



Enhanced Arithmetic Optimization Algorithm for Parameter Estimation of PID Controller

Mohamed Issa^{1,2}

Received: 3 March 2022 / Accepted: 19 July 2022
© The Author(s) 2022

Abstract

The Proportional-Integral-Derivative (PID) controller is a key component in most engineering applications. The main disadvantage of PID is the selection of the best values for its parameters using traditional methods that do not achieve the best response. In this work, the recently released empirical identification algorithm that is the Arithmetic Optimization Algorithm (AOA) was used to determine the best values of the PID parameters. AOA was selected due to its effective exploration ability. Unfortunately, AOA cannot achieve the best parameter values due to its poor exploitation of search space. Hence, the performance of the AOA exploit is improved by combining it with the Harris Hawk Optimization (HHO) algorithm which has an efficient exploit mechanism. In addition, avoidance of trapping in the local lower bounds of AOA–HHO is enhanced by the inclusion of perturbation and mutation factors. The proposed AOA–HHO algorithm is tested when choosing the best values for PID parameters to control two engineering applications namely DC motor regulation and three fluid level sequential tank systems. AOA–HHO has superiority over AOA and comparative algorithms.

Keywords PID controller · Arithmetic optimization algorithm (AOA) · Harris Hawk optimization algorithm (HHO)

Abbreviations

ZN	Ziegler–Nichol
SCA	Sine–cosine optimization algorithm
PSO	Particle swarm optimization
IMO	Ions motion optimization
MFO	Moth-flame optimization
IAE	Integral absolute error
dPSO	Dynamic PSO
IWOA	Improved whale optimization algorithm
OBL-HGS	Opposition Henry gas solubility
IWO	Invasive weed optimization
FPA	Flower pollination algorithm
GWO	Grey Wolf optimization
TLBO	Teaching learning based optimization
DE	Differential evolution

NFL	No-free-lunch
HHO	Harris Hawk optimization
CM	Cauchy mutation
GM	Gaussian mutation

1 Introduction

PID controller is used in manufacturing industries for process control due to their effectiveness, robustness and durability [1]. PID controller has common control parameters such as system stability, the time it takes for the process to settle (settling time), and bypass and error between the desired response and the actual response [1]. Due to the sharing of processes in factories, parameter setting is an important task, and proper configuration allows to obtain efficient transient performance in terms of minimum settling time, steady-state error, maximum bypass and rise time as possible. The three parameters of PID controller are proportional gain (K_p), integral gain (K_i) and derivative gain (K_d).

The main advantages of PID controller are concluded as in the following [1]:

✉ Mohamed Issa
Mamohamedali@eng.zu.edu.eg

¹ Computer and Systems Department, Faculty of Engineering, Zagazig University, Zagazig, Egypt

² Faculty of Computer Science, NAHDA University in Beni-Suef, New Beni Suef City, Egypt



1. The P controller is used for stabilizing the gain and producing a constant steady-state error
2. The I controller is used to eliminates or decreases the steady-state error
3. The D controller is used to decrease the rate of change of error, overshoot and settling time.

PID controller is used to regulate many industrial process such as pressure, temperature, flow rate, feed rate, weight, speed and position [1]. There are three categories for tuning the parameters of the PID controller, which are analytical methods, rule-based methods, and numerical methods [2]. The most common method is Ziegler–Nichol (ZN) [3] to be the classic method for adjusting parameters of the PID controller and has been classified as analytical method. ZN does not offer the best performance.

Heuristic optimization algorithms can be used to tune PID's parameters that have been classified as numerical methods and have been popular in the literature. Stochastic optimization methods such as heuristic algorithms [4] was suitable for tuning PID parameters because it treats the problem as a black box and adjusts the parameters and fitness tracking (fitness function) to achieve the optimum value. A meta-heuristic algorithm is a search-based algorithm that speeds up the exploration of problem's search space depending on a random motion to detect an acceptable solution in an acceptable time [4]. The meta-heuristic algorithm mimics the search methods from physics, humans, or nature. Sine—Cosine Optimization algorithm (SCA) [5] that drills into the search space by drawing search agents toward the best-established region based on cosine and sine factors. Besides, the Particle Swarm Optimization algorithm (PSO) [6] simulates the search strategy of birds flowing from nature. Also, there are a lot of released algorithms such as Ions Motion Optimization (IMO) [7], Lightning Attachment Procedure Optimization [8], Moth-Flame Optimization (MFO) [9], and other hundreds of algorithms are developed.

Meta-heuristic algorithms have successfully improved many engineering problems in fields as diverse as bioinformatics [10–17], Motor design [18, 19], Solar Energy [20], Robot design [21], Passive suspension system optimization [22] and many others. Many options are available to design the controllers with a lot of meta-heuristics algorithms in the literature. Such as particle swarm optimization algorithm was used for finding the optimum design of PID controller in the AVR systems [23] [24].

As shown in Fig. 1, meta-heuristic algorithm adjust the three parameters of the PID controller to improve the process response performance where $e(t)$ and $u(t)$ are the input and output signals of the PID controller, respectively. $e(t)$ is the error signal which is the difference between the set point signal ($h(t)$) and the output response of the process to be controlled ($y(t)$). $u(t)$ represents the controlled signal that

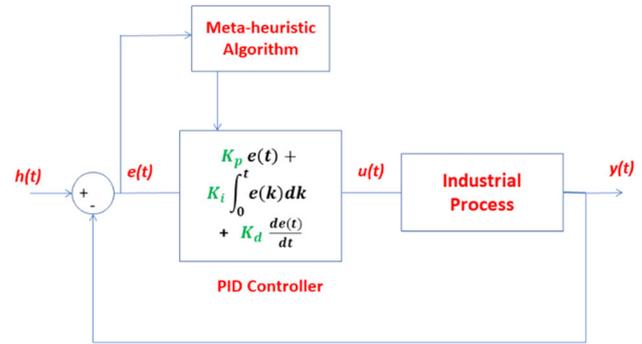


Fig. 1 Adjust PID parameters through meta-heuristic algorithm

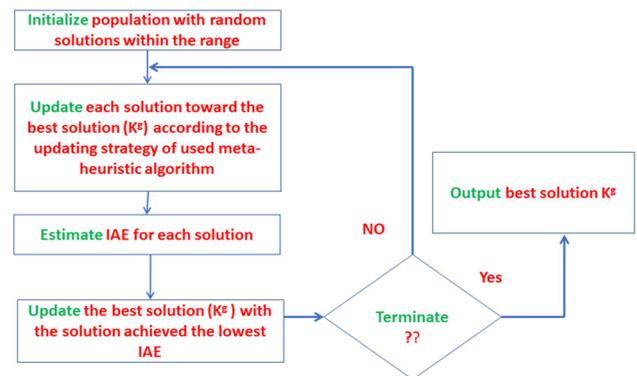


Fig. 2 The flowchart of tuning the PID's parameters using meta-heuristic algorithms

output from the PID controller and applied on the process to be controlled.

The objective function used to improve process performance is the minimum integral to absolute error (IAE). It is the sum of the differences between the desired response ($h(t)$) and the actual response ($y(t)$) which is represented as in Eq. (1) [1].

$$IAE = \int_0^{\infty} |y(t) - h(t)| dt \quad (1)$$

The procedure of estimating the values of PID's parameters (K_p , K_i and K_d) as shown in Fig. 2. The solutions are updated using the updating strategy of the meta-heuristic techniques for several iterations. During each iteration, the best solution K^g (K_p^g , K_i^g and K_d^g) which produce the best IAE is updated with the best solution founded during updating of solutions.

In the literature, there are several improvements in meta-inference for optimal design of a PID controller such as a constraint PSO (CPSO) [25], dynamic PSO (dPSO) [26], opposition-based Henry gas solubility optimization algorithm (OBL-HGS) [27], and improved whale optimization algorithm (IWOA) [28]. Besides various other descriptive

algorithms that have been used to optimize the parameters of the PID controller to improve the performance of DC motor as Invasive Weed Optimization (IWO) [29], Flower Pollination Algorithm (FPA) [30], FireFly [31] and Grey Wolf Optimization Algorithm (GWO) [32].

For other systems such as controlling the voltage regulator, Teaching Learning Based Optimization (TLBO) algorithm was used to optimize the parameters of PID controller [33]. Differential Evolution (DE) and its improved version (PSO-DE) [34] were used to optimize three liquid level tank systems.

The theory of No-Free-Lunch (NFL) [35] which states that “no one optimization algorithm can solve all engineering problems with the same efficiency”. Hence, upgraded versions of meta-inference based on embedding operators within algorithms or mixed meta-inference have been proposed to improve the algorithms for a particular engineering application. The nature of the optimization problem of the PID controller parameters is continuous because the search space has a large number of possible solutions which make the search fall into local minima besides its nonlinear behavior. Hence, a meta-heuristic algorithm containing efficient exploration and exploitation schemes is necessary to provide a better selection of PID controller parameters that achieved a better fit than those available in the literature.

Arithmetic Optimization Algorithm (AOA) [36] is a population-based algorithm that relies on the arithmetic operators of the search strategy. The main advantage of AOA is its effective exploration scheme which is defined as the ability of the search agents to visit most portions of the search space. However, the AOA exploitation scheme is poor resulting in a lack of microfitness and needs further enhancement. In this work, AOA has been improved by combining it with a heuristic algorithm which has an effective exploit strategy for tuning PID’s parameters.

A lot of researches were done on AOA; a version of AOA released was used for multiobjective optimization [37]. AOA was used for multilevel threshold segmentation optimization problem of COVID-19 images [38]. Besides, AOA was used for truss optimization [39], optimal installation of distribution system [40], Fog computing [41], energy storage system [42], economic load dispatching [43], optimal power flow problem [44], brain computer interface [45], photovoltaic solar cell parameter extraction [46], optimal energy resource planning [47], intrusion detection system [48], and PEM fuel cell parameters estimation [49]. AOA was merged with Slime Mold Algorithm for global optimization [50]. The hybridization was tested on 23 mathematical benchmark functions and three classical engineering problems. AOA was merged with genetic algorithm for feature selection problem [51]. An improved version of AOA was released based on using high-density distribution function (beta distribution) to enhance the exploration scheme of AOA [52]. The enhanced version

was tested on 30 mathematical benchmark function and engineering problems such as welded beam design, compression spring design and pressure vessel design. AOA was hybrid with Aquila optimizer for high-dimensional optimization problems [53]. A chaotic AOA was released for enhancing the speed convergence and avoiding local optima [54]. AOA was merged with differential evolution for truss structure optimization problem [55].

This paper presents a hybrid between AOA and Harris Hawk Optimization (HHO) [56] which has the benefits of efficient AOA exploration in addition to the efficient exploitation strategy of the hybrid algorithm. HHO algorithm is a population algorithm that inspired the attacking strategy of Harris Hawk for catching a prey [56]. The exploitation of search space by HHO is based on four schemes based on rabbit escape energy and rabbit escape chance. The four schemes diversify the movement patterns of hawksbill positions toward the best founding position (rabbit site), thus promoting intensification and head-avoidance in the local optima. HHO was used for optimizing many optimization problems such as feature selection [57], photovoltaic solar cell parameter extraction [58], color multilevel thresholding image segmentation [59], drug discovery [60], landslide susceptibility [61], and passive suspension system [22].

HHO was used for enhancing the exploitation of many meta-heuristic algorithms such as Nelder–Mead simplex optimization algorithm [62], grasshopper optimization algorithm [63], Salp swarm optimization algorithm [64], and equilibrium optimization [65],

The main contributions of this work are listed as follows:

1. Enhanced AOA’s exploitation by incorporating HHO for optimized PID controller design.
2. Avoidance of local AOA minima is enhanced based on the inclusion of disruption and mutation operators.
3. A DC motor and three liquid level tanks were used in pilot tests to test the performance of the developed AOA–HHO for the optimized design of the PID controller.

The rest of the paper is organized as follows: Sect. 2 describes the strategy of AOA and HHO, Sect. 3 describes the proposed hybrid algorithm (AOA–HHO), while Sect. 4 describes the procedure for estimating PID controller parameters based on AOA–HHO. Experimental results and discussion are presented in Sect. 5. Finally, the result of the proposed work is presented in Sect. 6.

2 Preliminaries

In this section, the procedure of AOA [36] and HHO [56] is presented.

2.1 AOA Algorithm

AOA uses arithmetic operators (addition, subtraction, multiplication, and division) to update solutions. To explore the search space, it depends on the multiplication and division operators, while the addition and subtraction operators are used for exploitation. The control parameter that balances diversification and intensification of search space is the accelerated Math Optimizer (M_{OA}) which is described in Eq. (2).

$$M_{OA}(t) = \frac{t}{T} \tag{2}$$

where t is the number of current iteration and T is the number of iterations.

Diversification of the search space based on division and multiplication factors is performed due to the high distribution of the generated values which are represented according to Eq. (3) for a condition ($rand > M_{OA}$) where $rand$ is a random generated number.

$$x_i^{(t)} = \begin{cases} best(x^{(t)}) \div ((M_{OP} + \varepsilon)((UB - LB)\mu + LB)) & r_1 < 0.5 \\ best(x^{(t)}) \times (M_{OP})((UB - LB)\mu + LB) & else \end{cases} \tag{3}$$

where x is the solution, i is the index of solution ($i=1:N$), $best(x)$ is the best global solution, ε and μ are constants, LB and UB are the lower and upper bound of the solutions, r_1 is a random generated number, Math Optimizer probability (M_{OP}) is a scaling parameter that produces more exploration and is estimated according to Eq. (4) where α is a constant parameter.

$$M_{OP}(t) = 1 - \frac{t^{1/\alpha}}{T^{1/\alpha}} \tag{4}$$

In terms of search space condensation, addition and subtraction operators are used due to the high density of solutions generated and implemented for a case ($rand > M_{OA}$). Equation 5 expresses the updating strategy during exploitation where r_2 is a random generated number.

$$x_i^{(t)} = \begin{cases} best(x^{(t)}) - ((M_{OP})((UB - LB)\mu + LB)) & r_2 < 0.5 \\ best(x^{(t)}) + (M_{OP})((UB - LB)\mu + LB) & else \end{cases} \tag{5}$$

AOA's procedure is listed in Algorithm 1.

Algorithm 1: The procedure of AOA

1:	Initialize $N, T, \mu,$ and ε
2:	Generate random solution ($x_i, i = 1:N$) in the range (LB, UB)
3:	Estimate the fitness of solutions
4:	Update the global best solution $best(x)$
5:	While T isn't reached yet
6:	Update M_{OA} and M_{OP}
7:	If ($rand > M_{OA}$)
8:	If ($r_1 < 0.5$), update solutions using division operator in Eq. 3
9:	Else , update solutions using multiplication operator in Eq. 3
10:	Else
11:	If ($r_2 < 0.5$), update solutions using subtraction operator in Eq. 5
12:	Else , update solutions using addition operator in Eq. 5
13:	End If
14:	Update the global best solution $best(x)$
15:	End While
16:	Output the best solution

2.2 HHO Algorithm

The HHO algorithm is a population-based heuristic that simulates the surprise-and-hunt mechanism of Harris Hooke. The prey is surprised by many hawks who cooperate to pounce on it, and according to the surrounding environment conditions and the escape method, the technique of chasing the hawk will be determined.

HHO's main feature is the hawks' cooperative way of attacking prey as more experienced hawks grab it. The attack mechanism and escape pattern of the prey are mathematically modeled where the search agents are represented by the hawks and the best solution is represented by the prey.

The HHO exploration stage is represented by the initial attack of the prey as it has a high energy which decreases during escape to a low energy level and can be attacked which represents the exploitation stage as shown in Eq. (6).

$$E^t = 2E_0(1 - \frac{t}{T}) \tag{6}$$

where (E) represents the energy of the prey, (E_0) is the initial energy of prey, and it has a value between (-1,1) which is assigned randomly, (t) is the current iteration number, and (T) is the total iterations number. For ($|E| > 1$), the exploration phase is executed, while for ($|E| \leq 1$) the exploitation phase is executed.

The exploration stage is carried out on two mechanisms: first, the falcons settle according to the positions of other falcons and the location of the prey. Second Mechanism Falcons can settle into a random position within the range of other falcons. Equation 7 simulates the exploration phase.

$$y^{t+1} = \begin{cases} y_{rand}^t - r_1 |y_{rand}^t - 2r_2 y^t| & c \geq 0.5 \\ (y_{rabbit}^t - y_{av}^t) - r_3(LB + r_4(UB - LB)) & c < 0.5 \end{cases} \tag{7}$$

where y^{t+1} is the solutions in the next iteration of search agents, y_{rand}^t is a hawk of the search agents is selected randomly, y^t is the solutions of the search agents at iteration (t), y_{rabbit}^t is the global solution among the search agents, and y_{av}^t is the average of all search agents' solutions at iteration (t). r_1, r_2, r_3 and r_4 and are random generator numbers within the range (0,1).

During the exploitation phase ($|E| \leq 1$), the hawks can attack hardly ($|E| < 0.5$) or softly ($|E| \geq 0.5$); in addition, there is a probability for escaping for the rabbit (r) where ($r < 0.5$) the rabbit escapes successfully and for ($r \geq 0.5$) it is caught. The various chasing style according to the energy (E) and the probability of escaping (r) are as follows:

2.2.1 A. Un-successful escaping ($r \geq 0.5$) and Soft besieges ($|E| \geq 0.5$)

Hawks surround the rabbit while it tries to escape and is tired and then pounced by the hawks. This process is simulated as in Eq. (8).

$$y^{t+1} = (y_{rabbit}^t - y^t) - E|Jy_{rabbit}^t - y^t| \tag{8}$$

where (J) simulates the rabbit's jump power randomly through escaping and $J = 2(1-r_5)$, r_5 is a parameter its value is selected randomly within the range [0,1].

2.2.2 B. Successful escaping ($r < 0.5$) and Soft besieges ($|E| \geq 0.5$)

The rabbit (the prey) escapes in a zigzag pattern simulated using mega-flying. The hawks search for the best direction of attack to catch the hare so that the hawks decide the next attack according to Eq. (9).

$$X = y_{rabbit}^t - E|Jy_{rabbit}^t - y^t| \tag{9}$$

Levy's flight function is used to simulate the random and irregular attacks of hawks to capture prey according to Eq. (10).

$$Z = X + s \times LF() \tag{10}$$

where s is a random value generated within the range (0,1) and LF() is the levy flight function represented according to Eq. (11).

$$LF() = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1 + \beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{(1+\beta)}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}} \tag{11}$$

where β is a set constant as a value (1.5) and u and v are randomly generated values within the range (0,1). The position of the falcon is estimated according to Eq. (12).

$$y^t = \begin{cases} X & \text{if } F(y^t) > F(X) \\ Z & \text{if } F(y^t) > F(Z) \end{cases} \tag{12}$$

where Z and X are estimated according to Eq. (10) and Eq. (9) in order.

2.2.3 C. Un-successful escaping ($r \geq 0.5$) and Hard besieges ($|E| < 0.5$)

The prey is ejected from the air and has a low energy to lunge, so the hawk veers strongly toward the hare to carry out Sally's attack. The update of the current position of the hawk toward the hare is simulated according to Eq. (13).

$$y^{t+1} = y_{rabbit}^t - E|y_{rabbit}^t - y^t| \tag{13}$$

2.2.4 D. Successful escaping ($r < 0.5$) and Hard besieges ($|E| < 0.5$)

The hawk glides toward the hare aggressively in order to deflate, but it attempts to limit prey by a small distance to the hawk's average positions. Besides, levy flight is used to simulate the zigzag way of prey and the rare movements of falcons. Equations (14) and (15) update the situation, while Eq. (16) determines the final position.

$$X = y_{rabbit}^t - E|Jy_{rabbit}^t - y_{av}^t| \tag{14}$$

$$Z = X + s \times LF() \tag{15}$$

$$y^t = \begin{cases} X & \text{if } F(y^t) > F(X) \\ Z & \text{if } F(y^t) > F(Z) \end{cases} \tag{16}$$

Algorithm (2) represents the steps of HHO algorithm.

Algorithm 2: The procedure of HHO

```

1: Commence (N) search agents (yi ; i ∈ (1: N))
2: While T isn't reached yet
3:   Estimate the fitness of the solutions
4:   Find the global best solution (yrabhi)
5:   Assess the escaping energy (E) as in Eq. 6.
6:   If (|E| ≥ 1) Then
7:     Update the movement according to Eq. 7
8:   Else
9:     If (|E| ≥ 0.5) and (|r| ≥ 0.5) Then
10:      Update the location according to Eq. 8
11:     Else If (|E| ≥ 0.5) and (|r| < 0.5) Then
12:      Update the location according to Eq. 9 to Eq. 12
13:     Else If (|E| < 0.5) and (|r| ≥ 0.5) Then
14:      Update the location according to Eq. 13
15:     Else
16:      Update the location according to Eq. 14 to Eq. 16
17:     End If
18:   End If
19:   Iterate from step (6) for (N) search agents
20: End While
21: Produce (yrabhi) as the global best solution.

```

3 The Proposed Hybrid Algorithm (AOA–HHO)

The main advantage of AOA is efficient exploration due to division and multiplication operators due to their strong ability to generate values with high distribution. However, it produced poor performance for PID controller's parameters estimation because it is trapped in local minima and has poor exploitability. Hence, this is the impetus to enhance avoidance of local minima by using the disruption operator and the mutation operator. Its poor exploit is improved by incorporating the exploitation scheme of the HHO algorithm (Heidari et al., 2019) where its main advantage is the different efficient exploitation mechanisms that balance the search focus on narrow areas and the avoidance of trapping in the local minima. The disruption factor and mutation operator are described in subsections 3.1 and 3.2, in order. In subsection 3.3, the proposed AOA–HHO hybrid procedure is described.

3.1 Disruption Operator

The disruption operator inspired by an astrophysical theory posits that “when a group of gravitationally bound particles (with a total mass m) is very close to a massive object (with a mass of M), the group becomes torn apart. Similar to this, when a solid body, held together by gravitational forces, approaches a much larger body” [66]. The disruption operator is used to enhance the diversity of the search space where the disruption operator is mathematically modeled as in Eq. (17) [67].

$$D_{OP} = \begin{cases} D_{i,j} \times U(-2, 2) & \text{if } D_{i,best} \geq 1 \\ 1 + D_{i,best} \times U\left(-\frac{10^{-4}}{2}, \frac{10^{-4}}{2}\right) & \text{Otherwise} \end{cases} \quad (17)$$

D_{OP} represents the disruption operator, and $D_{i,j}$ is the distance between the search agent (i) and nearest neighborhood search agent (j). $D_{i,best}$ is the distance between the search agent (i) and the best solution (best). $U(x, y)$ is a number generated randomly within the range (x, y) . For updating the solutions, the disruption operator is used to enhance its diversity as in Eq. (18).

$$X = \begin{cases} X^{past} \times D_{OP} & \text{if } \alpha > 0.5 \\ X^{past} & \text{Otherwise} \end{cases} \quad (18)$$

where X^{past} represents current value of the solution to be updated, X is the value after updating the solution, and α is a random number generated within the range $(0, 1)$. The disable factor has successfully improved many meta-algorithms in many applications such as feature selection [68, 69], optimal flow problem [70], image thresholding [71] and photovoltaic solar cell design [72].

3.2 Mutation Operator

The mutation factor is used to enhance the diversity of the search space and to avoid falling into local lower bounds where two well-known factors are the Cauchy mutation (CM) and the Gaussian mutation (GM). The mutation factor has been used to enhance the diversity of several heuristic algorithms in the literature that has motivated their use [73–78].

CM operator has better ability of efficient search than GM operator according to previous research [73, 77, 79, 80]. Since the CM operator has a wider distribution of searching in the horizontal direction than the vertical direction in contrast to the GM operator, so it is the main motive for the use of the CM operator.

The CM operator's density function is represented as follows as shown in Eq. (19):

$$f_{(0,g)}(\gamma) = \frac{g}{\pi(g + \gamma^2)}, \gamma = \tan(\pi(\text{rand} - 0.5)) \quad (19)$$

where g is the constant parameter with value (1) [77] and rand is a random number within the range $(0, 1)$.

3.3 The Proposed AOA–HHO Procedure

The enhancement of AOA was performed by embedding the CM operator in the exploration updating mechanism of AOA as in Eq. (20).

$$x_i^{(t)} = \begin{cases} \text{best}(x^{(t)}) \div (M_{OP} + \varepsilon) \times CM \times ((UB - LB)\mu + LB) & r_1 < 0.5 \\ \text{best}(x^{(t)}) \times (M_{OP}) \times CM \times ((UB - LB)\mu + LB) & \text{else} \end{cases} \quad (20)$$

where CM is the mutation factor and was estimated based on Eq. (19). Then the inactivation factor was applied by implementing Eq. (18). During the exploit phase, the AOA exploit mechanism (Eq. 4) was replaced by the update mechanism exploited the HHO algorithm (Eq. 8 to Eq. 16). The MOA operator that is estimated according to Eq. (2) balance between exploration and exploitation. The AOA–HHO hybrid procedure is described in Algorithm (3). The flowchart describing the proposed AOA–HHO procedure is shown in Fig. 3.

The main advantages of the proposed AOA–HHO algorithm are inferred in the following:

1. Promoting trapping avoidance in the local minima based on the use of CM operator.
2. Enhancing the diversification of the search space based on the disruption factor.
3. Enhancing the exploitation system based on the mechanism of exploiting of the HHO algorithm.

Algorithm 3: The procedure of AOA-HHO

```

1: Initialize N, T, μ, and ε
2: Generate random solution (xi, i = 1:N) in the range (LB, UB)
3: Estimate the fitness of solutions
4: Update the global best solution best (x)
5: While T isn't reached yet
6:   Update MOA and MOP
7:   If (rand > MOA)
8:     If (r1 < 0.5), update solutions using division operator in Eq. 20
9:     Else, update solutions using multiplication operator in Eq. 20
10:    Apply the disruption operator based on Eq. 18
11:   Else
12:     Asses the escaping energy (E) as in Eq. 6.
13:     If (|E| ≥ 0.5) and (|r| ≥ 0.5) Then
14:       Update the location according to Eq. 8
15:     Else If (|E| ≥ 0.5) and (|r| < 0.5) Then
16:       Update the location according to Eq. 9 to Eq. 12
17:     Else If (|E| < 0.5) and (|r| ≥ 0.5) Then
18:       Update the location according to Eq. 13
19:     Else
20:       Update the location according to Eq. 14 to Eq. 16
21:   End If
22: End If
23: Update the global best solution best (x)
24: End While
25: Output the best solution
    
```

4 PID's Parameter Estimation Based on AOA–HHO

The integral of the absolute error (IAE) function represented in Eq. (1) was used as a fitness function for PID controller's parameters estimation based on the proposed AOA–HHO algorithm shown in Fig. 4. Each AOA–HHO search factor

has a vector of three values (K_p , K_i and K_d), and each solution is initialized with a random value in the lower and upper bounds. The IAE function is estimated based on the sign of e(t) for each research agent to evaluate solutions. The best solution (K_p^g , K_i^g , K_d^g) is determined based on the search agents solution that achieved minimum relevance. The variable MOA controls the implementation of the exploration phase using the AOA or the exploit phase of the HHO algorithm. After completing the execution of iterations, the best solution (K_p^g , K_i^g , K_d^g) has been founded. Algorithm (4) describes the procedure for estimating the parameters of the PID controller using the proposed AOA–HHO algorithm.

Algorithm 4: Parameters Estimation of PID Controller using AOA-HHO

```

1: Initialize N, T, μ and ε
2: Generate real random solution (Xz (Kpz, Kiz, Kdz), z = 1:N), in the range (LB, UB)
3: Estimate the fitness (IAE) using Eq. 1 for each search agent (z = 1:N)
4: Update the global best solution Xg (Kpg, Kig, Kdg) which produces the minimum fitness.
5: While T isn't reached yet
6:   Update MOA and MOP
7:   If (rand > MOA)
8:     If (r1 < 0.5), update solutions using division operator in Eq. 20
9:     Else, update solutions using multiplication operator in Eq. 20
10:    Apply the disruption operator based on Eq. 18
11:   Else
12:     Asses the escaping energy (E) as in Eq. 6.
13:     If (|E| ≥ 0.5) and (|r| ≥ 0.5) Then
14:       Update the location according to Eq. 8
15:     Else If (|E| ≥ 0.5) and (|r| < 0.5) Then
16:       Update the location according to Eq. 9 to Eq. 12
17:     Else If (|E| < 0.5) and (|r| ≥ 0.5) Then
18:       Update the location according to Eq. 13
19:     Else
20:       Update the location according to Eq. 14 to Eq. 16
21:   End If
22: End If
23: Update the global best solution best (x)
24: End While
25: Output the best solution Xg (Kpg, Kig, Kdg)
    
```

5 The Experimental Results and Discussion

Experimental tests were carried out on two systems: the first was DC motor speed control, and it was common in many related studies [18, 27–29, 32, 81]. The second system was controlling the liquid level for three consecutive tank systems [34]. The experimental results were compared with relevant studies such as PSO [6], SCA [5], IWO [29], GWO [32], ASO, PSO-DE [34] and OBL-HGS [27] and Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [82].

The step response characteristics of controlled process response in time domain are delay time, rise time, peak time, settling time and overshooting as shown in Fig. 5 which are defined as follows [1]:

Fig. 3 The procedure for the proposed hybrid AOA–HHO algorithm

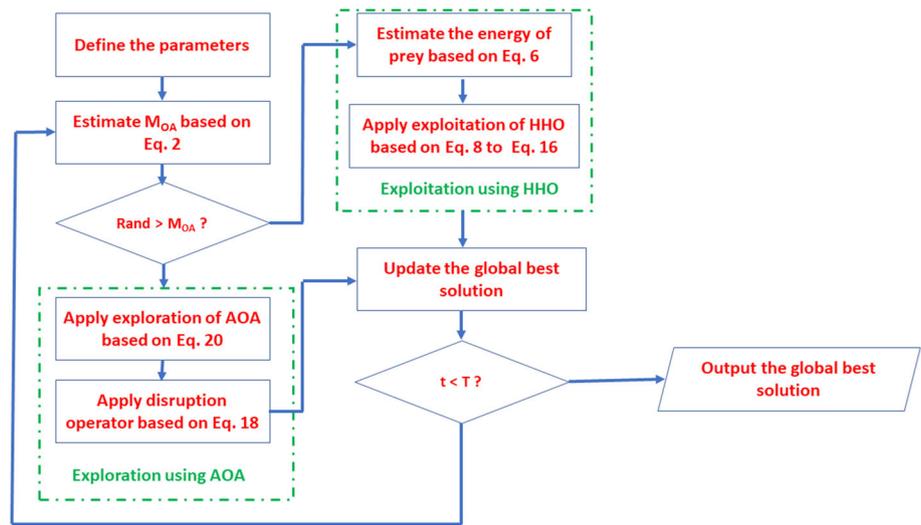


Fig. 4 Estimation of PID parameters based on the proposed AOA–HHO algorithm

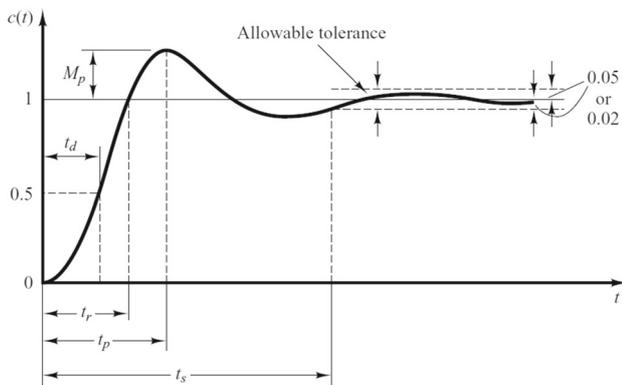
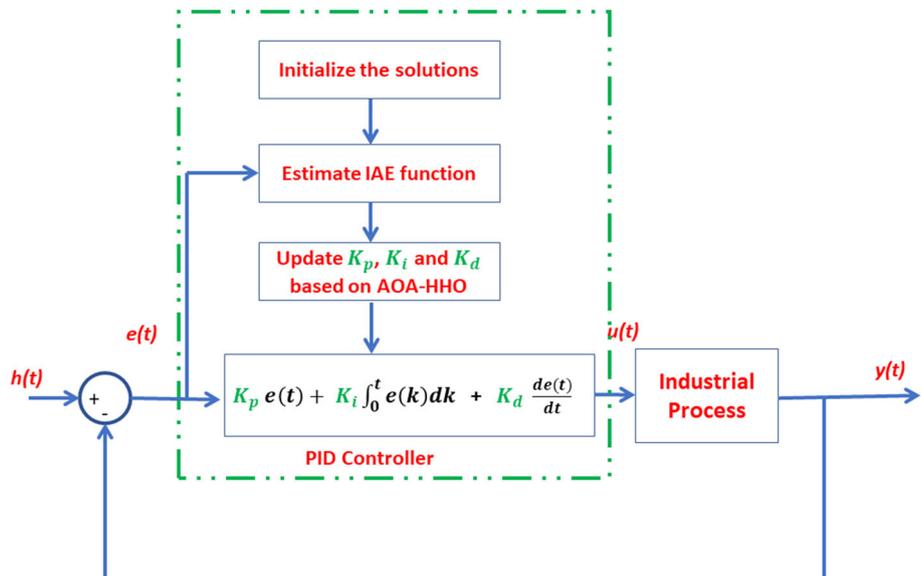


Fig. 5 Time domain specification of controlled process response

1. Delay time (t_d): it is the time required for the response to achieve half of the final value for the first time.

2. Peak time (t_p): it is the time required for the response to reach the first peak of the overshoot.
3. Rise time (t_r): it is the required time for the response to rise from 10% to 90% of its final value.
4. Settling time (t_s): it is the time required for the response curve to reach and stay within a range about the final value of size specified by absolute percentage of the final value (usually 2% or 5%).
5. Overshooting (M_p): it is the maximum peak value of the response curve measured from unity.

The fitness function that was used to evaluate solutions based on IAE according to Eq. (1). The measurement criteria that were used in the comparisons are as follows:

Table 1 The parameters setting of various algorithm for DC motor

	The parameter	Value
All algorithms	The population (N)	100
	Iteration Number (T)	20
	Independent run number	20
	Lower bound of (K_p , K_i and K_d)	[0.001,0.001,0.001]
	Upper bound of (K_p , K_i and K_d)	[20, 20, 20]
PSO	C_1	0.5
	C_2	0.5
	w	0.1
CMA-ES	$C_{\sigma'}$	0.2
	C_{μ}	0.2
	μ_{ω}	0.5
AOA-HHO, AOA	μ	0.5
	ϵ	2

Table 2 Parameters of DC motor [32]

Parameter	Value
R_a	0.4 Ω
L_a	2.7 H
J	0.0004 kg. m ²
D	0.0022 N.m.sec / rad
K	15 e – 03 kg. m / A
K_b	0.05 V.s

1. Integral absolute of difference error between actual and desired responses (IAE)
2. The step response characteristics as settling time, rise time and overshooting.
3. The frequency response of the systems.

5.1 Speed Regulator of DC Motor System

The speed regulation of electrical DC motor [27] was controlled by a PID controller where heuristic algorithms were used to select the best parameters that produce the optimal response. Parameters setting of PSO, CMA-ES, AOA and the proposed AOA-HHO algorithm are listed in Table 1 which were estimated experimentally to get the best results. The results of other algorithms (SCA, IWO, GWO, ASO, and OBL-HG) were obtained from their origin manuscript.

The values of the parameters of DC motor which was used as a case study are listed in Table 2 [32]. R_a represents

armature resistance, L_a represents inductance of armature winding, J represents the equivalent moment of inertia of motor and load referred to motor shaft, D is the equivalent friction coefficient of motor and load referred to motor shaft, K represents Motor torque constant, and K_b represents back EMF constant.

The transfer function of DC motor closed-loop speed control systems is expressed in Eq. (16).

$$G_1(S) = \frac{15}{1.08s^2 + 6.1s + 1.63} \tag{21}$$

Table 3 presents the best PID controller parameter values for optimizing DC motor speed regulation using AOA-HHO versus standard AOA, and other related algorithms were used in the comparative study. AOA-HHO optimizes single objective which is IAE where the solution is the best parameters of PID controller that achieve the minimum IAE. Other specifications such as set time, rise time and overshoot were measured according to the estimated parameters for AOA-HHO and other algorithms in the comparative.

As shown in Table 3, AOA-HHO has the superiority over other algorithms for finding the minimum IAE. It enhances the IAE of AOA from 14.6156 to 9.0465 which proves the enhancement of AOA-HHO using disruption and mutation operator for enhancing exploration and HHO algorithm for enhancing exploitation. PSO and GWO found the most nearest value of IAE to that founded by AOA-HHO, but that of other algorithms is far. AOA-HHO provides IAE better than that of the hybrid techniques such as CMA-ES and OBL-HG which prove the powerful of hybrid technique between AOA and HHO.

For overshoot measurement, AOA-HHO provides overshoot very smaller than that of AOA but not the optimum due to that of GWO and SCA is smaller as shown in Fig. 6. The overshoot of parameters estimated by ASO and OBL-HG is 0 which implies that the system response in this case is over-damping which is the best response shape, but IAE of ASO and OBL-HG is larger than that of AOA-HHO. The reason as mentioned before the single objective was optimize IAE not overshoot.

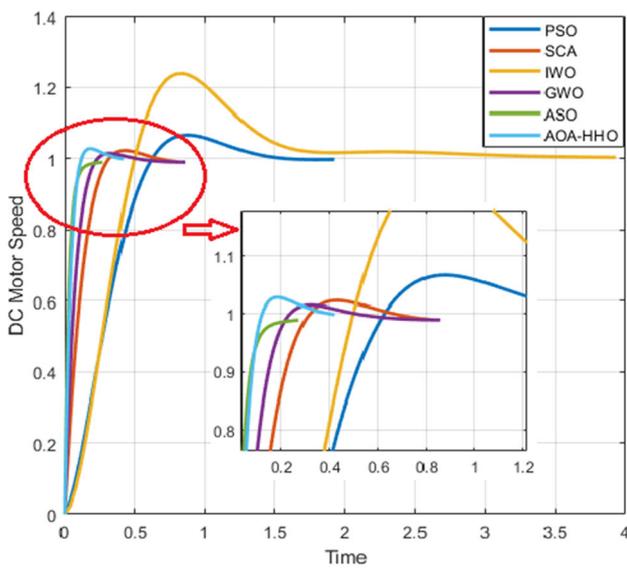
For rise time, AOA-HHO produces the smallest rise time according to results of Table 3, while PSO provides the largest one. That is guaranteed from Fig. 6. For set time, AOA-HHO provides set time smaller than that of AOA, PSO, SCA, IWO and CMA-EA. However, set time of AOA-HHO is not the smallest, but it is reasonable because it implies the time needed for the response to be set around the set point with a percentage 2% or 5%.

This results implies that AOA-HHO has the superiority over than AOA and other algorithms for finding the smallest IAE and reasonable overshoot, set time and rise time.

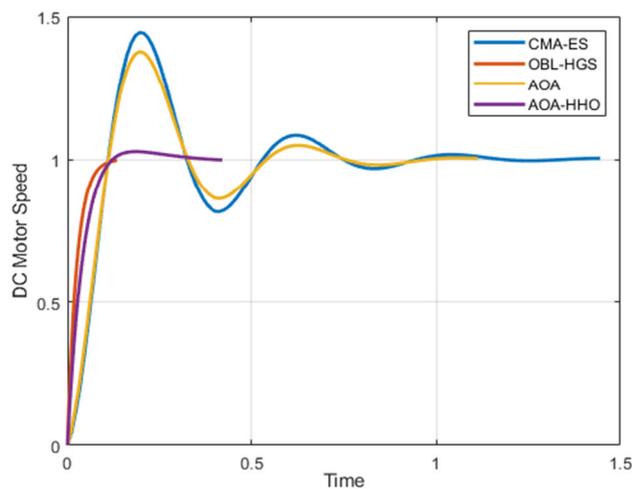
Figure 7 shows the Bode diagrams for regulating a DC

Table 3 Step response and IAE specification for various heuristic algorithms

Method	Kp	Ki	Kd	Set Time (Sec)	Rise Time (Sec)	Over-shoot %	IAE
PSO	1.5234	0.4372	0.0481	0.3549	1.8016	24	12.36
SCA [83]	4.5012	0.5260	0.5302	0.2037	0.4900	2.36	13.63
IWO [29]	1.5782	1.3801	0.0159	0.4190	1.2533	6.7	18.55
GWO [32]	6.898	0.5626	0.9293	0.1388	0.2053	1.5	10.99
ASO [84]	11.943	2.0521	2.4358	0.0692	0.1535	0	22.27
CMA-ES	17.3347	10.9710	0.2140	0.8170	0.0800	44.46	14.73
OBL-HG [27]	16.9327	0.9508	2.8512	0.0546	0.0949	0	21.58
AOA	17.057	4.8488	0.2917	0.7135	0.0821	37.75	14.6156
AOA-HHO	14.435	0.1636	1.7620	0.2508	0.0743	2.83	9.0465



(a)



(b)

Fig. 6 DC motor response versus time in seconds

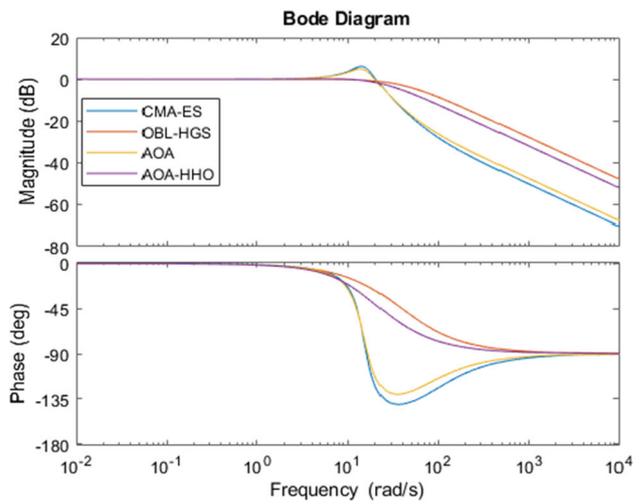
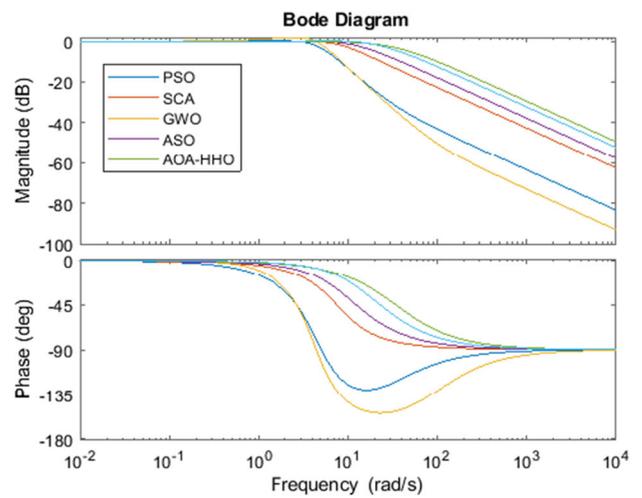


Fig. 7 Bode plots for DC motor based on PID controller

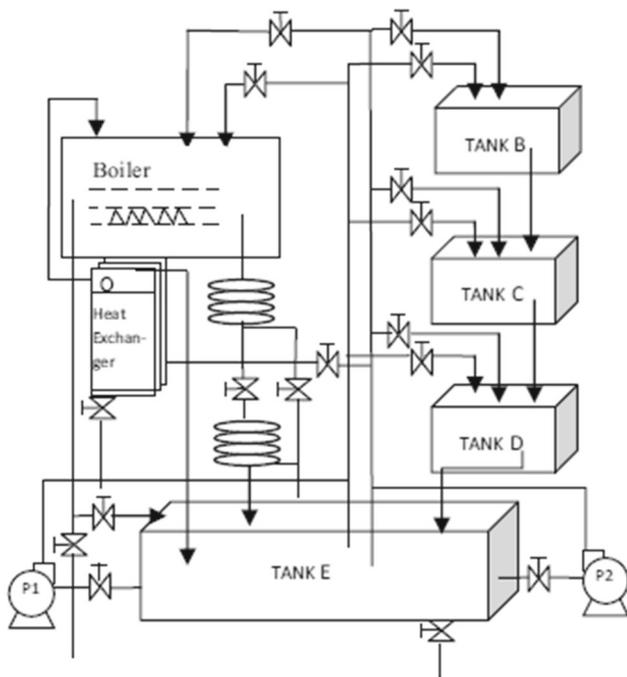


Fig. 8 Three cascaded tanks liquid level systems [34]

motor using a PID controller where its parameters are calculated using the proposed AOA–HHO algorithm and related study algorithms. As shown, AOA–HHO has a wider bandwidth better than that of algorithms used in the comparison except OBL-HGS. This guarantees that AOA–HHO has smaller rise time than other algorithms except OBL-HGS as shown in Table 3 and Fig. 6.

In addition, the magnitude margin of AOA–HHO has not any gain, while that of AOA and CMA-ES has a gain which implies that it produces larger overshoot which is guaranteed as shown in Fig. 6.

5.2 Liquid Level Tank

Three cascaded liquid level tank system was used to test the performance of improved DE with PSO [34] for estimating the parameters of PID controller. As shown in Fig. 8, three tanks B, C and D are cascaded, while tank E is the main tank. Equation 17 expressed the transfer function of the liquid level tank systems [34]. Table 4 presents the best estimated parameters of the PID controller using AOA–HHO and related algorithms that were used in the experimental tests. Besides, the step response characteristics have been added in the table as well as the trap value (IAE).

$$G_2(S) = \left(\frac{1}{4s + 0.2} \right)^3 = \frac{1}{64s^3 + 9.6s^2 + 0.48s + 0.008} \tag{22}$$

Table 4 The parameters setting of various algorithms for liquid level control

	The parameter	Value
All algorithms	The population (N)	100
	Iteration Number (T)	20
	Independent run number	20
	Lower bound of (K _p , K _i and K _d)	[0.001,0.001,0.001]
	Upper bound of (K _p , K _i and K _d)	[20, 20, 20]
PSO	C ₁	0.5
	C ₂	0.5
	w	0.1
SCA	a	3
GWO	\bar{a}_0	2
ASO	α, β	30
	δ	4
	CMA-ES	C _{σ}
	C _{μ}	0.2
	μ_ω	0.5
AOA–HHO, AOA	μ	0.5
	ϵ	2

Parameters setting of PSO, SCA, GWO, ASO, CMA-ES, AOA and the proposed AOA–HHO algorithm are listed in Table 4 which were estimated experimentally to get the best results. The results of PSO-DE were obtained from their origin manuscript.

As shown in Table 5, AOA–HHO has the minimum IAE which is better than that of AOA which has the largest IAE between the comparative algorithms. PSO-DE and GWO produce the most nearest IAE to that of AOA–HHO which is 9.13 and 10.76 in order.

For overshoot, AOA–HHO cannot produce the minimum value, but it is better than that of AOA by difference 50%. SCA produces the minimum overshoot, while it produces high value of IAE. GWO produces the largest overshoot, while it produces reasonable value of IAE. The reason is that the single objective is minimizing IAE not the specification of the response, which can be enhanced in the future work by handling the problem as multi-objectives.

For rise time, AOA–HHO produces reasonable rise time not the minimum but better than that of AOA by difference 65 s. PSO produces the minimum rise time, while AOA produces the largest value of rise time.

For set time, AOA–HHO produced 160.363 s of set time which is larger than that of AOA which has value of 2.65 s.

Table 5 Step response specification and IAE using meta-heuristic algorithms

Method	K_p	K_i	K_d	Set time (sec)	Rise time (Sec)	Over-shoot %	IAE
PSO	0.6060	0.0024	14.4250	80.68	2.4078	67.86	18.61
SCA	0.3039	0.1154	6.7231	207.18	3.5526	4.428	15.66
GWO	0.2928	0.0396	4.719	89.51	4.2374	73.22	10.76
ASO	0.1642	0.0048	12.922	70.1	2.6376	57.26	14.47
CMA-ES	0.051	0.0013	0.3914	238.58	15.0019	50.08	14.27
PSO-DE [34]	0.0419	0.0009	1.000	64.21	12.7790	12.45	09.13
AOA	0.407	0.1184	12.066	2.649	82.756	71.63	16.865
AOA-HHO	0.040	0.0005	0.4269	160.363	17.7783	20.2	8.293

CMA-ES produced the largest value of set time, while AOA produced the minimum value of set time.

Figure 10 shows Bode diagrams for three liquid level tank systems using a PID controller where its parameters are calculated using the proposed AOA-HHO algorithm and related study algorithms. As shown in the figure, the proposed AOA-HHO algorithm has narrower bandwidth than other algorithms; hence, it produces larger rise time as shown in Table 4 and Fig. 9. In addition, according to the magnitude margin of AOA-HHO it has small gain in comparison with that of other algorithms, which implies it has smaller overshoot as shown in Table 4 and Fig. 9.

As shown from the results of the two case studies, AOA-HHO has the superiority over than AOA and other algorithms. The reason is that tuning the parameters of PID controller is a continuous optimization problem which has huge number of allowable values in determined search space. Hence, the balance between exploration and exploitation is an important for achieving the optimal values. In AOA-HHO, Cauchy mutation operator enhances the avoidance of trapping in local optima, while disruption operator enhances the exploration capability of AOA. These two operators enhance the diversification of the AOA-HHO. Regarding the exploitation, HHO was integrated which has an efficient exploitation schemes (Fig. 10).

6 Conclusion

In this work, an enhanced version of AOA is presented to improve the estimation of PID controller parameters. The improvements were made by incorporating the efficient exploitation mechanism of the HHO algorithm instead of exploiting the AOA. In addition, trapping avoidance in the local minima of the proposed AOA-HHO algorithm was enhanced by including disruption and mutation factors which enhance the exploration capability. The AOA-HHO has been tested to select the best parameters of the PID controller for

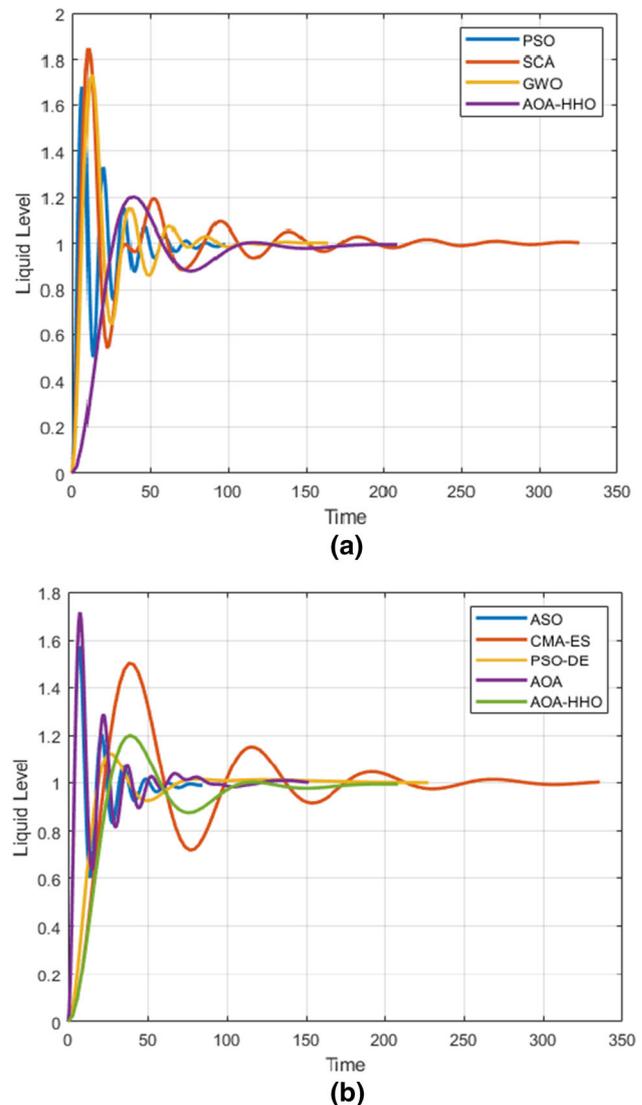


Fig. 9 Liquid level response versus time in seconds

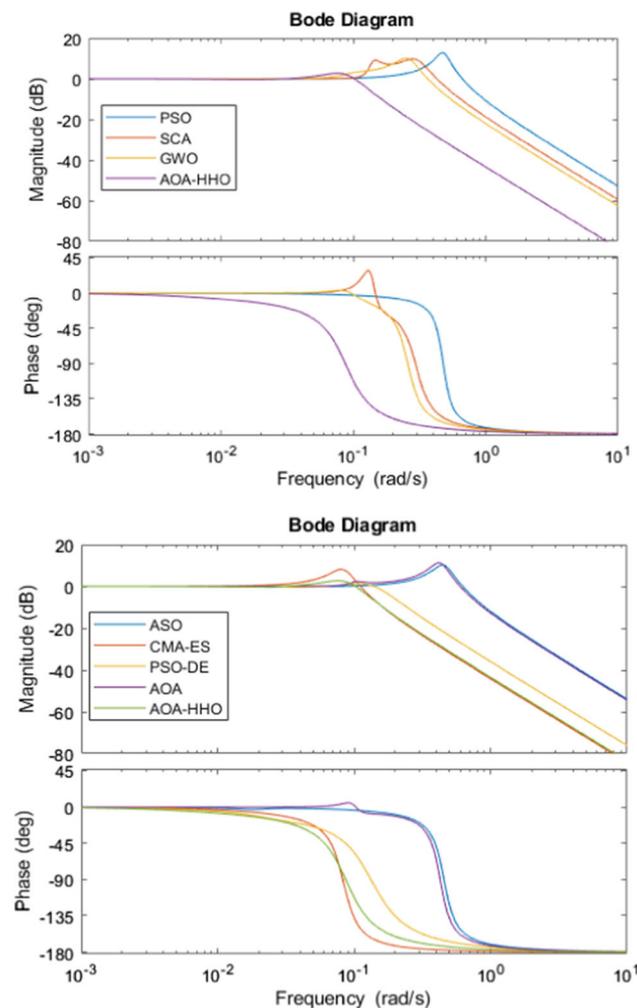


Fig. 10 Bode plots for liquid level tank system based on PID controller

controlling two engineering applications that are DC motor regulation and three cascading liquid level tank systems. The single objective function was an integral part of the absolute error (IAE) function.

AOA-HHO has outperformed AOA in terms of IAE and response specification such as overshoot, rise time and set time in controlling dc motor, while for controlling level of three cascaded liquid tanks only set time of AOA is better than that of AOA-HHO. The frequency response of AOA-HHO was measured which implies it produces reasonable bandwidth and gain magnitude margin better than that of AOA and other comparative algorithms. From the experimental study, AOA-HHO has the superiority over AOA and other comparative study for estimating efficient parameters of PID controlling which leads to efficient IAE.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the

source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Ogata, K.; Yang, Y.: Modern control engineering. Vol. 4. Prentice hall India, Prentice (2002)
- Mansour, T.: PID Control: Implementation and Tuning. BoD—Books on Demand (2011)
- Ziegler, J.G.; Nichols, N.B.: Optimum settings for automatic controllers. Trans. ASME **64**(11) (1942)
- Talbi E.-G.: Metaheuristics: From Design to Implementation. vol. 74. John Wiley & Sons, London (2009)
- Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. Knowl.Based Syst. **96**, 120–133 (2016)
- Kennedy: Particle swarm optimization. Neural Netw (1995)
- Javidy, B.; Hatamlou, A.; Mirjalili, S.: Ions motion algorithm for solving optimization problems. Appl. Soft Comput. **32**, 72–79 (2015)
- Nematollahi, A.F.; Rahiminejad, A.; Vahidi, B.: A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization. Appl. Soft Comput. **59**, 596–621 (2017)
- Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl.Based Syst. **89**, 228–249 (2015)
- Issa, M.; Abd Elaziz, M.: Analyzing COVID-19 virus based on enhanced fragmented biological Local Aligner using improved ions motion optimization algorithm. Appl Soft Comput, p. 106683 (2020)
- Issa, M. et al.: ASCA-PSO: Adaptive sine cosine optimization algorithm integrated with particle swarm for pairwise local sequence alignment. Expert Syst. Appl. **99**, 56–70 (2018)
- Issa, M.: Expeditious Covid-19 similarity measure tool based on consolidated SCA algorithm with mutation and opposition operators. Appl. Soft Comput. **104**, 107197 (2021)
- Issa, M.: Sequence analysis algorithms for bioinformatics application. GRIN Verlag (2014)
- Issa, M.; Hassanien, A.E.: Multiple sequence alignment optimization using meta-heuristic techniques. In: Handbook of Research on Machine Learning Innovations and Trends. IGI Global, p. 409–423 (2017)
- Issa, M. et al. Pairwise Global Sequence Alignment Using Sine-Cosine Optimization Algorithm. In: International Conference on Advanced Machine Learning Technologies and Applications. Springer (2018)
- Issa, M.; Helmi, A.: Two layer hybrid scheme of IMO and PSO for optimization of local aligner: COVID-19 as a case study. In: Artificial Intelligence for COVID-19, pp. 363–381. Springer (2021)
- Issa, M. et al.: A biological sub-sequences detection using integrated BA-PSO based on infection propagation mechanism: Case study COVID-19. Expert Syst. Appl. **189**, 116063 (2022)
- Bora, T.C.; L.d.S. Coelho, and L. Lebensztajn, *Bat-inspired optimization approach for the brushless DC wheel motor problem*. IEEE Transactions on magnetics, 2012. **48**(2): p. 947–950.

19. Issa, M. et al.: PID controller tuning parameters using meta-heuristics algorithms: comparative analysis. In: *Machine Learning Paradigms: Theory and Application*, pp. 413–430. Springer (2019)
20. Jordehi, A.R.: Parameter estimation of solar photovoltaic (PV) cells: A review. *Renew. Sustain. Energy Rev.* **61**, 354–371 (2016)
21. Pan, J.-S. et al.: A Multi-objective ions motion optimization for robot path planning. In: *International conference on engineering research and applications*. Springer (2018)
22. Issa, M.; Samn, A.: Passive vehicle suspension system optimization using Harris Hawk Optimization algorithm. *Math. Comput. Simul.* **191**, 328–345 (2022)
23. Gaing, Z.-L.: A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Trans. Energy Convers.* **19**(2), 384–391 (2004)
24. Issa, M.: Performance optimization of pid controller based on parameters estimation using meta-heuristic techniques: a comparative study. In: *Metaheuristics in Machine Learning: Theory and Applications*, pp. 691–709. Springer (2021)
25. Roy, A.; Srivastava, S.: Design of optimal PID controller for speed control of DC motor using constrained particle swarm optimization. In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT). IEEE (2016)
26. Khubalkar, S. et al.: Modeling and control of a permanent-magnet brushless DC motor drive using a fractional order proportional-integral-derivative controller. *Turk. J. Electr. Eng. Comput. Sci.* **25**(5), 4223–4241 (2017)
27. Ekinci, S.; Hekimoğlu, B.; Izci, D.: Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor. *Eng Sci Technol Int J* **24**(2), 331–342 (2021)
28. Razmjoo, N. et al.: Speed control of a DC motor using pid controller based on improved whale optimization algorithm. In: *Metaheuristics and Optimization in Computer and Electrical Engineering*, pp. 153–167. Springer (2021)
29. Khalilpour, M. et al.: Optimal control of DC motor using invasive weed optimization (IWO) algorithm. In: *Majlesi Conference on Electrical Engineering*, Majlesi New Town, Isfahan, Iran (2011)
30. Potnuru, D.; Mary, K.A.; Babu, C.S.: Experimental implementation of flower pollination algorithm for speed controller of a BLDC motor. *Ain Shams Eng. J.* **10**(2), 287–295 (2019)
31. Bendjehaba, O.: Continuous firefly algorithm for optimal tuning of PID controller in AVR system. *J. Electr. Eng.* **65**(1), 44 (2014)
32. Agarwal, J. et al.: Analysis of grey wolf optimizer based fractional order PID controller in speed control of DC motor. *Microsyst. Technol.* **24**(12), 4997–5006 (2018)
33. Chatterjee, S.; Mukherjee, V.: PID controller for automatic voltage regulator using teaching–learning based optimization technique. *Int. J. Electr. Power Energy Syst.* **77**, 418–429 (2016)
34. Moharam, A.; El-Hosseini, M.A.; Ali, H.A.: Design of optimal PID controller using hybrid differential evolution and particle swarm optimization with an aging leader and challengers. *Appl. Soft Comput.* **38**, 727–737 (2016)
35. Wolpert, D.H.; Macready, W.G.: No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**(1), 67–82 (1997)
36. Abualigah, L. et al.: The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **376**, 113609 (2021)
37. Premkumar, M. et al.: A new arithmetic optimization algorithm for solving real-world multiobjective CEC-2021 constrained optimization problems: diversity analysis and validations. *IEEE Access* **9**, 84263–84295 (2021)
38. Abualigah, L. et al.: A novel evolutionary arithmetic optimization algorithm for multilevel thresholding segmentation of covid-19 ct images. *Processes* **9**(7), 1155 (2021)
39. Khodadadi, N.; Snaes, V.; Mirjalili, S.: Dynamic arithmetic optimization algorithm for truss optimization under natural frequency constraints. *IEEE Access* **10**, 16188–16208 (2022)
40. Zellagui, M. et al.: Arithmetic optimization algorithm for optimal installation of DSTATCOM into distribution system based on various voltage stability indices. In: 2021 9th International Conference on Modern Power Systems (MPS). IEEE (2021)
41. Abd Elaziz, M. et al.: IoT workflow scheduling using intelligent arithmetic optimization algorithm in fog computing. *Comput. Intel. Neurosci.* (2021)
42. Kharrich, M. et al.: An improved arithmetic optimization algorithm for design of a microgrid with energy storage system: case study of El Kharga Oasis, Egypt. *J. Energy Storage* **51**, 104343 (2022)
43. Hao, W.-K. et al.: Arithmetic optimization algorithm based on elementary function disturbance for solving economic load dispatch problem in power system. *Appl. Intell.*; p. 1–27 (2022)
44. Garzon Rivera, O.D. et al.: Application of the Arithmetic Optimization Algorithm to Solve the Optimal Power Flow Problem in Direct Current Networks. Available at SSRN, 4069702.
45. Malibari, A.A. et al.: Arithmetic optimization with retinanet model for motor imagery classification on brain computer interface. *J. Healthcare Eng.* (2022)
46. Abbassi, A. et al.: Improved arithmetic optimization algorithm for parameters extraction of photovoltaic solar cell single-diode model. *Arab. J. Sci. Eng.*; p. 1–17 (2022)
47. Ahmadi, B. et al.: The Arithmetic optimization algorithm for optimal energy resource planning. In 2021 56th International Universities Power Engineering Conference (UPEC). IEEE (2021)
48. Mahboob, A.S.; Moghaddam, M.R.O.; Yousefi, S.: AOV-IDS: arithmetic optimizer with voting classifier for intrusion detection system. In: 2021 12th International Conference on Information and Knowledge Technology (IKT). IEEE (2021)
49. Sharma, A. et al.: A Novel opposition-based arithmetic optimization algorithm for parameter extraction of PEM fuel cell. *Electronics* **10**(22), 2834 (2021)
50. Zheng, R. et al.: Deep ensemble of slime mold algorithm and arithmetic optimization algorithm for global optimization. *Processes* **9**(10), 1774 (2021)
51. Ewees, A.A. et al.: Boosting arithmetic optimization algorithm with genetic algorithm operators for feature selection: case study on Cox proportional hazards model. *Mathematics* **9**(18), 2321 (2021)
52. Agushaka, J.O.; Ezugwu, A.E.: Advanced arithmetic optimization algorithm for solving mechanical engineering design problems. *PLoS ONE* **16**(8), e0255703 (2021)
53. Mahajan, S. et al.: Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks. *Soft Comput.* p. 1–19 (2022)
54. Aydemir, S.B.: A novel arithmetic optimization algorithm based on chaotic maps for global optimization. *Evol. Intell.* p. 1–16 (2022)
55. Do, D.T. et al.: A hybrid arithmetic optimization algorithm and differential evolution for optimization of truss structures subjected to frequency constraints. *J. Sci. Technol. Civil Eng. (STCE)-HUCE* (2022)
56. Heidari, A.A. et al.: Harris hawks optimization: algorithm and applications. *Futur. Gener. Comput. Syst.* **97**, 849–872 (2019)
57. Zhang, Y. et al.: Boosted binary Harris hawks optimizer and feature selection. *Eng. Comput.* **37**(4), 3741–3770 (2021)
58. Chen, H. et al.: Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts. *J. Clean. Prod.* **244**, 118778 (2020)
59. Bao, X.; Jia, H.; Lang, C.: A novel hybrid harris hawks optimization for color image multilevel thresholding segmentation. *IEEE Access* **7**, 76529–76546 (2019)
60. Houssein, E.H. et al.: A novel hybrid Harris hawks optimization and support vector machines for drug design and discovery. *Comput. Chem. Eng.* **133**, 106656 (2020)
61. Bui, D.T. et al.: A novel swarm intelligence—Harris hawks optimization for spatial assessment of landslide susceptibility. *Sensors* **19**(16), 3590 (2019)

62. Liu, Y. et al.: Horizontal and vertical crossover of Harris hawk optimizer with Nelder-Mead simplex for parameter estimation of photovoltaic models. *Energy Convers. Manage.* **223**, 113211 (2020)
63. Ramachandran, M. et al.: A hybrid grasshopper optimization algorithm and harris hawks optimizer for combined heat and power economic dispatch problem. *Eng. Appl. Artif. Intell.* **111**, 104753 (2022)
64. Singh, N. et al.: HSSAHHO: a novel hybrid Salp swarm-Harris hawks optimization algorithm for complex engineering problems. *J Ambient Intell Humanized Comput*, p. 1–37 (2022)
65. Zhong, C.; Li, G.: Comprehensive learning Harris hawks-equilibrium optimization with terminal replacement mechanism for constrained optimization problems. *Expert Syst. Appl.* **192**, 116432 (2022)
66. Harwit, M.: *Astrophysical Concepts*. Springer Science & Business Media, Berlin (2006)
67. Liu, H.; Ding, G.; Wang, B.: Bare-bones particle swarm optimization with disruption operator. *Appl. Math. Comput.* **238**, 106–122 (2014)
68. Hamidzadeh, J.: Feature selection by using chaotic cuckoo optimization algorithm with levy flight, opposition-based learning and disruption operator. *Soft. Comput.* **25**(4), 2911–2933 (2021)
69. Neggaz, N. et al.: Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst. Appl.* **145**, 113103 (2020)
70. Bansal, J.C.; Farswan, P.: A novel disruption in biogeography-based optimization with application to optimal power flow problem. *Appl. Intell.* **46**(3), 590–615 (2017)
71. Abd Elaziz, M. et al.: Improving image thresholding by the type II fuzzy entropy and a hybrid optimization algorithm. *Soft. Comput.* **24**(19), 14885–14905 (2020)
72. Fathy, A. et al.: Optimal parameter identification of triple-junction photovoltaic panel based on enhanced moth search algorithm. *Energy* **188**, 116025 (2019)
73. Salgotra, R.; Singh, U.: Application of mutation operators to flower pollination algorithm. *Exp. Syst. Appl.* **79**, 112–129 (2017)
74. Zhang, Q. et al.: Chaos-induced and mutation-driven schemes boosting salp chains-inspired optimizers. *IEEE Access* **7**, 31243–31261 (2019)
75. Jia, H. et al.: Dynamic harris hawks optimization with mutation mechanism for satellite image segmentation. *Remote sensing* **11**(12), 1421 (2019)
76. Xu, Y. et al.: Enhanced Moth-flame optimizer with mutation strategy for global optimization. *Inf. Sci.* **492**, 181–203 (2019)
77. Wang, H. et al.: Opposition-based particle swarm algorithm with Cauchy mutation. In: 2007 IEEE Congress on Evolutionary Computation. IEEE (2007)
78. Zhang, X. et al.: Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Syst. Appl.* **141**, 112976 (2020)
79. Wang, G.-G. et al.: Opposition-based krill herd algorithm with Cauchy mutation and position clamping. *Neurocomputing* **177**, 147–157 (2016)
80. Sapre, S.; Mini, S.: Opposition-based moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling for global optimization. *Soft. Comput.* **23**(15), 6023–6041 (2019)
81. Ekinici, S. et al.: Speed control of DC motor using improved sine cosine algorithm based PID controller. In: 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE (2019)
82. Hansen, N.; Auger, A.: CMA-ES: evolution strategies and covariance matrix adaptation. In: Proceedings of the 13th annual conference companion on Genetic and evolutionary computation (2011)s
83. Agarwal, J.; Parmar, G.; Gupta, R.: Application of sine cosine algorithm in optimal control of DC motor and robustness analysis. *Wulfenia J* **24**(11), 77–95 (2017)
84. Hekimoğlu, B.: Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. *IEEE Access* **7**, 38100–38114 (2019)

