Design of Fractional Order Controller Based on Particle Swarm Optimization

Jun-Yi Cao and Bing-Gang Cao

Abstract: An intelligent optimization method for designing Fractional Order PID (FOPID) controllers based on Particle Swarm Optimization (PSO) is presented in this paper. Fractional calculus can provide novel and higher performance extension for FOPID controllers. However, the difficulties of designing FOPID controllers increase, because FOPID controllers append derivative order and integral order in comparison with traditional PID controllers. To design the parameters of FOPID controllers, the enhanced PSO algorithms is adopted, which guarantee the particle position inside the defined search spaces with momentum factor. The optimization performance target is the weighted combination of ITAE and control input. The numerical realization of FOPID controllers uses the methods of Tustin operator and continued fraction expansion. Experimental results show the proposed design method can design effectively the parameters of FOPID controllers.

Keywords: Evolutionary computation, fractional calculus, fractional order controllers, particle swarm optimization.

1. INTRODUCTION

In recent years, researchers reported that controllers making use of factional order derivatives and integrals could achieve performance and robustness results superior to those obtained with conventional (integer order) controllers [1-4]. Special international symposiums and workshops organized by ASME and IFAC were held to promote international exchange and cooperation in fractional derivatives and their applications research.

Fractional order controllers are described by fractional order differential equations. Expanding derivatives and integrals to fractional orders can adjust control system's frequency response directly and continuously. This great flexibility makes it possible to design more robust control system. Letting control order be fractional, however, is not always straightforward. Several methods have been reported for FOPID design. A method based on pole distribution of the characteristic equation in complex plane was proposed [5]. Vinagre *et al.* presented a frequency domain approach based on the expected crossover frequency and phase margin [6]. A state-space design method based on feedback poles placement can be viewed in [7]. Doctor Ma Chengbin provided a two-stage or hybrid approach: use conventional controller's design method firstly and then improve the performance of designed control system by adding proper fractional order controller [8]. FOPID design is also a parameter optimization problem.

An evolutionary computation technique has become gradually popular to obtain global optimal solution in many areas. A particle swarm optimization (PSO), a stochastic optimization strategy from the family of evolutionary computation, is a biologically-inspired technique originally proposed by Kennedy and Eberhart [9]. PSO has been regarded widely as a promising optimization algorithm due to its combination of simplicity (in terms of its implementation), low computational cost and good performance [10]. What's more, the optimal problems solved by genetic algorithms (GA) can be obtained better solutions with PSO in comparison with conventional methods. These are precisely the main motivations that led us to apply PSO for FOPID controllers design.

This paper is organized as follows. Section 2 introduces the fractional order controllers, their digital realizations and stability. Section 3 presents the particle swarm optimization and its improvement. Section 4 presents the parameter optimization design

Manuscript September 8, 2005; revised June 21, 2006; accepted June 30, 2006. Recommended by Editorial Board member Hoon Kang under the direction of Editor Jin Young Choi. This research was supported by Xian Jiaotong University. The authors would like to thank the reviewers for their comments and suggestions, and also thank Yu Liu for discussion on PSO.

Jun-Yi Cao and Bing-Gang Cao are with the Department of Mechanical and Electronics Engineering, Xian Jiaotong University, P.O.Box 1659, Xian 710049, China (e-mails: caojy@mailst.xjtu.edu.cn, inte-cao@mail.xjtu.edu.cn).

process and results along with a detailed comparative analysis with genetic algorithms. Some conclusions are drawn in Section 5.

2. FRACTIONAL ORDER CONTROLLERS

Fractional order control systems are described by fractional order differential equations. Fractional calculus allows the derivatives and integrals to be any real number. The FOPID controller is the expansion of the conventional PID controller based on fractional calculus. FOPID controllers' parameters designed have five, and the derivative and integral orders improve the design flexibility.

2.1. Fractional calculus

There are several definitions of fractional derivatives [11]. Grunwald-Letnikov definition is perhaps the best known one due to its most suitable for the realization of discrete control algorithms. The m order fractional derivative of continuous function $f(t)$ is given by:

$$\begin{aligned} D^m f(t) &= \lim_{h\to 0} h^{-m} \sum_{j=0}^{[x]} (-1)^{-j} \binom{m}{j} f(t-jh) \\ &= \frac{d^m f(t)}{dt^m}, \end{aligned} \tag{1}$$

where $[x]$ is a truncation and $x = \frac{t-m}{h}$; $\binom{m}{j}$ is binomial coefficients, $\binom{m}{j} = \frac{m(m-1)...(m-j+1)}{j!}$, $\binom{m}{j} = 1,\ (j=0),$ it can be replaced by Gamma function, $\binom{m}{j} = \frac{\Gamma(m+1)}{j!\Gamma(m-j+1)}$.

The general calculus operator, including fractional order and integer order, is defined as:

$${}_aD_t^\alpha = \begin{cases} d^\alpha / dt & \mathbb{R}(\alpha)>0 \\ 1 & \mathbb{R}(\alpha)=0 \\ \int_a^t (d\tau)^{-\alpha} & \mathbb{R}(\alpha)>0, \end{cases} \tag{2}$$

where a and t are the limits related to operation of fractional differentiation, α is the calculus order. The Laplace transform of the fractional derivative of $f(t)$ is given by:

$$L\left\{D^\alpha f(t)\right\} = s^\alpha F(s) - \left[D^{\alpha-1} f(t)\right]_{t=0}, \tag{3}$$

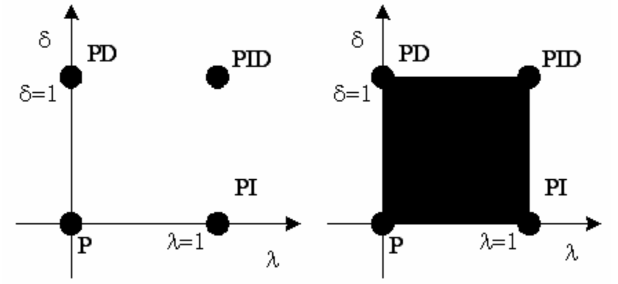

Fig. 1. PID controllers with fractional orders.

where $F(s)$ is the Laplace transform of $f(t)$. The Laplace transform of the fractional integral of $f(t)$ is given as follows:

$$L\left\{D^{-\alpha} f(t)\right\} = s^{-\alpha} F(s). \tag{4}$$

2.2. Fractional order controllers

The differential equation of fractional order controller $PI^\lambda D^\delta$ is described by [12]:

$$u(t) = K_p e(t) + K_i D_t^{-\lambda} e(t) + K_d D_t^\delta e(t). \tag{5}$$

The continuous transfer function of FOPID is obtained through Laplace transform, which is given by:

$$G_c(s) = K_p + K_i s^{-\lambda} + K_d s^\delta. \tag{6}$$

It is obvious that the FOPID controller not only need design three parameters K_p, K_i, and K_d, but also design two orders λ, δ of integral and derivative controllers. The orders λ, δ are not necessarily integer, but any real numbers. As shown in Fig. 1, the FOPID controller generalizes the conventional integer order PID controller and expands it from point to plane. This expansion could provide much more flexibility in PID control design.

2.3. Discretization methods

To realize fractional order controllers perfectly, all the past inputs should be memorized. These are two discretization methods: direct discretization and indirect discretization. In indirect discretization methods, frequency domain fitting in continuous time domain first and discretizing the fit s-transfer function. They couldn't guarantee the stable minimum phase discretization. Several direct discretization methods by finite differential or difference equation were proposed in recent researches, such as Short memory principle, Tustin Expansion, Lagrange function interpolation method [13].

Derived from Grunwald-Letnikov definition, the numerical calculation formula of fractional derivative can be achieved as:

$$ {}_{t-L}D_t^{\alpha}x(t) \approx h^{-\alpha} \sum_{j=0}^{[L/T]} b_j x(t-jh), \tag{7} $$

where L is the length of memory, T, the sampling time, always replaces the time increment h during approximation. The weighting coefficients b_j can be calculated recursively by:

$$ b_0 = 1, b_j = (1 - \frac{1+\alpha}{j}) b_{j-1}, \ (j \geq 1). \tag{8} $$

With generating function $s = \omega(z^{-1})$, the fractional order differentiator s^{α} can be transformed from s domain to z space. The well-known $s \to z$ schemes are Euler and Tustin method. To obtain the coefficients of the approximation equations for fractional calculus, we can consider the Tustin operator as generating function:

$$ s^{\alpha} = (\omega(z^{-1}))^{\alpha} = \left(\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} \right)^{\alpha}, \tag{9} $$

and then perform the continued fraction expansion, the discretized result is as follows:

$$ \begin{aligned} Z\left\{D^{\alpha}x(t)\right\} &= CFE\left\{ \left(\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} \right)^{\alpha} \right\} X(z) \\ &\approx \left(\frac{2}{T}\right)^{\alpha} \frac{P_p(z^{-1})}{Q_q(z^{-1})} X(z), \end{aligned} \tag{10} $$

where $CFE\{u\}$ denotes the continued fraction expansion of u; p and q are the orders of the approximation; P and Q are polynomials of degrees p and q. Normally, we can set $p = q = n.$

The above FOPID controller (6) can be approximated using discretization methods, which is given by:

$$ G_c(z) = K_p + K_i w_i(z) + K_d w_d(z), \tag{11} $$

where $w_i(z)$ is the discrete approximation equation of fractional order integral $s^{-\lambda}$, $w_d(z)$ is the discrete approximation equation of s^{δ}. The greater the truncation order, the better the approximation. That is, the discretized model with higher order is more closely approach to the real fractional order systems.

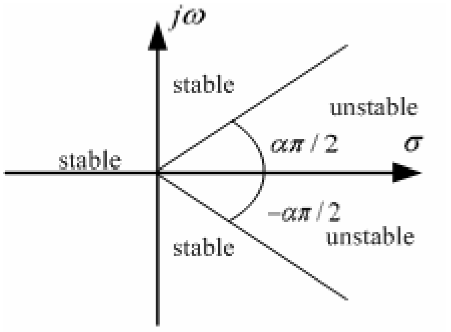

Fig. 2. Stability region of fractional order system.

2.4. Stability conditions

It is well-known that an integer order LTI system is stable if all the roots of the characteristic polynomial $P(s)$ are negative or have negative real parts if they are complex conjugate. This means that they are located on the left of the imaginary axis of the complex plane s. When dealing with fractional order system, the characteristic polynomial is a multivalued function of s, the domain of which can be viewed as a Riemann surface [14]. The stability region of fractional order systems is bounded by a cone, with vertex at the origin, and that extends into the right half of the complex plane s such that it encloses an angle of $\pm\alpha\pi/2$, as shown in Fig. 2. When $\alpha = 1$, we get the stability domain of the integer order system. Thus, when $\alpha = 0.5$, the stability domain is the entire s-plane less the area enclosed by the cone making $\pm 45^0$.

Hence, if all the roots of fractional order system are placed anywhere outside the cone in Fig. 2, it will be stable, Moreover, a controller that stabilizes the integer order system stabilize the integer order model as well as its fractional versions.

3. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a new population-based evolutionary computation. Unlike genetic algorithms, the PSO updates populations without any genetic operators such as crossover and mutation. The PSO algorithm attempts to mimic the natural process of group communication of individual knowledge, which occurs when such swarms flock, migrate, forage, etc, in order to achieve some optimum property such as configuration or location.

In PSO, the 'swarm' is initialized with a population of random solutions. Each particle in the swarm is a different possible set of the unknown parameters to be optimized. Representing a point in the solution space, each particle adjusts its flying toward a potential area according to its own flying experience and shares social information among particles. The goal is to efficiently search the solution space by swarming the particles toward the best fitting solution encountered in previous iterations with the intent of encountering better solutions through the course of the process and eventually converging on a single minimum error solution.

3.1. Standard PSO

Kennedy and Eberhart originated the original framework of PSO in 1995. In PSO, a swarm consists of N particles moving around in a D-dimensional search space. The random velocity assigned to each particle. Each particle modifies its flying based on its own and companion's experience at every iteration. The *i*th particle is denoted as $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$, whose best previous solution (*pbest*) is represented as $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$. Current velocity (position change rate) is described by $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$. Finally, the best solution achieved so far by the whole swarm (*gbest*) is represented as $P_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$.

At each time step, each particle moves toward *pbest* and *gbest* locations. The fitness function evaluates the performance of particles to determine whether the best fitting solution is achieved. The particles are manipulated according to the following equations:

$$\begin{aligned} v_{id} &:= v_{id} + c_1 * rand() * (p_{id} - x_{id}) \\ &\quad + c_2 * rand() * (p_{gd} - x_{id}), \end{aligned} \tag{12}$$

$$x_{id} := x_{id} + v_{id}, \tag{13}$$

where c_1 and c_2 are two positive constants, called cognitive learning rate and social learning rate respectively; $rand()$ is a random function in the range [0,1]. The velocity of the particles are limited in $[Vmin, Vmax]$. Since the original formula of PSO lacks velocity control mechanism, it has a poor ability to search at a fine grain [15]. A time decreasing inertia factor is designed by Eberhart and Shi to overcome this shortcoming in 1998 [16]:

$$\begin{aligned} v_{id} &:= w * v_{id} + c_1 * rand() * (p_{id} - x_{id}) \\ &\quad + c_2 * rand() * (p_{gd} - x_{id}), \end{aligned} \tag{14}$$

$$x_{id} := x_{id} + v_{id}, \tag{15}$$

where w is inertia factor which balances the global wide-rang exploitation and the local nearby exploration abilities of the swarm.

3.2. Improved PSO

In the former researches [9], most of them widely investigated on the improvement of the velocity update equation and imposed the limit on the velocity of particles. Few were mentioned about the limit on the positions of particles in their studies. If there is no limit imposed on positions, it is possible for particles to fly out of defined search space, which sometimes leads to invalid solutions. To confine particles in defined search space, common methods implemented in the code of traditional PSO are to check the validity of the positions of particles and then take some measures to rectify invalid solutions at every iteration. One rectification measure is to impose limit on positions, as it does on the velocity. If an element of the position is smaller then *Xmin*, it is set equal to *Xmin*; if greater than *Xmax*, then equal to *Xmax*. Another is to reject the invalid particle, and then repeatedly evaluate the velocity update equation, formula (12) and formula (14), until the position updating equation, formula (12), obtaining a valid solution. Though those measures can restrict particles in defined search space, at the same time, they bring some excess computation. An improved PSO with momentum factor is introduced to solve this disadvantage [17]. The new technique can limit the particles in defined search space without checking the boundary at every iteration. In improved PSO, the particles are manipulated by the following equations:

$$\begin{aligned} v_{id} &:= w * v_{id} + c_1 * rand() * (p_{id} - x_{id}) \\ &\quad + c_2 * rand() * (p_{gd} - x_{id}), \end{aligned} \tag{16}$$

$$x_{id} := (1 - mc) * x_{id} + mc * v_{id}, \tag{17}$$

where mc is momentum factor $(0 < mc < 1)$, and *Vmin* =*Xmin*; *Vmax*=*Xmax*.

4. OPTIMIZATION OF FOPID CONTROLLERS

Optimization of FOPID controllers firstly needs design the optimization goal, and then encode the parameters to be searched. PSO algorithm is running until the stop condition is satisfied. The particles of the last generation are the optimized parameters of the FOPID controller.

4.1. Representation of parameters

From (6), five parameters K_p, K_i, K_d, and λ, δ are required to be designed, according to control objectives. For the conventional PID controller design, we should ensure that all the poles of the close-loop transfer function are confined in the left half of the s plane. Based on the stability condition of fractional order system, a controller that stabilizes the integer order system stabilizes the fractional order system. The FOPID controller parameter can adopt the parameter of integer order controller and add fractional orders of integral and derivative. In this paper, the initial positions of the *i*th particles of the swarm can be represented by a 5 dimensional vector, and then the initial values are randomly generated based on the extreme values. The PSO technique is simple in encoding with real number, while GA with binary strings.

4.2. Selection of PSO factors

As a rule, PSO needs to predefine numerical coefficients (consisting of the maximum velocity or inertia weight, momentum factor, societal factor and individual factor), swarm size and topology. The ability of global optimization relies greatly on the setting of these parameters. The maximum velocity and inertia weight affect the ability of escaping from local optimization and refining global optimization. The societal factor and individual factor determine the ability of exploring and exploiting. The size of swarm balances the requirement of global optimization and computational cost. Lastly, the topology concerns both the ability of sharing information and the expense of communication.

The coefficients of PSO influence the optimization performance. In this paper, a time decreasing inertia weight w from 0.9 to 0.4, $c_1 = c_2 = 2.5$ with the addition of constant *mc* = 0.3 are adopted.

4.3. Fitness function

To evaluate the controller performance, there are always several criterions of control quality, which are given by:

$$J_1 = \int_0^\infty e^2(t)dt,\ J_2 = \int_0^\infty |e(t)|\,dt, \tag{18}$$

J_1 can track error quickly, but easily give rise to oscillation. J_2 can obtain good response, but its selection performance is not good. For getting good dynamic performance and avoiding large control input, the following control quality criterion is used in this paper,

$$J = \int_0^\infty (w_1 |e(t)| + w_2 u^2(t))dt. \tag{19}$$

The fitness function is given as follows:

$$F = {}^1\!/_J. \tag{20}$$

4.4. Stop criteria

The stop criteria used was the one that defines the maximum number of generations to be produced. When PSO algorithm runs, the new populations generating process is finished, and the best solution to complete the generation number is the one among the individuals better adapted to the evaluation function.

4.5. Simulation researches

The control objective is as follows:

$$G(s) = \frac{400}{s^2 + 50s}. \tag{21}$$

The sampling period is 0.001s. The control input is step signal. For reducing the time of optimization, the initial range of parameters are selected, these are $K_p \in [0,20]$, $K_i \in [0,1]$, $K_d \in [0,1]$, $\lambda \in [0,1]$, and $\delta \in [0,1]$. The fractional order controller can be digitally realized using the method of Tustin operator and continued fraction expansion, where sampling time is 0.001s; the approximation model order is 6. The population size of initial generation is 50. $w_1 = 0.999$, $w_2 = 0.001$; the maximum number of generations is set as 200. *Vmin* is set equal to *Xmin*; *Vmax* equal to *Xmax*.

Firstly, the standard PSO with gradually decreasing inertia factor is designed to optimize the parameters of FOPID controllers. The learning rate are $c_1 = c_2 = 2.5$; the inertia factor w decreases linearly between 0.9 and 0.4, it can calculate by the following equation:

$$w = (w_{max} - w_{min}) \times \frac{(Iter_{max} - Iter_{now})}{Iter_{max}} + w_{min}, \tag{22}$$

where $Iter_{max}$ is the maximum number of generations, $Iter_{now}$ is the current number of generations in the running PSO, so $Iter_{max} = 200$, $w_{max} = 0.9$, $w_{min} = 0.4$.

After 50 runs of the Standard PSO algorithm are performed, many optimized results are invalid and out of defined search space. As shown in Fig. 3, there are 16 optimized results of parameter λ out of the range[0,1]. At the same time, some optimized results of other parameters, including K_p, K_i, K_d and δ, also fly out of defined search range. So the Standard PSO with linear weight can not guarantee the valid solution.

With improved PSO, Its factors are set equal to the above standard PSO except momentum factor $mc = 0.3$. After 20 runs, all the results are in the predefined search range. The mean fitness value of the

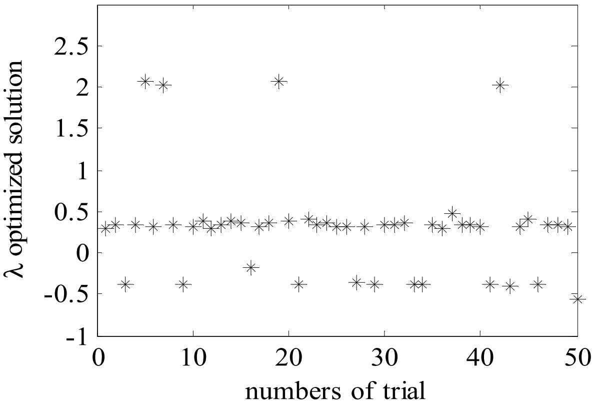

Fig. 3. Optimized results after 50 runs.

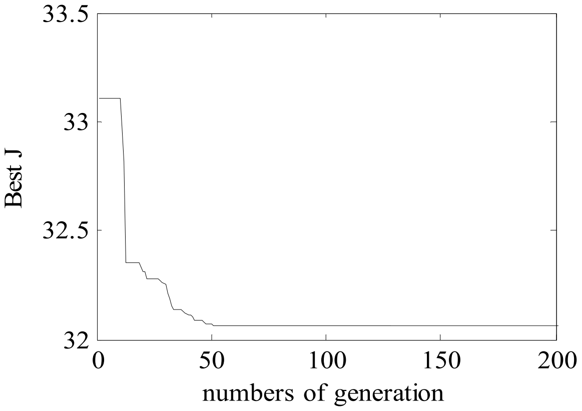

Fig. 4. Variety of performance function.

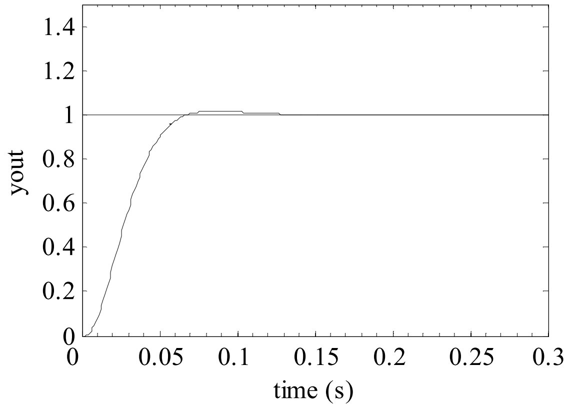

Fig. 5. Dynamic response after optimization.

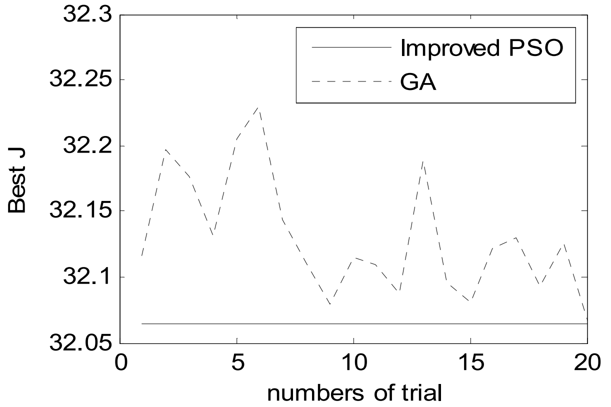

Fig. 6. Results using improved PSO and GA.

best solutions is 0.0312 and the mean value of the best objective function is $J = 32.063748$, one best solution is $K_p = 2.86$, $K_i = 0.000012$, $K_d = 1.0$, $\lambda = 0.00119$, $\delta = 0.4896$. The variety of performance function J of optimization process is shown as Fig. 4. The dynamic response using the optimized FOPID is shown as Fig. 5.

With binary genetic algorithms, FOPID controllers' parameters can be achieved for the same control objective (21). The Rank-based fitness assignment is selected in fitness calculations of reproduction operator; Crossover uses the binary valued uniform crossover with probability 0.6; Mutation adopts binary valued mutation with mutation probability 0.01. The population size of initial generation of numbers and the maximum number of generations are set equal to the optimization parameters of PSO. Through 20 runs, the mean fitness value of the best solutions is 0.0311, the best $J = 32.0671$. As shown in Fig. 6, the result of each run is more closely approach to the best solution using improved PSO in comparison with GA. To generally test the performance of improved PSO algorithms in designing fractional order controllers, various controlled plants, whose transfer function in appendix, are adopted. The objective function is also (19), and the factors of GA and the improved PSO are the same as the former. The optimization results of FOPID controllers for different plants using GA and the improved PSO algorithms can be shown as Table 1. Each result is the average of 20 runs with variance. The comparative results show the proposed PSO algorithms are preferable to GA in optimizing the parameters of FOPID controllers.

Table 1. The optimization results using GA and the improved PSO.

Plant	GA	Improved PSO
$G_1(s)$	20.8147 ± 0.0777	20.6685 ± 0.0065
$G_2(s)$	38.3095 ± 5.4925	34.1860 ± 6.8733
$G_3(s)$	68.9636 ± 0.0770	68.9091 ± 0.0413
$G_4(s)$	87.4638 ± 255.32	69.3680 ± 0.0105
$G_5(s)$	154.8318 ± 9731	95.2071 ± 42.136
$G_6(s)$	45.2443 ± 0.0002	45.2433 ± 0.0004

5. CONCLUSIONS

It has been demonstrated that the parameters optimization of fractional order controller based on modified PSO is highly effective. According to optimization target, the proposed method can search the best global solution for FOPID controllers' parameters and guarantee the objective solution space in defined search space. In contrast with GA method, the improved PSO can achieve faster search speed and better solution. Based on improved PSO, the design and application of FOPID will be appeared in various fields.

APPENDIX A

In the following, the transfer functions of controlled plants in Table 1 are described as follows:

$$G_1(s) = \frac{1553}{s^2 + 14s + 40.02}, \tag{23}$$

$$G_2(s)=\frac{523500}{s^3+83.75s+10470}, \tag{24}$$

$$G_3(s)=\frac{250s+500}{s^3+12s^2+100s}, \tag{25}$$

$$G_4(s)=\frac{300(s+100)}{s(s+10)(s+40)}, \tag{26}$$

$$G_5(z)=\frac{0.027z^3+0.0434z^2+0.0025z+0.0012}{z^4-0.5142z^3-0.6948z^2-0.0045z-0.0074}, \tag{27}$$

$$G_6(s)=\frac{4\times10^7}{s(s+250)(s^2+40s+9\times10^4)}. \tag{28}$$

REFERENCES

[1] I. Petras, L. Dorcak, and I. Kostial, “Control quality enhancement by fractional order controllers,” *Acta Montanistica Slovaca*, vol. 3, no. 2, pp. 143-148, April 1998.

[2] C.-B. Ma and Y. Hori, “Backlash vibration suppression in torsional system based on the fractional order Q-filter of disturbance observer,” *Proc. of the 8th IEEE International Workshop on Advanced Motion Control*, pp. 577-582, 2004.

[3] N. M. F. Ferreira and J. A. T. Machado, “Fractional-order hybrid control of robotic manipulators,” *Proc. of the 11th International Conf. on Advanced Robotics*, pp. 393-398, 2003.

[4] A. Kailil, N. Mrani, M. M. Touati, S. Choukri, and N. Elalami, “Low earth-orbit satellite attitude stabilization with fractional regulators,” *International Journal of Systems Science*, vol. 35, no. 10, pp. 559-568, August 2004.

[5] I. Petras, “The fractional order controllers: Methods for their synthesis and application,” *Journal of Electrical Engineering*, vol. 50, no. 9-10, pp. 284-288, October 1999.

[6] B. M. Vinagre, I. Podlubny, L. Dorcak, and V. Feliu, “On fractional PID controllers: A frequency domain approach,” *Proc. of IFAC Workshop on Digital Control - Past, Presentand Future of PID Control*, pp. 53-58, 2000.

[7] L. Dorcak, I. Petras, I. Kostial, and J. Terpak, “State-space controller design for the fractional-order regulated system,” *Proc. of the International Carpathian Control Conference*, pp. 15-20, 2001.

[8] C.-B. Ma and Y. Hori, “Fractional order control and its application of fractional order PID controller for robust two-inertia speed control,” *Proc. of the 4th International Power Electronics and Motion Control Conference*, pp. 1477-1482, 2004.

[9] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” *Proc. of the IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.

[10] M.-P. Song and G.-H, Gu, “Research on particle swarm optimization: A review,” *Proc. of the Third International Conf. on Machine Learning and Cybernetics*, pp. 2236-2241, 2004.

[11] A. Carpinteri and F. Mainardi, *Fractals and Fractional Calculus in Continuum Mechanics*, Springer-Verlag, Wien and New York, pp. 223-276, 1997.

[12] I. Podlubny, “Fractional-order systems and $PI^{\lambda}D^{\delta}$ controllers,” *IEEE Trans. on Automatic Control*, vol. 44, no. 1, pp. 208-213, January 1999.

[13] Y.-Q. Chen and K. L. Moore, “Discretization schemes for fractional-order differentiators and integrators,” *IEEE Trans. on Circuits and Systems*, vol. 49, no. 3, pp. 363-367, March 2002.

[14] I. Petras, Y.-Q. Chen, and B. M. Vinagre, “A robust stability test procedure for a class of uncertain LTI fractional order systems,” *Proc. of International Carpathian Control Conference*, pp. 247-252, 2002.

[15] P. J. Angelinc, “Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences,” *Proc. of the Seventh Annual Conference on Evolutionary Programming*, pp. 601-610, 1998.

[16] Y. Shi and R. C. Eberhart, “A modified particle swarm optimizer,” *Proc. of the IEEE Congress on Evolutionary Computation*, pp.69-73, 1998.

[17] Y. Liu, Z. Qin, and X.-S. He, “Supervisor-student model in particle swarm optimization,” *Proc. of the IEEE Congress on Evolutionary Computation*, pp. 542-547, 2004.

Jun-Yi Cao received the B.S. and M.S. degrees in Mechanical Engineering from Northwest Institute of Textile Technology in 2000 and 2003, respectively. He is currently working towards a Ph.D. degree at the Department of Mechanical Engineer-ing, Xian Jiaotong University, China. His research interests include fractional order control, system optimization and motion control.

Bing-Gang Cao received the B.S., M.S., and Ph.D. degrees in Mechanical Engineering from Xi’an Jiaotong University in 1976, 1982, and 1992, respectively. He is currently a Professor at the Department of Mechanical Engineering, Xian Jiaotong University, where he is also the Director of Electric Vehicle Center of Research and Development. His research interests include robust control, intelligent control of mechatronical system, noise and vibration control of liquid system, control technology of renewable energy.