An International Journal of Optimization and Control: Theories & Applications
ISSN: 2146-0957 eISSN: 2146-5703
Vol.15, No.1, pp.166-182 (2025)
https://doi.org/10.36922/ijocta.1620

RESEARCH ARTICLE

Recent metaheuristics on control parameter determination

Yagmur Olmez[1*], Gonca Ozmen Koca[2], Zuhtu Hakan Akpolat[3]

[1]*Department of Electrical and Electronics Engineering, Faculty of Engineering and Natural Sciences, Malatya Turgut Ozal University, Malatya, Turkiye*
[2]*Department of Mechatronics Engineering, Faculty of Technology, University of Firat, Elazig, Turkiye*
[3]*Department of Electrical Electronics Engineering, Faculty of Engineering, Fatih Sultan Mehmet Vakif University, Istanbul, Turkiye*
yagmur.olmez@ozal.edu.tr, gozmen@firat.edu.tr, zhakpolat@fsm.edu.tr

ARTICLE INFO

Article History:
Received: 27 May 2024
Accepted: 14 December 2024
Available Online: 31 January 2025

Keywords:
Metaheuristics
Optimal control
PI control
Two-wheeled robot

AMS Classification 2010:
13P25, 47N10

ABSTRACT

Metaheuristics have been widely used in recent years for tuning control parameters since they have a simple structure, are easy to apply, and provide efficient solutions. In this study, control of a two-wheeled mobile robot using the inverted pendulum principle is proposed. The performances of nine recent metaheuristics (Political Optimizer, Equilibrium Optimizer, Aquila Optimizer, Flow Directional Algorithm, Cheetah Optimizer, Golden Jackal Optimizer, Artificial Rabbit Optimization, Gazelle Optimizer, and Pelican Optimization) have been investigated for the balancing and speed control of a two-wheeled vehicle. In this context, a framework consisting of two cascaded PI controllers is designed to provide balance and speed control of the two-wheeled vehicle. The performances of the recent metaheuristics are also compared with previously introduced effective metaheuristic algorithms for further evaluation. The parameters of the controllers are tuned by using these metaheuristics. In experimental studies, quantitative and qualitative analyses are performed for evaluation of the metaheuristics. The dynamic system properties, convergence curves, computational times, and statistical results are provided to prove optimal control performances. The results show that 11 out of 14 compared algorithms produce similar optimal results in speed and balance control of the two-wheeled vehicle. The rest of them do not provide satisfactory results for the tuning of optimum control parameters of the two-wheeled vehicle.

1. Introduction

Metaheuristic methods play a crucial role in a broad range of applications across various disciplines. These algorithms offer powerful optimization strategies that may effectively solve complex and difficult issues. The metaheuristics are commonly utilized in the areas such as machine learning, feature selection,[1,2] engineering design,[3-7] image processing,[8,9] path planning,[10] signal processing,[11] control systems and applications[12] and so on. The most common use of these methods in the field of control systems is the determination of the controller parameters. These algorithms offer powerful optimization techniques that can effectively search for optimal solutions in complex and high-dimensional parameter spaces. Since determining the parameters of control systems is a time-consuming and challenging process, the use of metaheuristic

*Corresponding Author

methods in solving such problems has drawn the attention of scientists.

A novel fuzzy-PID method has been introduced for the Generation Control of the electric system by Arya.[13] The imperialist competitive optimization method has been used for tuning the parameters of the system. Mohanty et al. have presented an application to control the load frequency in a multisource-power model.[14] The authors have adapted the differential evolution algorithm to obtain the controller's parameters. Sathya et al. have developed a control structure relying on a Bat-inspired method for control of the load frequency on power systems.[15] The suggested approach is employed in the thermal power system by tuning PI control parameters. The outcomes show that the suggested approach outperforms the conventional PI and fuzzy-based PI control. Sahu et al. have introduced a hybrid-optimization algorithm based on Firefly and Pattern Search algorithms.[16] The implementation of the controlling of the automatic generation system is performed based on the proposed hybrid method. The experimental outcomes of the suggested method have been evaluated with a conventional parameter tuning method (Ziegler Nichols) and two modern optimization algorithms. The experimental results demonstrate the superior performance of the algorithm. Dash et al. have proposed a control framework for the automatic generation relying on the Cuckoo Search Optimizer.[17] In another study proposed by Dash et al., the Bat optimization method is applied to a PD-PID cascade control method.[18] The Cuckoo Search Algorithm has adapted to the PID control method for the DC Motor control.[19] The suggested control structure yields superior outcomes than the parallel-PID control. The parameter determination of the fuzzy-based PID frequency-control method is realized with an improved grey wolf algorithm for a power system.[20] To determine the optimal LQR control parameters, the Pareto-based optimization method has been developed by Wang et al.[21] Demirtas and Ahmad have introduced an optimization-based fractional-fuzzy-PI method for the power factor control in the AC voltage.[22] The PSO algorithm has been adapted to the problem to optimize the parameters of the PI controller. Basic PI, fuzzy-based PI, and fractional-order PI control the AC voltage. It is reported that the improved hybrid control method, which consists of fractional order and the basic PI, has produced better solutions than the basic PI and fuzzy-based control methods. In the other study, a PSO-based optimal-PI controller has been designed for the stability analysis of the various systems that have fractional-order delay.[23]

This paper has studied the efficiency of the recent metaheuristic methods on the controller parameter determination. To evaluate the performance of current metaheuristic algorithms, the PI-controller's gains are optimized to perform the balancing and speed controls of a two-wheeled robot developed utilizing the normally unstable principles of the inverted pendulum. The ten metaheuristic algorithms recommended over the past three years have been selected. The algorithms are Political, Equilibrium, Aquila Optimizers and Flow Directional, Cheetah, Artificial Rabbit, Golden Jackal, Gazelle, Pelican Optimization Algorithms. In terms of performance assessments, the application of selected algorithms in the optimization of control parameters, particularly of a system with a usually unstable structure, is significant. The PI control method was utilized as the control method, the balancing and the speed controls were performed separately, and the optimization algorithms were optimized for four parameters of the controllers. The selected optimization algorithms were categorized into two groups. First, despite being relatively young, algorithms such as Equilibrium Optimizer (EO), Aquila Optimizer (AO), Pelican Optimization Algorithm (POA), and Golden Jackal Optimization (GJO) algorithms are extremely popular and successful. The second category includes algorithms that have yet to be proposed and have few applications, such as the Political Optimizer (PO), Cheetah Optimization (CO), Artificial Rabbits Optimization (ARO), Gazelle Optimization Algorithm (GO), and Flow Directional Algorithm (FDA). In addition, the previously proposed algorithms are also included to the study for a fair evaluations of the recent algorithms. The mentioned older algorithms are Whale Optimization Algorithm (WOA), Grey Wolf Optimizer (GWO), Crow Search Algorithm (CSA), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Flower Pollination Algorithm (FPA).

The rest of the paper is structured as follows: Section 2 gives the introduction of the recent metaheuristic algorithms. Section 3 presents the experimental studies, and the results are also given in Section 4. As for Section 5, the conclusion of the study and its future directions are given in detail.

2. Metaheuristic methods

This study presents an investigation of control parameter determination for the recent metaheuristic methods which are PO, EO, AO, FDA, CO, ARO, GJO, GOA, and POA. These methods are compared with older metaheuristics, namely Whale, Grey Wolf, Crow Search, Covariance Matrix Adaptation Evolution Strategy, and Flower Pollination optimization algorithms. Table 1 presents the details of the considered methods.

The details of the methods, including the algorithm's inspiration, published years and journals, and author names are provided in Table 1. The determined methods for analyzing control parameter determination are selected from the last four years and they are inspired by various areas such as nature-inspired, physic-inspired, and socio-inspired.

2.1. Political optimizer

The Political Optimization (PO) algorithm, inspired by political processes, was proposed by Asgari et al in 2020.[24] All stages of political parties were modeled and their mathematical models were derived. The initial population is created by dividing it into parties and electoral districts. The algorithm is divided into five phases: party formation, voter allocation, election campaign, party change, and election affairs. Each population identifies one member and its votes are represented by the fitness values. Each party embodies a potential solution to the issue.

The main equation for the position update is found in the election campaign section, as given below Eqs. (1-2):

$$P^j_{i,k}(t+1) = \begin{cases} m^* + r(m^* - p^j_{i,k}(t)), & \text{if } p^j_{i,k}(t-1) \le p^j_{i,k}(t) \le m^* \text{ or } p^j_{i,k}(t-1) \ge p^j_{i,k}(t) \ge m^* \\ m^* + (2r-1)|m^* - p^j_{i,k}(t)|, & \text{if } p^j_{i,k}(t-1) \le p^j_{i,k}(t) \le m^* \text{ or } p^j_{i,k}(t-1) \ge p^j_{i,k}(t) \ge m^* \\ m^* + (2r-1)|m^* - p^j_{i,k}(t-1)|, & \text{if } p^j_{i,k}(t-1) \le p^j_{i,k}(t) \le m^* \text{ or } p^j_{i,k}(t-1) \ge p^j_{i,k}(t) \ge m^* \end{cases} \quad (1)$$

$$P^j_{i,k}(t+1) = \begin{cases} m^* + (2r-1)|m^* - p^j_{i,k}(t)|, & \text{if } p^j_{i,k}(t-1) \le p^j_{i,k}(t) \le m^* \text{ or } p^j_{i,k}(t-1) \ge p^j_{i,k}(t) \ge m^* \\ p^j_{i,k}(t-1) + r(p^j_{i,k}(t) - p^j_{i,k}(t-1)), & \text{if } p^j_{i,k}(t-1) \le m^* \le p^j_{i,k}(t) \text{ or } p^j_{i,k}(t-1) \ge m^* \ge p^j_{i,k}(t) \\ m^* + (2r-1)|m^* - p^j_{i,k}(t-1)|, & \text{if } m^* \le p^j_{i,k}(t-1) \le p^j_{i,k}(t) \text{ or } m^* \ge p^j_{i,k}(t-1) \ge p^j_{i,k}(t) \end{cases} \quad (2)$$

Where $p^j_{i,k}(t)$ indicates the jth member of the ith party with d-dimension.

2.2. Equilibrium optimizer

Equilibrium Optimizer (EO) is a physics-based optimization method developed by Faramarzi et al. in 2020.[25] In the updating stages, individuals are guided by a set of rules to an equilibrium position where the solutions are equally good. The method was designed with inspiration from control volume mass balance models. In these models, starting from the rule that the total mass of a substance is constant, the population is directed toward the position where the total fitness is in equilibrium. The particles' positions are revised using Eq. (3):

$$\vec{x} = \vec{x}_{eq} + (\vec{x} - \vec{x}_{eq}) \cdot \vec{F} + \frac{r}{\vec{\lambda} V} \cdot (1 - \vec{F}) \quad (3)$$

Where x denotes the concentration of the control volume, $\vec{x}_{eq}$ is the concentration at an equilibrium state. λ is a random vector in [0-1]. F denotes the exponential term and can be calculated with Eq. (4):

$$(F = e^{-\lambda(t-t_0)}) \quad (4)$$

Where t_0 is the initial time and t is a function linearly decreasing and calculated as the following Eq. (5):

$$t = \left(1 - \frac{it}{\max_{it}}\right)^{a_2\left(\frac{it}{max_{it}}\right)} \quad (5)$$

Where a_2 is the constant value related to the exploitation ability of the algorithm.

2.3. Aquila optimizer

Abualigah et al. developed the Aquila Optimizer (AO), a population-based metaheuristic algorithm, in 2021.[26] It was inspired by the Aquila's hunting behavior. It has four stages in the updating process, which are expanded exploration, narrowed exploration, expanded exploitation, and narrowed exploitation. The proposed updating equations at these four stages are given respectively as follows:

$$\begin{aligned}(x_1\,(t+1) = x_{best}\,(t) \,.\, \left(1 - \frac{it}{max_{it}}\right) \\ + (x_m\,(t) - x_{best}\,(t)\,.r)\end{aligned} \quad (6)$$

Table 1. Metaheuristic algorithms

No	Methods	Author	Inspiration	Year	Journal
1	Political Optimizer Engineering	Askari et al.[24]	Socio-inspired	2020	Knowledge-Based Systems Engineering
2	Equilibrium Optimizer Engineering	Faramarzi et al.[25]	Physics-inspired	2020	Knowledge-Based Systems Engineering
3	Aquila Optimizer	Abualigah et al.[26]	Nature-inspired	2021	Computers Industrial Engineering Engineering
4	Flow Directional Algorithm Engineering	Karami et al.[27]	Physics-inspired	2021	Computers Industrial Engineering Engineering
5	Cheetah Optimizer	Akbari et al.[28]	Nature-inspired	2022	Scientific Reports Engineering Engineering
6	Artificial Rabbits Optimization	Wang et al.[29]	Bio-inspired	2022	Engineering Applications of Artificial Intelligence
7	Golden Jackal Optimization	Chopra and Mohsin Ansari[30]	Nature-inspired	2022	Expert Systems with Applications
8	Gazelle Optimization Algorithm	Agushaka et al.[31]	Nature-inspired	2022	Neural Computing and Applications
9	Pelican Optimization Algorithm	Trojovský and Dehghani[32]	Nature-inspired	2022	Sensors
10	Crow Search Algorithm	Askarzadeh[33]	Nature-inspired	2016	Computers & Structures
11	Whale Optimization Algorithm	Mirjalili and Lewis[34]	Nature-inspired	2016	Advances in Engineering Software
12	Grey Wolf Optimizer	Mirjalili et al.[35]	Nature-inspired	2014	Advances in Engineering Software
13	Flower Pollination Algorithm	Xin-She Yang[36]	Nature-inspired	2012	Computing and Natural Computation
14	Adaptation Evolution Strategy	Hansen[37]	Evolution-inspired	2016	arXiv

$$(x_2(t+1) = x_{best}.Levy(D) + (x_r + (y-x).r) \tag{7}$$

$$(x_2(t+1) = x_{best}.Levy(D) + (x_r + (y-x).r) \tag{8}$$

$$x_4(t+1) = QF.x_{best}(t) - (G_1.x(t).r) - (G_2.Levy(D)) + r.G_1 \tag{9}$$

where x_{best} represents the best solution. $x_1(t+1)$, $x_2(t+1)$, $x_3(t+1)$, and $x_4(t+1)$ are the next solutions for four stages. it and max_{it} denote the current iteration and the max. number of the iterations, respectively. $x_m(t)$ is the mean of the population until t^{th} iteration. Levy represents the distribution function of the levy flight. x_r is a random solution. x and y parameters are used for spiral search behavior. a and δ are the exploitation parameters. QF denotes the quality function, which is used to balance between search behaviors. G_1 and G_2 are the motion parameter and slope of the flight, respectively. G_1 is in the range of [-1,1] and G_2 decreases from 2 to 0.

2.4. Flow directional algorithm

Flow Directional Algorithm (FDA) is a physic-based method developed by Karami et al in 2021.[27] The method is established by modeling the case of a fluid moving to the lowest outlet in a drainage basin to reach the optimal point. The Eq. (10) presents the updated the next value of the flow:

$$\mathrm{x}(\mathrm{i}+1) = \mathrm{x}(\mathrm{i}) + \mathrm{V}\frac{\mathrm{x}(\mathrm{i}) - \mathrm{nx}(\mathrm{j})}{||\mathrm{x}(\mathrm{i}) - \mathrm{nx}(\mathrm{j})} \tag{10}$$

Where nx(j) presents the value of the jth neighbor. x(i) and x(i+1) represent the position of the i^{th} and $(\mathrm{i}+1)^{\mathrm{th}}$ flow, respectively. V denotes the speed of the flow.

$$nx(j) = x(i) + \mathrm{randn} * \Delta \tag{11}$$

where rand is the random number distributed normally. Δ represent the flow direction, which can be obtained using Eq. (12):

$$\Delta = r \cdot x_r - r \cdot x(i) \cdot |x_{\mathrm{best}} - x(i)| \cdot w \tag{12}$$

Where r denotes the random value distributed uniformly and xr presents the randomly selected position. x_{best} represents the best solution. w is the random nonlinear weight in the range of (0, ∞) and determined by the following equation (13):

$$w = \left(\left(1 - \frac{it}{max_{it}} \right)^{2.randn} \right) . \left(rand. \left(\frac{it}{max_{it}} \right) \right) \quad (13)$$

where $randn$ denoted a random value in $[(-1)-(+1)]$. max_{it} is the number of the maximum iterations.

2.5. Cheetah optimization algorithm

Cheetah Optimizer is one of the methods inspired by nature and was introduced by Akbari et al. in 2022.[28] The algorithm is developed by taking inspiration from the behavior of the cheetahs during the hunting stage. According to hunting behaviors, the optimization method consists of three strategies, which are searching, sitting and waiting, and attacking. According to the search strategy, the new solutions are updated by using Eq. (14):

$$x_{i,j}(t+1) = x_{i,j}(t) + r_{i,j}^{-1}.a_{i,j}(t) \quad (14)$$

where $x_{(i,j)}(t)$ and $x_{(i,j)}(t+1)$ demonstrate the current and the next positions of the ith cheetah on d-dimension. $r_{(i,j)}^{-1}$ and $a_{(i,j)}(t)$ denote random parameters and the step length of the cheetah, respectively. According to the second strategy, sitting and waiting, the new solutions are calculated as in Eq. (15):

$$x_{i,j}(t+1) = x_{i,j}(t) \quad (15)$$

Finally, according to the third strategy, attacking, the new positions are found via the following equation (16):

$$x_{i,j}(t+1) = x_{\beta,j}(t) + r_{i,j}.\beta_{i,j}(t) \quad (16)$$

Where $r_{i,j}$ and $x_{\beta,j}(t)$ denote the turning parameter and position of the prey, respectively. $\beta_{i,j}(t)$ is the interaction parameter.

2.6. Artificial rabbit optimization algorithm

Artificial Rabbit Optimization is a nature-inspired algorithm proposed by Wang et al. in 2022.

$$\begin{aligned} v_i(t+1) = {} & x_j(t) + R.(x_i(t) - x_j(t)) \\ & + round(0.5.(0.05 + r_1)).n_1 \end{aligned} \quad (17)$$

where $v_i(t+1)$ is the position value of the ith rabbit at $(t+1)^{th}$ iteration. $x_j(t)$ denotes the current solution of the jth rabbit. n_1 and r_1 denote the parameter with normal distribution and the random parameter, respectively. The equations for the hiding stage are given in Eqs. (18-19):

$$v_i(t+1) = x_i(t) + R.(r_4.g - x_i(t)) \quad (18)$$

$$g(t) = x_i(t) + H.g_r.x_i(t) \quad (19)$$

Where r4 and H represent the random and the hiding parameters, respectively. g is the chosen burrow for hiding. Finally, an energy factor is modeled the provide balance between the exploration and the exploitation phases using Eq. (10).

$$A(t) = 4\left(1 - \frac{it}{max_{it}}\right)\ln\left(\frac{1}{r}\right) \quad (20)$$

where max_{it} indicates the number of the iterations (it).

2.7. Golden jackal optimization algorithm

Golden Jackal Optimization was also developed inspired by nature and was introduced by Chopra et al. in 2022.[30] It is based on the behavior of golden jackals during the hunting process. The hunting techniques of the jackals are modeled as three stages: searching, encircling, and attacking. The male jackal leads the prey. The jackals' positions are updated via Eq. (21) corresponding to the prey.

$$x(t+1) = \frac{x_m(t+1) + x_{fm}(t+1)}{2} \quad (21)$$

where $x_m(t+1)$ and $x_{fm}(t+1)$ demonstrate the male and female jackals obtained using Eqs. (22)-(23)

$$x_m(t+1) = x_m(t) - E.|x_m(t) - rl.p(t)| \quad (22)$$

$$x_{fm}(t+1) = x_f(t) - E.|x_f(t) - rl.p(t)| \quad (23)$$

where $x_m(t)$ and $x_m(t+1)$ denote the current and the next positions of the male jackal and $x_f(t)$ and $x_f(t+1)$ represent the current and the next value of the female jackal. p(t) is the prey position. E and rl indicate an energy function and a random vector generated with Levy distribution, respectively. When a jackal pair attacks the prey, its evading energy decreases, and the jackals surround the prey identified in the previous step.

The mathematical expression of this behavior is given in the following Eqs. (24-25).

$$x_m(t+1) = x_m(t) - E.\left|rl.x_m(t) - p(t)\right| \quad (24)$$

$$x_{fm}(t+1) = x_f(t) - E.\left|rl.x_f(t) - p(t)\right| \quad (25)$$

where E is the energy function and is calculated as $E = E_0.E_1.E_0$ and E_1 are the diminishing and starting energy of prey, respectively.

2.8. Gazelle optimization algorithm

The Gazelle Optimization Algorithm is one of the population-based methods developed by Agushaka et al in 2023.[31] Gazelles' survival skills are used as inspiration for developing the algorithm. It consists of two phases, which are exploration and exploitation. In the exploitation stage, the grazing gazelle is simulated while the predator is absent and chasing it. During the exploration phase, the gazelle notices the predator and moves towards a safe shelter. At the exploitation and exploration phase, the updated positions of the gazelles are calculated with Eq. (26).

$$x(t+1) = x(t) + s.R.R_B.(E(t) - R_B.x(t)) \quad (26)$$

where $x(t)$ is the current position of the gazelle. s-parameter indicates the velocity of the gazelle. R and R_B are random vectors represented by the uniform distribution and the Brownian motion. When the gazelle spotted the hunter, the motion of the gazelle was formulated by using Eq. (27).

$$x(t+1) = x(t) + s.\mu.R_L.(E(t) - R_L.x(t)) \quad (27)$$

where R_L is a random integers vector with Levy distribution. If the hunter chases the gazelle, the next motion of the gazelle is formulated as in Eq. (28).

$$\begin{aligned} x(t+1) &= x(t) \\ &+ s.\mu.CF.\ R.\ R_B.(E(t) - R_L.x(t)) \end{aligned} \quad (28)$$

where S denotes the upper level of the velocity. CF is formulated using Eq. (29).

$$C_F = \left(1 - \frac{it}{max_{it}}\right)^{\frac{2.it}{max_{it}}} \quad (29)$$

The hunter's success affects the gazelle's survival. This behavior allows the method to get rid of the local optimum and is formulated as the following equation (30).

$$x(t+1) = \begin{cases} x(t) + CF.[lb + R.(ub - lb), & \text{if } r \leq PSR \\ x(t) + PSR(1-r) + r].(x_{r1} - x_{r2}), & else \end{cases} \quad (30)$$

where PSR indicates the rate of the hunter's achievement. r_1 and r_2 indicate the randomly selected numbers. ub and lb denotes the bounds of the gazelles and U indicates the binary vector as generated by Eq. (31):

$$U = \begin{cases} 0, & if\ r < 0.34 \\ 1, & else \end{cases} \quad (31)$$

where r is a random number in [0,1].

2.9. Pelican optimization algorithm

Pelican Optimization Algorithm is one of the herd-based methods developed by observing and modeling the behavior of pelicans during the hunting stage. It was proposed by Trojovský et al in 2022.[32] The hunting stage consists of two stages: approaching the prey and flying over the water. After the pelican determines the position of the prey, it approaches the determined place. The behavior of the pelican is modeled mathematically as in Eq. (32).

$$x_{i,j}(t+1) = \begin{cases} x_{i,j}(t) + r.(p_j - I.x_{i,j}(t)), & if\ f_p < f_i \\ x_{i,j}(t) + r.(x_{i,j}(t) - p_j), & else \end{cases} \quad (32)$$

where $x_{i,j}(t)$, and $x_{i,j}(t+1)$ indicate the current and the next positions of the ith pelicans at the jth dimension, respectively. r is a constant number, I is the randomly chosen number as 1 or 2. p_j denotes the position of the prey at the jth dimension. f_p and f_i indicate the cost values of the prey and the i^{th} pelican, respectively. When they approach the water, pelicans stretch their wings, fly over it, and collect the captured fish in their pouches. Thus, they can catch and collect more fish in their hunting area. This demonstrates that the pelicans are quite effective at local search. The mathematical modeling of this behavior is presented in Eq. (33):

$$x_{i,j}(t) = x_{i,j}(t) + R\left(1 - \frac{it}{max_{it}}\right).(2r-1).x_{i,j}(t) \quad (33)$$

where max_{it} is the number of the iterations (it). R is a constant value and taken as 0.2. r denotes the random number.

3. Experimental studies

The experiments and coding are carried out on MATLAB software with version 2020a on a personal computer that contains an IntelCore-i7 9^{th} Gen processor, and 16GB RAM. The control of the speed and the balance angle for the two-wheeled robot is provided using the PI controllers. The mentioned metaheuristic methods in Section 2 are utilized for tuning the controllers' parameters. The model of two-wheeled robots is illustrated in Figure 1. In this figure, R denotes the radius of the wheel. The width, depth, and height of the body are represented as W, D, and H, respectively. Table 2 presents the vehicle's parameter and their definitions.

The model of the optimized control system for the two-wheeled vehicle is given in Figure 2. The fitness value of each candidate solution set generated by metaheuristic methods is sent to the control part of the robot. The equation of the fitness value is calculated using Eq. (34):

$$f\left(t\right)=0.6e+0.4M \tag{34}$$

Where M represents the maximum overshoot and e is the steady-state error of the system. The error acquired from the control model is passed through the function of the Integral Square Error (ISE) given in Eq. (35). In the metaheuristic algorithm, the fitness value is minimized by searching for the optimal solution until the stopping criteria are achieved.

$$ISE=\int e^{2}\left(t\right)dt \tag{35}$$

where the e(t) function is calculated as presented in Eq. (36).

$$e\left(t\right)\ =\ e_{velocity}\ \left(t\right)+\ e_{balance}\left(t\right) \tag{36}$$

A control signal is obtained with each candidate solution created by the metaheuristic algorithm, as shown in the control block diagram of the two-wheeled vehicle above. The effectiveness of the controller is assessed with the acquired control signal, and further effective solutions are attempted to be developed in conjunction with the prior solutions. As a result, the most efficient usage of the controller is ensured within the specified range. The settings of the parameters and their definitions for all metaheuristic methods are given in Table 3. Additionally, Table 4 presents the common control parameters used in all methods.

As shown in Table 4, the maxit and the number of solutions are taken to be the same in all algorithms, with values of 500 and 20, respectively. All considered algorithms run under 5-times. The parameters of the two PI controllers, which each have four parameters, denote the dimension of the problem and are set to the same value in all algorithms. To make an accurate comparison of the optimization algorithms, the ranges of PID control parameters were investigated in a wide search space as (0-800).

4. Results and discussion

Performance evaluations of the current metaheuristic algorithms are carried out on the same platform and the system. The performances of nine algorithms proposed in the last four years (detailed above) are analyzed by comparing them with five older algorithms, which are Whale, Grey Wolf, Crow Search, Covariance Matrix Adaptation Evolution Strategy, and Flower Pollination optimization algorithms. Each method is employed in the optimization stage of the control parameters to perform speed and balance controls of the two-wheeled vehicle. For a fair evaluation, all algorithms are performed under equal conditions. PI controllers are used for the balance and the speed control of the two-wheeled vehicle. The parameters of the PI controllers $(\mathrm{Kp}_1, \mathrm{Ki}_1, \mathrm{Kp}_2, \mathrm{Ki}_2)$ are obtained using the determined iteration and number of the search agents (see Table 4). The optimal PI controller parameter values obtained for all methods are presented in Table 5. In this table, Kp_1 and Ki_1 indicate the parameters of the speed controller, while Kp_2 and Ki_2 represent the parameters of the balance controller.

Table 5 shows that controller parameters are similar in the PO, AO, GOA, POA, GJO, FDA, EO, ARO, CSA, GWO, and FPA optimizers. For both controllers, Kp values which are obtained by using CO, WOA, and CMA-ES are bigger than the obtained Kp values with other methods. The Ki parameters of the balance controllers with CO and WOA are smaller, whereas these parameters acquired with CMA-ES are slightly bigger than the

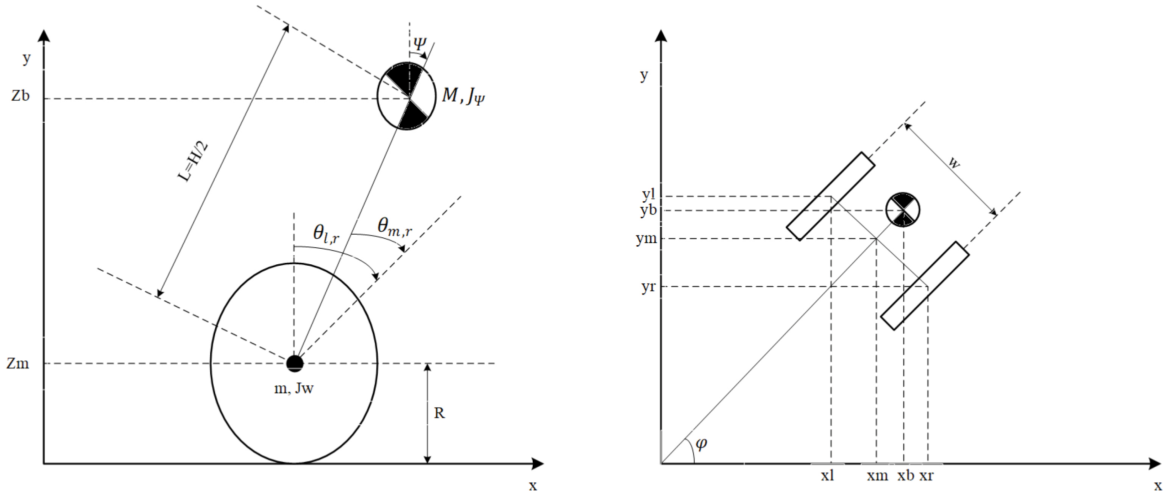

Figure 1. The model of two-wheeled vehicle[38]

Table 2. Parameters of two-wheeled vehicle

Param	Description	Values
W	Width of the body	0.15 m
H	Height of the body	0.2 m
D	Depth of the body	0.08 m
L	The distance from the body center	0.1
M	The weight of the body	0.8 kg
m	The weight of the wheels	0.05 kg
R	The radius of the wheels	0.05 m
J_w	Wheel inertia moment	$6.25e-05\,kg \cdot m^2$
J_m	Motor inertia moment	$1e-05\,kg \cdot m^2$

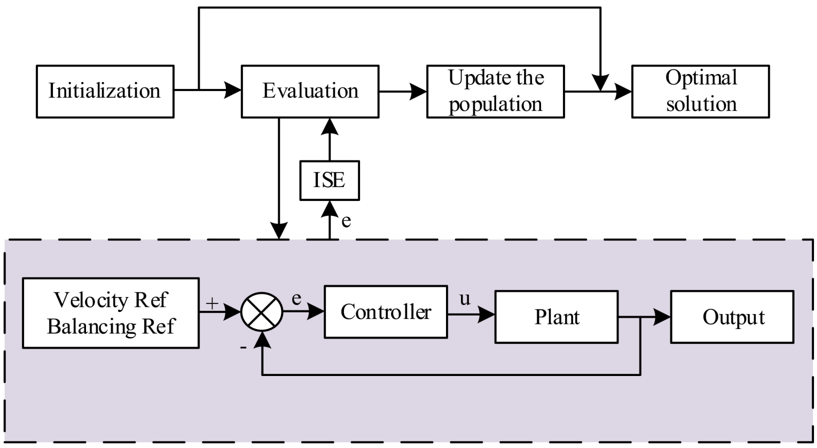

Figure 2. Block-diagram of the optimized control for the two-wheeled robot

other methods. WOA yields a smaller Ki parameter for the speed control, whereas CO finds this parameter bigger than other methods. To determine which of the obtained controller parameters is the best solution, the system responses shown in Figures 3-5 and Table 6-7 must be analyzed. The angle graphs and dynamic system properties of the two-wheeled vehicle produced when the obtained optimal controller parameters are applied to the system are presented in Figures 3-5. The responses of the balance angle for the two-wheeled vehicle are shown in Figure 3 for all considered metaheuristic algorithms. In the response produced using the CMA-ES method, it is seen that the maximum overshoot is high but it catches the reference at the zero point stably when it reaches stability.

The balance angle defines the position at which the vehicle remains upright. To ensure smooth and steady forward movement without tipping over, it is imperative that the robot navigates without excessive oscillations at the zero point of the balance angle. Figure 3 shows the balance

Table 3. Parameter settings of the metaheuristic methods

Method	Parameter and definition
PO	n = 0.8, number of the political parties.
	$\lambda = 1.0$, maximum limit of party switching rate.
AO	$\alpha = 0.1$, the constant value used in the exploitation phase.
	$\delta = 0.1$, the constant value used in the exploitation phase.
	$\beta = 1.5$, the constant value related to the Levy function.
	U = 0.265, a small value related to the narrowed exploration phase.
	$r_0 = 10$
	w = 0.005, a small value related to the narrowed exploration phase.
	$\varphi(0) = 3 * \text{pi}/2$.
	S = 0.01, the constant value related to the Levy function.
GOA	PSRs = 0.34, the success rate of the predator
	S = 0.88, maximum speed of the gazelle.
	$M = \{-1, 1\}$, direction of the gazelle
POA	-
GJO	$C_1 = 1.5$, constant value in Energy function.
	$\beta = 1.5$, the constant value related to the Levy function
FDA	$\beta = 1$, number of the neighbors.
EO	$a_1 = 2$, the constant value used for exploration ability.
	$a_2 = 1$, the constant value used for the exploitation ability.
	GP = 0.5, generation probability
ARO	-
CSA	pa = 0.25, the discovery rate of alien eggs/solutions.
	$\beta = 3/2$, levy parameter
WOA	b = 1; spiral coefficient
GWO	-
FPA	P = 0.8, probability switch.
	$\beta = 3/2$, levy parameter
CMA-ES	alpha_mu = 2, covariance update parameter

Table 4. Common control parameters of the metaheuristics

Parameter	Definition	Value
D	Problem dimension	4
Max_{it}	Maximum number of iteration	500
N	Number of the search agents	20
lb	Lower limit	1
ub	Upper limit	800
run	Number of runs	5

Table 5. Obtained optimal controller parameters for speed and the balance control

Methods	K_{p1}	K_{i1}	K_{p2}	K_{i2}	Methods	K_{p1}	K_{i1}	K_{p2}	K_{i2}
CO	78.5239	1.0038	388.3964	687.5466	EO	16.6323	3.5714	47.2498	553.1335
PO	16.7022	3.5738	47.4467	558.1976	ARO	16.5942	3.5767	47.1476	549.6950
AO	17.2016	3.8245	48.6753	621.7137	CSA	16.5347	3.4873	46.9722	545.7302
GOA	16.5988	3.5724	47.1621	549.8563	WOA	27.6981	1.0696	159.1850	1.01774
POA	16.5997	3.5458	47.1657	549.4160	GWO	16.6076	3.8515	47.1974	550.9385
GJO	16.5358	3.8075	47.0096	546.0188	FPA	16.7098	3.5376	47.4770	556.2386
FDA	16.6366	3.6054	47.2565	554.3169	CMA-ES	20.1506	30.848	53.1945	162.5135

responses are similar for PO, AO, GOA, POA, GJO, FDA, EO, ARO, CSA, GWO, and FPA optimizers. Whereas CO, PO, and CMA-ES methods yield the worst results according to the other optimizers. The reason for the difference between the answers is due to the change in the optimal control parameters given in Table 5. A very high Kp value causes the maximum overshoot to be high in the systems. However, although it reduces the steady-state error to some extent, it reduces the stability of the system. However, a very high Ki value also increases the overshoot. It is seen that the Kp2 and Ki2 parameters are larger in the CO method compared to the other methods. In the graph given for Region 2 in Figure 3, it is seen that the CO and PO methods produce a high

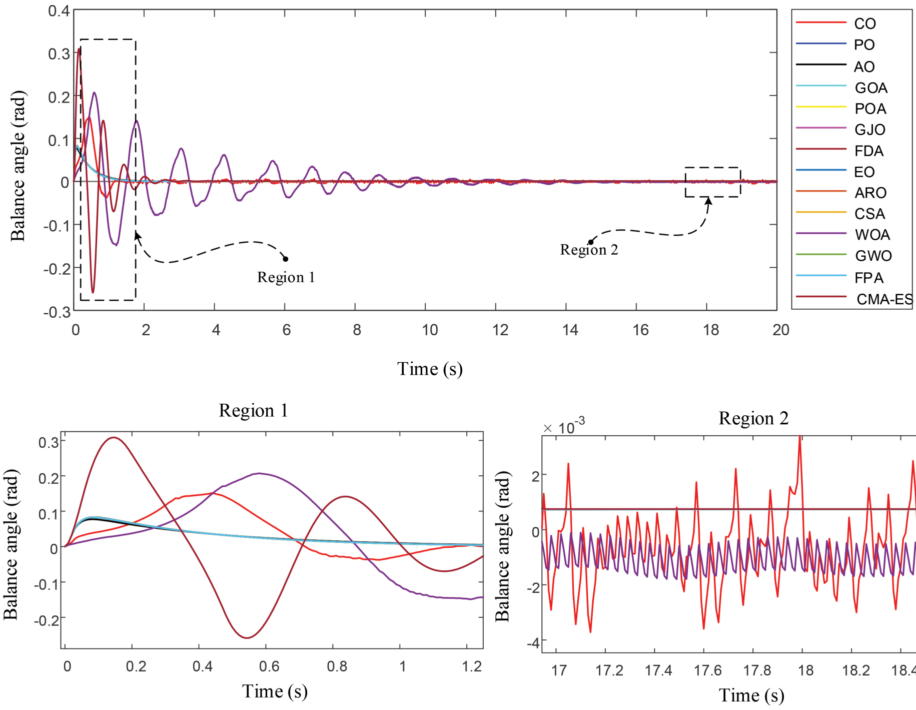

Figure 3. Balance response of two-wheeled vehicle for all methods

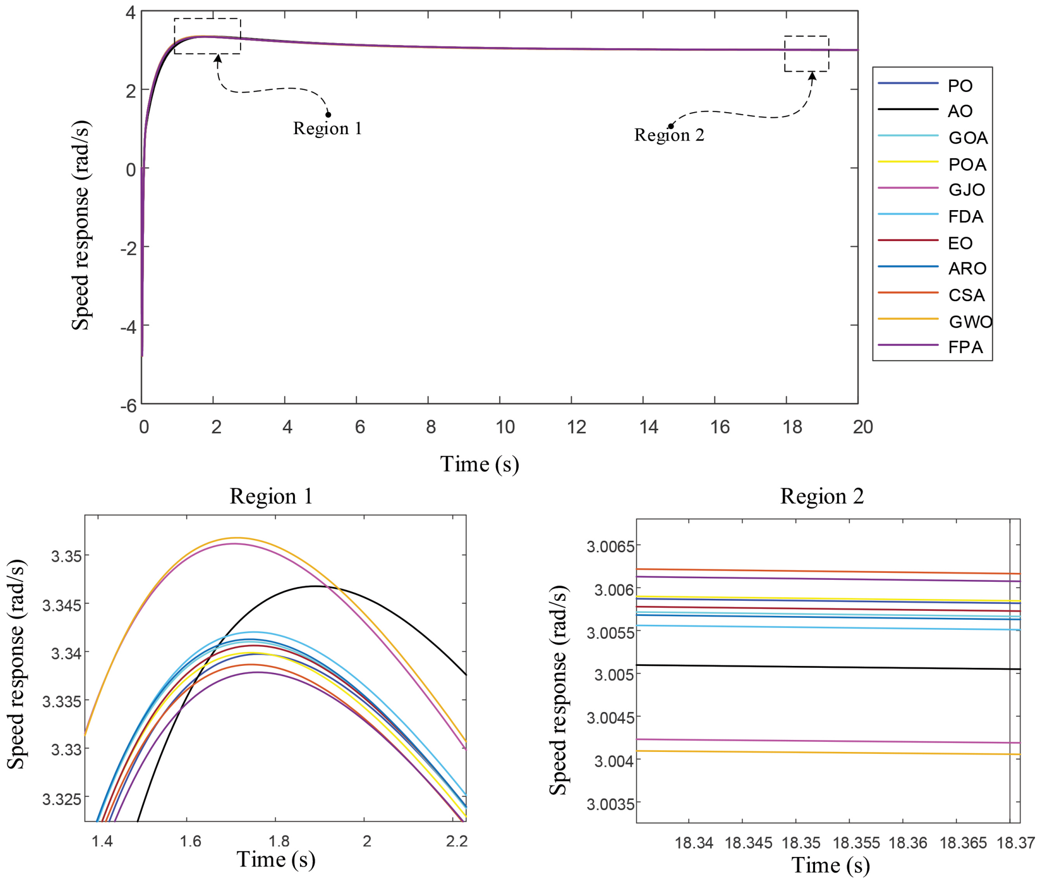

Figure 4. Speed responses of the two-wheeled vehicle with eleven metaheuristic algorithms

and excessive response due to the high Kp values. Since the parameter values of CO are higher, it also produces a larger oscillatory response compared to PO. As for the CMA-ES method, it is seen that there is an excessive oscillation in the response produced by this method at first, but then it reaches the reference at the zero point. The speed responses of the system are shown in Figure 4 for the following optimizers: PO, AO, GOA, POA, GJO, FDA, EO, ARO, CSA, GWO,

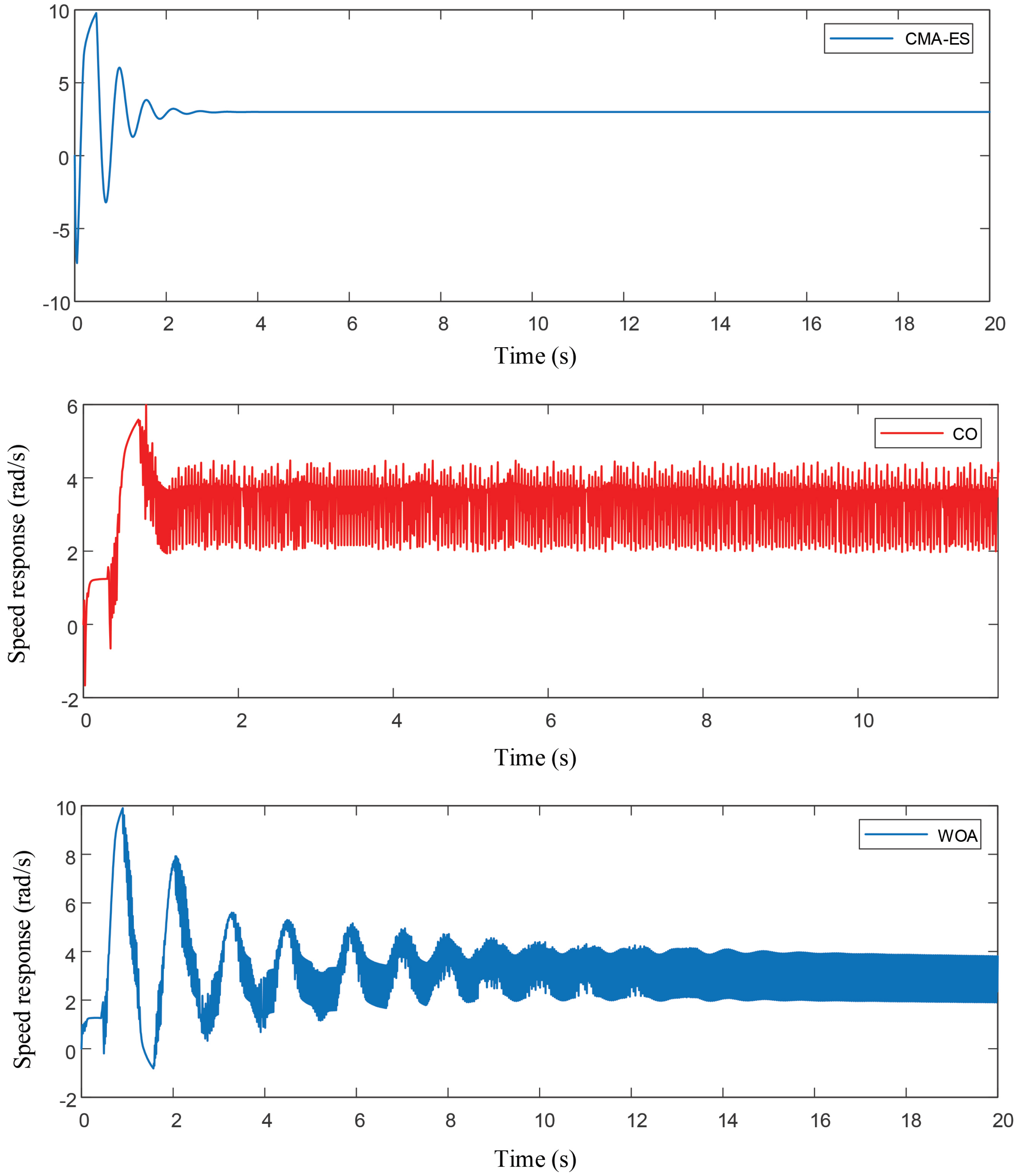

Figure 5. Speed responses of the two-wheeled vehicle with CMA-ES, CO, and WOA

and FPA. Additionally, Figure 5 illustrates the speed responses for the CMA-ES, CO, and WOA methods.

From Figure 4, the speed responses of the two-wheeled vehicle are found similar to each other for eleven metaheuristics. In this figure for Region 1-2, GJO and GWO have higher overshoot than the other methods and also have lower steady-state errors. In addition, the FPA and the CSA methods have the lowest overshoot and highest steady-state errors. Although there is a slight difference in the speed response curve due to small changes in the parameters, all algorithms given in Figure 4 produce similar and superior results for the speed control of the two-vehicle system. Due to the larger parameter differences in the CO, WOA, and CMA-ES methods, the speed signals are given separately in Figure 5. A step signal of 3 units is applied to the system as the speed reference.

Since the speed responses of the CO, WOA, and CMA-ES methods have differences from the responses of the others, the speed signals are presented separately in Figure 5. Firstly, it is seen from the figure that the Kp and Ki values for speed control are found as 20.1506 and 30.848 for the CMA-ES method (see Table 5). However, the control parameters for the GJO method, which

Table 6. Dynamic performance characteristics for balance angle

No	Method	T_r	T_s	T_p	M	ess
1	CO	0.0100	2.0400	0.4500	0.1521	1.20 e-03
2	PO	0	9.8800	0.1000	0.0817	7.47 e-04
3	AO	0	12.2000	0.0900	0.0766	7.47 e-04
4	GOA	0	11.0100	0.1000	0.0823	7.47 e-04
5	POA	0	12.8900	0.1000	0.0824	7.47 e-04
6	GJO	0	11.0400	0.1000	0.0827	7.47 e-04
7	FDA	0.0100	10.0900	0.1000	0.0820	7.47 e-04
8	EO	0	10.9800	0.1000	0.0821	7.47 e-04
9	ARO	0.0100	10.3500	0.1000	0.0823	7.47 e-04
10	CSA	0	11.0300	0.1000	0.0826	7.47 e-04
11	WOA	0.0100	19.8500	0.5900	0.2033	1.78 e-02
12	GWO	0	11.2200	0.1000	0.0824	7.47 e-04
13	FPA	0	10.0200	0.1000	0.0820	7.47 e-04
14	CMA-ES	0.01	4.38000	0.1600	0.3075	7.47 e-04

Table 7. Dynamic performance characteristics for speed responses

No	Method	T_r	T_s	T_p	M	ess
1	CO	0.4500	3.8270	0.7100	4.1331	0.0865
2	PO	0.6300	16.8500	1.7700	0.3397	2.5455e-05
3	AO	0.7000	14.6900	1.9000	0.3467	1.6699e-05
4	GOA	0.6300	16.8300	1.7500	0.3410	2.3981e-05
5	POA	0.6400	16.4900	1.7600	0.3399	2.5923e-05
6	GJO	0.6200	16.0000	1.7200	0.3511	1.1546e-05
7	FDA	0.6500	15.9600	1.7600	0.3420	2.2301e-05
8	EO	0.6400	15.1000	1.7600	0.3406	2.4552e-05
9	ARO	0.6200	12.9400	1.7500	0.3412	2.3618e-05
10	CSA	0.6400	15.0900	1.7600	0.3386	2.9485e-05
11	WOA	0.5800	15.8600	0.9000	7.0768	1.1954
12	GWO	0.6300	16.0600	1.7200	0.3518	1.0643e-05
13	FPA	0.6400	18.5100	1.7700	0.3378	2.8406e-05
14	CMA-ES	0.1600	4.95000	0.4800	6.7859	4.8849e-15

is determined to capture the reference in a very short time and with a very small steady-state error, are found to be 16.5358 and 3.8075. Although the Kp value of the CMA-ES method does not differ significantly, the unnecessary increase in the Ki value by approximately ten times causes excessive oscillation and delay in the system's speed response. Kp and Ki values of the CO algorithm are 78.5239 and 1.0038, respectively.

Figure 5 shows that the CO method produces an excessive oscillatory response due to its higher Kp value when compared to other methods. WOA finds the optimal parameter values for the two-wheeled vehicle as 27.6981 and 1.0696 for Kp and Ki, respectively. Similarly, the WOA method produces an oscillatory and delayed response as a result of the slightly higher Kp and lower Ki. The dynamic performance characteristics obtained from the system response for speed control are presented in Table 7. In the table, the terms of the Tr, Ts, and Tp represent the rise time, settling time, and peak time of the signals for all considered methods. M and ess denote the maximum overshoots and the steady-state error of the signals, respectively.

When the dynamic performance characteristics of the system, whose details are more clearly understood in Table 7, are examined, the lowest error is found with the CMA-ES method, while the overshoot is also at the highest value in this method. While WOA has the highest steady-state error, the lowest steady-state error after CMA-ES is obtained with the GJO method. When the balance of overshoot and steady-state error of the GJO and GWO methods is considered, it can be said that they are the methods that find the optimal solution with the most precision among the compared methods. However, the AO, FDA, ARO, GOA, EO, PO, POA, FPA, and CSA methods follow the GJO and GWO methods, respectively.

The convergence curves for all algorithms are presented in Figure 6. For a fair comparison, all algorithms are performed under the same conditions and on the same platform.

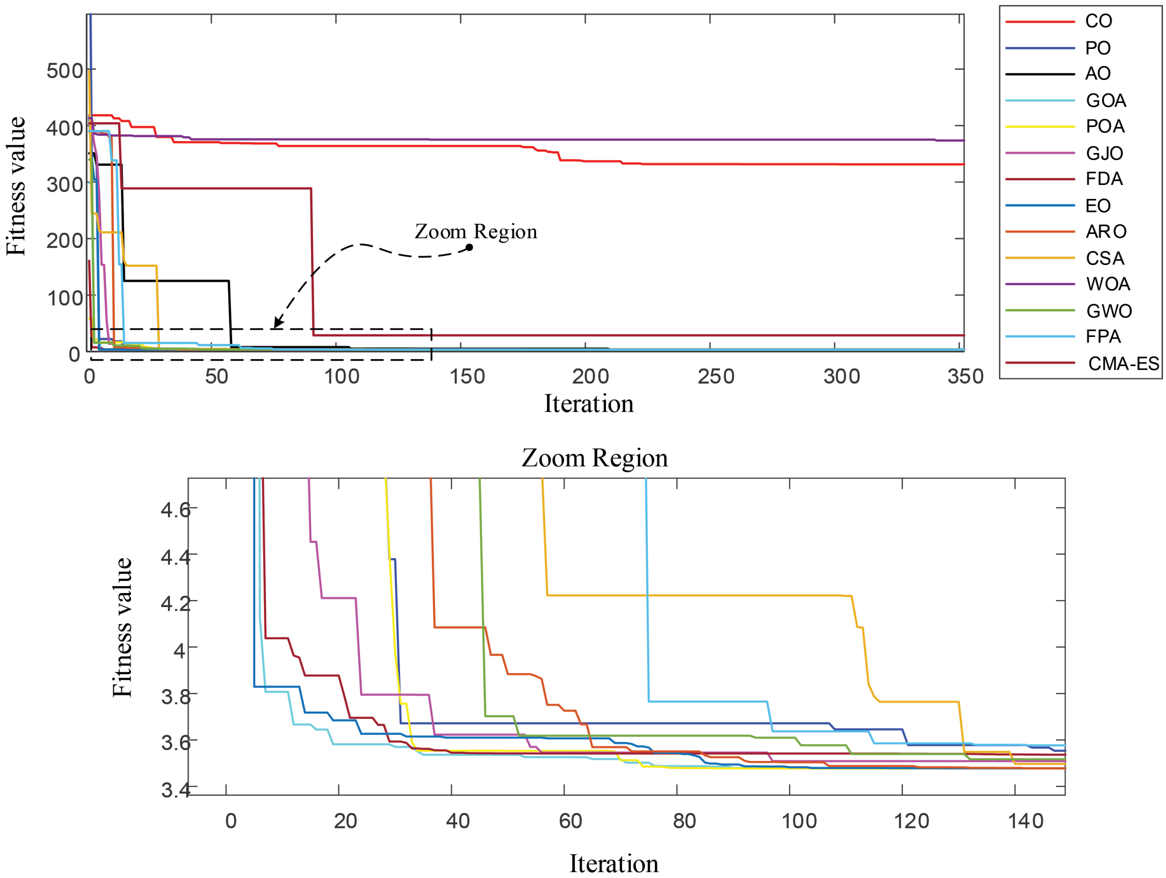

Figure 6. Convergence curves for all algorithms

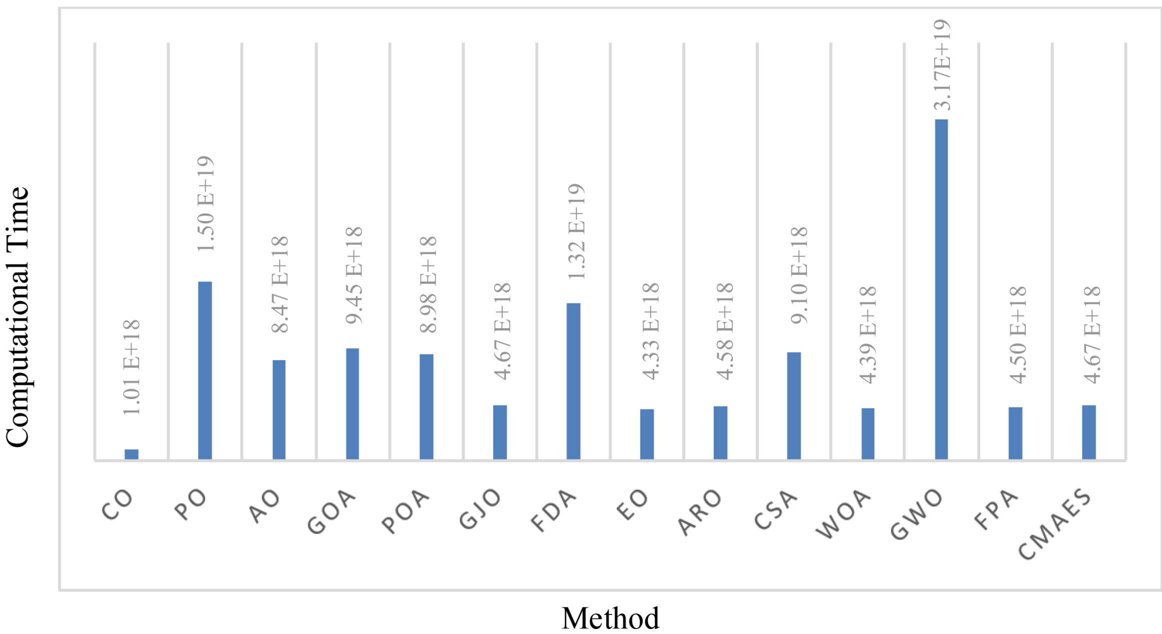

Figure 7. Computational times of the methods

In Figure 6, the fitness value given on the y-axis is calculated by considering the maximum overshoot and the steady-state error of the system, as expressed in Eq. (34). As can be seen in Figure 6, the worst results are produced with the WOA, CO, and CMA-ES optimizers, respectively. It is clearly seen in the given graphs that these methods cannot fully converge to the optimal point. When we look at the zoom region graph in Figure 6, the convergence speeds to the optimal point are given from the fastest to the slowest as: GOA, EO, FDA, GJO, POA, PO, GWO, ARO, FPA,

Table 8. Statistical results of the methods

No	Methods	Min	Mean	Max.	Std
1	CO	3.8723	75.7473	341.1709	148.4326
2	PO	3.5314	4.1219	5.0070	0.7211
3	AO	3.5161	4.6367	6.6439	1.3972
4	GOA	3.4950	3.5158	3.5716	0.0317
5	POA	3.4790	3.8902	4.9990	0.6589
6	GJO	3.4839	3.5038	3.5383	0.0222
7	FDA	3.4913	3.5625	3.6115	0.0518
8	EO	3.4775	3.5065	3.5461	0.0319
9	ARO	3.5161	4.6367	6.6439	1.3972
10	CSA	3.9982	73.3512	339.1747	148.6448
11	WOA	3.5171	6.8791e+218	3.4395e+219	Inf
12	GWO	3.4833	4.4826	6.8118	1.4526
13	FPA	3.6329	71.7655	338.1445	148.9295
14	CMAES	63.4304	317.4418	383.8003	142.0176

CSA. Computational times for all methods are presented in Figure 7. In terms of the computational times, CO method converged to the optimal solution in the shortest time. EO, GWO, FPA, ARO, and CMA-ES also converge to the optimal point shorter time than other algorithms. The GWO method is found the optimal point in the longest time.

The statistical results of all algorithms are presented in Table 8. The minimum, maximum, mean, and standard deviations are included in this table. The results for all algorithms are obtained with 5-runs.

In Table 8, PO, AO, GOA, POA, GJO, FDA, EO, ARO, CSA, GWO, and FPA methods produce similar results in terms of mean values, whereas the CO, WOA, and CMA-ES methods have the worst values. The GJO method is seen as the best algorithm, and EO, GOA, and FDA follow the GJO method in terms of the mean values. According to the minimum values of the algorithms, the EO, POA, GWO, and GJO algorithms produced the best results, respectively. Since the methods take a long time to run in such systems, it is important to reach the most optimal solution in a short time when solving such problems.

When examining the theoretical aspects of algorithms, it is clear that while the applications and common control parameters of metaheuristic algorithms are similar, the convergence behaviors in finding the optimal solution change due to the differences in the strategies used in the movement updates of the agents. This provides an overview of the effectiveness of the strategies employed in the methods.

5. Conclusion

This study presents a comprehensive analysis of current metaheuristics, most of which have never been used in this field, and an effective control design for a two-wheeled self-balancing vehicle. The optimal controller designed with considered metaheuristics, which are Flow Directional, Cheetah Algorithm and Politican, Equilibrium, Aquila, Artificial Rabbit, Golden Jackal, Gazelle, and Pelican Optimizers, are implemented to the two-wheeled vehicle. The balance and speed controller of the vehicle are performed with PI controllers. For this purpose, the cascade PI controller parameters are tuned with the metaheuristic methods. Since conventional PID tuning methods need more flexibility and robustness, metaheuristic methods-based tuning approaches are seen as crucial importance in solving complex engineering problems.[16, 38-40] The efficiencies of the optimization-based tuning algorithms are assessed, comparatively. The statistical results, convergence curves, dynamic system characteristics, and computation times are evaluated by performing all methods under equal conditions. In the qualitative and quantitative analysis, it is observed that 11 out of 14 compared algorithms, namely PO, AO, GOA, POA, GJO, FDA, EO, ARO, CSA, GWO, and FPA methods, produced similar optimal results in the speed and balance control of the two-wheeled vehicle. However, it is observed that CMA-ES, WOA, and CO methods do not give satisfactory results for tuning the optimal control parameters.

The control parameter determination with metaheuristics approaches is one of the best ways because they offer more efficient solution with their simple structures compared with the traditional methods. In addition, this problem is quite appropriate for demonstrating the success of the metaheuristic algorithms. This study will provide insights for researchers using metaheuristic methods in this field for where to start and which strategies and methods offer effective solutions. The limitation of this study is that the number of iterations and runs of the algorithms could not be taken very high due to the long computational time. In this respect, the number of iterations is taken as 500 at most and the number of runs could be increased up to 5. In the context of this disadvantage, it is understood that one of the most distinctive features of metaheuristic algorithms to be used in this field should be the running time. In future works, the performances of the considered algorithms can be improved with some modifications or hybrid methods can be developed. In addition, these methods can be effectively applied to determining different control parameters on various systems.

Acknowledgments

None.

Funding

None.

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization: Yagmur Olmez, Gonca Ozmen Koca
Formal analysis: Yagmur Olmez, Gonca Ozmen Koca
Methodology: All authors
Writing – original draft: All authors
Writing – review & editing: All authors

References

[1] Nssibi, M., Manita, G., & Korbaa, O. (2023). Advances in nature-inspired metaheuristic optimization for feature selection problem: A comprehensive survey. *Computer Science Review*, 49, 100559.

[2] Meenachi, L., & Ramakrishnan, S. (2021). Metaheuristic search based feature selection methods for classification of cancer. *Pattern Recognition*, 119, 108079.

[3] Yuan, X., Hu, G., Zhong, J., & Wei, G. (2023). HBWO-JS: jellyfish search boosted hybrid beluga whale optimization algorithm for engineering applications. *Journal of Computational Design and Engineering*, 10(4), 1615-1656.

[4] Han, M., Du, Z., Zhu, H., Li, Y., Yuan, Q., & Zhu, H. (2022). Golden-Sine dynamic marine predator algorithm for addressing engineering design optimization. *Expert Systems with Applications*, 210, 118460.

[5] Deng, L., & Liu, S. (2023). Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design. *Expert Systems with Applications*, 225, 120069.

[6] Calgan, H. (2023). Optimal C-type filter design for wireless power transfer system by using support vector machines. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 13(2), 151-160.

[7] Perin, D., Karaoglan, A. D., & Yilmaz, K. (2024). Rotor design optimization of a 4000 rpm permanent magnet synchronous generator using moth flame optimization algorithm. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 14(2), 123-133.

[8] Riaz, M., Bashir, M., & Younas, I. (2022). Metaheuristics based COVID-19 detection using medical images: A review. *Computers in Biology and Medicine*, 144, 105344.

[9] Olmez, Y., Koca, G. O., Tanyildizi, E., & Sengur, A. (2023). Multilevel image thresholding based on Renyi's entropy and golden sinus algorithm II. *Neural Computing and Applications*, 35(24), 17837-17850.

[10] Ozmen Koca, O. K., Dogan, S., & Yilmaz, H. (2018). A multi-objective route planning model based on genetic algorithm for cuboid surfaces. *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, 59(1), 120-130.

[11] Olmez, Y., Koca, G. O., Sengur, A., & Acharya, U. R. (2023). PS-VTS: particle swarm with visit table strategy for automated emotion recognition with EEG signals. *Health Information Science and Systems*, 11(1), 22.

[12] Yigit, T., & Celik, H. (2020). Speed controlling of the PEM fuel cell powered BLDC motor with FOPI optimized by MSA. *International Journal of Hydrogen Energy*, 45(60), 35097-35107.

[13] Arya, Y. (2018). AGC of two-area electric power systems using optimized fuzzy PID with filter plus double integral controller. *Journal of the Franklin Institute*, 355(11), 4583-4617.

[14] Mohanty, B., Panda, S., & Hota, P. K. (2014). Controller parameters tuning of differential evolution algorithm and its application to load frequency control of multi-source power system. *International journal of electrical power & energy systems*, 54, 77-85.

[15] Sathya, M. R., & Ansari, M. M. T. (2015). Load frequency control using Bat inspired algorithm based dual mode gain scheduling of PI controllers for interconnected power system. *International Journal of Electrical Power & Energy Systems*, 64, 365-374.

[16] Sahu, R. K., Panda, S., & Padhan, S. (2015). A hybrid firefly algorithm and pattern search technique for automatic generation control of multi area power systems. *International Journal of Electrical Power & Energy Systems*, 64, 9-23.

[17] Dash, P., Saikia, L. C., & Sinha, N. (2015). Comparison of performances of several FACTS devices using Cuckoo search algorithm optimized 2DOF controllers in multi-area AGC. *International Journal of Electrical Power & Energy Systems*, 65, 316-324.

[18] Dash, P., Saikia, L. C., & Sinha, N. (2015). Automatic generation control of multi area thermal system using Bat algorithm optimized PD–PID cascade controller. *International Journal of Electrical Power & Energy Systems*, 68, 364-372.

[19] Puangdownreong, D., Nawikavatan, A., & Thammarat, C. (2016). Optimal design of I-PD controller for DC motor speed control system by cuckoo search. *Procedia Computer Science*, 86, 83-86.

[20] Sahoo, B. P., & Panda, S. (2018). Improved grey wolf optimization technique for fuzzy aided PID controller design for power system frequency control. *Sustainable Energy, Grids and Networks*, 16, 278-299.

[21] Wang, L., Ni, H., Zhou, W., Pardalos, P. M., Fang, J., & Fei, M. (2014). MBPOA-based LQR controller and its application to the double-parallel inverted pendulum system. *Engineering Applications of Artificial Intelligence*, 36, 262-268.

[22] Demirtaş, M., & Ahmad, F. (2023). Fractional fuzzy PI controller using particle swarm optimization to improve power factor by boost converter. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA).*

[23] Özyetkin, M. M., & Birdane, H. (2023). The processes with fractional order delay and PI controller design using particle swarm optimization. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 13(1), 81-91.

[24] Askari, Q., Younas, I., & Saeed, M. (2020). Political Optimizer: A novel socio-inspired metaheuristic for global optimization. *Knowledge-based systems*, 195, 105709.

[25] Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-based systems*, 191, 105190.

[26] Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, 157, 107250.

[27] Karami, H., Anaraki, M. V., Farzin, S., & Mirjalili, S. (2021). Flow direction algorithm (FDA): a novel optimization approach for solving optimization problems. *Computers & Industrial Engineering*, 156, 107224.

[28] Akbari, M. A., Zare, M., Azizipanah-Abarghooee, R., Mirjalili, S., & Deriche, M. (2022). The cheetah optimizer: A nature-inspired metaheuristic algorithm for large-scale optimization problems. *Scientific reports*, 12(1), 10953.

[29] Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., & Zhao, W. (2022). Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 114, 105082.

[30] Chopra, N., & Ansari, M. M. (2022). Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Systems with Applications*, 198, 116924.

[31] Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2023). Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer. *Neural Computing and Applications*, 35(5), 4099-4131.

[32] Trojovský, P., & Dehghani, M. (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors*, 22(3), 855.

[33] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & structures*, 169, 1-12.

[34] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95, 51-67.

[35] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.

[36] Yang, X. S. (2012, September). Flower pollination algorithm for global optimization. In *International conference on unconventional computing and natural computation* (pp. 240-249). Berlin, Heidelberg: Springer Berlin Heidelberg.

[37] Hansen, N. (2016). The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772.*

[38] Olmez, Y., Koca, G. O., & Akpolat, Z. H. (2022). Clonal selection algorithm based control for two-wheeled self-balancing mobile robot. *Simulation Modelling Practice and Theory*, 118, 102552.

[39] Sharma, A., & Singh, N. (2024). Load frequency control of connected multi-area multi-source power systems using energy storage and lyrebird optimization algorithm tuned PID controller. *Journal of Energy Storage*, 100, 113609.

[40] Ozmen Koca, G., & Korkmaz, D. (2019). Neural network based control of a two-mass drive system. *International Journal of Intelligent Systems and Applications in Engineering*, 7(2), 92-98.

Yagmur Olmez *earned her M.Sc. and Ph.D. degrees in Mechatronics Engineering from the Institute of Science at Firat University, Elazig, Turkiye, in 2018 and 2024, respectively. Currently, she is a Research Assistant in the Department of Mechatronics Engineering at the Faculty of Engineering, Nigde Omer Halisdemir University. Her research interests encompass image processing, metaheuristic algorithms, artificial intelligence, system modeling, and advanced control techniques.*

https://orcid.org/0000-0002-1615-7390

Gonca Ozmen Koca *was born in Elazig, in 1980. She received a master's degree from the Department of Electronics and Computer Education, Institute of Science, Firat University, in 2005, and a Ph.D. degree from the Electric and Electronic Engineering Department, Firat University. She is currently an Associate Professor with the Control Division, Mechatronics Engineering Department, Technology Faculty, Firat University. Her research interests include control systems, path planning, biomimetic systems, intelligent control techniques, and optimization methods.*
https://orcid.org/0000-0003-1750-8479

Zuhtu Hakan Akpolat *was born in Elazig, Turkey, in 1967. He received his Ph.D. degree from the Department of Electrical and Electronics Engineering, University of Nottingham, Nottingham, U.K., in 1999. He is currently a Professor in the Department of Electrical and Electronics Engineering, Faculty of Engineering, Fatih Sultan Mehmet Vakif University, Istanbul, Turkey. His research interests include control of electrical drives, intelligent control systems, emulation of mechanical loads, fuzzy logic, optimization methods, and biomimetic systems.*
https://orcid.org/0000-0002-7935-7031

An International Journal of Optimization and Control: Theories & Applications
(https://accscience.com/journal/ijocta)