Research Article

Academic Platform Journal of Engineering and Smart Systems (APJESS) 11(3), 107-117, 2023

Received: 17-Mar-2023 Accepted: 31-Jul-2023

homepage: https://dergipark.org.tr/tr/pub/apjess

https://doi.org/10.21541/apjess.1266949

Optimizing PID Gains of a Vehicle using the State-of-the-art Metaheuristic Methods

[1]Mustafa Atakan AFŞAR, [*2]Hilal ARSLAN

[1] Department of Aerospace Engineering, Ankara Yıldırım Beyazıt University, Ankara, Türkiye, atakanafsar@gmail.com

[2] Department of Software Engineering, Ankara Yıldırım Beyazıt University, Ankara, Türkiye, hilalarslan@aybu.edu.tr

Abstract

PID controllers are important control methods that are widely used in industrial processes. Proper tuning of PID gains is critical for achieving state-of-the-art system performance. Therefore, the optimization of PID gains is an important research topic in control engineering. In this study, PID controller gains are automatically tuned using metaheuristic optimization methods. These methods use an iterative approach to calculate optimal values of PID controller gains based on different optimization techniques. The interaction between artificial intelligence and control systems requires a multidimensional approach across different disciplines. In the study, we perform Particle Swarm Optimization, Gray Wolf Optimization, Whale Optimization Algorithm, Firefly Algorithm, Harris Hawks Optimization, Artificial Hummingbird Algorithm, and African Vulture Optimization Algorithm to determine PID gains. In the simulation, step input is applied to the dynamic equation of the unmanned free-swimming submersible vehicle. The fitness function is determined for the controller integral square error, settling time value, and maximum percent overshoot value. We also evaluate the optimization time of the selected algorithms based on the fitness function. Experimental results show that the Artificial Hummingbird Algorithm, Gray Wolf Optimization, and Particle Swarm Optimization achieve significant performance. This underlines that using metaheuristic methods in PID gain optimization increases overall system performance.

Keywords: PID gain optimization; metaheuristic methods; artificial intelligence; PID controller design

1. INTRODUCTION

PID (Proportional-Integral-Derivative) controllers in control systems have great acceptance and reputation. Despite the emergence of numerous complex control theories and techniques, control loops generally continue to rely on proportional-integral-derivative (PID) controllers since they have a simple structure, provide easy design as well as implementation, and have a wide range of implementation. A PID controller has mainly a proportional gain (KP), an integral gain (KI), as well as a derivative gain (KD). The main goal of the PID controller is to minimize over a short time, which is computed by taking the difference between a set point and the measured process variable. Hence, determining the optimum value of the three gains (KP, KI, KD) is challenging, especially when the process to control does not have a linear behavior.

Tuning of the first PID controller parameters was achieved by Ziegler and Nichols (ZN) [1]. The main disadvantage of this tuning method is to have a large settling time as well as overshoot. Hence, each gain has to be refined later. Cohen and Coon (C-C) [2] proposed another method that optimizes three gains. The major limitation of this method is that all computed gains are constant as well as they are not able to be on-line reconfigured.

Various state-of-the-art methods have been proposed to develop a better adjustment mechanism for each optimal gain (KP, KI, KD) of the PID controller. These methods may be grouped as fuzzy control [3], robust control [4], adaptive control [5], and neural network control. With the advent of computer technology, several control design methods based on artificial intelligence have been proposed. The usage of artificial intelligence methods provides a quick and accurate method for tuning parameters to reach precise control.

Recently, various types of metaheuristic methods have been proposed to parameter tuning of PID controllers. In this study, we perform Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO), Whale Optimization Algorithm (WOA), Firefly Algorithm (FA), Harris Hawks Optimization (HHO), Artificial Hummingbird Algorithm (AHA) and African Vultures Optimization Algorithm (AVOA). The other sections are organized as follows: In section 2, related works about metaheuristic methods are given. In section 3, details of the proposed methods are given. Section 4 includes results and discussion. Section 5 concludes the study.

2. RELATED WORKS

Kim and Cho [6] used genetic algorithm (GA), PSO, and BFO methods for tuning the PID gains in the automatic voltage regulator system; and also implemented hybrid GA-PSO and BF-GA methods. The BF-GA hybrid algorithms gave the best results in their system. According to the simulation results, hybrid applications have a better response than GA and PSO applications. Srinivas et al. [7] compared traditional PID tuning methods with the GA method. According to the results of their study, GA was faster and had low overshoot compared to traditional methods such as the Z-N method, C-C method, and minimum error integral criteria method. El-Deen et al. [8] and Bassi and Dada [9] tuned the PID gains for DC motor speed control using GA and controlled the system. In their studies, GA was compared with the classical method Z-N and Active Set Optimization Algorithm (ASOA). In both studies, the GA method produced a better solution to control system dynamics. Emmanuel and Inyiama [10] tried to apply Z-N, Fuzzy Logic (FL), GA and Artificial Neural Network (ANN) methods in controller design for a robot manipulator. According to their studies, FL, GA and ANN methods are more suitable to obtain optimum system performance. Although these methods are more complex and slower to model than classical models, modern control methods have high accuracy.

Sandoval et al. [11] tuned PID gains with ACO to control the robotic arm and studied as experimentally. In their study, the Z-N method was compared with manual adjustment and ACO. The ACO was better than other methods. In addition, the system response improved as the number of ant was increased.

El-Telbany [12] compared the ABC, PSO, GA and ZN methods for DC motor controller optimization, and the best rise time and settling time for steady-state response were obtained with the ABC method. Liao et al. [13] used ABC to optimize DC motor PID gains. It has been stated that this method is effective for many optimization problems because it is easy to implement. Senberber and Bagis [14] stated that the application of the Fractional Order PID and ABC method greatly increased the performance of the system. Annisa et al. [15] tried to reduce the vibration at the endpoint by using a PID controller for a flexible robotic arm as a hybrid with PSO and ABC methods. According to the simulation results, it has been revealed that the PSO method was better than the ABC method. Zhi [16] applied the ABC, PSO, GA and ZN methods to determine the PID gains of the single-phase inverter system. In their study, the ABC method performed significantly better than other applications.

Kotteeswaran and Sivakumar [17] applied the BA method for tuning gains of the PI controller in the coal gasifier system. The controller controlled a complex system that was a non-linear multidimensional process. With the current controller, the system did not meet the performance criteria when sinusoidal pressure disturbance was applied while the system was at 0% load. With the optimum gain values determined by the BA, the system exhibited the desired performance under 0%, 50% and 100% load. Katal et al. [18] optimized PID gains with BA for the liquid-level control system. The BA presented better results than classical methods. Singh et al. [19] compared BA and PSO algorithms for servo motor PID gain optimization. The integral of the time square error was used as a stopping criterion in the study. According to the simulation results, the BA presented better rise time and settling time than the PSO application. Premkumar and Manikandan [20] used PSO, CS, and BA algorithms to determine gains of Fuzzy PD and Fuzzy PID controllers for brushless DC motor speed control applications. The BA method achieved faster results in Fuzzy PD optimization compared to other methods.

Bayoumi and Salem [21] applied the BFO method to design a robust PID controller. As a fitness function, they optimized percent overshoot, settling time, rise time, and steady-state error. The BFO-PID application showed better voltage regulation than the classical Z-N application. Benbouabdallah and Zhu [22] automatically tuned the PID gains of the mobile robot moving in an unknown route with BFO. The BFO and PSO applications were used as a hybrid to obtain optimum gains. According to the simulation results, the hybrid BFO-PSO application had better results than BFO, PSO and GA applications. Sivakumar et al. [23] used BFO to determine the PID gains in a MIMO system. Three different cost functions were compared. According to the simulation results, these functions presented very similar performances. Agarwal et al. [24] tried to minimize the maximum overshoot, peak time, settling time, and rise time with an integral absolute error of the system by applying BFO in their study. Compared with the classical Z-N method, the BFO method determined more optimum gain values. Jasim [25] tuned DC servo motor PID gains with BFO. Similar to other studies, fitness function was tried to be minimized. According to the simulation results, the high performance of the BFO method was emphasized in the study.

The PSO method is used quite frequently in control applications. Chang and Chen [26] optimized the PID gains for a MIMO system using PSO. An improved PSO method compared with different machine learning algorithms performed better than other algorithms. Rajesh and Ananda [27] optimized PID gains with the PSO method for the control of the camera moving in 2 axes and located on a UAV. The PSO algorithm was compared with classical Z-N and C-C applications. The PSO performed better than other applications. Lodhi and Saraf [28] tuned PID gains using the Real-valued Genetic Algorithm (RGA), GA, and PSO methods. According to the simulation results, it was stated in the study that PSO was better than other algorithms. Fister et al. [29] used two different evolutionary algorithms which are Differential Evolution (DE) and GA for determining PID gains on a SCARA-type robot with 2 degrees of freedom; and four different swarm intelligence-based algorithms, namely BA, Hybrid Bat Algorithm (HBA), PSO and CS, was applied experimentally. In their study, algorithms were compared according to many parameters. The HBA, BA, DE, PSO, CS, and GA operating under the same conditions produced results in 1.87, 1.93, 2.30, 2.64, 3.03, and 21.13 milliseconds from fast to slow, respectively. However, when the results in the robot axes were examined, the method with the least standard deviation compared to the average was obtained as PSO. According to these average results, HBA

presented the worst result. Joseph and Dada [30] automatically tuned PID gains using PSO for the control of the inverted pendulum system. The PSO showed better system performance than conventional methods.

Kumar et al. [31] used the CS method in the PID controller design of nonlinear systems such as ship roll dynamics, oscillators and inverted pendulums. integral square error, integral time square error and integral absolute error criteria were used as fitness functions. It was compared with the system designed with PSO by another researcher. According to the simulation results, the CS performed better than the PSO application. Gholap et al. [32] applied PSO, GA and SA to obtain optimum PID gains. They used two fitness functions, integral time absolute error and time domain specifications in their work. Different types of algorithms were compared in the study, taking into account parameters such as the number of iterations and cost. According to the results, although all three applications can control the system, the PSO and SA applications showed better results than GA applications. Bingul and Karahan [33] tuned the gains of the PID controller in the automatic voltage regulator system with CS. Performance criteria were determined as in the study of Bayoumi and Salem. The CS method proposed in the study performed better than the systems designed with PSO, ABC and BF-GA methods.

Bansal et al. [34] used the Multi-Objectives SA application for tuning PID gains. Single-Objective SA and classical Z-N methods were also applied in the study. The multi-objective SA application performed better than other methods. Vijay and Banu [35] optimized gains in attitude control with PID by the SA method. The fuel consumption of the thruster actuator in the attitude control system has been optimized. Lahcene et al. [36] used SA to optimally select the gains of the fractional order PID controller that controls the automatic voltage regulator system. Similar to other studies, the cost function was tried to be minimized. SA application performed better than Multi-Objective External Optimization and PSO methods. Shatnawi and Bayoumi [37] determined gains of PI and PID controllers designed for permanent magnet BLDC motors by SA methods. Maximum overshoot, rise time and settling time were optimized simultaneously. In their study, the SA applications performed better than PSO and classical Z-N applications.

Şen and Kalyoncu [38] tuned the gains of the PID controller for a quadruped robot system using GWO. For the same system, the GWO method performed faster and better than PSO and GA methods. Agarwal et al. [39] optimized gains of the fractional order PID controller for DC motor speed control with the GWO method. When the system response was compared, the GWO method was good and robust compared to Invasive Weed Optimization, Stochastic Fractal Search, and PSO methods. Yadav et al. [40] optimized PID control gains in the nonlinear inverted pendulum system with GWO. The integral square error was used as a cost function in the study. Sule et al. [41] determined the gains of the PI controller controlling the wind turbine system with the GWO method and compared them with the PSO and GA methods. For their system, the GWO method worked better and achieved faster results. Verma and Devarapalli [42] tuned gains of the fractional order PID controller that controls the automatic voltage regulator system using different types of GWO. The modified GWO method suggested by the researchers presented better performance than standard GWO methods.

Hekimoglu et al. [43] optimized PID gains in the DC-DC buck converter system using WOA and compared them with the GA method. According to the simulation results, the WOA method achieved better results. Kumar and Kumar [44] optimized PID gains controlling the drilling machine system with WOA and compared them with PSO and classical Z-N methods. The WOA method was better than other methods when compared to the settling time and rise time values. Mosaad et al. [45] optimized PID gains in the automatic voltage regulator system with the WOA method. In their study, PID gains were optimized with many evolutionary algorithms (8 different metaheuristic methods) and compared with WOA. The WOA was designed as a better controller than other applications. Loucif et al. [46] optimized gains in the PID controller of the robot arm with 2 degrees of freedom with WOA and compared them with the PSO and GWO methods. According to the performance criteria in the study, the WOA was the most effective method.

Bendjeghaba et al. [47] proposed an algorithm for tuning PID controller parameters using the firefly algorithm. They compared their results with Ziegler-Nichols method and achieved efficient results. Coelho and Mariani [48] applied the firefly algorithm for tuning the PID controller. They introduced a chaotic firefly algorithm based on the Tinkerbell map. Their experimental results presented that their method tuned multi-loop PID controllers. Finally, Ekinci et al. [49] presented an application of the Harris Hawks optimization algorithm to find the optimal parameters of the PID controller.

3. METHODS

3.1. Dynamics of the System

An Unmanned Free-Swimming Submersible (UFSS) vehicle changes its depth with the help of an elevator which is driven by a motor while moving forward. The moment occurs about the pitch axis of UFSS with the change of the elevator angle. The depth of the UFSS can be changed concerning the desired reference angle by controlling the moment of this axis. It is possible to develop an autonomous system by controlling the moment of the pitch axis with the help of a controller.

In this study, the UFSS is selected as a vehicle since its transfer function is well-known in literature and easy-to-design a controller relative to the multi degree of freedom systems. The transfer function can be found in the Control System Engineering book released by Norman S. Nise [50]. The block diagram of the system is shown in Figure 1.

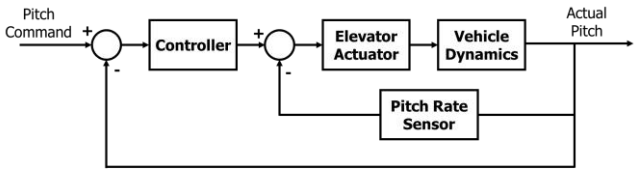

Figure 1. Closed-loop diagram of UFSS vehicle

The overall Transfer Function (OTF) in complex frequency domain, which expresses the actuator, vehicle dynamics and pitch rate sensor in the system, is shown in Equation 1.

$$OTF = \frac{-0.25\,s\ -\ 0.1087.}{s^4\ +\ 3.456\,s^3\ +\ 3.457\,s^2\ +\ 0.7193\,s\ +\ 0.04157} \qquad (1)$$

The coefficients describing the system dynamics in OTF are derived by considering all system elements. System elements and detailed block diagram of the system can be found in [50]. The transfer function of a real system is obtained through its mathematical model that takes into account the dynamics of the system.

3.2. Controller Design

In control applications, the main purpose is to reduce the error to zero. In this study, the PID controller is used to make the error closer to zero. The PID controller is frequently used in many simulation and experimental studies such as DC motor, Quadrotors and Inverted Pendulum. It is possible to apply a PID controller for UFSS vehicle. The equation of the PID controller is shown in Equation 2 [51].

$$u(t) = K_P e(t) + K_I \int_0^t e(t)d(t) + K_D \frac{de(t)}{dt} \qquad (2)$$

Table 1. Effect of increasing PID gains on system behavior

Close Loop Response	Rise Time	Overshoot	Settling Time	Steady-state Error
K_P	Decrease	Increase	Small Change	Decrease
K_I	Decrease	Increase	Increase	Decrease
K_D	Small Change	Decrease	Decrease	No Change

The proportional term adjusts control output proportionally depending on the size of the error. The integral term calculates the time integral of the error and adjusts the control output accordingly to prevent the control output from increasing with the accumulation of error over time. The derivative term allows control output to be changed quickly by calculating the rate of error. The effect of K_P, K_I and K_D gains on system behavior is shown in Table 1 [51].

3.3. Metaheuristic Algorithms

In this study, we apply the state-of-the-art metaheuristics methods, PSO, GWO, WOA, Firefly Algorithm (FA), Harris Hawks Optimization (HHO), Artificial Hummingbird Algorithm (AHA) and African Vultures Optimization Algorithm (AVOA) for optimizing PID gains. In the following, we summarize these methods.

In the PSO, new positions of particles are calculated at each iteration. This calculation is performed using the current position and best position of each particle. Each particle is repositioned at a new position determined by its velocity and the difference of its best position. This process continues until the particles reach a position that minimizes or maximizes their fitness function. Detailed steps of the PSO algorithm are given in Algorithm 1 [52].

Algorithm 1. Particle Swarm Optimization

Initialize Population: Positions and velocities of particles initialized as randomly
Define fitness function to evaluate each particle's fitness
Set initial personal best (pbest) for each particle to its current position
Set initial global best (gbest) to the position of the particle with the best fitness value
while termination criterion is not met **do**
  for each particle i **do**
    Update particle velocity using the following formula:
    $v_i = w * v_i + c1 * rand() * (pbest_i - x_i) + c2 * rand() * (gbest - x_i)$
    Update particle position using the following formula:
    $x_i = x_i + v_i$
    Evaluate fitness value of the new position
    if the new position is better than its personal best **then**
      Update personal best: $pbest_i = x_i$
    end if
    if the new position is better than the global best **then**
      Update global best: $gbest = x_i$
    end if
  end for
end while
return gbest

Algorithm 2. Gray Wolf Optimization

Initialize a population of gray 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 using the following formula:
    $x_i = \frac{\alpha+2\times\beta+2\times\delta-5\times x_i}{4}$
    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 3. Whale Optimization Algorithm

```
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 whale i in the population do
        Generate a 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 using the following formula:
                x_i = x_best − A × |C × x_best − x_i|
            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 using the following formula:
            x_i = x_j − A × |C × x_j − x_i|
        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 4. Firefly Algorithm

```
Initialize a population of fireflies randomly
Evaluate the fitness of each firefly
Set the best solution as the current global best
while stopping criterion 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 β(r_ij) of firefly j towards firefly i
                Move firefly i towards firefly j with a step size αβ(r_ij)
            end if
        end for
        Evaluate the fitness of the new position of firefly i
        if the new position of firefly i is better than its current position then
            Set the new position as the current position of firefly i
        end if
    end for
    Update the global best solution
end while
return the best solution found
```

Algorithm 5. Harris Hawks Optimization

```
Initialize a population of hawks randomly
Evaluate the fitness of each hawk
Set the best solution as the current global best
while stopping criterion not met do
    Sort the hawks in descending order of fitness
    Compute the position of the top predator hawk
    Compute the position of the other predator hawks
    for each prey hawk do
        Compute the position of the new prey hawk
        Evaluate the fitness of the new prey hawk
        if the new prey hawk is better than the current prey hawk then
            Replace the current prey hawk with the new prey hawk
        end if
    end for
    Update the global best solution
end while
return the best solution found
```

Algorithm 6. Artificial Hummingbird Algorithm

```
Initialize the population of hummingbirds with random positions and velocities
Evaluate the fitness of each hummingbird
while stopping criterion not met do
    for each hummingbird do
        Calculate the attraction towards the best hummingbird in the neighborhood
        Update the velocity of the hummingbird
        Update the position of the hummingbird
        Evaluate the fitness of the new position
        if new position is better than the current position then
            Replace the current position with the new position
        end if
    end for
end while
return the best hummingbird solution found
```

Algorithm 7. African Vultures Optimization Algorithm

```
initialize vulture population and their positions
initialize search radius and iteration counter
while not converged do
    for each vulture do
        update vulture's position based on its personal best and global best positions
        if vulture is outside the search radius then
            move it back inside
        end if
        if vulture is better than the current global best then
            update the global best position
        end if
    end for
    update the search radius based on the number of iterations
    update the iteration counter
end while
```

The GWO is based on three basic behaviors: hunting, sorting and mating. The wolf pack interacts with each other to optimize these behaviors. Hunting behavior involves exploring the search area to reach the best location. Sorting behavior allows wolves to adjust their position and strength against each other. Mating behavior, on the other hand, allows the creation of new wolves and the enlargement of the solution area. The Pseudocode of GWO is presented in Algorithm 2 [53].

The WOA works by maintaining a population of candidate solutions (whales) and improving them iteratively over several generations. At each iteration, each whale's position is updated using two random vectors, A and C, and a specific formula that simulates the hunting behavior of humpback whales. Some randomness is then introduced into the search process by applying the random join operator to each whale's new position. The fitness value of each whale's new position is calculated and compared against the best available position. If the new position is better than the current best position, it is assigned as the new current best position. This process continues until a certain termination criterion is met. The final solution is the best available position. Detailed steps of WOA are presented in Algorithm 3 [54].

In the FA, the difference in light intensity between fireflies is related to distance and the intensity of the environment. This difference determines each firefly's movement. The main purpose of the algorithm is to find the position where the difference in light intensity between many fireflies is the least. This location is a near-optimal solution. Using light intensities as motion vectors, the algorithm creates a swarm of fireflies that move through the solution range. Each firefly moves towards the light intensity value of the closest firefly, thus bringing the flock closer to the optimum solution. The Pseudocode of the FA is shown in Algorithm 4 [55].

The HHO is led by the strongest hawk chosen as the pack leader. Other hawks follow the leader's position and speed and move towards the positions the leader finds. Hawks also cooperate with each other to hunt non-leader hawks. This algorithm is particularly effective for multidimensional and complex optimization problems. An overview of the HHO is shown in Algorithm 5 [56].

The AHA is an optimization algorithm inspired by the interactions of flowers and hummingbirds in nature. The algorithm works by selecting the most suitable flowers using information such as the location of the flowers, the amount of nectar, and the taste value. The AHA can show high performance, especially in complex problems. The pseudocode of AHA is shown in Algorithm 6 [57].

The AVOA mimics the social interactions of vultures within the population and uses the leader-follower hierarchy. Initially, the fitness of each vulture is calculated. The population with the best fitness value is selected as the leader. Follower vultures keep track of the leader's position and update their position according to the leader's movement. Detailed steps of the AVOA are shown in Algorithm 7 [58].

Metaheuristic algorithms offer a more flexible and robust approach by exploring a larger solution space and considering multiple objective criteria than deterministic methods such as Ziegler-Nichols and Cohen-Cohen. They can handle nonlinear systems, take into account constraints, and find near-optimal solutions that outperform deterministic methods in terms of stability, performance, and robustness. Furthermore, they are particularly useful when dealing with complex control systems where the relationship between gains and system behavior is not well defined or where multiple conflicting objectives must be considered.

3.3.1. Input Parameters

The input parameters required to optimize the PID gains in the control problem are shown in Table 2. The number of search agents and the maximum number of iterations are set to 20 and 30, respectively. Solution quality may be enhanced by increasing these values. However, this increases the optimization time, and it has been observed that the effect on the result is quite low.

Lower boundary and upper boundary values are set to 0 and 100, respectively. It should be considered that these values may need to be adjusted for different control problems.

Table 2. Input parameters for optimization

Parameter	Value
Number of search agents	20
Maximum number of iterations	30
Lower boundary	[0 0 0]
Upper boundary	[100 100 100]
Problem Dimension	3

3.3.2. Fitness Function

The fitness function is used to evaluate the performance in the control problem, specifically for optimizing PID gains. In this study, the fitness function is calculated based on three metrics that are integral square error (ISE), overshoot (OS) and settling time (T_s). The formulation of the fitness function used in this study is given in Equation 3.

$$FitnessFunc = (J_1 \cdot ISE) + (J_2 \cdot \%OS) + (J_3 \cdot T_s) \quad (3)$$

where $J_1 = 5\, rad^{-2} \cdot sec^{-1}$, $J_2 = 10$ and $J_3 = 5\, sec^{-1}$. These values make the FitnessFunc variable unitless. Weights of ISE, OS and T_s are set by trial and error to get the step response more realistic. The reason why the OS value is larger than the ISE and T_s values is to prevent the system from large control signals. ISE and T_s weights are of equal importance for this problem. If the weights of these values are increased, oscillation occurs initially. This oscillation can cause instability in real system applications.

Input signals such as step, ramped, impulse, sinusoidal, parabolic etc. are applied to measure the performance of the controller. In this study, step input is applied to the system. ISE integrates the square of the error amount between the reference value (unit step = 1 rad) and the actual response numerically over the time interval of 0.1 sec. Thus, the overall error of the system is measured.

Overshoot is calculated by the difference between the reference value and the maximum response value. An overshoot penalty is applied if the maximum overshoot exceeds 1% threshold.

Settling time is measured according to 2% tolerance band. The response of the system is assigned to time vector by using loop which checks the specified tolerance band. If the settling time value is greater than 10 second threshold, the settling time penalty is applied.

4. RESULTS AND DISCUSSION

In this study, the depth of the UFSS vehicle is controlled by changing the elevator angle. The gain values of the PID controller used in the system are optimized with the PSO, GWO, WOA, FA, HHA, AHA, and AVOA metaheuristic methods. These algorithms are compared with respect to optimization time, error value, settling time value and % overshoot metrics.

The total optimization time of the algorithms performed in this study is shown in Figure 2. The time is determined based on the value of the fitness function. The fitness values achieved by the PSO, GWO, WOA, FA, HHO, AHA, and AVOA are 1.918, 1.912, 2.328, 1.911, 2.388, 1.924 and 2.012, respectively. As you can see in Figure 2, the FA requires much time when compared to the other methods and has a better cost value. The PSO, GWO, AHA, and AVOA converge in a short time with acceptable fitness values.

Error values in the proposed system where step input is applied are shown in Figure 3. All algorithms almost reach

the desired reference value. Error values of the AHA, PSO and FA are relatively low compared to other algorithms. Absolute error values of the PSO, GWO, WOA, FA, HHO, AHA, and AVOA are 0.00299 rad, 0.00381 rad, 0.05141 rad, 0.00306 rad, 0.00622 rad, 0.00237 rad, and 0.00535 rad, respectively. The error value of WOA is higher than other algorithms. However, it should be noted that all algorithms have fairly good error values for this control problem. For example, HHO has an error of 0.00622 radians (≈0.36°) for 1 radian (≈57.3°) input.

The settling time values, which are an important criterion for the evaluation of control problems, are compared in Figure 4. Settling time values of PSO, GWO, WOA, FA, HHO, AHA, and AVOA are obtained as 7.433 sec, 8.142 sec, 3.731 sec, 7.536 sec, 22.204 sec, 7.126 sec, and 25.924 sec, respectively. The WOA achieves the best performance when we focus on the settling time. The performances of the PSO, GWO, FA, and AHA are close to each other. On the other hand, HHO and AVOA methods require a larger time.

Another important benchmark for control problems is the percentage overshoot value that is presented in Figure 5. Percentage overshoot values of PSO, GWO, WOA, FA, HHO, AHA, and AVOA are calculated as 1.121, 1.022, 6.520, 1.029, 6.978, 0.747, and 2.414, respectively. While high overshoot may be appropriate in some systems, low overshoot may be appropriate in some systems. Overshoot evaluation is problem specific. In this study, a low overshoot value would be more appropriate. The performances of the AHA, GWO, and FA are close to each other and achieve the best results. Percentage overshoot values of WOA and HHO are greater than other algorithms.

Integral square error values of PSO, GWO, WOA, FA, HHO, AHA, and AVOA are calculated as 0.453 rad² · sec, 0.382 rad² · sec, 0.376 rad² · sec, 0.382 rad² · sec, 0.392 rad² · sec, 0.374 rad² · sec, and 0.384 rad² · sec. AHA, WOA and GWO methods have better ISE values than other algorithms. ISE values are shown in Figure 6.

AHA, GWO and PSO have better performance than other algorithms according to simulation time, best cost value, percentage error, settling time and percentage overshoot. AHA, GWO and PSO have almost same performance.

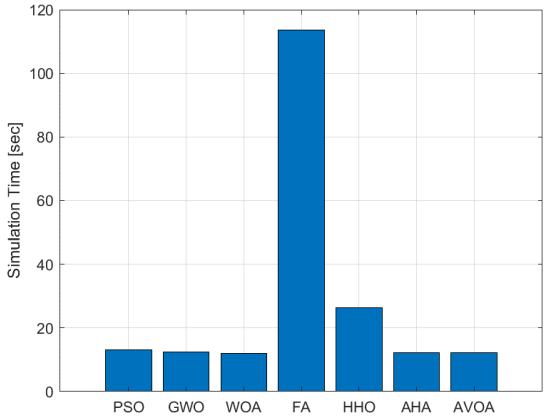

Figure 2. Total optimization time

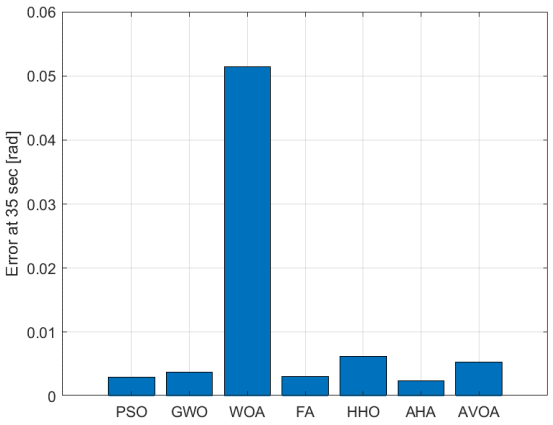

Figure 3. Error values at the end of simulation

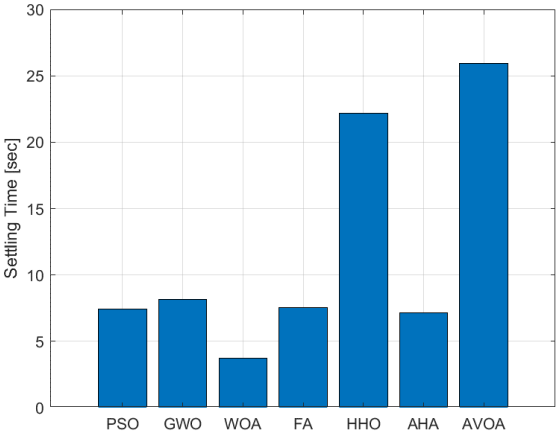

Figure 4. Settling time of step input

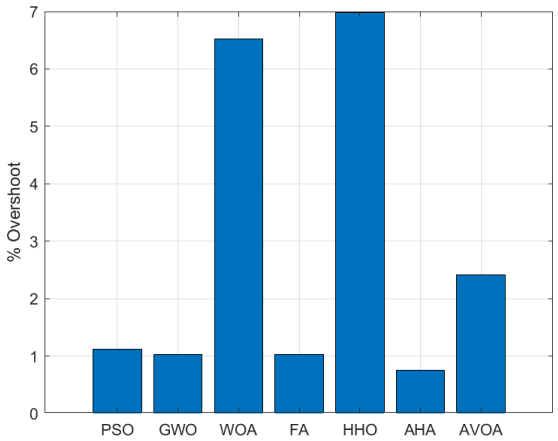

Figure 5. Percentage overshoot of step input

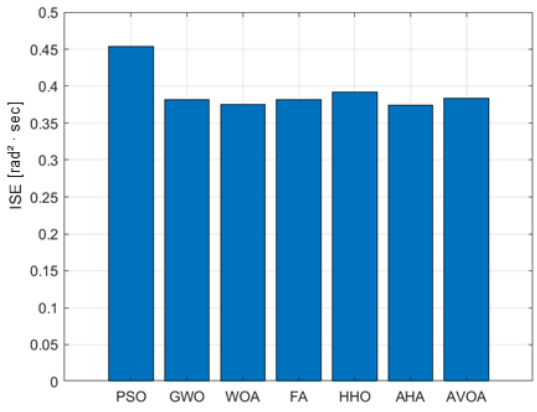

Figure 6. ISE values of all algorithms

Table 3. Optimized PID gains

Algorithm	K_P	K_I	K_D
PSO	6.026	0.551	31.180
GWO	5.862	0.505	31.551
WOA	7.051	0.000	29.674
FA	6.058	0.535	31.130
HHO	7.960	2.746	35.660
AHA	6.311	0.532	30.176
AVOA	4.763	0.634	35.348

Table 4. Results of metaheuristic algorithms

MN	RT (sec)	ST (sec)	OS (rad)	PE (rad)	PT (sec)	BCV	ISE (rad²·sec)	SIT (sec)
PSO	0.66	7.43	1.12	1.01	19.29	1.92	0.45	13.20
GWO	0.66	8.14	1.02	1.01	1.21	1.91	0.38	12.39
WOA	0.63	3.73	6.52	1.01	1.23	2.33	0.38	12.05
FA	0.66	7.54	1.03	1.01	1.21	1.91	0.38	113.59
HHO	0.57	22.20	6.98	1.07	1.09	2.39	0.39	26.42
AHA	0.68	7.13	0.75	1.01	1.21	1.92	0.37	12.34
AVOA	0.60	25.92	2.41	1.02	20.76	2.01	0.38	12.17

MN: Method Name, RT: Rise Time, ST: Settling Time, OS: Overshoot, PE: Peak, PT: Peak Time, BCV: Best Cost Value, ISE: Integral Square Error, SIT: Simulation Time

PID gains obtained by algorithms are shown in Table 3. Detailed results of the metaheuristic methods are also shown in Table 4. Furthermore, step responses are shown in Figure 7 based on optimized gains.

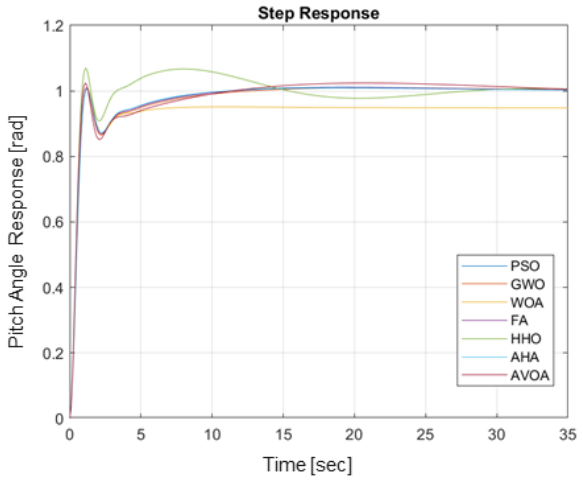

Figure 7. Step responses of all algorithms

Since this study is the application of metaheuristic methods in the field of control, control gains have been optimized by considering only the system dynamics. However, in real systems, the control effort should also be included into the fitness function. This ensures that the control signals sent to the actuator stay within acceptable limits. The control signal changes the angle of the elevator. This change is controlled by the torque produced by a motor. Both upper and lower limits need to be specified by the user, similar to setting the values for number of search agents, maximum number of iterations, and problem dimension. When selecting these limits, considering the control effort could be beneficial.

5. CONCLUSION

In this study, the gain values of the controller (PID) used in a control problem are optimized by new generation metaheuristics that are the PSO, GWO, WOA, FA, HHA, AHA and AVOA methods. To achieve a fair comparison, the algorithms are compared under the same conditions. A special fitness function is defined considering the optimization time, controller error, step input settling time and step input overshoot value in this study. Experimental results present that the AHA, GWO and PSO perform better than other algorithms. On the other hand, FA, HHO, and AVOA present lower performance.

The methods compared in the study are acceptable for PID gain optimization. Researchers can optimize PID gain values based on the fitness function they will determine in different control problems. The PSO, which is frequently used in other control problem studies in the literature, performed quite well in this study as well. As a result, the AHA, GWO and PSO can be preferred for PID gain optimization.

In future studies, new generation metaheuristics algorithms will be compared by optimizing PID gain values according to different fitness functions. Thus, the impact of the metrics can be compared as well. In addition, the optimized gain values can be used with the Fuzzy Logic method to design a better controller. Thus, the controller can produce a better response according to different reference signals. Furthermore, we will plan to implement the state-of-the-art metaheuristics by combining with traditional methods in parallel to save CPU time.

REFERENCES

[1] Ziegler JG, Nichols NB. Optimum Settings for Automatic Controllers. J Dyn Syst Meas Control 1993; 115:220–2. https://doi.org/10.1115/1.2899060.

[2] Cohen GH, Coon GA. Theoretical Consideration of Retarded Control. Trans Am Soc Mech Eng 1953; 75:827–34. https://doi.org/10.1115/1.4015451.

[3] Fereidouni, A, Masoum, M. A. S., & Moghbel, M. (2015). A new adaptive configuration of PID type fuzzy logic controller. In ISA Transactions (Vol. 56, pp. 222–240). Elsevier BV. https://doi.org/10.1016/j.isatra.2014.11.010

[4] Ali, R., Mohamed, T. H., Qudaih, Y. S., & Mitani, Y. (2014). A new load frequency control approach in an isolated small power systems using coefficient diagram method. In International Journal of Electrical Power & Energy Systems (Vol. 56, pp. 110–116). Elsevier BV. https://doi.org/10.1016/j.ijepes.2013.11.002

[5] Mittal, A., Kapoor, A., & Saxena, T. K. (2013). Adaptive tuning of PID controller for a nonlinear constant temperature water bath under set-point disturbances using GANFC. J. Auto Syst. Eng, 7, 143-63.

[6] Kim DH, Cho JH. A Biologically Inspired Intelligent PID Controller Tuning for AVR Systems. Int J Control Autom Syst 2006; 4:624–36.

[7] Srinivas P, Vijaya Lakshmi K, Kumar VN. A Comparison of PID Controller Tuning Methods for Three Tank Level Process. Int J Adv Res Electr Electron Instrum Eng 2007;3.

[8] El-Deen AT, Mahmoud AAH, El-Sawi AR, El-Deen AT, Mahmoud AAH, El-Sawi AR. Optimal PID Tuning for DC Motor Speed Controller Based on Genetic Algorithm. Int Rev Autom Control 2015; 8:80–5. https://doi.org/10.15866/IREACO.V8I1.4839.

[9] Bassi JS, Dada EG. Automatic Tuning of Proportional–Integral–Derivative Controller using Genetic Algorithm. Pacific J Sci Technol 2018; 19:51–7.

[10] Emmanuel AC, Inyiama H. A SURVEY OF CONTROLLER DESIGN METHODS FOR A ROBOT MANIPULATOR IN HARSH ENVIRONMENTS. Eur J Eng Technol 2015; 3:64–73.

[11] Sandoval D, Soto I, Adasme P. Control of direct current motor using Ant Colony optimization. CHILECON 2015 - 2015 IEEE Chil. Conf. Electr. Electron. Eng. Inf. Commun. Technol. Proc. IEEE Chilecon 2015, Institute of Electrical and Electronics Engineers Inc.; 2016, p. 79–82. https://doi.org/10.1109/CHILECON.2015.7400356

[12] E.El-Telbany M. Tuning PID Controller for DC Motor: An Artificial Bees Optimization Approach. Int J Comput Appl 2013; 77:18–21. https://doi.org/10.5120/13559-1341.

[13] Liao W, Hu Y, Wang H. Optimization of PID control for DC motor based on artificial bee colony algorithm. Int. Conf. Adv. Mechatron. Syst. ICAMechS, IEEE Computer Society; 2014, p. 23–7. https://doi.org/10.1109/ICAMECHS.2014.6911617.

[14] Senberber H, Bagis A. Fractional PID controller design for fractional order systems using ABC algorithm. Proc. 21st Int. Conf. Electron., Institute of Electrical and Electronics Engineers Inc.; 2017. https://doi.org/10.1109/ELECTRONICS.2017.7995218.

[15] Annisa J, Mat Darus IZ, Tokhi MO, Mohamaddan S. Implementation of PID Based Controller Tuned by Evolutionary Algorithm for Double Link Flexible Robotic Manipulator. 2018 Int. Conf. Comput. Approach Smart Syst. Des. Appl. ICASSDA 2018, Institute of Electrical and Electronics Engineers Inc.; 2018. https://doi.org/10.1109/ICASSDA.2018.8477615.

[16] Zhi D. Optimization of PID Controller for Single Phase Inverter Based on ABC. 2019 5th Int. Conf. Control. Autom. Robot. ICCAR 2019, Institute of Electrical and Electronics Engineers Inc.; 2019, p. 501–4. https://doi.org/10.1109/ICCAR.2019.8813361.

[17] Kotteeswaran R, Sivakumar L. Optimal partial-retuning of decentralised PI controller of coal gasifier using bat algorithm. Int. Conf. Swarm, Evol. Memetic Comput., Springer; 2013, p. 750–61.

[18] Katal N, Kumar P, Narayan S. Optimal PID controller for coupled-tank liquid-level control system using bat algorithm. Int. Conf. Power, Control Embed. Syst. ICPCES 2014, Institute of Electrical and Electronics Engineers Inc.; 2014. https://doi.org/10.1109/ICPCES.2014.7062818.

[19] Singh K, Vasant P, Elamvazuthi I, Kannan R. PID Tuning of Servo Motor Using Bat Algorithm. Procedia Comput Sci 2015; 60:1798–808. https://doi.org/10.1016/J.PROCS.2015.08.290.

[20] Premkumar K, Manikandan B V. Bat algorithm optimized fuzzy PD based speed controller for brushless direct current motor. Eng Sci Technol an Int J 2016; 19:818–40. https://doi.org/10.1016/J.JESTCH.2015.11.004.

[21] Bayoumi EH., Salem F. PID controller for series-parallel resonant converters using bacterial foraging optimization. Electromotion Sci J 2012; 19:64–79.

[22] Benbouabdallah K, Zhu QD. Bacterial foraging oriented by particle swarm optimization of a Lyapunov-based controller for mobile robot target tracking. Proc. - Int. Conf. Nat. Comput., IEEE Computer Society; 2013, p. 506–11. https://doi.org/10.1109/ICNC.2013.6818029.

[23] Sivakumar R, Deepa P, Sankaran D. A Study on BFO Algorithm based PID Controller Design for MIMO Process using Various Cost Functions. Indian J Sci Technol 2016; 9:1–6. https://doi.org/10.17485/IJST/2016/V9I12/89942.

[24] Agarwal S, Yadav D, Verma A. Speed control of PMSM drive using bacterial foraging optimization. 4th IEEE Uttar Pradesh Sect. Int. Conf. Electr. Comput. Electron. UPCON 2017, vol. 2018- January, Institute of Electrical and Electronics Engineers Inc.; 2017, p. 84–90. https://doi.org/10.1109/UPCON.2017.8251027.

[25] Jasim MH. Tuning of a PID Controller by Bacterial Foraging Algorithm for Position Control of DC Servo Motor. Eng Technol J 2018; 36:287–94. https://doi.org/10.30684/etj.36.3A.7.

[26] Chang W Der, Chen CY. PID controller design for MIMO processes using improved particle swarm optimization. Circuits, Syst Signal Process 2014;33:1473–90. https://doi.org/10.1007/S00034-013-9710-4/FIGURES/8.

[27] Rajesh RJ, Ananda CM. PSO tuned PID controller for controlling camera position in UAV using 2-axis gimbal. Proc. 2015 IEEE Int. Conf. Power Adv. Control Eng. ICPACE 2015, Institute of Electrical and Electronics Engineers Inc.; 2015, p. 128–33. https://doi.org/10.1109/ICPACE.2015.7274930.

[28] Lodhi RS, Saraf A. Survey on PID Controller Based Automatic Voltage Regulator. Int J Adv Res Electr Electron Instrum Eng 2016; 5:7424–9.

[29] Fister D, Fister I, Fister I, Šafarič R. Parameter tuning of PID controller with reactive nature-inspired algorithms. Rob Auton Syst 2016; 84:64–75. https://doi.org/10.1016/J.ROBOT.2016.07.005.

[30] Joseph SB, Dada EG. Proportional-integral-derivative (PID) controller tuning for an inverted pendulum using particle swarm optimisation (PSO) algorithm. FUDMA J Sci 2018; 2:73–9.

[31] Kumar P, Nema S, Padhy PK. PID controller for nonlinear system using cuckoo optimization. Int. Conf. Control. Instrumentation, Commun. Comput. Technol. ICCICCT 2014, Institute of Electrical and Electronics Engineers Inc.; 2014, p. 711–6. https://doi.org/10.1109/ICCICCT.2014.6993052.

[32] Gholap V, Dessai CN, Bagyaveereswaran V. PID controller tuning using metaheuristic optimization algorithms for benchmark problems. IOP Conf Ser Mater Sci Eng 2017; 263:052021. https://doi.org/10.1088/1757-899X/263/5/052021.

[33] Bingul Z, Karahan O. A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system. J Franklin Inst 2018; 355:5534–59. https://doi.org/10.1016/J.JFRANKLIN.2018.05.056.

[34] Bansal R, Jain M, Bhushan B. Designing of Multi-objective Simulated Annealing Algorithm tuned PID controller for a temperature control system. 6th IEEE Power India Int. Conf., Institute of Electrical and Electronics Engineers (IEEE); 2015, p. 1–6. https://doi.org/10.1109/POWERI.2014.7117716.

[35] Vijay D, Banu US. PID controller tuned using Simulated Annealing for assured crew re-entry vehicle with PWPF thruster firing. Proc. 2016 Online Int. Conf. Green Eng. Technol. IC-GET 2016, Institute of Electrical and Electronics Engineers Inc.; 2017. https://doi.org/10.1109/GET.2016.7916804.

[36] Lahcene R, Abdeldjalil S, Aissa K. Optimal tuning of fractional order PID controller for AVR system using simulated annealing optimization algorithm. 5th Int. Conf. Electr. Eng. - Boumerdes, ICEE-B 2017, vol. 2017- January, Institute of Electrical and Electronics Engineers Inc.; 2017, p. 1–6. https://doi.org/10.1109/ICEE-B.2017.8192194.

[37] Shatnawi M, Bayoumi E. Brushless DC motor controller optimization using simulated annealing. Int. Conf. Electical Drives Power Electron., vol. 2019-September, KoREMA; 2019, p. 292–7. https://doi.org/10.1109/EDPE.2019.8883924.

[38] ŞEN MA, KALYONCU M. Optimal Tuning of PID Controller Using Grey Wolf Optimizer Algorithm for Quadruped Robot. Balk J Electr Comput Eng 2018; 6:29–35. https://doi.org/10.17694/BAJECE.401992.

[39] Agarwal J, Parmar G, Gupta R, Sikander A. Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor. Microsyst Technol 2018; 24:4997–5006. https://doi.org/10.1007/S00542-018-3920-4/FIGURES/13.

[40] Yadav S, Nagar SK, Mishra A. Tuning of parameters of PID controller using Grey Wolf Optimizer. Proc. Int. Conf. Adv. Electron. Electr. Comput. Intell. 2019, Elsevier BV; 2019. https://doi.org/10.2139/SSRN.3575432.

[41] Sule AH, Mokhtar AS, Jamian JJ Bin, Khidrani A, Larik RM. Optimal tuning of proportional integral controller for fixed-speed wind turbine using grey wolf optimizer. Int J Electr Comput Eng 2020; 10:5251–61. https://doi.org/10.11591/IJECE.V10I5.PP5251-5261.

[42] Verma SK, Devarapalli R. Fractional order PID controller with optimal parameters using Modified Grey Wolf Optimizer for AVR system. Arch Control Sci 2022; 32:429–50. https://doi.org/10.24425/acs.2022.141719.

[43] Hekimoğlu B, Ekinci S, Kaya S. Optimal PID Controller Design of DC-DC Buck Converter using Whale Optimization Algorithm. Int. Conf. Artif. Intell. Data Process. IDAP 2018, Institute of Electrical and Electronics Engineers Inc.; 2018. https://doi.org/10.1109/IDAP.2018.8620833.

[44] Kumar AA, Kumar G, Anil A, Dr K, Giriraj Kumar SM. Application of Whale Optimization Algorithm for tuning of a PID controller for a drilling machine Atal Anil Kumar, S M Giriraj Kumar. Application of Whale Optimization Algorithm for tuning of a PID controller for a drilling machine Application of Whale Optimization Algorithm for tuning of a PID controller for a drilling machine. ICAARS 2018.

[45] Mosaad AM, Attia MA, Abdelaziz AY. Whale optimization algorithm to tune PID and PIDA controllers on AVR system. Ain Shams Eng J 2019; 10:755–67. https://doi.org/10.1016/J.ASEJ.2019.07.004.

[46] Loucif F, Kechida S, Sebbagh A. Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator. J Brazilian Soc Mech Sci Eng 2020; 42:1–11. https://doi.org/10.1007/S40430-019-2074-3/FIGURES/11.

[47] Bendjeghaba, O., Boushaki, S. I., & Zemmour, N. (2013, May). Firefly algorithm for optimal tuning of PID controller parameters. 4th International Conference on Power Engineering, Energy and Electrical Drives. Presented at the 2013 IV International Conference on Power Engineering, Energy and Electrical Drives (POWERENG), Istanbul, Turkey. doi:10.1109/powereng.2013.663579

[48] Coelho, L. dos S., & Mariani, V. C. (2012). Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. Computers & Mathematics with Applications (Oxford, England: 1987), 64(8), 2371–2382. doi: 10.1016/j.camwa.2012.05.007

[49] Ekinci, S., Izci, D., & Hekimoglu, B. (2020, June). PID speed control of DC motor using Harris hawks optimization algorithm. 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE). Presented at the 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Istanbul, Turkey. doi:10.1109/icecce49384.2020.9179308

[50] Nise NS. Control Systems Engineering. 7th Edition. John Wiley & Sons; 2013.

[51] Introduction: PID Controller Design n.d. https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID (accessed November 13, 2022).

[52] Kennedy J, Eberhart R. Particle swarm optimization. Proc. ICNN'95 - Int. Conf. Neural Networks, vol. 4, 1995, p. 1942–8 vol.4. https://doi.org/10.1109/ICNN.1995.488968.

[53] Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf Optimizer. Adv Eng Softw 2014; 69:46–61. https://doi.org/https://doi.org/10.1016/j.advengsoft.2013.12.007.

[54] Mirjalili S, Lewis A. The Whale Optimization Algorithm. Adv Eng Softw 2016; 95:51–67. https://doi.org/https://doi.org/10.1016/j.advengsoft.2016.01.008.

[55] Yang X-S. Firefly Algorithms for Multimodal Optimization. In: Watanabe O, Zeugmann T, editors. Stoch. Algorithms Found. Appl., Berlin, Heidelberg: Springer Berlin Heidelberg; 2009, p. 169–78.

[56] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: Algorithm and applications. Futur Gener Comput Syst 2019; 97:849–72. https://doi.org/https://doi.org/10.1016/j.future.2019.02.028.

[57] Zhao W, Wang L, Mirjalili S. Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. Comput Methods Appl Mech Eng 2022; 388:114194. https://doi.org/https://doi.org/10.1016/j.cma.2021.114194.

[58] Abdollahzadeh B, Gharehchopogh FS, Mirjalili S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng 2021; 158:107408. https://doi.org/https://doi.org/10.1016/j.cie.2021.107408.