

## Research Article

# Sliding Mode Control of Discrete Chaotic System Based on Multimodal Function Series Coupling

**Fengjun Hu**

*Institute of Information Technology, Zhejiang Shuren University, Hangzhou, Zhejiang 310014, China*

Correspondence should be addressed to Fengjun Hu; [jainism@msn.com](mailto:jainism@msn.com)

Received 9 December 2016; Revised 2 March 2017; Accepted 20 March 2017; Published 12 April 2017

Academic Editor: Eric Campos-Canton

Copyright © 2017 Fengjun Hu. 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.

A new sliding mode control model of discrete chaotic systems based on multimodal function series coupling is proposed to overcome the shortcomings of the standard PSO algorithm in multimodal function optimization. Firstly, a series coupled PSO algorithm (PP algorithm) based on multimodal function is constructed, which is optimized by multipeak solution on the basis of the standard PSO algorithm. Secondly, the improved PSO algorithm is applied to search all the extreme points in the feasible domain. Thirdly, the Powell method is used to perform the local optimization of the search results, and the newly generated extreme points are added to the extreme point database according to the same peak judgment operator. Finally, the long training time of PP algorithm can be overcome by the characteristics of fast convergence rate of the cloud mutation model. And also, both the population size and the redundancy can be reduced. Then, the clonal selection algorithm is used to keep the diversity of the population effectively. Simulation results of the sliding mode control of discrete chaotic systems show that the improved PSO algorithm obviously improves the response speed, overshoot, and so on.

## 1. Introduction

Chaotic system can be divided into integer order chaotic system and fractional chaotic system. Fractional chaotic system is a very complex nonlinear system [1]. Sliding mode control has become an important branch of modern control theory because of its good robustness and antijamming characteristics [2]. Fractional chaotic system sliding mode control, which can further improve the control performance of the system on the basis of traditional sliding mode control and become an important research area of modern nonlinear control.

Before introducing the sliding mode control of fractional order chaotic system, we mainly studied the control and synchronization of the fractional chaotic systems. At present, the control and synchronization of the fractional chaotic system are divided into two categories. One is the classical system stability theory and control method based on traditional integer order system. On the basis of the theory and control method, the stability analysis of the fractional order is carried

out to design the integer order controller [3]. The other is to utilize the characteristics of the fractional order system to put forward the fractional order controller design method; then, the control of the fractional order system can be realized [4]. The control methods can be divided into the following areas:

- (1) Feedback control based on pole placement of the fractional order system: Matignon [5] discussed the relationship between internal stability and external stability of the fractional differential equations. The system is linearized near the equilibrium point. And the author also described the characteristic equation of the system and the eigenvector based on the fractional calculus theory and the traditional algebraic polynomial, analyzed the relationship between the fractional system eigenvalues and the system matrix pole, and presented the sufficient conditions for the stability of the fractional order system.
- (2) The control of Gronwall inequality based on the fractional order: the Gronwall inequality is extensively applied to the integer order differential equation, and

the solution of the system is expressed analytically. The Gronwall inequality is taken to analyze the system solution to obtain the design of the controller, which is a common control design method for integer order systems. Lazarević and Spasić [6] applied the Gronwall inequality to the fractional time-delay system, analyzed the solution of the fractional order system, and realized the finite-time stabilization control of the fractional order. N'Doye and others [7] generalized the Bellman-Gronwall inequality, analyzed a class of nonlinear affine fractional order systems, and achieved the asymptotic stability of the fractional systems. Ye et al. [8] extended the integer order Gronwall inequality to the fractional systems, proposed the generalized Gronwall inequality, and successfully implemented the asymptotic stability control for the fractional time-delay systems.

- (3) Fractional order PID control: in the classical control theory and applications, PID controller has been developed very well and is widely applied to the actual system. In accordance with the PID control, the integral controller and differential controller of the PID controller can be partially extended to the fractional order; that is, the fractional integral and the fractional differential are introduced into the control design. Vasundhara Devi et al. [9] first proposed the fractional order PID controller and proved that the integer order PID controller belongs to a subset of the fractional PID controller. By adjusting the fractional order, the control effect of the fractional integral and fractional differential controller can be changed.
- (4) Feedback control based on linear matrix inequality: Sabatier and others [10] gave a sufficient and necessary condition for the stability of the fractional order time-invariant interval uncertain systems. Because of the state boundedness characteristic of chaotic systems, this condition has been applied to the control and synchronization of the fractional order chaotic systems. Chen and others [11] presented a sufficient and necessary condition for the fractional order systems with parametric perturbations and applied it to the fractional order chaotic systems to achieve robust synchronization of the fractional order chaotic systems. Dadras and Momeni [2] introduced the passive theory into the fractional order system and designed the fractional integral sliding surface by the linear matrix inequality to realize the passive sliding mode control of the time-varying uncertain fractional chaotic systems.
- (5) Sliding mode control of the fractional order system: the application of the integer order sliding mode to the fractional systems has been proved to be effective. The integral sliding mode surface and approach law are designed, respectively. And the effectiveness of the sliding surface is proved. The Lyapunov function is designed to analyze the stability of the fractional order system, and then, the sliding mode control law is obtained [12]. On the other hand, by combining

the design idea of the integral sliding mode surface with the characteristics of the fractional system, we can design the fractional sliding mode surface, and the integer integral of the sliding surface is changed to the fractional integral [13]. Similarly, the approximation of the integer order can be extended to the fractional systems; that is, the differential equation of integer order approximation law description can be generalized into the fractional order equation, which can realize the fractional order sliding mode approximation law [14].

In this paper, a PSO multiobjective optimization algorithm is introduced. Since the algorithm is easy to fall into the local optimization, a sliding mode control of discrete chaotic system based on multimodal function series coupling is proposed combining the characteristics of multimodal function optimization problems. Simulations show the effectiveness of this method.

## 2. Multimodal Function Optimization Problem Modeling

In many engineering optimization, such as complex system parameters and structural optimization, neural network weights, and structural optimization, we not only need to find the global optimal solution in the feasible region but also need to search several global optimal solutions and other valued local optimal solutions. Thus, multichoice and multi-information can be provided for decision makers. The above can be classified as multimodal function optimization problem or multipeak function optimization (MFO).

The multimodal optimization problem, as shown in (1), consists of two parts: the objective function and the search range.

$$\text{Minimize } f(x), \quad x = (x_1, x_2, \dots, x_n) \in S. \quad (1)$$

In (1), the search range  $S$  that can meet all the variables is called the feasible region, and the solution in feasible region is called the feasible solution. In the feasible region, the solution with the smallest objective function is the optimal solution. The optimal solution can be divided into global optimal solution and local optimal solution. In the multimodal function optimization, not only all the global optimal solutions of the optimization problem need to be found, but also the local optimal solutions need to be found as many as possible.

For the MFO problem of the objective function minimum value solution, the fitness function is defined as

$$\text{fit}(X) = f(X). \quad (2)$$

In (2),  $f(X)$  is the function value of  $X$ , and  $\text{fit}(X)$  is the fitness value.

For the MFO problem of the function maximum value solution, the limit construction method is used to solve the minimization problem. The fitness function is defined as

$$\text{fit}(X) = C_{\max} - f(X). \quad (3)$$

In (3),  $C_{\max}$  is the maximum estimate of  $f(X)$ .

From (2) and (3), we can see that the smaller the fitness value, the better the fitness.

### 3. Solution Algorithm of Multimodal Function Problem Based on Series Coupling

Based on PSO algorithm and Powell method, this paper presents PP algorithm based on series coupling. The PP algorithm has the maximum repetition search  $M$  ( $M > 1$ ) to avoid the situations in which not all the extreme values can be searched in an evolutionary iteration. In each iteration, firstly, the improved PSO algorithm is used to carry out the global search of all the extreme points in the feasible region. Secondly, Powell method is used to perform the local search towards the optimal solutions found by the improved PSO to improve the accuracy of the solution. Finally, the peak judgement operator adds the newly generated extreme points to the extreme point database and rerandomizes the initialization of the particles.

**3.1. PSO Algorithm Improvement.** In the case of the particle swarm with the  $N$  numbers of particles in the  $D$ -dimensional search space, denote the flying speed of particle  $i$  as  $V_i = (V_{i1}, V_{i2}, \dots, V_{iD})^T$ . The current position is  $X_i = (X_{i1}, X_{i2}, \dots, X_{iD})^T$ , the searched optimal position of particle  $i$  is  $P_i = \{P_{i1}, P_{i2}, \dots, P_{iD}\}$ , and the searched optimal position of the swarm is  $P_g = \{P_{g1}, P_{g2}, \dots, P_{gD}\}$ . In order to reduce the possibility of the particles leaving the search space during the process of evolutionary iterations, the speed is limited as  $V_{\min} \leq V_{id} \leq V_{\max}$ . The standard PSO is shown in

$$V_{id}(t+1) = \omega(t)V_{id}(t) + c_1 \text{rand}() (P_{id}(t) - X_{id}(t)) + c_2 \text{rand}() (P_{gd}(t) - X_{gd}(t)). \quad (4)$$

In (4),  $t$  is the current iteration number,  $V_{id}$  represents the  $d$ th dimensional component of the particle  $i$  velocity vector  $V$ , and  $1 \leq d \leq D$ .  $c_1$  represents the cognitive learning factor of the particle individual and  $c_2$  represents the social learning factor.  $\text{rand}()$  is the random number generated by the uniform distribution on  $[0, 1]$ .

In the multimodal optimization, in order to make each local optimal value become the “only way” of some particles in PSO algorithm, this paper improves the standard PSO and obtains the improved PSO.

- (1) Ideally, the self-cognition of each individual in the particle swarm represents each local optimization of the multipeak optimization. Thus, let  $c_2 = 0$ ; we can obtain not just the global optimal values, but all the local optimal values.
- (2) Consider that the random function  $\text{rand}()$  in (4) mainly increases the randomness of the particle motion so as to make the local optimal point jump out. In order to let each particle of the particle swarm converge to each local optimal value as soon as possible,  $\text{rand}()$  in (4) should be removed.

- (3) The linear decreasing strategy of inertia weight is simple and intuitive and has better searching ability, which makes the algorithm balance between the global search and the local search. Therefore,  $\omega$  uses the above strategy, which is shown in

$$\omega(t) = \omega_{\text{start}} - (\omega_{\text{start}} - \omega_{\text{end}}) \frac{t}{t_{\text{max}}}. \quad (5)$$

In (5),  $t$  represents the current PSO iteration algebra,  $t_{\text{max}}$  represents PSO maximum iteration number, and  $\omega_{\text{start}}$  and  $\omega_{\text{end}}$  represent the maximum and the minimum value of  $\omega$ .

Based on the above improvements, (4) can be changed to

$$V_{id}(t+1) = \omega(t)V_{id}(t) + c_1 (P_{id}(t) - X_{id}(t)). \quad (6)$$

The position change is performed according to (7). In order to avoid the particles leaving from the search space during the process of evolution iteration, the following condition should be satisfied:  $X_{\min} \leq X_{id} \leq X_{\max}$ .

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1). \quad (7)$$

**3.2. Nonlinear Powell Direct Search Method.** The nonlinear Powell direct search method is a nonlinear direct local search method for solving unconstrained optimization problems without using derivatives. It is considered to be a relatively effective method in direct search. Powell search method is known as the direction acceleration method as well. Using this method, the calculation of the derivative is not required and only the function value is used to carry out the one-dimensional search from one point towards two directions. Then, the minimal value can be obtained. Nonlinearity means that the search direction of the initial point is not fixed, the acceleration direction will change with the change of the initial point position, and the search is not along a straight line.

The steps of the Powell search method are as follows:

#### Begin

**Step 1.** The particle of the last search generation of PSO is taken as the initial point  $x^{(0)} \in S$ . Set the search precision of the Powell method as  $\varepsilon$ .  $D$  numbers of initial linearly independent search directions  $d^{(i)}$  are given, and  $i = 0, 1, \dots, D-1$ . Set  $k = 0$ .

**Step 2.** According to (9) below, one-dimensional search is performed accurately from point  $x^{(0)}$  to each search direction in turn  $d^{(0)}, d^{(1)}, \dots, d^{(D-1)}$ . Thus,  $x^{(0)}, x^{(1)}, \dots, x^{(D)}$  is obtained.

$$f(x^{(i)} + \alpha_i d^{(i)}) = \min_{\alpha \in S} f(x^{(i)} + \alpha d^{(i)}), \quad (8)$$

$$x^{(i+1)} = x^{(i)} + \alpha_i d^{(i)}. \quad (9)$$

In (8),  $\alpha$  and  $\alpha_i$  represent the step sizes and  $\alpha_i$  is obtained by the exact linear search; that is,  $\alpha_i$  is the solution of the one-dimensional optimization problem above.  $\alpha_i$  can be negative, which indicates that the exact linear search takes place over the entire real axis.

*Step 3.* Let  $d^{(D)} = x^{(D)} - x^{(0)}$ . If  $\|d^{(D)}\| \leq \varepsilon$ , and the solution  $x^{(D)}$  is obtained, calculation ends. Otherwise, the exact linear search is performed from point  $x^{(D)}$  to the direction of  $d^{(D)}$  to get  $x^{(D+1)}$ .

*Step 4.* According to (10), the indicator  $tl$  determined by the maximal decrease is calculated.

$$\begin{aligned} & f(x^{(tl)}) - f(x^{(tl+1)}) \\ &= \max_{0 \leq i \leq D-1} \{f(x^{(i)}) - f(x^{(i+1)})\}. \end{aligned} \quad (10)$$

*Step 5.* When (11) holds, it shows that  $d^{(0)}, d^{(1)}, \dots, d^{(D-1)}$  are still linearly independent, and the search directions of the next round are still  $d^{(0)}, d^{(1)}, \dots, d^{(D-1)}$ . Then,  $x^{(0)} = x^{(D+1)}$ . Return to Step 2.

$$\begin{aligned} & f(x^{(0)}) - 2f(x^{(D)}) + f(2x^{(D)} - x^{(0)}) \\ & \geq 2(f(x^{(tl)}) - f(x^{(tl+1)})). \end{aligned} \quad (11)$$

*Step 6.* When (11) does not hold, it shows that  $d^{(0)}, d^{(1)}, \dots, d^{(D-1)}$  are linearly dependent. Thus, set  $d^{(tl+1)} = d^{(tl+i+1)}$  to ensure linearly independence of the new generated search direction,  $x^{(0)} = x^{(D+1)}, k = k + 1$ . Return to Step 2.

**End**

**3.3. The Same Peak Judgement of the Extreme Points.** The extreme points found by the PP algorithm are compared with all the extreme points in the extreme point database EP in turn and make the same peak judgement. If it is judged as the same peak and greater than the fitness value of the existing extreme point, the extreme point is discarded. If it is judged as the same peak but less than the fitness value of the extreme point, the extreme point is replaced. And if it is judged to be different peak, it is added to set EP.

The same peak judgement operator uses the peak exploration method to perform the same peak judgement towards the global extreme points. In this paper, the peak exploration method is applied in the same peak judgement of all the global and local extreme points. The method is as follows.

Set  $A$  as an extreme point in EP and  $B$  as the extreme point found by the PP algorithm.  $(h-1)$  points are inserted between  $A$  and  $B$ , and  $AB$  is divided into  $h$  equal parts. The same peak operators of  $h$  parts are calculated, which is shown as

$$Tf(i) = \text{fit}(X_{i+1}) - \text{fit}(X_i). \quad (12)$$

In (12),  $X_{i+1}$  and  $X_i$  are two adjacent points between  $A$  and  $B$ ,  $1 \leq i \leq h$ . When  $i = 1$ ,  $X_1$  is point  $A$ , and when  $i = h$ ,  $X_{h+1}$  is point  $B$ . If value of  $Tf$  changes from positive to negative, it means that there is a mountain between points  $A$  and  $B$ . Then,  $A$  and  $B$  are different peak extreme points. Otherwise, they are the same peak extreme points.

**3.4. Performance Analysis of PP Algorithm in Multimodal Function Optimization Problem.** Let the numbers of particles

TABLE 1: Comparisons of PP algorithm with other evolutionary algorithms.

Method	Number	$\bar{X}$		Mean <sub>G</sub>	Time
		$x_1$	$x_2$		
ES	1	-2.9135	3.2658	648.9996	11.2
	2	3.2584	-1.5894	648.9996	
	3	-3.4584	-3.5814	648.9994	
	4	2.8595	2.1247	648.9991	
PSO	1	-2.9251	3.2671	648.9997	—
	2	3.2597	-1.5902	648.9998	
	3	3.4595	-3.5877	648.9998	
	4	2.8608	2.1211	648.9995	
PP	1	-2.9295	3.2613	650	2.5
	2	3.2657	-1.5945	650	
	3	3.4624	-3.5898	650	
	4	2.8695	2.1202	650	

$N$  be 20~105. Set the acceleration factor  $c_1 = 1.49445$ . Set the inertia weight:  $\omega_{\text{start}} = 0.9$ ,  $\omega_{\text{end}} = 0.4$ , and  $t_{\text{max}} = 30$ , respectively. The maximal repeated search number is  $M = 2\sim 10$ , and  $\varepsilon = 2 \times 10^{-8}$ . Select  $f_1$  as the test function; the comparison of the results of the PP algorithm, multipopulation coevolution strategy ES, and the standard PSO algorithm is shown in Table 1.

$$f_1(x) = 660 - (x_1^2 + x_2 - 11)^2 - (x_1 + x_2^2 - 7)^2. \quad (13)$$

In Table 1,  $\bar{X}$  represents the mean value of the optimal solution location at the time of the test. Mean<sub>G</sub> is the average of the global solutions of the 50 tests.

From Table 1, we can see that, compared with ES and the standard PSO, the PP algorithm can search all four global extreme points of the function  $f_1$  in 50 tests, and also there are no redundant or missing extreme points. The solution quality is significantly improved. Therefore, the PP algorithm has obvious advantages in search precision.

For function  $f_1$ , the standard PSO has no mean computing time, and we use “—” to express time. Time of the ES is 11.2 seconds, and time of the PP algorithm is 2.5 seconds. Thus, the PP algorithm has obvious advantages in the speed of convergence.

Then, the global convergence of PP algorithm is analyzed.

*Suppose.* (1) Definition domain  $\Omega$  of the multimodal function optimization problem is a bounded closed region of  $R^n$ . (2) The objective function  $f(X)$  is a continuous function on the region  $\Omega$ .

**Theorem 1.** Set  $\{X(k)\}$  ( $k$  is the cumulative iteration algebra) as the population sequence generated by the PP algorithm, where  $x^*(k) \in X(k)$  is the optimal point in the  $k$ th generation of the population; that is,  $x^*(k) = \arg \min_{x \in \Omega} f(x(k))$ . If the objective function and the feasible region of problem  $P$  satisfy hypothesis 1,  $p\{\lim_{k \rightarrow \infty} f(x^*(k)) = f^*\} = 1$  holds, where  $(f^* = \min\{f(x) : x \in \Omega\})$ ; that is, the population sequence converges to the global optimal solution with probability 1.

*Demonstration.* Define the cumulative iteration algebra  $k$  as  $k = t_{\max}(m - 1) + t$ , where  $m$  is the repeated search algebra ( $1 \leq m \leq M$ ),  $t_{\max}$  is the maximum iteration algebra, and  $t$  is the iterative algebra.

The change of the displacement of the two generations before and after the PP algorithm is  $\Delta X_{id}(t) = \omega(t)V_{id}(t) + c_1(P_{id}(t) - X_{id}(t))$ . Thus, when the population particles converge, the social cognitive part ( $P_{id}(t) - X_{id}(t)$ ) tends to 0 and less than 1, the particle velocity will drop rapidly to 0, and the particle swarm will stop moving. Therefore, the population particles may converge on the local solution and not on the global solution. For this reason, when the population particles fall into the evolutionary stagnation state, the particles are subjected to the mutation operation through the basic normal cloud generator. And the basic normal cloud generator based on the cloud model is subject to normal distribution.

Due to the mutation operation of the basic normal cloud generator with normal distribution, the displacement of particle  $X_i$  is subject to normal distribution; that is,  $\Delta X_i(t) \sim N(0, \sigma_i^2)$ , where  $N(0, \sigma_i^2)$  represents the normal distribution with average value 0 and variance  $\sigma_i^2$ .

Therefore, we can conclude that Theorem 1 holds.

#### 4. Optimization of PP Algorithm Based on Cloud Mutation Clonal Selection

In order to solve the problems of the multimodal function optimization, based on the above improved methods, this paper utilizes the characteristics of fast convergent speed of the cloud mutation model to compensate the shortcomings of fast training time of the PP algorithm. Meanwhile, it can reduce the population size and redundancy. And the clonal selection algorithm can effectively maintain the diversity of population. Hence, a particle swarm optimization algorithm based on cloud mutation clonal selection is proposed (WCPP).

*4.1. Optimization of Particle Space Distribution Based on Cloud Mutation.* The cloud model has the characteristics of randomness and stability tendency. The randomness can keep the individual diversity so as to search for more local extreme points, and the stable tendency can protect the better individual to perform adaptive localization of the global optimal values. Thus, in order to improve the accuracy of the PP algorithm and expand the search range to find other extreme points, a cloud mutation operator based on cloud model is introduced on the basis of the PP algorithm to update the particles.

Thresholds  $\varepsilon$  and  $t_N$  are given. When a particle satisfies (14), that is, the particle degenerates and degradation amplitude is smaller than  $\varepsilon$ , it is considered that the particle has a generation in the evolutionary stagnation state.

$$0 < \text{fit}(X_i(t)) - \text{fit}(X_i(t-1)) < \varepsilon, \quad (14)$$

$$\varepsilon = \text{abs}\left(\frac{\text{fit}(X_i(t))}{a}\right). \quad (15)$$

In (15),  $\text{fit}(X_i(t))$  represents the fitness value of  $X_i$  in the  $t$ th generation.  $a$  is a constant. When  $\text{fit}(X_i(t)) - \text{fit}(X_i(t-1)) < 0$ , it indicates the evolution of particle  $X_i$ . And when  $\text{fit}(X_i(t)) - \text{fit}(X_i(t-1)) > 0$ , it indicates the degradation of particle  $X_i$ .

The cloud mutation condition can be defined as the degradation of a particle of  $t_N$  successive generations and the degradation amplitude is less than  $\varepsilon$ . When a particle satisfies the condition of cloud mutation, mutation operation is performed to the particle by a basic normal cloud generator.  $t_N$  is set as 5 here. Since  $t_N$  is set too small, the mutation occurs too frequently and it is easy to miss the extreme point. On the other hand, the convergent speed of the algorithm can slow down.

One-dimensional normal cloud operator can be extended to  $D$  dimensions. And the cloud mutation operator for each dimension of  $X_i$  is as follows:

**Begin**

**For**  $d = 1 : D$

$Ex = X_{id}$ ;

$En = \text{variable search range}/cc1$ ;

$cc1 = 20 \text{ sqrt}(t)$ ;

$He = En/cc2$ ;

$cc2 = 10$ ;

A normal random number  $x_i$  is generated by the basic normal cloud generator, at the same time a random number  $\text{Temp}$  is also generated. When  $\mu_i > \text{Temp}$ ,  $X_{id}$  is updated by  $x_i$ .

**End For**

**End**

$En$  represents the horizontal width of the cloud, and the greater the horizontal width, the greater the scope of the particle search. When  $cc1$  is set as  $20 \text{ sqrt}(t)$ , the search range of the initial evolution is very large, which is helpful to find more extreme points. With the dynamic narrowing of the search range of  $t$ , it is favorable to the fine search of particles.

“Dimensional catastrophe problem” exists in dimension function. As the dimension increases, it is difficult to find the optimal solution in each dimension through the general evolutionary algorithms. The abnormal values on individual dimension lead to poor quality of the final solution or an inability to find the optimal solution to the optimization problem. Thus, in order to prevent the particles from falling into the local extreme points and failing to find the global optimal solution in high-dimensional multimodal function, the search particles are reinitialized with a certain probability  $p_s$  to find better solutions. When in the search state, the particles are rerandomly distributed in the search space as shown in

$$V_k(t+1) = \frac{(V_{\min} + V_{\max})}{2} + (V_{\min} - V_{\max})(\text{rand}() - 0.5),$$

$$X_k(t+1) = \frac{(X_{\min} + X_{\max})}{2} + (X_{\max} - X_{\min})(\text{rand}() - 0.5) \text{lk}. \quad (16)$$

After the initialization, particles are updated according to (16).

**4.2. Population Diversity Optimization Based on Clonal Selection Algorithm.** In order to ensure the diversity of the population, optimization is performed based on the clonal selection algorithm. The steps are divided into clonal amplification, adaptive wavelet mutation, and immune selection.

**(1) Clonal Amplification.** In the PSO immune system, the problem to be solved is the antigen. Each particle in the particle swarm has the potential to represent each local optimization value in the multipeak optimization. Therefore,  $N$  particles searched in the last iteration of the PP algorithm are all used as antibodies. The affinity function of antibodies and antigens uses the fitness function of the PP algorithm; that is,

$$\text{affinity}(a_i) = \text{fit}(a_i). \quad (17)$$

In (17),  $\text{fit}(a_i)$  is the fitness value of particle  $a_i$  and  $\text{affinity}(a_i)$  is the affinity value of the antibody  $a_i$  (the particle  $X_i$ ). From (17), we can see that the smaller the affinity value of the antibody, the better the affinity.

The antibodies in the antibody population were sorted in ascending order according to the affinity value before the clonal amplification. The clonal amplification operator for the sorted  $i$ th antibody  $a_i$  is calculated according to

$$\beta_i = \text{round}\left(a_c \frac{1}{\sqrt{i}} + b\right). \quad (18)$$

In (18),  $a_c$  is a constant greater than 1,  $\text{round}()$  is a round function, and  $b$  can ensure that each antibody has a constant larger than 1 with a certain number of clones. From (18), we can see that the higher the affinity, the greater the clonal multiple. In the clonal selection algorithm, when the current iteration number is  $ct$ , the  $i$ th cloned amplified antibody  $a_i$  is changed to  $a_i(ct) = \{a_{i1}(ct), a_{i2}(ct), \dots, a_{i\beta_i}(ct)\}$  after  $\beta_i$  times of clone.

**(2) Adaptive Wavelet Mutation.** Morlet wavelet has better probability distribution of upper and lower wings. Thus, Morlet wavelet is applied to immune mutation operation to make CSA have a wider range of searching, higher precision, and good regulating performance. Morlet wavelet mother function is shown in

$$\Psi_{a,b}(x) = \frac{1}{\sqrt{a}} e^{-((x-b)/a)/2} \cos\left(5 \frac{x-b}{a}\right). \quad (19)$$

When  $a = 1$  and  $b = 0$ , Morlet wavelet function is shown in Figure 1. In Figure 1, the horizontal axis is the independent variable, and the vertical axis is the wavelet mother function.

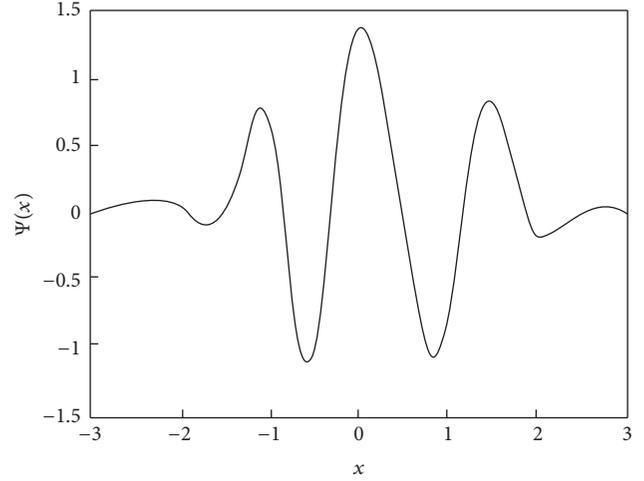


FIGURE 1: Morlet wavelet function.

With the progress of  $ct$ , the affinity of the antibody increases constantly. When the adaptive dynamic of the antibody mutation probability  $p_m$  reduces, the convergent speed of the algorithm can be improved.  $p_m$  is shown as

$$p_m(ct) = p_m(ct-1) \left(1 - 0.01 \frac{ct}{ct_{\max}}\right). \quad (20)$$

In (20),  $ct_{\max}$  is the maximum iteration number. When  $ct = 1$ , the initial value is set as 0.7.  $X_{id}$  mutation uses adaptive wavelet mutation operator based on the time-varying scale.  $X_{id}$  mutation formula is shown as follows:

$$X'_{id} = \begin{cases} X_{id} + \frac{X_{\max} - X_{id}}{m} \eta(ct) \sigma(ct), & \sigma(ct) \geq 0 \\ X_{id} - \frac{X_{id} - X_{\min}}{m} \eta(ct) \sigma(ct), & \text{else,} \end{cases} \quad (21)$$

where  $m$  is an increase constant.

$$\eta(ct) = 1 - r^{(1-ct/ct_{\max})^b}. \quad (22)$$

In (22),  $\eta(ct)$  is the time-based mutation scale,  $r$  is the random number between  $[0, 1]$ , and  $b$  is the system parameter, which determines the dependency degree of the random number disturbance on  $ct$ .

$$\sigma(ct) = \frac{1}{\sqrt{a(ct)}} e^{-(\varphi/a(ct))^2/2} \cos\left(5 \frac{\varphi}{a(ct)}\right). \quad (23)$$

In (23),  $\sigma(ct)$  is the wavelet mutation function, and  $\varphi$  is a random number between  $[-3a(ct), 3a(ct)]$ . Function  $a(ct)$  is shown as

$$a(ct) = e^{-\ln(g)(1-ct/ct_{\max})^c + \ln(g)}. \quad (24)$$

In (24),  $c$  is the shape parameter of  $a(ct)$  and is set to 1.  $g$  is the upper limit of  $a(ct)$  and is set to 10000. Thus, the value of  $a(ct)$  increases with the increase of  $ct$  between 1 and 10000, while the range of  $\sigma(ct)$  shrinks with the increase of  $ct$ .

TABLE 2: Comparing the results of PP algorithm with WCPP algorithm.

Method	Number	Mean <sub>G</sub>	Mean <sub>L</sub>	Time
PP	1	3.5216	2.7526	91.5
	2	3.5259	2.7568	
	3	3.5951	2.8692	
	4	3.6059	2.8925	
WCPP	1	3.4253	2.2682	50.8
	2	3.4158	2.2481	
	3	3.4682	2.2543	
	4	3.5060	2.4988	

From (21), we can see that, at the early stage of the evolution, for smaller  $r$ ,  $\eta(\text{ct}) \approx 1$ . When the value of  $\sigma(\text{ct})$  is large, the mutation space is also large, so that the probability of searching other extreme points can be increased. And in the middle and late stages of the evolution, adaptive of  $\eta(\text{ct})$  is smaller. When  $\text{ct}$  is approaching  $\text{ct}_{\max}$ ,  $\eta(\text{ct}) \approx 0$ , and the value of  $\sigma(\text{ct})$  is relatively small. Thus, by improving the local fine-tuning ability, the precision of the algorithm can be improved effectively.

(3) *Immune Selection.* Let  $S_i(\text{ct}) = T_s^c(a_i(\text{ct})) = \min\{\text{affinity}(a'_i(\text{ct}))\}$ , and  $S_i(\text{ct}) \cup a_i(\text{ct}) \rightarrow a_i(\text{ct} + 1)$ .  $a'_i(\text{ct})$  represents an antibody obtained by clonal amplification and mutation of the  $\text{ct}$ th iteration. Compression of population antibody is achieved by local selection, and the optimal solution of population is not deteriorated.

4.3. *Performance Analysis of WCPP Algorithm in Multimodal Function Optimization.* Select  $f_2$  as test function, the result comparison of the PP algorithm and the WCPP algorithm is shown in Table 2.

$$f_2(x) = x \sin(10\pi x) + 2. \quad (25)$$

In Table 2,  $\bar{X}$  represents the mean value of the optimal solution location at the time of the test. Mean<sub>G</sub> represents the average of the global solutions of the 50 tests.

The mean value Mean<sub>L</sub> of the local solution of the PP algorithm is larger than that Mean<sub>L</sub> of the WCPP algorithm, which shows that the local solution of the PP algorithm has higher accuracy than the WCPP algorithm in the case that 16 local extreme points can be searched for objective function  $f_2$ .

## 5. Application of WCPP Algorithm to Sliding Mode Control of Discrete Chaotic Systems

5.1. *Fractional Order Chaotic System Sliding Mode Control Model.* Suppose a control system as  $\dot{x} = f(x, u, t)$ ,  $x \in R^n$ ,  $u \in R^n$ ,  $t \in R$ . Switching function needs to be determined,  $s(x)$ ,  $s \in R^m$ . Then, the control function can be solved as follows:

$$u = \{u^+(x), s(x) > 0; u^-(x), s(x) < 0\}. \quad (26)$$

In (26),  $u^+(x) \neq u^-(x)$ . Then,

- (1) the sliding mode exists: that is, (26) holds;
- (2) the accessibility condition can be satisfied, and the movement points outside the switching surface will reach the switching surface with a limited time;
- (3) the stability of the sliding mode movement is ensured;
- (4) the dynamic quality requirement of the control system is satisfied.

If the first three basic problems can be satisfied, we can call the control as sliding mode control.

Henon fractional chaotic system is taken as an example; the sliding mode control model of fractional chaotic system is constructed. The Henon chaotic system with control is shown as

$$\begin{aligned} x_1(k+1) &= 1 - a(x_1(k))^2 + bx_2(k) + u(k), \\ x_2(k+1) &= x_1(k), \end{aligned} \quad (27)$$

where  $x \in R^n$ ,  $u \in R$ .

Define the switching function as  $s(k) = C_e[e(k), de(k)]$ , where  $C_e = [c_k, c_0]$ . Then,  $s(k+1) = C_e[e(k+1), de(k+1)]$ ,  $de(k+1) = (e(k+1) - e(k))/ts$ , where  $ts$  is the sampling time. And we can obtain (28) as follows:

$$\begin{aligned} s(k) &= c_k e(k) + c_0 de(k), \\ s(k+1) &= \left(c_k + \frac{c_0}{ts}\right) e(k+1) - \left(\frac{c_0}{ts}\right) e(k). \end{aligned} \quad (28)$$

When the fractional order chaotic system enters the ideal sliding mode,  $s(k)$  can satisfy the following:

$$s(k+1) = s(k). \quad (29)$$

That is,

$$\left(c_k + \frac{c_0}{ts}\right) e(k+1) - \left(\frac{c_0}{ts}\right) e(k) = c_k e(k) + c_0 de(k), \quad (30)$$

where  $e(k+1) = r(k+1) - x_1(k+1)$ . Define  $f(k) = 1 - ax_1^2(k) + bx_2(k)$ ; then, we can have  $u_{\text{eq}}(k) = x_1(k+1) - f(k)$ .

From the above, we can obtain the sliding mode control equivalent control part of the fractional order chaotic system  $u_{\text{eq}}(k)$ , which is shown in

$$\begin{aligned} u_{\text{eq}}(k) &= \frac{(c_k + c_0/ts)r(k+1) - (c_k + c_0/ts)e(k) - c_0 de(k)}{c_k + c_0/ts}. \end{aligned} \quad (31)$$

5.2. *Sliding Mode Control Flow of the Discrete Chaotic System Based on WCPP Algorithm*

(1) *Evaluation Function Selection.* The selection of the evaluation function not only needs to consider the stability, fastness, and accuracy of the system but also needs to consider the control energy problem, the form of which is shown as

$$J(p) = \int_0^{\infty} (w_1 |e(t)| + w_2 |u(t)| + w_3 t_u) dt. \quad (32)$$

In (32),  $e(t)$  is the system error,  $u(t)$  is the control variable,  $t_u$  is the initial step time, and  $w_1, w_2, w_3$  are the weights. The smaller the value of the evaluation function  $J(p)$  is, the closer the corresponding function  $p$  is to the global optimal solution. The evaluation function value decreases as the algorithm runs.

(2) *Sliding Mode Control Flow of the Discrete Chaotic System Based on WCPP Algorithm.* By using the WCPP algorithm, the parameters of the sliding mode controller are globally optimized to construct the sliding mode control method. The flow is as follows:

### Begin

*Step 1.* Initialize the sliding mode controller, other parameters, and a group of particles with size  $N$ . The initialization parameters include maximum repetition search number  $M$  of hybrid algorithm and the WCPP maximum number of iterations  $ct_{\max}$ .

*Step 2.* The evaluation function value of each particle is obtained by control of the discrete chaotic system according to the position of each particle and the optimal position of individual particle  $P_i$  and the optimal position of all particles  $P_g$  are recorded.

*Step 3.* If the repeated search algebra  $m$  is larger than  $M$ , the iteration is stopped, and the algorithm is finished. The output of the sliding mode controller controls the discrete chaotic system. Otherwise, turn to Step 4.

*Step 4.* If the WCPP algorithm iteration number  $t \leq t_{\max}$ ,

If the condition of the cloud mutation is satisfied,  
the cloud mutation is performed on the particles.

**Else**  
Speed and positions are updated on the particles.

**End If**  
Update particles  $P_i$  and  $P_g$ , and  
 $t + 1$ .

**End If**

*Step 5.* The particles in the last search generation of the WCPP algorithm are cloned and amplified.

Adaptive wavelet mutation operator performance is carried out.

Immune selection performance is carried out.

And,  $ct + 1$ .

### End If

update particles  $P_i$  and  $P_g$ ,

*Step 6.*  $m + 1$ , the positions and the speed are reinitialized.

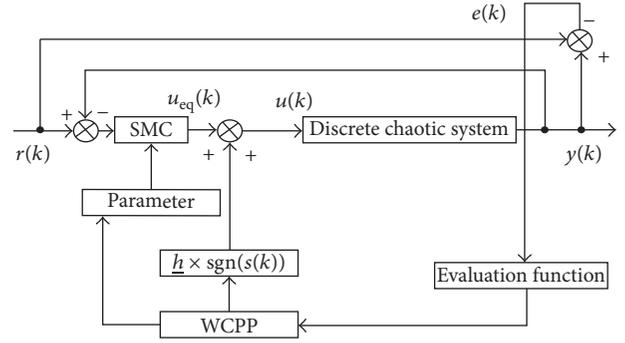


FIGURE 2: Sliding mode control of discrete chaotic system based on WCPP algorithm.

*Step 7.* The searched optimal fitness particles are assigned to the initialized particles and turn to Step 2.

### End

In order to improve the fastness of the sliding mode control method for the discrete chaotic systems based on the WCPP algorithm, the values of the algorithm parameters  $N$  and  $M$  are small.

Figure 2 is the discrete chaotic system sliding mode control structure based on the WCPP algorithm.

*5.3. Algorithm Simulations.* A discrete chaotic system is taken as the simulation object; the model of the controlled object is shown as

$$y(k+1) = \frac{y(k)}{1 + y(k)^2 + d(k)} + u(k)^3. \quad (33)$$

For the above discrete chaotic system, simulations of step input signal  $r(k) = 1$  are implemented based on the WCPP and other two algorithms, which is shown in Figure 3. In this simulation, the disturbance signal is set as  $d(k) = 0$ . And at this time the WCPP optimized parameters are  $\{20, 1\}$ .

From Figure 3, we can see that the WCPP algorithm has very strong search ability. It can search the global optimal value of the sliding mode controller parameters without overshoot, and the error, the response speed, and adjustment time all achieved the optimal performance. Therefore, the control effect of the WCPP algorithm is superior to the control effect of the PSO and the PP algorithm.

For the above discrete chaotic system, the simulations of the square wave input signal  $r(k) = 0.5 \text{ sign}(\sin(8\pi k \cdot ts))$  are implemented by the WCPP algorithm and other two algorithms, respectively, which is shown in Figure 4.

Another discrete chaotic system is also taken as the simulation object; the model of the controlled object is shown as

$$y(k) = \frac{-0.1y(k-1) + u(k-1)}{1 + y^2(k-1)}. \quad (34)$$

For the above discrete chaotic system, simulations of step input signal  $r(k) = 1$  are implemented based on the WCPP

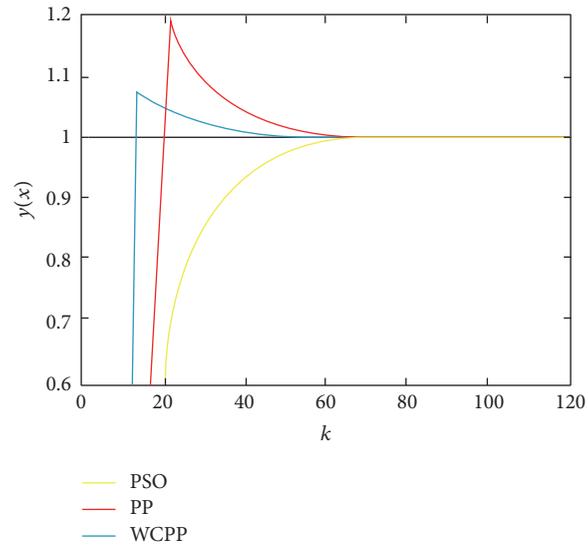


FIGURE 3: Simulation results of discrete chaotic systems.

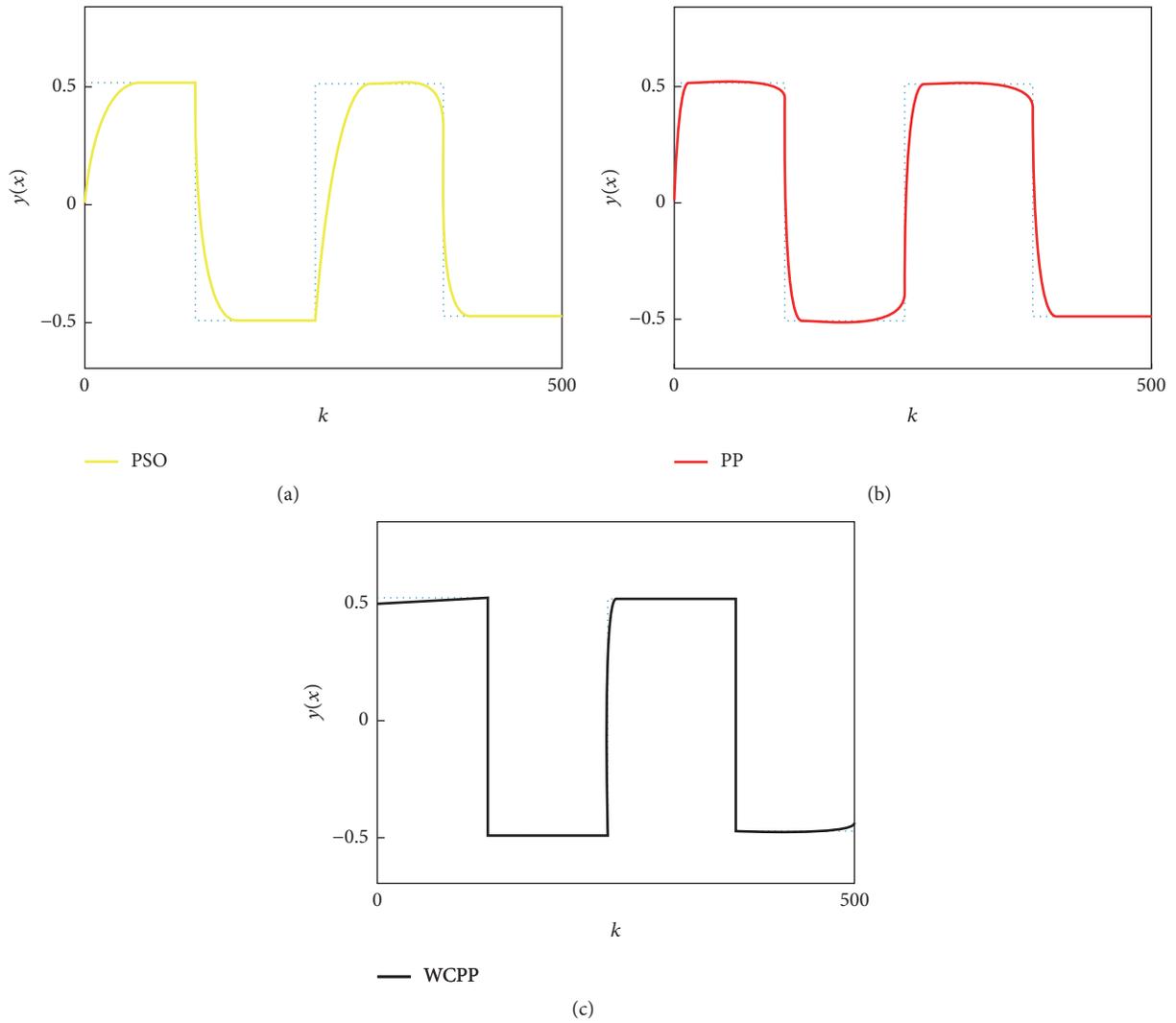


FIGURE 4: Square wave simulation of discrete chaotic system.

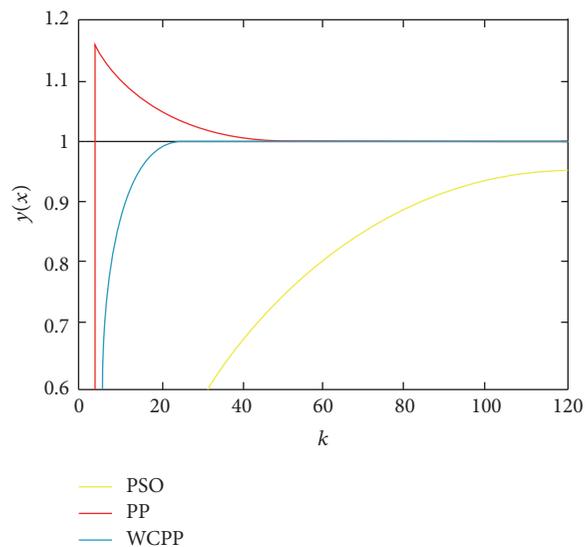


FIGURE 5: Simulation results of discrete chaotic systems.

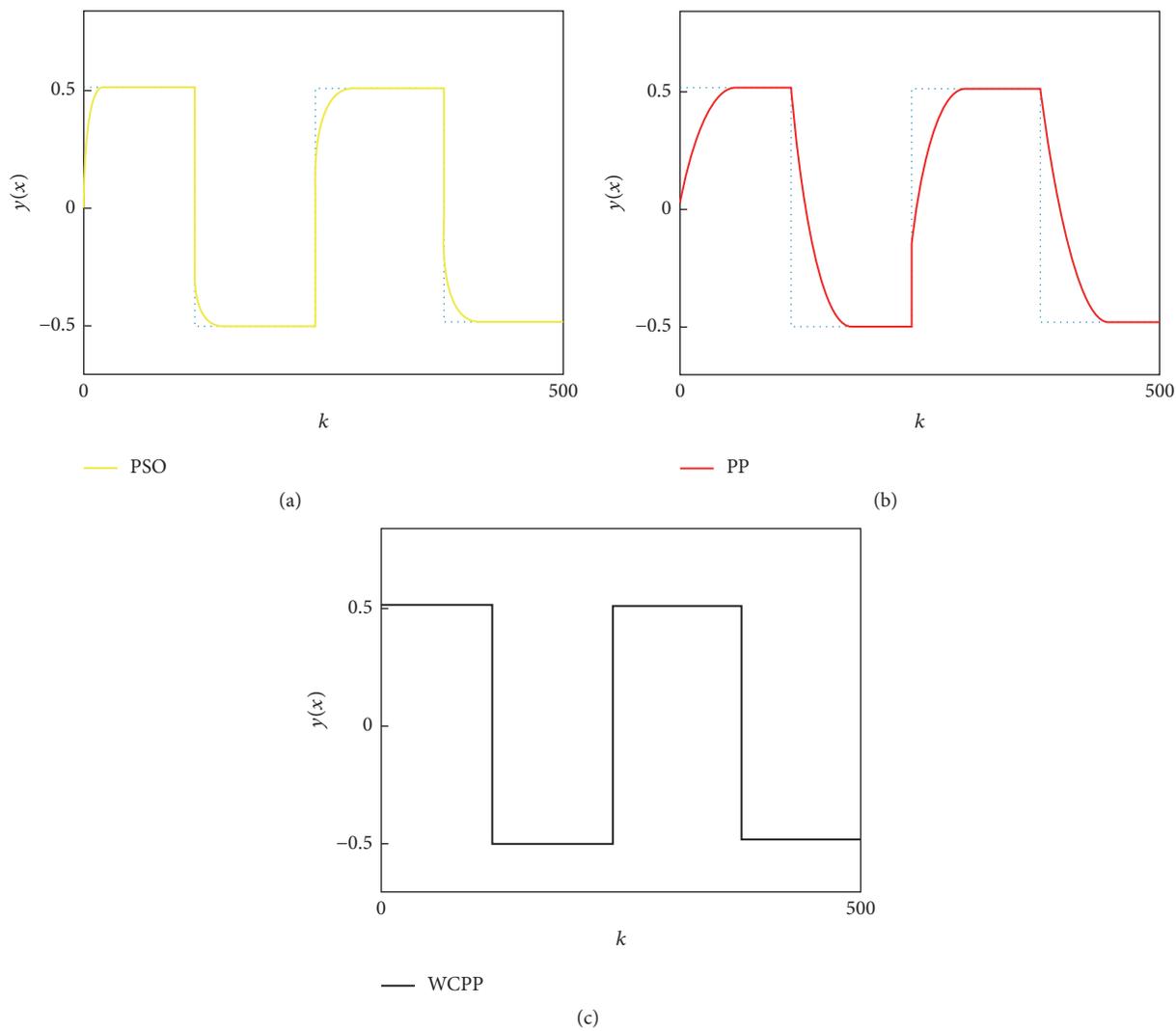


FIGURE 6: Square wave simulation of discrete chaotic system.

and other two algorithms, which is shown in Figure 5. The WCPP optimized parameters are  $\{1.000, 0.001\}$ .

From Figure 5, we can find that the WCPP algorithm is basically no overshoot and is able to achieve the desired value in the second iteration. The performances of the WCPP algorithm in error, response speed, and adjustment time are significantly better than those of PSO and PP algorithms. Therefore, WCPP algorithm has strong searching ability and good control quality.

For the above discrete chaotic system, the simulations of the square wave input signal  $r(k) = 0.5 \text{ sign}(\sin(8\pi k \cdot ts))$  are implemented by the WCPP algorithm and other two algorithms, respectively, which is shown in Figure 6.

It can be seen from Figures 4 and 6 that the response speed, overshoot, and other performances of the WCPP algorithm have been improved significantly.

## 6. Conclusions

Chaos and chaos systems are widely used in physics, engineering, biology, finance, and so forth. The modeling of complex systems based on the theory of calculus can more accurately reflect the dynamic characteristics of the systems. Discrete chaotic systems can be widely used in the field of chaotic secure communications. Thus, it is of great theoretical and applicable value in the applications of control and synchronization of the chaotic dynamic systems. In order to overcome the shortcoming of the standard PSO algorithm in multimodal function optimization problems, a sliding mode control model of discrete chaotic systems based on the coupled multimode functions is proposed. The simulation results show that the improved algorithm proposed in this paper has significantly improved the response speed and overshoot performance compared with the standard PSO algorithm.

## Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 51675490).

## References

- [1] Y.-H. Chang, C.-I. Wu, H.-W. Lin, H.-C. Chen, and C.-W. Chang, "Fractional order integral sliding-mode flux observer for direct field-oriented induction machines," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 7, pp. 4851–4868, 2012.
- [2] S. Dadras and H. R. Momeni, "Fractional terminal sliding mode control design for a class of dynamical systems with uncertainty," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 1, pp. 367–377, 2012.
- [3] A. H. J. De Ruiter, "Adaptive spacecraft attitude control with actuator saturation," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 5, pp. 1692–1696, 2010.
- [4] H. Liu, L. Guo, and Y. Zhang, "Composite attitude control for flexible spacecraft with simultaneous disturbance attenuation and rejection performance," *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 226, no. 2, pp. 154–161, 2012.
- [5] D. Matignon, "Stability properties for generalized fractional differential systems," *ESAIM: Proceedings and Surveys*, vol. 5, pp. 145–158, 2008.
- [6] M. P. Lazarević and A. M. Spasić, "Finite-time stability analysis of fractional order time-delay systems: Gronwall's approach," *Mathematical and Computer Modelling*, vol. 49, no. 3–4, pp. 475–481, 2009.
- [7] I. N'Doye, M. Zasadzinski, M. Darouach, and N.-E. Radhy, "Observer-based control for fractional-order continuous-time systems," in *Proceedings of the 48th IEEE Conference on Decision and Control held jointly with 28th Chinese Control Conference (CDC/CCC '09)*, pp. 1932–1937, December 2009.
- [8] H. Ye, J. Gao, and Y. Ding, "A generalized Gronwall inequality and its application to a fractional differential equation," *Journal of Mathematical Analysis and Applications*, vol. 328, no. 2, pp. 1075–1081, 2007.
- [9] J. Vasundhara Devi, F. A. Mc Rae, and Z. Drici, "Variational Lyapunov method for fractional differential equations," *Computers & Mathematics with Applications*, vol. 64, no. 10, pp. 2982–2989, 2012.
- [10] J. Sabatier, M. Moze, and C. Farges, "LMI stability conditions for fractional order systems," *Computers & Mathematics with Applications*, vol. 59, no. 5, pp. 1594–1609, 2010.
- [11] Y. Chen, H.-S. Ahn, and I. Podlubny, "Robust stability check of fractional order linear time invariant systems with interval uncertainties," *Signal Processing*, vol. 86, no. 10, pp. 2611–2618, 2006.
- [12] T. P. Sales, D. A. Rade, and L. C. G. De Souza, "Passive vibration control of flexible spacecraft using shunted piezoelectric transducers," *Aerospace Science and Technology*, vol. 29, no. 1, pp. 403–412, 2013.
- [13] J. Erdong and S. Zhaowei, "Passivity-based control for a flexible spacecraft in the presence of disturbances," *International Journal of Non-Linear Mechanics*, vol. 45, no. 4, pp. 348–356, 2010.
- [14] K. Lu, Y. Xia, Z. Zhu, and M. V. Basin, "Sliding mode attitude tracking of rigid spacecraft with disturbances," *Journal of the Franklin Institute. Engineering and Applied Mathematics*, vol. 349, no. 2, pp. 413–440, 2012.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

