



ELSEVIER

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: [www.elsevier.com/locate/isatrans](http://www.elsevier.com/locate/isatrans)

Research article

# Fractional order $PI^\lambda D^\mu$ controller design for satisfying time and frequency domain specifications simultaneously

Weijia Zheng<sup>a,b</sup>, Ying Luo<sup>a,c,\*</sup>, XiaoHong Wang<sup>b,\*</sup>, YouGuo Pi<sup>b</sup>, YangQuan Chen<sup>c</sup>

<sup>a</sup>School of Automation, Foshan University, Foshan, China

<sup>b</sup>School of Automation Science and Engineering, South China University of Technology, Guangzhou, China

<sup>c</sup>School of Engineering, University of California, Merced, 5200 North Lake Road, Merced, CA, USA



## ARTICLE INFO

### Article history:

Received 9 February 2016

Received in revised form

24 November 2016

Accepted 28 February 2017

Available online 17 March 2017

### Keywords:

Fractional order  $PI^\lambda D^\mu$  controller design

PMSM servo system

Differential Evolution algorithm

## ABSTRACT

In order to achieve a desired control performance characterized by satisfying specifications in both frequency-domain and time-domain simultaneously, an optimal fractional order proportional integral derivative ( $PI^\lambda D^\mu$ ) controller design strategy is proposed based on analytical calculation and Differential Evolution algorithm for a permanent magnet synchronous motor (PMSM) servo system in this paper. In this controller design, the frequency-domain specifications can guarantee the system stability with both gain margin and phase margin, and also the system robustness to loop gain variations. The time-domain specifications can ensure the desired step response performance with rapid rising curve, constrained overshoot, and proper power consuming. Compared with the  $PI^\lambda$  controller and the traditional PID controller,  $PI^\lambda D^\mu$  controller can get obvious benefits from two more degrees of freedom of the fractional orders  $\lambda$  and  $\mu$  on satisfying multiple constraints simultaneously and achieving better servo tracking performance for the PMSM servo system. PMSM speed tracking simulations and experiments are demonstrated to show the significant advantages of using the proposed optimal  $PI^\lambda D^\mu$  controller over the optimal fractional order  $PI^\lambda$  controller and traditional integer order PID controller.

© 2017 ISA. Published by Elsevier Ltd. All rights reserved.

## 1. Introduction

In recent years, fractional calculus has been widely used in system modeling [1–4] and control area [5–9]. The characteristics of real-world systems can be described more precisely using fractional order mathematical models [10,11]. Meanwhile, proportional integral derivative (PID) control has been the most widely used and developed control method in industrial control area [12–15]. Fractional order proportional integral derivative ( $PI^\lambda D^\mu$ ) controller has the potential to achieve better control performance over the traditional PID controller because the differential order and integral order are introduced as adjustable controller parameters, which increase the flexibility of the controller [16,7–9,17–20]. But the tuning of  $PI^\lambda D^\mu$  controller is more complicated over the traditional PID controller because two extra parameters are added [16,21–25]. Especially, how to design an optimized  $PI^\lambda D^\mu$  controller to achieve desired performance specified in both frequency-domain and time-domain is deserved to be investigated.

In present, the tuning methods for the fractional order  $PI^\lambda D^\mu$  controller mainly contain the frequency-domain design methods

[26] and other time-domain optimization methods [27–29]. The frequency-domain method is often applied to design the fractional order  $PI^\lambda$  or  $PD^\mu$  controllers [21,26]. As presented in [26], based on the given gain crossover frequency and phase margin, the controller parameters are calculated according to the gain robustness specification. The obtained control system can achieve the robustness to gain variations. However, this frequency-domain method may not be directly applied to design the fractional order  $PI^\lambda D^\mu$  controller, because two extra parameters are introduced. Meanwhile, the system gain margin is an important stability index in real industrial control applications, but it is always ignored in the frequency-domain method. The time-domain optimization methods search for the optimal controller parameters by optimizing an objective function [27,28]. The obtained control system can achieve the optimal time-domain dynamic performance. But the system stability with gain and phase margin, and the robustness performance specified in frequency-domain may not be able to be guaranteed simultaneously.

In order to obtain a controller to achieve good dynamic performance under the condition that both the requirements in time-domain and frequency-domain are satisfied, a fractional order  $PI^\lambda D^\mu$  controller design strategy is proposed in this paper and a fractional order  $PI^\lambda D^\mu$  controller is designed for a permanent magnet synchronous motor (PMSM) servo system, based on

\* Corresponding authors.

analytical calculation and Differential Evolution (DE) algorithm [30]. In frequency-domain, taking the loop-gain robustness specification as a constraint condition, the gain margin and phase margin specifications as the boundary conditions, in time-domain, taking the integral of time and absolute error (ITAE) [31] as the objective function, the step response overshoot upper-limit as the dynamic threshold, the power consumption upper-limit as the energy threshold, an optimal  $PI^\lambda D^\mu$  controller can be obtained by using the DE algorithm [30]. Under this controller design strategy, the frequency-domain specifications can guarantee the control system stability with not only phase margin but also gain margin; the system robustness to loop-gain variations can also be satisfied from a flat-phase specification in frequency-domain; the time-domain specifications can ensure the desired step response performance with rapid rising curve, constrained overshoot, and proper power consuming.

Using this proposed controller design method, the fractional order  $PI^\lambda D^\mu$  controller can obtain obvious benefits from two more degrees of freedom of the fractional orders  $\lambda$  and  $\mu$ . This designed  $PI^\lambda D^\mu$  is able to satisfy multiple constraints in both frequency-domain and time-domain simultaneously. Especially, it can achieve better servo tracking performance over the traditional PID controller for the PMSM servo system with a fractional order model which can describe the real PMSM servo system more precisely over the integer order model [32]. PMSM speed-tracking simulations and experiments are demonstrated to show the advantages of the proposed tuning method over the methods proposed in [29]. Besides, the significant advantages of using the proposed  $PI^\lambda D^\mu$  controller over the fractional order  $PI^\lambda$  controller and traditional integer order PID controller are also demonstrated.

The rest of this paper is arranged as follows: the model of the PMSM servo control system is discussed in Section 2; the fractional order  $PI^\lambda D^\mu$  controller design method is proposed in Section 3; PMSM speed control simulations are presented in Section 4. The obtained  $PI^\lambda D^\mu$  controller is compared with those obtained using the time-domain and frequency-domain tuning methods proposed in [29]. The dynamic performance of the obtained  $PI^\lambda D^\mu$  controller is also studied by comparison with the  $PI^\lambda$  controller and traditional PID controller; real PMSM speed control experiments are presented in Section 5; the conclusion is given in Section 6.

## 2. PMSM speed control system

According to our previous work [32], a fractional order model is able to describe the real PMSM servo system more precisely over the integer order model. Therefore, the fractional order model is applied for the PMSM servo system controller design in this paper. The block diagram of the fractional order model of the PMSM speed control system is shown in Fig. 1, where  $n_r$  is the reference speed,  $n$  is the actual speed,  $C_v(s)$  is the speed controller,  $i_{qr}$  is the speed controller output,  $i_q$  is the q-axis current,  $C_i(s)$  is the current controller,  $K_0$  is the voltage conversion factor,  $K_1$  is the current

conversion factor,  $T_i$  is the current filter time constant,  $K_2$  is the speed conversion factor,  $R$  is the resistor,  $L$  is the inductor,  $C_m$  is the torque constant,  $GD^2$  is the flywheel inertia,  $C_e$  is the induced voltage constant,  $\vartheta \in (0, 2)$  and  $\zeta \in (0, 2)$  are the fractional orders in the model.

In order to obtain the plant model of the speed control system, the current loop is properly simplified. Since the changing of the current is much faster than that of the motor speed, the induced voltage  $E$  can be considered unchanging when studying the variations of the current. Therefore, the influence of the induced voltage  $E$  can be eliminated [33] and the current loop is simplified as shown in Fig. 2.

A proportional-integral (PI) controller is chosen to be the current controller, whose integral time constant is set to be  $T_i$ , as described by (1),

$$C_i(s) = K_{pi} \left( 1 + \frac{1}{T_i s} \right). \tag{1}$$

Then the current loop model is simplified as shown in Fig. 3, where

$$\kappa = \frac{K_{pi} K_0}{T_i L}. \tag{2}$$

Therefore, the closed-loop transfer function of the current loop is described by (3),

$$G_i(s) = \frac{\kappa}{s^{\vartheta+1} + \frac{R}{L}s + K_i \kappa}. \tag{3}$$

The speed control loop can be converted into an unit feedback control system, as shown in Fig. 4.

The plant model of this speed control system can be generalized as the following form,

$$G(s) = \frac{d}{s^\alpha + as^\beta + bs^\gamma + c}, \tag{4}$$

where  $\alpha = \xi + \vartheta + 1$ ,  $\beta = \xi + 1$ ,  $\gamma = \xi$ ,  $a = \frac{R}{L}$ ,  $b = K_i \kappa$ ,  $c=0$ ,  $d = \frac{375 C_m K_0 \kappa}{GD^2}$ .

The fractional order  $PI^\lambda D^\mu$  controller is designed based on this PMSM speed control model described by (4) in this paper.

## 3. Fractional order $PI^\lambda D^\mu$ optimal design method

The fractional order  $PI^\lambda D^\mu$  controller is described by (5),

$$C(s) = K_p \left( 1 + \frac{K_i}{s^\lambda} + K_d s^\mu \right), \tag{5}$$

where,  $K_p$ ,  $K_i$  and  $K_d$  are proportional, integral and derivative gains, respectively;  $\lambda \in (0, 2)$  and  $\mu \in (0, 2)$  are the fractional orders.

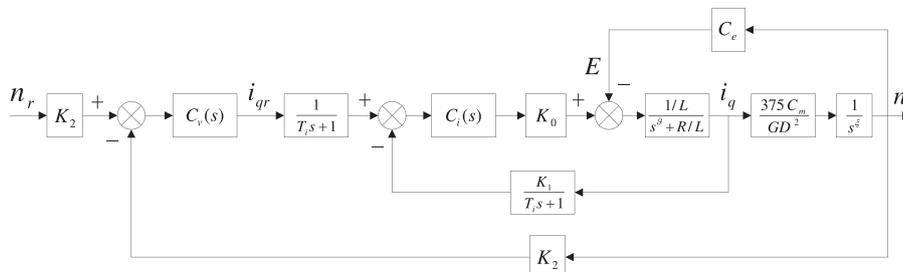


Fig. 1. Diagram of the feedback control system.

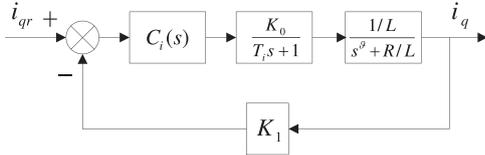


Fig. 2. The simplified model-I of the current loop in Fig. 1.

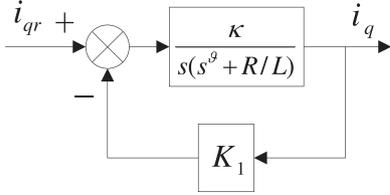


Fig. 3. The simplified model-II of the current loop in Fig. 1.

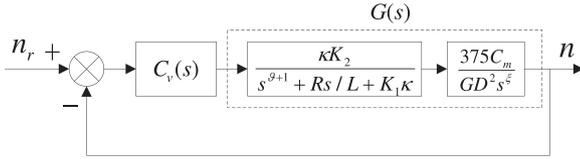


Fig. 4. The block diagram of the speed control loop model.

### 3.1. Controller design specifications

The controller design specifications are selected according to the following three aspects: stability, loop-gain robustness and dynamic performance. Using the frequency-domain design method [26], the phase margin is specified, but there is no constraint for the gain margin. Only the phase margin specification may not be enough to describe the stability of the control system. In order to ensure the comprehensive control system stability, both the phase margin and the gain margin specifications are introduced as the boundary conditions for the controller design in this paper.

The phase margin specification is shown in (6),

$$\text{Arg}[G(j\omega_c)] + \text{Arg}[C(j\omega_c)] \geq -\pi + \varphi_{mb}, \quad (6)$$

where  $\varphi_{mb}$  is the lower bound of the phase margin,  $\omega_c$  is the gain crossover frequency, satisfying

$$|G(j\omega_c)C(j\omega_c)| = 1. \quad (7)$$

Eq. (6) ensures that the phase margin of the control system will not be smaller than  $\varphi_{mb}$ .

The gain margin specification is shown in (8),

$$20\lg|G(j\omega_g)C(j\omega_g)| \leq -L_{gb}, \quad (8)$$

where  $L_{gb}$  is the lower bound of the gain margin,  $\omega_g$  is the phase crossover frequency, satisfying

$$\text{Arg}[G(j\omega_g)] + \text{Arg}[C(j\omega_g)] = -\pi. \quad (9)$$

Eq. (8) ensures that the gain margin of the control system will not be smaller than  $L_{gb}$ .

In order to achieve the robustness to loop-gain variations, the following robustness specification is introduced as the constraint condition,

$$\left. \frac{d[\text{Arg}[G(j\omega)C(j\omega)]]}{d\omega} \right|_{\omega=\omega_c} = 0. \quad (10)$$

The derivative of the phase-frequency characteristic curve is zero, namely, the phase Bode plot is flat around the gain crossover frequency. It means that the system is robust to loop-gain changes and the peak overshoots of the step responses are almost the same [26].

In order to obtain the optimal dynamic performance in time-domain, the ITAE is selected to be the objective function for the tuning algorithm to minimize. The ITAE index is described by (11) [31],

$$J_{ITAE} = \int_0^{\infty} t|e(t)|dt, \quad (11)$$

where  $e(t)$  represents the deviation between the expected output and the actual output.

### 3.2. Controller parameters calculation

According to (4), the amplitude and phase of the plant model are obtained as shown in (12) and (13),

$$|G(j\omega)| = \frac{d}{\sqrt{A(\omega)^2 + B(\omega)^2}}, \quad (12)$$

$$\text{Arg}[G(j\omega)] = -\arctan\left(\frac{B(\omega)}{A(\omega)}\right), \quad (13)$$

where

$$A(\omega) = \omega^\alpha \cos\left(\frac{\pi}{2}\alpha\right) + a\omega^\beta \cos\left(\frac{\pi}{2}\beta\right) + b\omega^\gamma \cos\left(\frac{\pi}{2}\gamma\right) + c, \quad (14)$$

$$B(\omega) = \omega^\alpha \sin\left(\frac{\pi}{2}\alpha\right) + a\omega^\beta \sin\left(\frac{\pi}{2}\beta\right) + b\omega^\gamma \sin\left(\frac{\pi}{2}\gamma\right). \quad (15)$$

According to (5), the amplitude and phase of the controller are obtained as shown in (16) and (17),

$$|C(j\omega)| = K_p \sqrt{P(\omega)^2 + Q(\omega)^2}, \quad (16)$$

$$\text{Arg}[C(j\omega)] = \arctan\left(\frac{Q(\omega)}{P(\omega)}\right), \quad (17)$$

where

$$P(\omega) = 1 + K_f \omega^{-\lambda} \cos\left(\frac{\pi}{2}\lambda\right) + K_d \omega^\mu \cos\left(\frac{\pi}{2}\mu\right), \quad (18)$$

$$Q(\omega) = K_d \omega^\mu \sin\left(\frac{\pi}{2}\mu\right) - K_f \omega^{-\lambda} \sin\left(\frac{\pi}{2}\lambda\right). \quad (19)$$

Given the gain crossover frequency  $\omega_c$ , the phase margin  $\varphi_m$  ( $\varphi_m \geq \varphi_{mb}$ ), the controller orders  $\lambda$  and  $\mu$ , from (6) it gives,

$$\tan(\text{Arg}[G(j\omega_c)] + \text{Arg}[C(j\omega_c)]) = \tan(-\pi + \varphi_m), \quad (20)$$

According to the trigonometric formula, it gives

$$\frac{\tan(\text{Arg}[G(j\omega_c)]) + \tan(\text{Arg}[C(j\omega_c)])}{1 - \tan(\text{Arg}[G(j\omega_c)])\tan(\text{Arg}[C(j\omega_c)])} = \tan(-\pi + \varphi_m). \quad (21)$$

Substituting (13) and (17) into (21), yields

$$\frac{-\frac{B(\omega_c)}{A(\omega_c)} + \frac{Q(\omega_c)}{P(\omega_c)}}{1 + \frac{B(\omega_c)Q(\omega_c)}{A(\omega_c)P(\omega_c)}} = \tan(-\pi + \varphi_m). \quad (22)$$

Denoting  $T$  as  $\tan(-\pi + \varphi_m)$ , substituting (18) and (19) into (22), yields,

$$K_d = s_1 K_i + s_0, \quad (23)$$

where

$$s_1 = \frac{A(\omega_c)\omega_c^{-\lambda}\sin\left(\frac{\pi}{2}\lambda\right) + B(\omega_c)\omega_c^{-\lambda}\cos\left(\frac{\pi}{2}\lambda\right) + TA(\omega_c)\omega_c^{-\lambda}\cos\left(\frac{\pi}{2}\lambda\right) - TB(\omega_c)\omega_c^{-\lambda}\sin\left(\frac{\pi}{2}\lambda\right)}{A(\omega_c)\omega_c^\mu\sin\left(\frac{\pi}{2}\mu\right) - B(\omega_c)\omega_c^\mu\cos\left(\frac{\pi}{2}\mu\right) - TA(\omega_c)\omega_c^\mu\cos\left(\frac{\pi}{2}\mu\right) - TB(\omega_c)\omega_c^\mu\sin\left(\frac{\pi}{2}\mu\right)}, \quad (24)$$

$$s_0 = \frac{B(\omega_c) + TA(\omega_c)}{A(\omega_c)\omega_c^\mu\sin\left(\frac{\pi}{2}\mu\right) - B(\omega_c)\omega_c^\mu\cos\left(\frac{\pi}{2}\mu\right) - TA(\omega_c)\omega_c^\mu\cos\left(\frac{\pi}{2}\mu\right) - TB(\omega_c)\omega_c^\mu\sin\left(\frac{\pi}{2}\mu\right)}, \quad (25)$$

Given the gain crossover frequency  $\omega_c$ , the controller orders  $\lambda$  and  $\mu$ , from flat-phase robustness constraint (10),

$$\left[ \frac{d[\text{Arg}[G(j\omega)]]}{d\omega} \right]_{\omega=\omega_c} + \left[ \frac{d[\text{Arg}[C(j\omega)]]}{d\omega} \right]_{\omega=\omega_c} = 0. \quad (26)$$

Substituting (17) into (26), the equation about  $K_i$  and  $K_d$  can be obtained, as shown in (27),

$$\begin{aligned} & \mu\omega_c^{\mu-1}\sin\left(\frac{\pi}{2}\mu\right)K_d + (\lambda + \mu)\omega_c^{\mu-\lambda-1}\sin\left(\frac{\pi}{2}(\lambda + \mu)\right)K_d K_i \\ & + \lambda\omega_c^{-\lambda-1}\sin\left(\frac{\pi}{2}\lambda\right)K_i + M\omega_c^{2\mu}K_d^2 + 2M\omega_c^{\mu-\lambda}\cos\left(\frac{\pi}{2}(\lambda + \mu)\right)K_d K_i \\ & + 2M\omega_c^\mu\cos\left(\frac{\pi}{2}\mu\right)K_d + M\omega_c^{-2\lambda}K_i^2 + 2M\omega_c^{-\lambda}\cos\left(\frac{\pi}{2}\lambda\right)K_i + M \\ & = 0, \end{aligned} \quad (27)$$

where

$$M = \left[ \frac{d[\text{Arg}[G(j\omega)]]}{d\omega} \right]_{\omega=\omega_c}. \quad (28)$$

Then substituting (23) into (27), the equation about  $K_i$  is obtained, as shown in (29),

$$Q_2 K_i^2 + Q_1 K_i + Q_0 = 0, \quad (29)$$

where

$$\begin{aligned} Q_2 = & M\omega_c^{2\mu}s_1^2 + (\lambda + \mu)\omega_c^{\mu-\lambda-1}\sin\left(\frac{\pi}{2}(\lambda + \mu)\right)s_1 \\ & + 2M\omega_c^{\mu-\lambda}\cos\left(\frac{\pi}{2}(\lambda + \mu)\right)s_1 + M\omega_c^{-2\lambda}, \end{aligned} \quad (30)$$

$$\begin{aligned} Q_1 = & 2M\omega_c^{2\mu}s_1s_0 + \mu\omega_c^{\mu-1}\sin\left(\frac{\pi}{2}\mu\right)s_1 + 2M\omega_c^{\mu-\lambda}\cos\left(\frac{\pi}{2}(\lambda + \mu)\right)s_0 \\ & + (\lambda + \mu)\omega_c^{\mu-\lambda-1}\sin\left(\frac{\pi}{2}(\lambda + \mu)\right)s_0 + 2M\omega_c^\mu\cos\left(\frac{\pi}{2}\mu\right)s_1 \\ & + \lambda\omega_c^{-\lambda-1}\sin\left(\frac{\pi}{2}\lambda\right) + 2M\omega_c^{-\lambda}\cos\left(\frac{\pi}{2}\lambda\right), \end{aligned} \quad (31)$$

$$Q_0 = M\omega_c^{2\mu}s_0^2 + \mu\omega_c^{\mu-1}\sin\left(\frac{\pi}{2}\mu\right)s_0 + 2M\omega_c^\mu\cos\left(\frac{\pi}{2}\mu\right)s_0 + M. \quad (32)$$

$K_i$  can be obtained by solving (29). Therefore, the procedure for calculating the controller parameters are obtained. First, the gain crossover frequency  $\omega_c$ , the phase margin  $\varphi_m$ , the controller orders  $\lambda$  and  $\mu$  are selected. Second,  $K_i$  is obtained by solving (29). Then  $K_d$  is obtained using (23). Finally,  $K_p$  is obtained by solving (7).

### 3.3. Controller optimal design process based on DE algorithm

As shown in [34,35], DE algorithm can be applied to search for the optimal controller parameters in control system design. In this paper, the DE algorithm is used for the optimal design of the  $PI^\lambda D^\mu$  controller.

#### 3.3.1. Initialization

The algorithm starts with the initialization of the population.  $N$  individuals are randomly generated. As a potential solution to the optimization problem, each individual contains four parameters: the gain crossover frequency  $\omega_c$ , the phase margin  $\varphi_m$ , the controller orders  $\lambda$  and  $\mu$ , as shown in (33),

$$X_{i,G} = (\omega_c, \varphi_m, \lambda, \mu), \quad (33)$$

where  $i$  represents the index of the individual,  $i = 1, 2, \dots, N$ ,  $G$  is the evolution generations count,  $G = 1, 2, \dots, G_m$ ,  $G_m$  is the maximum generation number.

Then the controller parameters are calculated according to the procedure mentioned in Section 3.2. All individuals are checked by the boundary conditions (6) and (8). Only those satisfying the boundary conditions are selected into the initialized population.

#### 3.3.2. Mutation

When the population initialization is finished, the mutation and crossover operations are implemented. Some individuals are selected to be the target individuals according to a certain probability (mutation rate  $P_m$ ). The mutation is implemented by disturbing each target individual using the difference vector between different individuals. Then a mutated individual corresponding to the target individual in the current generation is generated, as shown in (34),

$$V_{i,G} = X_{i,G} + F \cdot (X_{r_1,G} - X_{r_2,G}), \quad (34)$$

where  $V_{i,G}$  is the mutated individual,  $X_{i,G}$  is the target individual,  $X_{r_1,G}$  and  $X_{r_2,G}$  are different individuals,  $F$  is a scale factor.

#### 3.3.3. Crossover

The algorithm performs the crossover operation using binary crossover scheme. Each mutated individual exchanges its components with the corresponding target individual and then a trial individual is generated. The trial individual should also be checked by the boundary conditions. If the boundary conditions are not satisfied, the trial individual will be generated again until the boundary conditions are satisfied.

#### 3.3.4. Selection

The step response simulations are implemented after the crossover operation and then the ITAE of the trial and target individuals are calculated. The simulation time  $T$  is 10 s, with the sampling time  $\Delta t$  to be 0.0001 s. The ITAE of each individual is calculated using (35),

$$J = \sum_{k=1}^{\frac{T}{\Delta t}} t[k] |e[k]| \Delta t. \quad (35)$$

In order to obtain the controller that minimizes the ITAE, the fitness function of each individual is set as follows,

$$F = \frac{1}{J}. \quad (36)$$

The selection is implemented by comparing the trial individual with the corresponding target individual according to their fitness values. The individual with larger fitness will be selected into the next population, while that with smaller fitness will be abandoned. When the optimization process is finished, the individual with the largest fitness will be selected to be the optimal individual.

### 3.3.5. Improvement to the optimization process

In order to avoid too high overshoot of the step response, a penalty operation is implemented on the fitness of the individuals whose overshoot peaks are larger than a selected upper bound  $\sigma_b$ . If the overshoot peak is higher than  $\sigma_b$ , the fitness of the individual will be set as 0. Meanwhile, the power consumption of the controller output should also be limited to an upper bound  $J_b$ , as shown in (37),

$$\int_0^{\infty} |i_{qr}(t)| dt \leq J_b, \quad (37)$$

where  $i_{qr}(t)$  is the output of the speed controller  $C_v(s)$  in Fig. 1. If the power consumption is larger than  $J_b$ , the fitness of the individual will also be set to be 0.

In order to avoid the convergence to the local optimum during the optimization process, an improvement is implemented on the mutation rate  $P_m$  in this paper, as described by (38) [34],

$$P_m = P_0 \cdot 2^\lambda, \quad (38)$$

where

$$\lambda = e^{1 - \frac{G_m}{G_m - G + 1}}, \quad (39)$$

where  $P_0$  is the initial mutation rate,  $G$  is the current generation count and  $G_m$  is the maximum generation. The improved mutation rate is close to  $2P_0$  at the beginning of the optimization process, allowing the maintenance of the diversity of individuals, helping to avoid the population converging to some local optimums. In the later period of the optimization process, the mutation rate is close to  $P_0$ , reducing the risk of the optimal individual being damaged by mutation.

In order to present the procedure of this optimal controller design and tuning course based on DE algorithm clearly, a design flow chart diagram is shown in Fig. 5.

### 3.4. $PI^{\lambda}D^{\mu}$ optimal design for the PMSM speed control system

The fractional order model parameters of the PMSM speed control system used in this paper are identified using an output-error method [32]. Besides, the gain  $K_{pi}$  and integral time  $T_i$  of the current controller  $C_i(s)$  in Fig. 1 are set to be 1 and 0.02, respectively. Therefore, the plant model of PMSM speed control system can be described by (40),

$$G(s) = \frac{47979.2573}{s^{2.9544} + 127.38 s^{2.0463} + 9995.678 s^{1.0463}}. \quad (40)$$

In order to ensure the system stability, the lower bounds of the phase margin and gain margin are set as  $60^\circ$  and 15 dB, respectively. The range of the gain crossover frequency is set as 1 to 100 rad/s. The upper bound of the peak overshoot  $\sigma_b$  is 12% and that of the power consumption  $J_b$  is 0.16. The ranges of the controller fractional orders  $\lambda$  and  $\mu$  are both 0 to 2. The population size  $N$  is set as 50 and the maximum generation  $G_m$  is 300. The initial mutation rate  $P_0$  is 0.1. The ranges of the parameters in each individual in the optimal algorithm are shown in Table 1.

Using the optimal algorithm mentioned in Section 3.4, the optimal fractional order  $PI^{\lambda}D^{\mu}$  controller can be designed as shown in (41),

$$C(s) = 8.281 \left( 1 + \frac{3.5062}{s^{0.8371}} + 0.0229 s^{0.941} \right). \quad (41)$$

The average fitness of all individuals and the best fitness in each generation are shown in Fig. 6, which shows that both the average fitness and the best fitness have converged to the steady values at the end of the optimization process.

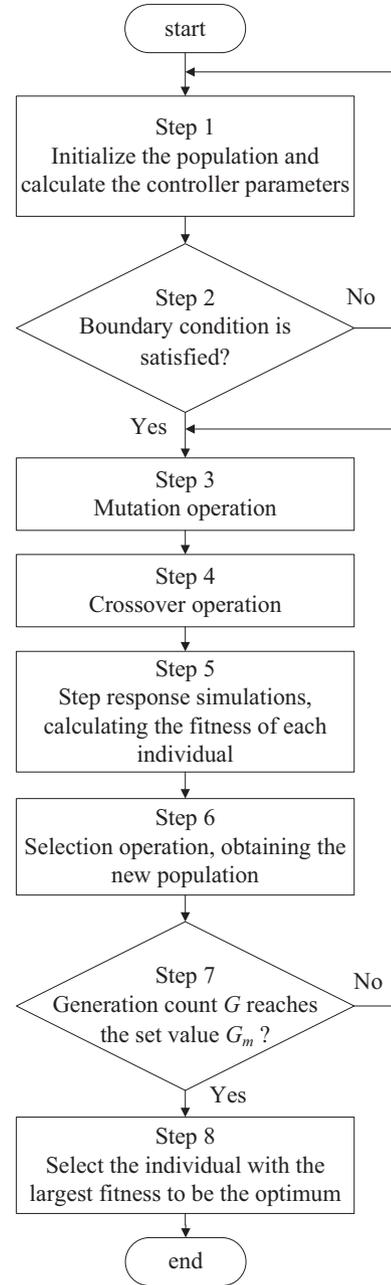


Fig. 5. The main procedure of the optimal controller design algorithm.

Table 1

The ranges of the parameters in each individual.

Parameters	$\omega_c$ (rad/s)	$\varphi_m$ (deg)	$\lambda$	$\mu$
minimum	1	60	0	0
maximum	100	180	2	2

The gain crossover frequency of the control system is  $\omega_c = 40.8$  rad/s, the phase margin is  $\varphi_m = 82.7^\circ$ . The phase crossover frequency is  $\omega_g = 1.04 \times 10^4$  rad/s and the gain margin is  $L_g = 82.8$  dB. Obviously, the boundary conditions of the phase margin and gain margin are both satisfied. The open-loop Bode plot is shown in Fig. 7.

It can be seen that the phase is flat around the gain crossover frequency, which satisfies the requirement for loop-gain robustness.

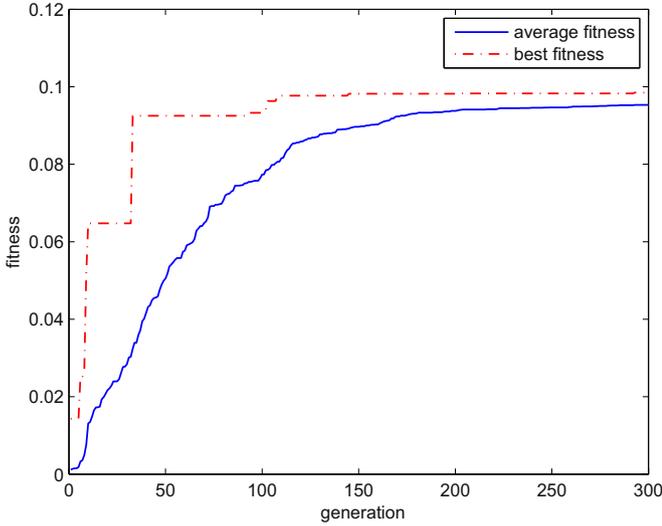


Fig. 6. The average fitness and the best fitness.

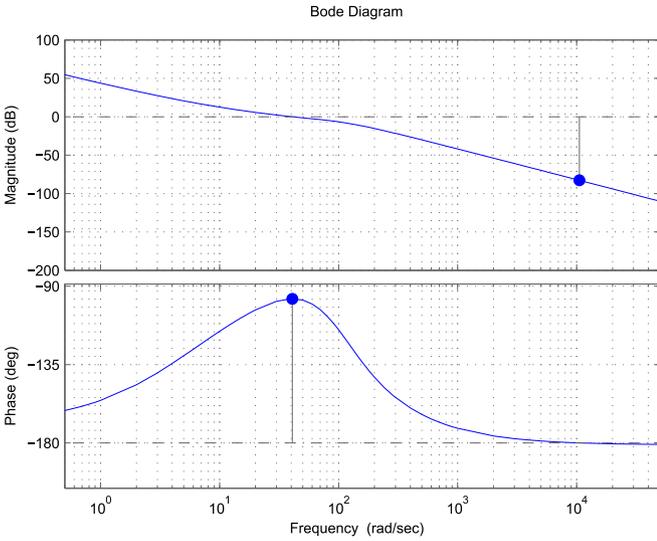


Fig. 7. Open-loop Bode plot of the control system using  $C(s)$ .

4. Simulation study

The optimal fractional order  $PI^\lambda D^\mu$  controller obtained using the design method proposed in Section 3 can achieve the desired stability, robustness to loop-gain variations, and optimal dynamic performance. In [29], a time-domain design method and a frequency-domain method for fractional order  $PI^\lambda D^\mu$  controller design are presented. In order to demonstrate the advantages of the tuning method proposed in this paper, two  $PI^\lambda D^\mu$  controllers are designed using the time-domain and frequency-domain methods presented in [29] in this section. PMSM speed-tracking simulations are implemented to compare the robustness and tracking performance of the controller obtained using the method proposed in this paper and those obtained using the methods presented in [29].

Besides, In order to verify the advantages of the fractional order  $PI^\lambda D^\mu$  controller, an optimal fractional order  $PI^\lambda$  controller as shown in (42) and an optimal integer order PID controller as (43) are designed using the similar design algorithm for comparison in this section.

$$C(s) = K_p \left( 1 + \frac{K_i}{s^\lambda} \right), \tag{42}$$

where,  $K_p$  and  $K_i$  are proportional and integral gains, respectively;  $\lambda \in (0, 2)$  is the fractional order.

$$C(s) = K_p \left( 1 + \frac{K_i}{s} + K_d s \right), \tag{43}$$

where,  $K_p$ ,  $K_i$  and  $K_d$  are proportional, integral and derivative gains, respectively.

PMSM speed-tracking simulations are implemented to test the tracking performance and the anti-load-disturbance performance of three control systems. The Oustaloup method [36,37] is used to approximate the fractional order operator  $s^\lambda$  in the simulation models.

4.1. Comparison with the time-domain method

The robustness specification is introduced as the constraint condition for the controller design in this paper, which ensures that the phase curve is flat at the gain crossover frequency. Therefore, when the system loop-gain has small variations, the overshoot of the step response will be almost constant. In contrast, the robustness specification is not introduced in the time-domain tuning method proposed in [29]. In order to guarantee the fair comparison, applying the ITAE index as the loss function, a  $PI^\lambda D^\mu$  controller is obtained using the time-domain design method proposed in [29], as shown in (44),

$$C_1(s) = 8.1909 \left( 1 + \frac{11.9094}{s^{1.1348}} + 0.081 s^{0.5514} \right). \tag{44}$$

The open-loop Bode plot is shown in Fig. 8. According to Fig. 8, the phase curve is not flat at the gain crossover frequency.

In order to verify the robustness of two controllers, the loop-gains of  $C(s)$  and  $C_1(s)$  are set to be 100%, 120% and 80% of their original values respectively to simulate the plant model uncertainties. PMSM speed-tracking simulations are implemented using two controllers to control the motor speed. The step response curves of two control systems are shown in Figs. 9 and 10, respectively.

Figs. 9 and 10 show that the overshoots of the response curves of the system using  $C(s)$  are close to each other, while those of the system using  $C_1(s)$  show obvious difference. Therefore, compared with the time-domain tuning method in [29], the tuning method

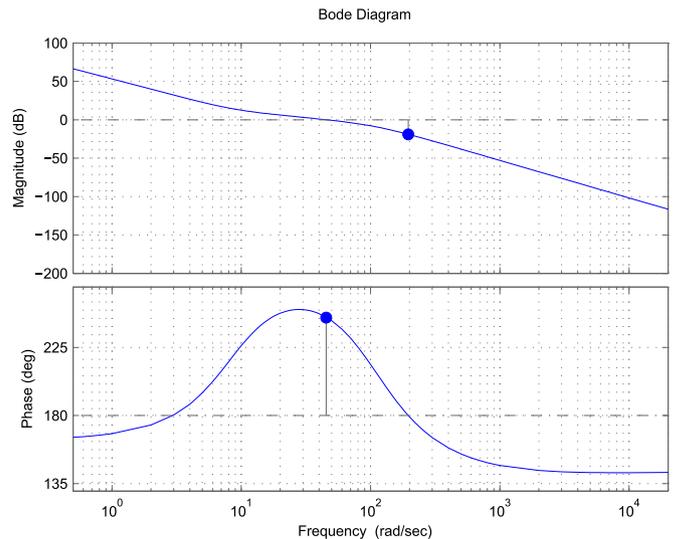


Fig. 8. Open-loop Bode plot of the control system using  $C_1(s)$ .

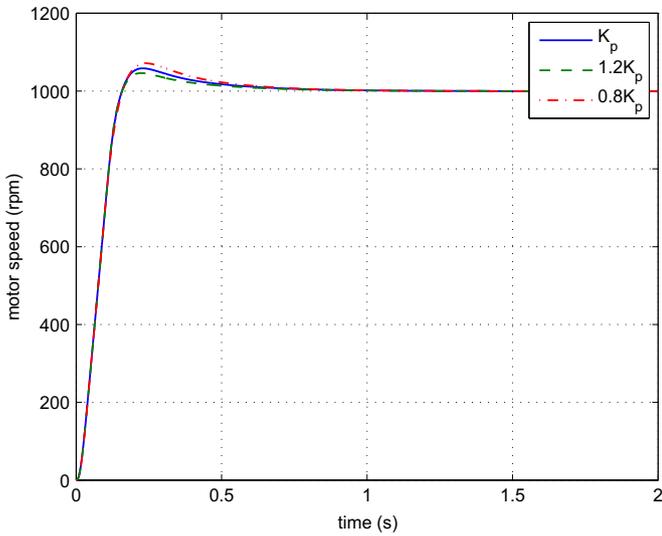


Fig. 9. Step response of the control system using  $C(s)$  with gain variations (simulation).

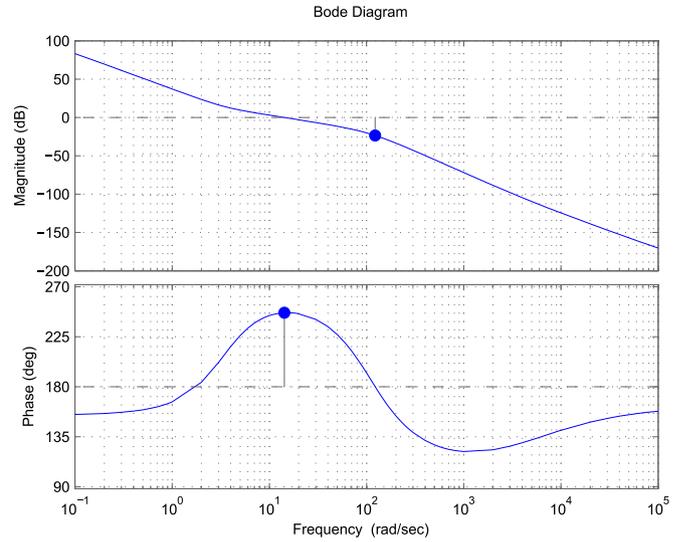


Fig. 11. Open-loop Bode plot of the control system using  $C_2(s)$ .

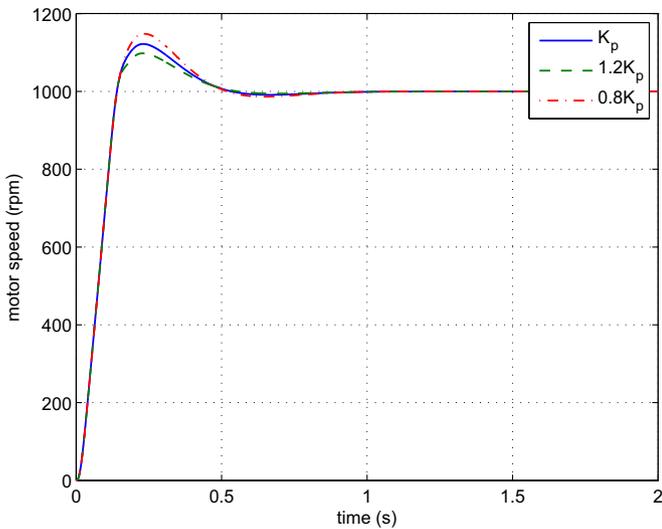


Fig. 10. Step response of the control system using  $C_1(s)$  with gain variations (simulation).

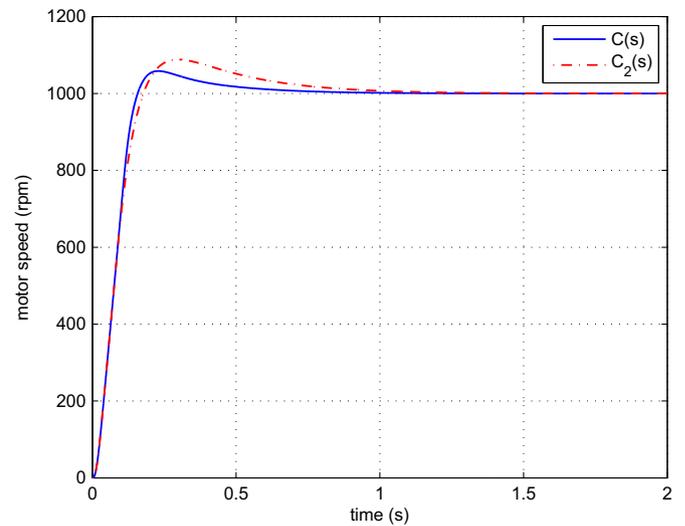


Fig. 12. Step response curves of two control systems using  $C(s)$  and  $C_2(s)$  (simulation).

proposed in this paper can ensure better robustness to gain variations.

4.2. Comparison with the frequency-domain method

The frequency-domain tuning method proposed in [29] searches for the control system with flat curve at the gain crossover frequency. Thus, the controller obtained using the frequency-domain can achieve good robustness to gain variations. However, without the consideration of the dynamic performance, this method cannot ensure the system to achieve good dynamic performance. In order to guarantee the fair comparison, a  $PI^2D^\mu$  controller is obtained using the frequency-domain design method proposed in [29], as shown in (45),

$$C_2(s) = 3.5276 \left( 1 + \frac{4.5108}{s^{1.2383}} + 0.0016 s^{0.7627} \right). \tag{45}$$

The open-loop Bode plot is shown in Fig. 11, which shows that the phase curve is flat at the gain crossover frequency. Therefore, the control system can achieve good robustness to gain variations.

PMSM speed step response simulation is implemented using  $C(s)$  and  $C_2(s)$  to control the motor speed. The response curves of

Table 2

Step response performance indexes of two control systems using  $C(s)$  and  $C_2(s)$  (simulation).

Controller	Rising time(s)	Settling time(s)	Overshoot(%)	$J_{ITAE}$
$C(s)$	0.157	0.471	5.84	10.152
$C_2(s)$	0.177	0.742	8.83	21.145

two control systems are shown in Fig. 12. The step response performance indexes of two control systems are shown in Table 2.

According to Fig. 12 and Table 2, compared with the response curve of the system using  $C_2(s)$ , the response curve of the system using  $C(s)$  has shorter rising time and settling time, smaller overshoot and ITAE. Therefore, the tuning method proposed in this paper can ensure the control system to achieve better dynamic performance.

4.3. Dynamic performance study

Applying the same constraint condition in (10), boundary conditions in (6) and (8), objective function  $J_{ITAE}$  in (11), fitness

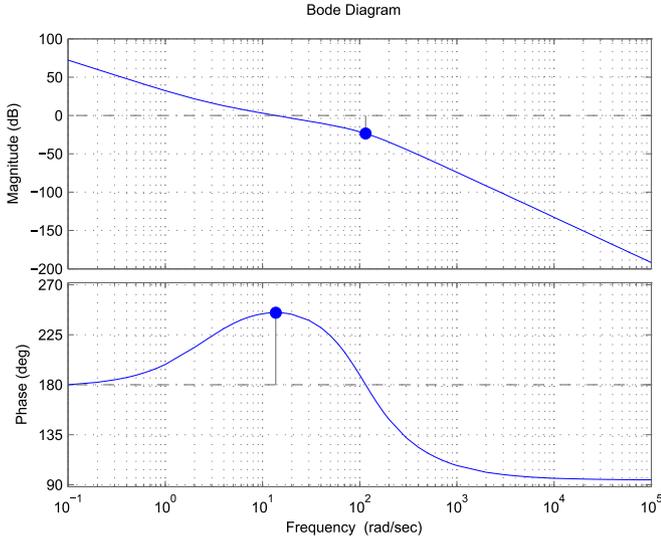


Fig. 13. Open-loop Bode plot of the control system using  $C_3(s)$ .

function  $F$  in (33), and the algorithm parameter settings of  $\omega_c$ ,  $\varphi_m$ ,  $L_g$ ,  $\sigma_b$ ,  $J_b$ ,  $\lambda$ ,  $N$  and  $P_0$ , DE algorithm [34,35] is applied to search for the optimal  $PI^\lambda$  controller parameters. As a potential solution to the optimization problem, each individual contains two parameters: the gain crossover frequency  $\omega_c$  and the integral order  $\lambda$ . Then  $K_I$  is calculated by solving (10) and  $K_p$  is calculated by solving (7). All the individuals are checked by the boundary conditions (6) and (8). Only those satisfying the boundary conditions are selected into the initialized population. Following the procedure shown in Fig. 5, the optimal fractional order  $PI^\lambda$  controller can be obtained as shown in (46),

$$C_3(s) = 3.1514 \left( 1 + \frac{2.5205}{s^{0.9802}} \right). \quad (46)$$

The gain crossover frequency of the control system is  $\omega_c = 13.7$  rad/s, the phase margin is  $\varphi_m = 64.8^\circ$ , the phase crossover frequency is  $\omega_g = 115$  rad/s and the gain margin is  $L_g = 23.6$  dB. The open-loop Bode plot is shown in Fig. 13, which shows that the phase curve of the system is flat at the gain crossover frequency, satisfying the robustness requirement.

Similarly, applying the constraint condition in (10), boundary conditions in (6) and (8), objective function  $J_{ITAE}$  in (11), fitness function  $F$  in (33), and the algorithm parameter settings of  $\omega_c$ ,  $\varphi_m$ ,  $L_g$ ,  $\sigma_b$ ,  $J_b$ ,  $N$  and  $P_0$ , DE algorithm [34,35] is applied to search for the optimal PID controller parameters as well. Each individual also contains two parameters: the gain crossover frequency  $\omega_c$  and the phase margin  $\varphi_m$ . Following the procedure shown in Fig. 5, an optimal integer order PID controller is designed as shown in (45),

$$C_4(s) = 8.3788 \left( 1 + \frac{2.6953}{s} + 0.0153 s \right). \quad (47)$$

The gain crossover frequency of the control system is  $\omega_c = 37.1$  rad/s, the phase margin is  $\varphi_m = 83.7^\circ$  and the gain margin is infinity. The open-loop Bode plot is shown in Fig. 14, which shows that the phase curve of the system is flat at the gain crossover frequency, satisfying the robustness requirement.

#### 4.3.1. Tracking performance study

PMSM speed step response simulation is implemented, using  $C(s)$ ,  $C_3(s)$  and  $C_4(s)$  to control the motor speed. The response curves of three control systems are shown in Fig. 15. The step response performance indexes of three control systems are shown in Table 3.

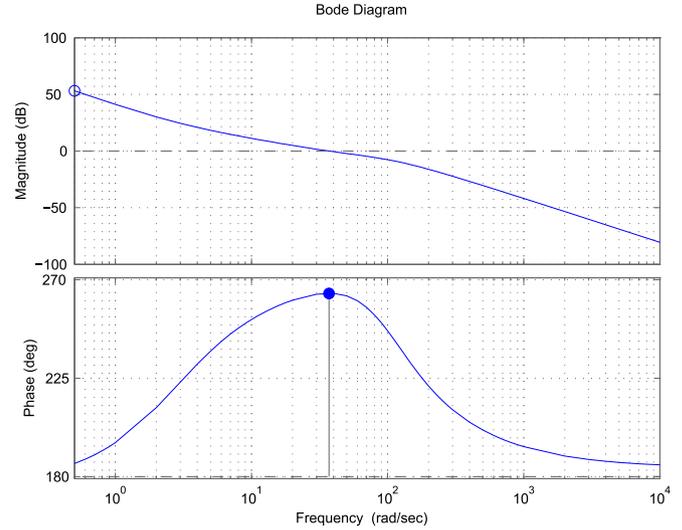


Fig. 14. Open-loop Bode plot of the control system using  $C_4(s)$ .

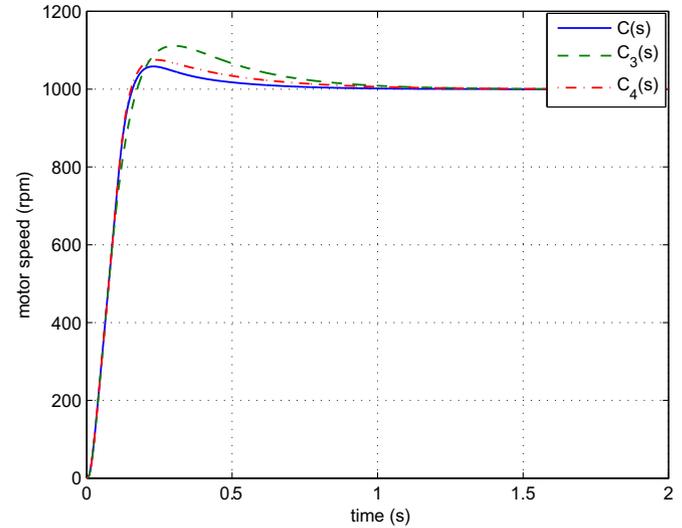


Fig. 15. Step response curves of the control systems using  $C(s)$ ,  $C_3(s)$  and  $C_4(s)$  (simulation).

Table 3

Step response performance indexes of the control systems using  $C(s)$ ,  $C_3(s)$  and  $C_4(s)$  (simulation).

Controller	Rising time(s)	Settling time(s)	Overshoot(%)	$J_{ITAE}$
$C(s)$	0.157	0.471	5.84	10.152
$C_3(s)$	0.173	0.802	11.15	25.904
$C_4(s)$	0.153	0.657	7.53	17.274

Fig. 15 and Table 3 show that the rising time of the response curve of the system using  $C(s)$  is close to that of the system using  $C_4(s)$ , which is shorter than that of the system using  $C_3(s)$ . Compared with those of the systems using  $C_3(s)$  and  $C_4(s)$ , the response curve of the system using  $C(s)$  has the shortest settling time, smallest overshoot and ITAE. Therefore, the control system using  $C(s)$  achieves the optimal tracking performance.

#### 4.3.2. Anti-load-disturbance performance study

In order to test the anti-load-disturbance performance of the control systems, load disturbance is injected when the motor speed becomes stable. The disturbance responses of three control

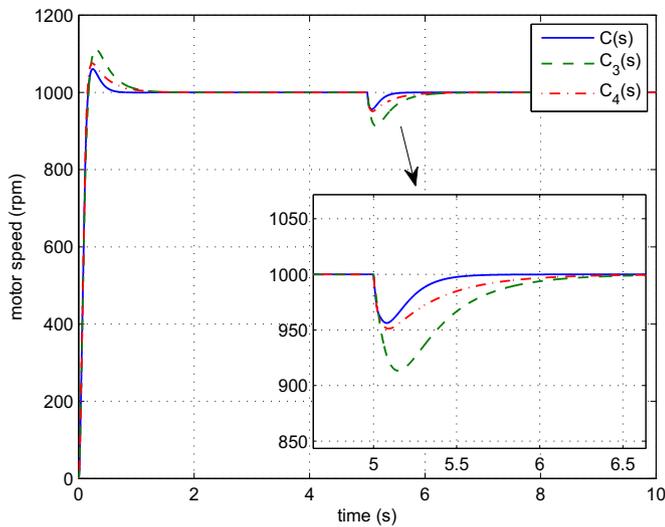


Fig. 16. Step responses with load disturbance of the control systems using  $C(s)$ ,  $C_3(s)$  and  $C_4(s)$  (simulation).

**Table 4**  
Anti-load-disturbance performance indexes of the control systems using  $C(s)$ ,  $C_3(s)$  and  $C_4(s)$  (simulation).

Controller	Recovery time(s)	Dynamic speed drop(%)
$C(s)$	0.24	4.37
$C_3(s)$	0.65	8.66
$C_4(s)$	0.41	4.88

systems are shown in Fig. 16. The anti-load-disturbance performance indexes of three control systems are shown in Table 4. In Table 4, the recovery time is the interval from the point where the speed drops out of the steady state range to the point where it recovers back to the range. The steady state range is defined as the range (98%, 102%) of the target steady state value.

Fig. 16 and Table 4 show that, compared with those of the system using  $C_3(s)$  and  $C_4(s)$ , the response curve of the system using  $C(s)$  has the shortest recovery time and smallest dynamic speed drop.

According to two groups of simulation results, the step response and anti-load-disturbance performances of the fractional order  $PI^{\lambda}D^{\mu}$  controller are better than those of the fractional order  $PI^{\lambda}$  controller and the integer order PID controller.

### 5. Experimental study

In this section, the robustness and dynamic performance of the proposed fractional order  $PI^{\lambda}D^{\mu}$  controller is studied by real PMSM speed control experiments. The fractional order operator  $s^{\lambda}$  is implemented by the impulse invariant discretization method [38] in time-domain.

#### 5.1. Experimental platform

The PMSM speed control platform is shown in Fig. 17. The motor used is Sanyo-P10B18200BXS PMSM. The servo drive is an embedded control system based on TI-DSP F2812 with the software development platform Code Composer Studio on PC. The DC generator is  $Z_2 = 32$  generator, which is coupled with the PMSM for load adjustment. The load disturbance is injected by closing the switch connecting the generator output terminal and the resistance box.

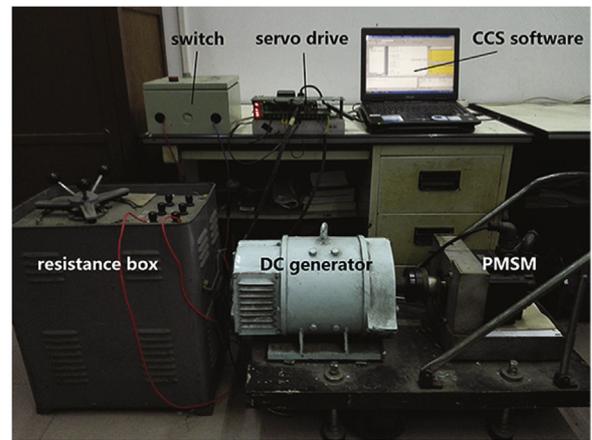


Fig. 17. PMSM speed control platform.

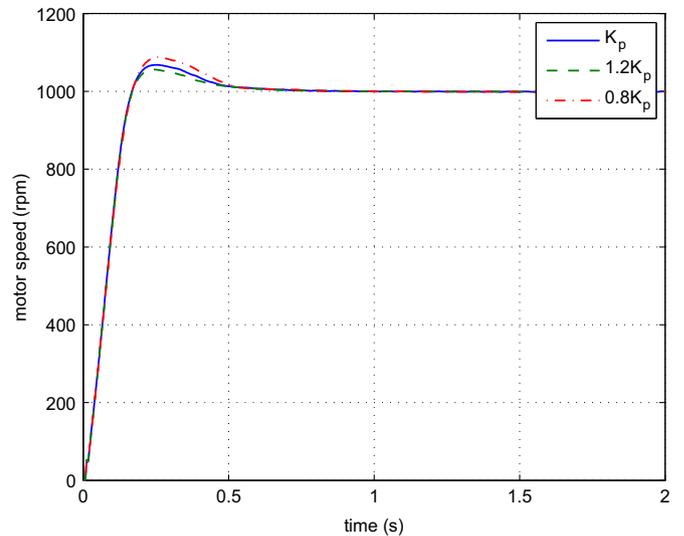


Fig. 18. Step response of the control system using  $C(s)$  with gain variations (experiment).

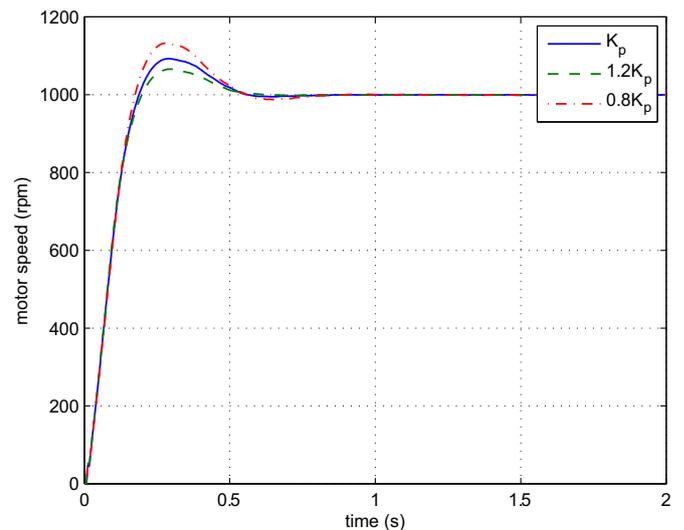


Fig. 19. Step response of the control system using  $C_1(s)$  with gain variations (experiment).

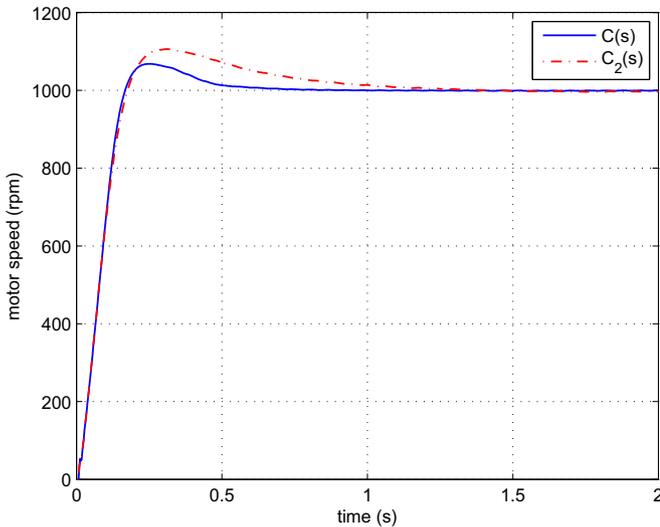


Fig. 20. Step response curves of two control systems using  $C(s)$  and  $C_2(s)$  (experiment).

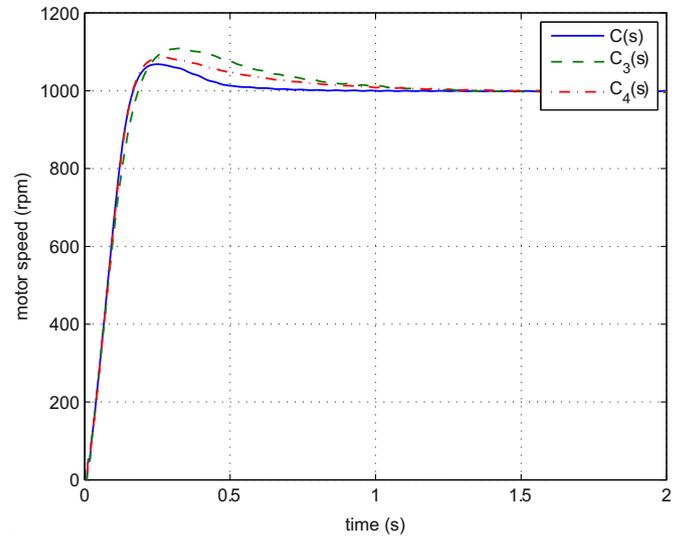


Fig. 21. Step response curves of the control systems using  $C(s)$ ,  $C_3(s)$  and  $C_4(s)$  (experiment).

5.2. Comparison with the time-domain method

PMSM speed control experiment is implemented, using  $C(s)$  and  $C_1(s)$  to control the motor speed. The step response curves of two control systems are shown in Figs. 18 and 19, respectively.

Similar to the simulation results, Figs. 18 and 19 show that the overshoots of the response curves of the system using  $C(s)$  are close to each other, while those of the system using  $C_1(s)$  show obvious difference. Therefore, compared with the time-domain tuning method in [29], the tuning method proposed in this paper can ensure better robustness to gain variations.

5.3. Comparison with the frequency-domain method

PMSM speed Step response experiment is implemented using  $C(s)$  and  $C_2(s)$  to control the motor speed. The response curves of two control systems are shown in Fig. 20. The step response performance indexes of two control systems are shown in Table 5.

According to Fig. 20 and Table 5, compared with the response curve of the system using  $C_2(s)$ , the response curve of the system using  $C(s)$  has shorter rising time and settling time, smaller overshoot and ITAE. Therefore, the tuning method proposed in this paper can ensure the control system to achieve better dynamic performance.

5.4. Tracking performance study

PMSM speed step response experiment is implemented, using  $C(s)$ ,  $C_3(s)$  and  $C_4(s)$  to control the motor speed. The response curves of three control systems are shown in Fig. 21. The step tracking performance indexes of three control systems are shown in Table 6.

Similar to the simulation results, Fig. 21 and Table 6 show that the rising time of the response curve of the system using  $C(s)$  is

Table 5 Step response performance indexes of two control systems using  $C(s)$  and  $C_2(s)$  (experiment).

Controller	Rising time(s)	Settling time(s)	Overshoot(%)	$J_{ITAE}$
$C(s)$	0.167	0.457	6.8	11.386
$C_2(s)$	0.176	0.889	10.61	33.392

Table 6

Step response performance indexes of the control systems using  $C(s)$ ,  $C_3(s)$  and  $C_4(s)$  (experiment).

Controller	Rising time(s)	Settling time(s)	Overshoot(%)	$J_{ITAE}$
$C(s)$	0.167	0.457	6.8	11.386
$C_3(s)$	0.173	0.864	10.91	33.907
$C_4(s)$	0.166	0.76	8.59	23.609

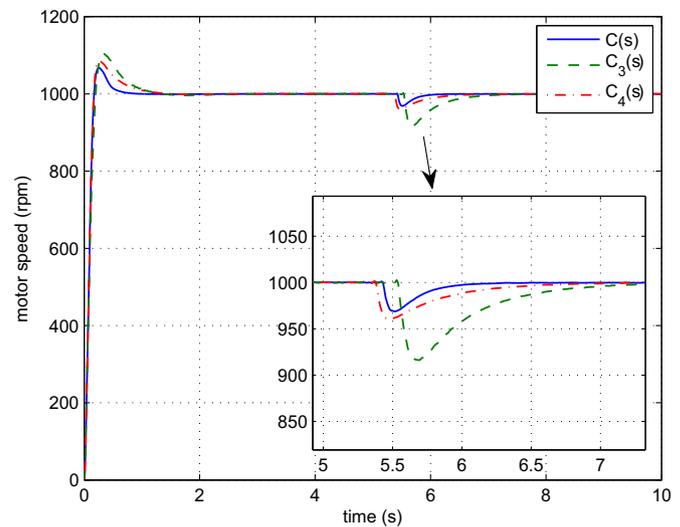


Fig. 22. Step responses with load disturbance of the control systems using  $C(s)$ ,  $C_3(s)$  and  $C_4(s)$  (experiment).

Table 7

Anti-load-disturbance performance indexes of the control systems using  $C(s)$ ,  $C_3(s)$  and  $C_4(s)$  (experiment).

Controller	Recovery time(s)	Dynamic speed drop(%)
$C(s)$	0.21	3.11
$C_3(s)$	0.86	8.39
$C_4(s)$	0.41	3.95

close to that of the system using  $C_4(s)$ , which is shorter than that of the system using  $C_3(s)$ . Compared with those of the systems using  $C_3(s)$  and  $C_4(s)$ , the response curve of the system using  $C(s)$  has the shortest settling time, smallest overshoot and ITAE. Therefore, the control system using  $C(s)$  achieves the optimal tracking performance.

### 5.5. Anti-load-disturbance performance study

In order to test the anti-load-disturbance ability of the control systems, load disturbance is injected when the motor speed becomes stable. The disturbance responses of three control systems are shown in Fig. 22. The anti-load-disturbance performance indexes of three control systems are shown in Table 7.

Similar to the simulation results, Fig. 16 and Table 4 show that, compared with those of the system using  $C_3(s)$  and  $C_4(s)$ , the response curve of the system using  $C(s)$  has the shortest recovery time and smallest dynamic speed drop.

According to two groups of experimental results, the step response and anti-load-disturbance performances of the fractional order  $PI^{\lambda}D^{\mu}$  controller are better than those of the fractional order  $PI^{\lambda}$  controller and the integer order PID controller.

## 6. Conclusion

A fractional order  $PI^{\lambda}D^{\mu}$  controller design strategy satisfying time and frequency domain specifications for a PMSM servo system is proposed in this paper. Taking the robustness specification as the constraint condition, the phase margin specification and gain margin specification as the boundary conditions, the ITAE as the objective function, the DE algorithm is applied to search the optimal controller parameters. PMSM speed control simulations and experiments are demonstrated, showing the significant advantages of the proposed  $PI^{\lambda}D^{\mu}$  controller by the comparisons with the controllers obtained using the time-domain and frequency-domain methods, the  $PI^{\lambda}$  controller and the traditional PID controller. Simulation and experimental results show that the proposed  $PI^{\lambda}D^{\mu}$  controller can satisfy the specifications in both frequency-domain and time-domain simultaneously and achieve optimal tracking performance.

## References

- [1] Benchellal A, Poinot T, Trigeassou JC. Approximation and identification of diffusive interfaces by fractional models. *Signal Process* 2006;86(10):2712–27.
- [2] Ngueteu GSM, Wofo P. Dynamics and synchronization analysis of coupled fractional-order nonlinear electromechanical systems. *Mech Res Commun* 2012;46:20–5.
- [3] Narang A, Shah SL, Chen T. Continuous-time model identification of fractional-order models with time delays. *IET Control Theory A* 2012;5(7):900–12.
- [4] Poinot T, Trigeassou JC. Identification of fractional systems using an output-error technique. *Nonlinear Dynam* 2004;38(1–4):133–54.
- [5] Luo Y, Chen YQ, Ahn HS, Pi YG. Fractional order robust control for cogging effect compensation in pmsm position servo systems: stability analysis and experiments. *Control Eng Pract* 2010;18(9):1022–36.
- [6] Bigdeli N. The design of a non-minimal state space fractional-order predictive functional controller for fractional systems of arbitrary order. *J Process Control* 2015;29:45–56.
- [7] Monje CA, Vinagre BM, Feliu V, Chen YQ. Tuning and auto-tuning of fractional order controllers for industry applications. *Control Eng Pract* 2008;16(7):798–812.
- [8] Valerio D, d Costa JS. Tuning of fractional PID controllers with ziegler-nichols-type rules. *Signal Process* 2006;86(10):2771–84.
- [9] Wang DJ, Li W, Guo ML. Tuning of  $PI^{\lambda}D^{\mu}$  controllers based on sensitivity constraint. *J Process Control* 2013;23(6):861–7.
- [10] Hartley TT, Lorenzo CF. Fractional-order system identification based on continuous order-distributions. *Signal Process* 2003;83(11):2287–300.
- [11] Gabano JD, Poinot T. Fractional modelling and identification of thermal systems. *Signal Process* 2011;91(3):531–41.
- [12] Wang DJ. Further results on the synthesis of PID controllers. *IEEE Trans Autom Control* 2007;52(6):1127–32.
- [13] Silva G, Datta A, Bhattacharyya S. New results on the synthesis of PID controller. *IEEE Trans Autom Control* 2002;47(2):241–52.
- [14] Astrom K, Hagglund T. *Advanced PID control*. Res Triangle Park: ISA- Instrum Syst Autom Soc 2005.
- [15] Astrom K, Hagglund T. *PID controllers: theory, design and tuning*. Instrum Soc Am, Res Triangle Park 1995.
- [16] Podlubny I. Fractional-order systems and  $PI^{\lambda}D^{\mu}$  controllers. *IEEE Trans Autom Control* 1999;44(1):208–14.
- [17] Biswas A, Das S, Abraham A, Dasgupta D. Design of fractional-order  $PI^{\lambda}D^{\mu}$  controllers with an improved differential evolution. *Eng Appl Artif Intel* 2009;22(2):343–50.
- [18] Hamamci SE, Tan N. Design of PI controllers for achieving time and frequency domain specifications simultaneously. *ISA Trans* 2006;45(4):529–43.
- [19] Soylemez MT, Munro N, Baki H. Fast calculation of stabilizing PID controllers. *Automatica* 2003;39(1):121–6.
- [20] Caponetto R, Dongola G, Fortuna L, Gallo A. New results on the synthesis of FO-PID controllers. *Commun Nonlinear Sci* 2010;15(4):997–1007.
- [21] Luo Y, Chen YQ. Stabilizing and robust FOPI controller synthesis for first order plus time delay systems. *Automatica* 2011;48(9):2040–5.
- [22] Hamamci S. An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers. *IEEE Trans Autom Control* 2007;52(10):1964–9.
- [23] Podlubny I. *Fractional Differential Equations*. New York: Academic Press; 1999.
- [24] Monje C, Chen Y, Vinagre B, Xue D, Feliu-Batlle V. *Fractional-order systems and controls: fundamentals and applications*. London: Springer; 2010.
- [25] Luo Y, Chen Y. *Fractional Order Motion Controls*. New York: Wiley; 2013.
- [26] Luo Y, Chen YQ. Fractional order [proportional derivative] controller for a class of fractional order systems. *Automatica* 2009;45(10):2446–50.
- [27] Zamani M, Karimi-Ghartemani M, Sadati N, Parniani M. Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Eng Pract* 2009;17(12):1380–7.
- [28] Lee CH, Chang FK. Fractional-order PID controller optimization via improved electromagnetism-like algorithm. *Expert Syst Appl* 2010;37(12):8871–8.
- [29] Martín F, Monje C, Moreno L, Balaguer C. De-based tuning of  $PI^{\lambda}D^{\mu}$  controllers. *ISA T* 2015;59:398–407.
- [30] Storn R, Price K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 1997;11:341–59.
- [31] Xu F, Li DH, Xue YL. Comparing and optimum seeking of PID tuning methods base on ITAE index. *Proc Chin Soc Electr Eng* 2003;23(8):206–10.
- [32] Zheng WJ, Luo Y, Chen YQ, Pi YG. Fractional-order modeling of permanent magnet synchronous motor speed servo system. *J Vib Control* 2016;22(8):2255–80.
- [33] Chen BS. *Electricity Pull Automation Control System*. 4th ed.. Beijing: China Machine Press; 2003.
- [34] Zhou YP, Gu XS. Development of differential evolution algorithm. *Control Instrum Chem Ind* 2007;34(3):6–10.
- [35] Liu B, Wang L, Jin YH. Advance in differential evolution. *Control Decis* 2007;22(7):721–9.
- [36] Oustaloup A. *la Dérivation Non Entière: Théorie, Synthèse Et Applications*. Paris: Editions Hermès; 1995.
- [37] Oustaloup A, François L, Benoît M, Florence M. Frequency-band complex noninteger differentiator: characterization and synthesis. *IEEE Trans Circuits-I* 2000;47(1):25–9.
- [38] Chen YQ. Impulse response invariant discretization of fractional order integrators/differentiators. *Filter Design and Analysis, MATLAB Central* 2008; (<http://www.mathworks.com/matlabcentral/fileexchange/21342-impulse-response-invariant-discretization-of-fractional-order-integrators-differentiators>).