Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Design of a fractional $PI^{\lambda}D^{\mu}$ controller using the cohort intelligence method

Pritesh SHAH[†‡1], Sudhir AGASHE[2], Anand J. KULKARNI[1,3]

[1]*Symbiosis Institute of Technology, Symbiosis International University, Pune 412115, India*
[2]*Department of Instrumentation and Control Engineering, College of Engineering, Pune 411005, India*
[3]*Odette School of Business, University of Windsor, Windsor N9B3P4, Canada*
[†]E-mail: pritesh.ic@gmail.com

Received Aug. 23, 2016; Revision accepted Feb. 14, 2017; Crosschecked Mar. 10, 2018

Abstract: The cohort intelligence (CI) method has recently evolved as an optimization method based on artificial intelligence. We use the CI method for the first time to optimize the parameters of the fractional proportional-integral-derivative (PID) controller. The performance of the CI method in designing the fractional PID controller was validated and compared with those of some other popular algorithms such as particle swarm optimization, the genetic algorithm, and the improved electromagnetic algorithm. The CI method yielded improved solutions in terms of the cost function, computing time, and function evaluations in comparison with the other three algorithms. In addition, the standard deviations of the CI method demonstrated the robustness of the proposed algorithm in solving control problems.

Key words: Cohort intelligence; Fractional calculus; Fractional PID controller; Tuning
https://doi.org/10.1631/FITEE.1601495 **CLC number:** TP391

1 Introduction

The fractional proportional-integral-derivative (PID) controller exhibits a more robust and stable performance when compared with the classical PID controller (Petráš, 2008). However, it is a challenging task to tune the five parameters in the fractional PID controller, in which two more parameters need to be tuned compared with the classical PID controller. The parameter-tuning methods can be classified by three categories: analytical methods, numerical methods, and rule-based methods (Valério and Costa, 2010). Rule-based methods can be applied only in specific plants whose response is similar to an S-shape curve (Valério and Costa, 2006). Based on the S-shape response, the parameters of the controller can be calculated using the predefined rules. When using analytical methods, the parameters of the controller can be obtained by solving the designed equations (Zhao et al., 2005). However, these equations always need to be solved using numerical methods such as those used by Runge-Kutta (Luo et al., 2011). In numerical methods, a cost function based on the control signal is minimized using optimization approaches (Cao et al., 2005). Most plants can be optimally tuned by numerical methods, which is an advantage over the other methods (Biswas et al., 2009). A self-tuning approach has also been developed using different methods such as a relay-based method (Monje et al., 2008) and a method based on process parameters (Keyser et al., 2016).

So far, the fractional PID controller has been designed by the genetic algorithm (GA) (Cao et al., 2005; Chang and Chen, 2009), particle swarm optimization (PSO), the modified version of PSO (Karimi-Ghartemani et al., 2007; Aghababa, 2016), the artificial bee colony algorithm (Rajasekhar et al.,

‡ Corresponding author
ORCID: Pritesh SHAH, http://orcid.org/0000-0002-7504-2323

2014), the improved electromagnetism-like (EM) algorithm (Lee and Chang, 2010), and improved differential evolution (DE) (Biswas et al., 2009). However, most of these methods cannot find the optimal global solution for most plants. In this paper, an emerging meta-heuristic method termed the cohort intelligence (CI) algorithm is applied to optimize the parameters in the fractional PID controller. The CI algorithm proposed by Kulkarni et al. (2013) has been used to solve several unconstrained test problems (Kulkarni AJ et al., 2013, 2017) and to deal with clustering test cases. It has also been used in various mechanical applications (Kulkarni O et al., 2016; Dhavle et al., 2017). In addition, it has been applied to solve the combinatorial problems such as the 0-1 knapsack problem (Kulkarni and Shabir, 2016) and large sized combinatorial problems in the health care domain, the multiple knapsack problem, and the cross-border shippers problem (Kulkarni AJ et al., 2016). The CI algorithm was inspired by the social behavior of cohort candidates. A cohort refers to a group of candidates which are interacting and competing with one another to achieve a common goal. In a cohort, each candidate iteratively attempts to improve its own behavior by observing the behavior of other candidates. Eventually, when all the candidates converge on a particular behavior, the learning procedure is terminated and the convergence/saturation condition is reached. Details about the CI algorithm can be found in Kulkarni AJ et al. (2016).

The novelty of this study is the application of the CI algorithm to design the fractional PID controllers for solving different benchmarks (Cao and Cao, 2006; Biswas et al., 2009; Lee and Chang, 2010). The control performance of the proposed method is compared with those of recent studies in terms of the cost function, the number of function counts, and the computation time. Besides, the closed-loop unit step responses of the proposed method are plotted for all plants.

2 Fractional calculus and the fractional PID controller

2.1 Fractional calculus

Although fractional calculus is predating classical calculus by more than 300 years, it has rarely been appreciated from a research point of view (De and Sen, 2011). However, over the last few decades some researchers have been exploring the potential applications of fractional calculus in different areas including control systems, speech signal processing, process modeling, chaos, and fractals (Das, 2011).

In fractional calculus, the differentiation-integration operator ${}_aD_t^\alpha$ (Chen et al., 2009) is defined as follows:

$$ {}_aD_t^\alpha = \begin{cases} \frac{\mathrm{d}^\alpha}{\mathrm{d}x^\alpha}, & \alpha > 0, \\ 1, & \alpha = 0, \\ \int_a^t (\mathrm{d}\tau)^{-\alpha}, & \alpha < 0. \end{cases} \tag{1} $$

The relevant definitions about fractional calculus are presented below.

2.1.1 Caputo definition

The caputo definition, which has been widely used in the engineering domain (Abramowitz, 1972; Cafagna, 2007; Monje et al., 2010), offers a straightforward relationship between the type of initial conditions and the fractional operator. The caputo definition is

$$ {}_aD_t^\alpha = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f^n(\tau)}{(t-\tau)^{\alpha-n+1}} \mathrm{d}\tau, \tag{2} $$

where n is an integer with $(n-1) \leq \alpha \leq n$ and α is a real number. For instance, if α is 0.52, then n would be 1 because $0 \leq 0.52 \leq 1$. The gamma function is defined as

$$ \Gamma(x) = \int_0^\infty t^{x-1} \mathrm{e}^{-t} \mathrm{d}t, \ \Re(x) > 0. \tag{3} $$

2.1.2 Riemann-Liouville definition

The Riemann-Liouville fractional definition is given by

$$ \begin{aligned} {}_aD_t^\alpha &= D^n J^{n-\alpha} f(t) \\ &= \frac{1}{\Gamma(n-\alpha)} \left(\frac{\mathrm{d}}{\mathrm{d}t}\right)^n \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} \mathrm{d}\tau, \end{aligned} \tag{4} $$

where n is an integer with $(n-1) \leq \alpha \leq n$, α is a real number, J is the integral operator, and a and t are the limits of integration.

2.1.3 Grunwald-Letnikov definition

The Grunwald-Letnikov definition is defined as

$$ {}_aD_t^\alpha = \lim_{h \to 0} \frac{1}{h^\alpha} \sum_{r=0}^{\left[\frac{t-a}{h}\right]} (-1)^r \binom{n}{r} f(t-rh), \tag{5} $$

where $[\frac{t-a}{h}]$ is an integer, h is the step size for differentiation, and a and t are the limits of differentiation.

2.2 Fractional PID controller

Classical PID controllers have been used for many decades in industry to solve the control problems. The main reason for their popularity is the simplicity of their design and their good performance (Astrom and Hagglund, 1995; Das et al., 2012). However, a PID controller can be extended to a fractional PID controller to improve its quality and robustness (Podlubny, 1999). A classical PID controller has the following transfer function (Anwar and Pan, 2014):

$$C(s) = K_{\mathrm{P}} + \frac{K_{\mathrm{I}}}{s} + K_{\mathrm{D}}s, \tag{6}$$

where $C(s)$ is the controller transfer function, K_{P} is the proportional constant gain, K_{I} is the integration constant gain, and K_{D} is the derivative constant gain.

A fractional-order controller was introduced for fractional order systems (Podlubny, 1994; Podlubny et al., 1997; Podlubny, 1999). It is an application of fractional calculus, which is also as old as classical calculus. The advantage of this controller is that it is less sensitive to the changes in the variables of a control system and the parameters of a controller (Luo et al., 2010; Malek et al., 2013). A fractional PID controller can also achieve the iso-damping property quickly and more robust than other classical controllers (Podlubny, 1994; Mishra et al., 2015). A block diagram of a fractional PID controller is shown in Fig. 1. It has the following structure (Shah et al., 2013; Barbosa et al., 2007):

$$\mathrm{GC}(s) = \frac{U(s)}{E(s)} = K_{\mathrm{P}} + \frac{K_{\mathrm{I}}}{s^{\lambda}} + K_{\mathrm{D}}s^{\mu},\ \lambda,\ \mu \geq 0. \tag{7}$$

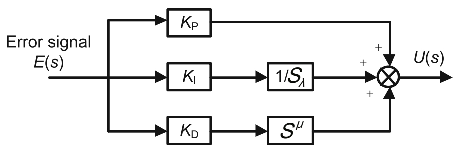


Fig. 1 Block diagram of the fractional PID controller

When $\lambda = \mu = 1$, the fractional PID controller becomes a classical PID controller (Fig. 2). Therefore, a fractional PID controller can be seen as a generalized version of a classical PID controller (Tang et al., 2012). The fractional PID controller has much more flexibility so that real systems can be controlled more accurately. The FOMCON toolbox is used for the simulation in this study (Tepljakov et al., 2011, 2013). Shah and Agashe (2016) gave a detailed review of fractional PID controllers. Valério and Costa (2010) and Shah and Agashe (2016) also introduced the details about different tuning methods.

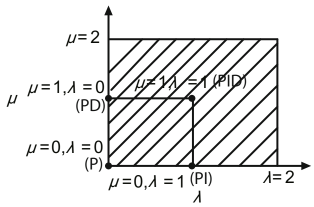


Fig. 2 The fractional PID controller plane

3 Framework of cohort intelligence

The cohort intelligence (CI) method is used to improve the closed-loop responses of various plants by minimizing the defined error function. In the CI method, the behavior can be improved through the interaction and competition amongst its candidates. The framework of the CI method (Kulkarni O et al., 2016) can be summarized as follows:

Step 1: The parameters of the controller are considered qualities of candidates in the process of optimization. These qualities are randomly generated within the upper and lower bounds of the search space.

Step 2: The behavior is evaluated for all the candidates, and is referred to as the cost function. In the case of the control system, the cost function is the error function to be minimized.

Step 3: The probability for each candidate followed by other candidates is evaluated based on its cost function. The candidate with the best cost function has the maximum probability of being followed by other candidates, and vice versa.

Step 4: Each candidate employs a roulette wheel approach to follow a behavior in the cohort, and it improves the value of its cost function by expanding or reducing the lower and upper bounds of the parameters.

Step 5: The value of the cost function is assumed to converge when the difference between the behavior of each candidate becomes insignificant.

4 Cohort intelligence for solving a transfer function

The process of CI optimization begins with the selection of candidates in cohort C, reduction interval factor $r \in [0,1]$, convergence parameter ε, and maximum iteration number $L_{\max}$. The values of C, r, ε, and $L_{\max}$ are selected based on preliminary runs of the algorithm. The lower and upper bounds of optimization variables (K_{P}, K_{I}, λ, K_{D} and μ) for plant 1 are chosen as follows (Cao and Cao, 2006): $\boldsymbol{\Psi}_{K_{\mathrm{P}}}=[0,10]$, $\boldsymbol{\Psi}_{K_{\mathrm{I}}}=[0,10]$, $\boldsymbol{\Psi}_{\lambda}=[0,2]$, $\boldsymbol{\Psi}_{K_{\mathrm{D}}}=[0,10]$, $\boldsymbol{\Psi}_{\mu}=[0,2]$. A fractional PID controller $\mathrm{GC}(s)$ is also chosen to design the plant transfer function $G(s)$.

The following steps describe the implementation of the CI method to optimize the fractional PID controller:

Step 1: Each candidate c ($c=1,2,\cdots,C$) generates its qualities (K_{P}, K_{I}, λ, K_{D}, and μ) within their associated sampling intervals ($\boldsymbol{\Psi}_{K_{\mathrm{P}}},\boldsymbol{\Psi}_{K_{\mathrm{I}}},\boldsymbol{\Psi}_{\lambda},\boldsymbol{\Psi}_{K_{\mathrm{D}}}$, and $\boldsymbol{\Psi}_{\mu}$) as follows:

$$K_{\mathrm{P}}^{c}=\min(\boldsymbol{\Psi}_{K_{\mathrm{P}}})+\big[\max(\boldsymbol{\Psi}_{K_{\mathrm{P}}})-\min(\boldsymbol{\Psi}_{K_{\mathrm{P}}})\big]\cdot\mathrm{rand}(\cdot), \tag{8}$$

where $\mathrm{rand}(\cdot)\in[0,1]$ is a random number generated between 0 and 1. The equations of other qualities are similar.

Step 2: The associated behavior of each candidate c ($c=1,2,\cdots,C$), i.e., the integral square error (ISE), is defined as follows:

$$\mathrm{ISE}^{c}=\int_{0}^{t}e^{2}(t)\mathrm{d}t, \tag{9}$$

where $e(t)$ is the error signal and is given for the unity feedback system with a step input signal:

$$e(t)=1-L^{-1}\left(\frac{1}{s}\frac{G(s)\mathrm{GC}(s)}{1+G(s)\mathrm{GC}(s)}\right). \tag{10}$$

Note that $L^{-1}\{F(s)\}$ represents the inverse Laplace transform of $F(s)$.

Step 3: Every candidate evaluates the probability of its behavior as follows:

$$P^{c}=\frac{1/\mathrm{ISE}^{c}}{\sum_{c=1}^{C}1/\mathrm{ISE}^{c}},\quad c=1,2,\cdots,C. \tag{11}$$

The probability of each candidate helps select the best solution.

Step 4: Each candidate decides to follow another behavior ISE^{c} and its associated qualities using the roulette wheel approach. The roulette wheel method is used to select a candidate from one generation to create the basis of the next generation. The better the solution, the higher the probability of being followed by the other candidates in the cohort.

Step 5: Each candidate reduces the sampling interval $\boldsymbol{\Psi}^{c}$ associated with each of its qualities (K_{P}, K_{I}, λ, K_{D}, and μ) as follows:

$$\boldsymbol{\Psi}_{K_{\mathrm{P}}}^{c}\in\Bigg[K_{\mathrm{P}}^{c}-\parallel(\max(\boldsymbol{\Psi}_{K_{\mathrm{P}}})-\min(\boldsymbol{\Psi}_{K_{\mathrm{P}}}))\frac{r}{2}\parallel, K_{\mathrm{P}}^{c}+\parallel(\max(\boldsymbol{\Psi}_{K_{\mathrm{P}}})-\min(\boldsymbol{\Psi}_{K_{\mathrm{P}}}))\frac{r}{2}\parallel\Bigg]. \tag{12}$$

The other equations for the rest parameters are similar.

Step 6: The cohort behavior can be considered saturated if there is no significant improvement in the behavior ISE^{c} of any candidate c ($c=1,2,\cdots,C$) in the cohort. The saturated condition is defined as follows:

$$\parallel\max(\mathrm{ISE}^{c})^{n}-\max(\mathrm{ISE}^{c})^{n-1}\parallel\leq\varepsilon, \tag{13}$$

$$\parallel\min(\mathrm{ISE}^{c})^{n}-\min(\mathrm{ISE}^{c})^{n-1}\parallel\leq\varepsilon, \tag{14}$$

$$\parallel\max(\mathrm{ISE}^{c})^{n}-\min(\mathrm{ISE}^{c})^{n}\parallel\leq\varepsilon. \tag{15}$$

Step 7: If neither of the two criteria listed below is satisfied, continue to step 1:

1. The maximum number of attempts $L_{\max}$ is exceeded.

2. Cohort saturates as explained in Eqs. (13)–(15).

Select any of the C behaviors from the current set of behaviors for the final solution. The description about the formulation of the CI method can be found in Kulkarni AJ et al. (2013, 2016).

5 Results and discussion

The CI algorithm discussed above was coded in MATLAB R2013B on an Intel Core i7 processor with 4 GB RAM and the Windows 7 Professional Operating system. The CI parameters are listed in Table 1. Each transfer function associated with each plan was solved 20 times to demonstrate the robustness of the CI algorithm. In the simulations, the transfer functions are shown in Table 2. The sampling intervals of the controller parameters are illustrated in Table 3.

These values were selected from the reference papers given in Table 2. The interval can be selected based on the knowledge and an initial guess. The interval for fractional order was considered between 0 and 2 because of stability issues for more than two orders. The r value was selected based on a trial-and-error method.

Table 1 Parameters for the CI algorithm

Parameter	Value
Number of cohort candidates C	2 to 5
Number of variables N	5
Discount factor r	0.97
Saturation/convergence constant ε	0.0001

Table 2 Transfer functions used for simulation

Plant	Transfer function	Reference
$P_1(s)$	$\frac{400}{s^2+50s}$	Cao and Cao (2006), Lee and Chang (2010)
$P_2(s)$	$\frac{1}{s^3+6s^2+7s}$	Aghababa (2016)
$P_3(s)$	$\frac{s^2+2s}{s^4+2s^3+5s^2+s+0.1}$	Aghababa (2016)
$P_4(s)$	$\frac{1}{s^2+2s}$	Aghababa (2016)

Table 3 Sampling intervals of controller parameters

Parameter	Value			
	Plant 1	Plant 2	Plant 3	Plant 4
K_{P}	[0, 10]	[0, 500]	[0, 500]	[0, 500]
K_{I}	[0, 10]	[0, 500]	[0, 500]	[0, 500]
λ	[0, 2]	[0, 2]	[0, 2]	[0, 2]
K_{D}	[0, 10]	[0, 500]	[0, 500]	[0, 500]
μ	[0, 2]	[0, 2]	[0, 2]	[0, 2]

As suggested in Cao and Cao (2006), Biswas et al. (2009), and Lee and Chang (2010), the integral square error (ISE) was considered the benchmark for plant 1. For plants 2, 3, and 4, the performance index J of time-domain specifications defined in Eq. (16) was used:

$$J=\frac{1}{1+\mathrm{e}^{-\alpha}}(T_{\mathrm{r}}+T_{\mathrm{ss}})+\frac{\mathrm{e}^{-\alpha}}{1+\mathrm{e}^{-\alpha}}(M_{\mathrm{P}}+E_{\mathrm{ss}}). \quad (16)$$

If α is set to zero, the weights of all specifications would be the same. By deepening the α value, the effect of the parameters can be changed. In the simulation, α was set to zero.

Results for plant 1 are shown in Table 4. Using the CI method, the cost function ISE was reduced to 0.0051. Using the other methods, the values of ISE were 0.0971 and 0.0811, respectively. There was an improvement of 93% in the ISE value compared with the other methods. The CI method yielded not only a better ISE value but also a lower computation time, compared with that produced by the EM algorithm (2.5055 s for CI vs. 4.107 s in the EM algorithm (Lee and Chang, 2010)), which was approximately 39% faster. The closed-loop response of plant 1 has the following time-domain specifications: rise time (T_{r}), 0.0012 s; maximum overshoot (M_{p}), 0%; peak time (T_{p}), 0.0239 s; settling time (T_{ss}), 0.2 s; and steady state error (E_{ss}), 0.

The CI method was also compared with the recent optimization method used by Aghababa (2016). The transfer functions of plants 2, 3, 4 were taken from Aghababa (2016). For these plants, the cost function composed of rise time, settling time, maximum overshoot, and the steady state error in Eq. (16) was considered. Results are shown in Table 6. The results of the cost function and function count also exhibited the superiority of the CI method over the other methods. The function count was reduced by 42% for plant 4, 38% for plant 3, and 18% for plant 2. The value of the cost function was reduced by 98% for plant 3, 83% for plant 4, and 69% for plant 2. The computation time required for CI was 8.8173 s, 11.0082 s, and 12.0895 s for plants 4, 2, and 3, respectively.

Step responses of the closed-loop system after optimization by the CI method are shown in Figs. 3 and 4. The responses of GA and PSO methods are also shown with the same initial conditions in Figs. 3 and 4. The results demonstrated that the CI method performs better than the other methods. Tables 5 and 6 show that the CI method has an improved performance in terms of the function count, computation time, and cost function. From Figs. 3 and 4, it can be concluded that the CI method yields better transient and steady-state time responses. The step response shows that the plants are settling quickly (within 1 s) with significantly less overshoot (below 1%). The convergence plane for plant 3 is shown in Fig. 5, exhibiting the self-supervising learning behavior of all candidates (Kulkarni AJ et al., 2013). Similarly, the convergence planes for the controller parameters for one of the trials are shown in Fig. 6. The control-and-error signals for plant 2 are shown in Fig. 7.

The robustness of the fractional PID controller

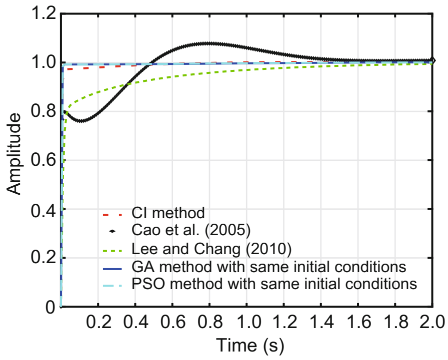
Table 4 Comparative results for plant 1

Algorithm	K_P	K_I	λ	K_D	μ	ISE
Cao et al. (2005)	1.6230	1.1908	0.8190	0.1135	1.5146	0.0971
Lee and Chang (2010)	1.0825	0.0477	0.9275	0.6317	0.8961	0.0811
CI method	4.7391	7.4476	0.3226	3.6877	1.0502	0.0051

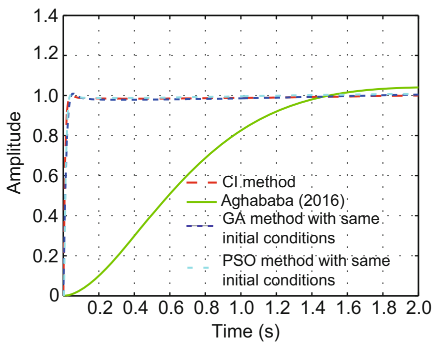
ISE: integral square error

Table 5 Comparative results for plants 2, 3, and 4

System	Percentage of performance improvement	
	Function count	Cost function
Plant 2	18%	69%
Plant 3	38%	98%
Plant 4	42%	83%



(a)

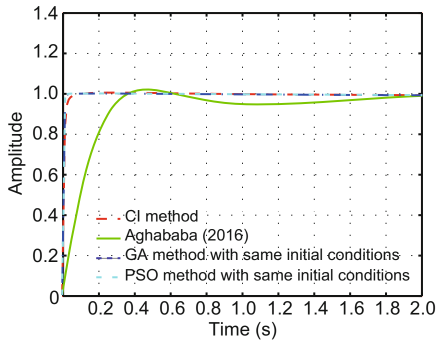


(b)

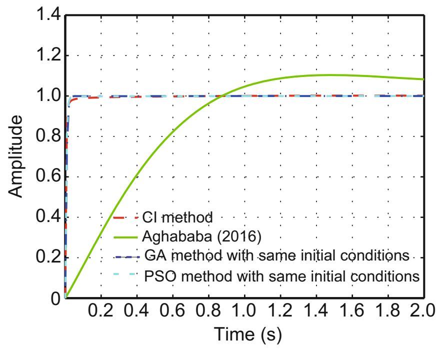
Fig. 3 Step response of the closed-loop system designed using CI: (a) plant 1; (b) plant 2

was validated by changing the value of the plant gain. By changing the K value, the closed-loop response remained almost unchanged. Fig. 7 also confirms that the fractional PID controller can achieve the iso-damping property much more easily compared with a classical controller. Fig. 8 presents the closed-loop responses of plant 1 with different plant gain values.

Each transfer function associated with each plant was solved 20 times using CI. The best, worst,



(a)



(b)

Fig. 4 Step response of the closed-loop system designed using CI : (a) plant 3; (b) plant 4

and mean of candidates, standard deviation (σ), average computation time, and average function evaluations are presented in Table 7 for four benchmarks. The results demonstrated that the fractional PID controller based on the CI method exhibits a robust performance.

6 Conclusions and future directions

It is the first time to use the emerging cohort intelligence (CI) method in the control domain to design a fractional PID controller. Simulations demonstrated that compared with other optimization methods (PSO, modified version of PSO, and EM), the CI method yielded better results when used to tune the parameters of the fractional PID

Table 6 Comparative results for plants 2, 3, and 4

Parameter		Value		
		Plant 2	Plant 3	Plant 4
Fractional PID controller parameters	$K_{\rm P}$	489.2877	345.3677	146.2190
	$K_{\rm I}$	75.3326	366.8687	349.4606
	λ	0.1115	0.1331	0.2177
	$K_{\rm D}$	484.6469	158.0005	232.0830
	μ	1.8262	1.2468	1.2984
Time domain parameters	$M_{\rm P}$(%)	0.9744	0.2128	0.1797
	$T_{\rm r}$(s)	0.0200	0.0400	0.0300
	$T_{\rm ss}$(s)	0.0400	0.0700	0.1200
	$E_{\rm ss}$	0.0003	0.0017	0.0000
J using CI		0.5173	0.1622	0.1648
J using the method proposed by Aghababa (2016)		1.6834	10.1812	1.0125
Function count using CI		130	105	105
Function count using the method proposed by Aghababa (2016)		158	169	183
Computation time using CI (s)		11.0082	12.0895	8.8173

Table 7 Performance of the CI algorithm

Parameter	Value			
	Plant 1	Plant 2	Plant 3	Plant 4
Best of candidates	0.005	0.4854	0.0112	0.0466
Mean of candidates	0.005 12	0.974 33	0.063 525	0.117 385
Worst of candidates	0.0052	1.9277	0.1893	0.2196
Standard deviation (σ)	6.15587E-05	0.385 95	0.058 958 95	0.049 338 01
Average computation time (s)	2.601 655	9.782 83	9.789 26	6.837 085
Average function evaluations	30	129	92	90

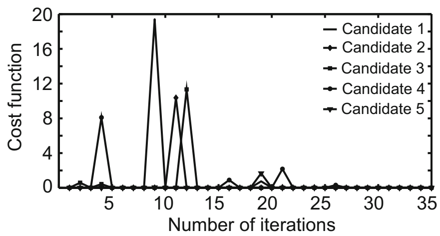


Fig. 5 Convergence plot of the candidates for $P_3(s)$

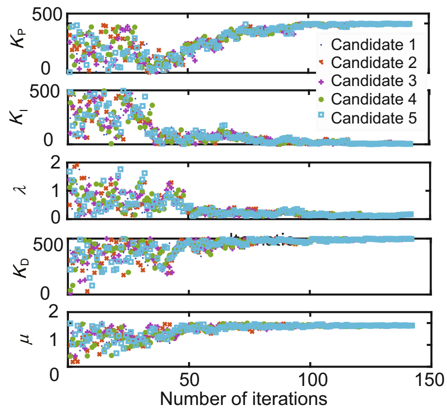
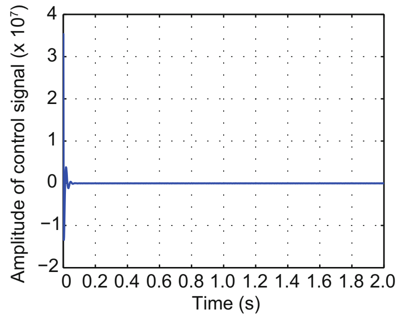


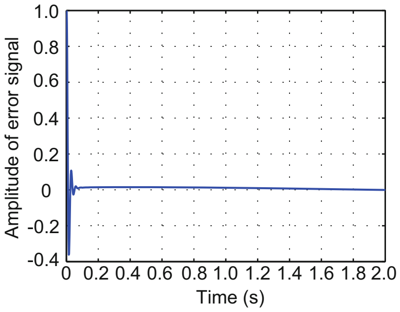
Fig. 6 Convergence graph for controller parameters

controller. For plant 1, the cost function and computation time were reduced by 93% and 39%, respectively. For plants 2, 3, and 4, the cost function value and functional count were reduced by up to 98% and 42%, respectively. The standard deviation of the CI method is very low, which indicates the robustness of the fractional PID controller. It can be concluded that the CI method provides the optimal closed-loop system performance for a fractional PID controller. It produces better response from various control systems, and benefits from the production of improved responses for a closed-loop system, which means that less control effort is required. The CI method converges towards the best values of the controller for different plants of a control system.

The method can be further applied for nonlinear and non-integer plant systems. In addition, it can be used for parameter estimations of integer and non-integer model structures for system identification (Sabatier et al., 2006; Li et al., 2016). Furthermore, the CI method can be extended to solve multi-objective problems such as those presented by Meng and Xue (2009) and Pan and Das (2012, 2015).



(a)



(b)

Fig. 7 Control signal (a) and error signal (b) for plant 2

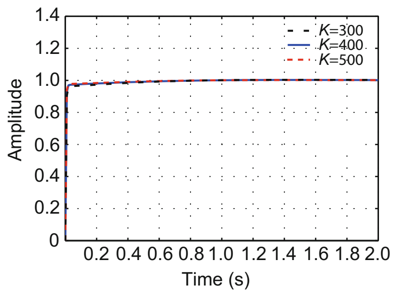


Fig. 8 Robustness of the FO-PID controller by changing *K*

References

Abramowitz M, 1972. Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables. Dover Publications Incorporated.

Aghababa MP, 2016. Optimal design of fractional-order PID controller for five bar linkage robot using a new particle swarm optimization algorithm. *Soft Comput*, 20(10):4055-4067. https://doi.org/10.1007/s00500-015-1741-2

Anwar MN, Pan S, 2014. A frequency domain design of PID controller for an AVR system. *J Zhejiang Univ-Sci C (Comput & Electron)*, 15(4):293-299. https://doi.org/10.1631/jzus.C1300218

Astrom KJ, Hagglund T, 1995. PID Controllers: Theory, Design, and Tuning (2^{nd} Ed.). The Instrument, Systems and Automation Society, Research Triangle Park, NC.

Barbosa RS, Machado JAT, Galhano AM, 2007. Performance of fractional PID algorithms controlling nonlinear systems with saturation and backlash phenomena. *J Vibr Contr*, 13(9-10):1407-1418. https://doi.org/10.1177/1077546307077499

Biswas A, Das S, Abraham A, et al., 2009. Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution. *Eng Appl Artif Intell*, 22(2):343-350. https://doi.org/10.1016/j.engappai.2008.06.003

Cafagna D, 2007. Fractional calculus: a mathematical tool from the past for present engineers. *IEEE Ind Electron Mag*, 1(2):35-40. https://doi.org/10.1109/MIE.2007.901479

Cao JY, Cao BG, 2006. Design of fractional order controllers based on particle swarm optimization. *Int J Contr Autom Syst*, 4(6):775-781.

Cao JY, Liang J, Cao BG, 2005. Optimization of fractional order PID controllers based on genetic algorithms. Int Conf on Machine Learning and Cybernetics, p.5686-5689. https://doi.org/10.1109/ICMLC.2005.1527950

Chang LY, Chen HC, 2009. Tuning of fractional PID controllers using adaptive genetic algorithm for active magnetic bearing system. *WSEA Trans Syst*, 8(1):158-167.

Chen YQ, Petras I, Xue DY, 2009. Fractional order control—a tutorial. American Control Conf, p.1397-1411. https://doi.org/10.1109/ACC.2009.5160719

Das S, 2011. Functional Fractional Calculus. Springer Berlin Heidelberg, p.51-100. https://doi.org/10.1007/978-3-642-20545-3

Das S, Pan I, Das S, et al., 2012. A novel fractional order fuzzy pid controller and its optimal time domain tuning based on integral performance indices. *Eng Appl Artif Intell*, 25(2):430-442. https://doi.org/10.1016/j.engappai.2011.10.004

De A, Sen S, 2011. Root locus method for any fractional order commensurate system. IEEE Students' Technology Symp, p.323-328. https://doi.org/10.1109/TECHSYM.2011.5783837

Dhavle SV, Kulkarni AJ, Shastri A, et al., 2017. Design and economic optimization of shell-and-tube heat exchanger using cohort intelligence algorithm. *Neur Comput Appl*, in press. https://doi.org/10.1007/s00521-016-2683-z

Karimi-Ghartemani M, Zamani M, Sadati N, et al., 2007. An optimal fractional order controller for an AVR system using particle swarm optimization algorithm. Large Engineering Systems Conf on Power Engineering, p.244-249. https://doi.org/10.1109/LESCPE.2007.4437386

Keyser DR, Muresan CI, Ionescu CM, 2016. A novel autotuning method for fractional order PI/PD controllers. *ISA Trans*, 62:268-275. https://doi.org/10.1016/j.isatra.2016.01.021

Kulkarni AJ, Shabir H, 2016. Solving 0-1 knapsack problem using cohort intelligence algorithm. *Int J Mach Learn Cybern*, 7(3):427-441. https://doi.org/10.1007/s13042-014-0272-y

Kulkarni AJ, Durugkar IP, Kumar M, 2013. Cohort intelligence: a self supervised learning behavior. IEEE Int Conf on Systems, Man and Cybernetics, p.1396-1400. https://doi.org/10.1109/SMC.2013.241

Kulkarni AJ, Baki MF, Chaouch BA, 2016. Application of the cohort-intelligence optimization method to three selected combinatorial optimization problems. *Eur J Oper Res*, 250(2):427-447. https://doi.org/10.1016/j.ejor.2015.10.008

Kulkarni AJ, Krishnasamy G, Abraham A, 2017. Cohort Intelligence: a Socio-inspired Optimization Method. Springer, Cham. https://doi.org/10.1007/978-3-319-44254-9

Kulkarni O, Kulkarni N, Kulkarni AJ, et al., 2016. Constrained cohort intelligence using static and dynamic penalty function approach for mechanical components design. *Int J Parall Emerg Distr Syst*, p.1-19. https://doi.org/10.1080/17445760.2016.1242728

Lee CH, Chang FK, 2010. Fractional-order PID controller optimization via improved electromagnetism-like algorithm. *Expert Syst Appl*, 37(12):8871-8878. https://doi.org/10.1016/j.eswa.2010.06.009

Li CS, Chang L, Huang ZJ, et al., 2016. Parameter identification of a nonlinear model of hydraulic turbine governing system with an elastic water hammer based on a modified gravitational search algorithm. *Eng Appl Artif Intell*, 50:177-191. https://doi.org/10.1016/j.engappai.2015.12.016

Luo Y, Chen YQ, Wang CY, et al., 2010. Tuning fractional order proportional integral controllers for fractional order systems. *J Process Contr*, 20(7):823-831. https://doi.org/10.1016/j.jprocont.2010.04.011

Luo Y, Chao HC, Di LP, et al., 2011. Lateral directional fractional order PI control of a small fixed-wing unmanned aerial vehicles: controller designs and flight tests. *IET Contr Theory Appl*, 5(18):2156-2167. https://doi.org/10.1049/iet-cta.2010.0314

Malek H, Luo Y, Chen YQ, 2013. Identification and tuning fractional order proportional integral controllers for time delayed systems with a fractional pole. *Mechatronics*, 23(7):746-754. https://doi.org/10.1016/j.mechatronics.2013.02.005

Meng L, Xue DY, 2009. Design of an optimal fractional-order PID controller using multi-objective GA optimization. Chinese Control and Decision Conf, p.3849-3853. https://doi.org/10.1109/CCDC.2009.5191796

Mishra P, Kumar V, Rana KPS, 2015. A fractional order fuzzy PID controller for binary distillation column control. *Expert Syst Appl*, 42(22):8533-8549. https://doi.org/10.1016/j.eswa.2015.07.008

Monje CA, Vinagre BM, Feliu V, et al., 2008. Tuning and auto-tuning of fractional order controllers for industry applications. *Contr Eng Pract*, 16(7):798-812. https://doi.org/10.1016/j.conengprac.2007.08.006

Monje CA, Chen YQ, Vinagre BM, et al., 2010. Fractional-Order Systems and Controls: Fundamentals and Applications. Springer, London. https://doi.org/10.1007/978-1-84996-335-0

Pan I, Das S, 2012. Chaotic multi-objective optimization based design of fractional order $PI^{\lambda}D^{\mu}$ controller in AVR system. *Int J Electr Power Energy Syst*, 43(1):393-407. https://doi.org/10.1016/j.ijepes.2012.06.034

Pan I, Das S, 2015. Fractional-order load-frequency control of interconnected power systems using chaotic multi-objective optimization. *Appl Soft Comput*, 29:328-344. https://doi.org/10.1016/j.asoc.2014.12.032

Petráš I, 2008. Stability of fractional-order systems with rational orders. *Fract Calc Appl Anal*, 10(3):269-298.

Podlubny I, 1994. Fractional-order systems and fractional-order controllers. Proc Conf on Int Francophone d'Automatique, p.43-54.

Podlubny I, 1999. Fractional Differential Equations: an Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications. Academic Press, San Diego.

Podlubny I, Dorcak L, Kostial I, 1997. On fractional derivatives, fractional-order dynamic systems and $PI^{\lambda}D^{\mu}$-controllers. IEEE Conf on Decision Control, p.4985-4990. https://doi.org/10.1109/CDC.1997.649841

Rajasekhar A, Jatoth RK, Abraham A, 2014. Design of intelligent PID/$PI^{\lambda}D^{\mu}$ speed controller for chopper fed DC motor drive using opposition based artificial bee colony algorithm. *Eng Appl Artif Intell*, 29:13-32. https://doi.org/10.1016/j.engappai.2013.12.009

Sabatier J, Aoun M, Oustaloup A, et al., 2006. Fractional system identification for lead acid battery state of charge estimation. *Signal Process*, 86(10):2645-2657. https://doi.org/10.1016/j.sigpro.2006.02.030

Shah P, Agashe S, 2016. Review of fractional PID controller. *Mechatronics*, 38:29-41. https://doi.org/10.1016/j.mechatronics.2016.06.005

Shah P, Agashe SD, Singh AP, 2013. Design of fractional order controller for undamped control system. Nirma University Int Conf on Engineering, p.1-5. https://doi.org/10.1109/NUiCONE.2013.6780171

Tang YG, Cui MY, Hua CC, et al., 2012. Optimum design of fractional order $PI^{\lambda}D^{\mu}$ controller for AVR system using chaotic ant swarm. *Expert Syst Appl*, 39(8):6887-6896. https://doi.org/10.1016/j.eswa.2012.01.007

Tepljakov A, Petlenkov E, Belikov J, 2011. FOMCON: fractional-order modeling and control toolbox for Matlab. Int Conf on Mixed Design of Integrated Circuits and Systems, p.684-689.

Tepljakov A, Petlenkov E, Belikov J, et al., 2013. Fractional-order controller design and digital implementation using FOMCON toolbox for MATLAB. IEEE Conf on Computer Aided Control System Design, p.340-345. https://doi.org/10.1109/CACSD.2013.6663486

Valério D, Costa JSD, 2006. Tuning of fractional PID controllers with Ziegler-Nichols-type rules. *Signal Process*, 86(10):2771-2784. https://doi.org/10.1016/j.sigpro.2006.02.020

Valério D, Costa JSD, 2010. A review of tuning methods for fractional PIDs. Proc 4th IFAC Workshop on Fractional Differentiation and its Applications.

Zhao CN, Xue DY, Chen YQ, 2005. A fractional order PID tuning algorithm for a class of fractional order plants. IEEE Int Conf on Mechatronics Automation, p.216-221. https://doi.org/10.1109/ICMA.2005.1626550