

Research Article

PID Controller Parameter Optimized by Reformative Artificial Bee Colony Algorithm

Hualong Du, Pengfei Liu, Qiuyu Cui, Xin Ma, and He Wang 

School of Mechanical Engineering and Automation, University of Science and Technology Liaoning, Anshan 114051, China

Correspondence should be addressed to He Wang; wanghe@ustl.edu.cn

Received 4 January 2022; Accepted 19 January 2022; Published 17 February 2022

Academic Editor: Sun Young Cho

Copyright © 2022 Hualong Du et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The PID parameters determine the PID controller performance. A reformative artificial bee colony (RABC) algorithm is proposed for the PID parameter optimization problem. The algorithm balances the exploitation capability and exploration capability of the ABC algorithm by introducing a global optimal solution and improving the food source probability. The proposed algorithm is validated by simulation with six benchmark functions, and the results show that the RABC algorithm has higher search accuracy and faster search speed than other variants of the artificial bee colony algorithm. The RABC algorithm-optimized PID controller has better control with minimum overshoot and fast response, as verified by comparison with PSO-PID, DE-PID, and GA-PID methods in three typical systems.

1. Introduction

The PID controller has been the most widely used and mature controller in the industrial production process [1]. Despite the emergence of various new controllers in the control field, PID controllers are still in the dominant position with their simple structure, easy implementation, and robustness. The PID controllers are widely used in chemical, electric power, metallurgy, and other industrial control sites. In the industrial control process, more than 95% of the loops have a PID structure [2]. The suitability of controller parameters has an important impact on the quality of the controller. With the development of modern industry, the traditional PID controller can no longer meet the requirements of control systems with high order, time lag, and nonlinearity. The traditional method of PID parameter optimization can no longer fully adapt to the exploitation of modern industry, so it is very important to study a new and efficient PID parameter optimization technique for engineering practice [3].

For the optimization of the parameters of the PID controller, the researchers used various optimization techniques. Feng et al. [4] proposed an improved genetic algorithm (IGA), to search for the PID controller parameters, for the robotic excavator. Özdemir et al. [5]

proposed a new metaheuristic optimization algorithm, optical inspired optimization (OIO) algorithm to optimize PID controllers. The method has better performance in terms of maximum overshoot and stabilization time. Chen et al. [6] proposed a fuzzy PID controller optimized by an improved ant colony algorithm. The improvements of nonlinear incremental evaporation rate and pheromone incremental update were proposed in the IACO algorithm to improve the quality of the solution. Hekimo glu et al. [7] proposed atomic search optimization (ASO) algorithm and chaotic ASO (ChASO) to determine the optimal parameters of fractional-order proportional + integral + derivative (FOPID) controller. Bingul et al. [8] proposed a new time-domain performance criterion for the rectification design of the proportional-integral-derivative (PID) controller in an automatic voltage regulator (AVR) using the cuckoo search algorithm. This performance criterion is chosen to minimize the maximum overshoot, rise time, stabilization time, and steady-state error of the terminal voltage. Ekinici et al. [9] used an improved kidney-inspired algorithm (IKA) and a new objective function. The main objective of the method is to optimize the transient response of the AVR system to obtain the optimal values of the three gains (K_p , K_i , and K_d) of the PID controller by minimizing the maximum overshoot, stabilization time, rise time, and peak

time values of the terminal voltage and eliminating the steady-state error. Batiha et al. [10] implemented two optimization algorithms, particle swarm optimization (PSO) and bacteria foraging optimization (BFO) algorithms, for the purpose of tuning the fractional-order PID-controller. Huang and Chuang [11] proposed an artificial bee colony optimization (ABC) algorithm incorporating fuzzy theory to optimize the PID controller by introducing fractional-order proportional-integral-derivative (FOPID) control strategy. Panoeiro et al. [12] optimized PID controller parameters by bionic optimization technique. Optimization of PID parameters based on the Ziegler-Nichols (ZN) rule was proposed in [13] as a fractional-order PID controller optimization method based on radial basis function (RBF) neural network. A fractional-order fuzzy proportional integral differential (FOFPID) controller was proposed by Sharma et al. [14]. The cuckoo search algorithm (CSA) optimization technique was used to optimize all the controller parameters. Chang et al. proposed a new adaptive genetic algorithm for PID controller design, and they found that the fractional-order PID controller significantly reduced the overshoot and stabilization time compared to the optimized conventional PID controller [15]. Bingul [16] used a differential evolution (DE) algorithm for tuning the PID controller for unstable and time-lagged integral processes. The results show that the PID-tuned DE has faster stabilization time, less or no overshoot, and higher robustness. Cao and Cao [17] demonstrated parameter optimization of a fractional-order controller based on improved PSO. The improved particle swarm algorithm has a faster search speed and better solution than the genetic algorithm. Maiti et al. [18] used PSO to design fractional-order PID controllers. They significantly reduced the overshoot percentage, rise, and adjustment time using the FOPID controller compared to the PID controller. Alfi and Modares [19] used a novel adaptive PSO (APSO) algorithm to find the optimal system parameters for unstable nonlinear systems and optimal parameters for PID controllers. Some scholars applied the particle swarm algorithm (PSO) [20–22] to optimize the PID controller parameters to improve the search speed. Improving the PID structure is too tedious. The exploitation capability and exploration capability of intelligent algorithms in parameter optimization of PID controllers need further improvement.

Artificial bee colony algorithm is a swarm intelligence optimization algorithm, proposed by Karaboga, to simulate the process of honey bee foraging [23]. The algorithm is easy to implement control with few parameters and has good optimization performance. Therefore, the contributions of this paper are as follows:

- (1) The RABC algorithm is proposed to improve the exploitation capability and exploration capability of the ABC algorithm.
- (2) The PID controller is based on the RABC algorithm (RABC-PID). The effectiveness of this controller is verified by comparing it with the controllers optimized by the other three methods.

The rest of this paper is organized as follows. Section 2 introduces the basic artificial swarm algorithm, the modified artificial swarm algorithm, and the superiority of the modified artificial swarm algorithm verified by six benchmark functions. The principle of optimized PID controller parameters and the whole optimization process of the modified artificial swarm algorithm is introduced. Section 3 establishes three typical models, and Section 4 conducts experimental simulations to validate the superiority of the RABC algorithm by comparing it with three other intelligent methods. Section 5 concludes the whole paper.

2. Proposed Method

2.1. Basic Artificial Bee Colony Algorithm. The basic artificial bee colony algorithm divides the population into three types: employed bees, following bees, and scout bees, and sets a search phase for each type of artificial bee, i.e., employed bee phase, following bee phase, and scout bee phase. Initially, the algorithm uses random initialization to generate the initial population, which is shown in the following:

$$X_i^j = X_{\min}^j + \text{rand}(0, 1)(X_{\max}^j - X_{\min}^j), \quad (1)$$

where $i = 1, \dots, \text{SN}$, $j = 1, \dots, D$. SN denotes the population size, D denotes the problem dimension, $\text{rand}(0, 1)$ is a random number between 0 and 1, and X_{\max}^j and X_{\min}^j denote the upper and lower bounds of the j th dimension of an individual, respectively.

In the employed bee phase, the employed bee searches for food sources by performing a random search of the feasible domain by equation (2) and passes the food source multiplication to the following bee waiting in the hive to search for the food source.

$$V_i^j = X_i^j + \varphi_i^j(X_i^j - X_k^j), \quad (2)$$

where $k \in (1, \dots, \text{SN})$ is a randomly selected indicator for different i , which means that there is only one randomly selected solution in generating the new candidate solution; $j \in (1, \dots, D)$ is a randomly selected indicator, which means that only one dimension has changed between the new candidate solution and the old one. φ_i^j is a random number uniformly distributed on $[-1, 1]$.

In the following bee stage, according to the food source information passed back to the hive by the employed bee, the following bee selects the food source using roulette according to the probability calculated in equation (3) below, and the nectar collection process still uses equation (2) to update the food source randomly.

$$P_i = \frac{\text{fit}_i}{\sum_{j=1}^{\text{SN}} \text{fit}_j}, \quad (3)$$

where fit_i denotes the adaptation value of the i th food source.

During the scout bee phase, scout bees discard all food sources that have already been mined for honey, exceeding the limit for all food sources. An employed bee on the abandoned food source will then transform into a scout bee,

randomly searching for new food sources according to equation (1). Throughout the search process of the algorithm, the food source corresponds to the candidate solution of the optimization problem, and the quality of the food source represents the merit of the candidate solution.

2.2. Reformative Artificial Bee Colony Algorithm. It is well known that the exploration ability [24] and the exploitation ability of an algorithm are two conflicting aspects that affect the performance of an algorithm. In other words, the enhanced exploration ability of the algorithm will inevitably affect the exploitation ability of the algorithm, which may reduce the convergence speed of the algorithm, while the enhanced exploitation ability of the algorithm will also inevitably affect the exploration ability of the algorithm, which may lead the algorithm to fall into local optimum. The artificial bee colony algorithm can be seen to have stronger exploration ability and weaker exploitation ability due to its random search feature. In this paper, a reformative artificial bee colony algorithm (RABC) is proposed. The optimal solution idea is proposed for the problem of the weak exploitation ability of the artificial bee colony algorithm. The algorithm selects the global optimal bee location and its food source location when updating the location of the food source. When the colony is updated, the bees are allowed to refer to the global bee with the best food source and move to the food source with better quality, and the global optimal is updated as the colony is constantly updated, which allows the bees to use the optimal food source as a reference when acquiring food source information and improves the exploration capability of the algorithm to some extent. Equation (2) for searching food sources is changed to the following equation:

$$V_i^j = X_{i,\text{best}}^j + \varphi_i^j (X_{i,\text{best}}^j - X_i^j), \quad (4)$$

where $X_{i,\text{best}}^j$ is the global current optimal food source.

The following probability of following bees is determined by the proportion of the current food source's fitness among all food sources. When some better food sources do not differ much from the optimal food source, it will lead to a lower probability of following the optimal food source by the following bees. For some bad food sources, due to the random selectivity of the following bees, they have the opportunity to follow them instead, which leads to a slower speed of finding the optimal food source and shows a slow convergence speed in the algorithm. Because of this, this paper proposes to take the current optimal food source of the colony as a reference, so that the following bees are more inclined to choose the food source with high quality and improve the speed of the colony to find the optimal food source. Thus, the probability P_i of following bees following the hiring bees is changed to the following equation:

$$P_i = \frac{0.8\text{fit}_i}{\max \text{fit}_i + 0.2}, \quad (5)$$

where $\max \text{fit}_i$ is the fitness value of the highest solution.

Both employed and following bees use equation (4) to update the food source location, which allows a portion of the bee's information to be exchanged with the globally optimal bee, ensuring that the bee is not disturbed by the locally optimal bee, but also that the bee moves to a better food source led by the globally optimal bee. Using equation (5) to judge the quality of food sources, following bees are more likely to follow honey-harvesting bees that have high-quality food sources and thus exploit high-quality food sources.

2.3. Benchmark Function Simulation Verification. To check the optimization performance of the RABC algorithm, the same ABC, GABC [25], and GBABC [26] algorithms are tested for comparison experiments on six benchmark functions. The basic characteristics of the test functions are given in Table 1. The test dimension of the test function is $D=50$. In the experiments, the population number SN is 100, the limit is 50, the maximum cycle number MaxCycle is set to 5000, and the algorithm is run 30 times independently. The experimental results are shown in Table 2.

Table 2 gives the experimental results of the six benchmark functions, including the mean and standard deviation. From the table, we can see that the RABC algorithm has the smallest standard deviation among the six benchmark functions, which proves that the RABC algorithm has the best stability. The mean value is closest to the optimal value, which proves that the RABC algorithm has the highest search accuracy.

Figure 1 gives the benchmark function convergence curve graph. From the figure, it can be seen that the RABC algorithm outperforms both the basic artificial bee colony algorithm and the other two improved artificial bee colony algorithms in terms of convergence speed performance and search accuracy. The experimental results indicate that RABC has better optimization performance.

2.4. PID Controller Based on RABC (RABC-PID). The RABC algorithm is used to optimize the PID parameters, which is essentially a parameter optimization problem based on a certain objective function, i.e., finding the optimal values in the parameter space of K_p , K_i , and K_d variables to optimize the control performance of the system. The control block diagram is shown in Figure 2.

The RABC algorithm optimizes the PID parameters by taking the error on the system as the evaluation function of the RABC, i.e., the fitness function input, calculating the value of the fitness function, and then adjusting the three PID parameters according to the fitness of the function to make the control performance of the system optimal.

The selection of the objective function is an important process. To achieve the optimal comprehensive performance of the whole system, we need to use some indicators that can reflect the comprehensive performance. At this stage, the common comprehensive performance evaluation criteria are mainly based on the connection between the deviation of the system $e(t) = r(t) - y(t)$ and time t . There are four main comprehensive performance evaluation criteria: integral of squared error (ISE), integral of time-weighted squared error

TABLE 1: Benchmark function definition domain and optimal value.

Function	Name	Definition domain	Optimal value
F1	Beale	(-4.5, 4.5)	0
F2	Camel3	(-5, 5)	0
F3	Hump	(-5, 5)	0
F4	Rastrigin	(-5.12, 5.12)	0
F5	Goldstein	(-2.2)	3
F6	Easom	(-100, 100)	-1

TABLE 2: Benchmark function experimental results.

Function		ABC	GABC	GBABC	RABC
F1	Mean	0.00218509	0.000172577	0.000144142	6.49866e-05
	Std	0.00129199	0.000147622	9.31112e-05	2.46584e-05
F2	Mean	0.00338069	0.0024648	5.02474e-05	9.78549e-06
	Std	0.00235156	0.0025263	1.80147e-05	1.96744e-06
F3	Mean	0.00130278	0.00134831	5.77273e-05	1.17531e-05
	Std	0.00040916	0.00025792	6.32839e-05	4.90377e-06
F4	Mean	0.0287649	0.0123041	4.1950e-10	1.02564e-11
	Std	0.0374303	0.0081333	4.3844e-10	8.64572e-11
F5	Mean	3.2728	3.03062	3.0111	3.00295
	Std	0.337852	0.00717374	0.00522465	0.00379628
F6	Mean	-0.0548194	-0.351209	-0.41055	-0.771196
	Std	0.3775246	0.352833	0.0580604	0.00293816

(ITSE), integral of absolute error (IAE), integrated time absolute error (ITAE) [27].

Integrated time absolute error (ITAE), which integrates the speed, stability, and accuracy of the system, is widely used to optimize the comprehensive performance index of the PID controller. ITAE can be expressed as

$$\text{ITAE} = \int_0^{\infty} t|e(t)|dt. \quad (6)$$

To prevent the control from being too large, the weighted integrated time absolute error (WITAE) is proposed. The ITAE performance index is used as the minimum adaptation function for parameter selection, and the squared term of the control input is added to the objective function. WITAE can be expressed as

$$\text{WITAE} = \int_0^t [J_1 t|e(t)| + J_2 t u(t)^2] dt, \quad (7)$$

where J_1, J_2 are the weights and WITAE is the adaptation value. Once the overshoot is generated and the penalty function is applied [28], equation (8) is used as the performance evaluation criteria.

$$\text{WITAE} = \int_0^t [J_1 t|e(t)| + J_2 t u(t)^2 + J_3 |e(t)|] dt, \quad (8)$$

where J_3 is the weight value, $J_3 \gg J_1$. Usually, $J_1 = 0.999$, $J_2 = 0.001$, and $J_3 = 100$. The fitness function is expressed as

$$\text{fit}_i = \begin{cases} \frac{1}{1 + \text{WITAE}}, & \text{WITAE} \geq 0, \\ 1 + |\text{WITAE}|, & \text{WITAE} < 0. \end{cases} \quad (9)$$

The higher the value of fit_i is, the higher the probability that the food source will be selected for search. The RABC

algorithm optimizes the PID controller parameters flow as shown in Figure 3.

3. Three Typical Models

3.1. DC Motor Modeling. The motor circuit structure is shown in Figure 4. The air-gap flux is constant, and the control of the motor is realized by adjusting the voltage of the armature circuit, to realize the regulation of the motor speed. For such a motor object, some of its parameter characteristics such as torque constant and viscous friction coefficient are not available, so it can be regarded as a black box in modeling, and the modeling can be completed by using input and output data and using system identification.

First, determine the object model structure, and the air-gap flux is known, so its electromagnetic torque T_i is proportional to the armature current i_a :

$$T_i = K_i i_a, \quad (10)$$

where K_i is the torque constant of the motor.

The armature circuit voltage balance equation can be expressed as

$$u_a = R_a i_a + L_a \frac{di_a}{dt} + e_a, \quad (11)$$

where R_a and L_a are the resistance and inductance of the armature circuit, respectively.

The relationship between the motor's counter-electromotive force e_a and its angular velocity ω is as follows:

$$e_a = K_e \omega, \quad (12)$$

where K_e is the counter-electromotive force constant. The equation of torque balance on the motor shaft can be expressed as

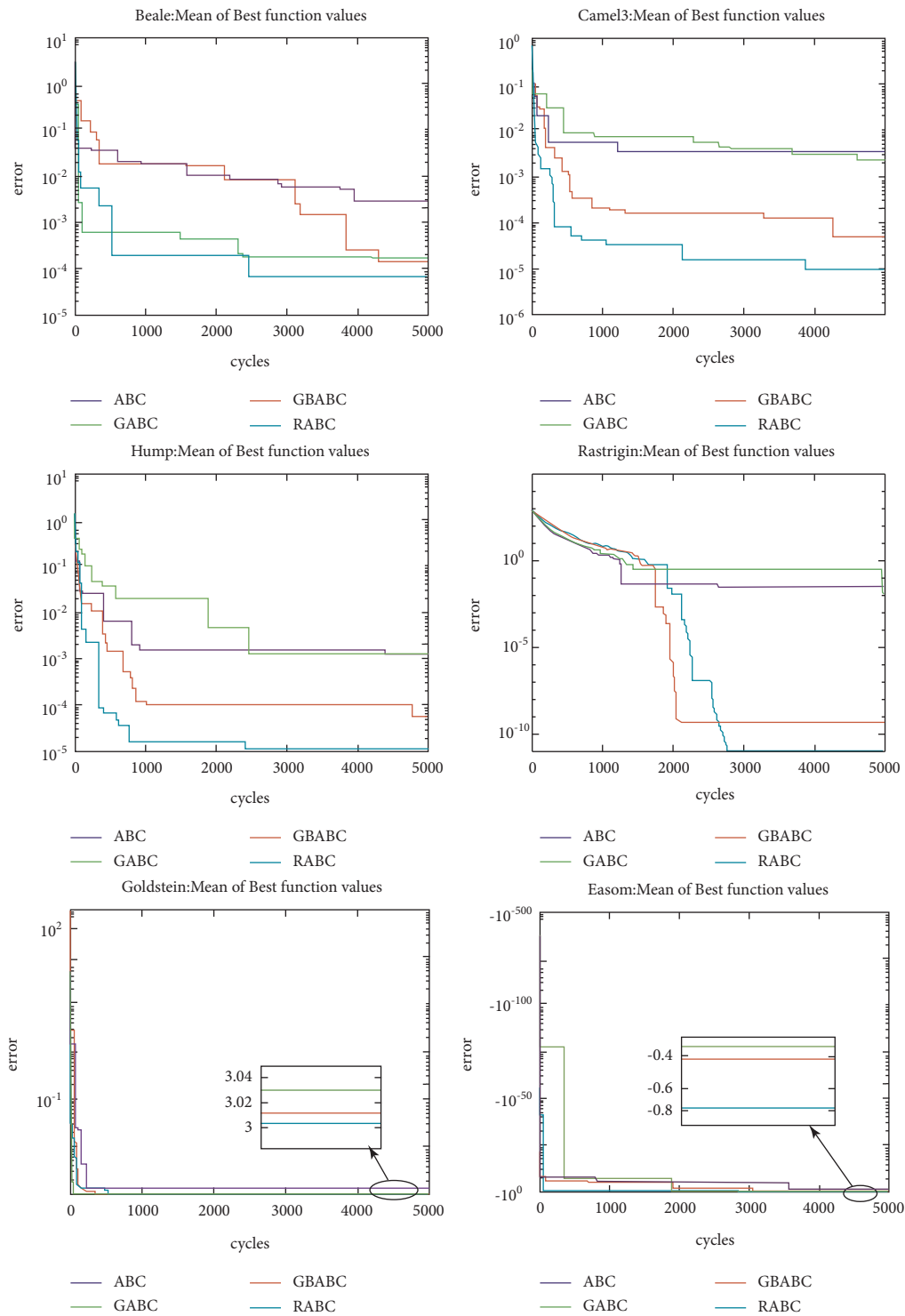


FIGURE 1: Convergence curve of benchmark function.

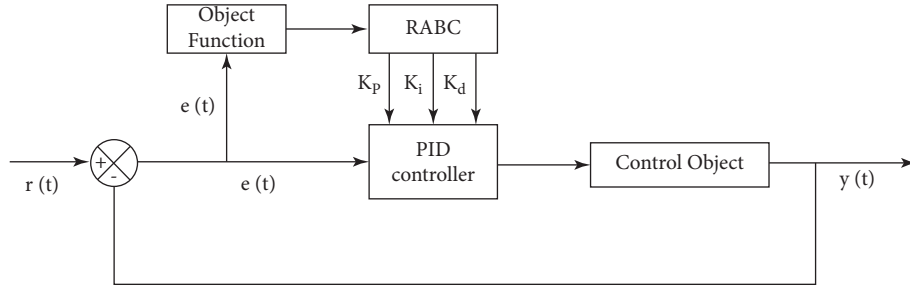


FIGURE 2: PID parameter tuning block diagram of reformative artificial bee colony algorithm.

$$T_i = J \frac{d\omega}{dt} + b\omega + T_L, \quad (13)$$

where J is the equivalent rotational inertia, b is the equivalent viscous friction coefficient, and T_L is the load torque. With u_a as input voltage and ω as output speed, the following differential equation is obtained by eliminating i_a , e_a , and T_i :

$$L_a J \frac{d^2\omega}{dt^2} + (R_a J + bL_a) \frac{d\omega}{dt} + (K_e K_t + R_a b)\omega + \frac{R_a T_L + L_a T}{K_t} = K_t u_a. \quad (14)$$

The term of $((R_a T_L + L_a T)/K_t)$ can be ignored without participating in the calculation. Therefore, differential equation (14) can be transformed into the following transfer function (15) according to the Laplace transformation.

$$G(s) = \frac{K_t}{L_a J s^2 + (R_a J + bL_a)s + (K_e K_t + R_a b)}. \quad (15)$$

At this point, the armature-controlled DC motor can be viewed as an oscillating link. Usually, the inductance L_a in the armature circuit is small, and if its effect is neglected, transfer function (15) can be approximated as a first-order transfer function, which can be expressed as follows:

$$G(s) = \frac{K}{Ts + 1}, \quad (16)$$

where $K = (K_t / (K_e K_t + R_a b))$ is the gain constant of the motor and $T = (R_a J / (K_e K_t + R_a b))$ is the time constant of the motor.

3.2. Mathematical Model of Double Capacity Water Tank (DCWT). The double-volume tank is a typical second-order time-lag system, which is schematically shown in Figure 5.

In Figure 5, A_1 and A_2 represent the bottom area of the tank, q_1 , q_2 , and q_3 represent the water flow, R_1 and R_2 represent the resistance of the valves V_1 and V_2 , which is called liquid resistance of valve resistance, and $\Delta q = (\Delta h/R)$. According to the material balance on tank 1, there is the following equation:

$$\Delta q_1 - \Delta q_2 = A_1 \frac{d\Delta h_1}{dt}, \quad (17)$$

$$\Delta q_2 = \frac{\Delta h_1}{R_2}. \quad (18)$$

Therefore, differential equations (17) and (18) can be transformed into the following transfer functions (19) and (20) according to the Laplace transformation.

$$\Delta Q_1(S) - \Delta Q_2(S) = A_1 S \Delta H_1(S), \quad (19)$$

$$\Delta Q_2(S) = \frac{\Delta H_1(S)}{R_2}. \quad (20)$$

Similarly, the differential equation and transfer function of tank 2 can be obtained:

$$\Delta q_2 - \Delta q_3 = A_2 \frac{d\Delta h_2}{dt},$$

$$\Delta q_2 = \frac{\Delta h_2}{R_3}, \quad (21)$$

$$\Delta Q_2(S) - \Delta Q_3(S) = A_2 S \Delta H_2(S),$$

$$\Delta Q_3(S) = \frac{\Delta H_2(S)}{R_2}.$$

The transfer function of DCWT can be expressed as follows:

$$\begin{aligned} W_o(S) &= \frac{\Delta H_2(S)}{\Delta Q_1(S)} = \frac{R_2}{(A_1 R_2 S + 1)(A_2 R_3 S + 1)} \\ &= \frac{K}{(T_1 S + 1)(T_2 S + 1)}, \end{aligned} \quad (22)$$

where $T_1 = A_1 R_2$ is the time constant of tank 1, $T_2 = A_2 R_3$ is the time constant of tank 2, and K is the amplification factor of the dual-capacity object. If the system also has a pure delay, transfer function (22) can be changed as follows:

$$W_o(S) = \frac{\Delta H_2(S)}{\Delta Q_1(S)} = \frac{K}{(T_1 S + 1)(T_2 S + 1)} e^{-\tau s}. \quad (23)$$

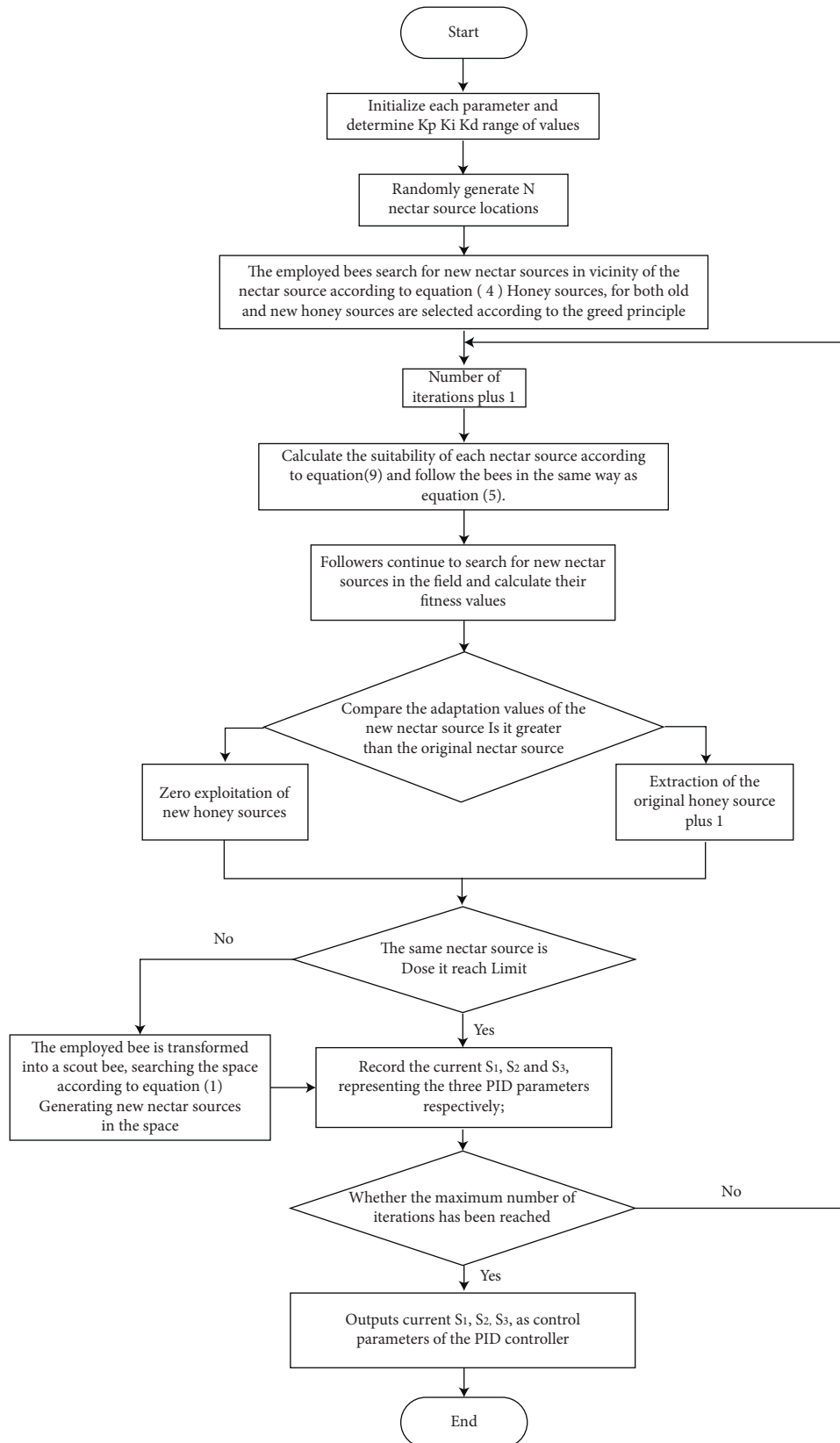


FIGURE 3: The flowchart of the reformative artificial bee colony algorithm for optimizing PID controller parameters.

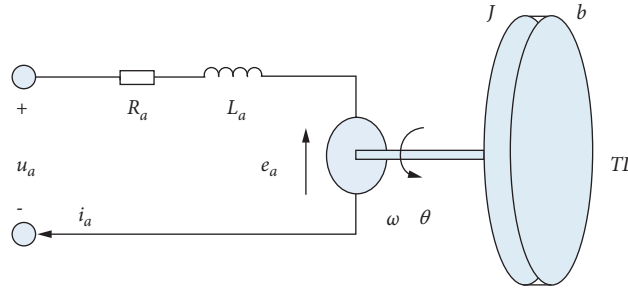


FIGURE 4: Motor circuit structure diagram.

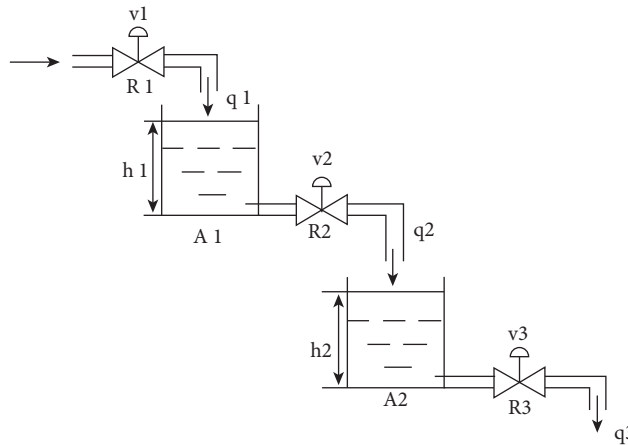


FIGURE 5: Schematic diagram of dual-capacity water tank.

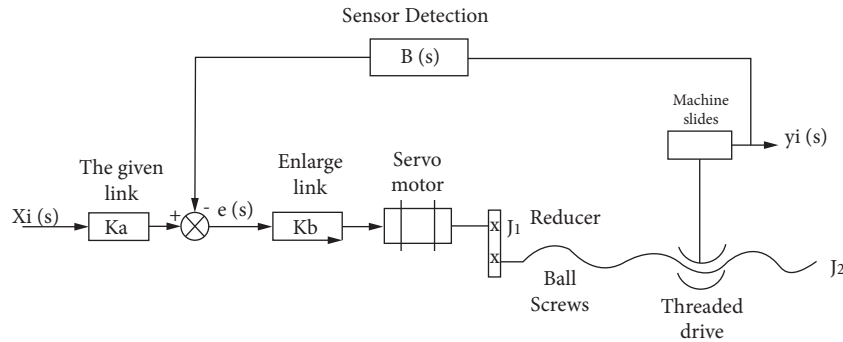


FIGURE 6: Schematic diagram of the position control system of the machine tool sliding table.

3.3. Position Control System for the Sliding Table (PCSST).

The schematic diagram of the PCSST is shown in Figure 6. In the figure, $x_i(s)$ is the given input signal; K_a is the given link; $y_i(s)$ is the output signal; and $B(s)$ is the sensor detection link. $e(s)$ is the error signal, which is converted to a voltage signal to drive the servo motor after the amplification link K_b . The modeling process of the PCSST is explained as follows.

3.3.1. Position Closed-Loop Model. The schematic diagram of the electro-hydraulic servo slide table position control system in this study is shown in Figure 6. There are

photoelectric sensors installed on both sides of the x -axis direction of the machine slide, which detect the slide motion position in real time and use the position closed-loop control to continuously correct the position error $e(s)$ to achieve zero-error position closed-loop control. Therefore, the closed-loop position control model is established as follows:

$$U_a(s) = K_b [p_{wc}(s) - p_{wf}(s)], \tag{24}$$

$$p_{wf}(s) = K_w x_h(s),$$

where $U_a(s)$ is the position loop voltage; K_b is the position amplification factor; $p_{wc}(s)$ represent the position loop

TABLE 3: Transfer function.

Model	Coefficient setting	Transfer function
DC	$K=5$ $T=6$	$G(s) = (5/(6s+1))$
DCWT	$\tau=5$ $K=10$ $T_1=3$ $T_2=5$	$G(S) = (10/(3s+1)(5s+1))e^{-5\tau}$
PCSST	$a_3=5, a_2=3$ $a_1=2.45, a_0=18$ $K_a K_b K_w K_T i_{pl} = 16$	$G(s) = (16/(5s^3+3s^2+2.45s+18))$

TABLE 4: The optimal parameters of the PID controllers of the three models.

		K_p	K_i	K_d
RABC	DC	15.8565	3.08603	-0.874396
	DCWT	8.456	5.61285	0.573094
	PCSST	5.9645	3.72379	0.86088
PSO	DC	12.5	2.9477	-1.0005
	DCWT	6.872	3.063	0.13548
	PCSST	5.2364	2.0749	0.18095
DE	DC	14.56	3.04633	-0.93457
	DCWT	5.465	2.871	0.15037
	PCSST	5.22386	1.959	0.18219
GA	DC	13.5	1.462	-1.02
	DCWT	4.2827	2.84296	0.13236
	PCSST	4.0379	1.6632	0.14135

initial pulses; $p_{wf}(s)$ represent the position loop sensor feedback pulses; K_w is the position gain factor; and $x_h(s)$ is the slide table movement.

3.3.2. *Servomotor Model.* The servomotor drive model can be expressed as follows:

$$\begin{bmatrix} \dot{i} \\ \ddot{\theta} \end{bmatrix} = \begin{pmatrix} \frac{r}{l} & \frac{p_k \lambda_f}{l} \\ \frac{3p_n \lambda_f}{2J_1} & 0 \end{pmatrix} \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{u}{l} \\ -\frac{T}{l} \end{bmatrix}, \quad (25)$$

where J_1 is the servomotor inertia; λ_f is the motor magnetic field coefficient; l is the inductance coefficient; p_k is the pole logarithm; i is the current value; θ is the rotation angle; and T is the servomotor torque, and it can be expressed as follows:

$$T = K_T i = \frac{3}{2} p_n \lambda_f i, \quad (26)$$

where K_T is the torque coefficient.

3.3.3. *Working Slide Drive Model.* The machine slide completes the movement along the X -axis direction under the ball screw thread drive, and the slide position movement model can be simplified to a ball screw linear motion model as follows:

$$x_l(s) = i_{pt} \theta_l = \frac{H_h i}{2\pi} \theta_m, \quad (27)$$

where $x_i(s)$ is the displacement of the table; i is the ratio of the ball screw; H_h is the total ball screw travel; and θ_m is the turning angle of the screw.

3.3.4. *Closed-Loop Total Transfer Model of Table Position.* From the above three models, closed-loop series connection to form the total transfer function of the electro-hydraulic position closed-loop control system of the working slide of the combined machine tool can be obtained.

$$G(s) = \frac{x(s)}{y(s)} = \frac{K_a K_b K_w K_T i_{pl}}{a_3 s^3 + a_2 s^2 + a_1 s^1 + a_0}. \quad (28)$$

4. Simulation Validation

The RABC balances the exploitation capability of the basic artificial bee colony algorithm with the exploration capability. In this paper, we verify the superiority of the RABC algorithm to optimize PID parameters by four intelligent algorithms.

4.1. *Performance Indicators of the Control System.* The performance indicators of control systems are divided into transient performance indicators and steady-state performance indicators. Transient performance refers to the transient behavior of the output of a control system during

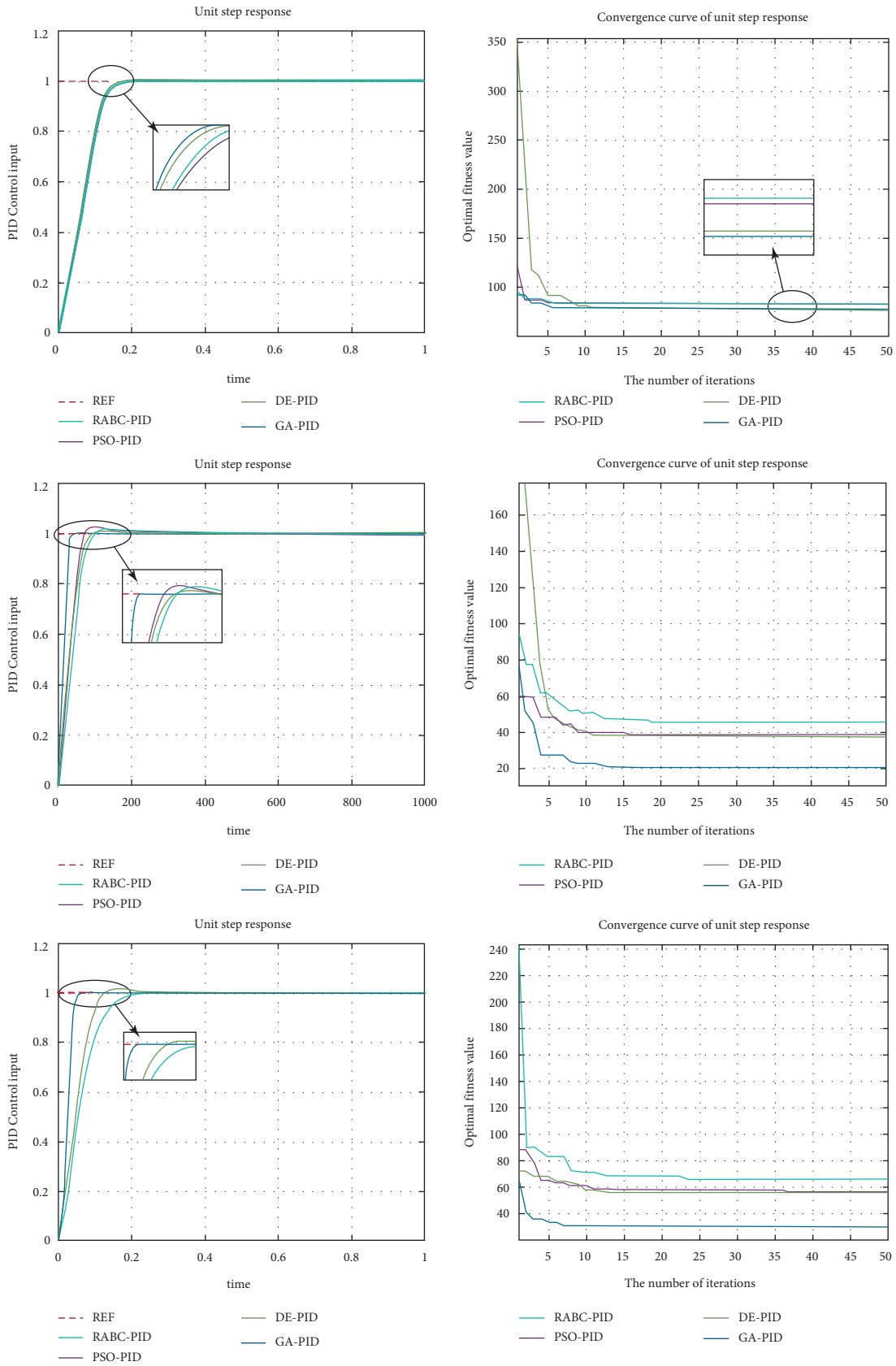


FIGURE 7: Unit step response curve and convergence curve.

TABLE 5: Performance index.

		t_r	t_p	σ %	t_s	WITAE
RABC	DC	0.114	0.176	0.038882	0.134	77.28
	DCWT	28	10	0	32	20.48
	PCSST	0.043	0.074	0.089273	0.052	30.28
PSO	DC	0.124	0.241	0.049552	0.157	82.04
	DCWT	60	100	3.1078	123	38.64
	PCSST	0.095	0.103	1.9173	0.117	55.95
DE	DC	0.116	0.183	0.075293	0.138	78.05
	DCWT	65	115	1.1421	82	37.38
	PCSST	0.096	0.164	1.7873	0.118	56.65
GA	DC	0.127	0.293	0.082143	0.166	82.86
	DCWT	68	116	1.5401	84	45.93
	PCSST	0.126	0.305	0.15743	0.177	66.34

the transition process, which is the so-called transition process in which the system is transferred from one steady state to another under the action of an external input signal. In control systems, the unit step response of the system is generally used to define the indicators of the transient performance of the system, which are usually rise time t_r , peak time t_p , peak time t_p , and overshoot σ %.

4.2. Simulation Verification. After simulation verification in Matlab, three typical mathematical model transfer functions are shown in Table 3.

To verify the optimization performance of the RABC algorithm, the comparison with three intelligent algorithms PSO, DE, and GA is verified. After the simulation verification in Matlab, the optimal parameter setting of each algorithm is obtained. The parameters of each algorithm were set as follows: PSO algorithm $C_1 = C_2 = 2$, initial value of W is 0.9, $W_{\max} = 0.9$, $W_{\min} = 0.4$, $V_{\max} = 1$, $V_{\min} = -1$, population size $SN = 50$, and iterations = 50. DE algorithm crossover probability $P_{cr} = 0.8$, scaling factor $F = 0.85$, population size $SN = 50$, and iterations = 50. GA algorithm crossover probability $P_c = 0.7$, variation probability $P_m = 0.3$, population size $SN = 50$, iterations = 50. RABC algorithm population size SN is set to 50, limit number of iterations is 50, and iteration number is set to 50.

The simulation time for DC and PCSST is set to 1 s. The simulation time for DCWT is set to 1000 s, since DCWT is a time-delay system. The experimental results are shown in Table 4.

From the unit step response curve in Figure 7, it can be seen that the RABC algorithm has no overshoot amount in the dual-capacity tank system, and the rise time, regulation time, and peak time are the shortest. The RABC algorithm has almost no overshoot in the other two systems, and the rise time, regulation time, and peak time are also the shortest. This indicates that the RABC algorithm optimizes

the PID controller parameters best. From the unit step response convergence curve in Figure 7, it can be seen that the RABC algorithm has the fastest convergence speed and the highest convergence accuracy among the three systems. This also proves that the RABC algorithm optimizes the PID controller parameters best.

From Table 5, it can be clearly seen that the adaptation values obtained by the RABC algorithm in optimizing the PID controller parameters are smaller than those of the other three optimization algorithms for the DC motor, dual-capacity water tank, and machine tool control systems. This also fully illustrates that the RABC algorithm is more accurate than the other three intelligent algorithms. From Table 5, it can be also seen that the PID controller of the dual-capacity water tank system optimized by the RABC algorithm has a fast response, no overshoot, and short regulation time and can enter the steady-state zone quickly, which reflects a better control effect. The PID controller of the other two systems optimized by the RABC algorithm has almost no overshoot, and the rise time, regulation time, and peak time are also the shortest. In summary, RABC algorithm-optimized PID controller has a better control effect.

A good controller has some robustness to cope with changes in parameters. For DC model, the parameter T is set to 10, 15, and 20, respectively. For DCWT model, the parameter T_1 is set to 5, 10, and 15, respectively. For PCSST model, the parameter a_3 is set to 15, 20, and 25, respectively. The simulation results are presented in Figures 8–10 and Tables 6–8. It can be seen from Figures 8–10 that the method still yields better results in the case of variation of model parameters. From Tables 6–8, we can see that the RABC-PID method still outperforms the other three methods in terms of overshoot, rise time, regulation time, and peak time when the parameters are changed through simulation experiments, which also proves that the RABC-PID has stronger robustness.

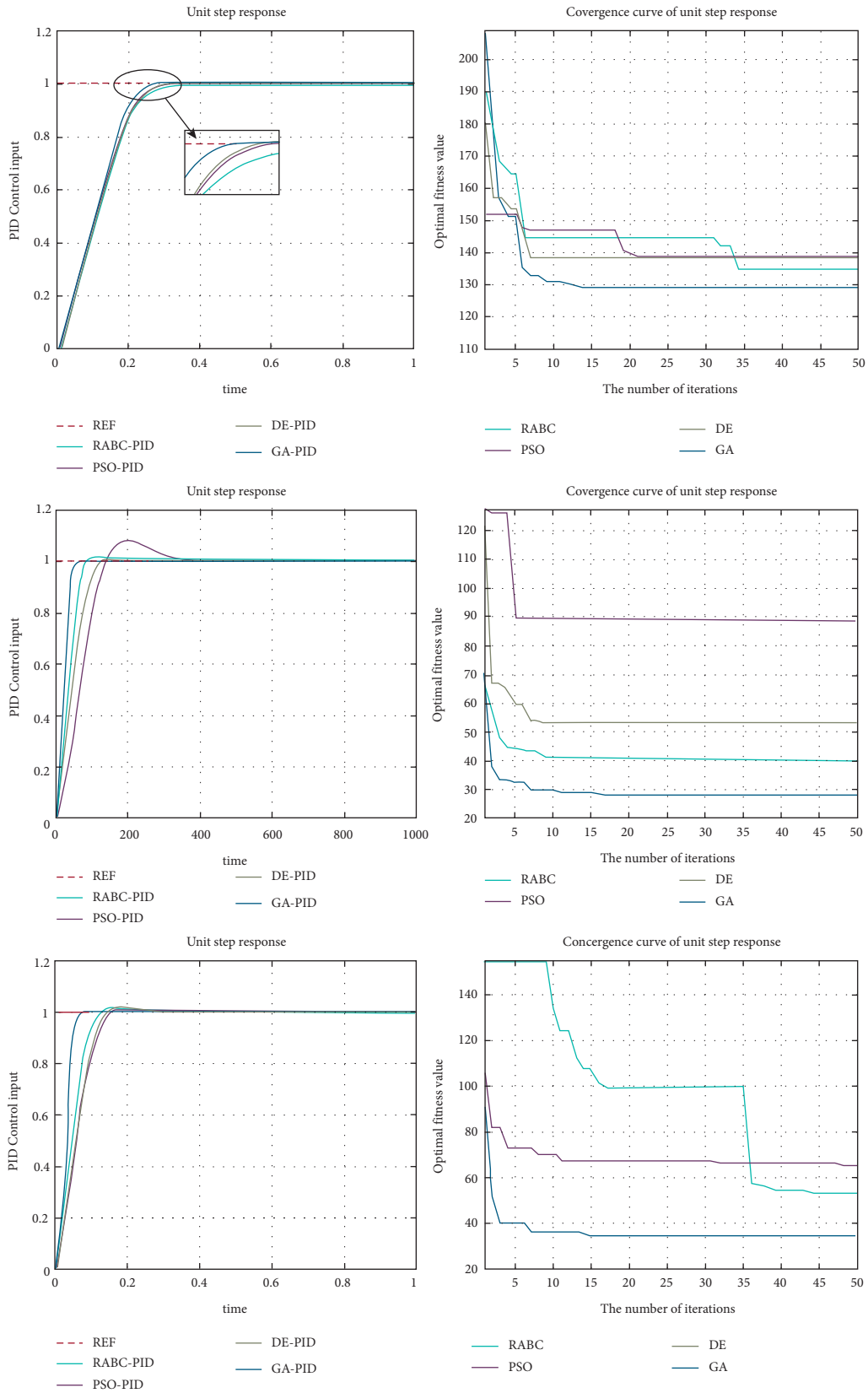


FIGURE 8: Unit step response curve and convergence curve when the parameters $T=10$, $T_1=5$, and $a_3=15$.

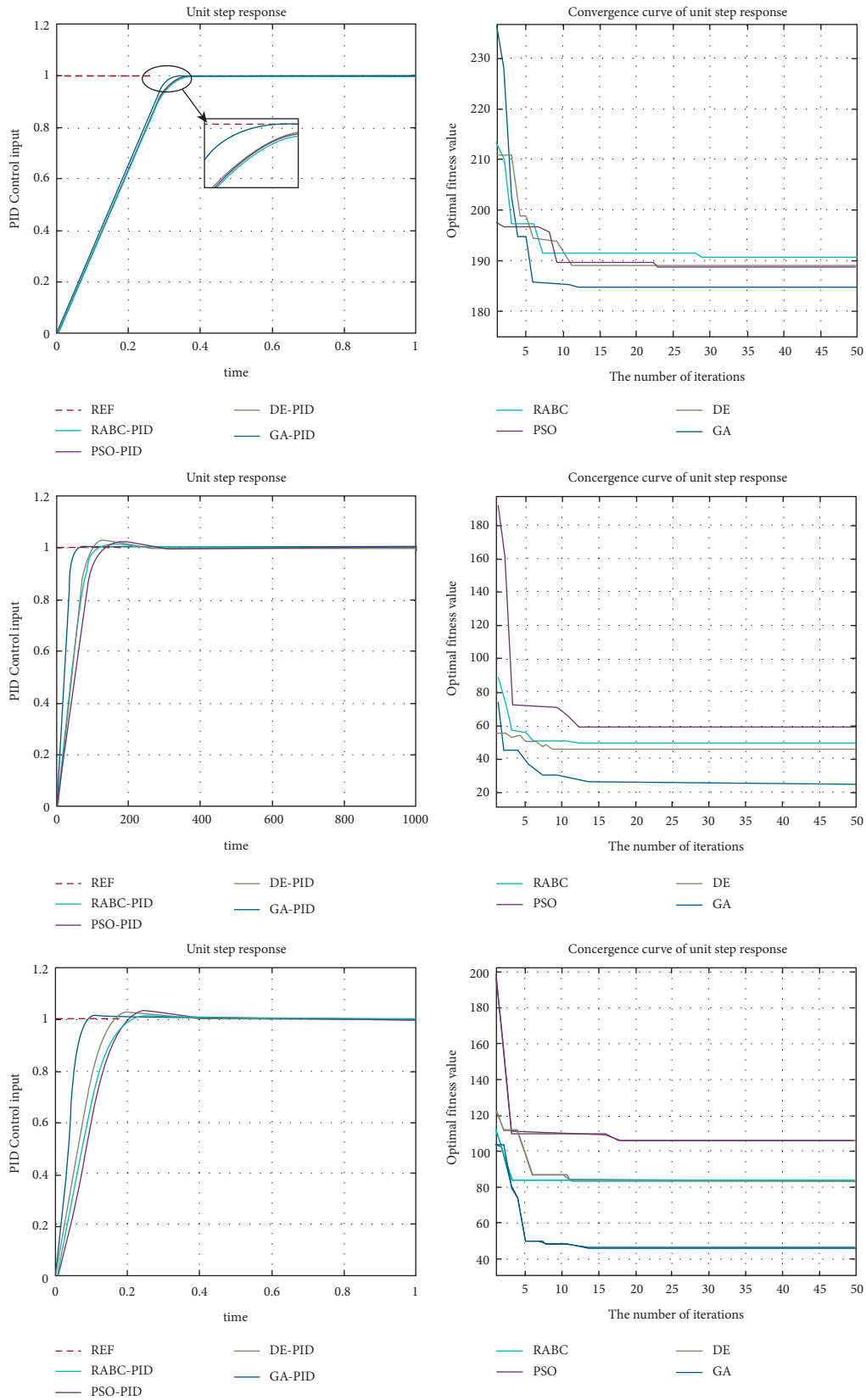


FIGURE 9: Unit step response curve and convergence curve when the parameters $T=15$, $T_1=10$, and $a_3=20$.

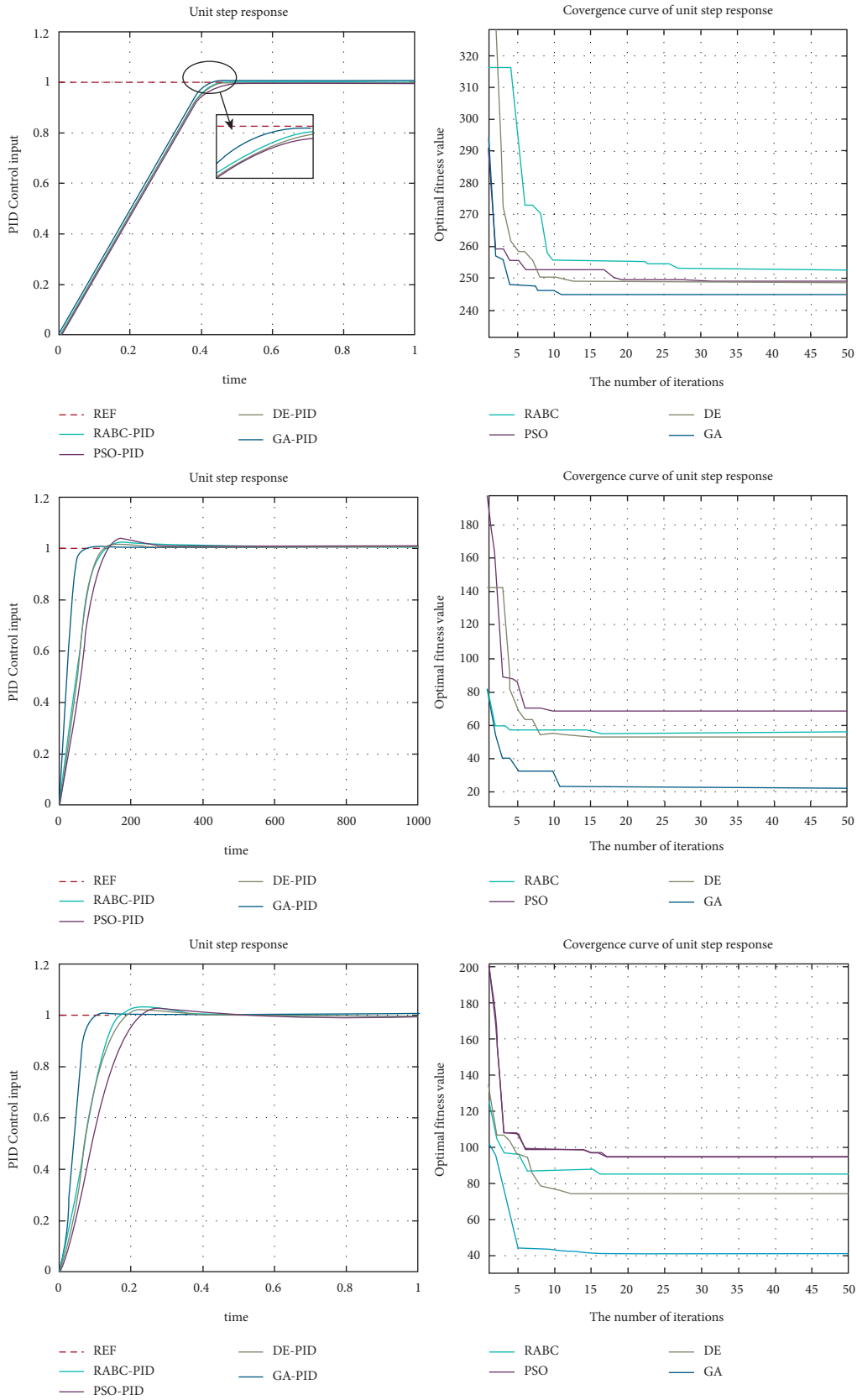


FIGURE 10: Unit step response curve and convergence curve when the parameters $T=20$, $T_1=15$, and $a_3=25$.

TABLE 6: Performance index when the parameters $T=10$, $T_1=5$, and $a_3=15$.

		t_r	t_p	$\sigma \%$	t_s	WITAE
RABC	DC	0.193	0.302	0.15831	0.232	128.9
	DCWT	40	69	0.073972	48	28.45
	PCSST	0.049	0.085	0.12111	0.06	34.49
PSO	DC	0.211	0.357	0.29852	0.263	138.5
	DCWT	119	201	7.5223	303	40.03
	PCSST	0.114	0.119	1.3367	0.143	52.98
DE	DC	0.209	0.345	0.42965	0.259	138.4
	DCWT	91	156	1.6836	112	52.83
	PCSST	0.11	0.187	2.051	0.196	64.6
GA	DC	0.211	0.711	0.16068	0.289	134.9
	DCWT	66	112	1.813	80	89.15
	PCSST	0.09	0.156	1.4905	0.112	65.28

TABLE 7: Performance Index when the parameters $T=15$, $T_1=10$, and $a_3=20$.

		t_r	t_p	$\sigma \%$	t_s	WITAE
RABC	DC	0.277	0.381	0.004674	0.307	184.8
	DCWT	40	71	0.248.2	50	25.63
	PCSST	0.064	0.111	0.54352	0.081	46.17
PSO	DC	0.283	0.438	0.080586	0.336	188.9
	DCWT	98	169	1.9875	121	58.68
	PCSST	0.157	0.266	2.8769	0.323	105.6
DE	DC	0.283	0.425	0.12588	0.333	188.8
	DCWT	79	137	1.4829	98	45.6
	MTCS	0.125	0.213	2.2948	0.239	84.63
GA	DC	0.285	0.422	0.28155	0.336	190.6
	DCWT	83	142	2.02663	146	50.28
	PCSST	0.152	0.271	1.0545	0.193	83.17

TABLE 8: Performance index when the parameters $T=20$, $T_1=15$, and $a_3=25$.

		t_r	t_p	$\sigma \%$	t_s	WITAE
RABC	DC	0.368	0.496	0.008282	0.408	245
	DCWT	46	81	0.33053	57	29.48
	PCSST	0.071	0.123	0.64848	0.089	41.43
PSO	DC	0.374	0.611	0.013246	0.443	249.1
	DCWT	110	186	3.024	228	68.17
	PCSST	0.117	0.3	2.767	0.36	94.74
DE	DC	0.373	0.545	0.10481	0.434	248.8
	DCWT	90	156	1.6576	112	52.67
	PCSST	0.139	0.236	2.4837	0.273	74.49
GA	DC	0.373	0.502	0.89717	0.423	253
	DCWT	94	162	1.9269	116	55.42
	PCSST	0.136	0.228	3.3499	0.288	85.31

5. Conclusions

In this paper, we propose an RABC algorithm to optimize the PID controller, to address the defects of the traditional ABC algorithm that the following bees follow the employed bees with too much randomness and the bee colony is not easy to approach the optimal food source, resulting in slow convergence and low accuracy, to increase the probability of following the employed bees that have a higher quality food source, and to introduce the global optimal bees to lead the other bees to move to a better food source. The RABC algorithm can quickly obtain the optimized value of the theory and improve the convergence speed and convergence

accuracy. The PID controller optimized by the RABC algorithm has the characteristics of no overshoot and fast response and has a better control effect.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was fully supported by the Liaoning Provincial Department of Education Basic Research Projects for Higher Education Institutions, China (no. LJKZ0301), the Scientific Research Foundation of the Education Department of Liaoning Province, China (no. 2017LNQN22), and the Young Teachers Foundation of University of Science and Technology Liaoning, China (no. 2017QN04).

References

- [1] B. Hekimoğlu, S. Ekinici, and S. Kaya, "Optimal PID controller design of DC-DC buck converter using whale optimization algorithm," in *Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, Turkey, 2018.

- [2] S. Tang and M. X. Wang, *Auto-Tuning PID Control for Long Time Delay Process*, IFAC, New York, NY, USA, 2002, in Chinese.
- [3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, 1995.
- [4] H. Feng, C.-B. Yin, W.-w. Weng et al., "Robotic excavator trajectory control using an improved GA based PID controller," *Mechanical Systems and Signal Processing*, vol. 105, pp. 153–168, 2018.
- [5] M. T. Özdemir and D. Öztürk, "Comparative performance analysis of optimal PID parameters tuning based on the optics inspired optimization methods for automatic generation control," *Energies*, vol. 10, p. 2134, 2017.
- [6] G. Chen, Z. Li, Z. Zhang, and S. Y. Li, "An improved ACO algorithm optimized fuzzy PID controller for load frequency control in multi area interconnected power systems," *IEEE Access*, vol. 8, pp. 6429–6447, 2019.
- [7] B. Hekimoğlu, "Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm," *IEEE Access*, vol. 7, pp. 38100–38114, 2019.
- [8] Z. Bingul and O. Karahan, "A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system," *Journal of the Franklin Institute*, vol. 355, no. 13, pp. 5534–5559, 2018.
- [9] S. Ekinici and B. Hekimoğlu, "Improved kidney-inspired algorithm approach for tuning of PID controller in AVR system," *IEEE Access*, vol. 7, pp. 39935–39947, 2019.
- [10] I. M. Batiha, J. Oudetallah, A. Ouannas, A. A. Al-Nana, and I. H. Jebri, "Tuning the fractional-order PID-controller for blood glucose level of diabetic patients," *International Journal of Advances in Soft Computing and its Applications*, vol. 13, no. 2, pp. 2710–1274, 2021.
- [11] H.-C. Huang and C.-C. Chuang, "Artificial bee colony optimization algorithm incorporated with fuzzy theory for real-time machine learning control of articulated robotic manipulators," *IEEE Access*, vol. 8, pp. 192481–192492, 2020.
- [12] N. M. Panoeiro, R. N. N. Koury, and L. Machado, "Development of an adaptive PID controller for superheating control employing artificial bee colony algorithm," in *Proceedings of the 16th International Refrigeration and Air Conditioning Conference*, Purdue University, IND, USA, 2016.
- [13] B. Ou, L. Song, and C. Chang, "Tuning of fractional PID controllers by using radial basis function neural networks," in *Proceedings of the IEEE ICCA 2010*, Xiamen, China, 2010.
- [14] R. Sharma, K. P. S. Rana, and V. Kumar, "Performance analysis of fractional order fuzzy PID controllers applied to a robotic manipulator," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4274–4289, 2014.
- [15] L. Y. Chang and H. C. Chen, "Tuning of fractional PID controllers using adaptive genetic algorithm for active magnetic bearing system," *WSEAS Transactions on Systems*, vol. 8, no. 2, pp. 158–167, 2009.
- [16] Z. Bingul, "A new PID tuning technique using differential evolution for unstable and integrating processes with time delay," in *Proceedings of the International Conference on Neural Information Processing*, Kolkata, India, 2004.
- [17] J. Y. Cao and B. G. Cao, "Design of fractional order controllers based on particle swarm optimization," in *Proceedings of the 2006 1ST IEEE Conference on Industrial Electronics and Applications*, Singapore, 2006.
- [18] D. Maiti, S. Biswas, and A. Konar, "Design of a fractional order PID controller using particle swarm optimization technique," 2008, <https://arxiv.org/abs/0810.3776>.
- [19] A. Alfi and H. Modares, "System identification and control using adaptive particle swarm optimization," *Applied Mathematical Modelling*, vol. 35, no. 3, pp. 1210–1221, 2011.
- [20] X. Li, Y. Wang, N. Li, M. Han, Y. Tang, and F. Liu, "Optimal fractional order PID controller design for automatic voltage regulator system based on reference model using particle swarm optimization," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 5, pp. 1595–1605, 2017.
- [21] D. Hu, Z. Qi, Y. Tang, and Y. He, "Research on fractional order PID controller applied to PEMFC pre-stage power conversion," in *Proceedings of the 2017 29th Chinese Control and Decision Conference (CCDC)*, in Chinese, Chongqing, China, 2017.
- [22] E. Sahin, M. S. Ayas, and I. H. Altas, "A PSO optimized fractional-order PID controller for a PV system with DC-DC boost converter," in *Proceedings of the 2014 16th International Power Electronics and Motion Control Conference and Exposition*, Antalya, Turkey, 2014.
- [23] S. F. Hussain, A. Pervez, and M. Hussain, "Co-clustering optimization using artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol. 97, Article ID 106725, 2020.
- [24] R. Zhang, X. Zhu, and W. Zhu, "Improved sample efficiency by episodic memory hit ratio deep Q-networks," *Journal of Applied and Numerical Optimization*, vol. 3, no. 3, pp. 513–519, 2021.
- [25] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, 2010.
- [26] X. Zhou, Z. Wu, and H. Wang, "Gaussian bare-bones artificial bee colony algorithm," *Soft Computing*, vol. 20, no. 2, pp. 907–924, 2016.
- [27] D.-L. Zhang, Y.-G. Tang, and X.-P. Guan, "Optimum design of fractional order PID controller for an AVR system using an improved artificial bee colony algorithm," *Acta Automatica Sinica*, vol. 40, no. 5, pp. 973–979, 2014.
- [28] N. T. An, P. D. Dong, and X. Qin, "Robust feature selection via nonconvex sparsity-based methods," *Journal of Nonlinear and Variational Analysis*, vol. 5, no. 1, pp. 59–77, 2021.