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," *Applied Soft Computing Journal*, vol. 75, pp. 29–45, 2019.

[18] A. Bahadir, and Ö. Aydoğdu, "Modeling of a brushless dc motor driven electric vehicle and its pid-fuzzy control with dSPACE," *Sigma Journal of Engineering and Natural Sciences*, vol. 41, no. 1, pp. 156–177, 2023.

[19] B. Suna, "D.C Motorda Kontrol Yöntemlerinin Simülasyonu," Yüksek lisans tezi, Elektrik Mühendisliği Bölümü, Sakarya Üniversitesi, Sakarya, Türkiye, 2009.

[20] C. Guo-qiang, and Z. Zhi-rui, "Mechanical analysis of the industrial robot to upgrade to the gaming robot," *13th Global Congress on Manufacturing and Management, GCMM 2016*, 2017, pp. 1077–1083.

[21] A. Derdiyok, B. Soysal, F. Arslan, Y. Ozoglu, and M. Garip "An adaptive compensator for a vehicle driven by DC motors," J*ournal of The Franklin Institute.*, vol. 342, pp. 273–283, 2005.

[22] T. Szecsi, "A DC motor based cutting tool condition monitoring system," *Journal of Materials Processing Technology*, vol. 93, pp. 350–354, 1999.

[23] A. Bawdaka, and İ. Kaya, "DC Motor Sürücüsü İçin Model Öngörülü Denetleyici Tasarımı," *DÜMF Mühendislik Dergisi*, c. 10, s. 3, ss. 899–910, 2019.

[24] Pololu. (2024, Feb 02). *Metal Gearmotor 25Dx64L mm HP 6V with 48 CPR Encoder* [Online]. Available: https://www.pololu.com/product/2273

[25] D. Tilbury, B. Messner, R. Hill, J. D. Taylor, S. Das, and M. Hagenow. (2024, Feb 02). *Time-Response Analysis of a DC Motor* [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_DCmotor

[26] A. Ozturk, B. Bozali, and S. Tosun, "Investigating voltage and frequency stability problems in the electrical power system using Gravitational Search Algorithms," *Journal of Optoelectronics and Advanced Materials*, vol. 18, no. 1–2, pp. 153–159, 2016.

[27] M. S. Tavazoei, "Notes on integral performance indices in fractional-order control systems," J*ournal of Process Control* vol. 20, no. 3, pp. 285–291, 2010.

[28] A. A. Kesarkar, and N. Selvaganesan, "Tuning of optimal fractional-order PID controller using an artificial bee colony algorithm," *Systems Science and Control Engineering*, vol. 3, no. 1, pp. 99–105, 2015.

[29] B. Bozali, "Elektrik Güç Sistemlerinde Kararlılık Problemlerinin Yerçekimi Algoritması ile İncelenmesi," Yüksek lisans tezi, Elektrik Eğitimi Bölümü, Düzce Üniversitesi, Düzce, Türkiye, 2012.

[30] J. Kennedy, and E. Russell, "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[31] P. Dutta and S. K. Nayak, "Grey Wolf Optimizer Based PID Controller for Speed Control of BLDC Motor," *Journal of Electrical Engineering and Technology*, vol. 16, no. 2, pp. 955–961, 2021.

[32] H. Yiğit, S. Ürgün, and S. Mirjalili, "Comparison of recent metaheuristic optimization algorithms to solve the SHE optimization problem in MLI," *Neural Comput. Appl.*, vol. 35, no. 10, pp. 7369–7388, 2023.

[33] N. Subaş, "Sürekli/İkili Parçacık Sürü Optimizasyonu ve Destek Vektör Makinelerinin Hibrit Kullanımı ile Özellik Seçimi," Yüksek lisans tezi, İstatistik Programı, Mimar Sinan Güzel Sanatlar Üniversitesi, İstanbul, Türkiye, 2019.

[34] B. Bozali, "Türkiye 400 kV'luk Güç Sistemi İçin Sezgisel Yöntemler Kullanılarak Optimal Fazör Ölçüm Birimlerinin Yerleşim Noktalarının Belirlenmesi," Doktora tezi, Elektrik Elektronik Mühendisliği Bölümü, Düzce Üniversitesi, Düzce, Türkiye, 2022.

[35] N. H. A. Rahman, A. F. Zobaa, and M. Theodoridis, "Improved BPSO for optimal PMU placement," *Proceedings of the Universities Power Engineering Conference*, 2015, pp. 1–4.

[36] L. Abualigah, M. A. Elaziz, A. M. Khasawneh, M. Alshinwan, R. A. Ibrahim, M. A. A. Al-qaness, S. Mirjalili, P. Sumari, and A. H. Gandomi, "Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results," *Neural Computing and Applications*, vol. 34, no. 6, pp. 4081–4110, 2022.

[37] Baeldung. (2024, Feb 21). *How does particle swarm optimization work* [Online]. Available: https://www.baeldung.com/cs/pso

[38] H. Jagtap, A. Bewoor, R. Kumar, M. H. Ahmadi, and G. Lorenzini, “Markov-based performance evaluation and availability optimization of the boiler–furnace system in coal-fired thermal power plant using PSO,” *Energy Reports*, vol. 6, pp. 1124–1134, 2020.

[39] E. S. Ali, “Optimization of Power System Stabilizers using BAT search algorithm,” *International Journal of Electrical Power & Energy Systems*, vol. 61, pp. 683–690, 2014.

[40] D. K. Sambariya, and R. Prasad, “Robust tuning of power system stabilizer for small signal stability enhancement using metaheuristic bat algorithm,” *International Journal of Electrical Power & Energy Systems*, vol. 61, pp. 229–238, 2014.

[41] M. A. Akbari, M. Zare, R. Azizipanah-abarghooee, S. Mirjalili, and M. Deriche, “The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems,” *Nature Publishing Group*, UK, Nov. 12:10953, 2022.

[42] Z. A. Memon, M. A. Akbari, and M. Zare, “An improved cheetah optimizer for accurate and reliable estimation of unknown parameters in photovoltaic cell and module models,” *Applied Sciences*, vol. 13, no. 18, 2023.