Black Sea Journal of Engineering and Science
doi: 10.34248/bsengineering.1539753

Open Access Journal
e-ISSN: 2619 – 8991

Research Article
Volume 7 - Issue 6: 1177-1186 / November 2024

AN OPTIMIZED PID CONTROLLER DESING FOR BLDC MOTOR USING NATURE-INSPIRED ALGORITHMS

Batıkan Erdem DEMİR[1]*

[1]*Karabuk University, Faculty of Engineering, Department of Electrical and Electronics, 78050, Karabük, Türkiye*

Abstract: For the optimal control of speed in a brushless DC motor, it is crucial to appropriately adjust the parameters of the PID controller. This study addresses the determination of PID controller parameters using nature-inspired metaheuristic optimization algorithms. Initially, the dynamic model of the brushless DC motor is formulated in the MATLAB/Simulink environment. The grey wolf optimization algorithm, whale optimization algorithm, and firefly algorithm are successively applied to the simulation model to optimize the PID controller parameters. The integral time absolute error objective function is utilized to compare the error performances of these algorithms. Additionally, performance evaluations are conducted concerning parameters such as rise time, settling time, and maximum overshoot. As a result of the comparison based on the fitness criteria, it was determined that the grey wolf optimization algorithm is 35% more successful than the algorithm that provided the next closest result.

Keywords: Brushless DC motor, Nature-inspired algorithms, PID control

***Corresponding author:** Karabuk University, Faculty of Engineering, Department of Electrical and Electronics, 78050, Karabük, Türkiye
E mail: bedemir@karabuk.edu.tr (B. E. DEMİR)
Batıkan Erdem DEMİR https://orcid.org/0000-0001-6400-1510

Received: August 28, 2024
Accepted: October 02, 2024
Published: November 15, 2024

Cite as: Demir BE. 2024. An optimized PID controller desing for BLDC motor using nature-inspired algorithms. BSJ Eng Sci, 7(6): 1177-1186.

1. Introduction

DC motors are efficient electrical machines with ideal operating characteristics for variable speed drives. Their most significant disadvantage is the requirement for a commutator and brushes, both of which wear out and need replacement. Maintenance-free, efficient motors can be obtained by using solid-state switches that take on the role of the commutator and brushes. These motors are called brushless direct current (BLDC) motors (Ehsani et al., 2021; Chittajallu and Lanka, 2023). A BLDC motor consists of three-phase concentrated windings on the stator and permanent magnets on the rotor. Commutation is electronically achieved using a three-phase static inverter powered by a continuous DC source. The Hall sensor serves as a position sensor and is mounted on the stator (Santra et al., 2022; Potnuru et al., 2022; Çetintaş et al., 2023). Due to their characteristics such as high dynamic response, efficiency, silent operation, high torque, and low volume, the use of BLDC motors is increasingly prevalent in many industrial applications. Additionally, in cases where space and weight are critical, they are preferred due to the larger torque provided per motor size. In this study, a control strategy is proposed for the speed control of BLDC motors. PI controllers are simple and widely used for many industrial applications. When BLDC motors are considered as a system, there are uncertainties in their mathematical model due to their advanced nonlinear structures. Therefore, achieving the best performance when tuning PI controller parameters using traditional methods is challenging. Among the traditional tuning methods, Ziegler-Nichols is the most well-known. However, manual tuning can take longer and may potentially damage the hardware during the control process. Moreover, rule-based approaches may not cope well with certain high-level systems. In many current applications, optimization-based techniques developed with a specific objective function are utilized for tuning the parameters of PI or PID controllers.

The study compares the control of a BLDC motor using a traditional PI controller with artificial neural network-based control in a simulation environment (Ch and Palakeerthi, 2015). It was found that the artificial neural network controller exhibited superior performance compared to the PI controller in tracking speed reference changes and stabilizing output speed during load variations. It is addressed the optimization of the commutation angle of the BLDC motor and its impact on speed, current, efficiency, and noise spectrum (Bober, 2017). It is aimed to minimize losses, production costs, and motor volume by optimizing geometric parameters in the design of BLDC motors, utilizing methods such as cuckoo search, genetic algorithm, and particle swarm optimization (Azari et al., 2017).

It is tackled the issue of the inability to adapt to parameter adjustments and system behavior changes encountered in speed control of BLDC motors with a PI controller (Praptodiyono et al., 2020). An optimal adaptive PI controller combining particle swarm optimization and fuzzy logic control was proposed. It is presented the determination of the optimal PI parameters for speed control of a permanent magnet synchronous motor using six swarm intelligence-based optimization algorithms and compared their performances (Aguilar-Mejía et al., 2020). It is utilized algorithms like the equilibrium optimizer (EO), grey wolf optimizer (GWO), and whale optimizer (WO) to maximize the efficiency of BLDC motors and minimize total mass for optimal design (Premkumar et al., 2021).

It is proposed an effective controller design for BLDC motor drivers using the nature-inspired whale optimization algorithm (WOA) (Chittajallu and Lanka, 2023). PI controller parameters were optimized using WOA with integral square error (ISE) as the objective function, resulting in improved performance of the controller. It is addressed the optimization of PID controller parameters controlling the speed of a permanent magnet BLDC motor using metaheuristic algorithms (Abdolhosseini and Abdollahi, 2023). The performance of 14 different algorithms was observed considering settling time, rise time, overshoot, and step response stability of the system. It is suggested a particle swarm algorithm-adjusted fuzzy logic-PI controller for speed control of BLDC motors (Jun et al., 2022). The performance of the optimized PI controller was measured using Integral Absolute Error (ITAE), Integral of Time-Squared Error (ITSE) and ISE error-based performance indicators, demonstrating superior performance.

It is proposed an equilibrium optimization algorithm-based fractional-order PID controller as a solution to sudden set point and parameter changes in traditional PID control of BLDC motor speed (Temir and Durmuş, 2023). The performance of the proposed equilibrium optimization algorithm in optimizing controller parameters was compared with particle swarm optimization (PSO), differential evolution (DE), and golden jackal optimization (GJO) algorithms, showing better results. In this study, a newly proposed metaheuristic optimization algorithm, WOA, was used to optimize PID gains for non-linear BLDC motors.

The optimization algorithms mentioned in the literature review can be applied to various optimization problems, but they exhibit different performance characteristics depending on their attributes and the specific problem being addressed. This performance can vary between good and poor, and understanding this is crucial. From this perspective, there is a need for a comparative study that evaluates some popular optimization algorithms used for optimizing the speed controller of BLDC motors.

In this study, speed control of the BLDC motor is achieved in a simulation environment utilizing the PID method and metaheuristic algorithms. These algorithms comprise the GWO, WOA, and FA (Firefly Algorithm), respectively. The reason for selecting these algorithms, which are frequently encountered in the literature, within the scope of this study is their novel and competitive nature. Performance comparisons of the algorithms used for determining PID parameters are conducted in detail, and the results are presented in the results section. In order to perform a dynamic simulation of a brushless DC motor, its mathematical model must be derived. The subsequent Section 2 provides an in-depth examination of the mathematical model of the BLDC motor, detailing specific aspects related to its dynamic modeling. Section 3 elaborates on the BLDC model created in the MATLAB/Simulink environment and the algorithms utilized. Experimental studies and the obtained findings are analyzed in detail in Section 4, followed by a discussion of the results in Section 5.

2. Materials and Methods

The modeling of the BLDC motor can be developed similarly to a three-phase synchronous machine. Due to the rotor being mounted with permanent magnets, certain dynamic characteristics differ. The equivalent circuit of the BLDC motor, shown in Figure 1, includes a permanent magnet mounted on the rotor and three stator phase windings connected in a star configuration. The motor is powered by a three-phase voltage source. Not only sinusoidal but also square wave or even other waveforms can be applied. The modeling of the BLDC motor is based on several assumptions:

(i) The motor is unsaturated and should operate with rated current.

(ii) The resistances of the three stator phase windings are equal.

(iii) Self-inductance and mutual inductance are constant.

(iv) Iron and copper losses are negligible.

(v) The three phases are balanced.

(vi) Uniform air gap.

(vii) Hysteresis and eddy current losses are neglected.

(viii) Semiconductor switches are ideal.

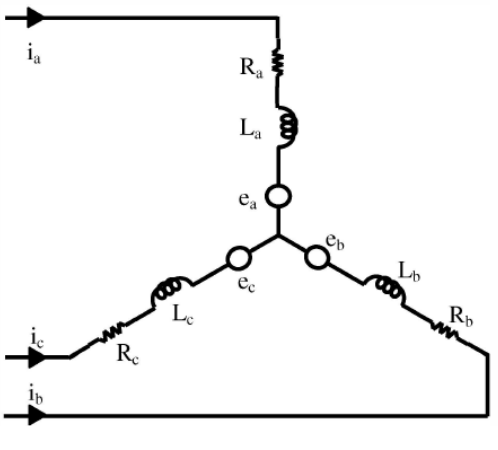

Figure 1. Equivalent circuit of a BLDC motor.

BLDC motors are equipped with a rotor containing permanent magnets and trapezoidal electromotive force (EMF) (Mondal et al., 2015). Typically, a three-phase inverter is used to drive BLDC motors, requiring the utilization of a rotor position sensor element for the operation of the inverter module. The three-phase inverter employs a six-step commutation mechanism to drive the BLDC motor. Position sensors should be employed for the proper commutation sequence, and initially, Hall effect sensors should be utilized. Within each phase, there will be an interval of 120° between executions.

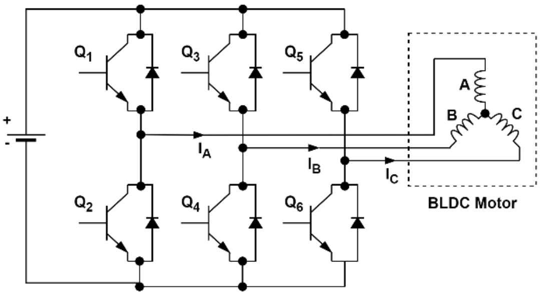

Figure 2. Three phase simplified Inverter.

As depicted in Figure 2, the execution sequence will be 1-4, 1-6, 3-6, 3-2, 5-2, and 5-4. The applied current defining the operation of a BLDC motor should be synchro-nous with the back electromotive force voltage signal. The resultant currents are rectangular in shape, and the motor's switching operation consists of six different steps controlled by a six-step control mechanism. The rotational position can be determined either by position sensors or sensorless methods (Mahmud et al., 2020). The model of the armature winding of a BLDC motor is expressed as follows (equations 1, 2 and 3):

$$V_a = Ri_a + L\frac{di_a}{dt} + e_a \tag{1}$$

$$V_b = Ri_b + L\frac{di_b}{dt} + e_b \tag{2}$$

$$V_c = Ri_c + L\frac{di_c}{dt} + e_c \tag{3}$$

In this context, R_a = R_b = R_c = R represents the stator resistance per phase, L_a = L_b = L_c = L represents the stator inductance per phase, V_a, V_b, and V_c denote the stator phase voltages, i_a, i_b, and i_c represent the stator phase currents, and e_a, e_b, and e_c denote the Back Electromotive Forces (Back EMF) of the motor. The expression of the BLDC motor equation in matrix format is as follows (equation 4):

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} R+pL & 0 & 0 \\ 0 & R+pL & 0 \\ 0 & 0 & R+pL \end{bmatrix} . \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \tag{4}$$

In this context p represents d/dt. When a BLDC motor rotates, each winding generates a voltage known as Back EMF, which opposes the main voltage supplied to the winding according to Lenz's law. The polarity of the Back EMF is opposite to that of the source voltage. The direction of the Back EMF is related to the rotor position function, and there is a 120-degree difference between each phase. The Back EMF is primarily dependent on three factors: (i) the angular speed of the rotor, (ii) the magnetic field generated by the rotor magnets, (iii) the number of turns in the stator windings. The equation for each phase is as follows (equations 5, 6 and 7):

$$e_a = k_e.\omega.f(\theta) \tag{5}$$

$$e_b = k_e.\omega.f\left(\theta - \frac{2\pi}{3}\right) \tag{6}$$

$$e_c = k_e.\omega.f(\theta + \frac{2\pi}{3}) \tag{7}$$

Where, k_e represents the Back EMF constant [V/rad/s]. θ denotes the electrical rotor angle. ω stands for the mechanical speed of the rotor [rad/s]. The permanent magnet also influences the torques generated due to the trapezoidal flux linkage. Considering k_t as the torque constant, the generated torques is as follows (equations 8, 9 and 10):

$$T_a = k_t.f(\theta).i_a \tag{8}$$

$$T_b = k_t.f\left(\theta - \frac{2\pi}{3}\right).i_b \tag{9}$$

$$T_c = k_t.f\left(\theta + \frac{2\pi}{3}\right).i_c \tag{10}$$

The total torque, denoted as T_e, can be represented as the simulation of each phase. Consequently, the equation for total torque can be defined as follows (equation 11):

$$T_e = \frac{(e_a i_a + e_b i_b + e_c i_c)}{\omega} = T_a + T_b + T_c \tag{11}$$

In order to establish the complete mathematical model of an electromechanical system, it is necessary to introduce the motion equations of the motor. At this point, the generated electromagnetic torque in motion can be expressed as follows (equation 12):

$$T_e = J\frac{d\omega}{dt} + T_L + B\omega \tag{12}$$

Where, J represents the rotor inertia [kgm2], and B is the damping constant with T_L load torque, unit N-m. If the Laplace transformation of the equations pertaining to phase a is written, it will be as follows (equations 13, 14, 15 and 16):

$$V(s) = (R + sL) + E_a(s) \tag{13}$$

$$E_a(s) = k_e\omega(s) \tag{14}$$

$$T_e(s) = k_t I_a(s) \tag{15}$$

$$T_e(s) = T_L(s) + (B + sJ)\omega \tag{16}$$

The rotor speed ω is determined as follows using the superposition method (equation 17):

$$\omega(s) = \frac{k_t V(s)}{(R+sL)(B+sJ)+k_t k_e} - \frac{(R+sL)T_L(s)}{(R+sL)(B+sJ)+k_t k_e} \tag{17}$$

Assuming the motor is running without load and considering the conditions B = 0 and T_L = 0, the following reduced equation 18 is obtained.

$$G(s)=\frac{\omega(s)}{V(s)}=\frac{k_t}{sJ(R+sL)+k_t k_e} \quad (18)$$
$$=\frac{1}{k_e\left(s^2\frac{LJ}{k_t k_e}+s\frac{RJ}{k_t k_e}+1\right)}$$

According to the equation above, the mechanical time constant (τ_m) and the electrical time constant (τ_e) are defined as follows (equation 19):

$$\tau_m=\frac{RJ}{k_t k_e},\tau_e \quad (19)$$
$$=\frac{L}{R}$$

With the definitions, the transfer function of the motor is expressed as follows (equation 20):

$$G(s)=\frac{\omega(s)}{V(s)}=\frac{1}{k_e\left(s^2\frac{LJ}{k_t k_e}+s\frac{RJ}{k_t k_e}+1\right)}$$
$$=\frac{1}{k_e(s^2\tau_m\tau_e+s\tau_m+1)} \quad (20)$$

The parameters of the permanent magnet BLDC motor modelled for utilization in the simulation are provided in Table 1.

Table 1. Parameters of the Brushless DC (BLDC) Motor (Khubalkar et al., 2016)

Parameters	Values	Specification
V	48 V	Voltage
R	2.870 Ω	Stator resistance/phase
L	2.7 mH	Stator inductance/phase
J	0.0005 kg-m^2	Moment of inertia
k_t	0.042 Nm/A	Torque constant
k_e	0.042 V/rad/s	Back EMF constant
np	4	Number of poles

After substituting all parameter values from Table 1 into equation 21, the transfer function of the BLDC motor is obtained as follows (Abdolhosseini and Abdollahi, 2023; Khubalkar et al., 2016):

$$G(s)=\frac{\omega(s)}{V(s)} \quad (21)$$
$$=\frac{1}{3.214\cdot10^{-4}s^2+0.3423\,s+0.0042}$$

2.1. Control and Optimization

2.1.1. PID controller

The PID controller, widely employed across diverse systems worldwide, is a linear controller renowned for its typically simplistic and robust architecture, as well as its propensity for delivering satisfactory performance outcomes. The output of the PID controller is defined by the following tracking equation 22.

$$U_{PID}= k_p e(t)+ k_i\int e(t)dt + k_d\frac{de(t)}{dt} \quad (22)$$

Where, the control parameters k_p, k_i, and k_d are proportional, integral, and derivative gains respectively, while *e(t)* represents the error (Joseph et al., 2022; Demir and Demir, 2023). PID controllers measure the output error via a feedback loop and generate the control signal by combining three main control terms: Proportional, Derivative, and Integral. The ability of the PID controller to demonstrate the desired performance relies on the proper tuning of its parameters. Various methods exist for parameter tuning, ranging from classical methods requiring mathematical modeling and analysis of system response to techniques based on metaheuristic optimization algorithms for finding optimal parameter values of PID controllers (Nisi et al., 2019).

Among the classical PID controller tuning methods, the Ziegler-Nichols method, Chien-Hrones-Reswick (CHR) method, Cohen-Coon method, Integral Performance Criterion (IMC), and gain and phase margin-based design methods are the most well-known. These methods attempt to find suitable values for controller parameters utilizing the mathematical model of the system. However, these methods require accurate and complete knowledge of the system model, and sometimes experimentation and adjustments may be needed to achieve the desired performance.

PID controller tuning methods based on optimization algorithms use metaheuristic algorithms to optimize controller parameters. These methods typically require the system's mathematical model or detailed analysis. Additionally, they can automatically adjust controller parameters. However, the optimization process can sometimes lead to computational complexity and require high processing power. The choice of which PID tuning method to use should be determined based on system characteristics, performance requirements, and available data (Águila-León et al., 2020).

Figure 3 illustrates the tuning structure of the PID controller. The adjustment of k_p, k_i, and k_d parameters of the PID controller using an intelligent metaheuristic to optimize the objective function is depicted in Figure 3. The output of the controller (control voltage) controls the speed of the motor.

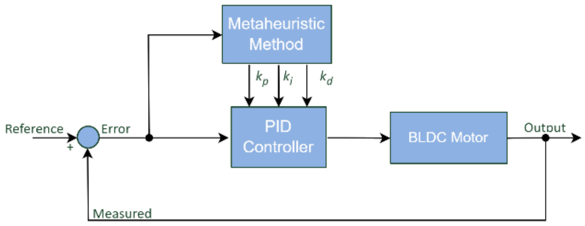

Figure 3. The block diagram of the control system.

2.1.2. Optimization

Optimization problem solutions that satisfy all constraints are feasible solutions. Optimal solutions may encompass objectives such as minimizing the cost of a process or maximizing the efficiency of a system. Control theory deals with dynamic systems and optimizations over time. Therefore, the aim is to find a control for a dynamic system, thus optimizing an objective function. Nature-inspired metaheuristic optimization algorithms can be employed for tuning parameters in PID control applications. Common tasks of these optimization algorithms include randomly selecting initial solutions, evolving solutions based on a fitness function, eliminating the worst solutions, and generating new solutions. The manner in which new solutions are generated constitutes a fundamental difference among these algorithms. It may be desired to assess the im-pact of these algorithms on control. In this case, system performances such as over-shoot and settling time are considered. In multi-objective problems, dynamic error-based performance indices are used instead of multiple fitness functions. These include IAE, ITAE, ISE, and ITSE (Demir and Demir, 2023). In this study, the ITAE performance criterion given in equation 23 is determined as the fitness function to be minimized.

$$f_{ITAE} = \int_0^T t|e(t)|dt \tag{23}$$

When compared with performance indices such as the IAE or the ISE, the ITAE performance index exhibits smaller oscillations and overshoots. Moreover, it possesses greater sensitivity and better selectivity. Additionally, it enjoys computational advantages over the ITSE index. The speed control of a BLDC motor is treated as an optimization problem. This is assessed based on a fitness function, and the optimal parameters are obtained as proportional, integral, and derivative gains. Each problem response is evaluated with optimization algorithms to determine the three parameters that yield the best response as the PID index. In this study, the cuckoo, whale, and firefly algorithms are employed. Subsequently, nature-inspired algorithms utilized for PID index optimization are introduced in the following section.

2.2.3. Grey wolf optimization algorithm

The grey wolf optimization algorithm is based on three fundamental behaviors: hunting, ranking, and mating. A pack of wolves interacts with each other to optimize these behaviors. Hunting behavior involves exploring the search space to reach the best position. Grey wolves encircle prey during the hunt. In order to mathematically model encircling behavior, the following equations 24 and 25 are proposed:

$$\vec{D} = |\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)| \tag{24}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.\vec{D} \tag{25}$$

In the equation t, expresses the current iteration, $\vec{A}$ and $\vec{C}$ coefficient vectors, and $\overrightarrow{X_p}$ position vector of the prey and $\vec{X}$ indicates the position vector of a grey wolf. Vectors are calculated in the below equations 26 and 27:

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a} \tag{26}$$

$$\vec{C} = 2.\overrightarrow{r_2} \tag{27}$$

Where components of $\vec{a}$ are linearly decreased from 2 to 0 over the course of iterations and $\overrightarrow{r_1}$, $\overrightarrow{r_2}$ are random vectors in [0, 1]. Ranking behavior allows wolves to adjust their positions and strengths relative to each other. On the other hand, mating behavior facilitates the creation of new wolves and the expansion of the solution space. In order to mathematically simulate the hunting behavior of grey wolves, we assume that the alpha (best candidate solution), beta, and delta have better knowledge about the potential location of prey. Therefore, we save the first three best solutions obtained so far and require the other search agents (including the omegas) to update their positions according to the position of the best search agents. The following equations 28, 29 and 30 are proposed in this regard:

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}|, \overrightarrow{D_\beta} = |\overrightarrow{C_2}.\overrightarrow{X_\beta} - \vec{X}|, \overrightarrow{D_\delta} = |\overrightarrow{C_3}.\overrightarrow{X_\delta} - \vec{X}| \tag{28}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}.(\overrightarrow{D_\alpha}), \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}.(\overrightarrow{D_\beta}), \overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}.(\overrightarrow{D_\delta}) \tag{29}$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \tag{30}$$

The pseudo-code for GWO is presented in Algorithm 1 (Mirjalili et al., 2014).

2.2.4. Whale optimization algorithm

The whale optimization algorithm operates by maintaining a population of candidate solutions (whales) and iteratively improving them over several generations. In each iteration, the position of each whale is updated using two random vectors, de-noted as A and C, along with a specific formula simulating whale hunting behavior. In order to mathematically model this behavior, the following equations 31 and 32 are proposed:

$$\vec{D} = |\vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t)| \tag{31}$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A}.\vec{D} \tag{32}$$

Where t indicates the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, X^* is the position vector of the best solution obtained so far, $\vec{X}$ is the position vector. The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows (equations 33 and 34):

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{33}$$

$$\vec{C} = 2.\vec{r} \tag{34}$$

Where $\vec{a}$ is linearly decreased from 2 to 0 over the course of iterations (in both exploration and exploitation phases) and $\vec{r}$ is a random vector in [0,1]. Subsequently, randomness is introduced into the search process by applying a randomization operator to each whale's new position. The fitness value of each whale's new position is computed and compared with the current best position. If the new position is superior to the current best position, it is assigned as the new best position. This process continues until a specific termination criterion is met. The final solution corresponds to the best current position. The detailed steps of WOA are presented in Algorithm 2 (Mirjalili and Lewis, 2016).

2.2.5. Firefly Algorithm

In the firefly algorithm, the disparity in luminosity among fireflies is associated with distance and the density of the environment. This variance delineates the movement of each firefly. The primary objective of the algorithm is to identify the position with the least disparity in luminosity among numerous fireflies. This position nearly represents an optimal solution. Utilizing luminosity differentials as motion vectors, the algorithm establishes a swarm of fireflies navigating within the solution space. Each firefly moves towards the luminosity value of the nearest firefly, thus progressively converging the swarm towards the optimum solution. The proportionality of attractiveness depends on the intensity of light seen by another firefly. Thus, the attractive-ness variation β on r distance is expressed as (equation 35):

$$\beta = \beta_0 e^{-\gamma r^2} \tag{35}$$

Here r equals to 0 and β_0 denotes the attractiveness. Let us assume that there are two fireflies x_i and x_j. Movement of *i*th firefly towards *j*th firefly due to more brightness is defined as (equation 36):

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha_t \epsilon_i^t \tag{36}$$

Where, the second part of Equation 36 explains the attraction. The third part of Equation 36 is for randomization with α_t. ϵ_i^t defines the vector of random numbers generated by either Gaussian distribution or it can be introduced by uniform distribution with time t. The pseudocode for FA is presented in Algorithm 3 (Kumar and Kumar, 2021).

Algorithm 1. GWO Pseudocode

```
Initialize a population of grey wolves randomly
Evaluate fitness function to determine the fitness value of each wolf
Set alpha, beta, and delta as the three best wolves in the population
while termination criterion is not met do
    for each wolf i in the population do
        Calculate the distance between the current wolf i and alpha, beta, and delta
        Update the position of wolf i
        Apply a randomization operator to the new position of wolf i
        Evaluate the fitness value of the new position of wolf i
        if the new position is better than the position of alpha, beta or delta then
            if the new position is better than alpha then
                Set the new position as alpha
            else if the new position is better than beta then
                Set the new position as beta
            else
                Set the new position as delta
            end if
        end if
    end for
end while
return alpha
```

Algorithm 2. WOA Pseudocode

```
Initialize a population of whales randomly
Evaluate fitness function to determine the fitness value of each whale
Set the current best position as the position of the whale with the best fitness value
while termination criterion is not met do
        for each wolf i in the population do
        Generate random vector A and C
        if |A| < 0.5 then
            if |C| < 1 then,
                Update the position of whale i towards the current best position
            else
                Update the position of whale i randomly within the search space
            end if
        else
            Update the position of whale i towards a randomly selected whale j
        end if
        Apply a randomization operator to the new position of whale i
        Evaluate the fitness value of the new position of whale i
        if the new position is better than the current best position then
            Set the new position as the current best position
        end if
   end for
end while
return current best position
```

Algorithm 3. FA Pseudocode

```
Initialize a population of fireflies randomly
Evaluate fitness function to determine the fitness value of each firefly
Set the best solution as the current global best
while termination criterion is not met do
for each firefly i do
for each firefly j do
if firefly j is brighter than firefly i then
Compute the distance $r_{ij}$ between fireflies i and j
Compute the attractiveness $\beta(r_{ij})$ of firefly j towards firefly i
Move firefly i towards firefly j with a step size $\alpha\beta(r_{ij})$
end if
end for
Update the global best solution
end while
return the best solution found
```

4. Results and Discussion

In academic studies, the comparative utilization of different approaches is a factor that influences the value of the study. In this study, GWO, WOA, and FA metaheuristic algorithms have been comparatively employed for the purpose of tuning PID controller parameters to minimize error. After modeling the BLDC motor in MATLAB/Simulink, simulations were conducted using different optimization methods for the reference speed value. The MATLAB/Simulink model of the developed BLDC motor speed control system within the scope of this study is presented in Figure 4.

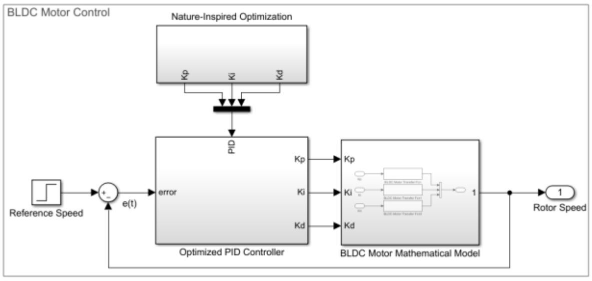

Figure 4. MATLAB / Simulink Model of BLDC Motor Speed Control System.

The population sizes were adjusted to be equal, as the number of individuals in the optimization algorithms affects the simulation results. Common parameters of the optimization algorithms, namely a population size of 50 and a maximum iteration number of 50, were selected. For the accuracy of the results, each optimization algorithm was executed 20 times. The reference value for the output speed (ω) to be used in the simulation was set to 500 rpm. The range of values for the PID controller parameters is also between [0 - 1]. The graph showing the variation in the fitness value of the optimization algorithms for the specified reference values is presented in Figure 5.

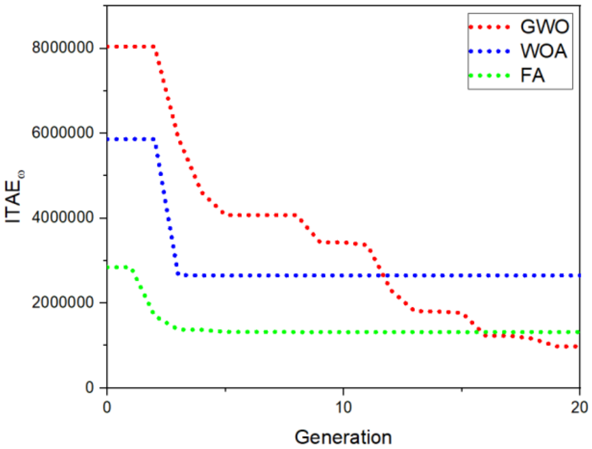

Figure 5. Fitness results.

Furthermore, the ITAE results of each algorithm are presented in Table 2, while the control parameters determined by the optimization algorithms are provided in Table 3.

Table 2. Fitness (ITAE)

Refence Speed	GWO	WOA	FA
500 rpm	977125.99	2654118.29	1320036.50

Table 3. PID parameter in optimization algorithm.

Algorithm	k_p	k_i	k_d
GWO	0.7870	0.0059	0
WOA	0.9699	0.3268	0.0893
FA	0.9241	0.8793	0.0483

Nature-inspired optimization algorithms use stochastic operators. Therefore, the algorithms can produce different results each time they are run. The results in Table 2 were obtained by selecting the best outcomes after running each algorithm 20 times. The optimal outcome was achieved when the speed (ω) control parameter of GWO was evaluated according to the fitness function ITAE. According to the data in Table 2, the results of WOA and FA differ significantly from those of GWO. The disparity be-tween FA in second place and GWO is approximately 35%, whereas the disparity be-tween WOA in third place and GWO is approximately 2.7-fold. The speed (ω) graph obtained using PID parameters determined based on the best outcomes of various optimization algorithms is presented in Figure 6.

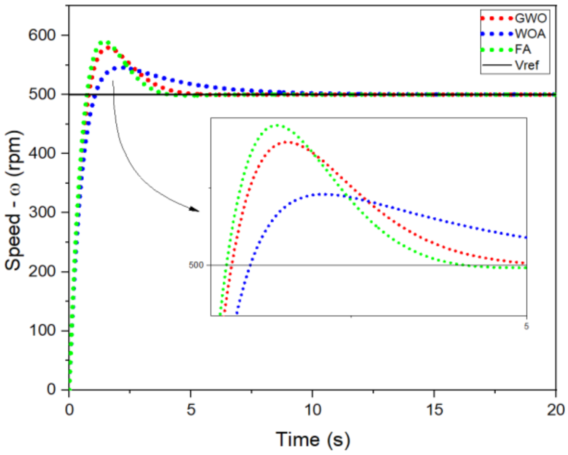

Figure 6. Speed (ω) graphs of BLDC motor. GWO: grey wolf optimizer; WOA: whale optimization algorithm; FA: firefly algorithm.

An optimization algorithm repeats a standard process in each iteration while searching for the optimum value in the solution space. It reaches the optimum value after several iterations. A small number of iterations takes less time, indicating that the algorithm is fast. A speed comparison was made for GWO, WOA, and FA based on the number of iterations required to reach the optimum result. The fastest algorithm is FA, reaching the best result in the 8th iteration. WOA is the second fastest, reaching the best result in the 9th iteration. GWO is the third, reaching the best result in the 19th iteration.

The performance comparisons of optimization algorithms based on the speed graph are provided according to the values of overshoot, settling time, and rise time in Table 4. The minimum overshoot in the speed (ω) response occurs in WOA, while the largest overshoot is observed in FA. According to the data in Table 4, there is an ap-proximate twofold difference between the lowest and highest overshoot values. The overshoot values of GWO are close to those of FA.

The best result in terms of settling time is obtained from FA, whereas the longest settling time is observed in WOA. The difference between the shortest and longest settling times is 91%. The difference between GWO, which ranks second, and FA is 17%. The difference between WOA, which ranks third, and GWO is 63%. In terms of rise time, the best result is obtained from FA, while the longest rise time is observed in WOA. The difference between the shortest and longest rise times is 35%. The difference between GWO, which ranks second, and FA is 9%.

Table 4. Controller performance in the speed parameter (ω)

Algorithm	Overshoot (%)	Settling Time (s)	Rising Time (s)
GWO	15.8585	3.9312	0.5912
WOA	9.1307	6.4128	0.7335
FA	18.0210	3.3566	0.5421

5. Conclusions

In this study, a BLDC motor model with a PID controller was developed and simulated. Controller parameters for the output speed of the developed model were determined using nature-inspired algorithms, namely GWO, WOA, and FA. The performance of these optimization algorithms was compared based on ITAE selected as the fitness function. For the reliability of the obtained results, each algorithm was run 20 times.

According to the data in Table 2, the most successful result in terms of controlling the output speed (ω) was achieved with GWO. GWO was 35% and 2.7-fold more successful than the method that gave the closest and farthest results, respectively. Additionally, controller performances optimized based on overshoot, settling time, and rise time values in the obtained graphs were also examined.

According to the data in Table 4, the best result in terms of overshoot was obtained with WOA. WOA was 1.7-fold and 1.9-fold more successful than the methods that gave the closest and farthest results, respectively. Regarding settling and rise times, the best performances were achieved with FA. In terms of settling and rise times, FA was 17% and 9% more successful than the method that gave the closest results, respectively.

The primary objective of obtaining these controller parameters through optimization methods is to develop an optimal controller without being subject to system limitations. This objective has been achieved as a result of this study. In future research, the comparative use of multi-objective functions and hybrid optimization algorithms is planned for the optimization of control parameters in different types of engines.

Author Contributions

The percentage of the author contributions is presented below. The author reviewed and approved the final version of the manuscript.

	B.E.D.
C	100
D	100
S	100
DCP	100
DAI	100
L	100
W	100
CR	100
SR	100
PM	100
FA	100

C=Concept, D= design, S= supervision, DCP= data collection and/or processing, DAI= data analysis and/or interpretation, L= literature search, W= writing, CR= critical review, SR= submission and revision, PM= project management, FA= funding acquisition.

Conflict of Interest

The author declared that there is no conflict of interest.

Ethical Consideration

Ethics committee approval was not required for this study because of there was no study on animals or humans.

References

Abdolhosseini M, Abdollahi, R. 2023. Performance analysis of PID controller-based metaheuristic optimisation algorithms for BLDC motor. Australian J Elect Electron Engin, 20(4): 400–411.

Águila-León J, Chiñas-Palacios CD, Vargas-salgado C, Hurtado-perez E, García, EXM. 2020. Optimal PID parameters tunning for a DC-DC boost converter: a performance comparative using grey wolf optimizer, particle swarm optimization and genetic algorithms. In 2020 IEEE Conference on Technologies for Sustainability (SusTech), Santa Ana, CA, USA, pp: 1–6.

Aguilar-Mejía O, Minor-Popocatl H, Tapia-Olvera R. 2020. Comparison and ranking of metaheuristic techniques for optimization of PI controllers in a machine drive system. Applied Sci, 10(18): 6592.

Azari MN, Samami M, Pahnehkolaei SMA. 2017. Optimal design of a brushless DC motor , by cuckoo optimization algorithm. Inter J Engin, 30(5): 668–677.

Bober P. 2017. Measurement of objective function for BLDC motor optimization. Acta Electrotech Inform, 17(4): 43–49.

Çetintaş OG, Akgül K, Ergene LT. 2023. Position Sensorless speed control of BLDC motor with using back-EMF method. In 2023 14th Inter Conference on Electrical and Electronics Engin (ELECO), pp: 1–6.

Ch L, Palakeerthi R. 2015. BLDC drive control using artificial intelligence technique. Inter J Computer Applicat, 118(4): 5–9.

Chittajallu T, Lanka, RS. 2023. An effective controller design for BLDC motor drive with nature inspired heuristic algorithm. In: International Conference on Artificial Intelligence Techniques for Electrical Engin Systems, pp: 268–280.

Demir BE, Demir F. 2023. Comparison of metaheuristic optimization algorithms for quadrotor PID controllers. Tehnički Vjesnik, 30(4): 1096–1103.

Ehsani M, Singh KV, Bansal HO, Mehrjardi RT. 2021. State of the Art and trends in electric and hybrid electric vehicles. Proceed IEEE, 109(6): 967–984.

Joseph SB, Dada EG, Abidemi A, Oyewola DO, Khammas BM. 2022. Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems. Heliyon, 8(5): e09399.

Jun S, Qingtao M, Weifeng C, Lintao Z. 2022. Optimizing BLDC motor drive performance using particle swarm algorithm-tuned fuzzy logic controller. SN Applied Sci, 4(293).

Khubalkar SW, Chopade AS, Junghare SA, Aware MV. 2016. Design and tuning of fractional order PID controller for speed control of permanent magnet brushless DC motor. In: 2016 IEEE First Inter Conference on Control, Measurement and Instrumentation (CMI), pp: 320–326.

Kumar V, Kumar D. 2021. A systematic review on firefly algorithm: past, present, and future. Archiv Computat Methods Engin, 28(4): 3269–3291.

Mahmud M, Motakabber SMA, Alam AHMZ, Nordin AN. 2020. Control BLDC motor speed using PID controller. Inter J Adv Comput Sci Applicat, 11(3): 477–481.

Mirjalili S, Lewis A. 2016. The whale optimization algorithm. Adv Engin Software, 95: 51–67.

Mirjalili S, Mirjalili SM, Lewis A. 2014. Grey wolf optimizer. Adv Engin Software, 69: 46–61.

Mondal S, Mitra A, Chattopadhyay M. 2015. Mathematical modeling and simulation of brushless DC motor with ideal back EMF for a precision speed control. In: 2015 IEEE Inter Conference on Electrical, Computer and Communication Technologies (ICECCT), pp: 1–5.

Nisi K, Nagaraj B, Agalya A. 2019. Tuning of a PID controller using evolutionary multi objective optimization methodologies and application to the pulp and paper industry. Inter J Machine Learn Cybernet, 10: 2015–2025.

Potnuru D, Ayyarao TSLV, Kumar LVS, Kumar YVP, Pradeep DJ, Reddy CP. 2022. Salp swarm algorithm based optimal speed control for electric vehicles. Inter J Power Electron Drive Systems, 13(2): 755–763.

Praptodiyono S, Maghfiroh H, Hermanu C. 2020. BLDC motor control optimization using optimal adaptive PI algorithm. J Elektron Dan Telekom, 20(2): 47–52.

Premkumar M, Sowmya R, Jangir P, Nisar KS, Aldhaifallah M. 2021. A new metaheuristic optimization algorithms for brushless direct current wheel motor design problem. Comput, Mater Continua, 62(2): 2227-2242.

Santra SB, Chatterjee A, Chatterjee D, Padmanaban S, Bhattacharya K. 2022. High efficiency operation of brushless DC motor drive using optimized harmonic minimization based switching technique. IEEE Transact Indust Applicat, 58(2): 2122–2133.

Temir A, Durmuş B. 2023. Equilibrium optimizer based fractional order PID control of brushless DC motor. European J Sci Technol, 51: 153–161.