ICIC Express Letters
Volume **12**, Number **1**, January **2018**
ICIC International ©2018 ISSN 1881-803X
pp. **87–96**

DESIGN OF FRACTIONAL ORDER PID CONTROLLER FOR VELOCITY OF MICRO INTELLIGENT VEHICLES

Hong Guan[1,2,3] and Yangzhou Chen[1,2,3,*]

[1]College of Metropolitan Transportation
Beijing University of Technology
[2]Beijing Key Laboratory of Transportation Engineering
[3]Beijing Collaborative Innovation Center for Metropolitan Transportation
No. 100, Pingleyuan, Chaoyang Dist., Beijing 100124, P. R. China
liangguanhong@emails.bjut.edu.cn; *Corresponding author: yzchen@bjut.edu.cn

Received July 2017; accepted September 2017

Abstract. *Fast and precise control of velocity is one of the key factors for the vehicles to obtain good control quality in their movement. The paper introduces fractional order $PI^\lambda D^\mu$ (FOPID) controller into velocity control of micro intelligent vehicles (MicroIVs). Because the selection of parameters for FOPID controller with two additional parameters λ and μ is more difficult than traditional integer order PID (IOPID) controller, an improved back propagation (BP) neural network is proposed and used in the parameters tuning of FOPID controllers. The implementation of FOPID controller is also difficult because fractional calculus operators of FOPID controller cannot be directly implemented in numerical calculation. Fractional order calculus is transformed from continuous time domain to discrete time domain by using the Al-Alaoui generating function, and is discretized by using the continued fraction expansion (CFE). FOPID controller is implemented in motor velocity control for MicroIVs, and the robustness and rapidity of FOPID controller are verified by the methods of this paper. The results of experiment show that FOPID controller has better control performance than IOPID controller.*
Keywords: Fractional order $PI^\lambda D^\mu$ controller, BP neural network, Parameters self-tuning, Al-Alaoui and CFE

1. **Introduction.** Integer order PID (IOPID) controller has been widely used in various fields of industrial control, e.g., metallurgy, machinery, electric power, and chemical industry [1,2], because of its simple principle, easy to use, strong adaptability and robustness, and so on. IOPID controller is in a dominant position for motor control. With control object becoming more complex, controllers are put forward of higher requirements, and the conventional IOPID controller is difficult to obtain satisfactory control effect [3,4]. Fractional order $PI^\lambda D^\mu$ (FOPID) controller retains all the excellent characteristics of IOPID controller, and has more flexible control capability for complex control object because of the expansion of calculus in IOPID controller.

The theory of fractional calculus can be traced back to the work of Lebnitz, one of the founders of calculus, which is only in the study of theory in the early stages. Fractional calculus began to be applied in some fields in recent years, and its application in control systems is mainly the modeling of fractional order systems and the design of fractional order controllers. At present, there are four types of fractional order controllers: TID controller [5], CRONE controller [6], fractional order $PI^\lambda D^\mu$ controller [7] and lead-lag compensator [8]. Among them, fractional order $PI^\lambda D^\mu$ controller proposed by Podlubny in 1999 is of great significance to the application and development of fractional order calculus. Its structure is the same as IOPID controller, but the parameters that affect the performance of controller are increased by two, integral order λ and derivative order μ.

Compared with IOPID controller, FOPID controller makes control system obtain better dynamic performance and robustness.

Parameters tuning is a hot topic in the research of fractional order control theory, but it has not yet explored a suitable parameter tuning method for fractional order controller. With the development of simulation tools, researchers put forward some new tuning methods [9,10] for the parameters tuning of FOPID controller. Cao et al. [11] proposed an intelligent optimization method for designing FOPID controller based on genetic algorithms (GA), and analyzed the optimization design process in detail. ITAE is used as the optimization performance target. In [12], the authors presented an intelligent optimization method for designing FOPID controller by optimizing the SIWPSO position values to minimize the IAE fitness value in an iteration process. PSO technique is simple in encoding with real number, while GA with binary strings. Maamar and Rachid [13] combined the properties of FOPID and internal model control (IMC) to form IMC-PID-fractional-filter controller. A PSO-BP neural networks based PID self-tuning controller is constructed in [14] by utilizing particle swarm optimization algorithm to optimize the connection weight matrix of BP neural networks, and its effectiveness is verified only by simulation. In this paper, parameters tuning method of FOPID is based on an improved BP neural network, and the effectiveness of the proposed method is verified through simulation and experiments of real MicroIVs.

In order to apply FOPID controller in the control of velocity for real vehicles, it is necessary to convert fractional calculus in FOPID controller into a programmable integer order expression by discretizing fractional calculus. In general, the discretization methods of fractional calculus are divided into direct and indirect discretization method [15], and the commonly used direct discretization method is the method based on the combination of Euler, Tustin or Al-Alaoui generating function and continued fraction expansion (CFE) [16-18]. In this paper, Al-Alaoui generating function and CFE are utilized to discretize fractional calculus in FOPID.

The main contributions of this paper are summarized in the following threefold. Firstly, an effective parameters tuning method of FOPID controller based on an improved BP neural network is proposed. The steps of parameters tuning of FOPID based on the improved BP neural network are introduced in detail, which provides a new idea for parameter self-tuning. Secondly, an idea that FOPID controller is discretized and transformed into programmable expression is presented. Thirdly, the effectiveness of FOPID is verified by the experiments on real MicroIVs, which provides practical significance for practical application of FOPID controller.

The remainder of the paper is organized as follows. The concepts of fractional calculus and FOPID are described in Section 2. The method of parameters tuning of FOPID based on BP neural network is proposed in Section 3. Discretization method of FOPID is provided in Section 4, and commonly used generating functions are introduced. In Section 5, the results of parameter tuning and discretization of FOPID are verified on the velocity tuning of MicroIVs in cooperative vehicle-infrastructure simulation platform (CVISP). Finally, the conclusion is provided and the future work is discussed.

2. **Fractional Order $PI^{\lambda}D^{\mu}$ Controller.** Fractional calculus mainly studies the properties and applications of differential and integral operators of any order. The mathematical model of fractional calculus can describe the dynamic response of system more accurately, and improve the ability of the design, characterization and control of the dynamic system. IOPID controller is the most widely used and the most mature technology in the control system, so it is necessary to research the practical application of FOPID controller. FOPID controller extends the integral and differential order in the IOPID controller to any real number. It means that the tuning parameters of FOPID controller are five and the design flexibility of controller is improved.

2.1. **Fractional calculus.** The order of fractional calculus can be real or complex, and the fundamental operator of fractional calculus is ${}_aD_t^\alpha$, where t and a are the upper-lower limit of operator respectively, and α is the fractional calculus order. The fractional calculus is defined by

$$ {}_aD_t^\alpha f(t) = \begin{cases} \dfrac{d^\alpha}{dt^\alpha} f(t), & \mathrm{Re}(\alpha) > 0 \\ f(t), & \mathrm{Re}(\alpha) = 0 \\ \int_a^t f(\tau)(d\tau)^{-\alpha}, & \mathrm{Re}(\alpha) < 0 \end{cases} \tag{1} $$

where the $f(t)$ is an integral or differential object, and $\mathrm{Re}(\alpha)$ is the real part of α.

Up to now, there is no uniform definition for fractional calculus. In the process of establishing the theory of fractional calculus, some of the commonly used definitions are the Grünwald-Letnikov (GL), Riemann-Liouville (RL), Caputo and Cauchy definition [19] of fractional calculus, and the best known one of them is GL definition of fractional calculus given by

$$ {}_aD_t^\alpha f(t) = \lim_{h\to 0} h^{-\alpha} \sum_{j=0}^{[(t-\alpha)/h]} (-1)^j \begin{pmatrix} \alpha \\ j \end{pmatrix} f(t - jh) \tag{2} $$

where $[(t-\alpha)/h]$ is approximate recursive number of items and $\begin{pmatrix} \alpha \\ j \end{pmatrix} = \frac{\alpha!}{j!\cdot(\alpha-j)!} = \frac{\Gamma(\alpha+1)}{j!\cdot\Gamma(\alpha-j+1)}$ is recursive function coefficient. The Laplace transform of ${}_aD_t^\alpha f(t)$ is described as

$$ L\left\{{}_0D_t^\alpha f(t)\right\} = s^\alpha F(s) - \sum_{k=0}^{w-1} s^k {}_0D_t^{\alpha-k-1} f(t)|_{t=0} \tag{3} $$

where w is a positive integer satisfying $w - 1 < \alpha < w$. For fractional order differential equations, the general expression of fractional order transfer function can be given as

$$ G(s) = \frac{b_m s^{\beta_m} + \cdots + b_1 s^{\beta_1} + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + \cdots + a_1 s^{\alpha_1} + a_0 s^{\alpha_0}} \tag{4} $$

and the discrete transfer function is

$$ G(z) = \frac{b_m(\omega(z^{-1}))^{\beta_m} + \cdots + b_1(\omega(z^{-1}))^{\beta_1} + b_0(\omega(z^{-1}))^{\beta_0}}{a_n(\omega(z^{-1}))^{\alpha_n} + \cdots + a_1(\omega(z^{-1}))^{\alpha_1} + a_0(\omega(z^{-1}))^{\alpha_0}} \tag{5} $$

where a_n and b_m are the corresponding denominator coefficients and numerator coefficients of fractional order transfer function for control object, respectively. $\omega(z^{-1})$ is the generating function from the s domain to the z domain.

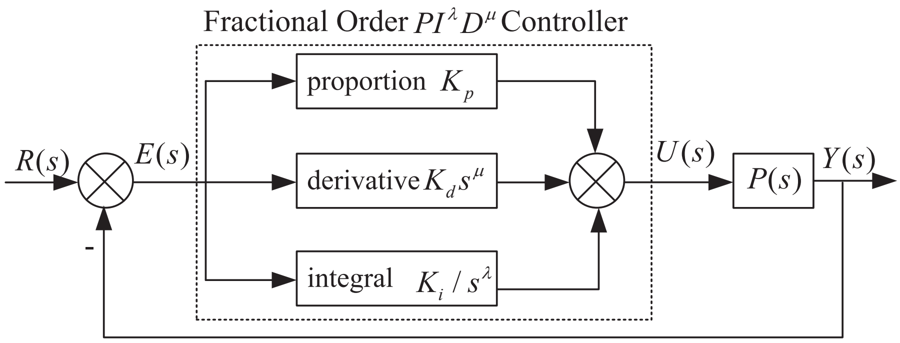
2.2. **The fractional order $PI^\lambda D^\mu$ controllers.** Figure 1(a) shows a closed loop control system with fractional order $PI^\lambda D^\mu$ controller, and Figure 1(b) shows the range of integral order λ and derivative order μ in the FOPID controllers, respectively. $R(s)$, $E(s)$, $U(s)$ and $Y(s)$ are the Laplace transform of the reference signal of control system $r(t)$, the system error signal $e(t)$, the output signal of controller $u(t)$, and the actual output signal of control system $y(t)$, respectively. K_p, K_i and K_d are the coefficients of proportion, integral and differential, respectively. $P(s)$ is the transfer function of control object.

The FOPID controller in Figure 1(a) is

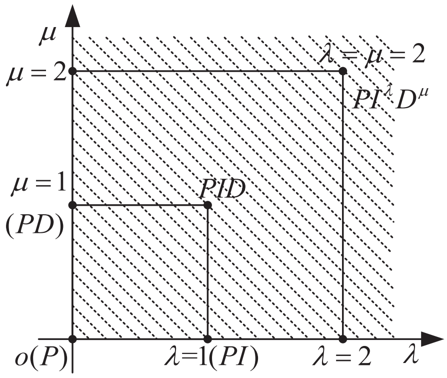
$$ u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t) \tag{6} $$

and its continuous transfer function from $e(t)$ to $u(t)$ is

$$ G_c(s) = K_p + K_i s^{-\lambda} + K_d s^\mu \tag{7} $$



(a) System structure based on FOPID controller



(b) Different order of FOPID controllers

FIGURE 1. FOPID controllers

Figure 1(b) shows that FOPID expands the order of calculus in IOPID from point to plane, and the order of calculus in FOPID can be any real number greater than zero. FOPID will be IOPID if $\lambda = \mu = 1$. It enhances the design flexibility of controller. For FOPID controller, λ mainly affects the steady-state error of system, e.g., the system will be unstable if λ is too large. The dynamic characteristics of system can be improved when μ increases, but too large μ may result in increased tuning time and system unstable. Experiments show that the values of λ and μ are usually between 0 and 2.

3. **Design of FOPID Controller Based on BP Neural Network.** The back propagation (BP) neural network was proposed firstly by professor Rumelhart and McClelland [20] in 1986. The structure of BP neural network includes an input layer, one or more hidden layers and one output layer. Based on the approximation theory of multilayer feedforward network, the number of neurons in the hidden condition enough, arbitrary precision of arbitrary nonlinear mapping from the input space to output space can be achieved by three neural networks with nonlinear activation function of sigmoid. Therefore, this paper only discusses the three layer feedforward neural network. In order to ensure that the obtained useful input information is sufficient without increasing the turning burden of the hidden layer, four related information with the system error are selected as the input layer. Because the FOPID controller has five parameters required tuning, the number of output layer nodes of BP neural network is set to be five. And IOPID has three parameters required tuning, and the number of output nodes only needs to be changed to three when tuning the parameters of IOPID controller. Therefore, the parameter tuning method for FOPID controller based on BP neural network proposed in this paper, which can tune the parameters of FOPID and IOPID controllers at the same time. The structure of the method is shown in Figure 2.

The number of nodes for input layer and hidden layer is set as 4 and 5, respectively. As Figure 2 shows, e, e_{-1} and e_{-2} are the current error, last error and previous error. a,

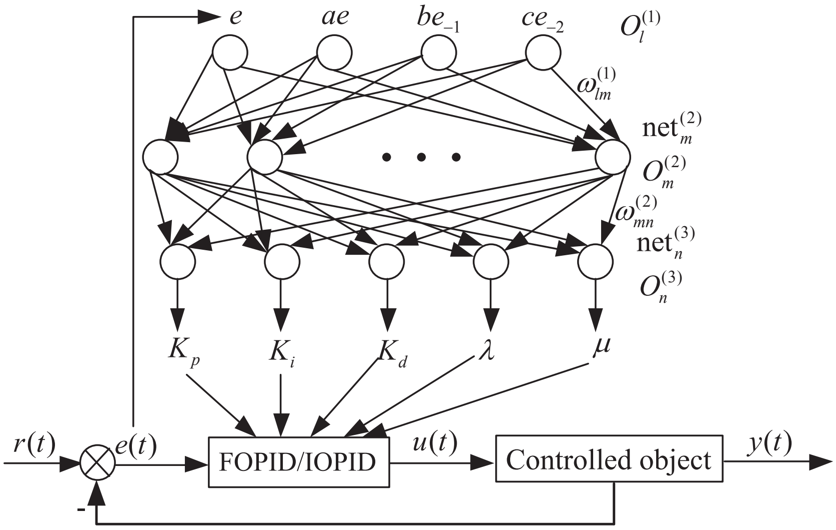


Figure 2. Design of FOPID controller based on BP neural network

b and c are the random coefficients, and $\mathrm{net}_m^{(2)}$ and $O_m^{(2)}$ represent the input and output of the mth node for the second layer of BP neural network. $\omega_{lm}^{(1)}$ is the weight from the output of the lth node in the first layer to the input of the mth node in the second layer. Combined with the structure of BP neural network for FOPID in Figure 2, the expression of input layer can be described as

$$O^{(1)}(k) = \left[O_1^{(1)}(k), O_2^{(1)}(k), O_3^{(1)}(k), O_4^{(1)}(k)\right] = [e, ae, be_{-1}, ce_{-2}] \tag{8}$$

where k is the training times. And the expression of hidden layer can be described as

$$\mathrm{net}_m^{(2)}(k) = \sum_{k=1}^{4} \omega_{lm}^{(1)}(k) O_l^{(1)}(k) \tag{9}$$

$$O_m^{(2)}(k) = f(\mathrm{net}_m^{(2)}(k)), \quad m = 1, 2, 3, 4, 5 \tag{10}$$

where $f(x) = r \cdot (e^x - e^{-x})/(e^x + e^{-x})$ is a tan-sigmoid function, $r > 0$ is a constant which can be appropriately chosen according to the type of controllers, and the value of r is generally set as $r = 1$. The expression of output layer is given as

$$\mathrm{net}_n^{(3)}(k) = \sum_{m=1}^{5} \omega_{mn}^{(2)}(k) O_m^{(2)}(k) \tag{11}$$

$$O_n^{(3)}(k) = g\left(\mathrm{net}_n^{(3)}(k)\right), \quad n = \begin{cases} 1, 2, 3, 4, 5 & \text{for FOPID} \\ 1, 2, 3 & \text{for IOPID} \end{cases} \tag{12}$$

which means $O^{(3)}$ is $[K_p, K_i, K_d, \lambda, \mu]$ for FOPID and $[K_p, K_i, K_d]$ for IOPID. This paper only discusses parameters tuning of FOPID controller. The parameters K_p, K_i, K_d, λ and μ in FOPID controller are non-negative, so the transfer function of output layer is set to a non-negative function as

$$g(x) = v \cdot (1 + \tan\mathrm{sigmoid}(x))/2 = v \cdot e^x / (e^x + e^{-x}) \tag{13}$$

where $\tan\mathrm{sigmoid}(x) = (e^x - e^{-x})/(e^x + e^{-x})$, and v is a constant greater than zero, and the value of v generally meets $0 < v \leq 2$ for FOPID controller.

So far, the design structure of FOPID controller based on BP neural network has been completed of a large part. The formula of error evaluation index is selected as

$$E(k) = \frac{1}{2} e(k)^2 = \frac{1}{2}(r(k) - y(k))^2 \tag{14}$$

where $r(k)$ and $y(k)$ are the input and output of control system, and $e(k)$ is their error. The tuning of parameters of FOPID will be stopped when the error enters the required range. To tune the weights of BP, the gradient descent method is introduced in the parameters tuning of weights in BP neural network. According to the gradient descent method, the weights tuning formula is

$$\Delta\omega_{mn}(k+1) = (1-\gamma)\eta\frac{\partial E(k)}{\partial \omega_{mn}} + \gamma\Delta\omega_{mn}(k) \tag{15}$$

where γ and η are momentum factor and adaptive learning rate of neural network, and γ is generally about 0.95 and η is generally taken from 0.01 to 1. Formula (15) is calculated by the following formulas:

$$\gamma = \begin{cases} 0 & E(k+1) > 1.04E(k) \\ 0.95 & E(k+1) < E(k) \\ \gamma & \text{other} \end{cases} \tag{16}$$

$$\eta(k+1) = \begin{cases} 1.05\eta(k) & E(k+1) > 1.04E(k) \\ 0.7\eta(k) & E(k+1) < E(k) \\ \eta(k) & \text{other} \end{cases} \tag{17}$$

$$\frac{\partial E(k)}{\partial \omega_{mn}(k)} = \frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial O_{mn}^{(3)}(k)} \cdot \frac{\partial O_{mn}^{(3)}(k)}{\partial \omega_{mn}(k)} \tag{18}$$

4. **Implementation and Performance Analysis of FOPID.** Fractional calculus has been applied to the fields of controller, filter design and analysis, image processing and signal analysis. As the fractional order controller is equivalent to the infinite order integer order controller, the key problem of the fractional order controller is the effective discretization of fractional order calculus. At present, the discretization methods of fractional calculus mainly include direct and indirect methods, and direct methods are considered in this paper. The idea of direct discretization is to replace the fractional calculus operator $s^{\pm\alpha}$ with a generating function $(w(z^{-1}))^{\pm\alpha}$, and the irrational function $(w(z^{-1}))^{\pm\alpha}$ is approximated by using a finite order rational function in discrete time domain, z domain.

The commonly used generating functions are Eular, Tustin, and Al-Alaoui generating functions. And Al-Alaoui generating function is obtained by a linear combination of 3/4 Eular and 1/4 Tustin, which is used to overcome the approximation error of high frequency. For Eular, Tustin and Al-Alaoui, $(w(z^{-1}))^{\pm\alpha}$ is $((1-z^{-1})/T)^{\pm\alpha}$, $(2/T \cdot (1-z^{-1})/(1+z^{-1}))^{\pm\alpha}$ and $((8/7T) \cdot (1-z^{-1})/(1+z^{-1}/7))^{\pm\alpha}$, respectively, where T is sampling period. Al-Alaoui generating function has the best phase characteristics.

After the fractional calculus operator is transformed from the s domain to the z domain by using generating function, the discrete approximation function $H(z^{-1})$ can be obtained by the methods of power series expansion (PSE) and continued fraction expansion (CFE) or other methods [21]. In this paper, CFE is adopted. and the combination of CFE with Al-Alaoui generating function can get a more ideal fractional integrator. The discrete model is given as

$$D^{\pm\alpha}(z) = \left(\frac{8}{7T}\right)^{\pm\alpha} \text{CFE}\left\{\left(\frac{1-z^{-1}}{1+\frac{z^{-1}}{7}}\right)^{\pm\alpha}\right\}_{p,q} \tag{19}$$

where $-1 \le \alpha \le 1$ is calculus order, and CFE$\{f\}$ is the CFE of function f. And the expression is simplified for $D^{\pm\alpha}(z) = (8/7T)^{\pm\alpha} \cdot P_p(z^{-1})/Q_q(z^{-1})$, and p and q are the order of $P_p(z^{-1})$ and $Q_q(z^{-1})$ which are the polynomials of variable z^{-1}. The values of p and q are generally equal and are positive integers.

TABLE 1. The expression of $D^{\pm\alpha}(z)$ based on Al-Alaoui method

p	q	$P_p(z^{-1})\ (\kappa=1),\ Q_q(z^{-1})\ (\kappa=0)$
1	1	$((-1)^\kappa 4\alpha - 3)z^{-1} + 7$
2	2	$(16\alpha^2 - (-1)^\kappa 36\alpha + 11)z^{-2} + ((-1)^\kappa 84\alpha - 126)z^{-1} + 147$
3	3	$((-1)^\kappa 64\alpha^3 - 288\alpha^2 + (-1)^\kappa 284\alpha + 27)z^{-3}$ $+(672\alpha^2 - (-1)^\kappa 2520\alpha + 1827)z^{-2}$ $+((-1)^\kappa 2940\alpha - 6615)z^{-1} + 5145$

In Table 1, the approximate numerator and denominator expression of $D^{\pm\alpha}(z)$ based on Al-Alaoui method are given while $p = q = 1, 2, 3$. According to the experience of previous researchers, and that the computational complexity and control performance of system are considered, the value of p and q is generally set as 3. The fractional calculus will be divided into an integer calculus and a decimal calculus when the absolute value of calculus order is greater than 1. For example, $D^{1.5}(s)$ can be expressed as $D^{1.5}(s) = s \cdot D^{0.5}(s)$. The discrete approximation function of $D^{0.5}(s)$ can be obtained by the above method, which is shown as Formula (20) while $p = q = 2$.

$$D^{0.5}(s) = \frac{33.8062z^2 - 38.6356z + 7.5891}{z^2 - 0.5714286z - 0.0204082} \tag{20}$$

For FOPID controller, the discretization function of Formula (7) is

$$G_c(z) = K_p + \frac{K_i}{\left(\frac{8}{7T}\right)^{\pm\lambda} \frac{P_p(z^{-1},\lambda)}{Q_q(z^{-1},\lambda)}} + K_d \left(\frac{8}{7T}\right)^{\pm\mu} \frac{P_p(z^{-1},\mu)}{Q_q(z^{-1},\mu)} \tag{21}$$

where we make $K_\lambda = (8/7T)^{\pm\lambda}$, $K_\mu = (8/7T)^{\pm\mu}$, $P_\lambda = P_p(z^{-1},\lambda)$, $Q_\lambda = Q_q(z^{-1},\lambda)$, $P_\mu = P_p(z^{-1},\mu)$ and $Q_\mu = Q_q(z^{-1},\mu)$, and Formula (21) can be simplified as

$$G_c(z) = \frac{K_p K_\lambda P_\lambda Q_\mu + K_i Q_\lambda Q_\mu + K_d K_\lambda K_\mu P_\lambda P_\mu}{K_\lambda P_\lambda Q_\mu} \tag{22}$$

For P_λ in Formula (22), the value of p is set as 2 and P_λ is given as

$$\begin{aligned} P_\lambda &= (16\lambda^2 + 36\lambda + 11)z^{-2} + (-84\lambda - 126)z^{-1} + 147 \\ &= \left[\begin{pmatrix} 16 & 36 & 11 \\ 0 & -84 & -126 \\ 0 & 0 & 147 \end{pmatrix}\begin{pmatrix} \lambda^2 \\ \lambda^1 \\ 1 \end{pmatrix}\right]^T \begin{pmatrix} z^{-2} \\ z^{-1} \\ 1 \end{pmatrix} = \left[P\bar{\lambda}\right]^T Z \end{aligned} \tag{23}$$

By the methods of Formula (23), all the polynomials in Formula (22) are expressed in matrix, and the discretization and programmability of FOPID are realized.

Based on the above, FOPID is implemented by the simulation in MATLAB firstly. An example of heating furnace is considered in [9], and the control object model of integer and fractional order for heating furnace are given. We take the integer order controlled object as the example in this paper, which means $P(s) = 1/(73043s^2 + 4893s + 1.93)$. Different papers give different examples for this classic example, and [9] gives an IOPID controller. The methods in [11] and this paper are utilized to tune the parameters in FOPID for the control object of heating furnace, so two kinds of FOPID controllers are produced. For FOPID controller of this paper, only the value of K_p increased obviously in all output when v in Formula (13) was increased, the sampling period $T = 0.01$. After tuning, $K_p = 31726.913$, $K_i = 0.865$, $K_d = 0.286$, $\lambda = 0.772$ and $\mu = 0.461$. The step response of system based on these three controllers is shown as Figure 3.

Figure 3 shows that the method of this paper for parameters tuning of FOPID controllers has a great advantage in the aspect of rising time, regulating time and overshoot,

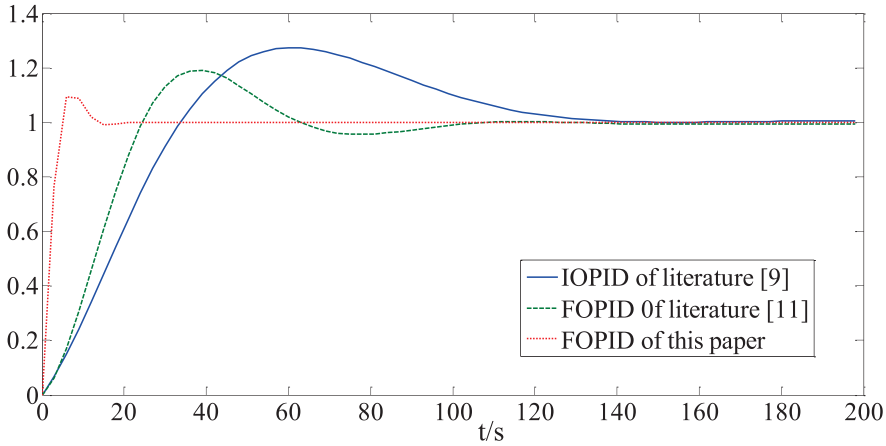


FIGURE 3. Step responses of system with different controllers

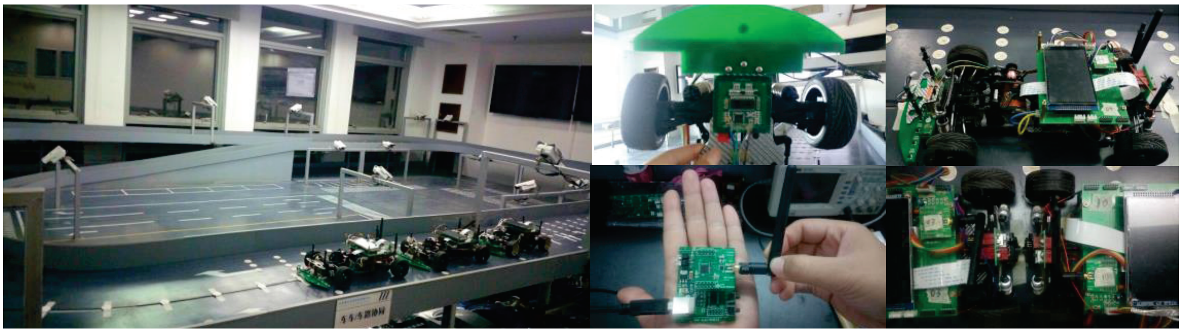


FIGURE 4. Cooperative vehicle-infrastructure simulation platform

and so on. At the same time, the tuning of r in Formula (10) and v in Formula (13) improves flexibility and range of parameter tuning.

5. **Application of FOPID in Velocity Control of Micro Intelligent Vehicles.** In this section, we apply the designed FOPID to the velocity control of a MicroIV, which shows that the proposed FOPID has a practical signification. The application is based on the micro intelligent vehicles in cooperative vehicle-infrastructure simulation platform (CVISP) as Figure 4. The quality of velocity adjustment for micro intelligent vehicles is mainly based on its rapidity and accuracy, and the experiment will show that the tuning quality of FOPID is better than IOPID on the same platform and the same road environment.

For the experiment of velocity tuning of real vehicles, IOPID controller and FOPID controller are applied in a same motor of MicroIV, respectively. The results about velocity tuning with IOPID and FOPID controllers are shown in Figure 5 and Figure 6, the vertical axis is the displacement of vehicle, and the horizontal axis is time. Among them, Figure 5 and Figure 6 are the displacement curves of vehicle based on IOPID controller and FOPID controller, respectively.

The velocity of vehicle is set as 40 cm/s in Figure 5. And the velocity of vehicle is set as 40 cm/s at first, after a period of time, which is set to 60 cm/s at the position of the red dot in Figure 6. Obviously, there are many small polyline in the displacement curve in Figure 5, which means that the value of velocity is fluctuating and it will be more obvious when vehicle moves on the ground with protrusions or depressions. However, the displacement curve is a smooth polyline without small polyline in Figure 6, even when the velocity changes sharply, the adjustment of the velocity is also fast and there are not

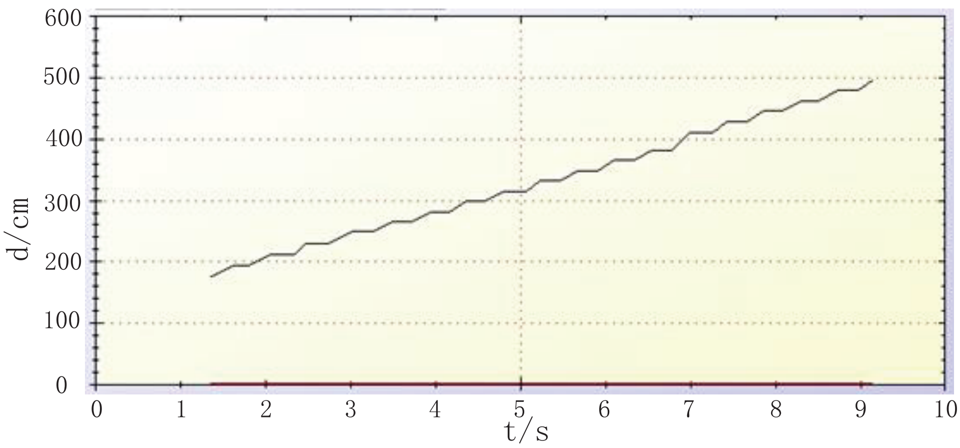


Figure 5. The displacement of vehicle based on IOPID

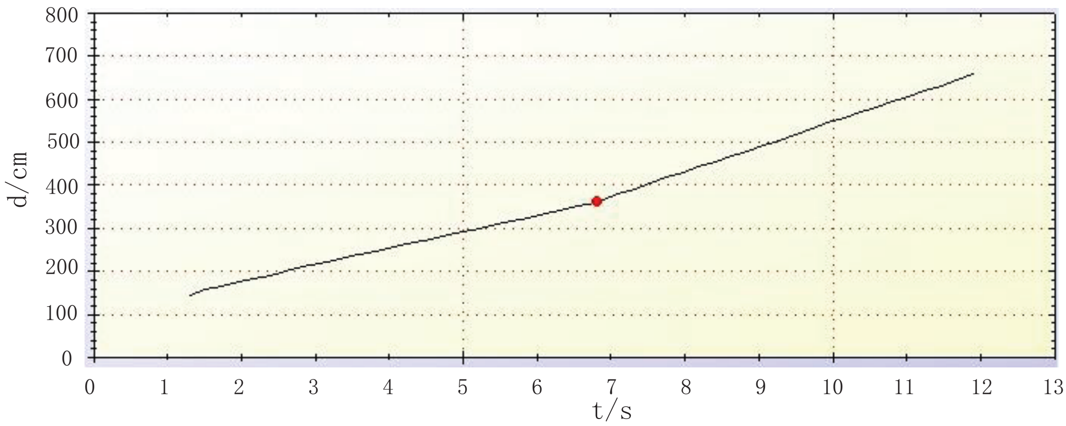


Figure 6. The displacement of vehicle based on FOPID

large fluctuations. The results show that FOPID controller has shorter adjustment time and faster regulation speed than IOPID controller for the control system.

6. **Conclusions.** The paper presented an effective parameter tuning method of fractional order PID controller based on BP neural network. The connection function between the hidden layer and output layer had been modified a little, which improved the flexibility of parameter self-tuning methods based on BP neural network. What is more, the method of discretization and programmability for FOPID was presented. The method was used in the velocity tuning of MicroIVs successfully and made the application fractional calculus become more vivid and practical significance.

In the future research, we will tune parameters of FOPID controller in more complex control system. For improving the adaptability of the method of this paper, we will tune the output function or other links of BP neural network by comparing the results of parameters tuning of controllers with the different methods in different control systems. Finally, a more efficient and easier method to implement parameter self-tuning will be proposed.

Acknowledgment. The research was supported by National Natural Science Foundation of China (Grant No. 61573030, 61511130044).

REFERENCES

[1] H. Z. Li, L. Li and M. X. Kang, PID plus fuzzy logic method for torque control in traction control system, *International Journal of Automotive Technology*, vol.13, no.3, pp.441-450, 2012.

[2] M. D. Pham and H. H. Lee, Adaptive control of DC-DC converter based on hybrid fuzzy PID controller, *International Conference on Intelligent Computing*, Springer, Cham, pp.253-263, 2017.

[3] M. A. S. Aboelela, M. F. Ahmed and H. T. Dorrah, Design of aerospace control systems using fractional PID controller, *Journal of Advanced Research*, vol.3, no.3, pp.225-232, 2012.

[4] A. Mesbahi and M. Haeri, Robust non-fragile fractional order PID controller for linear time invariant fractional delay systems, *Journal of Process Control*, vol.24, no.9, pp.1489-1494, 2014.

[5] B. J. Lurie, *Three-Parameter Tunable Tilt-Integral-Derivative (TID) Controller: US*, US 5371670 A, 1994.

[6] A. Oustaloup, B. Mathieu and P. Lanusse, The CRONE control of resonant plants: Application to a flexible transmission, *European Journal of Control*, vol.1, no.2, pp.275-283, 1995.

[7] I. Podlubny, Fractional-order systems and PID-controllers, *IEEE Trans. Automatic Control*, vol.44, no.1, pp.208-214, 1999.

[8] J. F. Wang and L. I. Y. Kai, Design for two types of fractional-order controllers: $PI^{\lambda}D^{\mu}$ controller and lead-lag compensator, *Journal of Circuits and Systems*, vol.11, no.5, pp.21-25, 2006.

[9] D. Y. Xue and C. N. Zhao, Fractional-order PID controller design for the fractional-order system, *Control Theory and Application*, vol.24, no.5, pp.771-776, 2007.

[10] S. E. Hamamci, Stabilization using fractional-order PI and PID controllers, *Nonlinear Dynamics*, vol.51, no.1, pp.329-343, 2008.

[11] J. Y. Cao, J. Liang and B. G. Cao, Optimization of fractional order PID controllers based on genetic algorithms, *International Conference on Machine Learning and Cybernetics*, vol.9, pp.5686-5689, 2005.

[12] J. W. Perng, G. Y. Chen and Y. W. Hsu, FOPID controller optimization based on SIWPSO-RBFNN algorithm for fractional-order time delay systems, *Soft Computing*, vol.21, no.14, pp.4005-4018, 2017.

[13] B. Maamar and M. Rachid, IMC-PID-fractional-order-filter controllers design for integer order systems, *ISA Transactions*, vol.53, no.5, pp.1620-1628, 2014.

[14] K. Guo, Y. San and Y. Zhu, PID controller parameters optimization based on PSO-BP neural networks, *Electronic Design Engineering*, vol.4, no.21, 2012.

[15] B. T. Krishna and K. V. V. S. Reddy, Design of fractional order digital differentiators and integrators using indirect discretization, *Fractional Calculus and Applied Analysis*, vol.11, no.2, pp.143-151, 2008.

[16] Y. Q. Chen and K. L. Moore, Discretization schemes for fractional-order differentiators and integrators, *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol.49, no.3, pp.363-367, 2002.

[17] C. Li, Y. Q. Chen and J. Kurths, Fractional calculus and its applications, *Philosophical Transactions of the Royal Society*, 2013.

[18] B. M. Vinagre, I. Petras and P. Merchan, Two digital realizations of fractional controllers: Application to temperature control of a solid, *European Control Conference (ECC)*, pp.1764-1767, 2001.

[19] J. Spanier, *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*, Academic Press, New York, 1974.

[20] D. E. Rumelhart and J. L. McClelland, *PDP Research Group, Parallel Distributed Processing*, MIT Press, Cambridge, MA, USA, 1987.

[21] T. V. Moghaddam, N. Bigdeli and K. Afshar, Tuning a fractional order PID controller with lead compensator in frequency domain, *World Academy of Science, Engineering and Technology*, vol.75, pp.1278-1283, 2011.