Received 22 May 2023, accepted 3 June 2023, date of publication 9 June 2023, date of current version 15 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3284969

RESEARCH ARTICLE

Neural Network Intelligent Control Based on MPSO

AIJUN KOU[1] AND XIAOJUN LI[2]
[1]School of Electronic Information, Huzhou College, Huzhou 313000, China
[2]School of Information Industry, Shanxi College of Technology, Shuozhou 036000, China
Corresponding author: Aijun Kou (hzkaj2022@163.com)

ABSTRACT With the increasing complexity of mechanical equipment, the control effectiveness of traditional intelligent control systems can no longer meet the needs of modern industrial production. In order to reduce errors in intelligent control systems while ensuring system performance, this study proposes a new Particle Swarm Optimization (PSO) optimization scheme. The study simplified the PSO algorithm from three aspects: algorithm parameters, speed, and position formula, and corrected the formulas for individual optimal values and global optimal values. Research will name the optimized algorithm Modified PSO (MPSO). On the basis of the MPSO algorithm, neural network intelligent control has been innovatively improved. In the experimental results, the MPSO optimized controller controlled the error within 0.01 within 0.02 seconds. At this time, the Whale Optimization Algorithm (WOA) optimized error was 0.072, and the PSO optimized error was 0.478. Compared to PSO and WOA, the control error of MPSO has decreased by 98.95% and 93.06%, respectively. In addition, the proposed method not only has the best control effect, but also has the shortest system response time, with an average time of 1.294 seconds. Compared to PSO and WOA optimization, it reduces by 61.48% and 43.07%, respectively.The results verified that the proposed method in this study can effectively improve the accuracy of intelligent control and control the error within the target range within 0.02 seconds. The research not only simplifies the calculation of the PSO algorithm, but also effectively reduces the error of the algorithm, providing a reference for research in the field of intelligent control.

INDEX TERMS Intelligent control, nonlinear problems, neural network, particle swarm optimization (PSO).

I. INTRODUCTION

In recent years, with the continuous development of industrial technology, the requirements for control systems have become increasingly high. However, traditional control methods are difficult to meet complex control requirements, and neural network control technology has been widely used in the control field due to its advantages of simplicity and strong robustness. Researchers have used neural network technology to design a new type of controller, which has achieved significant results in the field of control [1]. Neural network intelligent control technology can predict future behavior and results by learning and analyzing historical data of the system, and reduce control errors while ensuring system performance. Neural network technology can automatically adjust the network structure based on the current input data to achieve the optimal control effect [2].However, as the level of science and technology continues to improve and the complexity of control systems increases significantly, the existing neural network intelligent control technology can no longer meet the needs of modern industrial control systems [3]. Especially in industrial production, most of the controlled objects are multi-input and multi-output multivariable systems, which are extremely difficult to control. Moreover, due to the influence on individual inputs, the components of the control system form a coupled control system. Coupling is now widely seen in the control of engine speed, turbine temperature and so on. Severe coupling can lead not only to equipment damage, but also to serious production safety accidents. For this reason, the study of multivariable control systems has become the focus of research in the field of control at this stage. Meanwhile, the PSO algorithm, as a

The associate editor coordinating the review of this manuscript and approving it for publication was Felix Albu.

population intelligence algorithm, has been improved by a large number of experts and scholars in recent years, resulting in a variety of improved algorithms based on PSO.Adnan R M et al. proposed a new model combining PSO with the gray wolf algorithm for improving the accuracy of runoff prediction in reservoir operation [4].Yang et al. on the basis of the traditional wavelet neural network method developed a particle swarm optimization algorithm to solve the problems such as slow convergence speed of the traditional method in determining gear cracks in gearboxes [5].Although there has been a large amount of research on PSO optimization algorithms and certain achievements have been made, there is still relatively little research on the application of PSO algorithms to intelligent control systems. Therefore, a new PSO improvement scheme is proposed and innovatively applied to intelligent control systems.In order to reduce the lengthy learning time and the computational effort and error in the intelligent control system of neural networks, this study optimized the multivariate controller based on the standard Particle Swarm Optimization (PSO) algorithm. To address the problems of insufficient performance of the PSO algorithm in seeking and slow convergence of the algorithm, the study optimizes the PSO and proposes the MPSO algorithm. The study simplifies the PSO algorithm in terms of three aspects: algorithm parameters, velocity, and position equation. Based on the simplified PSO algorithm, the individual optimal position and the global optimal position in the position update formula are corrected using a linear combination of the individual optimal value and the global optimal value. In addition, the study innovatively proposes an inertia weight based on the cosine function and introduces a dynamic adjustment strategy obeying the Beta Distribution (BD) to enhance the population diversity and improve the exploration ability of the algorithm. The study optimizes thePIDNN network of Proportion Integration Differentiation (PID) combined with Artificial Neural Network (ANN) through the improved MPSO algorithm to solve the problem of weight correction in PIDNN.The research aims to optimize neural network intelligent control systems through MPSO, shorten the processing time of intelligent control systems and improve control effectiveness, provide solutions for solving complex system control problems, and indirectly improve the efficiency and safety of industrial production and management.In addition, the optimization approach adopted in the study provides a unique ideological guidance for the optimization method of PSO algorithm.

II. RELATED WORKS

With the endless research on optimization of various algorithms, numerous scholars have studied about intelligent control. Qiao B et al. proposed an intelligent control algorithm for tactile sharing in the field of automatic driving. Nash game is used to help determine the degree of intervention and driver's attention assessment. The simulation experiment verifies that the method proposed in this study can assist the driver in decision-making and improve driving safety [6]. Zhang et al. proposed a simplified pipeline intelligent control mode in natural gas transportation engineering. The test results in the actual pipeline operation data show that the optimized intelligent control can effectively and reliably predict the pipeline operation status [7]. Li et al. believed that the intelligent control algorithm can reduce the noise in the signal interference, so the research proposed an intelligent control algorithm that can accurately predict the flow. The research used Long Short Term Memory (LSTM) to predict the traffic demand and feedback the control scheme according to the results. This method achieved the balance between network speed and energy consumption [8]. Zhai et al. proposed a new intelligent control system based on knowledge transfer learning, which can ensure the accurate control of massage position and force when the massage robot is working. This research combines particle swarm optimization with knowledge transfer learning, and realizes personalized recognition of skin parameters on this basis. The experimental results indicate that the optimized control system can enable the massage robot to have the ability to accurately locate and control the force during operation [3]. Chen et al. proposed a finite-time intelligent control algorithm for performance regulation of cable parallel robot with output constraints. First of all, the system's dynamic model was established to ensure the normal operation of cables. The proposed controller's effectiveness of was verified by simulation of the control object [9]. Qassar et al. proposed a new control design to address the dynamic motion of aircraft caused by wing effects and improve the maneuverability of the aircraft during ascent. The method is based on super torsion Sliding mode control, and Lyapunov stability analysis is carried out to ensure the gradual convergence of the error. However, as the parameters of this method directly affect the dynamic performance of the controlled system, a modern technology based on whale optimization algorithm is proposed to ensure the optimal performance of the controller. The effectiveness of this method was verified through numerical simulation in MATLAB software [10].

At the same time, improved particle swarm optimization algorithms also emerge in endlessly. In industrial structure design, Zakian et al. proposed a weight minimization optimization design using heuristic improved particle swarm optimization, and introduced stress ratio, drift and size constraints in the optimization process. The comparison of simulation design experiments on pipe rack structure elaborated that the optimization method is more effective [11]. Priyadarshi et al. proposed a tracking algorithm built on PSO to enhance the power point of the Internet of Things (IoT) system. The study first optimized the converter and then used it as an interface to the solar PV voltage and modulated the converter using particle swarm algorithm to obtain more power. The monitoring of PV voltage by the controller is implemented on the basis of this method. The results show that the algorithm is superior to the techniques such as ant colony optimization and artificial bee colony optimization which are widely used at this stage [12]. In the field of solar power generation forecasting, Dash et al. designed a new hybrid forecasting method. This method optimized parameters by particle swarm optimization of sine and cosine flight, and used depth architecture instead of neural network to process huge database. To test the prediction accuracy of this method, the research selected the solar energy historical data of different seasons for testing, and verified the effectiveness of this method [13]. Ammar et al. designed a new multi-objective optimization PSO algorithm to deal with conflicting objective functions. The method deals with the decision variables of a multi-project capacity problem by a new particle coding and decoding approach. It also effectively improves the problem constraints by combining the overall violation minimization and constraint handling approach.The research used five performance indicators to compare the effectiveness of the algorithm, and verified that the efficiency of the algorithm is greater than other similar algorithms [14]. Fakoor et al. proposed an algorithm based on PSO in the research of satellite payload precise positioning. The optimal value of the controller variable and the normalized integral square error was searched by improving PSO. The simulation results demonstrated that the target detection algorithm based on improved PSO has improved in different degrees on the four indicators of error control. This verified that the proposed method can accurately track the time-varying payload of the satellite [15]. Wang et al. also proposed an improved particle swarm algorithm to solve the problems that the particle swarm algorithm is prone to too fast convergence and unbalanced local search and global search when dealing with complex problems. This study introduced an adaptive strategy to balance performance and accuracy in the search process. And the improved particle swarm algorithm was experimentally verified to have higher performance compared to the existing algorithm [16]. Wang et al. also proposed an improved particle swarm algorithm to solve the problems that the particle swarm algorithm is prone to too fast convergence and unbalanced local search and global search when dealing with complex problems. This study introduced an adaptive strategy to balance performance and accuracy in the search process. And the improved particle swarm algorithm was experimentally verified to have higher performance compared to the existing algorithm [12].

In summary, in the field of intelligent control, scholars have applied various optimization algorithms and conducted extensive research on the problems existing in intelligent control. However, the current optimization objectives are mostly limited to the application of control systems in different fields, ignoring the problems of long response time and large control errors inherent in intelligent control systems. At the same time, particle swarm optimization (PSO) algorithm has also demonstrated its unique advantages in solving complex problems, so improvements to PSO algorithm have also received a lot of attention. Based on the above research. This study optimizes neural network intelligent control systems based on particle swarm optimization, aiming to provide more effective improvement solutions for intelligent control systems.

III. NEURAL NETWORK INTELLIGENT CONTROL BASED ON MPSO OPTIMIZATION

A. RESEARCH ON MPSO OPTIMIZATION ALGORITHM

In recent years, with the continuous growth of computing power and data scale, the problems encountered in industrial production have become increasingly complex. Traditional optimization methods cannot meet the needs of modern industry in terms of solution accuracy and convergence speed when solving high-dimensional and complex problems. In order to break through the limitations of traditional algorithms, natural heuristic optimization algorithms have been widely applied and developed. A large number of practices and experiments have verified that swarm intelligence algorithm has more efficient performance in solving complex problems. At present, the widely used swarm intelligence algorithms include Ant Colony Optimization (ACO), particle swarm optimization (PSO) and Whale Optimization Algorithm (WOA). Among them,PSO algorithm is evolved from the theory of Complex Adaptive System (CAS), which originates from scholars' research on bird swarm system. Each bird in the flock is the subject of CAS, and each subject can communicate with other subjects and the environment, thus having adaptability. In communication, the subject changes the existing knowledge structure and behavior through learning. In the PSO algorithm, each bird is a particle, and each particle may be the optimal solution of the algorithm. The definition of particles consists of position, speed and fitness values [17]. Among them, the position of particles is random, and the speed controls the flying distance and direction. The fitness value reflects the quality of particles. There are also individual optimal values and global optimal values in PSO algorithm. The individual optimal value represents the best position in the individual particle experience, and the global optimal value is the best position in the group particle experience. In the search space of PSO algorithm, the expression of particle speed and position is shown in equation (1) [18].

$$\begin{cases} V_i^{t+1} = V_i^t + c_1 r_1 (p_{best} - X_i^t) + c_2 r_2 (g_{best} - X_i^t) \\ X_i^{t+1} = X_i^t + V_i^{t+1} \end{cases} \tag{1}$$

In equation (1), V_i^t denotes the velocity of the i th particle when the number of iterations is; tX_i^t denotes the position of the i th particle when the number of iterations is t; p_{best} denotes the optimal position obtained by the i th particle, i.e., the individual optimal value; g_{best} denotes the optimal position obtained by the particle swarm as a whole, i.e., the global optimal value; c_1, c_2 is the learning factor, $c_1 = c_2 = 2$; r_1, r_2 is two independent random numbers, $r_1, r_2 \in [0, 1]$. In order to coordinate the optimization capability of the PSO algorithm, an inertia weight w has been introduced into the PSO algorithm. The PSO speed formula after introducing the inertia weight is shown in Equation (2) [19].

$$V_i^{t+1} = wV_i^t + c_1 r_1 (p_{best} - X_i^t) + c_2 r_2 (g_{best} - X_i^t) \tag{2}$$

The introduction of inertia weights coordinates to a certain extent the optimization-seeking ability of the PSO algorithm, increasing the global search ability in the early stage of the algorithm search and the local search ability in the late stage of the algorithm search. However, the reduction of particle velocity in the late stage of algorithm search due to the introduction of inertia weights may lead to the reduction of population diversity, thus causing the algorithm to fall into local optimum. In addition, the PSO algorithm suffers from insufficient search performance and slow convergence of the algorithm. To solve these problems, this study first simplifies the PSO algorithm in terms of three aspects: algorithm parameters, velocity, and position equation. Then the individual optimal values and global optimal values in the position equation of the algorithm are corrected. Finally, BD's are introduced to dynamically adjust the inertia weights. This study names the improved PSO algorithm as MPSO. The simplification of PSO in this study first removes the particle velocity term from the position update formula, and the simplified PSO algorithm position update formula is shown in equation (3) [20].

$$X_i^{t+1} = wX_i^t + c_1r_1(p_{best} - X_i^t) + c_2r_2(g_{best} - X_i^t) \quad (3)$$

As shown in equation (3), the simplified PSO algorithm iterates with only the particle position term, thus turning the second-order differential equation into a first-order differential equation [11]. In order to accelerate the convergence speed of the PSO algorithm, the study introduces a linear combination to optimize equation (3). The optimized position update equation is shown in equation (4).

$$\begin{aligned} X_i^{t+1} = wX_i^t + c_1r_1(\frac{p_{best} + g_{best}}{2} - X_i^t) \\ + c_2r_2(\frac{p_{best} - g_{best}}{2} - X_i^t) \end{aligned} \quad (4)$$

In equation (4), $c_1r_1(\frac{p_{best}+g_{best}}{2} - X_i^t)$ leads the particle to move from the current position to the mean of the individual optimum and the global optimum; $c_2r_2(\frac{p_{best}-g_{best}}{2} - X_i^t)$ leads the particle to move from the current position to the negative direction of the mean of the individual optimum and the global optimum. After the linear combinatorial optimization, the information of the particle itself and the global position is applied more effectively, thus bringing the particle closer to the optimal solution. BD is a continuous probability distribution in the interval (0, 1), and its probability density function is shown inequation (5) [21].

$$f(x) = \frac{x^{a-1}(1-x)^{b-1}}{B(a,b)} \quad (5)$$

In equation (5), B represents the beta function, and a and b are the parameters. When the parameters a, b take different values, the probability density function can fit different function shapes. The expression of inertia weight after introducing BD is shown in equation (6).

$$\begin{aligned} w_0 = w_{\min} + (w_{\max} - w_{\min}) \times \cos(\pi t/2T_{\max}) \\ + \sigma \times betarnd(a, b) \end{aligned} \quad (6)$$

In equation (6), σ denotes the inertia adjustment factor; $betarnd(a, b)$ denotes the random number obeying BD. The optimized inertia weights can control the range of taking values in the range of 0.4∼0.9, thus enhancing the search ability of the algorithm. Moreover, the deviation degree of inertia weights is controlled under the influence of BD, so that the distribution of their values is more reasonably adjusted [22]. The inertia weights of MPSO show a nonlinear variation overall and decreasing with the algorithm search. After adding the adjustment factor of obeying BD, it produces larger adjusted values of inertia weights to improve the particle population diversity at the early stage of algorithm search, and larger inertia weights to improve the search accuracy at the later stage of algorithm search. the algorithm flow of PSO and MPSO is shown in Figure 1.

B. APPLICATION OF MPSO OPTIMIZATION ALGORITHM IN NEURAL NETWORK INTELLIGENT CONTROL

Intelligent control is an automation technology that allows him to achieve control objectives by artificial intelligence methods without the need for explicit mathematical models and precise control rules [23]. Intelligent control can be applied to many fields, such as industrial production, transportation, medical, energy and so on. In intelligent control systems, the controller can automatically adjust the control parameters and algorithms according to the characteristics and requirements of the controlled object, so as to achieve the optimal control of the controlled object [24]. Compared with traditional control algorithms, intelligent control systems have stronger robustness and self-adaptability, and can achieve better control effects in complex and changing situations. The basic principle structure of the intelligent control system, as shown in Figure 2.

Figure 2 shows the working principle of a representative intelligent control system. Taking the intelligent robot control system as an example, the broad object includes the robot arm as well as the object being operated; sensors include force sensors, tactile sensors, and visual sensors; sensory information represents the raw data obtained by sensors; cognitive systems are used for information reception, storage, analysis, and decision making; communication interfaces establish human-robot interaction and links between different modules; planning and control systems can specific control according to the task requirements and act on the control object through actuators [25]. Neural network intelligent control is based on control theory and intelligence theory as the basis for implementation, and intelligent operation as the implementation step. In the field of neural network intelligent control, Proportion Integration Differentiation (PID) control is the most commonly used control law [12].The mainstream way to intelligentize nonlinear PID control is to apply neural network control systems, transforming the control object into an Artificial Neural Network (ANN) controller, thereby achieving intelligent control. The principle of PID and the basic structure of ANN nonlinearcontrol system is shown in Figure 3.

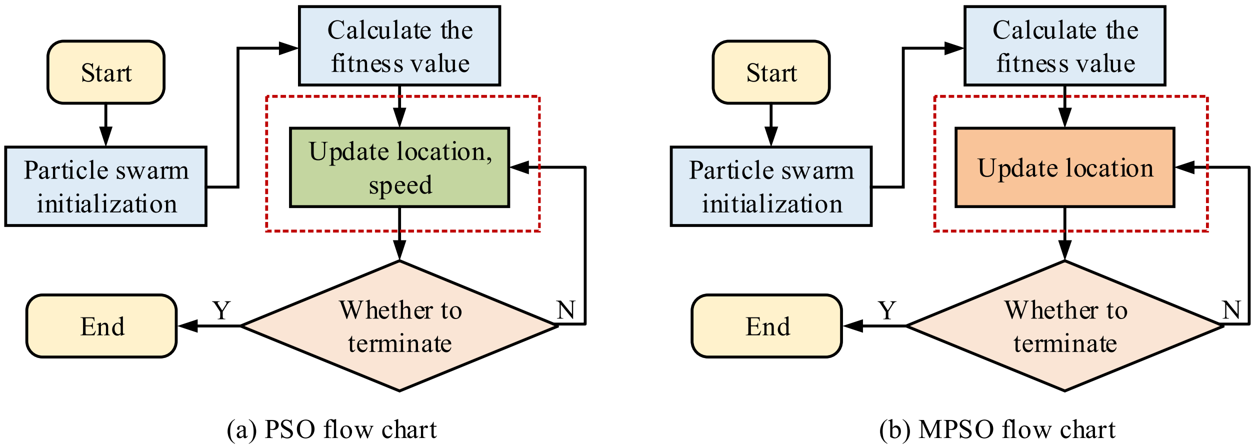

FIGURE 1. PSO and MPSO flow chart.

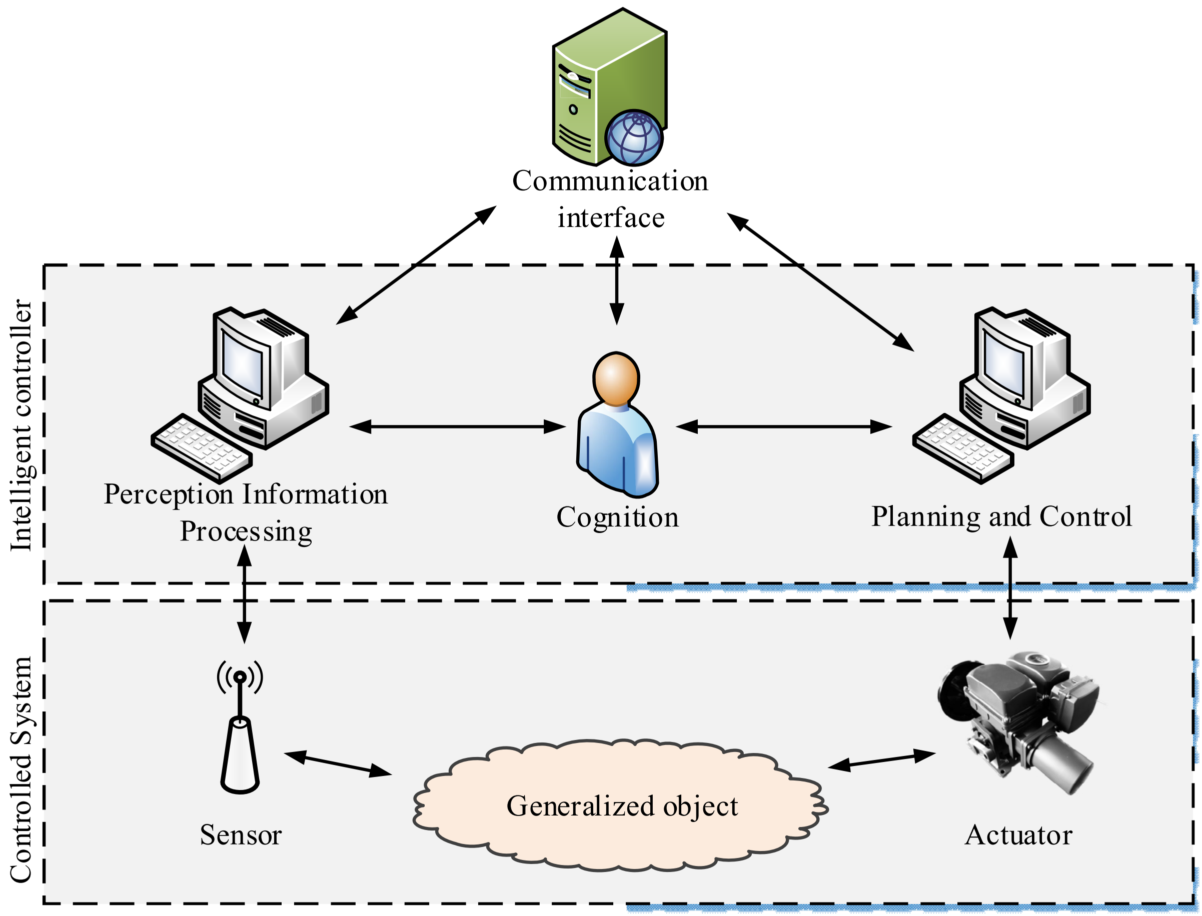

FIGURE 2. Improved MPSO algorithm running process.

As shown in Figure 3(a), there are three correction steps in the traditional PID controller: proportional, integral and differential. Among them, the proportion correction is carried out in real time. As long as there is deviation, it will be corrected immediately. Therefore, the response speed is fast, but it is easy to overshoot. The integral correction is used to reduce the overshoot and achieve no static error control. The ANN in Figure 3(b) is a network system constructed by simulating the human brain's nervous system. Compared with the traditional PID controller, the construction of ANN-based controller has nothing to do with the mathematical model of the controlled object, and the structure is easy to implement.

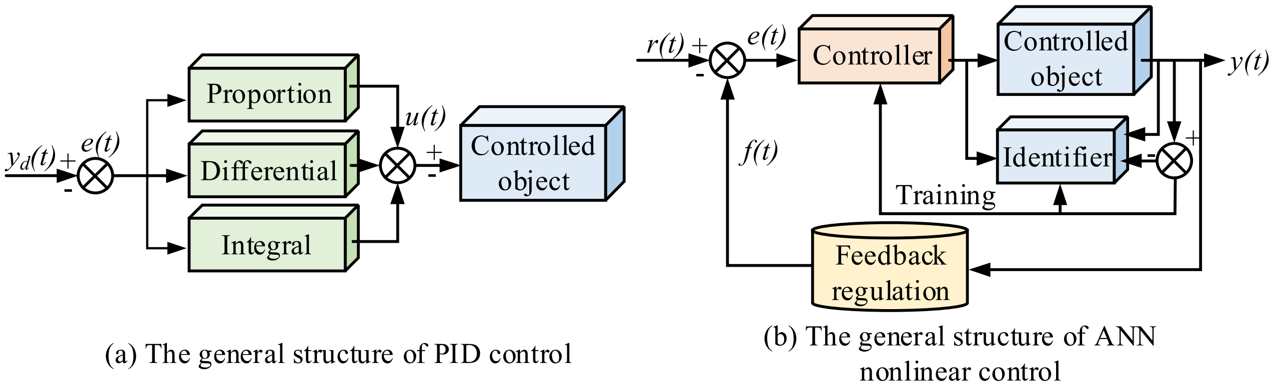

FIGURE 3. **PID principle and basic structure of ANN nonlinear control system.**

However, with the continuous improvement of science and technology, in the modern industrial intelligence control system, not only the controlled object becomes more complex, but also the system is affected by many control variables at the same time. Such problems not only lead to poor stability of the intelligence control system, but also lead to equipment damage and safety accidents. For this reason, this study combines a PID controller with an ANN control machine to obtain a PIDNN multivariable nonlinear controller. And based on this, MPSO algorithm is applied to optimize it. When the sampling moment is k, the output of the input layer neurons in the summarized PIDNN controller is shown in equation (7) [26].

$$x_{si}(k) = \begin{cases} 1, & u_{sj}(k) > 1 \\ u_{si}(k), & -1 \leq u_{sj}(k) \leq 1 \\ -1, & u_{sj}(k) < -1 \end{cases} \tag{7}$$

In equation (7), $y_s(k)$ indicates the system controlled quantity; $r_s(k)$ indicates the system given value; i is the input layer serial number in the subnet; s is the subnet serial number. The output of the summarized implicit layer is shown in equation (8) [27].

$$x'_{sj}(k) = \begin{cases} 1, & u'_{sj}(k) > 1 \\ u'_{sj}(k), & -1 \leq u'_{sj}(k) \leq 1 \\ -1, & u'_{sj}(k) < -1 \end{cases} \tag{8}$$

In equation (8), j is the implicit layer neuron serial number. The weighted sum of all the output values of the implicit layer constitutes the output of the neurons in the output layer, and the expression is summarized as shown in in equation (9) [28].

$$x_h{}''(k) = \begin{cases} 1, & u_h{}''(k) > 1 \\ u_h{}''(k), & -1 \leq u_h{}''(k) \leq 1 \\ -1, & u_h{}''(k) < -1 \end{cases} \tag{9}$$

In equation (9), h is the serial number of the neuron in the output layer. PIDNN generates control quantity error in the control process, and for this reason, the study corrects the weights in PIDNN by the gradient correction method to make the control quantity close to the target value. Firstly, the control quantity error is calculated as shown in equation (10).

$$J = \frac{1}{2}\sum_{k=1}^{3}[y_h(k) - r(k)]^2 \tag{10}$$

In equation (10), y_h denotes the predicted output value and r denotes the control target. After getting the error, the weights from the input layer to the implied layer and from the implied layer to the output layer are corrected separately, and the correction formula for the weights from the input layer to the implied layer is shown in equation (11).

$$w_{ij}(k+1) = w_{ij}(k) - \eta_1 \frac{\partial J}{\partial w_{ij}} \tag{11}$$

The weight correction formula for the implied layer to the output layer is shown in equation (12).

$$w_{jk}(k+1) = w_{jk}(k) - \eta_2 \frac{\partial J}{\partial w_{jk}} \tag{12}$$

In equation (12), η_1 and η_2 denote the corresponding learning rates, respectively. Therefore, the closed-loop control system formed by the neural network and the controlled system in the PIDNN is shown in Fig. 4.

In Figure 4, r denotes the control target; u denotes the control rate; and y denotes the current control amount. To address the problem that the controller depends on the initial weights, this study uses MPSO for optimization to improve the control performance of the controller by dynamically adjusting the initial weights of the network.The dynamic adjustment process of MPSO is to set the relevant parameters according to the specific problem. First, the dimensionality of the particle population is estimated based on the topology of the network, and the dimensionality is used as the connection weight or threshold of the network. The fitness of the particles increases as the training error decreases. The algorithm can only terminate the iteration when the fitness of the particles is lower than a certain setting or the number of iterations exceeds a certain setting, and the particle with the highest fitness is the optimal solution at that time. the flow of MPSO optimization for each parameter is shown in Figure 5.

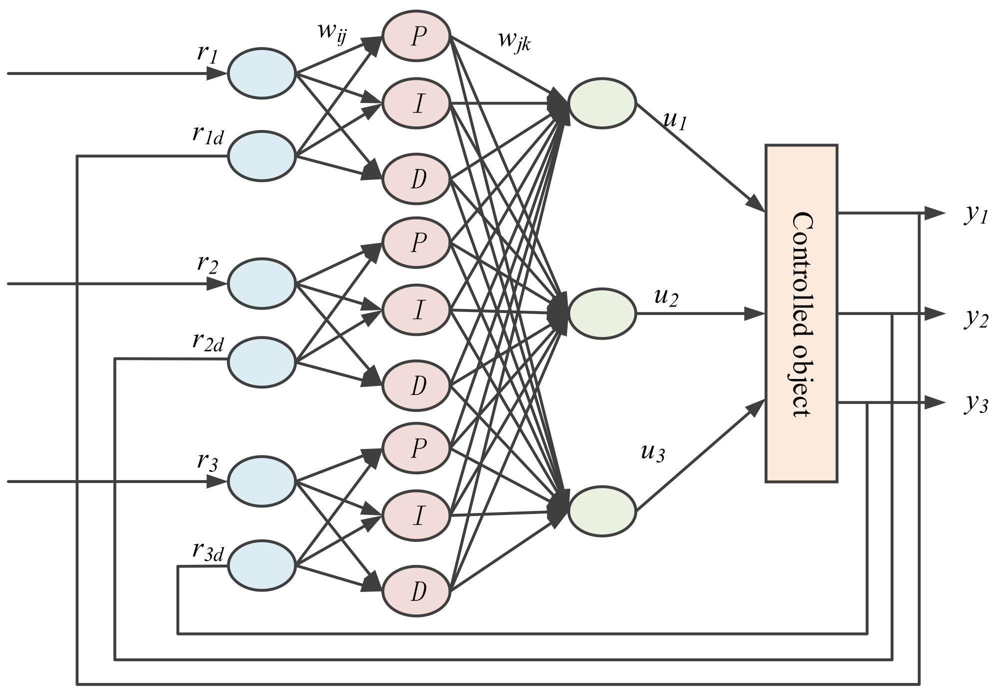

FIGURE 4. PIDNN Closed loop control system.

IV. COMPARISON OF ALGORITHM PERFORMANCE AND EXPERIMENTAL ANALYSIS OF CONTROL SYSTEM

A. MPSO ALGORITHM PERFORMANCE COMPARISON

To ensure that the computer equipment will not produce errors, this experiment uses the same computer equipment for simulation testing.The computing graphics card used for the experiment is GTX 1080ti; the CPU is Inter Xeon E5; the memory is 64 GB; the operating system is Windows 10; and the deep learning framework used for the experiment is TensorFlow 1.8. Before testing the performance of the population intelligence optimization algorithm, it is also necessary to select different standard test functions.The study selected the multidimensional unimodal function Quartic with random noise, the unimodal function Rosenbrock that is difficult to solve, and the multimodal function Griewank that is prone to trapping the algorithm into local optima as the standard test functions for this study. The detailed information of the test function is shown in Table 1.

The study determined the range of optimal parameters for the MPSO algorithm through preliminary experiments. In order to further determine the optimal parameter settings for the MPSO algorithm, the study conducted tests on different subgroup numbers and particle count parameters. To verify the effect of different subgroup numbers on the performance of MPSO algorithm, simulation experiments were conducted on the basis of three standard test functions, and the results are shown in Figure 6.

In this experiment, the number of subgroups set in the study is 3 groups, 6 groups and 9 groups, the total number of particles is 90, and the other parameters remain unchanged. Overall, using the test function for testing, the MPSO algorithm with different subpopulations showed a faster decline in the first 50 iterations and a smoother trend after 50 iterations. Figure 6(a) is the test result on the function Quartic. When the number of subgroups is 9, the algorithm converges in 48 iterations with a fitness of 0.31. When the number of subgroups is 6, the algorithm converges in 43 iterations, and the fitness is 0.52. When the number of subgroups is 3, it converges at 38 iterations, and the fitness is 0.58. It can be seen that when the number of subgroups is 3 and 6, the curve of fitness value is relatively close to that of fitness value of 9 subgroups. Figure 6 (b) is the test result on the function Resenblock. When the number of subgroups is 9, the algorithm converges in 43 iterations with a fitness of 82.34. When the number of subgroups is 6, the algorithm converges in 41 iterations with a fitness of 88.73. When the number of subgroups is 3, it converges at 40 iterations, and the fitness is 132.17. It can be seen that when the number of subgroups is 6 and 9, the curve of fitness value is relatively close, and the curve of fitness value is quite different from that of 3 subgroups.Figure 6(c) is the test result on the function Griewank. When the number of subgroups is 9, the algorithm converges in 46 iterations with a fitness of 0.34. When the number of subgroups is 6, the algorithm converges in 32 iterations with a fitness of 0.48. When the number of subgroups is 3, it converges at 29 iterations, and the fitness is 1.48.It can be seen that when the number of subgroups is 6 and 9, the curve of fitness value is relatively close to that of fitness value of 3 subgroups. From the experimental results, it can be seen that when the number of subgroups is 3, there is a faster convergence speed; When

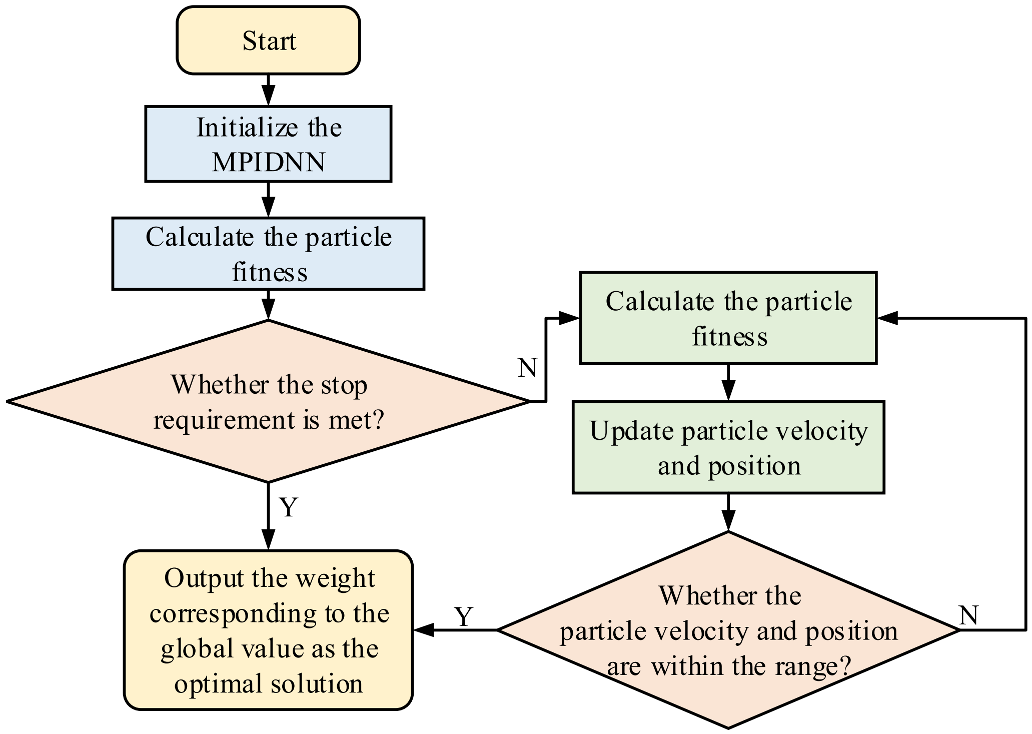

FIGURE 5. MPSO optimizes the flow chart of each parameter.

TABLE 1. Information on standard test functions.

Function	Formula	Global optimal function value	Characteristic
Quartic	$f_1(x)=\sum_{i=1}^{30} ix_i^4 + random[0,1)$	$f_1(x)=0$	Multidimensional unimodal functions with random noise
Rosenbrock	$f_2(x)=\sum_{i=1}^{n}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2)$	$f_2(x)=0$	Unimodal function, difficult to solve
Griewank	$f_3(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	$f_3(x)=0$	Multimodal function, difficult to solve

the number of subgroups is 9, it has a better fitness value. The relationship between the convergence rate, fitness value and the number of subgroups is revealed. Therefore, in practical applications, the number of subgroups can be set based on specific engineering goals and requirements.

Further verifying the influence of different total particle numbers on the MPSO algorithm's performance when the number of sub-groups is the same, as shown in Figure 7.

In this experiment, the number of subgroups is set to 6 groups, the total number of particles is set to 60, 90 and 120 respectively, and the other parameters remain unchanged. From Figure 7, it can be seen that as the total number of particles increases, the search accuracy and convergence time of the three test functions all increase. Overall, using the test function for testing, the MPSO algorithm with different particle counts experienced a rapid decline in the first 50 iterations and tended to flatten out after 50 iterations.As the sum of particles increases in Figure 7, the search accuracy of the three test functions increases, and the time of the optimal solution decreases slightly. Figure 7(a) is the test result on the function Quartic. When the sum of particles is 60 and 90, the algorithm converges in 27 iterations with a fitness of 1.22 and 0.82. When the sum of particles is 120, it converges at 31 iterations, and the fitness is 0.38. Figure 7(b) is the test result on the function Resenblock. When the total number of particles is 60, 90, 120, the algorithm converges in 24, 25, 27 iterations with a fitness of 132.45, 81.82 and 57.23 respectively. It can be seen that when the total number of particles is 90 and 120, the fitness value curve is relatively close, and it is quite different from the fitness value curve when the total number of particles is 60.Figure 7(c) is the test result on the function Griewank. When the sum of particles is 60, 90,

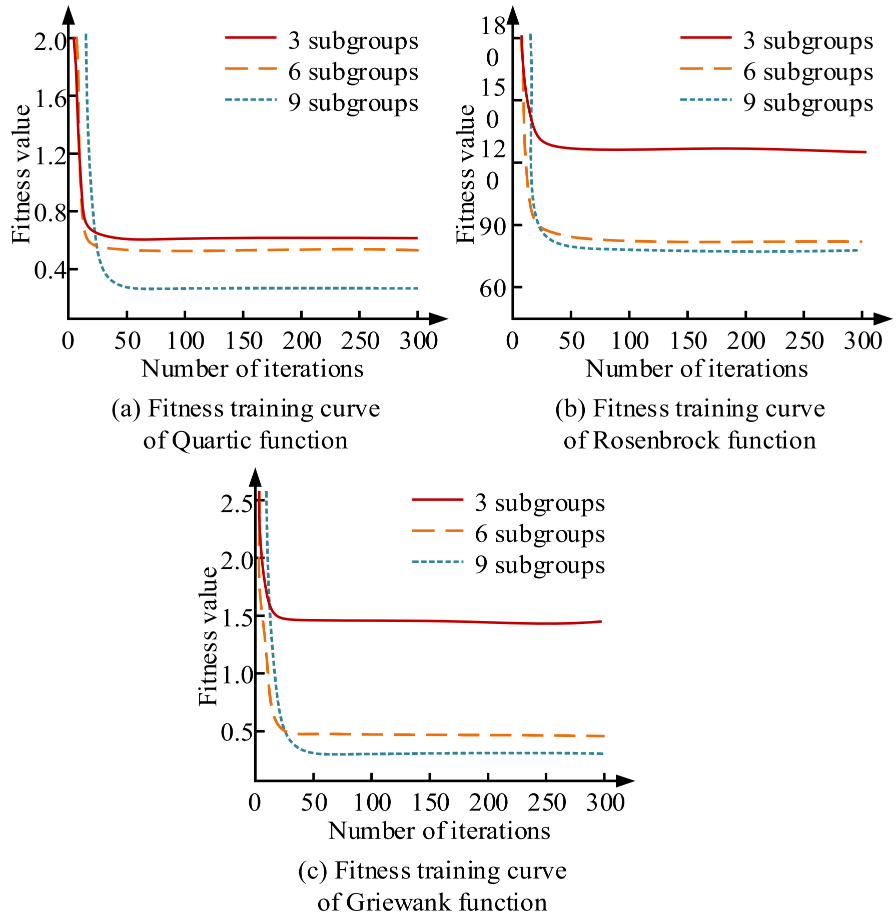

FIGURE 6. MPSO fitness changes with different subgroup numbers.

120, the algorithm converges in 32, 34, 36 iterations with a fitness of 0.67, 0.48 and 0.32 respectively. It can be seen that when the total number of particles is 90 and 120, the fitness value curve is relatively close, and it is quite different from the fitness value curve when the total number of particles is 60. From the experimental results, it can be seen that when the total number of particles is 60, there is a faster convergence speed; When the total number of particles is 120, it has a better fitness value. The relationship between the convergence rate, fitness value and the number of particles is revealed. Provide reference for the application of algorithms in specific projects.The conclusion can be drawn that the improvement of the algorithm depends on the representation of the optimal information of the subpopulation in the subpopulation and the difference between the subpopulations, which is consistent with the influence of the number of subpopulations in the simulation test on the optimization effect.

In the performance comparison of the algorithms, the standard PSO algorithm, the hybrid PSO algorithm (Crossover PSO,CRPSO) and the whale optimization algorithm (WOA), which are widely used at present, are chosen as the comparison algorithms in the study. Training on test functions Rosenbrock and Griewank was performed to obtain the algorithm performance comparison results, as shown in Figure 8.

Figure 8(a) shows the training results on the Rosenbrock function, where MPSO reaches convergence with 62 iterations, PSO reaches convergence with 83 iterations, CRPSO reaches convergence with 121 iterations, and WOA reaches convergence with 76 iterations. where MPSO converges significantly faster than PSO and CRPSO, and slightly faster than WOA, but at convergence time MPSO has the best fitness among the four types of algorithms involved in the comparison. Figure 8(b) shows the training results on the Griewank function, where MPSO reaches convergence with 81 iterations, PSO reaches convergence with 93 iterations, and CRPSO reaches convergence with 99 iterations. WOA reaches convergence with 112 iterations. although the fitness value of WOA is close to that of MPSO, MPSO converges significantly faster than PSO, CRPSO, and WOA, From Fig. 8, it can be seen that for each test function, MPSO has better finding accuracy and good finding speed compared to PSO and CRPSO.

B. EXPERIMENTAL ANALYSIS OF OPTIMIZED PIDNN INTELLIGENT CONTROL SYSTEM

In order to better compare the optimization control capabilities of different algorithms, the study compares the effects of WOA algorithm optimization, PSO algorithm optimization,

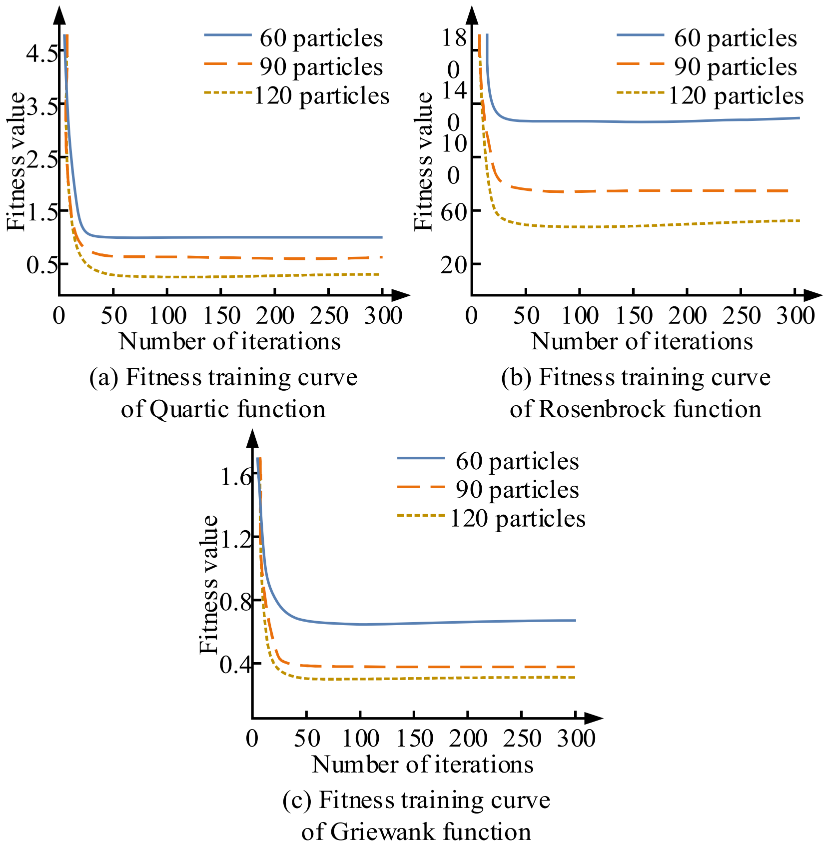

FIGURE 7. **MPSO fitness changes with different particle numbers.**

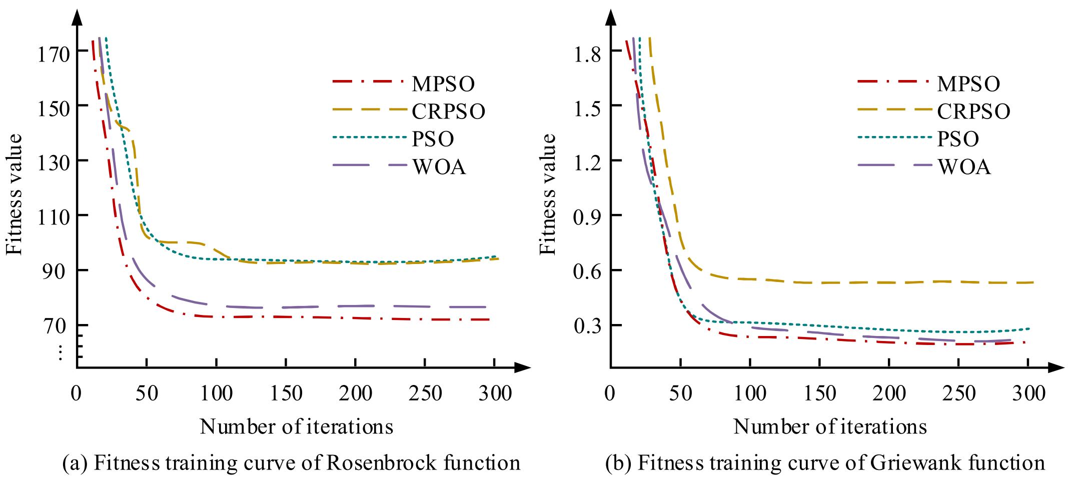

FIGURE 8. **Performance comparison of different algorithms.**

and the MPSO algorithm optimization proposed in this study. Study the control error of each optimization model under different target control quantities using a motor control system commonly used in PIDNN control as the controlled object. As shown in Figure 9.

Figure 9(a) shows the comparison of the control effects of the motor when the control amount is 0.7. Among them, the PSO optimization control reached stability at 0.091s, with a stable control value of 0.642; The WOA optimization control reached stability at 0.043 seconds, with a stable control value of 0.641; The MPSO optimized control reached stability at 0.027 seconds, with a stable control value of 0.696. Figure 9(b) shows the comparison of the control effects of the motor when the control amount is 0.4. Among them, the

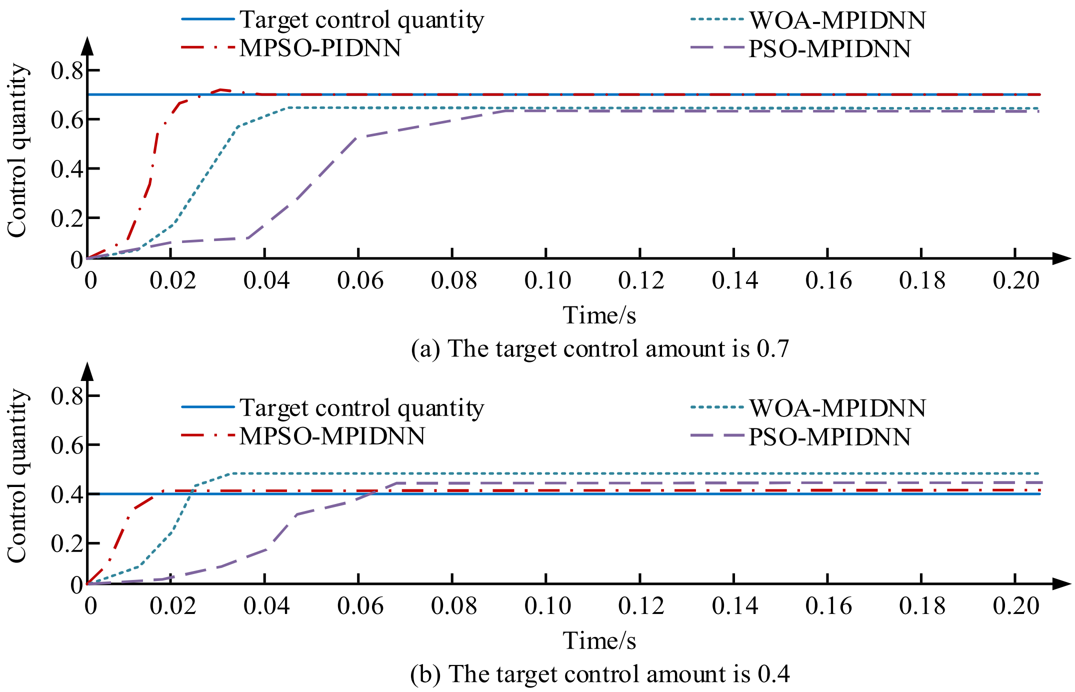

FIGURE 9. Comparison of control effect of different algorithms.

TABLE 2. Performance comparison of various optimization control methods.

Evaluating indicator		Method		
		MPSO-PIDNN	PSO - PIDNN	WOA-PIDNN
Absolute error	0.4	0.003	0.087	0.083
	0.7	0.004	0.058	0.059
Relative error	0.4	0.75%	21.75%	20.75%
	0.7	0.57%	8.29%	8.43%
Stability time	0.4	0.018s	0.091s	0.031s
	0.7	0.027s	0.091s	0.043s
System response time	0.4	1.245s	3.211s	2.231s
	0.7	1.344s	3.506s	2.314s

PSO optimization control reached stability at 0.091s, with a stable control value of 0.487; The WOA optimization control reached stability at 0.031s, with a stable control value of 0.483; The MPSO optimization control reached stability at 0.018s, with a stable control value of 0.403. The absolute error, relative error and system response time when the target control quantity of the motor is 0.4 and 0.7 are counted, as shown in Table 2.

From Table 2, it can be seen that the proposed method not only has the best control effect, but also has the shortest system response time, with an average time of 1.294 seconds. Compared to PSO and WOA optimization, it has decreased by 61.48% and 43.07%, respectively. The feasibility and advantages of using MPSO to optimize neural network intelligent control in this study have been verified. The error control curves for comparing different algorithms are shown in Figure 10.

As shown in Figure 10, in the error curve of each algorithm optimization NN-intelligence control system, the error of MPSO optimization control is the smallest, and the time to reach the minimum error is the shortest. The error of MPSO is controlled within 0.01 within 0.02s. At this time, the error of PSO optimization control is 0.072, and the error of BP optimization control is 0.478. At 0.1s, all three models participating in the comparison achieved stable control effects, with a control error of 0.067 for MPSO-PIDNN; The control error of WOA-PIDNN is 0.132; The control error of PSO-PIDNN is 0.198.Compared with PSO and WOA optimization control,

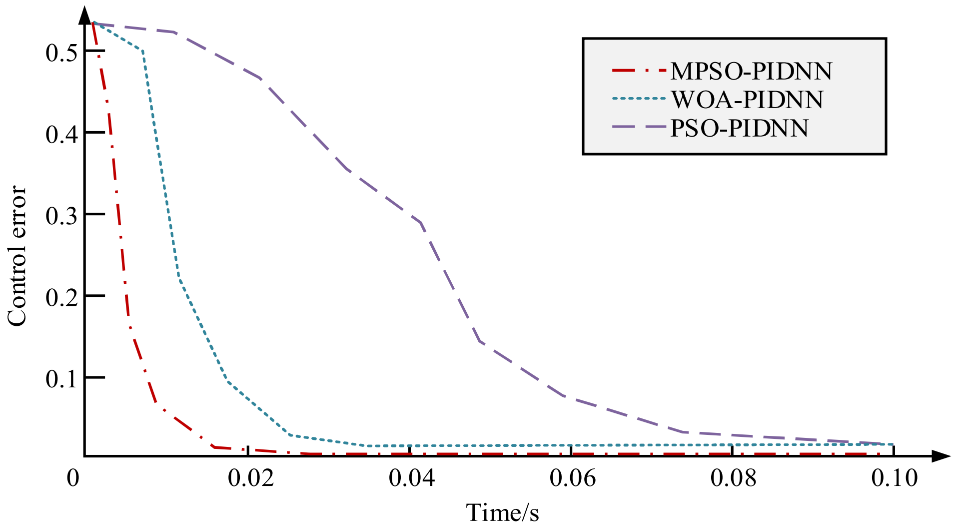

FIGURE 10. Comparison of error control curves of different algorithms.

the error of MPSO optimization control at 0.02s decreased by 98.95% and 93.06%, respectively; At 0.04s, the errors decreased by 72.23% and 98.38% respectively; The errors decreased by 66.16% and 49.24% at 0.1s, respectively. The experiment proved the PIDNNintelligence control system, which is based on MPSO optimization, is accurate and effective, and it validated the usefulness of the improvement approach suggested in this study. However, in this experiment, the experimental parameters were set too ideal, the common random disturbances were not analyzed exhaustively, and the experimental sample was not large enough, so there may still be large errors. If the applicability of this study is improved, it will be the main research direction in the future.

V. CONCLUSION

With the continuous innovation of science, technology and means, the intelligence control system also urgently needs to be optimized. In order to improve the response time and control effectiveness of neural network control, this study optimized neural network intelligent control based on the PSO improved MPSO algorithm. In the experimental results, the iteration number of MPSO algorithm is 62 when it converges on Rosenbrook function and 81 when it converges on Griewank function, which is significantly lower than other algorithms and has higher accuracy. When the control amount is 0.7, the PSO optimization control reaches stability at 0.091s, with a stable control value of 0.642; The WOA optimization control reached stability at 0.043 seconds, with a stable control value of 0.641; The MPSO optimized control reached stability at 0.027 seconds, with a stable control value of 0.696; When the control amount is 0.4, the PSO optimization control reaches stability at 0.091s, with a stable control value of 0.487; The WOA optimization control reached stability at 0.031s, with a stable control value of 0.484. In addition, compared to PSO optimization and WOA optimization, the error of the PIDNN controller optimized by MPSO decreased by 98.95% and 93.06% at 0.02s, respectively; The errors at 0.04s were reduced by 72.23% and 98.38%, respectively, and at 0.1s, the errors were reduced by 66.16% and 49.24%, respectively. The experimental results show that the proposed method can effectively reduce control errors and reduce the time consumption of intelligent control, verifying the effectiveness of this method. However, in this experiment, if the experimental parameters are set too ideal, the common random interferences have not been thoroughly analyzed, and the experimental samples are not large enough, so there may still be significant errors. How to improve the applicability of this study will become the main research direction in the future.

REFERENCES

[1] Y. Wu, J. Liu, Y. Wei, D. An, Y. Duan, W. Li, B. Li, Y. Chen, and Q. Wei, "Intelligent control method of underwater inspection robot in netcage," *Aquaculture Res.*, vol. 53, no. 5, pp. 1928–1938, Apr. 2022.
[2] Q. Liu and K. R. Thorup, "Intelligent control method of accounting information based on multi-objective evolution," *Int. J. Infomation Technol. Manag.*, vol. 21, no. 1, pp. 97–114, 2022.
[3] J. Zhai, X. Zeng, and Z. Su, "An intelligent control system for robot massaging with uncertain skin characteristics," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 49, no. 4, pp. 634–644, Jun. 2022.
[4] R. M. Adnan, R. R. Mostafa, O. Kisi, Z. M. Yaseen, S. Shahid, and M. Zounemat-Kermani, "Improving streamflow prediction using a new hybrid ELM model combined with hybrid particle swarm optimization and grey wolf optimization," *Knowl.-Based Syst.*, vol. 230, no. 6, 2021, Art. no. 107379.
[5] L. Yang and H. Chen, "Fault diagnosis of gearbox based on RBF-PF and particle swarm optimization wavelet neural network," *Neural Comput. Appl.*, vol. 31, no. 9, pp. 4463–4478, Sep. 2019.
[6] B. Qiao, H. Li, and X. Wu, "Intelligent-assist algorithm for remote shared-control driving based on game theory," *J. Shanghai Jiaotong Univ.*, vol. 26, no. 5, pp. 615–625, Oct. 2021.
[7] T. Zhang, H. Bai, and S. Sun, "A self-adaptive deep learning algorithm for intelligent natural gas pipeline control," *Energy Rep.*, vol. 7, pp. 3488–3496, Nov. 2021.
[8] J. Li, J. Wen, and M. Sheng, "Intelligent power control algorithm in heterogeneous wireless cellular networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8823–8837, Sep. 2021.

[9] Z. S. Chen, Y. H. Cheng, and X. S. Wang, "Prescribed performance control of redundantly-actuated cable driving parallel robots subjected to output constraint," *Acta Automatica Sin.*, vol. 48, no. 7, pp. 1704–1717, 2022.

[10] A. A. Al-Qassar, A. S. M. Al-Obaidi, A. F. Hasan, A. J. Humaidi, A. R. Nasser, A. Alkhayyat, and I. K. Ibraheem, "Finite-time control of wing-rock motion for delta wing aircraft based on whale-optimization algorithm," *Indonesian J. Sci. Technol.*, vol. 6, no. 3, pp. 441–456, Aug. 2021.

[11] P. Zakian, B. Ordoubadi, and E. Alavi, "Optimal design of steel pipe rack structures using PSO, GWO, and IGWO algorithms," *Adv. Struct. Eng.*, vol. 24, no. 11, pp. 2529–2541, Aug. 2021.

[12] N. Priyadarshi, S. Padmanaban, J. B. Holm-Nielsen, M. S. Bhaskar, and F. Azam, "Internet of Things augmented a novel PSO-employed modified zeta converter-based photovoltaic maximum power tracking system: Hardware realisation," *IET Power Electron.*, vol. 13, no. 13, pp. 2775–2781, Oct. 2020.

[13] D. R. Dash, P. K. Dash, and R. Bisoi, "Short term solar power forecasting using hybrid minimum variance expanded RVFLN and sine-cosine levy flight PSO algorithm," *Renew. Energy*, vol. 174, pp. 513–537, Aug. 2021.

[14] H. B. Ammar, W. B. Yahia, O. Ayadi, and F. Masmoudi, "Design of efficient multiobjective binary PSO algorithms for solving multi-item capacitated lot-sizing problem," *Int. J. Intell. Syst.*, vol. 37, no. 2, pp. 1723–1750, Feb. 2022.

[15] M. Fakoor, S. Nikpay, and A. Kalhor, "On the ability of sliding mode and LQR controllers optimized with PSO in attitude control of a flexible 4-DOF satellite with time-varying payload," *Adv. Space Res.*, vol. 67, no. 1, pp. 334–349, Jan. 2021.

[16] R. Wang, K. Hao, L. Chen, T. Wang, and C. Jiang, "A novel hybrid particle swarm optimization using adaptive strategy," *Inf. Sci.*, vol. 579, pp. 231–250, Nov. 2021.

[17] A. Hamza and N. B. Yahia, "Heavy trucks with intelligent control of active suspension based on artificial neural networks," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 235, no. 6, pp. 952–969, Jul. 2021.

[18] C. Sancak, F. Yamac, M. Itik, and G. Alici, "Force control of electro-active polymer actuators using model-free intelligent control," *J. Intell. Mater. Syst. Struct.*, vol. 32, no. 17, pp. 2054–2065, Oct. 2021.

[19] P. K. Muthusamy, M. Garratt, H. Pota, and R. Muthusamy, "Real-time adaptive intelligent control system for quadcopter unmanned aerial vehicles with payload uncertainties," *IEEE Trans. Ind. Electron.*, vol. 69, no. 2, pp. 1641–1653, Feb. 2022.

[20] Y. Sun, H. Zhao, Z. Chen, X. Zheng, M. Zhao, and B. Liang, "Fuzzy model-based multi-objective dynamic programming with modified particle swarm optimization approach for the balance control of bicycle robot," *IET Control Theory Appl.*, vol. 16, no. 1, pp. 7–19, Jan. 2022.

[21] K. P. Dijkman, T. Mohns, J. P. Dieleman, C. van Pul, T. G. Goos, I. K. Reiss, P. Andriessen, and H. J. Niemarkt, "Predictive intelligent control of oxygenation (PRICO) in preterm infants on high flow nasal cannula support: A randomised cross-over study," *Arch. Disease Childhood-Fetal Neonatal Ed.*, vol. 106, no. 6, pp. 621–626, Nov. 2021.

[22] E. Kapici, E. Kutluay, and R. Izadi-Zamanabadi, "A novel intelligent control method for domestic refrigerators based on user behavior," *Int. J. Refrig.*, vol. 136, pp. 209–218, Apr. 2022.

[23] A. P. S. Subramanian, B. S. Sutha, and K. R. A. Britto, "Cohesive DS-PID and FQL control mechanisms to enhance the performance of the electric vehicle system," *Elektronika IR Elektrotechnika*, vol. 28, no. 2, pp. 46–58, Apr. 2022.

[24] X. Li, J. Chen, D. Zhou, and Q. Gu, "A modified biogeography-based optimization algorithm based on cloud theory for optimizing a fuzzy PID controller," *Optim. Control Appl. Methods*, vol. 43, no. 3, pp. 722–739, May 2022.

[25] V. S. Özsoy, "The determination of the most suitable inertia weight strategy for particle swarm optimization via the minimax mixed-integer linear programming model," *Eng. Comput.*, vol. 38, no. 4, pp. 1933–1954, Jun. 2021.

[26] I. Dagal, B. Akin, and E. Akboy, "Improved salp swarm algorithm based on particle swarm optimization for maximum power point tracking of optimal photovoltaic systems," *Int. J. Energy Res.*, vol. 46, no. 7, pp. 8742–8759, Jun. 2022.

[27] A.-H. H. Bacar and S. C. Rawhoudine, "An attractors-based particle swarm optimization for multiobjective capacitated vehicle routing problem," *RAIRO-Oper. Res.*, vol. 55, no. 5, pp. 2599–2614, Sep. 2021.

[28] B. Zhang, J. Song, S. Zhao, H. Jiang, J. Wei, and Y. Wang, "Prediction of yarn strength based on an expert weighted neural network optimized by particle swarm optimization," *Textile Res. J.*, vol. 91, nos. 23–24, pp. 2911–2924, Dec. 2021.

AIJUN KOU was born in Linyi, Shandong, China, in 1985. He received the master's degree from Jiangsu University, China. He is currently with the School of Electronic Information, Huzhou College. His research interests include artificial intelligence, machine learning, and computer vision.

XIAOJUN LI was born in Shuozhou, Shanxi, China, in 1987. He received the master's degree from Jiangsu University, China. He is currently with the School of Information Industry, Shanxi College of Technology. His research interests include artificial intelligence, machine learning, and computer vision.