Alexandria Engineering Journal (2020) 59, 3247–3254

HOSTED BY

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

A variable-order fractional proportional-integral controller and its application to a permanent magnet synchronous motor

Liping Chen [a], Gang Chen [a], Penghua Li [b], António M. Lopes [c], J.A. Tenreiro Machado [d], Shuiqing Xu [a]

[a] School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, China
[b] Automotive Electronics Engineering Research Center, College of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[c] UISPA–LAETA/INEGI, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
[d] Institute of Engineering, Polytechnic of Porto, Department of Electrical Engineering, R. Dr. António Bernardino de Almeida, 431, 4249-015 Porto, Portugal

Received 29 April 2020; revised 30 July 2020; accepted 28 August 2020
Available online 14 September 2020

KEYWORDS

Fractional-order PI controller;
Permanent magnet synchronous motor;
Stability

Abstract The time-domain approximation of the Grünwald–Letnikov fractional derivative is intuitive and widely adopted in the design of fractional-order proportional-integral (FOPI) controllers. To solve the accuracy reduction caused by truncating the series, an optimized discrete FOPI is presented. The effectiveness of the new FOPI is highlighted and compared with the one exhibited by a controller implemented based on the Oustaloup method. Furthermore, to improve the performance of the FOPI, a variable-order fractional proportional-integral (VFPI) controller is proposed. The response of the VFPI is verified in the control of a permanent magnet synchronous motor. Simulation and experimental results show the superior performance of the VFPI.

1. Introduction

Fractional calculus has been widely used in the area of systems' modeling and control due to its advantages over the classical integer calculus [1–8]. The fractional-order proportional-integral (FOPI) controller is a generalization to non-integer orders of the integer proportional-integral (IOPI) controller. The FOPI has three parameters, yielding the IOPI when the order of the integral term is equal to one. The extra degree of freedom allows the FOPI to achieve superior control results than the IOPI and leads to better dynamic response and higher robustness [9,10]. Therefore, as the controlled systems become more complex and the requirements for control performance increase, the FOPI emerges as a relevant option, as demonstrated by several engineering applications [11–15].

Several numerical methods were proposed in the area of fractional calculus [16–19]. For fractional-order (FO) con-

Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2020.08.046

troller design, the time-domain [20], s-domain [21,22] and z-domain [23,24] numerical approximations are often adopted. A discrete FOPI based on the time-domain approximation was discussed in [25] by discretizing and approximating the Grünwald-Letnikov (GL) fractional derivative. This controller is easy to implement, but its accuracy is limited due to the series truncation. The s-domain approximation implies the replacement of the FO integral and derivative by integer operators with frequency characteristics that approximate those of the FO operators. Common approximations in the s-domain consist of the Oustaloup and the continued fraction expansion (CFE) methods. In what concerns the z-domain approximation we need two steps. The first is to discretize the FO operators using the Tustin or Euler's formulas. The second is to approximate the expression by a rational one, using the CFE [26] or the Taylor expansion [27,28].

Adopting accurate models and optimizing the control algorithms are effective methods to improve the performance of complex systems. However, since we often have limited knowledge about the system dynamics, the effort on the optimization of the control algorithm is more productive. In recent years, the combination of different control methods has become a promising approach. For example, FO sliding mode control [29,30] and adaptive fuzzy FO sliding mode control [31,32] have achieved good results. However, the practical implementation of any FO controller requires accurate numerical methods. Indeed, approximating the FO integrator in the time domain using the GL approximation may induce oscillations due to the series truncation. In this paper, a method to solve the finite memory length issue is proposed, leading to an improved FOPI. Moreover, a variable-order fractional proportional integral (VFPI) controller is proposed by combining the FOPI with variable order. Simulation and experimental results show that the VFPI yields superior performance when applied in the control of a permanent magnet synchronous motor (PMSM).

The rest of this paper is organized as follows. Section 2 introduces preliminary concepts regarding the FO derivative and the PMSM model. Section 3 develops the improved FOPI and the VFPI controllers. Section 4 presents simulation and experimental results of the PMSM control. Finally, Section 5 outlines the main conclusions.

2. A brief outline about FO derivatives and the permanent magnet synchronous motor

In this section we introduce the GL FO derivative, the Oustaloup approximation method, and the mathematical model of the PMSM.

2.1. The GL derivative

The GL FO derivative of a causal function, $f(t)$, can be written as:

$$\begin{aligned} {}_aD_t^{\alpha} f(t) &= \lim_{h\to 0} \frac{1}{\Gamma(\alpha)h^{\alpha}} \sum_{j=0}^{[(t-\alpha)/h]} \frac{\Gamma(\alpha+j)}{\Gamma(j+1)} f(t-jh) \\ &= \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{[(t-\alpha)/h]} (-1)^j \times \binom{\alpha}{j} f(t-jh), \end{aligned} \tag{1}$$

where $\Gamma(\cdot)$ is the gamma function and

$$\begin{aligned} \binom{\alpha}{j} &= \frac{\alpha(\alpha-1)(\alpha-2)\cdots(\alpha-j+1)}{j!} \\ &= \frac{\alpha!}{j!(\alpha-j)!}. \end{aligned} \tag{2}$$

Several methods have been developed to approximate (1) by means of finite-order transfer functions. In the Laplace domain, the FO operators (derivative and integral) are, respectively:

$$L\{{}_0D_t^{\alpha} f(t)\} = s^{\alpha}F(s) - \sum_{k=0}^{n-1} s^k [{}_0D_t^{\alpha-k-1} f(t)|_{t=0}], \tag{3}$$

$$L\{{}_0D_t^{-\alpha} f(t)\} = s^{-\alpha}F(s), \tag{4}$$

where $L\{\cdot\}$ denotes the Laplace transform and $m-1 \leqslant \alpha < m, m \in \mathbb{N}$. For zero initial conditions, we have the transfer functions:

$$H(s) = s^{\alpha}, G(s) = 1/s^{\alpha}. \tag{5}$$

The Oustaloup method approximates s^{α} through a $(2N+1)$th-order rational transfer function, yielding:

$$H(s) = \left(\frac{w_u}{w_h}\right)^{\alpha} \prod_{k=-N}^{N} \frac{1+s/w_{k1}}{1+s/w_{k2}}, \tag{6}$$

where

$$w_u = (w_b w_h)^{1/2}, w_{k1} = w_b \left(\frac{w_h}{w_b}\right)^{\frac{k+N+1/2(1-\alpha)}{2N+1}},$$

$$w_{k2} = w_b \left(\frac{w_h}{w_b}\right)^{\frac{k+N+1/2(1+\alpha)}{2N+1}}.$$

Obviously, the accuracy of the approximation within the bandwidth $[w_b, w_h]$ increases with N.

2.2. Mathematical model of a PMSM

The PMSM is an electrical motor excited by permanent magnets. Compared with three-phase asynchronous motors, the PMSM has lower loss, better power factor and higher efficiency. In this paper a PMSM is used as the controlled plant, and its model is briefly presented in the follow-up.

The mathematical model of a PMSM in the $d-q$ frame can be given by:

$$\begin{pmatrix} \dot{i}_d \\ \dot{i}_q \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} -\frac{R_s}{L_d} & \rho_n\omega & 0 \\ -\rho_n\omega & -\frac{R_s}{L_q} & -\frac{\rho_n\Psi_f}{L_q} \\ 0 & \frac{\rho_n\Psi_f}{J} & -\frac{B}{J} \end{pmatrix} \begin{pmatrix} i_d \\ i_q \\ \omega \end{pmatrix} + \begin{pmatrix} \frac{u_d}{L_d} \\ \frac{u_q}{L_q} \\ -\frac{T_L}{J} \end{pmatrix}, \tag{7}$$

where the model parameters are

u_d, u_q	d- and q-axis stator voltages;
i_d, i_q	d- and q-axis stator currents;
L_d, L_q	d- and q-axis inductors;
R_s	stator resistance;
ρ_n	number of pole pairs;
ω	angular velocity;
Ψ_f	rotor flux linkage;
T_L	load torque;
J	rotor inertia;
B	viscous friction coefficient.

The electromagnetic torque of the PMSM is given by:

$$T_e = \frac{3}{2}\rho_n[\Psi_f i_q - (L_q - L_d)i_d i_q], \tag{8}$$

In order to maximize T_e, the current $i_d = 0$, and (8) yields:

$$T_e = \frac{3}{2}\rho_n \Psi_f i_q, \tag{9}$$

Since for a motor the parameters ρ_n and Ψ_f are constant, the torque T_e can be controlled by means of i_q. This mode is called $i_d = 0$ control mode. After starting, and without any perturbation, the PMSM accelerates or decelerates until reaching stability. At this point, the electromagnetic and load torques, T_e and T_L, are equal.

In the field of industrial control, there are various physical constraints on both the control action and the state of the PMSM control system. For example, the current i_q is limited either by the PMSM or by the inverter. This paper aims to study the performance of the control algorithm. In order to reduce the interference of external factors, measures are taken to make the actual operating conditions of the PMSM below the rated ones.

3. Controller design

In this section, the time domain approximation method is used to design the FOPI. The reasons that lead the controller to cause oscillation are analyzed. Then, an improved discrete FOPI is proposed and its performance is discussed. Finally, based on the improved discrete FOPI, a VFPI is designed.

3.1. The problem of time domain approximation method

Expression (2) can be rewritten as:

$$\begin{aligned}\binom{\alpha}{j} &= \frac{\alpha!}{j!(\alpha-j)!} \\ &= \frac{\alpha-j+1}{j} \times \frac{\alpha!}{(j-1)!(\alpha-j+1)!} \\ &= \frac{\alpha-j+1}{j}\binom{\alpha}{j-1}.\end{aligned} \tag{10}$$

Let $q_{\alpha,j} = (-1)^j\binom{\alpha}{j}$ and $q_{\alpha,0} = 1$. From (10) we obtain

$$\begin{aligned}q_{\alpha,j} &= -\frac{\alpha-j+1}{j} \times (-1)^{j-1}\frac{\alpha!}{(j-1)!(\alpha-j+1)!} \\ &= \left(1 - \frac{1+\alpha}{j}\right)q_{\alpha,j-1}.\end{aligned} \tag{11}$$

Substituting (11) into (1) yields

$$\begin{aligned}{}_aD_t^\alpha f(t) &= \lim_{h\to 0}\frac{1}{h^\alpha}\sum_{j=0}^{[(t-\alpha)/h]}(-1)^j\binom{\alpha}{j}f(t-jh) \\ &= \lim_{h\to 0}\frac{1}{h^\alpha}\sum_{j=0}^{[(t-\alpha)/h]}q_{\alpha,j}f(t-jh).\end{aligned} \tag{12}$$

It can be seen from (12) that the length of the array q tends to infinite when the step size h approaches zero. Since the size of q must be limited, we set the memory length to L and approximate (12) by

$${}_aD_t^\alpha f(t) \approx \frac{1}{h^\alpha}\sum_{j=0}^{M}q_{\alpha,j}f(t-jh), \tag{13}$$

where $M = \min\left\{\left[\frac{t}{h}\right], \left[\frac{L}{h}\right]\right\}$.

Using expression (13), the control output $u(t)$ of the FOPI controller can be written as:

$$u(k) = K_p e(k) + K_i h^\lambda \sum_{j=0}^{M} q_{-\lambda,j}e(k-j), \tag{14}$$

where λ denotes the order of the integral component.

For most cases, when the system is stable, the output of the controller is not zero. In some cases, the controller and the system outputs are directly related. Suppose that, when a system reaches stability, the output of the controller is u_m, the sampling time is h_1, and the adjusting time is t_1. Suppose that $M = M_1$ in formula (14) and $M_1 > t_1/h_1$. When $t = 2M_1$, one can get from formula (14):

$$u(2M_1) = K_p e(2M_1) + K_i h^\lambda \sum_{j=M_1}^{2M_1} q_{-\lambda,j}e(k-j) = 0. \tag{15}$$

Since $u = u_m$ when the system is stable, then it is unstable when $t = 2M_1$. If $u(2M_1) = u_m$, then $\sum_{j=M_1}^{2M_1} e(k-j) > 0$. Therefore, the controller obtained by (14) can not achieve null static error.

As an illustrative example, let us consider the second-order system:

$$\ddot{y}(t) + 50\dot{y}(t) + 100y(t) = x(t), \tag{16}$$

where $x(t)$ is the input and $y(t)$ is the output.

The parameters of the FOPI are set to $K_p = 50, K_i = 500$ and $\lambda = 0.9135$. The sampling time is $h = 0.01$ s and the memory length is $M = 1000$.

Figs. 1 and 2 depict the system response to a unit step and the control action, respectively. We verify from Fig. 1 that the settling time is $t_s = 5$ s, and the system oscillates after 10 s. From Fig. 2 it can be seen that when the system is stable, the output of the controller is $u_m = 99.8$. After 10 s, the output of the controller drops rapidly and then oscillates.

For $5\text{ s} \leqslant t_{s1} \leqslant 10$ s, the system output is given by the following expression:

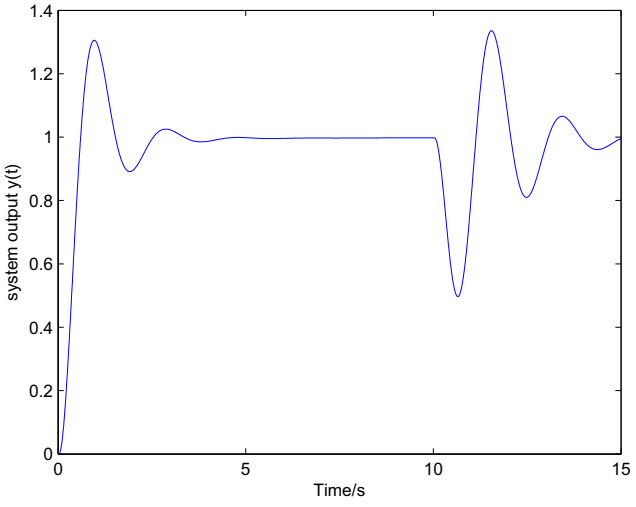

Fig. 1 Unit step response $y(t)$ of the closed-loop system.

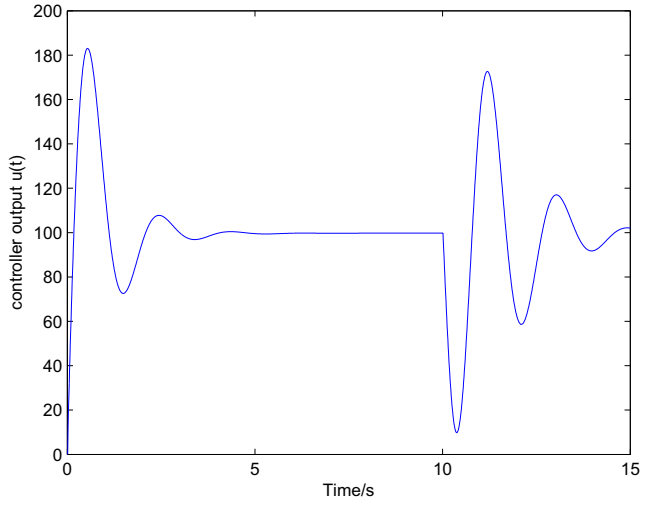

Fig. 2 The controller $u(t)$ output for the unit step response of the closed-loop system.

$$\begin{aligned} u(t_{s1}) &= K_p e(t_{s1}) + K_i h^{\lambda} \sum_{j=0}^{t_s/h} q_{-\lambda,j} e(k-j) \\ &\quad + K_i h^{\lambda} \sum_{j=t_s/h+1}^{t_{s1}/h} q_{-\lambda,j} e(k-j) \\ &= 0 + 99.8 + 0 = 99.8 = u_m, \end{aligned} \tag{17}$$

For $10 < t_{s2} \leqslant 10 + \triangle t$, where $\triangle t$ is a small time increment, the system is still stable and the output is given by:

$$u(t_{s2}) = K_p e(t_{s2}) + K_i h^{\lambda} \sum_{j=\triangle t/h+1}^{t_s/h} q_{-\lambda,j} e(k-j) + K_i h^{\lambda} \sum_{j=t_s/h+1}^{t_{s2}/h} q_{-\lambda,j} e(k-j). \tag{18}$$

It follows from $K_i h^{\lambda} \sum_{j=0}^{\triangle t/h} q_{-\lambda,j} e(k-j) > 0$ that

$$u(t_{s2}) < 99.8. \tag{19}$$

Therefore, when $t > t_s$ the system output has an oscillation.

For the FOPI, the error signal is only valid within the memory length. The consequence is that the controller can not guarantee a null static error, and may led to system to oscillations. For solving this problem, the discrete scheme (14) needs to be modified.

3.2. An improved discrete FOPI controller

Herein an improved FOPI is proposed, where the error signal beyond the memory length is multiplied by $q_{-\lambda,M+1}$ and accumulated. The new scheme is given by:

$$\begin{aligned} u(k) = \quad & K_p e(k) + \sum_{j=0}^{M} \gamma_1 K_i h^{\lambda} q_{-\lambda,j} e(k-j) \\ & + \sum_{j=k-M+1}^{k-1} \gamma_2 K_i h^{\lambda} q_{-\lambda,M+1} e(k-j), \end{aligned} \tag{20}$$

where $k > M$, and γ_1 and γ_2 are constants.

For validating the improved FOPI we show in the follow-up that: 1) it yields an IOPI for particular values of the parameters; 2) it achieves results similar to those obtained by the Oustaloup method; and 3) expression (20) gives better performance than (14):

1. If $\lambda = 1$, then $q_{-\lambda,n} = 1, \forall n \in \mathbb{N}$. Taking $\gamma_1 = \gamma_2 = 1$, formula (20) can be written as follows:

$$\begin{aligned} u(k) &= K_p e(k) + \sum_{j=0}^{M} K_i h e(k-j) + \sum_{j=k-M+1}^{k-1} K_i h e(k-j) \\ &= K_p e(k) + K_i h \sum_{j=1}^{k} e(j), \end{aligned} \tag{21}$$

   Therefore, we obtain an IOPI controller.
2. Comparing the discrete form of (20) and the Oustaloup method for the example (16), with $K_p = 50, K_i = 500, \lambda = 0.9135$ and $\gamma_1 = \gamma_2 = 1$, we obtain the unit step responses depicted in Fig. 3.We note that the unit step response obtained by the two methods are identical, with only slight differences at some instants. This is mainly caused by the value of $q_{-\lambda,M+1}$. The differences between the two methods can be reduced by adjusting the parameter γ_2. The results show that the controller (20) can still be regarded as a FO controller.
3. The parameters γ_1 and γ_2 can be adjusted for improving the controller performance. Fig. 4. shows the unit step response for the system (16) and the optimal parameters are $K_p = 50, K_i = 500, \lambda = 0.4451, \gamma_1 = 2$ and $\gamma_2 = 1.415$.We note that the overshoot obtained with the Oustaloup method is $\sigma = 27.5\%$ and the adjusting time is $t_s = 3.4$ s. The overshoot with the controller (20) is $\sigma = 7\%$ and the adjusting time is $t_s = 2.2$ s. Therefore, the control action (20) is superior to the one produced by the Oustaloup method.

3.3. A variable-order fractional PI controller

For the FOPI, the integration order determines the weight of the error at different time instants. The proportional gain can be regarded as the weight of the error at the current moment, and the integral gain can be viewed as the ratio of

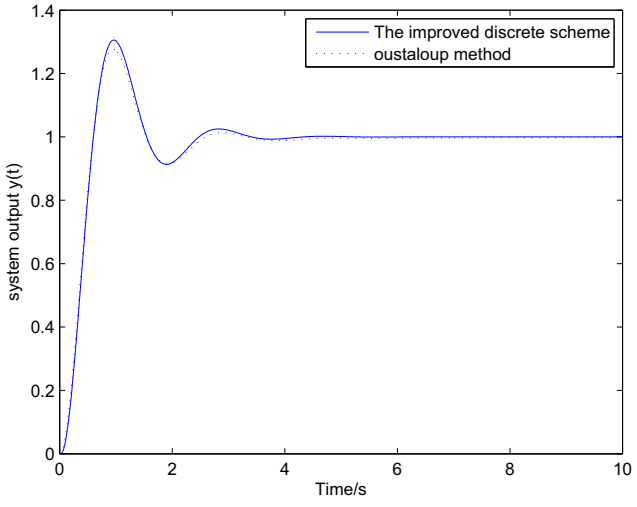

Fig. 3 Unit step responses obtained with (20) and the Oustaloup method.

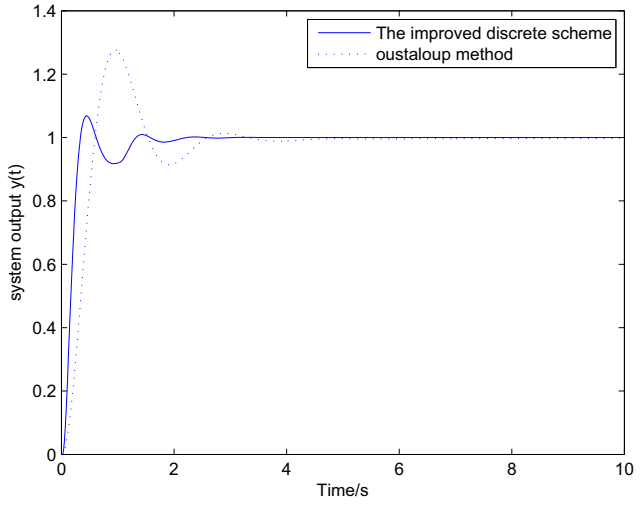

Fig. 4 Comparison of closed loop system response for formula (20) and the Oustaloup method.

the weight of the error at each moment. Comparing the functions of the three control parameters in formula (20), we verify that the integration order is the key parameter that affects the change of the control action. If we compare the FOPI and the IOPI, then, in order to improve the performance, we advance with a VFPI to make the integration order λ adaptive in time. Its discrete formulation is as follows:

$$u(k) = K_p e(k) + \sum_{j=0}^{M} \gamma_1 K_i h^{\lambda(t)} q_{-\lambda(t),j} e(k-j) + \sum_{j=k-M+1}^{k-1} \gamma_2 K_i h^{\lambda(t)} q_{-\lambda(t),M+1} e(k-j), \tag{22}$$

where $\lambda(t) = a + be^{-ct}$ or $\lambda(t) = a - be^{-ct}$.

When $c = 0$ and $\lambda = a + b$, the VFPI yields the FOPI controller. Therefore, expression (22) can be regarded as a generalization of (20).

For $t = t_L$ and t_L large, we have $e^{-ct_L} \to 0$. When $t > t_L$, we verify that the motor starts repeatedly. At this time, the VFPI algorithm is equivalent to a FOPI controller that has an integral order with value a. In order to increase the effectiveness of the VFPI in repeated start-up, it is necessary to add a zero-return operation in the control scheme, which means setting $t = 0$ when the speed set point is such that $e(k) = n_{ref}$. The speed loop of the VFPI controller is implemented is Steps 1--3 as follows:

Step 1	Calculate the error signal $e(k)$ at the current time, if $e(k) = n_{ref}$, set $t = 0$ and go to step 2; otherwise, go to step 2;
Step 2	Set $t = t + h$ and calculate $\lambda(t)$ and $q_{\lambda(t),j}$ from t;
Step 3	Update the control signal $u(k)$ using formula (22).

For some values of the parameters the VFPI can be viewed as a FOPI controller. Therefore, its performance is, in general, better than the one achieved by the FOPI. In the follow-up the control effect of the VFPI controller is verified when applied in the control of a PMSM.

Remark 1. Due to the variable integration order, the change of the control action is more flexible, and often a better control effect can be achieved. The integration order $\lambda(t) = a + be^{-ct}$ is adopted. It should be noted that other functions are also possible if the order $\lambda(t)$ approaches a constant value as time t tends to infinity.

4. Control experiment of PMSM

4.1. Simulation analysis of PMSM

The space vector pulse width modulation (SVPWM) and a double closed-loop structure that includes speed and current loops are used to control the PMSM, as shown in the block diagram in Fig. 5. The SVPWM makes the output current of the inverter close to the ideal sinusoidal waveform and drives the motor. The position and speed of the PMSM are obtained with a 2500 ppr encoder. In order to maximize the electromagnetic torque the value of i_d^* is equal to 0.

The PMSM specifications are given in Table Table 1. The reference speed n_{ref} is 600 rpm and the integral order is decreasing. The control parameters used in the simulations are shown in Table Table 2. The Matlab/Simulink is used to simulate and analyze the performance of the VFPI and the FOPI.

The PMSM load torque is $T_L = 5$ Nm and its starting performance is shown in Fig. 6. We verify that the FOPI leads to a much higher speed overshoot than the one obtained with the VFPI. The adjusting times are 0.04 s and 0.015 s, respectively. Therefore, we conclude that with respect to the starting performance of the PMSM, the VFPI controller is superior.

To analyse the anti-interference performance of the PMSM, the load torque of the motor is removed at 0.2 s. The corresponding speed curve is shown in Fig. 7.

When the PMSM suddenly unloads the load torque, the speed fluctuation of obtained with the FOPI is greater than the one exhibited by the VFPI. Thus, the FOPI has better anti-interference performance than the VFPI controller.

4.2. Experimental set-up and results

The experimental platform to test the PMSM is shown in Fig. 8. In the experiment, the control chip is the *TMS320F28335*, and the control algorithm is programmed in the software CCS6.1 using C language. The driver is the *IR2136* integrated circuit and the motor rating is 200 W@48 V. The load torque is provided by means of a magnetic powder brake.

The experimental data are read into a computer. The reference speed n_{ref} is 600 rpm, and the load torque of the PMSM is $T_L = 0.4$ N m. The starting characteristics of the PMSM are shown in Fig. 9.

The vibration of the motor shaft is caused by the rotation of the load, since they are coupled. Therefore, the speed exhibits some fluctuations when the motor is stable.

In the experiment, the speed overshoot of the PMSM controlled by means of the FOPI is 8.67%, and the adjusting time is about 0.4 s. The speed curve obtained with the VFPI has negligible overshoot, and the adjusting time is close to 0.28 s.

The load torque of the motor is removed at 5 s, and the speed curve of PMSM is shown in Fig. 10.

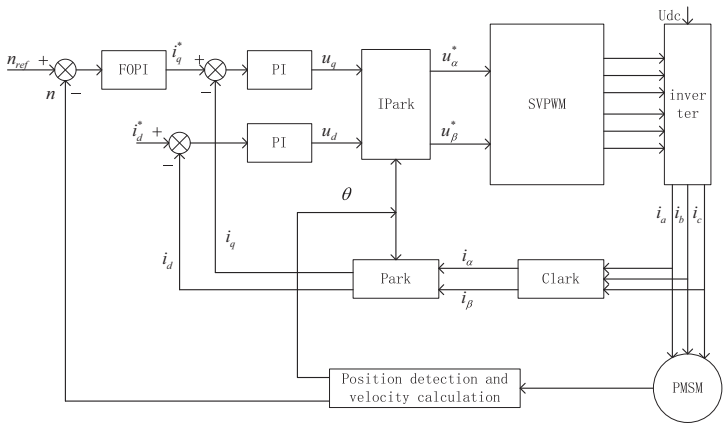

Fig. 5 The PMSM vector control diagram.

Table 1 Specification of the PMSM.

stator resistance R_s	1.014 Ω
stator inductance L_s	0.00079 H
pole pairs ρ_n	4
rotor flux linkage Ψ_f	0.1496 Wb
rotor inertia J	0.002 kg·m^2
viscous frictional coefficient B	0.0001

Table 2 Control parameters used in the simulations.

Controller	parameters
FOPI	$K_p = 0.1, K_i = 20, \lambda = 0.9$
	$\gamma_1 = 0.32, \gamma_2 = 0.21$
VFPI	$K_p = 0.1, K_i = 20, \gamma_1 = 2.4$
	$\gamma_2 = 0.0213, a = 0.5, b = 0.4, c = 100$

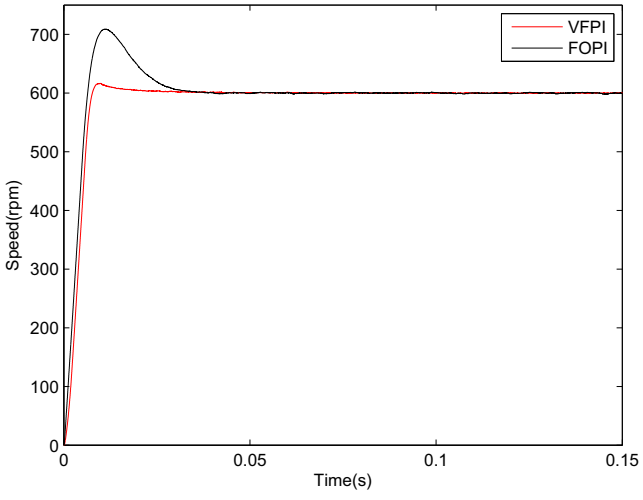

Fig. 6 Starting performance of the PMSM under close-loop control using the VFPI and the FOPI.

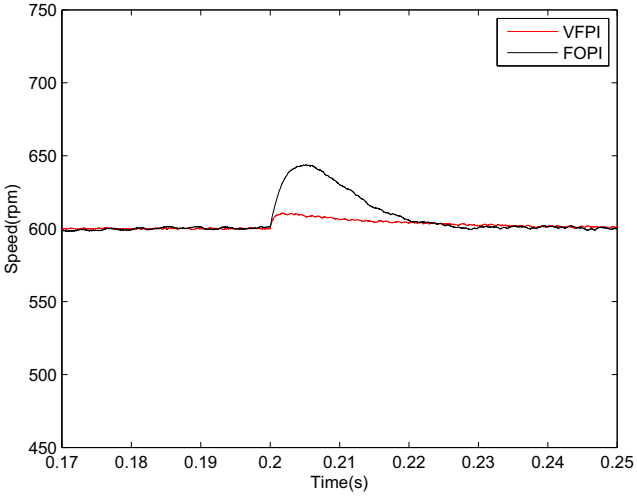

Fig. 7 Anti-interference performance of the PMSM under closed-loop control using the VFPI and the FOPI.

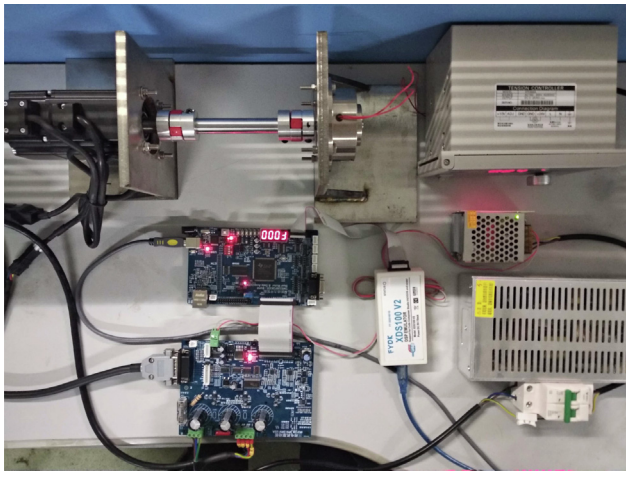

Fig. 8 Experimental platform of the PMSM control system.

We note that the speed response of the VFPI has smaller fluctuations, which means that the VFPI has a superior anti-interference action.

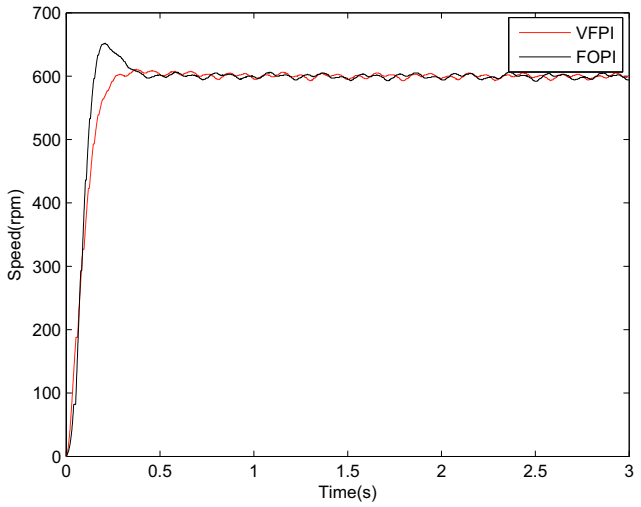

Fig. 9 Experimental starting characteristic of the PMSM under closed-loop control using the VFPI and the FOPI.

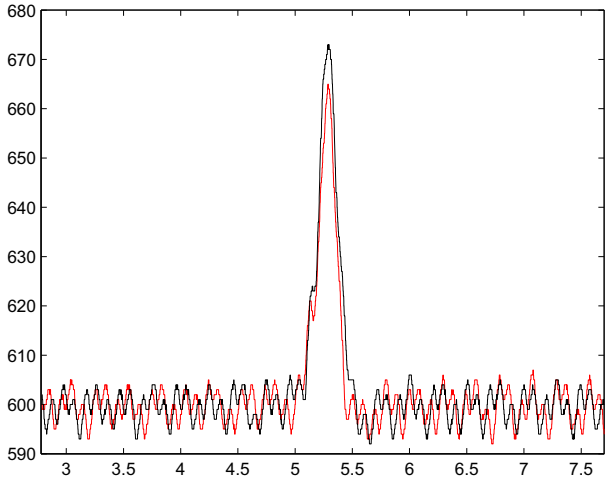

Fig. 10 The anti-interference performance chart of the PMSM under closed-loop control using the VFPI and the FOPI.

In summary, both the simulation and experimental results demonstrate that the VFPI controller has good performance.

5. Conclusion

A discrete FOPI controller was presented. The FOPI overcomes the performance degradation caused by using finite length memory to approximate the FO operators in the time domain. Moreover, a new VFPI controller that further improves the FOPI action was proposed. The two algorithms were applied in the control of a PMSM. Simulation and experimental result demonstrated the effectiveness of the control algorithms and that the VFPI has the best performance.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to express their deep gratitude to the Editors and the anonymous referees for their helpful comments and suggestions, which have greatly improved the paper. This work was supported by the National Natural Science Funds of China (No. 11971032), the Fundamental Research Funds for the Central Universities (No. JZ2019HGTB0090), Ministry of Education China Mobile Research Fund (MCM20180404) and Chongqing Basic Research and Frontier Exploration Project (cstc2018jcyjAX0167).

References

[1] S. Arshad, D. Baleanu, Y. Tang, Fractional differential equations with bio-medical applications, in: Dumitru Baleanu, António Mendes Lopes (Eds.), Applications in Engineering, Life and Social Sciences, Part A in Handbook of Fractional Calculus with Applications, Berlin, De Gruyter, 2019.

[2] L. Chen, R. Wu, Y. He, L. Yin, Robust stability and stabilization of fractional-order linear systems with polytopic uncertainties, Appl. Math. Comput. 257 (2015) 274–284.

[3] B.M. Hossein Jafari, D. Baleanu, Fractional calculus for modeling unconfined groundwater, in: Dumitru Baleanu, António Mendes Lopes (Eds.), Applications in Engineering, Life and Social Sciences, Part A in Handbook of Fractional Calculus with Applications, Berlin, De Gruyter, 2019.

[4] R. Yury, S. Marina, Fractional calculus models in dynamic problems of viscoelasticity, in: Dumitru Baleanu, António Mendes Lopes (Eds.), Applications in Engineering, Life and Social Sciences, Part A in Handbook of Fractional Calculus with Applications, Berlin, De Gruyter, 2019.

[5] L. Chen, H. Yin, T. Huang, L. Yuan, S. Zheng, L. Yin, Chaos in fractional-order discrete neural networks with application to image encryption, Neural Netw. 125 (2020) 174–184.

[6] T. Jan, Fractional heat conduction models and their applications, in: Dumitru Baleanu, António Mendes Lopes (Eds.), Applications in Engineering, Life and Social Sciences, Part A in Handbook of Fractional Calculus with Applications, Berlin, De Gruyter, 2019.

[7] K. Xu, L. Chen, M. Wang, A.M. Lopes, J. Tenreiro Machado, H. Zhai, Improved decentralized fractional PD control of structure vibrations, Mathematics 8 (3) (2020) 326.

[8] J. Pinheiro Neto, D. Valério, S. Vinga, Variable-order derivatives and bone remodeling in the presence of metastases, in: Dumitru Baleanu, António Mendes Lopes (Eds.), Applications in Engineering, Life and Social Sciences, Part A in Handbook of Fractional Calculus with Applications, Berlin, De Gruyter, 2019.

[9] J. Zhong, L. Li, Tuning fractional-order $PI^{\lambda}D^{\mu}$ controllers for a solid-core magnetic bearing system, IEEE Trans. Control Syst. Technol. 23 (4) (2015) 1648–1656.

[10] K. Erenturk, Fractional-order $PI^{\lambda}D^{\mu}$ and active disturbance rejection control of nonlinear two-mass drive system, IEEE Trans. Industr. Electron. 60 (9) (2013) 3806–3813.

[11] R. Duma, P. Dobra, M. Trusca, Embedded application of fractional order control, Electron. Lett. 48 (24) (2012) 1526–1528.

[12] S. Folea, C.I. Muresan, R. De Keyser, C.M. Ionescu, Theoretical analysis and experimental validation of a simplified fractional order controller for a magnetic levitation system, IEEE Trans. Control Syst. Technol. 24 (2) (2015) 756–763.

[13] D. Wang, D. Zhao, M. Gong, B. Yang, Research on robust model predictive control for electro-hydraulic servo active suspension systems, IEEE Access 6 (2018) 3231–3240.

[14] A. Dabiri, B.P. Moghaddam, J.T. Machado, Optimal variable-order fractional PID controllers for dynamical systems, J. Comput. Appl. Math. 339 (2018) 40–48.
[15] L. Chen, G. Chen, R. Wu, A.M. Lopes, J.A.T. Machado, H. Niu, Variable coefficient fractional-order PID controller and its application to a SEPIC device, IET Contr. Theor. Appl. 14 (6) (2020) 900–908.
[16] O. Abu Arqub, Application of residual power series method for the solution of time-fractional schrödinger equations in one-dimensional space, Fundamenta Informaticae 166 (2) (2019) 87–110.
[17] O. Abu Arqub, Numerical algorithm for the solutions of fractional order systems of Dirichlet function types with comparative analysis, Fundamenta Informaticae 166 (2) (2019) 111–137.
[18] O.A. Arqub, N. Shawagfeh, Application of reproducing kernel algorithm for solving Dirichlet time-fractional diffusion-gordon types equations in porous media, J. Porous Media 22 (4) (2019) 411–434.
[19] O.A. Arqub, Numerical solutions of systems of first-order, two-point bvps based on the reproducing kernel algorithm, Calcolo 55 (3) (2018) 1–28.
[20] I. Podlubny, Fractional Differential Equations, Academic Press, San Diego, 1999.
[21] D. Saptarshi, P. Indranil, On the mixed H_2/H_∞ loop-shaping tradeoffs in fractional-order control of the AVR system, IEEE Trans. Industr. Inf. 10 (4) (2014) 1982–1991.
[22] P. Indranil, D. Saptarshi, Kriging based surrogate modeling for fractional order control of microgrids, IEEE Trans. Smart Grid 6 (1) (2015) 36–44.
[23] Y. Chen, Discretization schemes for fractional-order differentiators and integrators, IEEE Trans. Circ. Syst. I: Reg. Pap. 49 (3) (2002) 363–367.
[24] G. Maione, On the Laguerre rational approximation to fractional discrete derivative and integral operators, IEEE Trans. Autom. Control 58 (6) (2013) 1579–1585.
[25] Z. Wang, Z. Wang, G. Cao, X. Zhu, Digital implementation of fractional order PID controller and its application, J. Syst. Eng. Electron. 16 (1) (2005) 116–122.
[26] J.-Y. Cao, B.-G. Cao, Digital realization and characteristics of fractional order controllers, Contr. Theor. Appl. 23 (5) (2006) 791–799.
[27] L. Angel, J. Viola, Design and statistical robustness analysis of FOPID, IOPID and SIMC PID controllers applied to a motor-generator system, IEEE Latin Am. Trans. 13 (12) (2015) 3724–3734.
[28] J. Viola, L. Angel, Factorial design for robustness evaluation of fractional PID controllers, IEEE Latin Am. Trans. 13 (5) (2015) 1286–1293.
[29] N.Z. Davijani, G. Jahanfarnia, A.E. Abharian, Nonlinear fractional sliding mode controller based on reduced order FNPK model for output power control of nuclear research reactors, IEEE Trans. Nucl. Sci. 64 (1) (2017) 713–723.
[30] G. Sun, Z. Ma, J. Yu, Discrete-time fractional order terminal sliding mode tracking control for linear motor, IEEE Trans. Industr. Electron. 65 (4) (2018) 3386–3394.
[31] M.Ö. Efe, Fractional fuzzy adaptive sliding-mode control of a 2-DOF direct-drive robot arm, IEEE Trans. Syst. Man Cybernet. Part B (Cybernet.) 38 (6) (2008) 1561–1570.
[32] S. Xu, G. Sun, Z. Ma, X. Li, Fractional-order fuzzy sliding mode control for the deployment of tethered satellite system under input saturation, IEEE Trans. Aerosp. Electron. Syst. 55 (2) (2019) 747–756.