Journal of Southeast University (English Edition) Vol. 34, No. 1, pp. 6 - 14 Mar. 2018 ISSN 1003—7985

Parameter identification of the fractional-order systems based on a modified PSO algorithm

Liu Lu[1] Shan Liang[1] Jiang Chao[2] Dai Yuewei[1] Liu Chenglin[3] Qi Zhidong[1]

([1]School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China)

([2]Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA)

([3]Key Laboratory of Advanced Process Control for Light Industry of Ministry of Education, Jiangnan University, Wuxi 214122, China)

Abstract: In order to better identify the parameters of the fractional-order system, a modified particle swarm optimization (MPSO) algorithm based on an improved Tent mapping is proposed. The MPSO algorithm is validated with eight classical test functions, and compared with the POS algorithm with adaptive time varying accelerators (ACPSO), the genetic algorithm (GA), and the improved PSO algorithm with passive congregation (IPSO). Based on the systems with known model structures and unknown model structures, the proposed algorithm is adopted to identify two typical fractional-order models. The results of parameter identification show that the application of average value of position information is beneficial to making full use of the information exchange among individuals and speeds up the global searching speed. By introducing the uniformity and ergodicity of Tent mapping, the MPSO avoids the extreme value of position information, so as not to fall into the local optimal value. In brief, the MPSO algorithm is an effective and useful method with a fast convergence rate and high accuracy.

Key words: particle swarm optimization; Tent mapping; parameter identification; fractional-order systems; passive congregation

DOI: 10.3969/j.issn.1003-7985.2018.01.002

Fractional calculus and integral calculus were proposed 300 years ago. With the development of computer technology, the fractional calculus theory has drawn increasing attention[1-2]. This is because we have found that some systems, such as economic systems[3], biology[4], thermal systems[5], electric power systems[6], can be described in detail by fractional differential equations.

The fractional-order model is widely used in all types of fields due to its lower order, fewer parameters and higher modelling accuracy. At present, the research of the fractional-order system identification is still in the initial stage, and there is no uniform identification method. Since the physical meaning of fractional calculus is still not certain, many identification methods for the fractional-order system do not take into account the structural parameters of the model. Therefore, many experts have focused on the study of the identification of fractional-order structure parameters[7-8].

Recently, considerable attention has been paid to evolutionary algorithms, such as differential evolution (DE)[9], the genetic algorithm (GA)[10] and particle swarm optimization (PSO)[11]. Especially, compared with other evolutionary methods, PSO has the advantages of high speed, high efficiency and is easy to understand. A number of identification results show that the PSO algorithm is effective in the fractional-order system identification[12-13].

However, the classical PSO can easily fall into a local optimum, especially in a complex and high dimensional situation. In order to improve the performance of the traditional PSO, various PSO variants were proposed. One of the PSO variants proposed by Shi and Eberhart introduced the concept of inertia weight[14]. The inertia weight is added to balance the global optimization and local optimization abilities. Some PSO variants try to improve the convergence performance of the algorithm by adding the characteristics of other methods[15-16]. Aghababa[17] proposed a new PSO with adaptive time varying accelerators (ACPSO). However, these improvements did not change the three components of the traditional PSO algorithm. Therefore, the information transmission among individuals cannot be carried out, which ignores some useful information. Considering the integrity of group information, He et al.[18] proposed an improved PSO with passive congregation (IPSO). This algorithm only considers the information of a single individual and does not make full use of the information of the group. In this paper, a modified PSO algorithm (MPSO) is presented. This method can not only enable the individuals to obtain the information of the global best individual, but also achieve the information exchange among individuals.

Received 2017-09-27, **Revised** 2018-01-13.

Biographies: Liu Lu (1990—), male, Ph. D. candidate; Shan Liang (corresponding author), male, doctor, associate professor, shanliang@njust.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 61374153, 61473138, 61374133), the Natural Science Foundation of Jiangsu Province (No. BK20151130), Six Talent Peaks Project in Jiangsu Province (No. 2015-DZXX-011), China Scholarship Council Fund (No. 201606845005).

Citation: Liu Lu, Shan Liang, Jiang Chao, et al. Parameter identification of the fractional-order systems based on a modified PSO algorithm [J]. Journal of Southeast University (English Edition), 2018, 34(1): 6 - 14. DOI: 10.3969/j.issn.1003-7985.2018.01.002.

1 Theory

1.1 Definition of fractional-order derivatives and integrals

The development of fractional calculus has gone through a long process, which can be regarded as the generalization of integral calculus. The basic operational factor of fractional calculus is ${}_a\mathrm{d}_t^{\alpha}$, where a and t are the upper and lower bounds of the operational factor, respectively. α is the order of calculus, and its value can be any complex number. The difference of calculus operator depends on the value of α, as shown below.

$$ {}_a\mathrm{d}_t^{\alpha}=\begin{cases}\dfrac{\mathrm{d}^{\alpha}}{\mathrm{d}t^{\alpha}} & \mathrm{Re}(\alpha)>0\\ 1 & \mathrm{Re}(\alpha)=0\\ \displaystyle\int_a^t(\mathrm{d}\tau)^{-\alpha} & \mathrm{Re}(\alpha)<0\end{cases} \tag{1}$$

where $\mathrm{Re}(\alpha)$ is the real part of α.

So far, fractional calculus has no uniform definition. In the development of fractional calculus, there are several definitions of fractional calculus. Three most commonly used definitions are Caputo, Riemann-Liouville (RL) and Grumwald-Letnikov (GL)[19]. For example, the definition of GL is defined as

$$ {}_a\mathrm{d}_t^{\alpha}f(t)=\lim_{h\to 0}\frac{1}{h^{\alpha}}\sum_{j=0}^{(t-a)/h}w_j^{(\alpha)}f(t-jh) \tag{2}$$

$$ w_j^{(\alpha)}=\frac{(-1)^j\Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)} \tag{3}$$

where $\Gamma(\cdot)$ is the Gamma function; $[(t-a)/h]$ is the largest integer that is not greater than $(t-a)/h$, and h is the sampling period.

1.2 Fractional-order systems

Conventionally, integer-order calculus is used to describe natural phenomena, but many phenomena in nature cannot be accurately described by the traditional integer-order differential equations. Therefore, it is necessary to extend the traditional calculus to describe such phenomena. Fractional differential equation is an extension of traditional calculus, which can describe complex physical systems more accurately.

The general expression of the SISO linear fractional-order system differential equation is shown below.

$$ \begin{aligned}&a_nD^{\alpha_n}y(t)+a_{n-1}D^{\alpha_{n-1}}y(t)+\cdots+a_0y(t)=\\ &\quad b_mD^{\beta_m}u(t)+b_{m-1}D^{\beta_{m-1}}u(t)+\cdots+b_0u(t)\end{aligned} \tag{4}$$

where a_i and b_i are real numbers; α_i and β_i are calculus orders; $u(t)$ and $y(t)$ are the input and output signals of the system, respectively. Under a zero initial condition, the expression of the transfer function is described as

$$ G(s)=\frac{Y(s)}{U(s)}=\frac{b_ms^{\beta_m}+b_{m-1}s^{\beta_{m-1}}+\cdots+b_0}{a_ns^{\alpha_n}+a_{n-1}s^{\alpha_{n-1}}+\cdots+a_0} \tag{5}$$

At present, it is difficult to identify the general fractional-order system, because it is difficult to determine the order of the system. In Ref. [20], the concept of continuous order distribution was proposed, and the general expression of the fractional-order system was transformed into an expression with common factor order, which is shown below.

$$ \begin{aligned}&a_nD^{n\alpha}y(t)+a_{n-1}D^{(n-1)\alpha}y(t)+\cdots+a_0y(t)=\\ &\quad b_mD^{m\beta}u(t)+b_{m-1}D^{(m-1)\beta}u(t)+\cdots+b_0u(t)\end{aligned} \tag{6}$$

The corresponding transfer function is

$$ G(s)=\frac{Y(s)}{U(s)}=\frac{b_ms^{m\beta}+b_{m-1}s^{(m-1)\beta}+\cdots+b_0}{a_ns^{n\alpha}+a_{n-1}s^{(n-1)\alpha}+\cdots+a_0} \tag{7}$$

1.3 PSO variants

1.3.1 ACPSO algorithm

PSO is a new evolutionary algorithm, which seeks for the optimal solution by sharing information among individuals. In the process of optimization, each particle updates its velocity and position as follows[15]:

$$ v_i(t+1)=wv_i(t)+c_1r_1(P_i(t)-x_i(t))+c_2r_2(G(t)-x_i(t)) \tag{8}$$

$$ x_i(t+1)=x_i(t)+v_i(t+1) \tag{9}$$

where w is the inertia weight; $v_i(t)$ is the velocity of particle i at iteration t; c_1 and c_2 are two positive constants between 0 and 1; r_1 and r_2 are two random numbers between 0 and 1; $P_i(t)$ is the optimum value of particle i at iteration t; $x_i(t)$ is the current position of particle i at iteration t; $G(t)$ is the optimum value of the particle population.

In Ref. [14], a linearly decreasing inertia weight strategy was introduced in conventional PSO. The modified inertia weight w is shown below.

$$ w(t)=w_{\max}-\frac{w_{\max}-w_{\min}}{T}t \tag{10}$$

The corresponding velocity updating formula is

$$ \begin{aligned}v_i(t+1)=&w(t)v_i(t)+c_1r_1(P_i(t)-x_i(t))+\\ &c_2r_2(G(t)-x_i(t))\end{aligned} \tag{11}$$

where $w_{\max}$ is the maximum value of inertia weight w; $w_{\min}$ is the minimum value of inertia weight w; T is the maximum number of iterations. Usually, the inertia weight w is set from 0.9 to 0.4[21].

In Ref. [17], in order to improve the convergence rate of PSO, the ACPSO is proposed, which is expressed as

$$ \begin{aligned}v_i(t+1)=&wv_i(t)+c_1r_1\frac{f(P_i(t)-x_i(t))}{f(P_i(t))}(P_i(t)-x_i(t))+\\ &c_{22}\frac{f(G(t)-x_i(t))}{f(G(t))}(G(t)-x_i(t))\end{aligned} \tag{12}$$

where $f(P_i(t))$ is the best fitness function found by the i-th particle at iteration t; $f(G(t))$ is the best objective function found by the swarm up to the iteration t. With this method, the velocity can be updated adaptively,

which accelerates the convergence of the algorithm.

1.3.2 IPSO algorithm

The classical PSO algorithm only uses the individual optimal value and the optimal value of the population, and does not take into account the information of other individuals in the population. With this in mind, an improved PSO with the passive congregation was proposed in Ref. [18]. The velocity updating of the IPSO algorithm is shown as

$$v_i(t+1) = w(t)v_i(t) + c_1r_1(P_i(t) - x_i(t)) + c_2r_2(G(t) - x_i(t)) + c_3r_3(P(t) - x_i(t)) \tag{13}$$

where $P(t)$ is the current position of a random particle at iteration t. The introduction of the passive congregation can further increase the information exchange between groups, so that the population can find the optimal value more quickly.

2 Proposed MPSO Algorithm

In order to increase the exchange of information among individuals, and make full use of the information in the population, so as to improve the search speed and avoid the local optimum, the average value of position is introduced in this paper. Furthermore, in order to avoid the extreme value existing in the position information, a modified Tent mapping is proposed.

2.1 Modified Tent mapping

Tent mapping is a type of piecewise linear one-dimensional chaotic mapping, which has the advantages of simple form, uniform power spectrum density and good correlation properties. Compared with other chaotic sequences, such as logistics sequence, the improved Tent mapping has better uniformity and is more suitable for computer digital computation. The recursive formula of Tent mapping is

$$f(x_n) = \begin{cases} 2x_n & x_n \in [0, 0.5] \\ 2(1 - x_n) & x_n \in (0.5, 1] \end{cases} \tag{14}$$

There are small cycles and some unstable periodic points in the Tent sequence, e. g. (0.2, 0.4, 0.8, 0.6) and (0.25, 0.5, 0.75). In order to prevent the iterations from falling into the small cycles, we design the modified sequence generation method. The specific steps are as follows.

1) Take a point x_0 that does not fall into the small cycle, and set $z(1) = x_0$, $i = j = 1$.

2) Generate sequence x according to Eq. (14), $i = i + 1$.

3) Suppose that the maximum iteration number is M_1, if $i > M_1$, then execute Step 5); else if $x(i) \in \{0, 0.25, 0.5, 0.75\}$ or $x(i) = x(i-k)$, $k \in \{0, 1, 2, 3, 4\}$, then return to Step 4), else return to Step 2).

4) Change the initial value of iteration: $x(i) = z(j+1) = z(j) + \varepsilon$, $j = j + 1$, return to Step 2).

5) Terminate the operation and save sequence x.

The initial value and the iteration number are set to be 0.435 8 and 1 000, respectively. Fig. 1 shows the generated sequence x.

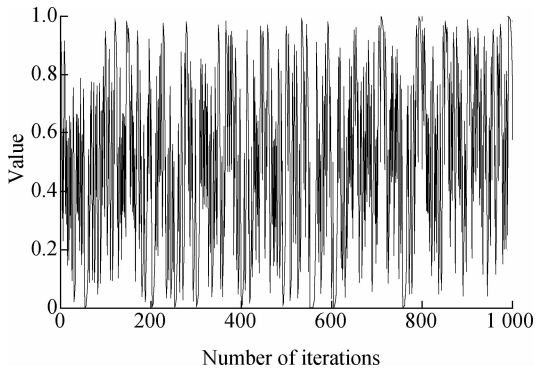

Fig. 1 The generated sequence diagram

2.2 MPSO algorithm

Suppose that the number of particles is N, the specific steps of the Tent chaotic search are shown below.

1) The average value M of position is obtained, and the definition of M is

$$M = \sum_{i=1}^{N} \frac{x_i(t)}{N}$$

2) The optimization variables from the interval $[a, b]$ are transformed into chaotic variables z from the interval $[0, 1]$,

$$z_i = (M_i - a)/(b - a) \qquad i = 1, 2, \cdots, D$$

where D is the number of variable dimensions.

3) N iterations are executed to generate z_i^n according to the generation method of chaotic sequence in Section 2.1, $n = 1, 2, \cdots, N$.

4) The chaotic sequence is translated to the original variable range as follows:

$$M_i^n = a + (b - a)z_i^n$$

Then, N feasible solutions can be obtained by

$$M^n = (M_1^n, M_2^n, \cdots, M_D^n)$$

5) The average value is calculated as

$$P_{\text{av}} = \sum_{i=1}^{N} \frac{M}{N}$$

The velocity updating of the MPSO algorithm is shown as

$$v_i(t+1) = w(t)v_i(t) + c_1r_1(P_i(t) - x_i(t)) + c_2r_2(G(t) - x_i(t)) + c_3r_3(P_{\text{av}}(t) - x_i(t)) \tag{15}$$

The inertia weight w in Eq. (15) is updated according to Eq. (10), and the position is updated according to Eq. (9).

2.3 Performance evaluation

2.3.1 Classical test functions

In the test stage, eight classical test functions are intro-

duced to test the performance of the MPSO and other algorithms as follows:

Ackley

$$f_1(x) = 20 + e - 20\exp\left(-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)$$

Schewefel

$$f_2(x) = 418.982\ 9D + \sum_{i=1}^{D}\left(-x_i\sin(\sqrt{x_i})\right)$$

Rastrigrin function

$$f_3(x) = \sum_{i=1}^{D}\left[x_i^2\cos(2\pi x_i) + 10\right]$$

Griewank function

$$f_4(x) = \sum_{i=1}^{D}\frac{x_i^2}{4\ 000} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

Rosenbrock function

$$f_5(x) = \sum_{i=1}^{D-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$$

Sphere function

$$f_6(x) = \sum_{i=1}^{D} x_i^2$$

Sum squares

$$f_7(x) = \sum_{i=1}^{D} ix_i^2$$

Dixon-price

$$f_8(x) = (x_1 - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_{i-1})^2$$

f_1 to f_8 have a common global minimum. Besides, f_1 to f_4 are multimodal test functions, while f_5 to f_8 are unimodal test functions. Unlike the unimodal test functions, the number of local optimal values in the multimodal test functions increases with the increase of the dimension. The basic information of the eight test functions is listed in Tab. 1.

Tab. 1 Basic information of the test functions

Function	Dimension	Solution space	Optimal value
f_1	30	$[-100,100]^D$	0
f_2	30	$[-500,500]^D$	0
f_3	30	$[-100,100]^D$	0
f_4	30	$[-100,100]^D$	0
f_5	30	$[-10,10]^D$	0
f_6	30	$[-10,10]^D$	0
f_7	30	$[-10,10]^D$	0
f_8	30	$[-10,10]^D$	0

2.3.2 Parameter analysis

In the MPSO algorithm, the selection of c_3 influences the performance of the algorithm. In order to analyze the effect of c_3 on the performance of the algorithm, four multimodal functions f_1 to f_4 and two unimodal functions f_5 and f_6 are used to test the algorithm with different c_3 values. In the MPSO algorithm, the population size N is set to be 50; the inertia weight w adopts the linear decreasing strategy in Ref. [14], and its variation range is 0.7-0.9; the learning factors c_1 and c_2 are set to be 0.5-10. The test results for c_3 are shown in Tab. 2.

Tab. 2 Average fitness values of functions f_1-f_6 with different c_3

c_3	Function					
	f_1	f_2	f_3	f_4	f_5	f_6
0	$7.347\ 5\times10^{-2}$	64.872	45.294	$9.231\ 7\times10^{-3}$	$1.235\ 4\times10^{2}$	$1.061\ 4\times10^{-15}$
0.1	$2.391\ 5\times10^{-3}$	5.165 4	48.150	$5.865\ 2\times10^{-4}$	79.659	$1.317\ 6\times10^{-16}$
0.2	$5.931\ 7\times10^{-4}$	6.153 3	39.223	$1.643\ 4\times10^{-4}$	77.891	$1.320\ 7\times10^{-14}$
0.3	$1.934\ 6\times10^{-5}$	0.494 28	51.126	$3.928\ 2\times10^{-4}$	96.976	$8.478\ 3\times10^{-13}$
0.4	$8.279\ 1\times10^{-4}$	0.287 35	$1.208\ 5\times10^{2}$	$7.883\ 4\times10^{-4}$	92.040	$2.815\ 2\times10^{-12}$
0.5	$2.438\ 9\times10^{-3}$	0.763 94	$1.893\ 3\times10^{2}$	$4.090\ 9\times10^{-4}$	63.385	$2.134\ 3\times10^{-10}$
0.6	$7.648\ 5\times10^{-3}$	1.835 7	$2.083\ 4\times10^{2}$	$8.142\ 7\times10^{-3}$	66.271	$3.398\ 3\times10^{-4}$
0.7	$4.638\ 1\times10^{-2}$	36.825	$2.195\ 3\times10^{2}$	$2.853\ 5\times10^{-2}$	52.547	$5.462\ 4\times10^{-3}$
0.8	0.650 48	$5.369\ 8\times10^{2}$	$2.271\ 2\times10^{2}$	0.140 26	65.894	$9.384\ 9\times10^{-2}$
0.9	2.394 5	$7.628\ 8\times10^{3}$	$2.324\ 0\times10^{2}$	0.334 19	80.602	0.284 66
1.0	8.182 7	$9.749\ 5\times10^{3}$	$2.517\ 6\times10^{2}$	0.623 57	97.674	1.582 4
1.1	36.474	$1.297\ 6\times10^{4}$	$2.917\ 3\times10^{2}$	1.286 4	$3.456\ 8\times10^{2}$	31.685
1.2	$4.816\ 5\times10^{2}$	$2.648\ 7\times10^{4}$	$3.047\ 4\times10^{2}$	3.598 4	$7.264\ 8\times10^{2}$	$5.678\ 4\times10^{2}$

From Tab. 2, it can be seen that MPSO obtains good results on function f_1 when $c_3 = 0.3$. The best result for function f_2 is obtained when $c_3 = 0.4$. MPSO obtains good results on function f_6 when $c_3 = 0.1$. For function f_5, the best result is obtained when $c_3 = 0.7$. With $c_3 = 0.2$, MPSO obtains the best results for functions f_3 and f_4. The test results of MPSO on function f_4 and f_6 deteriorate with $c_3 \geqslant 0.8$. Considering the situation mentioned above, the value range of c_3 is set to be 0.1-0.7.

2.3.3 Evaluation results

The population size of all the algorithms is set to be 50. For the GA algorithm, the crossover and mutation probabilities are $P_c = 0.7$ and $P_m = 0.15$, respectively. For the IPSO algorithm, the learning factors c_1 and c_2 are

set to be 0.5, and the inertia weight w is set to be [0.7, 0.9]. For the ACPSO algorithm, the learning factors c_1 and c_2 are set to be 2, and the inertia weight w is the same as that in Ref. [18]. The proposed MPSO algorithm is tested by the classical test functions f_1 to f_8, and is compared with GA, IPSO and ACPSO algorithms. The maximum number of iterations is 1 000. In order to make the results more convincing, 100 experiments were carried out. The mean, the standard deviations (SD) and the average number of iterations (I_{ter}) of the test results are listed in Tab. 3.

Tab. 3 Performance comparisons of MPSO, ACPSO, IPSO and GA algorithms

Function	GA		IPSO		ACPSO		MPSO	
	Mean(SD)	I_{ter}	Mean(SD)	I_{ter}	Mean(SD)	I_{ter}	Mean(SD)	I_{ter}
f_1	0.583 47 (0.928 64)	256	0.253 78 (0.483 67)	201	$7.728\ 1\times10^{-3}$ ($1.359\ 2\times10^{-2}$)	175	$2.247\ 5\times10^{-5}$ ($5.613\ 7\times10^{-5}$)	186
f_2	$4.284\ 1\times10^{3}$ ($7.482\ 5\times10^{3}$)	183	$1.762\ 9\times10^{3}$ ($2.967\ 4\times10^{3}$)	96	$1.186\ 2\times10^{2}$ ($2.857\ 3\times10^{2}$)	80	0.372 86 (0.786 48)	83
f_3	$1.605\ 3\times10^{2}$ ($1.882\ 0\times10^{2}$)	286	$1.051\ 2\times10^{2}$ ($1.452\ 2\times10^{2}$)	135	28.955 (30.848)	117	$7.506\ 7\times10^{-2}$ (0.179 09)	128
f_4	0.374 98 (0.279 88)	247	0.191 24 (0.216 11)	162	$1.383\ 6\times10^{-2}$ ($1.836\ 8\times10^{-2}$)	145	$1.657\ 6\times10^{-5}$ ($2.171\ 9\times10^{-5}$)	149
f_5	$1.446\ 7\times10^{3}$ ($6.122\ 5\times10^{3}$)	219	$2.672\ 1\times10^{2}$ ($8.429\ 4\times10^{2}$)	128	23.034 (52.787)	102	1.484 52 (2.420 25)	108
f_6	$4.328\ 8\times10^{-3}$ ($9.516\ 7\times10^{-3}$)	186	$1.597\ 8\times10^{-5}$ ($6.326\ 5\times10^{-5}$)	134	$2.240\ 6\times10^{-8}$ ($7.516\ 1\times10^{-8}$)	82	$2.341\ 2\times10^{-10}$ ($4.479\ 8\times10^{-10}$)	97
f_7	3.521 3 (7.473 4)	282	0.231 75 (0.782 44)	201	$4.073\ 8\times10^{-2}$ ($8.341\ 7\times10^{-2}$)	157	$2.642\ 4\times10^{-3}$ ($6.327\ 5\times10^{-3}$)	164
f_8	$2.519\ 7\times10^{2}$ ($7.627\ 4\times10^{2}$)	302	57.162 ($1.138\ 4\times10^{2}$)	173	0.653 72 (0.986 72)	131	0.105 17 (0.312 16)	137

From Tab. 3, compared with GA and IPSO algorithms, it can be clearly found that the MPSO algorithm has better performance in terms of both the mean and standard deviation. Compared with the ACPSO algorithm, the overall convergence rate of MPSO is relatively slow, but the MPSO algorithm has better convergence precision. One reason is that the ACPSO algorithm uses an adaptive learning factor, which can speed up the convergence rate of the algorithm. On the other hand, the proposed MPSO algorithm increases the information exchange among individuals in the population, and can search the solution space more carefully, so as to achieve better convergence precision.

In conclusion, the introduction of the average value of position takes full advantage of the information in the population, and the application of Tent mapping avoids the emergence of extreme values, which improves the convergence precision and the searching efficiency of the MPSO algorithm.

3 Simulations

To demonstrate the effectiveness of the proposed MPSO algorithm, based on the systems with known model structures and unknown model structures, the MPSO algorithm is compared with some typical algorithms, including GA, IPSO and ACPSO algorithms. The basic parameter settings are similar to the settings in Section 2.3. Only considering the influence of population size (P_N) on time complexity, the analysis of time complexity for the proposed MPSO algorithm and other algorithms being compared in a single run is concluded as follows: For the MPSO algorithm, the time complexity for the initialization process is $O(P_N)$; the time complexity for calculating the objective function values is $O(P_N)$; the time complexity for updating and selecting the individual and global optimal values is $O(P_N)$; the time complexity for calculating the mean of the position information in passive congregation terms is $O(P_N)$; the time complexity for updating the location information of particles using Tent mapping is $O(P_N)$; the time complexity for generating new velocities and positions is $O(P_N)$. Thus, the overall time complexity of the MPSO algorithm in one iteration is: $O(P_N)+O(P_N)+O(P_N)+O(P_N)+O(P_N)+O(P_N)$, which can be regarded as $O(\mathrm{MPSO})=O(P_N)$. For the GA algorithm, the time complexity is mainly composed of roulette wheel selection, crossover and mutation operations, and the corresponding time complexities are $O(P_N^2)$, $O(P_N)$ and $O(P_N)$, respectively. Therefore, the time complexity of the GA algorithm can be simplified as $O(\mathrm{GA})=O(P_N^2)$. Similarly, the time complexities of the IPSO and ACPSO algorithms are $O(P_N)$ and $O(P_N)$, respectively. In short, we can conclude that $O(\mathrm{GA})>O(\mathrm{MPSO})\ncong O(\mathrm{IPSO})\ncong O(\mathrm{ACPSO})$.

The simulations are implemented using MATLAB 7.11 on Intel(R) Core(TM) i5-2320 CPU, 3.00 GHz with 4 GB RAM. The specific identification steps are as follows.

1) Determine the vector to be searched (model param-

eters and order).

2) Determine the performance index (evaluation function). In this paper, the step signal is introduced to be as the input signal. The sum of the squared errors between the output $y(t)$ and the true value $y_0(t)$, i.e., $J = \int_0^T [y(t) - y_0(t)]^2 \mathrm{d}t$ is used as the performance evaluation function.

3) The parameters of the algorithm are initialized to generate random search vectors.

4) According to the steps of the MPSO algorithm, the parameters in the fractional-order system are identified.

5) The iteration is repeated until the performance index is satisfactory. Output the identification results.

The schematic diagram of parameter estimation for the fractional-order system is shown in Fig. 2.

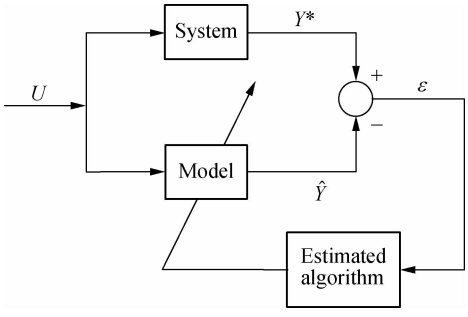

Fig. 2 Schematic diagram of parameter estimation for the fractional-order system

3.1 Identification of known fractional-order model structure

Assume that the transfer function of the identified object is

$$G = \frac{1}{2.4S^{2.1} + 1.3S^{1.2} + 1} \tag{16}$$

The range of parameters is set to be $[0, 3]$, and the random number for the model order changes from -0.05 to 0.05. The identification vector is defined as

$$\boldsymbol{x} = [a_1 \quad a_2 \quad a_3 \quad b_1 \quad b_2]$$

where a_1, a_2, a_3 are the model parameters and b_1, b_2 are the orders of the model. The statistical results of the best, the mean and the worst estimated parameters over 20 independent runs are shown in Tab. 4.

We can see from Tab. 4 that the obtained estimated values by MPSO are closer to the true values, which shows that the MPSO algorithm is more accurate than GA, IPSO and ACPSO algorithms. Furthermore, it can also be easily found that the best objective function values obtained by MPSO are better than those obtained by other algorithms.

The estimated parameters generated by various algorithms in a single run are shown in Fig. 3. Fig. 3 reveals that the performance of the MPSO algorithm surpasses the

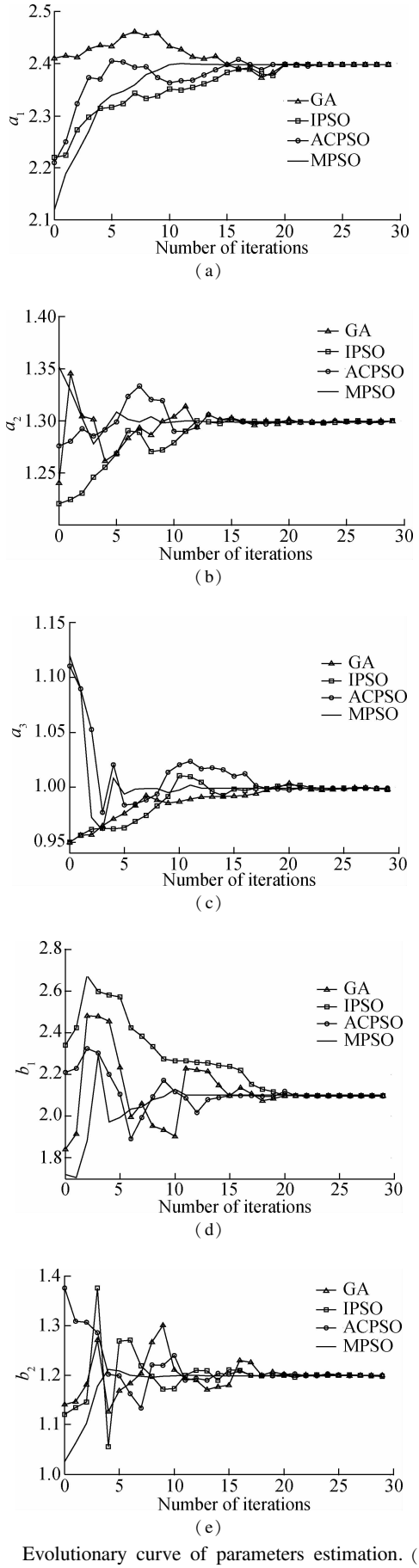

Fig. 3 Evolutionary curve of parameters estimation. (a) a_1; (b) a_2; (c) a_3; (d) b_1; (e) b_2

Tab. 4 The estimation results of various algorithms for Eq. (16)

Algorithm		GA	IPSO	ACPSO	MPSO
Best	a_1	2.381 0	2.384 8	2.389 7	2.398 2
	a_2	1.295 4	1.297 0	1.298 2	1.299 8
	a_3	0.990 2	0.991 1	0.994 3	0.999 4
	b_1	2.088 9	2.089 9	2.095 0	2.099 9
	b_2	1.194 5	1.195 0	1.198 2	1.199 6
	J	0.245 1	0.209 4	$7.820\,4\times10^{-2}$	$6.406\,8\times10^{-4}$
Mean	a_1	2.375 2	2.379 1	2.385 3	2.394 8
	a_2	1.271 0	1.279 8	1.286 2	1.298 7
	a_3	0.982 6	0.984 7	0.995 8	0.998 2
	b_1	2.074 3	2.078 1	2.090 1	2.098 1
	b_2	1.180 7	1.185 8	1.194 8	1.198 2
	J	0.733 2	0.593 3	$4.841\,5\times10^{-2}$	$7.701\,4\times10^{-3}$
Worst	a_1	2.352 7	2.359 9	2.382 9	2.411 7
	a_2	1.263 7	1.275 1	1.283 5	1.283 8
	a_3	0.977 5	0.981 7	0.994 2	0.996 4
	b_1	2.065 6	2.073 6	2.087 2	2.096 4
	b_2	1.178 4	1.180 3	1.192 6	1.191 4
	J	1.247 2	0.836 1	$9.350\,3\times10^{-2}$	$1.781\,5\times10^{-2}$

Tab. 5 The estimation results of various algorithms for Eq. (16)

Algorithm		GA	IPSO	ACPSO	MPSO
Best	a_1	2.271 8	2.283 7	2.290 8	2.354 7
	a_2	1.116 7	1.128 7	1.138 4	1.182 1
	a_3	0.933 7	0.943 7	0.948 4	0.992 3
	b_1	1.926 8	1.947 1	1.977 5	2.035 7
	b_2	1.063 2	1.076 8	1.093 7	1.168 5
	J	14.138 0	10.603 0	7.657 1	0.492 0
Mean	a_1	2.209 5	2.214 6	2.217 8	2.321 9
	a_2	1.051 3	1.051 8	1.127 6	1.167 2
	a_3	0.892 7	0.903 7	0.940 9	0.989 2
	b_1	1.951 6	1.952 0	1.960 7	2.026 3
	b_2	1.049 7	1.052 8	1.089 4	1.144 6
	J	24.489 0	21.098 0	11.148 0	1.313 2
Worst	a_1	1.980 8	2.027 3	2.113 8	2.212 1
	a_2	0.986 8	0.995 4	1.087 4	1.148 5
	a_3	0.873 7	0.897 6	0.927 5	0.954 7
	b_1	1.858 2	1.872 3	1.910 6	1.937 1
	b_2	0.975 7	0.982 8	1.097 2	1.115 4
	J	37.857 0	26.896 0	15.804 0	9.255 7

IPSO algorithm due to the full use of information among individuals. Moreover, in the early stage of evolution, the convergence rate of the ACPSO algorithm is slightly faster due to the introduction of an adaptive learning factor. However, the estimated parameters generated by MPSO yield a better precision as compared to that of other comparison algorithms, which suggests that our presented algorithm is more susceptible to a small change in parameters.

Moreover, considering the time complexity, we can easily see that the MPSO algorithm outperforms the GA algorithm whether in the convergence characteristics or in the time complexity. In addition, under the same level of time complexity with the IPSO and ACPSO algorithms, the MPSO algorithm has a better convergence rate and accuracy.

In order to study the influence of the variation range of order on the identification, the variation range of order is set to be $[-0.5, 0.5]$, and other conditions remain unchanged. The statistical results are shown in Tab. 5.

From Tab. 5, we can see that the identification effect can be improved significantly with the decrease of the random number range of order. The main reason is that the objective function value affected by the order changes exponentially, which indicates that there is a significant change in the function value with a minor change of order. Therefore, the appropriate reduction of the order helps to shorten the convergence time of the algorithm and improve search ability.

3.2 Identification for unknown fractional-order model structure

In the model identification, the model structure is often unknown. When the assumed model structure matches the real model, the error is smaller than that of other models. In the actual identification, the high-order system is relatively small, and the higher order fractional-order system can be reduced to one of the models in Tab. 6. Therefore, the three models in Tab. 6 are selected for simulation study. Assume that the transfer function of the identified object is

$$G=\frac{1}{2.5S^{2.3}+1.4S^{1.4}+0.8S^{0.6}+1} \tag{17}$$

Tab. 6 The model structure

Model	Structure	Parameter
1	$\frac{1}{a_1S^{b_1}+a_2}$	a_1, a_2, b_1
2	$\frac{1}{a_1S^{b_1}+a_2S^{b_2}+a_3}$	a_1, a_2, a_3, b_1, b_2
3	$\frac{1}{a_1S^{b_1}+a_2S^{b_2}+a_3S^{b_3}+a_4}$	$a_1, a_2, a_3, a_4, b_1, b_2, b_3$
4	$\frac{1}{a_1S^{b_1}+a_2S^{b_2}+a_3S^{b_3}+a_4S^{b_4}+a_5}$	$a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4$

The statistical results are shown in Tab. 7. From Tab. 7, we notice that the MPSO algorithm outperforms other algorithms being compared. In addition, we can also find that the performance index J of model 3 is significantly small, so model 3 is the closest to the target. In addition, taking the J value of model 3 as the dividing point, the J value decreases first and then increases. Furthermore, it is clear that the difficulty of parameter identification greatly increases with the increase in the number of the parameters in the system.

4 Conclusions

1) A new parameter identification scheme based on a modified PSO algorithm is proposed. The introduction of the average value of position information makes full use of the information exchange among individuals in the population, which can improve the precision and efficiency of optimization. By utilizing the uniformity and ergodicity

Tab.7 The estimation results of various algorithms for Eq. (17)

Algorithm		GA	IPSO	ACPSO	MPSO
Model 1	a_1	3.8946	3.781 7	2.977 4	2.812 7
	a_2	1.713 7	1.637 1	1.498 7	1.418 3
	b_1	1.597 5	1.582 1	1.393 7	1.265 1
	J	$1.777\ 8\times10^2$	$1.479\ 4\times10^2$	$1.090\ 6\times10^2$	91.345 0
Model 2	a_1	3.631 5	3.624 2	3.583 4	3.478 5
	a_2	1.325 8	1.337 5	1.391 7	1.406 1
	a_3	1.044 2	1.053 3	1.017 5	1.089 3
	b_1	2.367 2	2.356 1	2.310 7	2.257 0
	b_2	0.921 8	0.913 7	0.901 6	0.849 7
	J	$1.081\ 8\times10^2$	88.850 0	39.057 0	15.404 0
Model 3	a_1	2.897 1	2.873 2	2.857 4	2.841 7
	a_2	1.219 3	1.184 1	1.171 8	1.187 3
	a_3	0.901 2	0.892 4	0.887 3	0.875 1
	a_4	1.138 5	1.104 7	1.054 2	1.017 6
	b_1	2.354 1	2.283 1	2.267 5	2.271 5
	b_2	1.375 4	1.369 1	1.360 1	1.358 2
	b_3	0.679 1	0.670 2	0.668 2	0.661 9
	J	17.506 0	8.181 6	1.675 2	0.385 7
Model 4	a_1	0.257 4	0.302 4	0.348 3	0.295 4
	a_2	2.094 8	2.337 5	2.277 3	2.183 2
	a_3	1.106 7	1.078 6	1.107 4	1.139 3
	a_4	1.293 8	1.215 2	1.164 8	1.131 7
	a_5	1.347 4	1.128 7	1.114 3	1.073 6
	b_1	2.478 1	2.246 4	2.207 4	2.156 1
	b_2	2.602 1	2.682 8	2.628 7	2.505 9
	b_3	1.621 4	1.593 9	1.507 2	1.494 7
	b_4	0.862 5	0.838 9	0.806 7	0.781 6
	J	$1.064\ 3\times10^2$	72.682 0	36.149 0	7.955 3

of Tent mapping, MPSO can avoid the extreme value of position information, so as not to fall into local optimal values.

2) Compared with the other three algorithms, the identification results verify the fine searching capability and efficiency of the presented algorithm. For the systems with known model structures and unknown model structures, the proposed approach can obtain the estimated parameters with a faster convergence rate and higher convergence precision.

3) In conclusion, the proposed MPSO algorithm is an efficient and promising approach for parameter identification of the fractional-order system.

We mainly study the fractional-order linear model with constant coefficient in this paper. In future work, we will further study the parameter identification of the fractional-order system which contains nonlinearity models and input signals with fractional differential components.

References

[1] Diethelm K. An efficient parallel algorithm for the numerical solution of fractional differential equations [J]. *Fractional Calculus and Applied Analysis*, 2011, **14**(3): 475 - 490. DOI:10.2478/s13540-011 - 0029-1.

[2] Gepreel K A. Explicit Jacobi elliptic exact solutions for nonlinear partial fractional differential equations [J]. *Advances in Difference Equations*, 2014, **2014**(1): 1 - 14. DOI:10.1186/1687-1847 - 2014-286.

[3] Laskin N. Fractional market dynamics [J]. *Physica A: Statistical Mechanics and Its Applications*, 2000, **287**(3): 482 - 492. DOI:10.1016/s0378-4371(00)00387-3.

[4] Ionescu C, Machado J T, de Keyser R. Fractional-order impulse response of the respiratory system [J]. *Computers & Mathematics with Applications*, 2011, **62**(3): 845 - 854. DOI:10.1016/j.camwa.2011.04.021.

[5] Badri V, Tavazoei M S. Fractional order control of thermal systems: Achievability of frequency-domain requirements [J]. *Nonlinear Dynamics*, 2014, **80**(4): 1773 - 1783. DOI:10.1007/s11071-014 - 1394-1.

[6] Vikhram Y R, Srinivasan L. Decentralised wide-area fractional order damping controller for a large-scale power system [J]. *IET Generation, Transmission & Distribution*, 2016, **10**(5): 1164 - 1178. DOI:10.1049/iet-gtd.2015.0747.

[7] Safarinejadian B, Asad M, Sadeghi M S. Simultaneous state estimation and parameter identification in linear fractional order systems using coloured measurement noise [J]. *International Journal of Control*, 2016, **89**(11): 2277 - 2296.

[8] Taleb M A, Béthoux O, Godoy E. Identification of a PEMFC fractional order model [J]. *International Journal of Hydrogen Energy*, 2017, **42**(2): 1499 - 1509. DOI: 10.1016/j.ijhydene.2016.07.056.

[9] Storn R, Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces [J]. *Journal of Global Optimization*, 1997, **11**(4): 341 - 359.

[10] Tao C, Zhang Y, Jiang J J. Estimating system parameters from chaotic time series with synchronization optimized by a genetic algorithm [J]. *Physical Review E*, 2007, **76**(2): 016209. DOI: 10.1103/PhysRevE.76.016209.

[11] Kennedy J, Eberhart R. Particle swarm optimization [J]. *Proceedings of ICNN'95—International Conference on Neural Networks*. Perth, Australia, 1995: 1942 - 1948. DOI:10.1109/icnn.1995.488968.

[12] Gupta R, Gairola S, Diwiedi S. Fractional order system identification and controller design using PSO [C]//2014 *Innovative Applications of Computational Intelligence on Power, Energy and Controls with Their Impact on Humanity*. Ghaziabad, India, 2015: 149 - 153. DOI: 10.1109/cipech.2014.7019053.

[13] Hammar K, Djamah T, Bettayeb M. Fractional hammerstein system identification using particle swarm optimization [C]//2015 *7th IEEE International Conference on Modelling, Identification and Control*. Sousse, Tunisia, 2015: 1 - 6. DOI:10.1109/icmic.2015.7409483.

[14] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]//1998 *IEEE International Conference on Evolutionary Computation*. Anchorage, USA, 1998: 69 - 73. DOI:10.1109/icec.1998.699146.

[15] Koulinas G, Kotsikas L, Anagnostopoulos K. A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem [J]. *Information Sciences*, 2014, **277**: 680 - 693. DOI:10.1016/j.ins.2014.02.155.

[16] Gong Y J, Li J J, Zhou Y, et al. Genetic learning particle swarm optimization [J]. *IEEE Trans Cybern*, 2016,

46(10): 2277 - 2290. DOI: 10.1109/TCYB.2015.2475174.

[17] Aghababa M P. Optimal design of fractional-order PID controller for five bar linkage robot using a new particle swarm optimization algorithm [J]. *Soft Computing*, 2016, **20**(10): 4055 - 4067. DOI: 10.1007/s00500-015-1741-2.

[18] He S, Wu Q H, Wen J Y, et al. A particle swarm optimizer with passive congregation [J]. *Biosystems*, 2004, **78**(1/2/3): 135 - 147. DOI: 10.1016/j.biosystems.2004.08.003.

[19] Rossikhin Y A, Shitikova M V. Application of fractional calculus for dynamic problems of solid mechanics: Novel trends and recent results [J]. *Applied Mechanics Reviews*, 2009, **63**(1): 010801. DOI: 10.1115/1.4000563.

[20] Hartley T T, Lorenzo C F. Fractional-order system identification based on continuous order-distributions [J]. *Signal Processing*, 2003, **83**(11): 2287 - 2300. DOI: 10.1016/s0165-1684(03)00182-8.

[21] Gaing Z L. A particle swarm optimization approach for optimum design of PID controller in AVR system [J]. *IEEE Transactions on Energy Conversion*, 2004, **19**(2): 384 - 391. DOI: 10.1109/tec.2003.821821.

基于改进粒子群算法的分数阶系统参数辨识

刘 璐[1] 单 梁[1] 蒋 超[2] 戴跃伟[1] 刘成林[3] 戚志东[1]

([1] 南京理工大学自动化学院, 南京 210094)

([2] Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA)

([3] 江南大学轻工过程先进控制教育部重点实验室, 无锡 214122)

摘要: 为了更好地辨识分数阶系统的参数, 提出了一种基于 Tent 映射的改进粒子群算法(MPSO). 采用 8 个经典测试函数对 MPSO 算法的性能进行了测试,并与自适应时变加速器算法(ACPSO)、改进的被动聚集粒子群算法(IPSO)以及遗传算法(GA)进行对比, 验证了所提算法的有效性. 在已知模型结构和未知模型结构的基础上,利用所提算法对 2 种典型分数阶模型进行参数辨识. 参数辨识结果表明,应用位置信息的平均值有利于充分共享个体间的信息,从而能够加快全局搜索速度; Tent 映射具有的均匀性和遍历性能够防止位置信息中极值的产生,避免算法陷入局部最优. MPSO 算法收敛速度快、精度高,是一种有效且实用的方法.

关键词: 粒子群优化;Tent 映射;参数辨识;分数阶系统;被动聚集

中图分类号: TP301.6