IET Control Theory & Applications

Brief Paper

IET Journals
The Institution of Engineering and Technology

Variable coefficient fractional-order PID controller and its application to a SEPIC device

ISSN 1751-8644
Received on 1st April 2019
Revised 6th November 2019
Accepted on 2nd January 2020
E-First on 16th March 2020
doi: 10.1049/iet-cta.2019.0361
www.ietdl.org

Liping Chen[1] ✉, Gang Chen[1], Ranchao Wu[2], António M. Lopes[3], José António Tenreiro Machado[4], Haihong Niu[1]

[1]*School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, People's Republic of China*
[2]*School of Mathematics, Anhui University, Hefei 230039, People's Republic of China*
[3]*UISPA–LAETA/INEGI, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal*
[4]*Department of Electrical Engineering, Institute of Engineering, Polytechnic of Porto, R. Dr. António Bernardino de Almeida, 431, 4249-015 Porto, Portugal*
✉ *E-mail: lip_chenhut@126.com*

Abstract: The fractional-order proportional–integral–derivative (FOPID) controller has two more parameters than the integer-order proportional–integral–derivative (PID). Such characteristic makes the controller design more flexible and leads to superior performance. This study proposes a variable coefficient FOPID (VCFOPID) with optimal single step parameters, combining discrete synthesis and variable control parameters. The new algorithm is compared with previous FOPID discrete methods via several examples. Since the energy losses of the single-ended primary-inductor converter (SEPIC) cannot be ignored, the standard models are insufficient and a new model is derived using quantum-behaved particle swarm optimisation. The VCFOPID is applied to the SEPIC and both the effectiveness of the controller and the model are verified experimentally.

1 Introduction

The fractional-order proportional–integral–derivative (FOPID) controller has been proved to have better performance than the standard integer-order proportional–integral–derivative (IOPID) in many engineering applications [1–4]. The development of the FOPID is closely related to the definition, approximation, and discretisation of fractional-order (FO) operators [5, 6]. The continuous fractional expansion or the frequency-based methods proposed by Oustaloup and Carlson are possible alternatives for approximating such operators [7, 8]. The Oustaloup approximation is often used and several researchers introduced improvements on the original method [9–11]. Since the analogue circuit-based realisation of FO operators is not straightforward and its accuracy is difficult to guarantee [12, 13], the digital implementation is often chosen, thus requiring the use of discretising methods. The most used approach in industrial applications is the direct discrete method [14, 15]. However, this procedure needs to use the short-term memory rule to ensure the storage space needed for the controller. Moreover, the number of loop points affects the performance of the control algorithm. Therefore, some researchers proposed simpler discrete schemes by means of approximate techniques [16–20].

In spite of these advantages, a broader perspective reveals that, in the presence of complex dynamical effects, the FOPID can still have difficulties in achieving satisfactory results. In those cases, it is necessary to improve the FOPID to enhance control performance. For this purpose, several researchers combined fractional calculus (FC) with different methods to design new controllers, such as the FO terminal sliding mode [21], fuzzy fractional integral sliding mode [22], and other algorithms [23]. On the other hand, it was shown that one reason for the good performance of the model reference adaptive controller is that its parameters are constantly adapting to the operating conditions [24, 25]. Motivated by this idea, a new controller, viz. the variable coefficient FOPID (VCFOPID) is proposed by combining the direct discrete implementation of the FO operator with the method of variable control parameters.

Some papers introduced high performance intelligent optimisation algorithms to adjust the parameters of the FOPID and the results revealed some success [26–28]. For example, the quantum-behaved particle swarm optimisation (QPSO) algorithm, a recent variant of the standard particle swarm optimisation (PSO) with high randomicity and global convergence was adopted [29, 30], demonstrating relevant results.

In this study, we propose an improved version of the FOPID, namely the VCFOPID, to control a single-ended primary-inductor converter (SEPIC). Since the loss factor is not considered, the standard SEPIC mathematical models are not sufficiently accurate. Therefore, we adopt a QPSO algorithm to fit the open loop response curve of the SEPIC and we derive a new SEPIC model. The superior performance of the VCFOPID is verified experimentally.

The rest of this paper is organised as follows. Section 2 introduces the FOPID ($PI^{\lambda}D^{\mu}$) and the QPSO algorithm. Section 3 overviews several FOPID schemes proposed in the literature, introduces the VCFOPID, and compares the performance of the controllers by means of several examples. Section 4 addresses the modelling and control of a SEPIC. Section 5 analyses the maximum sensitivity of several examples and the SEPIC. Finally, Section 6 outlines the main conclusions.

2 Preliminary concepts

In this subsection, the fundamental aspects of the $PI^{\lambda}D^{\mu}$ controller and the QPSO algorithm are introduced.

2.1 $PI^{\lambda}D^{\mu}$ controller

The FC is essentially an arbitrary order calculus, where the order can be any real or complex number. The fractional derivative is expressed as

$$ {}_aD_t^q = \begin{cases} \dfrac{\mathrm{d}^q}{\mathrm{d}t^q}, & R(q) > 0, \\ 1, & R(q) = 0, \\ \displaystyle\int_a^t (\mathrm{d}\tau)^{-q}, & R(q) < 0, \end{cases} \tag{1} $$

where $q \in \mathbb{C}$ represents the order, a and t denote the upper and lower bounds of the FO operation, respectively, and $R(q)$ is the real part of q.

Two frequently used definitions of fractional derivatives are the Riemann–Liouville and Grünwald–Letnikov formulations given by

$$ {}_aD_t^q f(t) = \frac{1}{\Gamma(n-q)} \frac{\mathrm{d}^n}{\mathrm{d}t^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{q-n+1}} \mathrm{d}\tau, $$

$$ {}_aD_t^q f(t) = \lim_{h \to 0} \left(\frac{1}{h^q} \right) \sum_{j=0}^{[(t-q)/h]} (-1)^j \binom{q}{j} f(t - jh), $$

where $n - 1 < q < n,\ n \in \mathbb{N}$, $\Gamma(\cdot)$ denotes the Gamma function so that $\Gamma(s) = \int_0^\infty t^{s-1} e^{-t} \mathrm{d}t$, and

$$ \binom{q}{j} = \frac{\Gamma(q+1)}{\Gamma(j+1)\Gamma(q-j+1)}. $$

For the $PI^\lambda D^\mu$ controller, the control signal $u(t)$ in the time domain can be expressed as

$$ u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} D^{-\lambda} e(t) + K_\mathrm{d} D^\mu e(t), \tag{2} $$

where $\lambda, \mu \in \mathbb{R}^+$ are the orders of the integral and derivative actions, and K_p, K_i, and K_d stand for the proportional, integral, and derivative gains, respectively.

Applying the Laplace transform to formula (2), the FO transfer function can be obtained as follows:

$$ G_c(s) = K_\mathrm{p} + \frac{K_\mathrm{i}}{s^\lambda} + K_\mathrm{d} s^\mu. \tag{3} $$

As shown in (3), the FOPID has five control parameters that provide higher freedom in the design process, but, on the other hand, make the parameters tuning a more laborious task.

2.2 Quantum-behaved particle swarm optimisation

Sun *et al*. [31] first proposed the QPSO, combining the quantum mechanism and swarm intelligence. In terms of algorithm flow, there is no significant difference between the QPSO and the classic PSO. The QPSO generates new solutions through the following probability model of the wave function:

$$ X = p \pm \frac{L}{2} \ln(1/u), \tag{4a} $$

$$ p = \frac{c_1 r_1 p_\mathrm{best} + c_2 r_2 g_\mathrm{best}}{c_1 r_1 + c_2 r_2}, \tag{4b} $$

where $u, r_1, r_2 \sim U(0, 1)$, p_best is the individual optimum, and g_best is the global optimum positions, respectively. The parameter L is controlled by

$$ L_{i,j}(t) = 2\alpha \left| C_j(t) - X_{i,j}(t) \right|, $$

$$ \alpha = (1 - 0.5)(maxiter - iter)/(maxiter + 0.5), $$

where *maxiter* is the maximum number of iterations and *iter* is the number of current iterations. Therefore, the updating rule of the particle position can be written as

$$ X_{i,j}(t+1) = p_{i,j}(t) \pm \alpha \left| C_j(t) - X_{i,j}(t) \right| \ln[1/u_{i,j}(t)], $$

where α denotes the contraction–expansion coefficient that diminishes with the increasing of the number of iterations, and $C(t) = (1/M) \sum_{i=1}^{M} P_\mathrm{i}(t)$.

The fitness function is a key point of the algorithm that often represents the main goal of the optimisation. For the controller parameter optimisation problem, a common fitness function is the integral of time multiplied by the absolute value of the error (ITAE) as follows:

$$ J = \int t \left| e(t) \right| \mathrm{d}t. \tag{5} $$

In this study, the QPSO algorithm is used not only for the optimisation of controller parameters but also for the SEPIC modelling. The parameters adopted in the QPSO algorithm are shown in Table 1.

3 Design and analysis of the controller

In this subsection, the VCFOPID algorithm is presented. When implemented in digital signal processors (DSPs), the computational load is reduced by changing the discrete format. The VCFOPID is compared with existing FOPID controllers through several examples to illustrate its effectiveness.

3.1 Discrete schemes

The implementation of a digital controller requires the discretisation of the continuous control signal in the time domain. Using the GL FO operator, the signal $u(t)$ of formula (2) is discretised as [14, 15]

$$ \begin{aligned} u(k) = K_\mathrm{p} e(k) + K_\mathrm{i} h^\lambda \sum_{j=0}^{m} d_j e(k-j) \\ + K_\mathrm{d} h^{-\mu} \sum_{j=0}^{m} q_j e(k-j), \end{aligned} \tag{6} $$

where

$$ \begin{cases} d_0 = 1; d_j = \left(1 - \dfrac{1+\lambda}{j}\right) d_{j-1}, & j = 1, 2, 3, \ldots \\ q_0 = 1; q_j = \left(1 - \dfrac{1+\mu}{j}\right) q_{j-1}, & j = 1, 2, 3, \ldots \end{cases} $$

and m is the number of circulant points in the storage space of a digital controller.

As shown in (6), the integral and differential terms include memory, i.e. related to the error at previous time instants. Since the memory space of a DSP is limited, the short-term memory rule is used to determine the number of samples. The output waveforms obtained in the absence or in the presence of the short-term memory rule are shown in Figs. 1 and 2, respectively, where the inputs are pulse waveforms with different duty cycles and the outputs are voltage signals.

The absence of the short-term memory rule can cause erroneous control signals and lead to an oscillatory output. These problems may burn the controlled circuit. Therefore, it is necessary to set up the number of loop points in the short-term memory rule when using the discrete scheme. Hereafter, the controller designed in discrete form (6) is named as DFOPID.

The number of loop points affects the performance of the DFOPID, making its design more complex. In order to simplify the

Table 1 Parameter values adopted with the QPSO algorithm

number of particles	Maxiter	c_1	c_2
100	100	2	2.1

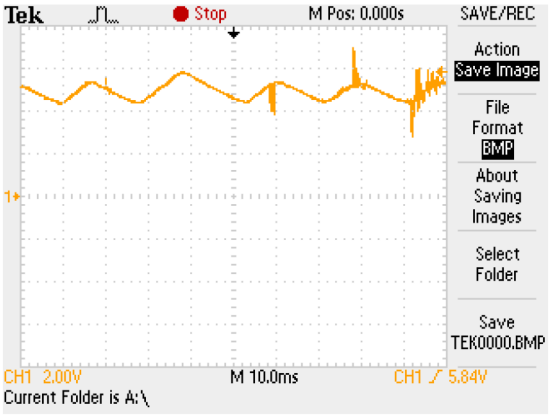

Fig. 1 *Output waveform in the absence of short-term memory rule*

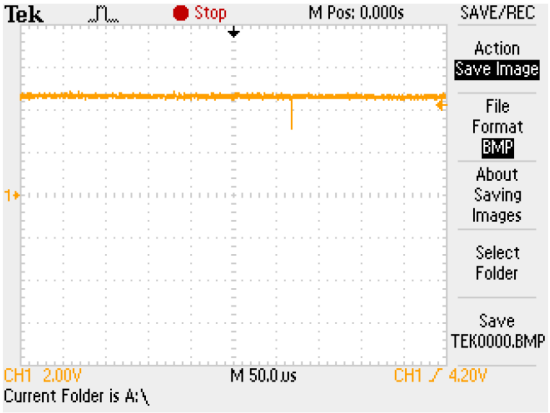

Fig. 2 *Output waveform in the presence of short-term memory rule*

process, a different discrete method is proposed. In this case, the Laplace variable s in (3) is substituted by some function of z and then the power series expansion (PSE) of the resulting expression is obtained in terms of z.

Let us assume that the sampling time is T and the pre-warped Tustin method is applied to the derivative item of (3). Then, we have

$$s^{\mu} = \left[\frac{\omega_c}{\tan(\omega_c T/2)} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right]^{\mu} = \alpha^{\mu}\left[\frac{1 - z^{-1}}{1 + z^{-1}}\right]^{\mu}, \tag{7}$$

where $\alpha = \omega_c/\tan(\omega_c T/2)$ and ω_c is the gain crossover frequency.

Assuming that $\omega = z^{-1}$ and $|z| > 1$, the PSE on the right-hand-side of expression (7) can be written as follows:

$$\begin{aligned} \alpha^{\mu}\left[\frac{1 - \omega}{1 + \omega}\right]^{\mu} &= \alpha^{\mu}\sum_{k=0}^{\infty} f_k(\mu)\omega^k, \quad \left|\omega\right| < 1, \\ s^{\mu} &= \alpha^{\mu}\sum_{k=0}^{\infty} f_k(\mu)z^{-k}, \quad \left|z\right| < 1, \end{aligned} \tag{8}$$

where

$$f_k(\mu) = \frac{1}{k!} \cdot \frac{d^k}{d\omega^k}\left(\frac{1 - \omega}{1 + \omega}\right)_{|\omega = 0}. \tag{9}$$

In order to find a series approximation for $s^{-\lambda}$ in terms of z^{-1}, we write $s^{-\lambda}$ as $(1/s) \cdot s^{1-\lambda}$ and apply the pre-warped Tustin method obtaining

$$s^{-\lambda} = \alpha^{-\lambda}\frac{1 + z^{-1}}{1 - z^{-1}}\sum_{k=0}^{\infty} f_k(1 - \lambda)z^{-k}, \quad \left|z\right| > 1, \tag{10}$$

where $f_k(1 - \lambda)$ is calculated from formula (9).

Substituting (8) and (10) into (3), yields

$$\begin{aligned} C_d(z) = K_p + K_d\alpha^{\mu}\sum_{k=0}^{\infty} f_k(\mu)z^{-k} \\ + K_i\alpha^{-\lambda}\frac{1 + z^{-1}}{1 - z^{-1}}\sum_{k=0}^{\infty} f_k(1 - \lambda)z^{-k}. \end{aligned}$$

Limiting the number of memory units to M, the difference equation relating $e(k)$ to $u(k)$ is as follows:

$$u(k) = u(k-1) + u_p(k) + u_d(k) + u_i(k), \tag{11}$$

where

$$\begin{aligned} u_p(k) &= K_p[e(k) - e(k-1)], \\ u_d(k) &= \sum_{n=0}^{M} K_d\alpha^{\mu} f_n(\mu)[e(k-n) - e(k-n-1)], \\ u_i(k) &= \sum_{n=0}^{M} K_i\alpha^{-\lambda} f_n(1 - \lambda)[e(k-n) - e(k-n-1)]. \end{aligned}$$

Often the value $M = 5$ is adopted.

The controller designed by means of discrete form (11) will be called IFOPID1 [5]. proposed other discrete schemes, such as in the control of the DC motor [20].

The control input is

$$u(k) = u_p(k) + u_d(k) + u_i(k), \tag{12}$$

where

$$\begin{aligned} u_p(k) &= K_p e(k), \\ u_d(k) &= K_d\alpha^{\mu}[e(k) - f_1(u)e(k-1)] \\ &\quad + \sum_{n=2}^{6} f_n(u)e(k-n), \\ u_i(k) &= K_i\alpha^{-\lambda}[e(k) - e(k-1)] \\ &\quad -[f_1(1-\lambda) - 1]u_i(k-1) \\ &\quad + \sum_{n=2}^{6} -[f_n(1-\lambda) - f_{n-1}(1-\lambda)]u_i(k-n) \\ &\quad + f_6(1-\lambda)u_i(k-7). \end{aligned}$$

The controller based on discrete form (12) will be called IFOPID2.

If we compare the discrete schemes of DFOPID, IFOPID1, and IFOPID2, we find that the computational complexity of IFOPID1 and IFOPID2 is smaller than the one obtained for the DFOPID. However, the structure of the DFOPID is more intuitive, and its accuracy can be adjusted by the number of loop points. Therefore, the three controllers have their own pros and cons. In order to determine the discrete form of the FOPID, the performance of the three algorithms is analysed in the follow-up with various examples.

3.2 Design of the VCFOPID controller

Here, we present the new VCFOPID algorithm. The control parameters are placed in the storage space of the DSP. The error signal is sampled and m points are stored using the stack storage mechanism, i.e. following the first-in first-out mode. In the initial transient, the VCFOPID parameters adapt quickly, producing a good system performance. When the system approaches the steady-state, the control parameters tend to be constant, and the controller can be seen as the standard FOPID.

The VCFOPID is obtained by combining discrete form (6) with the variable control parameters. The discrete scheme is given by

$$u(k) = K_{\mathrm{p}}(k)e(k) + K_{\mathrm{i}}(k)h^{\lambda(k)}\sum_{j=0}^{m} d_j e(k-j) + K_{\mathrm{d}}(k)h^{-\mu(k)}\sum_{j=0}^{m} q_j e(k-j), \tag{13}$$

so that

$$\begin{cases} d_0 = 1; d_j = \left(1 - \dfrac{1-\lambda}{j}\right)d_{j-1}, & j = 1, 2, 3, \ldots \\ q_0 = 1; q_j = \left(1 - \dfrac{1+\mu}{j}\right)q_{j-1}, & j = 1, 2, 3, \ldots \end{cases}$$

where K_{p}, K_{i}, K_{d}, λ, and μ are arrays of length m_{l}.

Since the FO parameters λ and μ change for every time sample, the values of arrays d and q need to be re-calculated. The larger the value of m_{l}, the higher the load of the calculation. Therefore, the value of m_{l} should not be too large and, in general, we have $m_{\mathrm{l}} < m$. The VCFOPID plays only an important role in the start-up process of the system. When the value of the counter is $i > m_{\mathrm{l}}$, the controller is switched to the DFOPID. The control parameters of the DFOPID are written into the arrays K_{p}, K_{i}, K_{d}, λ, and μ, respectively. In other words, the element $K_{\mathrm{p}}[m_{\mathrm{l}} + 1]$ is the control parameter of the DFOPID. The control flow is as follows:

- Step 1. Determine the value i of the counter.
  If $i < m_{\mathrm{l}}$, turn to the step 2;
  otherwise, turn to the step 3.
- Step 2. Let $i = i + 1$ and $u(k)$ is updated by formula (13).
- Step 3. Let $i = i + 1$ and $u(k)$ is updated by formula (6).

If all elements in the arrays K_{p}, K_{i}, K_{d}, λ, and μ are identical, then the VCFOPID can be regarded as a DFOPID controller. Therefore, formula (13) can be interpreted as the general form of expression (6). Since the control parameters of VCFOPID are variable, the variation range of the controlled quantity u increases, leading to a VCFOPID control performance superior to the one exhibited by the DFOPID.

3.3 Comparative analysis

In this subsection, we present three examples for illustrating the performance of VCFOPID.

Example 1: Viola *et al*. [16] used the least square method to fit the motion trajectory of a pendulum and to obtain its dynamic model, given by the transfer function

$$G(s) = \frac{0.49}{0.496s^2 + 1.3s + 1}. \tag{14}$$

For this system, it is assumed that the control parameters are constant, namely $K_{\mathrm{p}} = 10$, $K_{\mathrm{i}} = 1$, $K_{\mathrm{d}} = 1$, $\lambda = 0.3$, and $\mu = 0.7$. Fig. 3 shows the simulation results obtained with Simulink and by the digital formulation (13).

The curves obtained by the two simulation methods are basically consistent and the subtle differences may be caused by the number of sampling points in the digital simulation. The results show the effectiveness of control algorithm (13).

For the system shown in formula (14), the QPSO algorithm is used to find the controller parameters. The optimisation intervals for setting the parameters are $K_{\mathrm{p}} \in [0, 20]$, $K_{\mathrm{i}}, K_{\mathrm{d}} \in [0, 5]$, $\lambda, \mu \in [0, 1]$ and $m_{\mathrm{l}} = 20$. Table 2 summarises the parameters values of the three controllers. Figs. 4 and 5 depict the output and control signals, respectively, for the IOPID, DFOPID and VCFOPID algorithms.

We verify that the curves for the FOPID have better initial performance than those achieved by the IOPID. The VCFOPID exhibits the best performance, yielding the smallest setting time.

Example 2: Here the parameters of the FOPID are adjusted in the frequency domain for the second-order system [18]

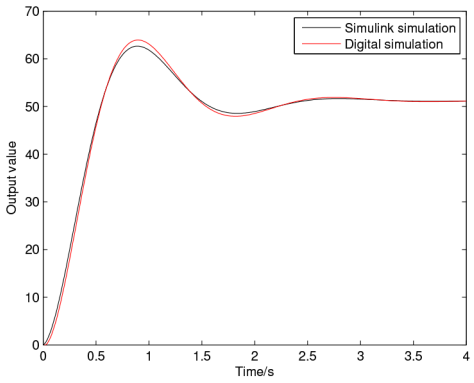

Fig. 3 *Simulation results obtained with Simulink and digital formulation (13)*

Table 2 Parameter values of the VCFOPID, FOPID and IOPID controllers in Example 1

VCFOPID	K_{p}	[0 0 20 0 0 20 0 0 2.9336 0 20 0 0 20 20 20 20 20 20 20 20]
	K_{i}	[5 0 5 0 5 0 0 5 5 0 0 0 0 0.0017 0 0 5 5 5 5 5]
	K_{d}	[5 0 5 0 4.9463 5 5 0 5 0 0 0 5 0 0 5 0 5 0 5 5]
	λ	[1 1 1 1 1 1 1 1 0 0 0 0 1 0 0.2579 1 0 0 1 0 1]
	μ	[0 0 1 1 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0.3069 1 0.9623]
FOPID	$K_{\mathrm{p}} = 5.8206$ $K_{\mathrm{i}} = 5$ $K_{\mathrm{d}} = 2.0772$ $\lambda = 1$ $\mu = 0.8994$	
IOPID	$K_{\mathrm{p}} = 6.3295$ $K_{\mathrm{i}} = 5$ $K_{\mathrm{d}} = 2.1864$ $\lambda = 1$ $\mu = 1$	

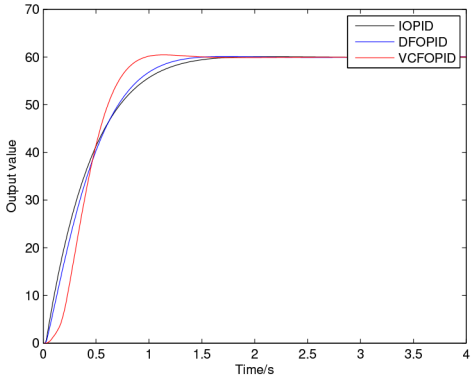

Fig. 4 *Output responses of the closed-loop system in Example 1 for the IOPID, DFOPID, and VCFOPID algorithms*

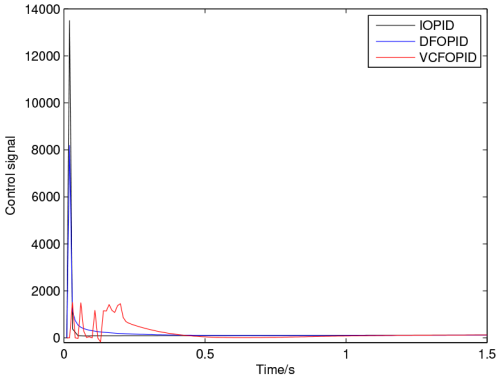

Fig. 5 *The control signals in Example 1 for the IOPID, DFOPID and VCFOPID algorithms*

Table 3 Parameter values of the FOPID and IFOPID1 controllers in Example 2

	K_p	K_i	K_d	λ	μ	α
FOPID	0.415	1.743	1.50	0.758	0.682	–
IFOPID1	12.476	5	5	0.6627	0.3375	1000

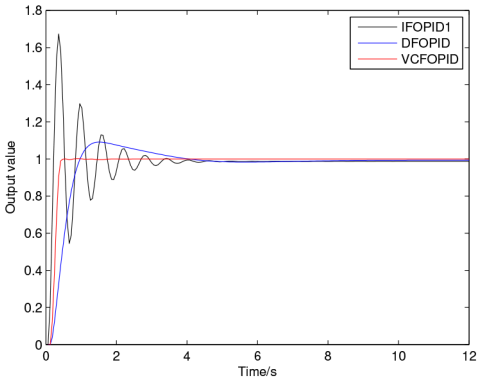

Fig. 6 *Unit step responses of the closed-loop system in Example 2 for the IFOPID1, DFOPID and VCFOPID algorithms*

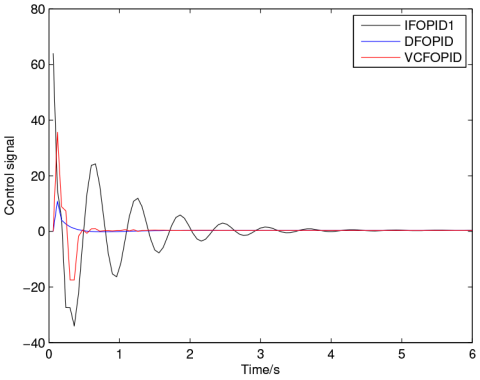

Fig. 7 *Control signals in Example 2 for the IFOPID1, DFOPID and VCFOPID algorithms*

$$G(s) = \frac{2}{s^2 + 1.66s + 0.666}.$$

The VCFOPID and IFOPID1 are designed using the QPSO algorithm. For the IFOPID1, we consider α as a variable. The sampling time is set to 0.1 s and the values of the control parameters are listed in Table 3.

As shown in Fig. 6, the IFOPID1 produces a large overshoot and high oscillations, while the VCFOPID exhibits the shortest settling time and no overshoot. The control signals are shown in Fig. 7.

Example 3: The following time delay transfer function, adopted in the DC motor model [20], is considered:

$$G(s) = \frac{0.63606}{(42.77s + 1)(7.45s + 1)} e^{-0.61s}.$$

For this system, the performance of the IFOPID2 is analysed. The optimisation ranges for setting the parameters are $K_p \in [0, 20]$, $K_i, K_d, \lambda, \mu \in [0, 1]$, and $\alpha \in [0, 10,000]$. It is assumed that the sampling time is 0.5 s, and the optimisation is accomplished by means of a QPSO algorithm, yielding the parameters values listed in Table 4.

In Fig. 8 we verify that the overshoot of the step response is about 150% for the IFOPID2 and that the system starts to shock after 70 s. In fact, the parameters can be adjusted for stabilising the step response, but the overshoot amplitude and the settling time of the response are increased by 160% and 100 s, respectively.

Table 4 Parameter values of the DFOPID and IFOPID2 controllers in Example 3

	K_p	K_i	K_d	λ	μ	α
DFOPID	8.343	0.178	1	1	0.374	–
IFOPID2	20	0.9998	1	0.334	0.3128	10,000

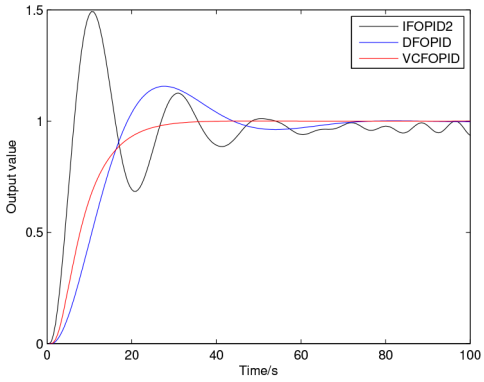

Fig. 8 *Unit step responses of the closed-loop system in Example 3 for the IFOPID2, DFOPID and VCFOPID algorithms*

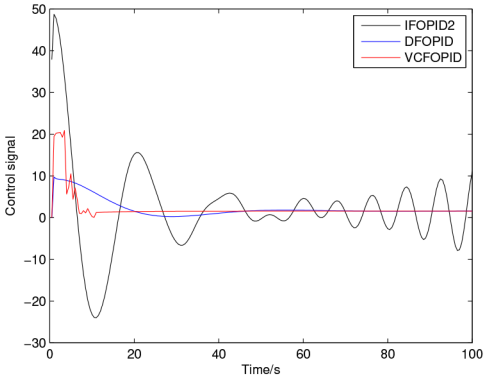

Fig. 9 *Control signals in Example 3 for the IFOPID2, DFOPID, and VCFOPID algorithms*

Therefore, the performance of the IFOPID2 is inferior to those exhibited by DFOPID and VCFOPID. Fig. 9 shows the control signals.

To sum up, IFOPID1 and IFOPID2 simplify the algorithm structure, making the design and implementation of the controller less demanding, but their performance is inferior to the one revealed by the DFOPID. Formula (6) is chosen for the discrete form of the FOPID, and the VCFOPID is obtained by improving the FOPID. In fact, the design of the VCFOPID is slight more complex, but the control performance is the best. The purpose of the VCFOPID design is to achieve a smaller overshoot and to shorten the adjustment time at the system starting phase.

From the analysis of the system transient in the above three examples, one can observe that the overshoot and adjustment time for the VCFOPID are inferior to those of the FOPID.

4 Modelling and testing

In this section, the QPSO algorithm is adopted in two distinct applications: (i) to obtain a fit between the experimental open-loop response of the SEPIC and its mathematical model and (ii) to optimise the parameters of the VCFOPID for controlling the SEPIC.

4.1 Modelling of SEPIC

The SEPIC is an electronic system for rising or dropping the voltage. The converter input and output currents are continuous, and the polarity of the input and output voltages is identical. The circuit diagram of the SEPIC is shown in Fig. 10.

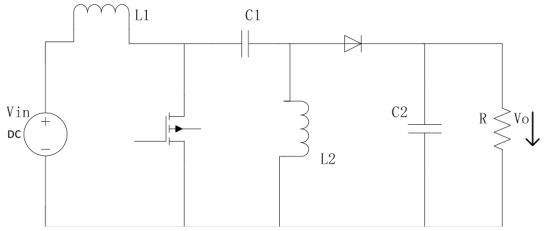

Fig. 10 *Circuit diagram of the SEPIC*

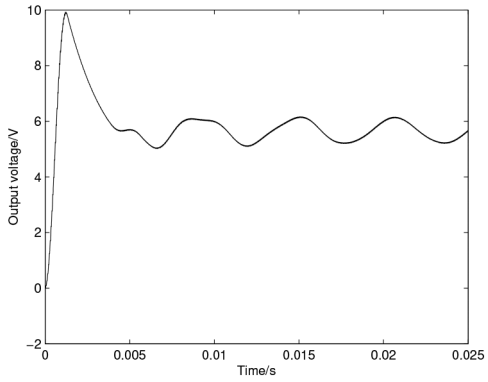

Fig. 11 *Step response of model (15)*

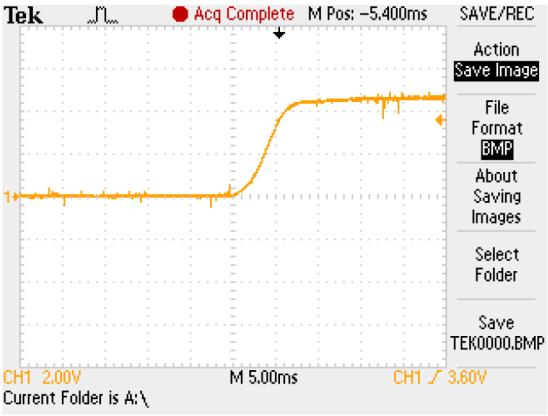

Fig. 12 *Actual step response of the SEPIC*

The values adopted for the components in the circuit are $V_{in} = 12\,\text{V}$, $L_1 = 330\,\mu\text{H}$, $L_2 = 470\,\mu\text{H}$, $C_1 = 1\,\text{mF}$, $C_2 = 330\,\mu\text{F}$, and $R = 10\,\Omega$. Let us consider that $V_{ref} = 7\,\text{V}$, so that $D = 0.37$ and $D' = 1 - D = 0.63$. The standard analysis of the SEPIC includes the transfer function modelling and the Simulink simulation. The transfer function can be obtained by using the following formula:

$$G(s) = \frac{D'(D + C_1L_2s^2)}{M_1s^4 + M_2s^3 + M_3s^2 + M_4s + D'^2}, \tag{15}$$

where $M_1 = L_1L_2C_1C_2$, $M_2 = L_1L_2C_1/R$, $M_3 = D'^2(L_1C_1 + L_1C_2 + L_2C_1 + L_2C_2)$, and $M_4 = (D^2L_1 + D'^2L_2)/R$. However, experiments with model (15) reveal that it is far from adequate and that its step response presents irregular oscillations. The step response obtained by Simulink is presented in Fig. 11, and the actual response of the SEPIC is shown in Fig. 12. We verify that the step response obtained by Simulink is quite different from the actual one. The reason for the discrepancy is that both the inductance and the capacitance have internal resistances that produce heat losses during operation. Moreover, the losses on the switch tube are proportional to the frequency and cannot be ignored. In addition, the welding process also increases the total resistance.

In order to establish an effective model of the SEPIC, the data of the step response in the experiment are fitted numerically to a third-order transfer function.

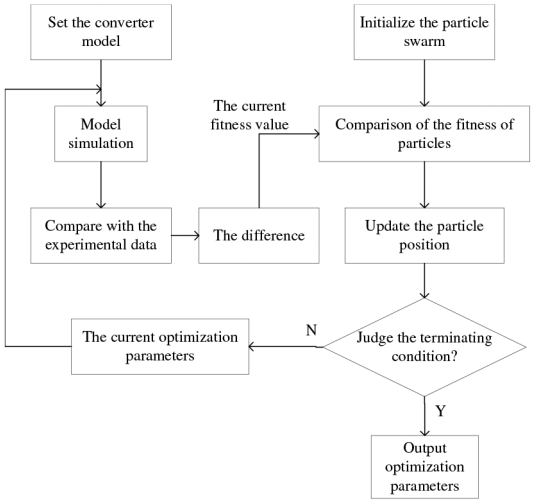

Fig. 13 *Flowchart of the model fit by means of the QPSO*

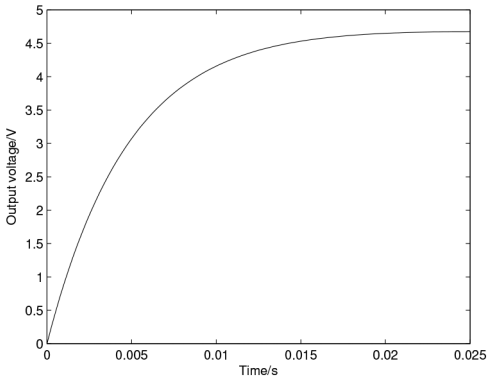

Fig. 14 *Step response of model (16)*

The QPSO algorithm is used to optimise the coefficients of the transfer function. The flowchart is shown in Fig. 13. In each iteration, the step response of the model is compared with the experimental data, and the difference is taken as the fitness value of the QPSO. After several iterations, a mathematical model of the SEPIC consistent with the experimental data is obtained. The optimised model is given by

$$G(s) = \frac{645.8265s^2 + 0.1353s + 10.0344}{4.5902s^3 + 945.4176s^2 + 1000s + 657.9755}. \tag{16}$$

The reliability of the model is calculated by using the following formula:

$$r = 1 - \left|\frac{\sum(t_n) - \sum(s_n)}{\sum(t_n)}\right|,$$

where t_n and s_n denote arrays of experimental and model data.

The results show that the reliability of the model is $r = 91.13\%$. The step response of model (16) is shown in Fig. 14.

4.2 Experiment

An experimental set-up is built to verify the performance of DFOPID and VCFOPID (Fig. 15). The control chip is TMS320F28335 and the switch tube drive chip is IR2125.

The switching frequency of the SEPIC circuit is 20 kHz, and the optimisation ranges of the control parameters are $K_p \in [0, 20]$, $K_i, K_d \in [0, 5]$ and $\lambda, \mu \in [0, 1]$. The control parameters of the DFOPID, obtained by using the QPSO, are $K_p = 20$, $K_i = 5$, $K_d = 5$, $\lambda = 0.67$, and $\mu = 0.2458$. The simulation and control inputs for the DFOPID and VCFOPID algorithms are depicted in Figs. 16 and 17, respectively.

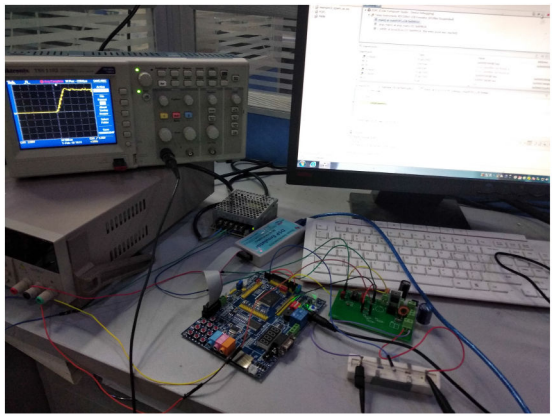

Fig. 15 *Experimental set-up*

For the DFOPID, the overshoot is 7.3% and the settling time is 1.8 ms, while for the VCFOPID the system response has no overshoot and the settling time is 0.2 ms. We conclude that the performance of the VCFOPID is much superior to the one exhibited by the DFOPID.

The control parameters obtained by the simulation are stored in the DSP, and the test responses of the two controllers are shown in Figs. 18 and 19, respectively.

In the experiment, it is observed that the overshoot of the DFOPID is 8.57% and the settling time is 3.5 ms. On the other hand, the settling time of the VCFOPID is about 1.3 ms. We verify that there are still some minor differences between the experimental and simulation results. One possible reason is the clamping action of the capacitance at both ends of the load resistor. The capacitor not only plays the role of filtering but also ensures that the voltage does not change suddenly. This is the main reason for the increase in settling time.

In synthesis, the experiment and simulation results in the SEPIC prove that the VCFOPID leads to better performance than the DFOPID.

5 Sensitivity analysis

The sensitivity analysis can make use of open loop control systems, closed loop control systems, and disturbances of control systems. As pointed out in [32], optimisation is a powerful tool for the design of controllers that can solve whatever criterion is formulated. Particularly, for proportional–integral–derivative (PID) or (FOPID) control, it is of key importance to introduce robustness constraints. This is often overlooked when using optimisation for PID or (FOPID) control. Indeed, making the closed-loop system stable is the minimal requirement for the controllers. However, in practice, it is not enough to require that the system is stable. There must also be some margins of stability to accommodate disturbances. In general, the maximum sensitivity can serve as a stability margin, telling the worst-case amplification of the disturbances [32], which is defined as

$$M_{\mathrm{s}} = \max_{\omega \in [0,\,+\infty)} \left| \frac{1}{GC(\mathrm{j}\omega)} \right|, \tag{17}$$

where $GC(\mathrm{j}\omega)$ is the loop transfer function.

In the following, the maximum sensitivity of the above examples and SEPIC will be calculated. The relation between the maximum sensitivity, M_{s}, and phase and gain margins, φ_m and g_m, is given by

$$\begin{aligned} g_m &\geq \frac{M_{\mathrm{s}}}{M_{\mathrm{s}} - 1}, \\ \varphi_m &\geq 2\arcsin\left(\frac{1}{2M_{\mathrm{s}}}\right). \end{aligned} \tag{18}$$

Table 5 shows the maximum sensitivity, phase margin (φ_m), and gain margin (g_m) of the above examples and SEPIC. One can find

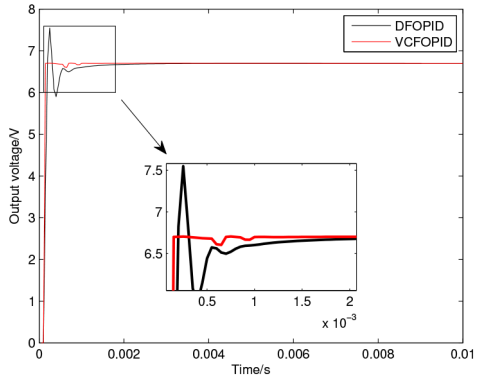

Fig. 16 *Step responses of model (16) for the VCFOPID and DFOPID controllers*

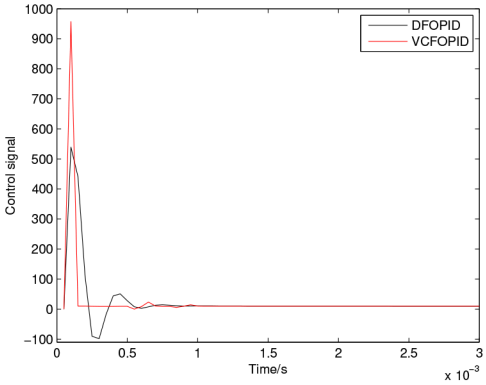

Fig. 17 *Control input signals for the DFOPID and VCFOPID algorithms of system (16)*

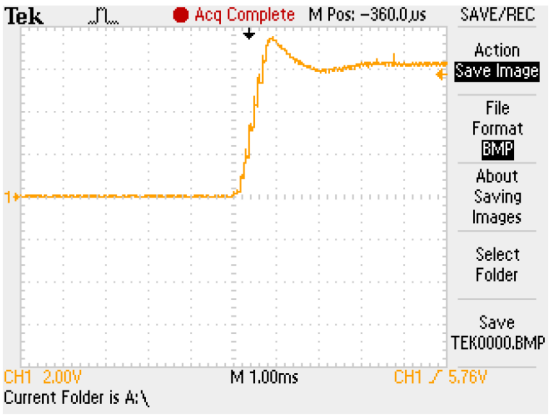

Fig. 18 *Test response of the SEPIC with the DFOPID controller*

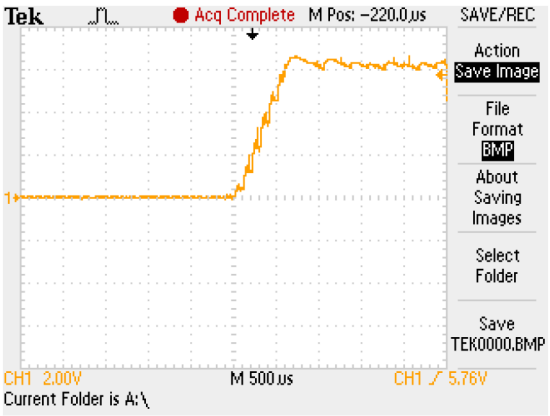

Fig. 19 *Test response of the SEPIC with the VCFOPID controller*

Table 5 Maximum sensitivity, phase margin (φ_m), and gain margin (g_m) of Examples 1–3 and SEPIC

	M_s	φ_m	g_m
Example 1	1.0319	≥ 57.9651°	≥ 32.3480
Example 2	1.0826	≥ 55.0132°	≥ 13.1065
Example 3	1.4035	≥ 41.7404°	≥ 3.4783
SEPIC	1.3901	≥ 42.1619°	≥ 3.5634

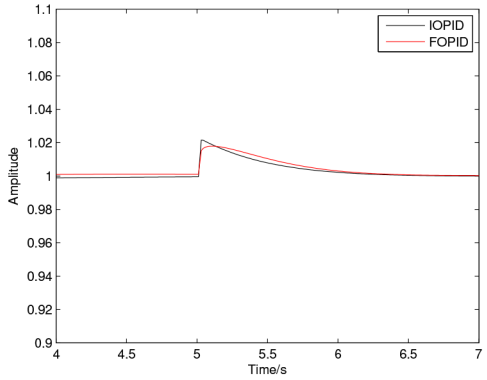

Fig. 20 *Disturbance responses for the system in Example 1 with FOPID and IOPID*

that the control system with the proposed controller has better stability and robustness.

Remark 1: The ITAE (5) is used in fractional-order (FO) controller design due to the good relation between absolute error and settling time. For the IOPID, arbitrarily small values of the cost function (5) can also be achieved by increasing the gain of the controller. However, this will result in poor interference rejection capability.

In fact, the comparison between the IOPID and FOPID should consider not only the optimisation index between input–output for a given (fixed) operating condition but also other aspects, such as robustness against parameter variations or external interference, after tuning to a given set point. To prevent the occurrence of controller parameters with an excessive value, the parameters of the (FO) PID controllers should be set to an acceptable interval beforehand in the optimisation process. To illustrate this issue, let us take Example 1. To be fair, the controller parameters optimisation range of the IOPID and FOPID are the same. The system response to a disturbance excitation (a pulse) at 5 s is simulated and the result is shown in Fig. 20. We verify that the disturbance attenuation achieved with the FOPID is superior to the one of the IOPIDs.

6 Conclusion

An improved VCFOPID was proposed in the study. In the first phase, the VCFOPID was compared with three known FO controllers under the light of several examples. The simulations showed that VCFOPID produces the best control results. In the second phase, a laboratory SEPIC was adopted. A QPSO algorithm performed a numerical fit between the experimental results and a new model of the SEPIC. The third-order transfer function revealed to be much more accurate than the standard mathematical model. Then, the VCFOPID was applied to the SEPIC. Both the simulation and experimental results confirm the good performance of the proposed controller.

7 Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments, which greatly improved the quality of this paper. This work was supported by the National Natural Science Funds of China (nos. 61403115 and 11971032).

8 References

[1] Jauregui, C., Mermoud, M.D., Lefranc, G., *et al.*: 'Conical tank level control with fractional PID', *IEEE Lat. Am. Trans.*, 2016, **14**, pp. 2598–2604
[2] Jesus, I.S., Machado, J.T.: 'Fractional control of heat diffusion systems', *Nonlinear Dyn.*, 2008, **54**, pp. 263–282
[3] Estakhrouiyeh, M.R., Vali, M., Gharaveisi, A.: 'Application of fractional order iterative learning controller for a type of batch bioreactor', *IET Control Theory Appl.*, 2016, **10**, pp. 1374–1383
[4] Zhong, J., Li, L.: 'Tuning fractional-order $PI^{\lambda}D^{\mu}$ controllers for a solid-core magnetic bearing system', *IEEE Trans. Control Syst. Technol.*, 2015, **23**, pp. 1648–1656
[5] Merrikh-Bayat, F., Mirebrahimi, N., Khalili, M.R.: 'Discrete-time fractional-order PID controller: definition, tuning, digital realization and some applications', *Int. J. Control Autom. Syst.*, 2015, **13**, pp. 81–90
[6] Gao, Z., Liao, X.: 'Discretization algorithm for fractional order integral by Haar wavelet approximation', *Appl. Math. Comput.*, 2011, **218**, pp. 1917–1926
[7] Machado, J.T.: 'Analysis and design of fractional-order digital control systems', *Syst. Anal. Model. Simul.*, 1997, **27**, pp. 107–122
[8] Machado, J.T., Galhano, A.M., Oliveira, A.M., *et al.*: 'Approximating fractional derivatives through the generalized mean', *Commun. Nonlinear Sci. Numer. Simul.*, 2009, **14**, pp. 3723–3730
[9] Aoun, M., Malti, R., Levron, F., *et al.*: 'Numerical simulations of fractional systems: an overview of existing methods and improvements', *Nonlinear Dyn.*, 2004, **38**, pp. 117–131
[10] Gao, Z., Liao, X.: 'Improved Oustaloup approximation of fractional-order operators using adaptive chaotic particle swarm optimization', *J. Syst. Eng. Electron.*, 2012, **23**, pp. 145–153
[11] Machado, J.T.: 'Optimal tuning of fractional controllers using genetic algorithms', *Nonlinear Dyn.*, 2010, **62**, pp. 447–452
[12] Freeborn, T.J., Maundy, B., Elwakil, A.S.: 'Field programmable analogue array implementation of fractional step filters', *IET Circuits Devices Syst.*, 2010, **4**, pp. 514–524
[13] El-Khazali, R.: 'On the biquadratic approximation of fractional-order Laplacian operators'. IEEE 56th Int. Midwest Symp. on Circuits and Systems (MWSCAS), Ohio State University, August 2013, pp. 04–07
[14] Wang, Z.B., Wang, Z.L., Cao, G.Y., *et al.*: 'Digital implementation of fractional order PID controller and its application', *J. Syst. Eng. Electron.*, 2005, **16**, pp. 116–122
[15] Li, R., Shan, L., Li, X., *et al.*: 'The digital implementation of fractional control algorithm on AC servo system', *Trans. China Electrotech. Soc.*, 2014, **29**, pp. 177–183
[16] Viola, J., Angel, L.: 'Identification, control and robustness analysis of a robotic system using fractional control', *IEEE Lat. Am. Trans.*, 2015, **13**, pp. 1294–1302
[17] Angel, L., Viola, J.: 'Design and statistical robustness analysis of FOPID, IOPID and SIMC PID controllers applied to a motor-generator system', *IEEE Lat. Am. Trans.*, 2015, **13**, pp. 3724–3734
[18] Viola, J., Angel, L.: 'Factorial design for robustness evaluation of fractional PID controllers', *IEEE Lat. Am. Trans.*, 2015, **13**, pp. 1286–1293
[19] Chen, L., Wu, R., Chu, Z., *et al.*: 'Pinning synchronization of fractional-order delayed complex networks with non-delayed and delayed couplings', *Int. J. Control*, 2017, **90**, (6), pp. 1245–1255
[20] Viola, J., Angel, L., Sebastian, J.M.: 'Design and robust performance evaluation of a fractional order PID controller applied to a DC motor', *IEEE/CAA J. Autom. Sin.*, 2017, **4**, pp. 304–314
[21] Sun, G., Ma, Z., Yu, J.: 'Discrete-time fractional order terminal sliding mode tracking control for linear motor', *IEEE Trans. Ind. Electron.*, 2018, **65**, pp. 3386–3394
[22] Balasubramaniam, P., Muthukumar, P., Ratnavelu, K.: 'Theoretical and practical applications of fuzzy fractional integral sliding mode control for fractional-order dynamical system', *Nonlinear Dyn.*, 2015, **80**, pp. 249–267
[23] Machado, J.T.: 'The effect of fractional order in variable structure control', *Comput. Math. Appl.*, 2012, **64**, pp. 3340–3350
[24] Maiti, S., Chakraborty, C., Hori, Y., *et al.*: 'Model reference adaptive controller-based rotor resistance and speed estimation techniques for vector controlled induction motor drive utilizing reactive power', *IEEE Trans. Ind. Electron.*, 2008, **55**, pp. 594–601
[25] Shyu, K.-K., Yang, M.-J., Chen, Y.-M., *et al.*: 'Model reference adaptive control design for a shunt active-power-filter system', *IEEE Trans. Ind. Electron.*, 2008, **55**, pp. 97–106
[26] Zamani, M., Karimi-Ghartemani, M., Sadati, N., *et al.*: 'Design of a fractional order PID controller for an AVR using particle swarm optimization', *Control Eng. Pract.*, 2009, **17**, pp. 1380–1387
[27] Bucanovic, L.J., Lazarevic, M.P., Batalov, S.N.: 'The fractional PID controllers tuned by genetic algorithms for expansion turbine in the cryogenic air separation process', *Hem. Ind.*, 2014, **68**, pp. 519–528

[28] Pires, E.S., Machado, J.T., de Moura Oliveira, P., *et al.*: 'Particle swarm optimization with fractional-order velocity', *Nonlinear Dyn.*, 2010, **61**, pp. 295–301
[29] Omkar, S.N., Khandelwal, R., Ananth, T.V.S., *et al.*: 'Quantum behaved particle swarm optimization (QPSO) for multi-objective design optimization of composite structures', *Expert Syst. Appl.*, 2009, **36**, pp. 11312–11322
[30] Yang, K., Maginu, K., Nomura, H.: 'Parameters identification of chaotic systems by quantum-behaved particle swarm optimization', *Int. J. Comput. Math.*, 2009, **86**, pp. 2225–2235
[31] Sun, J., Xu, W., Feng, B.: 'A global search strategy of quantum-behaved particle swarm optimization'. IEEE Conf. on Cybernetics and Intelligent Systems, Singapore, 2004, vol. 1, pp. 111–116
[32] Åström, K.J., Hägglund, T., Astrom, K.J.: '*Advanced PID control*' (ISA-The Instrumentation, Systems, and Automation Society, Research Triangle Park, North Carolina, 2006)