IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 25, 2020, accepted December 31, 2020, date of publication January 5, 2021, date of current version January 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049421

Optimal Design and Simulation for PID Controller Using Fractional-Order Fish Migration Optimization Algorithm

BAOYONG GUO[1], ZHONGJIE ZHUANG[2], JENG-SHYANG PAN[2], (Senior Member, IEEE), AND SHU-CHUAN CHU[2,3]

[1]College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China
[2]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[3]College of Science and Engineering, Flinders University, Clovelly Park, SA 5042, Australia

Corresponding author: Shu-Chuan Chu (scchu0803@gmail.com)

ABSTRACT Proportional Integral Derivative (PID) controller is one of the most classical controllers, which has a good performance in industrial applications. The traditional PID parameter tuning relies on experience, however, the intelligent algorithm is used to optimize the controller, which makes it more convenient. Fish Migration Optimization (FMO) is an excellent algorithm that mimics the swim and migration behaviors of fish biology. Especially, the formulas for optimization were obtained from biologists. However, the optimization effect of FMO for PID control is not prominent, since it is easy to skip the optimal solution with integer-order velocity. In order to improve the optimization performance of FMO, Fractional-Order Fish Migration Optimization (FOFMO) is proposed based on fractional calculus (FC) theory. In FOFMO, the velocity and position are updated in fractional-order forms. In addition, the fishes should migration back to a position which is more conducive to survival. Therefore, a new strategy based on the global best solution to generate new positions of offsprings is proposed. The experiments are performed on benchmark functions and PID controller. The results show that FOFMO is superior to the original FMO, and the PID controller tuned by FOFMO is more robust and has better performance than other contrast algorithms.

INDEX TERMS Fish migration optimization, fractional calculus, PID controller, swarm intelligence.

I. INTRODUCTION

As is known to all, PID is one of the earliest control strategies. Since its simple structure, good robustness, and high reliability, PID controller plays an important role in the closed industrial system [1], [2]. The PID controller is designed based on the error of the system, which uses proportion, integral, and differential to calculate the control quantity in order to achieve excellent performance. The traditional tuning methods include Ziegler-Nichols and Hägglund-Aström, etc. However, the researchers have proposed various intelligent tuning techniques in last few decades, including the methods based on genetic algorithm, fuzzy reasoning, and neural network. The traditional PID algorithms have low efficiency and no prominent effect, while the methods based on intelligent algorithms can achieve high efficiency and have a good effect [3], [4]. Soft computing is an effective intelligent algorithm to tune PID parameters. The common techniques of soft computing include fuzzy logic, neural networks, probability reasoning, and meta-heuristic algorithms, etc. Fuzzy logic is a science based on multi-valued logic that uses fuzzy sets to study fuzzy thinking, language forms, and their laws [5]. Neural networks process information by adjusting the interconnections between a large number of internal nodes [6], [7]. Probabilistic reasoning is a form of reasoning that people make decisions based on uncertain information [8]. Meta-heuristic algorithm is an improvement of heuristic algorithm, it is the product of combining random algorithm and local search algorithm.

In particular, influenced by bionics, meta-heuristic algorithms were generated in the 1960s. Meta-heuristic algorithms can obtain optimal solutions without having any

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Da Lin.

specific requirements [9], [10]. Furthermore, meta-heuristic algorithms can be separated into three categories including swarm intelligence algorithms, evolutionary algorithms, and algorithms based on mathematical or physical models. Among them, swarm intelligence algorithms are inspired by the ethology of group animals. For example, Particle Swarm Optimization (PSO) [11]–[14] is inspired by birds' foraging behavior. Ant Colony Optimization (ACO) [15], [16] mimicked the behavior of ants in finding the path in the process of searching for food. Bat Algorithm (BA) [17], [18] imitated the echolocation behavior of bats. Artificial Bee Colony (ABC) [19], [20] simulated the honey gathering process of bees, this algorithm has a fast convergence speed to find global optimal solution. Cat Swarm Optimization (CSO) [21], [22] depicted the cats' search and tracking strategy. Grey Wolf Optimization (GWO) [23], [24] is an optimal search method which is designed by the gray wolves' predation activities. Cuckoo Search Algorithm describes the parasitism behavior of cuckoo birds [25], [26]. Pigeon Inspired Optimization (PIO) simulated the behavior of pigeons going home [27]. Grasshopper Optimisation Algorithm (GOA) is proposed based on the behaviour of grasshopper swarms in nature for solving optimisation problems [28]. However, evolutionary algorithms are inspired based on the theory of biological evolution in nature, including Genetic Algorithm (GA) [29]–[31], Differential Evolution (DE), [32], [33], and QUasi-Affine TRansformation Evolutionary (QUATRE) [34]–[37], etc. GA is designed based on natural selection and genetic mechanism of Darwinian biological evolution. DE is a heuristic random search algorithm based on population difference. QUATRE is an excellent algorithm which improved the drawback of DE that did not achieve equilibrium search in search space without prior knowledge and moveover, it generalized the crossover operation of DE from vector to matrix. In algorithms based on physical or mathematical models, Simulated Annealing (SA) [38], [39] originates from the principle of solid annealing; Gravitational Search Algorithm (GSA) [40], [41] mainly uses the law of gravitation between two objects to guide the motion optimization of each particle to search for the optimal solution; Sine Cosine Algorithm (SCA) [42], [43] is achieved by iteration of sine and cosine functions.

Particularly, Fish Migration Optimization (FMO) [44], [45] is proposed in 2010 which is a swarm intelligence algorithm. It simulated the growth, migration processes, and predation strategy of fish biology. The difference of the FMO and other meta-heuristic algorithms is that the formulas for optimization were obtained from biologists. Compared to PSO, the FMO has higher accuracy and acceptable time consumption. However, the optimization effect of the FMO for low-dimensional complex functions is not very excellent. For low-dimensional complex functions, integer order speed update is easy to skip the optimal solution and thus cannot achieve good optimization effect, while fractional order speed can use fraction to update step size and can learn from historical speed, so more accurate results can be obtained.

Fractional Calculus (FC) [46] is an generalization of the classical concept of calculus. Similar to classical calculus, FC mainly includes fractional derivatives and fractional integrals. The difference between the two kinds of calculus is, the orders of derivatives and integrals in classical one are integers, while in FC, the orders can be fractions. Compared to classical calculus, FC can describe memory and inherited properties of various substances and their evolutionary processes accurately. Since the concept of FC appeared, the related theory has been successfully applied to many fields. Many researchers realize that they can also be used to describe some non-classical phenomena in natural sciences and engineering applications. In Meta-heuristic algorithm, a novel Fractional-Order Darwinian PSO was presented in paper [47]. In [48], BA algorithm based on FC was shown. In Ref. [49], Fractional-Order Cuckoo Search Algorithm is designed for financial systems. The Fractional calculus-based firefly algorithm was described in [50] and applied to parameter estimation of chaotic systems, and enhanced fractional chaotic whale optimization algorithm (WOA) was designed in paper [51] for Parameter Identification of isolated wind-diesel power systems. The paper [52] introduced the augmented Lagrangian PSO with fractional order velocity for fractional fixed-structure H_∞ controller. In addition, the generalization of the PSO algorithm based on complex-order is proposed in paper [53] and obtained excellent performance.

The problem of tuning PID controller is a low-dimensional complex functions because it has only three parameters. Therefore, Fractional-Order Fish Migration Optimization (FOFMO) is proposed in this manuscript since it is reasonable to improve the performance of the the FMO based on fractional order velocity. The rest of the paper is organized as following. Section II describes related works including the FMO algorithm, the FC theory, and the PID controller. In Section III, the FOFMO algorithm is presented in detail. In Section IV, the experiments on benchmark functions are shown. The PID simulation experiments are described in Section V. Section VI depicts the main work of the paper and gives some suggestions for further work.

II. RELATED WORKS

A. FISH MIGRATION OPTIMIZATION ALGORITHM

Every species in nature has its own way of survival, predation and reproduction. In addition, they must be able to against dangerous environment, since they constantly suffer from capture by natural enemies, and not every fish can grow to adult favourably. Biologists found that fish swim in water for many purposes. The FMO algorithm is proposed by taking grayling as an example. The life cycle of grayling can be divided into five stages as follows in the algorithm.

Stage 0+: newborn and young(age from 0 to 1 year).
Stage 1+: juvenile(age from 1 to 2 year).
Stage 2+: sub-adult(age from 2 to 3 years).
Stage 3+: adult(age from 3 to 4 years).
Stage 4+: adult(age from 4 to 5 years).

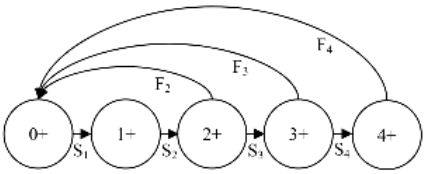


FIGURE 1. Life cycle graph of the grayling.

The fish in every stage have different fecundity rates. FIGURE 1 shows the life cycle graph of the grayling, where F_2, F_3, and F_4 are the fecundity rates in stage 2+, 3+, and 4+.

The FMO algorithm is designed to achieve the optimization through two processes including the swim process and the migration process. The swim process imitates the grayling swims and grows in the water to find food sources. In this process, energy consumption follows the movement of the fish that is defined by Eq. (1).

$$E_{r,d} = rand \cdot E \tag{1}$$

where $E_{r,d}$ is the energy consumed in d dimension, *rand* is a random number, E is a constant defines the maxima energy consumption in one dimension. In this algorithm, we set $E = 2$.

The functional relationship between the moving distance and energy consumption is

$$dis_{offset,d} = \frac{E_{r,d} \cdot U_{s,d}}{a + b \cdot (U_{s,d})^x} \tag{2}$$

where $dis_{offset,d}$ represents the moving distance and $U_{s,d}$ is the swimming speed in d dimension; a, b, and x are all constants, a denotes standard metabolic rate, b represents a scaling constant, x is a speed exponent, in literature, a, b, and x are 2.25, 36.2, and 2.23, respectively.

By updating the value of each dimension, the new position can be obtained by (3), where d_{offset} is the moving distance of the particle.

$$p_{new} = p_{old} + d_{offset} \tag{3}$$

If the fitness value of P_{new} is better than that of P_{old}, P_{old} will be updated by P_{new} utilizing (3). Meanwhile, the velocity will be updated by (4).

$$U_s = 2 \cdot U_s \tag{4}$$

As the fish are mature, some of them migrate back to their birthplace to reproduce offspring, that is migration process. Due to the fish in stage 0+ and stage 1+ are incapable of reproducing, the migration process only appears in stage 2+, stage 3+, and stage 4+. The fecundity rates of the three stages are 5%, 10%, and 100%, respectively. When fish find a new candidate point, the coordinate will be updated by (5)

$$P = (d_{max} - d_{min}) \cdot rand + d_{min}, \tag{5}$$

where d_{max}, d_{min} denote the maximum and maximum values of all dimensions of the fish, respectively.

Calculate the fitness value of the new candidate, the velocity will be updated by the following equation

$$U = \begin{cases} \pi \cdot U_s, & F(P) < F(P_{best}), \\ U_s, & otherwise, \end{cases} \tag{6}$$

where U_s denotes the initial velocity.

B. FRACTIONAL CALCULUS (FC)

The FC originates from the classical calculus and has a history of more than 300 years. Due to the development of various applied disciplines such as fluid mechanics, cybernetics, and biology, the FC has not made great progress until modern times. Then people gradually realized the practical significance of the FC, and more and more scholars began to study FC. As a branch of mathematical analysis, the FC has many advantages as follows. First of all, it reflects the inevitability of historical development from the mathematical point of view. Furthermore, memorability is a good feature of the FC. In addition, compared to the nonlinear model, the FC model can fit the real world better because of its concise expression.

Three definitions of fractional derivatives (FDs) are described as follows, where α is the order of fractional derivative.

Definition 1 (Riemann-Liouville FD):

$$\begin{aligned} {}_{RL}D^{\alpha}_{a,t} f(x) &= \begin{cases} \dfrac{d^n f}{dt^n}, & \alpha = n, \\ \dfrac{1}{\Gamma(n-\alpha)} \dfrac{d^n}{dx^n} \displaystyle\int_a^x (x-t)^{n-\alpha-1} f(t) dt, & n-1 < \alpha < n. \end{cases} \end{aligned} \tag{7}$$

Definition 2 (Caputo FD):

$$\begin{aligned} {}_{C}D^{\alpha}_{a,t} f(x) &= \begin{cases} \dfrac{d^n f}{dt^n}, & \alpha = n, \\ \dfrac{1}{\Gamma(n-\alpha)} \displaystyle\int_a^x (x-t)^{n-\alpha-1} f^n(t) dt, & n-1 < \alpha < n. \end{cases} \end{aligned} \tag{8}$$

Definition 3 (Grünwald-Letnikov FD):

$${}_{GL}D^{\alpha} f(x) = \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{k=0}^{+\infty} \frac{(-1)^k \Gamma(\alpha+1) f(x-kh)}{\Gamma(k+1)\Gamma(\alpha-k+1)}. \tag{9}$$

For Grünwald-Letnikov FD, the discrete time situation can be approximated by (10)

$${}_{GL}D^{\alpha} f(t) = \frac{1}{T^{\alpha}} \sum_{k=0}^{r} \frac{(-1)^k \Gamma(\alpha+1) f(t-kT)}{\Gamma(k+1)\Gamma(\alpha-k+1)}. \tag{10}$$

where T is the time increment, r is the truncation order.

$\Gamma(x)$ is Gamma function in above three FD definitions.

The above three FC has the following properties:

(a) Linearity $D^{\alpha}[af(x) + bg(x)] = aD^{\alpha} f(x) + bD^{\alpha} g(x)$.

(b) The index law $D^{\alpha+\beta} f(x) = D^{\alpha} D^{\beta} f(x)$.

(c) Generalized Leibniz rule
$D^{\alpha}[f(x)\cdot g(x)]=\sum_{i=0}^{\infty} D^{i}f(x)\cdot D^{\alpha-i}g(x)$,
where a, b are constants.

C. PID CONTROLLER

The PID controller has three parameters that have different effects on PID controller. The proportional gain K_P can adjust the deviation proportionally, quickly, and timely to improve the control sensitivity, but there is a steady-state error and the control accuracy is low. The integral gain K_I can eliminate the steady-state error, but it will affect the stability of the system. The integral gain K_D can speed up the response of the system, reduce the overshoot, reduce the oscillation, and "predict" the dynamic process.

The input-output equation in the time domain of the classical PID controller is

$$e(t)=y(t)-r(t), \tag{11}$$

$$u(t)=K_P e(t)+K_I\int_0^t e(\tau)d\tau+K_D\frac{de(t)}{dt}. \tag{12}$$

where $r(t)$, $y(t)$, $u(t)$ are system input, controller output, and system output, respectively, $e(t)$ is error signal.

The corresponding transfer function of PID controller is

$$G(s)=\frac{U(s)}{E(s)}=K_P+\frac{K_I}{s}+K_D s. \tag{13}$$

III. PROPOSED FRACTIONAL-ORDER FISH MIGRATION OPTIMIZATION ALGORITHM

In this section, a novel Fractional-Order Fish Migration Optimization (FOFMO) algorithm is described detailly. The FOFMO algorithm combines the FMO algorithm with the concept of Grünwald-Letnikov FD. The difference between the FMO algorithm and the FOFMO algorithm lies in two aspects. On the one hand, the update strategy of fractional-order velocity is used in the FOFMO. On the other hand, the new offspring position of the FOFMO is produced based on the global best particle.

A. FRACTIONAL-ORDER VELOCITY

Based on the analysis in Section 2.1, the velocity in the FMO is updated by (4) and (6). Although the FMO algorithm has a strong advantage in searching global optimal solution, it still has a weak exploitation ability since it takes a lot of time to make explore. In order to improve the ability of exploitation, the FC is used to update velocity. Supposing the time interval is 1, then

$$d_{offset}=\frac{E_r\cdot U_s^t}{a+b\cdot(U_s^t)^x},$$
$$U_s^t=P^t-P^{t-1}. \tag{14}$$

From (14), due to $a=2.25, b=36.2, x=2.23$, it can be find that for the whole fraction, compared to numerator, denominator is much bigger. As the iterative process continues, the speed of the fish gradually slows down, which will cause the algorithm to stagnate. In order to avoid the algorithm falling into the local optimal solution, the concept of Grünwald-Letnikov FD is introduced in this manuscript.

From (9), let $\alpha=1$, we obtain

$${}_{GL}D^{\alpha}[f_{t+1}]=f_{t+1}-f_t. \tag{15}$$

Equation (15) is the derivative of order 1 in the discrete case.

For $\Gamma(x)$, we have

$$\begin{aligned}\Gamma(\alpha+1)&=\alpha\Gamma(\alpha),\\ \Gamma(\alpha+1)&=\alpha(\alpha-1)\Gamma(\alpha-1),\\ \Gamma(\alpha+1)&=\alpha(\alpha-1)(\alpha-2)\Gamma(\alpha-2),\\ \Gamma(\alpha+1)&=\alpha(\alpha-1)(\alpha-2)(\alpha-3)\Gamma(\alpha-3),\end{aligned} \tag{16}$$

In order to generalize, let $T=1$, $r=4$ [50]–[52], Eq. (17) can be obtained.

$$\begin{aligned}{}_{GL}D^{\alpha}(f_{t+1})&=\frac{1}{T^{\alpha}}\sum_{k=0}^{r}\frac{(-1)^k\Gamma(\alpha+1)f_{t+1-kT}}{\Gamma(k+1)\Gamma(\alpha-k+1)}\\ &=f_{t+1}-\alpha f_t-\frac{1}{2}\alpha(1-\alpha)f_{t-1}\\ &\quad-\frac{1}{6}\alpha(1-\alpha)(2-\alpha)f_{t-2}\\ &\quad-\frac{1}{24}\alpha(1-\alpha)(2-\alpha)(3-\alpha)f_{t-3}.\end{aligned} \tag{17}$$

For the proposed algorithm, assume the population size of fish is ps and the dimension is Dim, let the position matrix $P=P(ps,Dim)=[p_1,p_2,\cdots,p_{ps}]^T$, where $p_i=[p_{i,1},p_{i,2},\cdots,p_{i,Dim}]$ is the position of particle i. Similarly, let $P^{pre}=[P^{pre1},P^{pre2},P^{pre3},P^{pre4}]$ denote the historical position of the particles which is used to calculate fractional-order velocity. Specifically, $P^{pre\{h\}}=P^{pre\{h\}}(ps,Dim)=[p_1^{pre\{h\}},p_2^{pre\{h\}},\cdots,p_{ps}^{pre\{h\}}]^T$, $p_i^{pre\{h\}}=[p_{i,1}^{pre\{h\}},p_{i,2}^{pre\{h\}},\cdots,p_{i,Dim}^{pre\{h\}}]$ where $h=1,2,3,4$; $i=1,2,\cdots,ps$. Therefore, the velocity of the particle is updated by (18), where $U_{s,d}$ is the velocity of dimension d.

$$\begin{aligned}U_{s,d}&=p_{i,d}-\alpha p_{i,d}^{pre1}-\frac{1}{2}\alpha(1-\alpha)p_{i,d}^{pre2}\\ &\quad-\frac{1}{6}\alpha(1-\alpha)(2-\alpha)p_{i,d}^{pre3}\\ &\quad-\frac{1}{24}\alpha(1-\alpha)(2-\alpha)(3-\alpha)p_{i,d}^{pre4}.\\ p_{i,d}^{new}&=\frac{E_{r,d}\cdot U_{s,d}}{a+b\cdot(U_{s,d})^x}.\\ p_{i,d}^{pre4}&=p_{i,d}^{pre3}.\\ p_{i,d}^{pre3}&=p_{i,d}^{pre2}.\\ p_{i,d}^{pre2}&=p_{i,d}^{pre1}.\\ p_{i,d}^{pre1}&=p_{i,d}^{new}.\end{aligned} \tag{18}$$

B. NEW POSITIONS OF THE OFFSPRINGS

Moveover, the graylings migration back to reproduce the new offsprings when they have group to maturity. The graylings should breed in a position that is more conducive to survival.

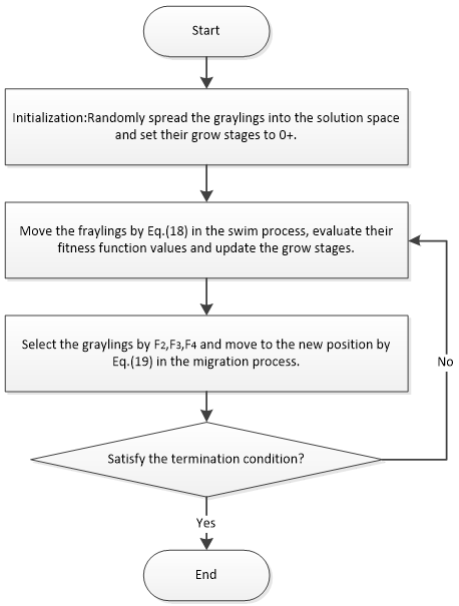


FIGURE 2. The diagram of the proposed FOFMO.

Therefore, the positions of the new offsprings should close to the global best particle. Equation (5) is replaced by (19).

$$p_i^{new} = p_{gbest} + rand \cdot (p_i^{old} - p_{gbest}) \tag{19}$$

C. THE PROPOSED ALGORITHM

In this section, the novel algorithm based on the fractional-order velocity and the new positions is described in detail. The diagram of the FOFMO is given in FIGURE 2.

TABLE 1 shows the pseudo code of the FOFMO algorithm, where $Rate_{fecundity}$ is the fecundity rates at specific stage, E_{min} is the minimum energy to eliminate, $x_{g,i}$ is the grow status, and $x_{eng,i}$ is the energy of particle i.

IV. EXPERIMENTAL RESULTS AND ANALYSIS ON BENCHMARK FUNCTION

The value of α affects significantly on the memory of the FOFMO. In this section, 23 classical benchmark functions utilized by many researchers [23], [24] are used to examine the performance of the proposed FOFMO algorithm with different α. In detail, TABLE 2 contains seven unimodal functions, TABLE 3 shows the details of six common multimodal functions, and TABLE 4 displays the details of ten multimodal functions in low dimension. Unimodal functions have only one global optimal solution, but there is no local optimal solution, this can verify whether the algorithm has the ability to search the global optimal solution. Due to multimodal functions have many local optimal solutions, it is possible to verify if the algorithm falls into the local optimal solution. Multimodal functions in low dimension can verify the convergence of the algorithm under more strict conditions. *Search space* denotes the boundary of search space; D means the dimension of the function and f_{min} represents the optimal value of the function.

TABLE 1. The pseudo code of FOFMO algorithm.

Algorithm 1 Pseudo code of FOFMO algorithm

Input:
The dimension Dim, popolation size ps, a, b, x,
maximum iterations $Maxitn$.
Initialization:
Initialize the searching space V, $itn = 1$, position matrix P^{itn},
historical position matrix $P^{pre,itn}$, the energy X_{eng}^{itn},
calculate fitness function values $f(p_i^{itn})$, set grow status $X_g^{itn} = 0+$.
Iteration:
while $itn < Maxitn || !stopCriterion$ **do**
  for $i = 1 : ps$ **do**
    for $d = 1 : Dim$ **do**
      Calculate consuming energy by (1).
      Calculate fractional-order position $p_{i,d}^{itn}$, update the history by (18).
    end for
    Calculate the fitness value of new position $p_{i,d}^{itn}$.
    if $f(p_i^{itn}) < f(p_{gbest}^{itn})$ then
      $f(p_{gbest}^{itn}) = f(p_i^{itn})$
      $p_{gbest}^{itn} = p_i^{itn}$
      Increased energy $x_{eng,i} = x_{eng,i} + rand \cdot E_{max}$.
    end if
    Consuming energy by $x_{eng,i} = x_{eng,i} - f(p_i^{itn}) / \sum_i f(p_i^{itn})$.
    if $x_{eng,i} < E_{min}$, then
      $x_{g,i} = 5$,the grayling died.
    end if
    if $x_{g,i} = 0 + || x_{g,i} = 1+$ then
      $x_{g,i} = [x_{g,i} + 1]+$
    else if $x_{g,i} = 2 + || x_{g,i} = 3+$ then
      if rand()<$Rate_{fecundity}$ then
        Immigrate and produce offspring by (19).
        $x_{g,i} = 0$
      else
        $x_{g,i} = [x_{g,i} + 1]+$
      end if
    else if $x_{g,i} = 4+$ then
      Immigrate and produce offspring by (19).
      $x_{g,i} = 0$
    end if
  end for
  $itn = itn + 1$
end while
output:
The global optimal solution: $p_{gbest}, f(p_{gbest})$.

In order to find the optimal fractional order α in (10), the FMO, FOFMO($\alpha = 0.1$), FOFMO($\alpha = 0.3$), FOFMO($\alpha = 0.5$), FOFMO($\alpha = 0.7$), and FOFMO($\alpha = 0.9$) algorithms are examined in the experiment. The experiment runs 30 times and 1000 iterations on each benchmark function. The population has 100 individuals. TABLE 5 shows the statistical results of the FMO and the five FOFMO algorithm with different value. AVG is the mean of the result and STD is the standard deviation of the results of 30 times. The red data represent the optimal result. The blue data represent all algorithms acquire the optimal result. The last line represents the number of times of each algorithm achieves the optimal result.

From TABLE 5, it can be seen that the effect becomes worse as the value of α increases. In detail, these six algorithms obtain the same optimal solutions on eight benchmark functions. Among them, three unimodal functions, six common multimodal functions, and two multimodal functions in low dimension. The FMO only achieves 2 optimal results. However, The FOFMO($\alpha = 0.1$) obtains 9 optimal (red) results, the FOFMO($\alpha = 0.3$) and the FOFMO($\alpha = 0.5$)

TABLE 2. Unimodal benchmark functions.

Function Name	Function Expression	Search space	D	f_{min}
Shpere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]	30	0
Schwefel's function 2.21	$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	30	0
Schwefel's function 1.2	$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	[-100,100]	30	0
Schwefel's function 2.22	$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	[-100,100]	30	0
Rosenbroke	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30,30]	30	0
Step	$f_6(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	[-100,100]	30	0
Dejong's noisy	$f_7(x) = \sum_{i=1}^{n} i x_i^4 + rand[0, 1)$	[-1.28,1.28]	30	0

TABLE 3. Common multimodal benchmark functions.

Function Name	Function Expression	Search space	D	f_{min}
Schwefel	$f_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	[-500,500]	30	-12569
Rastringin	$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	30	0
Ackley	$f_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2})$ $-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	[-32,32]	30	0
Griewank	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	30	0
Generalized penalized 1	$f_{12} = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n}(y_i - 1)^2$ $[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4) y_i$ $= 1 + \frac{x_i+1}{4} u(x_i, a, k, m)$ $= \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	[-50,50]	30	0
Generalized penalized 2	$f_{13}(x) = 0.1\{\sin^2(3\pi x_1)$ $+ \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)]$ $+ (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\}$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	[-50,50]	30	0

TABLE 4. Multimodal benchmark functions in low dimension.

Function Name	Function Expression	Search space	D	f_{min}
Fifth of Dejong	$f_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	[-65,65]	2	1
Kowalik	$f_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	[-5,5]	4	0.00030
Six-hump camel back	$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	[-5,5]	2	-1.0316
Branins	$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	[-5,5]	2	0.398
Goldstein-Price	$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2$ $+6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1$ $+12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[-2,2]	2	3
Hartman 1	$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	[1,3]	3	-3.86
Hartman 2	$f_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	[0,1]	6	-3.32
Shekel 1	$f_{21}(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	4	-10.1532
Shekel 2	$f_{22}(x) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	4	-10.4028
Shekel 3	$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	4	-10.5363

obtain 3 optimal results, respectively. The FOFMO with these 3 values is superior to the FMO algorithm. In special, the FOFMO($\alpha = 0.1$) obtains the most optimal results and performs best. Therefore, $\alpha = 0.1$ is the optimal order

TABLE 5. The statistical results of FMO, FOFMO($\alpha = 0.1$), FOFMO($\alpha = 0.3$), FOFMO($\alpha = 0.5$), FOFMO($\alpha = 0.7$), and FOFMO($\alpha = 0.9$).

Function	FMO		FOFMO($\alpha = 0.1$)		FOFMO($\alpha = 0.3$)		FOFMO($\alpha = 0.5$)		FOFMO($\alpha = 0.7$)		FOFMO($\alpha = 0.9$)	
	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD
f1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0001	0.0001	0.0004	0.0001	0.0001
f3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f5	27.7043	0.6060	27.6661	0.3868	27.7479	0.3055	27.7570	0.3364	27.9002	0.3855	27.9611	0.4798
f6	5.0443	0.1478	4.9849	0.2156	4.9112	0.2726	5.0311	0.2070	5.1267	0.2459	5.1630	0.1872
f7	0.0001	0.0001	0.0002	0.0002	0.0003	0.0003	0.0004	0.0003	0.0004	0.0005	0.0003	0.0003
f8	-3837.4621	250.5054	-4484.2525	309.4060	-4163.7633	308.4231	-3934.1873	256.8661	-3982.8045	225.5486	-3990.6394	280.7772
f9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f12	0.5842	0.0479	0.5785	0.0449	0.5887	0.0321	0.5927	0.0402	0.5912	0.0649	0.6129	0.0513
f13	2.5000	0.0531	2.4376	0.0969	2.4709	0.0681	2.5077	0.0585	2.5244	0.0858	2.5202	0.0816
f14	2.1885	0.9972	1.0537	0.2235	0.9983	0.0007	0.9989	0.0022	1.0205	0.0677	1.3517	0.7368
f15	0.0005	0.0003	0.0005	0.0003	0.0005	0.0003	0.0004	0.0002	0.0005	0.0002	0.0007	0.0004
f16	-1.0316	0.0000	-1.0316	0.0000	-1.0316	0.0000	-1.0316	0.0000	-1.0316	0.0000	-1.0316	0.0000
f17	0.3987	0.0008	0.3984	0.0005	0.3986	0.0005	0.3986	0.0009	0.3985	0.0005	0.3984	0.0004
f18	3.0000	0.0000	3.0000	0.0000	3.0000	0.0000	3.0000	0.0000	3.0000	0.0000	3.0000	0.0000
f19	-3.8538	0.0028	-3.8561	0.0032	-3.8558	0.0027	-3.8554	0.0034	-3.8562	0.0031	-3.8552	0.0027
f20	-2.9371	0.1522	-2.9649	0.2358	-2.9479	0.1589	-2.9817	0.1469	-2.9639	0.1595	-2.9065	0.1588
f21	-4.7807	0.1435	-5.0009	0.7831	-4.8021	0.1573	-4.7308	0.1546	-4.7097	0.1105	-4.6606	0.2008
f22	-4.7344	0.1704	-5.1078	0.8941	-4.8919	0.1706	-4.8320	0.1222	-4.6789	0.2137	-4.7198	0.1407
f23	-4.6536	0.2113	-5.0086	0.3830	-4.8794	0.1259	-4.8260	0.1494	-4.7724	0.1687	-4.8627	0.3543
count	2+8		9+8		3+8		3+8		1+8		1+8	

in this experiment and it can be concluded that (18) is effective.

TABLE 6 shows the runtime of the FMO and the FOFMO($\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$) in the experiment. We can see that the FOFMO algorithm has a longer runtime than the FMO. The main reason is that the velocity and position update by (18) of the FOFMO are more complicate.

V. SIMULATION EXPERIMENT ON THE PID CONTROLLER

A. PERFORMANCE ANALYSIS

In this section, we design and optimize the PID controller such that it has good performance. From (12) and (13), we find that the PID controller has three parameters: K_P, K_I, and K_D, so we need to optimize them only. For the design and optimization experiment of PID controller, there are many performance evaluation criterions. The common performance evaluation criterions are integral absolute error (IAE), integral square error (ISE), integral time absolute error (ITAE), and integral time square error (ITSE).

$$J_{IAE} = \int_0^\infty |e(t)|dt$$
$$J_{ISE} = \int_0^\infty e^2(t)dt$$
$$J_{ITAE} = \int_0^\infty t|e(t)|dt$$
$$J_{ITSE} = \int_0^\infty te^2(t)dt, \tag{20}$$

where $e(t)$ is error signal. In this paper, we apply the ITAE performance evaluation criterion, so J_{ITAE} is the fitness function.

In this paper, the system (21) is examined for the PID tuning problem.

$$G_1(s) = \frac{s+2}{s^4 + 8s^3 + 4s^2 - s + 0.4} \tag{21}$$

In order to verify the performance of the proposed algorithm, we utilize the FOFMO($\alpha = 0.1$) with PSO [11], fractional-order particle swarm optimization algorithm FOPSO($\alpha = 0.9$) [12] and the FMO [44] for design and optimization of the PID controller, respectively, and compare the results. The parameter $\alpha = 0.1$ in the experiment refers to the order of fractional derivative, because FOFMO performs best at this value. Similarly, $\alpha = 0.9$ is the best parameter of the FOPSO. In the experiment, all algorithms run 5 times, 500 iterations, and 20 particles on fitness function. The PID controller is shown in FIGURE 3. We set the PID controller

TABLE 6. The time consumption of the test algorithms.

Function	FMO	FOFMO($\alpha = 0.1$)	FOFMO($\alpha = 0.3$)	FOFMO($\alpha = 0.5$)	FOFMO($\alpha = 0.7$)	FOFMO($\alpha = 0.9$)
f1	0.2825	0.3290	0.3227	0.3186	0.3529	0.3093
f2	0.2987	0.3474	0.3397	0.3342	0.3239	0.3240
f3	1.0803	1.1000	1.1364	1.1130	1.3144	1.1212
f4	0.2864	0.3112	0.3289	0.3181	0.3437	0.3111
f5	0.3804	0.4083	0.4170	0.4158	0.4598	0.4102
f6	0.2808	0.3098	0.3092	0.3330	0.3187	0.3102
f7	0.7245	0.7641	0.6221	0.5674	0.6693	0.6097
f8	0.4234	0.4295	0.4581	0.5152	0.5378	0.4231
f9	0.3057	0.3416	0.3442	0.3434	0.4008	0.4141
f10	0.3120	0.3727	0.3624	0.3552	0.3559	0.3620
f11	0.4087	0.4429	0.4465	0.4422	0.4934	0.4413
f12	1.6051	1.4863	1.3888	1.3468	1.3655	1.1806
f13	1.5295	1.2858	1.1824	1.3134	1.1650	1.1712
f14	2.5514	2.4220	2.4150	2.6054	2.4327	2.3973
f15	0.1352	0.1458	0.1417	0.1428	0.1461	0.1430
f16	0.1219	0.1243	0.1136	0.1134	0.1172	0.1277
f17	0.0863	0.0874	0.0870	0.0889	0.0896	0.0886
f18	0.0931	0.0840	0.0880	0.0857	0.0850	0.0833
f19	0.1599	0.1627	0.2301	0.1627	0.1660	0.1652
f20	0.1853	0.1974	0.1881	0.1892	0.1894	0.2114
f21	0.6873	0.6669	0.6769	0.7306	0.6814	0.6859
f22	0.8846	0.8948	0.8836	0.9469	0.9251	0.8994
f23	1.2446	1.2131	1.2383	1.6114	1.2557	1.2464

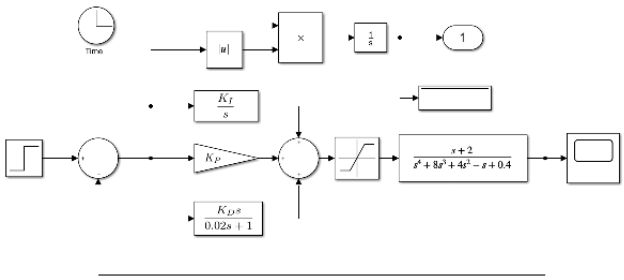


FIGURE 3. Block diagram of PID controller.

TABLE 7. The statistical results of PSO, FOPSO($\alpha = 0.9$), FMO, FOFMO($\alpha = 0.1$) for $G_1(s)$ ($0 \le K_P, K_I, K_D \le 300$).

	PSO	FOPSO($\alpha = 0.9$)	FMO	FOFMO($\alpha = 0.1$)
AVG	4.3209	2.3139	1.2730	1.2001
STD	0.8452	1.4217	0.0278	0.0568

parameters $0 \le K_P \le 300, 0 \le K_I \le 300$, and $0 \le K_D \le 300$.

TABLE 7 shows the statistical results of the PSO, FOPSO($\alpha = 0.9$), FMO, and FOFMO($\alpha = 0.1$) when $0 \le K_P, K_I, K_D \le 300$. In TABLE 7, AVG is the mean of the optimal values of 5 times, STD is the standard deviation of the optimal values of 5 times. We find that the FOPSO has smaller AVG value than the PSO and FOFMO has smaller AVG value than the FMO. At the same time, the FOFMO($\alpha = 0.1$) achieves the smallest AVG value. FIGURE 4 shows the curves of the fitness value of the results of 5 times.

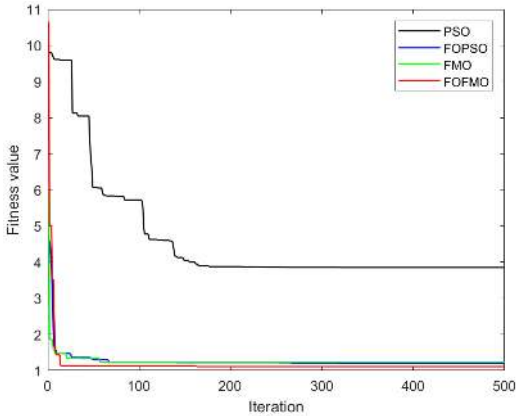


FIGURE 4. The fitness values of PSO, FOPSO, FMO, and FOFMO.

TABLE 8. The parameter values of PID controller tuned by four algorithms for $G_1(s)$ ($0 \le K_P, K_I, K_D \le 300$).

	PSO	FOPSO($\alpha = 0.9$)	FMO	FOFMO($\alpha =$
K_P	191.9216	180.4817	233.2300	70.7816
K_I	95.0081	0.0000	0.0002	0.1762
K_D	4.3209	208.0404	263.2398	82.2974
Optimal value	3.8531	1.1988	1.2346	1.1060

TABLE 8 shows the parameter values (K_P, K_I, and K_D) corresponding to the optimal values when $0 \le K_P, K_I, K_D \le 300$. The results show that the FOPSO has smaller fitness value than PSO and FOFMO has smaller optimal value than FMO. At the same time, FOFMO($\alpha = 0.1$) achieves the smallest optimal value.

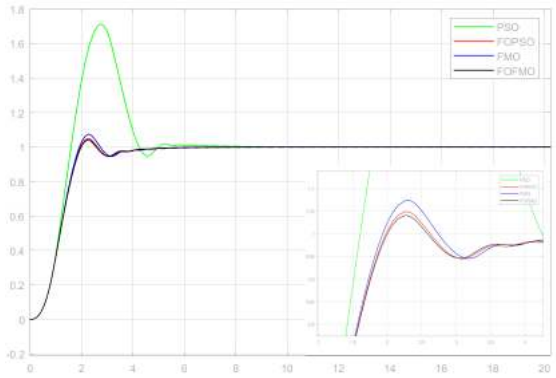


FIGURE 5. The terminal voltage step response of PID controller tuned by four algorithms for $G_1(s)$ ($0 \le K_P, K_I, K_D \le 300$).

TABLE 9. The overshoots (OS) of PID controller tuned by four algorithms for $G_1(s)$ ($0 \le K_P, K_I, K_D \le 300$).

	PSO	FOPSO($\alpha = 0.9$)	FMO	FOFMO($\alpha = 0.1$)
OS	71.1%	4.9%	7.4%	4.0%

TABLE 10. The parameter values and performance of PID controller tuned by four algorithms for $G_2(s)$ ($-50 \le K_P, K_I, K_D \le 50$).

0.1)	PSO	FOPSO($\alpha = 0.9$)	FMO	FOFMO($\alpha = 0.1$)
K_P	-47.8946	-26.1628	-18.0509	-14.6076
K_I	-5.4199	-41.4758	-50.0000	-50.0000
K_D	-2.0196	-0.7790	-0.7327	-0.6710
AVG	0.0516	0.0424	0.0385	0.0372
STD	0.0054	0.0063	0.0012	0.0010
Optimal value	0.0511	0.0437	0.0387	0.0371

FIGURE 5 shows the terminal voltage step response of the PID controller which tuned by the PSO, FOPSO($\alpha = 0.9$), FMO, and FOFMO($\alpha = 0.1$) when $0 \le K_P, K_I, K_D \le 300$. In TABLE 9, we can find that the overshoots of the PSO and FMO become smaller with the help of fractional derivative. Therefore, fractional derivative is effective for improving the performance of the PID controller. Meanwhile, the PID controller tuned by the FOFMO has the smallest overshoot. As a whole, the results show that the FOFMO algorithm has good performance for the design and optimization of the PID controller.

In order to further examine the performance of the proposed algorithm, another system (22) [54] is examined for the PID tuning problem.

$$G_2(s) = \frac{-1.39s^2 - 1.99s - 0.2577}{s^3 + 1.408s^2 + 0.656s + 0.1013} \tag{22}$$

The result is shown in TABLE 10. The FOFMO achieves the smallest AVG value and optimal value. This is the same conclusion as in Table 7 and Table 8.

B. ANALYSIS ABOUT THE RANGE OF K_P, K_I, K_D

TABLE 11 shows the statistical results of the PSO, FOPSO($\alpha = 0.9$), FMO, and FOFMO($\alpha = 0.1$) when $0 \le K_P, K_I, K_D \le 100$. The conclusion is similar to that in TABLE 7. That is, the FOPSO has smaller AVG value than the PSO and the FOFMO has smaller AVG value than the FMO.

TABLE 11. The statistical results of PSO, FOPSO($\alpha = 0.9$), FMO, FOFMO($\alpha = 0.1$) for $G_1(s)$ ($0 \le K_P, K_I, K_D \le 100$).

	PSO	FOPSO($\alpha = 0.9$)	FMO	FOFMO($\alpha = 0.1$)
AVG	3.9358	3.9265	1.1836	1.1271
STD	0.0081	0.0141	0.1899	0.0716

TABLE 12. The parameter values of PID controller tuned by four algorithms for $G_1(s)$ ($0 \le K_P, K_I, K_D \le 100$).

	PSO	FOPSO($\alpha = 0.9$)	FMO	FOFMO($\alpha = 0.1$)
K_P	56.8084	62.3900	39.3489	42.4726
K_I	28.0921	31.1346	19.6675	0.1815
K_D	67.9142	74.7298	47.1779	49.6289
Optimal value	3.9367	3.9258	1.1260	1.1225

TABLE 13. The statistical results of PSO, FOPSO($\alpha = 0.9$), FMO, FOFMO($\alpha = 0.1$) for $G_1(s)$ ($0 \le K_P, K_I, K_D \le 30$).

	PSO	FOPSO($\alpha = 0.9$)	FMO	FOFMO($\alpha = 0.1$)
AVG	3.5240	3.5154	1.1177	1.0877
STD	1.2691	1.3649	0.0348	0.0204

TABLE 14. The parameter values of PID controller tuned by four algorithms for $G_1(s)$ ($0 \le K_P, K_I, K_D \le 30$).

	PSO	FOPSO($\alpha = 0.9$)	FMO	FOFMO($\alpha =$
K_P	25.3187	25.3272	25.3244	26.1102
K_I	11.9753	11.9650	11.9710	0.1745
K_D	30.0000	30.0000	30.0000	30.0000
Optimal value	4.1259	4.1258	1.1125	1.0971

Similarly, FOFMO($\alpha = 0.1$) achieves the smallest AVG value. TABLE 12 shows the parameter values corresponding to the optimal values when $0 \le K_P, K_I, K_D \le 100$. The results show that the FOPSO has smaller optimal value than the PSO and FOFMO has smaller optimal value than FMO. At the same time, the FOFMO($\alpha = 0.1$) achieves the smallest optimal value.

TABLE 13 shows the statistical results of the PSO, FOPSO($\alpha = 0.9$), FMO, and FOFMO($\alpha = 0.1$) when $0 \le K_P, K_I, K_D \le 30$. The conclusion is similar to that in TABLE 7 and TABLE 11. That is, the FOPSO has smaller AVG value than the PSO and FOFMO has smaller AVG value than FMO. At the same time, the FOFMO($\alpha = 0.1$) achieves the smallest AVG value. TABLE 14 shows the parameter values corresponding to the optimal values when $0 \le K_P, K_I, K_D \le 30$. The results show that the FOPSO has smaller optimal value than the PSO and FOFMO has smaller optimal value than the FMO. At the same time, the FOFMO($\alpha = 0.1$) achieves the smallest optimal value.

C. ROBUSTNESS ANALYSIS

Robustness is an important property to evaluate the stability and reliability of a controller [55]–[58]. In this section, the robustness of the PID controller for different tuning methods for $G_1(s)$ is tested. The system slightly different from (21) is used as (23).

$$G_{td}(s) = \frac{s+2}{s^4 + 8s^3 + 4s^2 - s + 0.4} e^{-0.01s} \tag{23}$$

TABLE 15. The parameter values and performance of PID controller tuned by four algorithms for $G_{td}(s)$ ($0 \le K_P, K_I, K_D \le 300$).

	PSO	FOPSO($\alpha = 0.9$)	FMO	FOFMO($\alpha = 0.1$)
K_P	167.7314	164.5787	104.1152	78.1361
K_I	122.9517	75.7975	0.0000	0.2124
K_D	234.5847	197.0660	121.4009	92.0926
AVG	4.3511	4.5404	1.2447	**1.1765**
STD	1.7685	0.1725	0.1615	**0.0376**
Optimal value	6.4304	4.6419	1.5325	**1.2365**
OS	80.8%	71.0%	6.2%	**5.3%**
$t_s(\pm 2\%)$	4.1110	2.8400	2.0090	**1.6110**

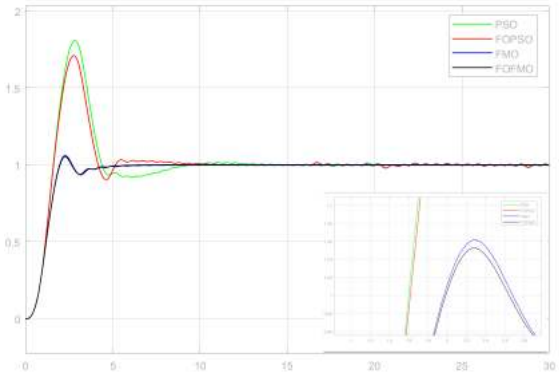


FIGURE 6. The terminal voltage step response of PID controller for $G_{td}(s)$.

TABLE 16. The running time of PSO, FOPSO($\alpha = 0.9$), FMO, FOFMO($\alpha = 0.1$) for $G_1(s)$ ($0 \le K_P, K_I, K_D \le 100$).

	PSO	FOPSO($\alpha = 0.9$)	FMO	FOFMO($\alpha = 0.1$)
time	717.3926	802.7293	341.0879	459.3857

In order to test the robustness of these different tuning methods, the time delay is added to $G(s)$. TABLE 15 shows the the performance on tuning the PID controller with the PSO, FOPSO($\alpha = 0.9$), FMO, and FOFMO($\alpha = 0.1$). In TABLE 15, OS is the overshoot, and $t_s(\pm 2\%)$ is the settling time.

FIGURE 6 shows the terminal voltage step response of the PID controller which tuned by the PSO, FOPSO($\alpha = 0.9$), FMO, and FOFMO($\alpha = 0.1$).

From TABLE 15 and FIGURE 6, we can see that the PID controller tuned by the FOFMO is more robust and has better performance in terms of the AVG, STD, optimal value, overshoot, and settling time compared to other PID controllers.

D. RUNNING TIME

TABLE 16 shows the running time of the PSO, FOPSO($\alpha = 0.9$), FMO, FOFMO($\alpha = 0.1$) when $0 \le K_P, K_I, K_D \le 100$. We find that both the FMO and FOFMO are much faster than the PSO and FOPSO. This result is consistent with the result of paper [44]. Furthermore, the PSO is faster than the FOPSO and the FMO is faster than the FOFMO. The main reason is that the velocity and position update by fractional-order are more complicate.

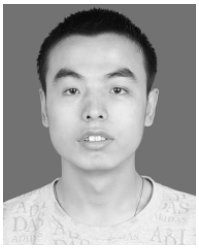
VI. CONCLUSION

In order to improve the performance on tuning the PID controller, a novel FOFMO algorithm is proposed in this manuscript based on the FC concepts. In particular, the velocity and position in the FOFMO are updated in fractional-order form. Meanwhile, the discrete fractional derivative used in the FOFMO is derived in detail. The experiment on benchmark functions shows that the proposed algorithm is superior to the original FMO on performance. In addition, the FOFMO algorithm is utilized to tune the PID controller by simulation experiment. The result reveals the PID controller tuned by the FOFMO has the best performance of the four algorithms including the PSO, FOPSO, and FMO. Therefore, it is effective to use the FOFMO algorithm for optimal control. The initial particle numbers affects the performance of many algorithms, and there are many other outstanding algorithms proposed in recent years. These will be tested in our future work.

REFERENCES

[1] P. D. Domanski, "Statistical measures for proportional–integral–derivative control quality: Simulations and industrial data," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 232, no. 4, pp. 428–441, Apr. 2018.
[2] O. Khan, C. M. R. Madhuranthakam, P. Douglas, H. Lau, J. Sun, and P. Farrell, "Optimized PID controller for an industrial biological fermentation process," *J. Process Control*, vol. 71, pp. 75–89, Nov. 2018.
[3] B. Nagaraj and N. Murugananth, "A comparative study of PID controller tuning using GA, EP, PSO and ACO," in *Proc. Int. Conf. Commun. Control Comput. Technol.*, Oct. 2010, pp. 305–313.
[4] G. Jahedi and M. M. Ardehali, "Genetic algorithm-based fuzzy-PID control methodologies for enhancement of energy efficiency of a dynamic energy system," *Energy Convers. Manage.*, vol. 52, no. 1, pp. 725–732, Jan. 2011.
[5] A. Riesco, B. Santos-Buitrago, M. Knapp, G. Santos-García, E. H. Galilea, and C. Talcott, "Fuzzy matching for cellular signaling networks in a choroidal melanoma model," in *Proc. Int. Conf. Practical Appl. Comput. Biol. Bioinf.* Cham, Switzerland: Springer, 2020, pp. 80–90.
[6] H. Dong, Y. Gao, X. Meng, and Y. Fang, "A multifactorial short-term load forecasting model combined with periodic and non-periodic features—A case study of Qingdao, China," *IEEE Access*, vol. 8, pp. 67416–67425, 2020.
[7] B. Cui, X. Chen, and Y. Lu, "Semantic segmentation of remote sensing images using transfer learning and deep convolutional neural network with dense connection," *IEEE Access*, vol. 8, pp. 116744–116755, 2020.
[8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Amsterdam, The Netherlands: Elsevier, 2014.
[9] X. Xue, J. Chen, and J.-S. Pan, *Evolutionary Algorithm Based Ontology Matching Technique*. Beijing, China: Science Press, 2018.
[10] X. Wang, J.-S. Pan, and S.-C. Chu, "A parallel multi-verse optimizer for application in multilevel image segmentation," *IEEE Access*, vol. 8, pp. 32018–32030, 2020.
[11] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007.
[12] E. J. S. Pires, J. A. T. Machado, P. B. de Moura Oliveira, J. B. Cunha, and L. Mendes, "Particle swarm optimization with fractional-order velocity," *Nonlinear Dyn.*, vol. 61, nos. 1–2, pp. 295–301, Jul. 2010.
[13] K.-C. Chang, Y.-W. Zhou, H.-C. Wang, Y.-C. Lin, K.-C. Chu, T.-L. Hsu, and J.-S. Pan, "Study of PSO optimized BP neural network and smith predictor for MOCVD temperature control in 7 nm 5G chip process," in *Proc. Int. Conf. Adv. Intell. Syst. Inform.* Cham, Switzerland: Springer, 2020, pp. 568–576.
[14] S. Qin, C. Sun, G. Zhang, X. He, and Y. Tan, "A modified particle swarm optimization based on decomposition with different ideal points for many-objective optimization problems," *Complex Intell. Syst.*, vol. 6, pp. 263–274, Mar. 2020.

[15] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.
[16] U. J, N. Metawa, K. Shankar, and S. K. Lakshmanaprabu, "Financial crisis prediction model using ant colony optimization," *Int. J. Inf. Manage.*, vol. 50, pp. 538–556, Feb. 2020.
[17] X. Yang and A. Hossein Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464–483, Jul. 2012.
[18] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "A compact bat algorithm for unequal clustering in wireless sensor networks," *Appl. Sci.*, vol. 9, no. 10, p. 1973, May 2019.
[19] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Oct. 2007.
[20] L. Tang, Z. Li, J. Pan, Z. Wang, K. Ma, and H. Zhao, "Novel artificial bee colony algorithm based load balance method in cloud computing," *J. Inf. Hiding Multimedia Sig. Process*, vol. 8, no. 2, pp. 460–467, 2017.
[21] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," in *Proc. Pacific Rim Int. Conf. Artif. Intell.* Berlin, Germany: Springer, 2006, pp. 854–858.
[22] X.-F. Ji, J.-S. Pan, S.-C. Chu, P. Hu, Q.-W. Chai, and P. Zhang, "Adaptive cat swarm optimization algorithm and its applications in vehicle routing problems," *Math. Problems Eng.*, vol. 2020, pp. 1–14, Apr. 2020.
[23] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[24] P. Hu, J.-S. Pan, and S.-C. Chu, "Improved binary grey wolf optimizer and its application for feature selection," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105746.
[25] P.-C. Song, J.-S. Pan, and S.-C. Chu, "A parallel compact cuckoo search algorithm for three-dimensional path planning," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106443.
[26] J.-S. Pan, P.-C. Song, S.-C. Chu, and Y.-J. Peng, "Improved compact cuckoo search algorithm applied to location of drone logistics hub," *Mathematics*, vol. 8, no. 3, p. 333, Mar. 2020.
[27] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, and W.-M. Zheng, "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, p. 767, Jan. 2020.
[28] J.-S. Pan, X. Wang, and S.-C. Chu, "A multi-group grasshopper optimisation algorithm for application in capacitated vehicle routing problem," *Data Sci. Pattern Recognit.*, vol. 4, no. 1, pp. 41–56, 2020.
[29] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994.
[30] C.-J. Weng, S. J. Liu, J. S. Pan, L. Liao, W. D. Zeng, P. Zhang, and L. Huang, "Enhanced secret hiding mechanism based on genetic algorithm," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*. Singapore: Springer, 2020, pp. 79–86.
[31] T.-T. Nguyen, Y. Qiao, J.-S. Pan, T.-K. Dao, T.-T.-T. Nguyen, and C.-J. Weng, "An improvement of embedding efficiency for watermarking based on genetic algorithm," *J. Inf. Hiding Multimedia Signal Process.*, vol. 11, pp. 79–89, Jun. 2020.
[32] K. V. Price, "Differential evolution," in *Handbook of Optimization*. Berlin, Germany: Springer, 2013, pp. 187–214.
[33] N. Liu, J. S. Pan, J. Lai, and S. C. Chu, "An efficient differential evolution via both top collective and p-best information," *J. Internet Technol.*, vol. 21, no. 3, pp. 629–643, 2020.
[34] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.
[35] S.-C. Chu, Y. Chen, F. Meng, C. Yang, J.-S. Pan, and Z. Meng, "Internal search of the evolution matrix in QUasi-Affine TRansformation Evolution (QUATRE) algorithm," *J. Intell. Fuzzy Syst.*, vol. 38, pp. 5673–5684, Jan. 2020.
[36] Z. Meng and J.-S. Pan, "QUasi-affine TRansformation evolution with external ARchive (QUATRE-EAR): An enhanced structure for differential evolution," *Knowl.-Based Syst.*, vol. 155, pp. 35–53, Sep. 2018.
[37] B.-Q. Jiang and J.-S. Pan, "A parallel quasi-affine transformation evolution algorithm for global optimization," *J. Netw. Intell.*, vol. 4, no. 2, pp. 30–46, 2019.
[38] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*. Dordrecht, The Netherlands: Springer, 1987, pp. 7–15.
[39] E. Kurtulus, A. R. Yildiz, S. M. Sait, and S. Bureerat, "A novel hybrid Harris hawks-simulated annealing algorithm and RBF-based metamodel for design optimization of highway guardrails," *Mater. Test.*, vol. 62, no. 3, pp. 251–260, Mar. 2020.
[40] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.
[41] Z.-K. Feng, S. Liu, W.-J. Niu, S.-S. Li, H.-J. Wu, and J.-Y. Wang, "Ecological operation of cascade hydropower reservoirs by elite-guide gravitational search algorithm with Lévy flight local search and mutation," *J. Hydrol.*, vol. 581, Feb. 2020, Art. no. 124425.
[42] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
[43] Q. Yang, S.-C. Chu, J.-S. Pan, and C.-M. Chen, "Sine cosine algorithm with multigroup and multistrategy for solving CVRP," *Math. Problems Eng.*, vol. 2020, pp. 1–10, Mar. 2020.
[44] J.-S. Pan, P.-W. Tsai, and Y.-B. Liao, "Fish migration optimization based on the fishy biology," in *Proc. 4th Int. Conf. Genetic Evol. Comput.*, Dec. 2010, pp. 783–786.
[45] Q.-W. Chai, S.-C. Chu, J.-S. Pan, and W.-M. Zheng, "Applying adaptive and self assessment fish migration optimization on localization of wireless sensor network on 3-D terrain," *J. Inf. Hiding Multimedia Signal Process.*, vol. 11, no. 2, pp. 90–102, 2020.
[46] R. Gorenflo and F. Mainardi, "Fractional calculus," in *Fractals and Fractional Calculus in Continuum Mechanics*. Vienna, Austria: Springer, 1997, pp. 223–276.
[47] M. S. Couceiro, R. P. Rocha, N. M. F. Ferreira, and J. A. T. Machado, "Introducing the fractional-order darwinian PSO," *Signal, Image Video Process.*, vol. 6, no. 3, pp. 343–350, Sep. 2012.
[48] R. Boudjemaa, D. Oliva, and F. Ouaar, "Fractional Lévy flight bat algorithm for global optimisation," *Int. J. Bio-Inspired Comput.*, vol. 15, no. 2, pp. 100–112, 2020.
[49] D. Yousri and S. Mirjalili, "Fractional-order cuckoo search algorithm for parameter identification of the fractional-order chaotic, chaotic with noise and hyper-chaotic financial systems," *Eng. Appl. Artif. Intell.*, vol. 92, Jun. 2020, Art. no. 103662.
[50] Y. Mousavi and A. Alfi, "Fractional calculus-based firefly algorithm applied to parameter estimation of chaotic systems," *Chaos, Solitons Fractals*, vol. 114, pp. 202–215, Sep. 2018.
[51] Y. Mousavi, A. Alfi, and I. B. Kucukdemiral, "Enhanced fractional chaotic whale optimization algorithm for parameter identification of isolated wind-diesel power systems," *IEEE Access*, vol. 8, pp. 140862–140875, 2020.
[52] E. S. A. Shahri, A. Alfi, and J. A. T. Machado, "Fractional fixed-structure H8 controller design using augmented lagrangian particle swarm optimization with fractional order velocity," *Appl. Soft Comput.*, vol. 77, pp. 688–695, Apr. 2019.
[53] J. T. Machado, S. M. A. Pahnehkolaei, and A. Alfi, "Complex-order particle swarm optimization," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 92, Jan. 2021, Art. no. 105448.
[54] Y. Mousavi and A. Alfi, "A memetic algorithm applied to trajectory control by tuning of fractional order proportional-integral-derivative controllers," *Appl. Soft Comput.*, vol. 36, pp. 599–617, Nov. 2015.
[55] H. A. Varol and Z. Bingul, "A new PID tuning technique using ant algorithm," in *Proc. Amer. Control Conf.*, vol. 3, 2004, pp. 2154–2159.
[56] Z. Bingul, "A new PID tuning technique using differential evolution for unstable and integrating processes with time delay," in *Proc. Int. Conf. Neural Inf. Process.* Berlin, Germany: Springer, 2004, pp. 254–260.
[57] Z. Bingul and O. Karahan, "Comparison of PID and FOPID controllers tuned by PSO and ABC algorithms for unstable and integrating systems with time delay," *Optim. Control Appl. Methods*, vol. 39, no. 4, pp. 1431–1450, Jul. 2018.
[58] Z. Bingul and O. Karahan, "A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system," *J. Franklin Inst.*, vol. 355, no. 13, pp. 5534–5559, Sep. 2018.



BAOYONG GUO received the B.S. degree from Shandong Agricultural University, China, in 2012, and the M.S. degree in basic mathematics from the Shandong University of Science and Technology, China, in 2015, where he is currently pursuing the Ph.D. degree in system engineering. His current research interests include differential equation, Soliton theory, integrable system, control theory, and swarm intelligence.

ZHONGJIE ZHUANG received the B.S. degree in network engineering from Shandong Agricultural University, and the M.S. degree in computer application technology from the Shandong University of Science and Technology, where she is currently pursuing the Ph.D. degree. Her current research interests include swarm intelligence, pattern recognition, and image processing.

JENG-SHYANG PAN (Senior Member, IEEE) received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology, in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from The University of Edinburgh, U.K., in 1996. He is currently a Professor of the Shandong University of Science and Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section.

SHU-CHUAN CHU received the Ph.D. degree from the School of Computer Science, Engineering and Mathematics, Flinders University, Australia, in 2004. She joined Flinders University, Australia, in December 2009. After nine years, she was with Cheng Shiu University, Taiwan. She has been a Research Fellow with the College of Science and Engineering, Flinders University, since December 2009. She is currently a Ph.D. Advisor with the College of Computer Science and Engineering, Shandong University of Science and Technology, since September 2019. Her research interests are mainly in swarm intelligence, intelligent computing, and data mining.