Adaptive Particle Swarm Algorithm for Parameters Tuning of Fractional Order PID Controller

Chao-bo Chen¹, Li Hu¹, Lei Wang¹, Song Gao*,¹, Chang-hong Li²
¹School of Electronics Information Engineering, Xi'an Technological University, Xi'an 710021, China
²Northwest Institute of Mechanical & Electrical Engineering, Xianyang 712099, NO.2 Xuefuzhong Avenue Weiyang District Xi'an, Shaanxi Province, P.R. China
*Corresponding author, e-mail: gaos@xatu.edu.cn

Abstract

In order to optimize the parameters of fractional order PID controller of complex system, an adaptive particle swarm optimization (PSO) method is proposed to realize the parameters adjustment. In this algorithm, the tuning particle population is divided into three subgroups firstly, and through introducing the swarm-aggregation degree factor and the evolution speed factor of particle, dynamically adjusting the inertia weight and size of subgroups respectively, setting to find optimal objective according to the time-domain performance index of the system, and then the controller parameter tuning is realized by iterative calculation. Finally, adaptive particle swarm optimization method of fractional order PID controller is applied to integer order and fractional order of the controlled system for performance simulation in time domain analysis. The experimental results show that the proposed method could improve the performance of the control system and has strong anti-interference ability.

Keywords: Fractional order PID controller, Adaptive particle swarm optimization, Parameter tuning, Self-adjusting

1. Introduction

The fractional order PID (POPID) algorithm [1] is powerful and widely used for precise analysis of the dynamic response of the fractional order PID control system performance, which was proposed by professor Podlubny I. in 1999. Although there are no clear conceptions of the fractional order calculus geometric explanation and the physical meaning, the fractional order controller has better flexibility and control accuracy and robustness than integer order PID controller in application of control system. Therefore, the fractional order theory research and optimizing designation and application with intelligent algorithms of fractional PID controller have become one of the hot research areas, simultaneously have obtained some remarkable achievements in recent years.

In the reference [2], an improved differential evolution (DE) algorithm is used to optimize the parameters of FOPID in order to obtain higher time-domain performance of the system. In general, it is hard to find an excellent optimization methodology for solving the optimal PID controller. The fractional controller can also be synthesized by a designed method based on Bode ideal transfer function, it can make the control system a good dynamic performance and robustness to parameter variation in reference [3]. Aiming at the high precision digital fractional calculus and FOPID controller in practical application of complex engineering system, a tuning method based on optimal Oustaloup fractional order PID parameters is put forward and obtains the good control effect in the reference [4]. In the reference [5], the optimal parameters are obtained by setting the fractional order PID controller according to the overshoot and adjusting time, and because of the optimal parameter values based on the maximum sensitivity index, the overall performance of the closed-loop system is optimal. In the reference [6], a FOPID controller with internal model is designed to improve tracking performance, disturbance rejection and robustness properties of the system. Most researchers combine many optimization techniques with intelligent methods for optimal control parameters and better system predictable performance. In the reference [7], function parameters can be predicted through applying particle swarm optimization to FOPID, and the higher control accuracy and the stronger adaptability of the fractional order PID controller has been improved. But despite the controller...
which can improve the performance of the control system, those methods are still under parameter variation in the computational complexity of coordination system which are not good enough with high control precision, real-time performance.

As we all know, the fractional order PID controller has such advantages over classical counterparts on flexible control in time-domain and frequency-domain specifications. However, it is necessary to solve the problem of high dimension variables, complex tuning process and system uncertainties. Moreover, as the particle swarm optimization algorithm (PSO) can effectively solve highly nonlinear, discontinuous, non-differential and multi-extremes optimization problems, it is simple and easy to implement parameters dynamic regulation but being trapped in local optimum that caused premature convergence, then, the control system properties could not reach an optimum time-domain performance.

In this paper, an adaptive parameter tuning method has been introduced into particle swarm optimization (PSO) of fractional order PID controller, an ensemble of particles is divided into three sub populations according to its fitness value, then aggregation degree factor and degree of evolutionary factor are introduced to rectify adaptively on the inertia weight and the size of subgroups respectively. The global optimized solution would be realized by the iterative calculation through the optimization objective function of the system. This algorithm is employed to design an FOPID controller that simulated within different scenarios and its property is compared with those of optimally designed FOPID controllers. The results show that the designed fractional order PID controller is able to enhance the real time property, high precision and robustness of the system.

The rest of this paper is organized as follows. Section 2 and 3 describes the rudiments of fractional calculus, fractional-order controller and basic PSO algorithm. Adaptive particle swarm optimization algorithm was showed in detail and the design procedure of FOPID controller expressed in section 4. Section 5 presented and discussed the simulation strategies and experimental results. Section 6 concludes the paper.

2. Fractional Order PID
2.1. Fractional-order Calculus

For the study of fractional order system, fractional calculus is a basic starting point and a branch of calculus in which the differentiation and integration of a function is generalized to non-integer order. The fundamental Integral-differential operator is \(D^\alpha\) [8], in which \(a\) and \(t\) is respectively the upper and lower limitation of the operation. \(\alpha \in \mathbb{R}\), is defined as:

\[
a D^\alpha_t = \begin{cases} 
\frac{d^n}{dt^n} t^\alpha \quad & R(\alpha) > 0 \\
1 \quad & R(\alpha) = 0 \\
\int_0^t (\tau)^{\alpha} d\tau \quad & R(\alpha) < 0
\end{cases}
\]

(1)

There are several different definitions for fractional derivatives. The Grünwald-Letnikov (GL) definition is given by:

\[
a D^\alpha_t f(t) = \lim_{h \to 0} h^{-\alpha} \sum_{j=0}^{[t/a]} (-1)^j \binom{\alpha}{j} f(t - jh)
\]

(2)

Where, \([.]\) means the integer part. GL is a unified expression of FOC, which is based on the definition of integer order differential, and extended to the fractional order. The Riemann-Liouville (RL) definition can be written as:

\[
a D^\alpha_t f(t) = \frac{1}{\Gamma(m-\alpha)} \frac{d^m}{dt^m} \int_0^t (t-\tau)^{-(m-\alpha)} d\tau
\]

(3)

In particular, the choice of \(\alpha\) derives \(m-1\) to \(m\) and \(m \in \mathbb{N}\), as well as \(\Gamma(\cdot)\), is the usual Euler’s gamma function. The integral definition of RL is the positive non-integer \(\alpha\) order.
derivative of the function $f(t)$, and the $m - \alpha$ order integral is advanced in the $m$. Similarly, the Caputo definition is in the form of:

$$\alpha D^\alpha_t f(t) = \frac{1}{\Gamma(m - \alpha)} \int_0^t \frac{f^{(m)}(\tau)}{(t - \tau)^{\alpha - m + 1}} d\tau$$

(4)

Where, $m - 1 < \alpha < m$, $m \in N$.

2.2. Approximation of Fractional Calculus Operator

Fractional order system is theoretically unlimited and its characteristic equation is generally a pseudo polynomial with a complex variable, so it is used to describe the fractional order system with finite differential equations. In order to implement a fractional order control law, Oustaloup’s band-limited frequency domain rational approximation technique is used in reference [9]. However, few recent research results show that band-limited implementation of fractional order PID controllers using higher order rational transfer function approximation of the integral-differential operators give a satisfactory properties in industrial automation.

Set induce Improved Oustaloup approximation method for calculus operator $s^\alpha$, select a fitting finite frequency section $[\omega_b, \omega_h]$ and the approximation order $2j + 1$.

$$s^\alpha = C * p(s) \prod_{k=1}^j \frac{s + \omega_k^\alpha}{s + \omega_k^b}$$

(5)

Where, $\alpha \in R^+$, the poles, zeros, and gain of the filter can be recursively evaluated as:

$$C = \left(\frac{d \omega_b}{b}\right)^\alpha, \quad p(s) = \frac{d s^2 + b \omega_b s}{d (1 - \alpha)s^2 + b \omega_b s + d \alpha}$$

(6)

Here, $b$ and $d$ are two added adjustable parameters.

$$\omega_k^\alpha = \omega_b \left(\frac{\omega_b}{\omega_b}\right)^{\frac{k + j}{2j + 1}(1 - \alpha)}$$

(7)

$$\omega_k^b = \omega_b \left(\frac{\omega_b}{\omega_b}\right)^{\frac{k + j}{2j + 1}(1 - \alpha)}$$

(8)

Thus, arbitrary signal $f(t)$ can be passed through the higher order analog filter (5) and its output can be considered as an approximation to the fractional differentiated or integrated signal $D^\alpha f(t)$.

2.3. Fractional PID Controller

Fractional order differentiators and integrators are products of fractional calculus-the theory of differentiation and integration in arbitrary order. To date, fractional order controllers are mainly classified of 4 kinds of controllers-fractional PID controller [1], TID controller [10], CRONE controller [11] and lead lag compensator [12]. Fractional Order PID controller proposed by Podlubny is more representative. Compared with the traditional integer order PID controller, this controller has two more variable parameters, which make the parameter setting more complex. In general, the parameters are $K_p$, $K_i$, $K_d$, integral order $\lambda$ and differential order $\mu$.

Because of the added two parameters, fractional calculus performs more effectively for the controller design than integer order calculus with arbitrary integral and derivative orders of real number. Taking $\lambda = 1$ and $\mu = 1$, we will have an integer order PID controller. In a graphical way, the control possibilities using a FOPID controller is shown in Figure 1, extending the four
points of the classical PID to the range of control points of quarter-plane block limited by the values of \( \lambda \) and \( \mu \).

![Derivation order](image)

Figure 1. Fractional-order PID plant

The fractional order PID makes the parameters governed by the system in the entire plane of the PID controller. The parameters are adjusted through fractional PID control according to the linear combination of the proportional, differential and integral of the input and output of the controller. The expression of the transfer function is as follows:

\[
G(s) = K_p + \frac{K_d}{s^\mu} + K_i s^\lambda
\]

Where, the orders of integration and differentiation are respectively \( \lambda \) and \( \mu \) (both positive real numbers, not necessarily integers).

In Figure 2, \( G(s) \) is as the transfer function of the controlled object. \( R(s) \) and \( Y(s) \) are the input and output of the system, \( U(s) \) is the output of the controller.

![Structure of the FOC system](image)

Figure 2. Structure of the FOC system

### 3. Basic PSO Algorithm

Particle swarm optimization algorithm [13-15], which is proposed by Kennedy and Eberhart in 1995, is a swarm intelligence optimization algorithm. The PSO algorithm is used to attempt to imitate the behavior of the birds in the biology, and to achieve the goal of finding the optimal solution through the continuous iteration of the particles. In the optimization process of PSO, the performance of each particle depends on the fitness value of the objective function of the optimization problem, and the movement position depends on the precious position and present speed of the particle[16]. Particles search the current optimal value particle direction in the range space. Particles can be represented as an N dimensional vector and the position and velocity of the \( i \)-th particle are given respectively by:

\[
X_i(k) = [X_{i1}(k), X_{i2}(k), ... , X_{iN}(k)]
\]

\[
V_i(k) = [V_{i1}(k), V_{i2}(k), ... , V_{iN}(k)]
\]
Where, \( k \) is the generation number. In each iteration of PSO, each particle is updated through following two extreme values. One is the individual extreme point (pbest position) that particles find the best precious solution themselves, the other one is called global extreme value point (gbest position) which can let the whole population find the optimal solution. The particles update their positions and velocities according to the following formula:

\[
V_i(k+1) = w(k+1)V_i(k) + c_1r_1(P_i(k) - X_i(k)) + c_2r_2(P_g(k) - X_i(k))
\]

\[\text{(12)}\]

\[
X_i(k+1) = X_i(k) + V_i(k+1)
\]

Where, \( w \) is inertial weight that balances the global wide-range exploitation and the local nearby exploration abilities of the swarm. \( c_1 \) and \( c_2 \) represent velocity coefficients which are positive constants, \( r_1 \) and \( r_2 \) are two random functions in the range \([0,1]\), \( P_i \) and \( P_g \) are the personal best and global best respectively.

Compared with other optimization algorithms, the PSO algorithm can achieve the optimization goal with a shorter time and better convergence performance. However, the particle swarm optimization algorithm also has the phenomenon of premature convergence. In the search process, it is easy to converge to the local optimum for loss of population diversity. Therefore, in this paper, the aggregation degree factor and evolutionary degree factor are introduced to detect population aggregation extent and speed of evolution, which can increase the global search speed and the diversity of the population when the population aggregation degree becomes higher and evolutionary rate changes rapidly.

4. Adaptive PSO Algorithm Turning Fractional Order Controller

For implementation of standard PSO, this paper induces the adaptive particle swarm optimization (APSO) algorithm that applied to reduce the probability of searching the local optimal solution with the iterative process so that more particles search the larger inertia weight on the range space.

4.1. Population Initialization

The random initialization of population, which reduces the efficiency of the search of the global optimal solution, is not beneficial to the global search for the particle swarm optimization. Some studies [17-20], proved that using the good point set method is beneficial to search the global optimization of the initial particle swarm. Thus, the good point set method is used in this paper to initialize the random particle swarm so that the solution parameters distribute in the whole search space. Meanwhile, the probability of the local optimal solution cannot be difficult to find.

4.2. Optimization Objective Function

Essentially, parameter tuning is the parameters optimization method based on the given indexes. In general, there are two kinds of control performance indexes, which are deterministic and robust ones. In this paper, the objective function is the same as the fitness function. In order to obtain the satisfactory dynamic characteristics of controlled object, the function was added the time integral performance index of the error absolute value and the square part of the input of the controller to prevent overmuch control volume in the fitness function. Furthermore, the penalty function was added to the target function to suppress the overshoot of the system, when it was over regulated. Fitness function is given as in Equation (14).

\[
\text{Fitness} = \int_{t_0}^{t_{\text{max}}} (\beta_1|e(t)| + \beta_2u^2(t))\,dt + \beta_3\text{sigma}
\]

Where, \( e(t) \) is sampling error value, \( \text{sigma} \) is overshoot, and \( u(t) \) is the input of the controller, \( \beta_1 \), \( \beta_2 \) and \( \beta_3 \) are the weight coefficients of each performance index, \( imax \) is the maximum number of iteration.
4.3. Aggregation Factor and Evolution Speed

In the particle swarm optimization algorithm, the particle is easy to fall into local optimal solution. In reference [21], PSO algorithm meet the needs of control precision of multi-leaves’ position, obviously, control index of high precision depends on the the value of objective function. This paper introduces the aggregation factor to reflect the diversity of particle swarm and find the minimum value of fitness function. The aggregation factor is defined as:

\[ J_d = \frac{1}{N} \sum_{t=0}^{N} \frac{F(x(t))}{F(x_g(t))} \]  

Where, \( F(x_g(t)) \) is the current optimal fitness value of all particles, \( F(x(t)) \) is the current optimal fitness value of the i-th particle and \( 0 \leq jd \leq 1 \). In Equation (15) equation, the closer of the value of \( J_d \) is to 1, the higher of the aggregation degree of the particles becomes. Similarly, evolution speed factor reflects the evolution extent of the population, which is given as:

\[ s_d = \frac{F(x_g(t))}{F(x_g(t-1))} \]  

Where, \( s_d \) ranges from zero to one. The closer the evolution speed factor is to 1, the slower the evolution rate is.

4.4. Adaptive Adjustment of the Inertia Weight of the Sub-populations

In our adaptive algorithm, the particle swarm is divided into three sub-groups based on the fitness value of the particle swarm, each of which was employed with different weight values, respectively. For the three subgroups, their computational formulas are listed below (17)-(19).

\[ w_1(k) = 0.9 - 0.5 \frac{k}{i_{\text{imax}}} \]  

\[ w_2(k) = (1 + 0.5j_d)w_1(k) \]  

\[ w_3(k) = (1 + j_d)w_1(k) \]  

Where, \( k \) is the current iteration number. When the population aggregation degree is greater, i.e. the diversity is less, the inertia weights of the 2nd and 3rd sub-population will be adjusted to the bigger one. Thus, the speed of the population evolution could be accelerated for avoiding particle swarm into local optimal solution.

4.5. Adjustment of the Subgroup Size

Firstly, the particle swarm optimization was initialized with the good point set method, so that it can be uniformly distributed in the search space. The population was subdivided into three subgroups by the fitness value of the population. The individual number of each group was assigned with a random position, and the random velocities is \( n_1, n_2 \) and \( n_3 \). \( N \) is the size of the population, that \( N \) equals the sum of \( n_1, n_2 \) and \( n_3 \). \( N \) is the size of the population, that \( N \) equals the sum of \( n_1, n_2 \) and \( n_3 \). Next, each individual of the population would find their own fitness value based on the corresponding fitness function. The \( n_1 \) individual, which is the optimal value of fitness values, is used as group 1. The fitness value of \( n_2 \) is 2, and the fitness value which is inferior to the \( n_3 \) individual is set as subgroup 3 in their iterative processes. Likewise, the evolution speed of the particle swarm was calculated and the size of the population was adjusted dynamically according to the evolution of the particle swarm. The numbers of subgroups were calculated according to the following formula.

\[ n_1 = (1 - s_d)N/3 \]  

\[ n_2 = N/3 \]  

\[ n_3 = (1 + s_d)N/3 \]
When the evolution speed of the particle swarm grew slowly, which led to increasing of subgroup numbers, we need focus on the global search in order to avoid the particle swarm optimizing to the local optimal solution.

4.6. Adaptive PSO for FOPID Steps

The PSO algorithm attempts to mimic the natural process of group communication of individual knowledge, which occurs when the social swarm elements flock, migrate, and forage, etc. Our approach is based on this basic method to design the non-integer PID controllers for searching the optimal or near optimal controller parameters, i.e. $K_p$, $K_i$, $K_d$, $\lambda$, and $\mu$. Each individual $X$ contains five parameters of FOPID controllers. The working of APSO algorithm for FOPID is explained in the following procedures:

Procedure 1: A good point set method of particle swarm initialization. It includes particle swarm scale, search range, maximum speed, inertia weight, accelerated coefficient and maximum number of iterations, and the particle velocity and position, etc.

Procedure 2: According to the fitness function defined previously, the fitness value of each particle is calculated.

Procedure 3: For each particle in the sub-population, compare its current fitness value with the individual optimal fitness value. If its value is better, update individual optimal fitness value with its best value. Similarly, compare its fitness value with the global optimal fitness value, then update the global optimal fitness value with the best one.

Procedure 4: Based on formulas (20), (21) and (22), round the numbers of subgroups, and classify particles in the swarm into the corresponding subgroups.

Procedure 5: Modify the inertia weight of particle swarm using Equation (17), (18) and (19).

Procedure 6: Update the current velocity and position of the particle in each iteration by the formula (12) and (13) respectively.

Procedure 7: If it meets the ending condition, then exit. Otherwise return to Procedure 2 and continue the iteration.

The individual that generates the latest global value is the optimal controller parameter.

5. Experimental Simulation and Results

We chose the integer order PID controller’s and fractional order PID controller’s objects to carry on the simulation experiment in order to verify the superiority of the algorithm. In this paper, the parameter of the simulation experiment is $n = 60$, the maximum iteration step size is $i_{\text{max}} = 100$, the maximum rate of iteration $v_{\text{max}}$ is the maximum speed of evolution of the solution parameter, $0 < v_{\text{max}} \leq (u_d - l_d)/5$, where $u_d$ and $l_d$ are the upper bound and lower bound of the search space. A single integer order controlled object selected from [22] and its transfer function is:

$$G_{p1}(s) = \frac{1}{4.32s^2 + 19.1801s + 1}$$

(23)

According to the parameter adaptive differential evolution (PSA-DE) algorithm proposed by [22], the integer order PID controller $G_1$ and fractional order PID controller $G_2$ are got respectively.

$$G_1 = 175.9591 + \frac{3.8301}{s} + 25.9113s$$

(24)

$$G_2 = 201.6255 - \frac{63.9625}{s^{0.4088}} + 26.9022s^{0.6088}$$

(25)

Where, select the search range for $K_p \in [0.200], K_i \in [0, 50], K_d \in [0.50], \mu \in [0, 2], \lambda \in [0.2], c_1 = c_2 = 2$, the FOPID controller is obtained by applying the adaptive particle swarm optimization method mentioned in this paper and its transfer function is:
\[ G_3 = 184.1678 + \frac{17.5974}{s^{0.326}} + 38.112s^{0.923} \]  

In Figure 3, the unit-step response of the closed-loop system with plant model \( G_{p1} \) is controlled by the conventional PID controller \( G_1 \) and non-integer PID controller in \( G_2 \) and \( G_3 \). The comparison shows that satisfactory feedback control performance of fractional-order system is better to use a conventional PID controller. For the IOPID controlled process, the maximum peak overshoot is 7.6\%, the rise time and the settling time are 0.336 and 1.17 seconds. For the FOPID with PSA-DE algorithm controlled process, the maximum peak overshoot is 2.9\%, the rise time is 0.371 seconds and the settling time is 0.697 seconds. However, the method applied by the FOPID mentioned in this paper controlled process, the overshoot is less than 1\%, the rise time is only 0.252 seconds and the settling time decreased to 0.341 seconds. Besides, the steady-state error jumped to 0.28\% over the full scale [0.32\%, 1.14\%] (the steady-state error are 0.32\% used \( G_2 \) controller and 1.14\% used \( G_3 \) controller, respectively). Also, we plotted the time response for unit step input in Figure 3 for uncontrolled system open-loop response and found it unsteady.

![Figure 3. Comparison of Unit Step Response of \( G_{p1} \) With Different Controllers](image1)

As is shown from Figure 3, system controlled by the fractional order PID controller has good time-domain performance, especially the maximum peak overshoot and rapidity. In order to test the robustness of these controllers, we introduced the load disturbance of amplitude 0.2 after 2 seconds in Figure 4, the unit step response of the control system is:

![Figure 4. Comparison of Disturbance Response of \( G_{p1} \) With Different Controllers](image2)
In Figure 4, it can be seen that the $P^\lambda D^\mu$ proposed in this paper could give the best convergence characteristics and strongest immunity. In addition, the step response under the same conditions yielded by PSA-DE and improved APSO in this paper are highlighted in bold. Results show that all responses in the slope of the curve, the blue curve is the highest (G3-POPID Controller), followed by the fuchsia dotted line (G2-POPID Controller), and finally the solid red line (G1-PID Controller). It is proved that G3-POPID controller made the system more stable and have strong anti-interference ability.

The fractional order model in reference [23] is used to set the parameters of fractional order PID controller, and the transfer function is defined as:

$$G_{p2} = \frac{1}{0.88s^{2.2} + 0.5s^{0.9} + 1}$$  \hspace{1cm} (27)

The fractional order PD controller in [24] by Podlubny et al. is designed for the system.

$$G_{cf1}=20.5+3.7343s^{1.15}$$  \hspace{1cm} (28)

In [25], Xue Dingyu et al. designed a fractional PID controller for the system using the method of amplitude margin and phase margin and the controller is written as:

$$G_{cf2}=138.1817+2.8914s^{0.2}+12.3820s^{1.1}$$  \hspace{1cm} (29)

For the controller using Bode ideal transfer function method for the fractional order PID controller in [26] written by He Yiwen and his brethren is defined as:

$$G_{cf3}=18.92s^{0.87}+10.77s^{0.43}+21.54s^{1.33}.$$  \hspace{1cm} (30)

The selection of search range for $K_p$ is from 0 to 200, $K_i \in [0,50]$, $K_d \in [0,50]$, $\lambda \in [0,2]$, $\mu \in [0,2]$ , $c_1 = c_2 = 2$, then, fractional order PID controller can be obtained by the APSO method.

$$G_{cf4}=36.135+51.2158s^{0.9278}+30.011s^{1.1388}.$$  \hspace{1cm} (31)

These four controllers are applied to the fractional order control systems, and the step responses are compared in Figure 5.

![Figure 5. Comparison of unit step response of $G_{p2}$ with different controllers](image-url)

By comparing the unit-step response of the same controlled plant $G_{p2}$ with different four controller methods- the fractional PID controller designed by [21], the controller based on the ITAE criteria used in [22], the FOPID controller designed by the Bode ideal transfer function method applied to the [23], all the time-domain performance indexes are improved and have better control effects by using the controller designed with the method mentioned in this paper.
As is shown from Figure 5, the results of the simulation show that the output of the proposed controller has the shortest rise time and settling time, smallest steady-state error and the minimum of overshoot. Note, however, that for the given common performance criteria on the maximum peak overshoot and settling time, the other three fractional order controllers achieve better results except the controller in equation (15) on the rise time performance index, they also make the steady-state error of control system smaller. Finally, we plot the time response for step input from control system controlled by different controllers with best property indexes – the rise time (0.029s), the maximum peak overshoot (1.3%), settling time (0.036s) and steady-state error (0.001).

The system has been designed tested by one type disturbance of amplitude 0.2 at sample time being 2 seconds, the step response of the comparison control system is presented in Figure 6.

![Figure 6. Comparison of disturbance response of Gp2 with different controllers](image)

The main advantage of the fractional order controller is the robustness of the system whenever a disturbance and uncertainty occurred in the parameters. Figure 6 shows the comparison of unit-step and perturbation response, system controlled by Gcf1-FOPD controller has the slowest speed to recover the original one, system controlled by Gcf2-FOPID controller caused the maximum amount of disturbance and the system under the Gcf2-FOPID controller used by He et al. drove the system performance between these two fractional order controllers. The simulation results for various non-integer controllers with disturbance putting described in Figure 6 proved that the method proposed in this paper is effective and makes the system recover stable faster with strong anti-interference ability.

6. Conclusion

Fractional order PID controller design is flexible, but the parameters tuning is complex. In this work the APSO algorithm was utilized to find the five optimal parameters of FOPID controller which performs the efficient search that was tested on spreading in steps. It is clear from the simulation results that the novel design method, which was assisted with the introduction of aggregation degree factor and the evolution speed derived from the adaptive adjustment of the inertia weight and the number of subgroups, can obtain higher quality solution more easily and quickly compared with the basic PSO approach. In addition, application of the method to two integer order plants with diverse controllers simulation respectively showed that the proposed algorithm can make the control system favorable time domain properties and stronger anti-interference performance.

Acknowledgements

Work presented in this paper was partially supported by the Shaanxi International Cooperation Key Projects (Grant No.2015KW-024) and Weapons Research Support Fund of China(Grant No.62201070317).
References


