

# A Nature Inspired PID like Fuzzy Knowledge-Based Fractional-Order Controller for Optimization



Ambreesh Kumar and Rajneesh Sharma

**Abstract** Conventional PID (CO-PID) controllers have dominated industrial process control applications. Though their use in industry is still prevalent, new avenues have emerged with the advent of soft computing tools. Several soft computing techniques for implementing conventional PID control have been proposed, e.g., cruise control using genetic algorithm (GA), conical tank regulation using ANT colony optimization (ACO), PID control using multi-objective ACO, automatic voltage regulator system (AVR) using particle swarm optimization (PSO), DC motor control using GA, evolutionary programming (EP) and PSO. The fract-order PID controllers have an advantage over conventional PID controllers in terms of availability of additional tuning parameters. In this research work, the authors propose what may be termed “optimized” tuning method for FFPID controllers using a combination of GA and ant colony techniques on a fuzzy logic platform. The research work attempts to design a controller for integer-order and fract-order plants by unifying nature inspired optimization techniques with proportional–integral–derivative (PID) like fuzzy knowledge-based control. The controller employs genetic algorithms (GA) and ANT colony algorithms for offline tuning of fract-order PID controller. Subsequently, fuzzy knowledge-based PID formulation fine-tunes the controller. The authors propose a modified GA-ANT approach wherein the inputs to the ANT system are generated in an optimal manner by using GA. They have simulated it on two distinct plants: (i) DC motor and (ii) a standard fract-order system. Simulation results and comparisons thereof show its superiority and feasibility for control of fract-order plants.

**Keywords** Integer-order plant · Fract-order plants · Fuzzy knowledge-based control · Modified GA-ANT approach

---

A. Kumar  
EC Department, Mewar University, Rajasthan, India  
e-mail: [kumar\\_amb@hotmail.com](mailto:kumar_amb@hotmail.com)

R. Sharma (✉)  
Division of Instrumentation and Control Engineering, NSIT, New Delhi, India  
e-mail: [rajneesh496@gmail.com](mailto:rajneesh496@gmail.com)

© Springer Nature Singapore Pte Ltd. 2020  
M. Pant et al. (eds.), *Computational Network Application Tools for Performance Management*, Asset Analytics,  
[https://doi.org/10.1007/978-981-32-9585-8\\_3](https://doi.org/10.1007/978-981-32-9585-8_3)

## 1 Introduction

Fractional-order systems can be represented as

$$G_f(s) = K_P + \frac{K_I}{s^\lambda} + K_D s^\mu; \quad \lambda, \mu \in (0, 1) \quad (1)$$

where  $K_P$ ,  $K_I$ ,  $K_D$  are the parameters of the conventional PID, and  $\lambda$  and  $\mu$  are fract-order proportional–integral–derivative (FO-PID) controller parameters, respectively [1]. For analyzing any fract-order system (FOS), it needs to be approximated to an equivalent integer-order system (IOS). Several techniques have been reported in the literature for this conversion [2, 3].

Once this approximation has been made, the design of a FO-PID controller is straightforward using various optimization techniques and random search methods. Conventional PID (CO-PID) controllers have dominated industrial process control applications. Though their use in industry is still prevalent, new avenues have emerged with the advent of soft computing tools.

Several soft computing techniques for implementing conventional PID control have been proposed, e.g., cruise control using genetic algorithm (GA) [4], conical tank regulation using ANT colony optimization (ACO) [5], PID control using multi-objective ACO [6], automatic voltage regulator system (AVR) using particle swarm optimization (PSO) [7], DC motor control using GA, evolutionary programming (EP) and PSO [8, 9]. In [10], authors propose linear quadratic regulator and PID controller for inverted pendulum wherein the parameters of the PID controller are tuned using ACO.

As mentioned earlier, fract-order PID controllers have an advantage over conventional PID controllers in terms of availability of additional tuning parameters. Authors in [11] tune both CO-PID and FO-PID controllers by adjusting GM and PM for temperature control; in [12], authors have applied GA for tuning of seven and eight parameters of the FO-PID controller and test it on integer-order plants (IOP) with delay.

In [13], authors have performed control and stability studies on fractional-order chaotic systems, and in [14, 15], fract-order systems for IOPs and AVRs are tuned using PSO. Zamani et al. [16] presents the design of an  $H_\infty$  optimal FO-PID controller. Tuning is performed using multiple random search techniques in [17, 18] for control of robotic manipulators and UAV autopilot. A FO-PID heart rate controller is designed in [19] and control of blood glucose levels in diabetes patients is developed in [20] using GA.

Fuzzy logic-based fract-order PID formulation provides better flexibility and control. It is suitable for the control of highly nonlinear systems and facilitates easy computations online. Robust fuzzy PID (FPID) controller for control of permanent magnet synchronous motor (PMSM) has been proposed in [21], and FPID controller for the inverted pendulum is described in [22]. Here, the authors have used ACO to estimate the initial parameters of the FPID.

In [23–25], FFPID controllers have been proposed where fract-order parameters are optimized using GA. Other notable approaches include sliding mode control, e.g., steam distillation process and PMSM for electric vehicles using PSO [26]. An FFPID approach using ACO for the control of IOP and FOP is detailed in [27]. Finally, a variable-order FFPID controller is discussed in [28].

In this paper, we propose what may be termed “optimized” tuning method for FFPID controllers using a combination of GA and ant colony techniques on a fuzzy logic platform. The controller is simulated on integer-order and fractional-order systems for validation. Based on minimization of the integral of the square of error, GA outputs a set of fract-order parameters as well as initial input parameter values for the ACO. Thereafter, ACO fine-tunes these initial fract-order parameters. In the final step, PID like fuzzy logic is used with error and derivative of error as inputs for generating a set of fract-order parameters. All the three fract-order parameter sets are algebraically combined to get the resulting nature inspired FFPID controller.

The key benefit is harnessing the generalization capability of fuzzy systems in tandem with optimality provided by nature inspired algorithms for generating a versatile fract-order controller. Section 2 gives brief details on the nature inspired algorithms used in this work, PID like fuzzy knowledge-based controller and our nature inspired fuzzy fractional controller. Section 3 details simulation results and associated discussions, and Sect. 4 concludes the paper.

## 2 Nature Inspired Algorithms for Fract-Order Systems

### 2.1 ANT Colony Optimization (ACO)

ACO [29] belongs to the family of algorithms collectively bunched “nature inspired” wherein optimization is carried out on the behavioral pattern of ants. Ants practice an excellent way of optimizing the path they take for food search and other tasks. First, they select an arbitrary path, and then this path is marked for other ants by releasing pheromones.

Next step is to choose the best path based on the probability of pheromone trails. An interesting phenomenon is the disappearance of lower quality pheromones trails. Thus, only the best or most optimal path gets reinforced by repeated pheromone secretion by other ants. The algorithm has the following steps:

- (i) A pheromone factor matrix  $\tau$  is chosen for obtaining a good solution.

$$\tau = \{\tau_{pq}\} \text{ and } \tau_{pq} = \tau_i \forall (p, q) \quad (2)$$

$p$  and  $q$  are nodes along the ant path.

- (ii) Next, the probability that an ant will choose the node  $q$  from node  $p$  is evaluated as

$$P_{pq}(t) = \frac{[\tau_{pq}]^\alpha [\eta_{pq}]^\beta}{\sum [\tau_{pq}]^\alpha [\eta_{pq}]^\beta} \forall p, q \in T^1 \quad (3)$$

In each iteration, a solution is obtained using the probability  $P_{pq}(t)$ . In (2),  $\eta_{pq} = \frac{1}{k_q}$  is a factor which is heuristic in nature;  $\alpha$  and  $\beta$  are constants which determine the influence of pheromone and heuristic values on the choice of the trail taken by an ant; and  $T^1$  is the path decided by an ant at any given time. When a new trail is chosen, the weaker pheromone trails disappear. The quantity of pheromone on a path is

$$\Delta\tau_{pq}^1 = \begin{cases} \frac{L^{\min}}{L^1} & \text{if } p, q \in T^1 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The structure formula [28] for updating of a pheromone is

$$\tau_{pq}(t) = \rho\tau_{pq}(t-1) + \sum_1^N \Delta\tau_{pq}^1(t) \quad (5)$$

where  $N$  is the number of ants and the evaporation rate is  $\rho \in (0, 1]$ .

## 2.2 Genetic Algorithm

Genetic algorithm is a stochastic search technique based on human genome selection procedure and is classified as “evolutionary algorithm.” GA uses the concept of natural selection and genetics and has been widely used for optimization tasks. GA is structured on the processes of natural selection, recombination and mutation of chromosomes. As with the case of any other nature inspired optimization algorithm, this technique follows the principle of survival of the fittest. The algorithm has the following steps:

- (i) **Selection:** First a random choice of individuals from a population is done, and the fittest one is selected for crossover. Selection is done by evaluating a fitness function corresponding to each individual.
- (ii) **Modification:** Modification is implemented by the process of crossover (recombination) or mutation. Crossover is the process of combining parent solutions and producing an offspring. The process of crossover and mutation attempts at generating a more fit set of individuals. Mutation prevents the algorithm from being trapped in a local minimum. Mutation maintains genetic diversity in the population. It introduces new genetic structures in the population by randomly modifying some of the individuals of the string.

The algorithm terminates when either a maximum number of generations have been produced or a sufficient fitness level has been achieved for the target population.

### 2.3 PID like Fuzzy Knowledge-Based Control (FKBC)

Fuzzy logic imitates human thinking and modeling of systems and processes. Implementing fuzzy logic-based control typically involves: (a) fuzzifying the inputs or converting crisp inputs to fuzzy membership functions; (b) coding human or process knowledge in terms of a rule base; (c) using fuzzy inputs along with rule base for inferring outputs called fuzzy inference system and (d) converting the aggregated fuzzy output from the fuzzy inference system to a crisp (usable) output by defuzzification.

Though any process or state variables can be chosen as input variables for the fuzzy logic-based control, a natural and more efficient method of implementing fuzzy knowledge-based control (FKBC) is to use error ( $e$ ) and derivative of error ( $\partial e$ ) as inputs to the fuzzy logic controller also termed as “PID like FKBC” in the literature. In this work, fuzzy membership functions (laid over the input variables universe of discourse) have been chosen as Gaussian, and the FIS is “Mamdani” [29]. Table 1 gives the rule base for the PID like FKBC for the variables  $K_p$ ,  $K_d$ ,  $K_i$ ,  $\lambda$ ,  $\mu$ . It is worthwhile to note that we have framed rules for not only the conventional PID parameters  $K_p$ ,  $K_d$ ,  $K_i$  but also for the additional fractional-order parameters  $\lambda$ ,  $\mu$ .

These rules have been framed based on empirical knowledge gained during simulations of fract-order PID control. Three fuzzy subsets are chosen, i.e., “Negative,” “Zero” and “Positive” for the inputs  $e$  and  $\partial e$ . The output variables  $K_p$ ,  $K_d$ ,  $K_i$ ,  $\lambda$  and  $\mu$  have been fuzzified using five fuzzy subsets: NL (negative large), NLE (negative less), ZR (zero), PLE (positive less), PL (positive large). Defuzzification has been performed using the centroid method [30]. The plots for the membership functions laid over the inputs and outputs are shown in Fig. 1. This completes the design of the PID like FKBC.

### 2.4 Nature Inspired Fuzzy Fractional PID (NIFFPID)

In this work, we propose to unify all the three approaches outlined above. This is done by obtaining an algebraic sum of all the approaches, i.e., combining the PID like FKBC, ANT colony-based optimizer and the GA-based control. The aim here is to find the best possible values for the fractional PID parameters by using all the three approaches to aid each other in this process. The added advantage is that the inaccuracy of one of the approaches could possibly be covered by the others and the results indeed point to this. Complete structure of the proposed nature inspired fuzzy fractional PID controller is shown in Fig. 2.

As can be seen from the figure that each of nature inspired algorithms, i.e., GA and ACO output a set of fract-order parameters which are algebraically added to the parameters provided by the FKBC, and the combined strategy is applied to the plant being controlled.

**Table 1** Fract-order PID like FKBC rule base

$K_P$			$K_I$			$K_D$			$\lambda$			$\mu$		
IF	AND	THEN	IF	AND	THEN	IF	AND	THEN	IF	AND	THEN	IF	AND	THEN
$e$	? $e$	$K_p$	$e$	? $e$	$K_i$	$e$	? $e$	$K_d$	$e$	? $e$		$e$	? $e$	$\mu$
Negative	Negative	Positive large	Negative	Negative	Negative large	Negative	Negative	Positive large	Negative	Negative	Positive large	Negative	Negative	Positive less
Negative	Zero	Zero	Negative	Zero	Negative large	Negative	Zero	Positive less	Negative	Zero	Positive less	Negative	Zero	Positive less
Negative	Positive	Positive less	Negative	Positive	Negative large	Negative	Positive	Positive large	Negative	Positive	Positive less	Negative	Positive	Positive large
Zero	Negative	Negative large	Zero	Negative	Negative large	Zero	Negative	Negative less	Zero	Negative	Zero	Zero	Negative	Zero
Zero	Zero	Negative large	Zero	Zero	Negative less	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero	Zero
Zero	Positive	Positive large	Zero	Positive	Negative less	Zero	Positive	Positive less	Zero	Positive	Negative less	Zero	Positive	Positive less
Positive	Negative	Positive large	Positive	Negative	Positive less	Positive	Negative	Positive less	Positive	Negative	Positive less	Positive	Negative	Positive large
Positive	Zero	Positive less	Positive	Zero	Zero	Positive	Zero	Positive large	Positive	Zero	Positive large	Positive	Zero	Positive less
Positive	Positive	Positive large	Positive	Positive	Positive less	Positive	Positive	Positive large	Positive	Positive	Positive large	Positive	Positive	Positive large

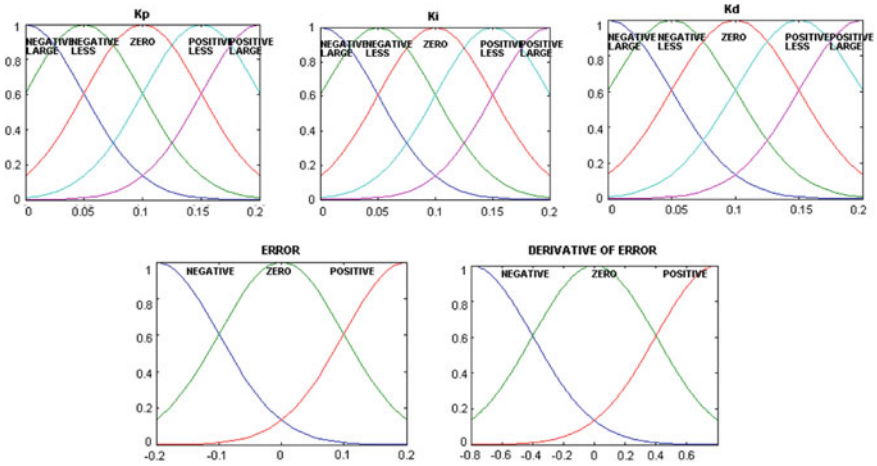
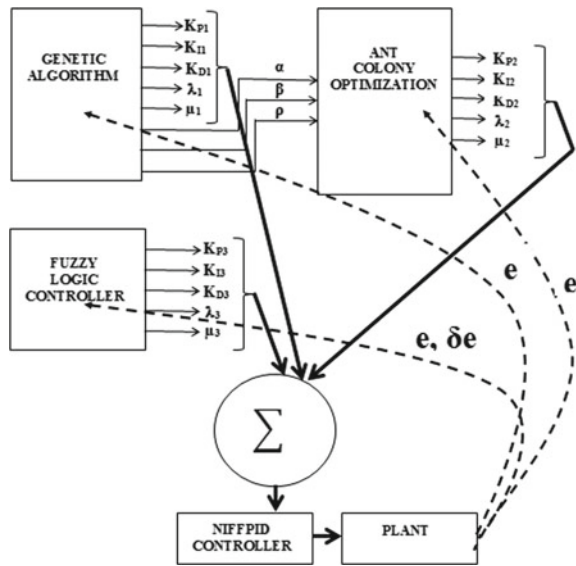


Fig. 1 Membership functions for inputs:  $e$  and  $\partial e$ ; outputs:  $K_p, K_i, K_d$

Fig. 2 Block diagram of NIFFPID controller



### 3 Simulation of Nature Inspired PID like FKBC

The proposed approach is tested on two systems:

### 3.1 DC Motor

The first plant considered is a DC motor having a transfer function

$$G_1 = \frac{-0.05963s^2 + 1.254s + 0.8736}{s^3 + 0.9708s^2 + 1.796s + 0.8736} \quad (6)$$

Unconstrained minimization with GA is used to evaluate the parameters of the fract-order PID controller and the same are fed as inputs to the ANT colony algorithm. Performance index chosen is the integral of square error (ISE). GA-based optimization yields the following parameters:

(i) Fract-order PID parameters

$$K_{p1} = 0.1833, K_{i1} = 1.6168, K_{d1} = 1.1348, L_1 = 0.0001, M_1 = 0.8712 \quad (7)$$

The GA controller transfer function is

$$F_{GA\_FOPID\_1}(s) = 0.1833 + \frac{1.6168}{s^{0.0001}} + 1.1348s^{0.8712} \quad (8)$$

(ii) ANT colony initial parameter estimate

$$\alpha = 0.0187, \beta = 0.6283, \rho = 0.0780 \quad (9)$$

Using these values, the ANT colony algorithm (2) is implemented, and the ANT colony-based fract-order PID parameters are obtained:

$$K_{p2} = 0.11, K_{i2} = 0.18, K_{d2} = 0.19, L_2 = 0.01, M_2 = 0.05 \quad (10)$$

Transfer function of ACO fract-order PID is

$$F_{ANT\_FOPID\_1}(s) = 0.11 + \frac{0.18}{s^{0.01}} + 0.19s^{0.05} \quad (11)$$

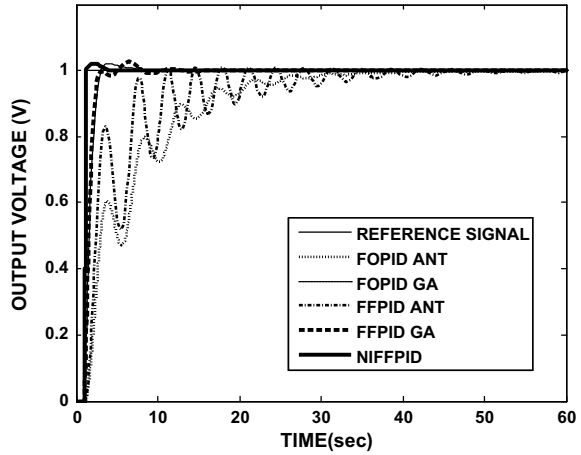
Finally, we merge the outputs of these fractional-order controllers, viz. GA, ANT and FKBC to generate our nature inspired fuzzy fractional PID (NIFFPID) controller with transfer function:

$$F_{NIFFPID\_1}(s) = 0.0871 + \frac{0.2445}{s^{0.0204}} + 2.0204s^{0.9242} \quad (12)$$

Figure 3 gives the responses obtained on the DC motor with different controllers. The NIFFPID controller proposed in this paper achieves a rise time of 0.27 s which



**Fig. 3** Step response of various controllers on DC motor



is significantly lower than the best rise time achieved by other controllers (FOPID, FFPID using GA) of 2.143 s. Our NIFFPID control achieves peak time of 1.102 s, settling time of 1.646 s and a peak overshoot of 0.022. These values point to pretty good time response performance.

### 3.2 *Fract-Order System*

The second case study is a fractional-order system with transfer function

$$G_2(s) = \frac{5}{s^{2.3} + 1.3s^{0.9} + 1.25} \tag{13}$$

Once again, GA is used to get the fract-order PID parameters:

$$K_{p2} = 2.3117, K_{i2} = 0.5424, K_{d2} = 0.9052, L_2 = 0.0172, M_2 = 0.9997 \tag{14}$$

Resulting GA-FOPID controller transfer function

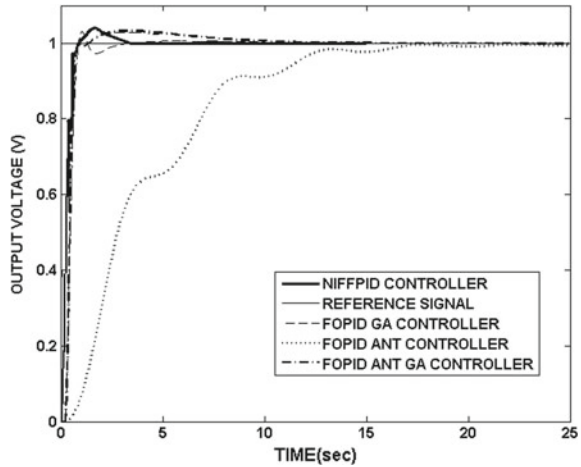
$$F_{GA-FOPID\_2}(s) = 2.3117 + \frac{0.5424}{s^{0.0172}} + 0.9052s^{0.9997} \tag{15}$$

ACO initial parameters obtained are

$$\alpha = 0.3, \beta = 0.7, \rho = 0.1 \tag{16}$$

These parameters are used to generate an ACO fract-order PID controller

**Fig. 4** Step response of various controllers for fract-order system



$$K_{p2} = 0.06, K_{i2} = 0.02, K_{d2} = 0.40, L_2 = 0.13, M_2 = 0.2 \quad (17)$$

ACO transfer function of FOPID controller becomes

$$F_{\text{ANT-FOPID}_2}(s) = 0.06 + \frac{0.02}{s^{0.13}} + 0.40s^{0.2} \quad (18)$$

We combine the three controller realizations to generate NIFFPID controller with transfer function:

$$F_{\text{NIFFPID}_2}(s) = 2.4704 + \frac{0.6618}{s^{0.1484}} + 1.4052s^{0.9998} \quad (19)$$

Simulation results using different controllers on this plant are given in Fig. 4. Our NIFFPID controller (Fig. 4) is seen to achieve the best response parameters of rise time of 0.89 s and settling time of 3.00 s. Thus, an amalgamation of nature inspired approaches with fuzzy logic control technique, viz. GA, ACO and FKBC does lead to an improved controller response.

## 4 Conclusion and Future Work

An attempt has been made to hybridize two nature inspired algorithms in a PID like fuzzy setting. Our main objective is to use nature-based optimization for tuning of fuzzy logic-based fract-order PID controller. The methodology is implemented on two systems and performance compared with other soft computing-based fract-order controllers. Results indicate that our proposed controller gives better closed loop time response over other controllers.

In future, the authors would like to implement the proposed approach on some fract-order systems. We would also augment our methodology by employing type-2 fuzzy systems and self organizing fuzzy inference system structure.

## References

1. Y.Q. Chen, K.L. Moore, Discretization schemes for fractional-order differentiators and integrators. *IEEE Trans. Circuits Syst. I Fund. Theory. Appl.* **49**(3), 363–367 (2002)
2. D.Y. Xue, Y.Q. Chen, Sub-optimum  $H_2$  rational approximations to fractional-order linear systems, in *IDETC/CIE Conference*, California (2005), pp. 1–10
3. A. Oustaloup, J. Sabatier, P. Lanusse, From fractional robustness to crone control. *Fract. Calc. Appl. Anal.* **2**(1), 1–30 (1999)
4. M.K. Rout, D. Sain, S.K. Swain, S.K. Mishra, PID controller design for cruise control system using genetic algorithm, in *International Conference on Electrical Electronics and Optimization Techniques (ICEEOT)*, Chennai (2016), pp. 4170–4174
5. K.V.L. Narayana, V.N. Kumar, M. Dhivya, R.P. Raj, Application of ant colony optimization in tuning a PID controller to a conical tank. *Indian J. Sci. Technol.* **8**(S2), 217–223 (2015)
6. I. Chiha, N. Liouane, P. Borne, Tuning PID controller using multiobjective ant colony optimization. *J. Appl. Comput. Intell. Soft Comput.* **11** (2012)
7. M.F. Aranza, J. Kustija, B. Trisno, D.L. Hakim, Tuning PID controller using particle swarm optimization algorithm on automatic voltage regulator system, in *IOP Conference Series: Materials Science and Engineering*, vol. 128 (2016), pp. 1–9
8. N.P. Adhikari, A. Gupta, PID controller tuning using soft computing techniques, in *All India Seminar on Biomedical Engineering*, Lecture Notes in Bioengineering (Springer, India, 2012)
9. N. Tandan, K.K. Swarnkar, PID controller optimization by soft computing techniques-a review. *Int. J. Hybrid Inf. Technol.* **8**(7), 357–362 (2015)
10. A. Jacknoon, M.A. Abido, Ant colony based LQR and PID tuned parameters for controlling inverted pendulum, in *International Conference on Communication, Control, Computing and Electronics Engineering*, Khartoum (2017), pp. 1–8
11. C.I. Muresan, Simplified optimization routine for tuning robust fractional order controllers. *Am. J. Comput. Math.* **3**, 7–12 (2013)
12. Y. Zhang, J. Li, Fractional-order PID controller tuning based on genetic algorithm, in *International Conference on Business Management and Electronic Information*, Guangzhou (2011), pp. 764–767
13. A. Soukkou, S. Leulmi, Controlling and synchronizing of fractional-order chaotic systems via simple and optimal fractional-order feedback controller. *Int. J. Intell. Syst. Appl.* **8**, 56–69 (2016). <https://doi.org/10.5815/ijisa.2016.06.07>
14. A.K. Mahmood, B.F. Mohammed, Design of fractional order PID controller based particle swarm. *Diyala J. Eng. Sci.* **7**(4), 42–93 (2014)
15. N.R. Raju, P.L. Reddy, Robustness study of fractional order PID controller optimized by particle swarm optimization in AVR System. *Int. J. Electr. Comput. Eng.* **6**(5), 2033–2040 (2016)
16. M. Zamani, M.K. Ghartemani, N. Sadate, Design of an  $H_\infty$ —optimal FOPID controller using particle swarm optimization, in *CCC*, Hunan (2007), pp. 435–440
17. D. Fani, E. Shahraki, Two-link robot manipulator using fractional order PID controllers optimized by evolutionary algorithms. *Biosci. Biotechnol. Res. Asia* **13**(1), 589–598 (2016)
18. G. Li, C. Guo, Y. Li, W. Deng, Fractional-order PID controller of USV course-keeping using hybrid GA-PSO algorithm, in *8th International Symposium Computational Intelligence and Design*, Hangzhou (2015), pp. 506–509
19. S.P. Arunachalam, S. Kapa, S.K. Mulpuru, P.A. Friedman, E.G. Tolkacheva, Intelligent fractional-order PID (FOPID) heart rate controller for cardiac pacemaker, in *IEEE Healthcare Innovation Point-Of-Care Technologies Conference*, Cancun (2016), pp. 105–108

20. M. Goharimanesh, A. Lashkaripour, A.A. Mehrizi, Fractional order PID controller for diabetes patients. *JAMECH* **46**(1), 69–76 (2015)
21. B.T. Zhang, Y. Pi, Robust fractional order proportion-plus-differential controller based on fuzzy inference for permanent magnet synchronous motor. *IET Control Theory Appl.* **6**(6), 829–837 (2012)
22. H. Boubertakh, M. Tadjine, P.Y. Glorennec, S. Labiod, Tuning fuzzy PID controllers using ant colony optimization, in *17th Mediterranean Conference Control and Automation*, Thessaloniki (2009), pp. 13–18
23. S. Das, I. Pan, S. Das, A. Gupta, A novel fractional order fuzzy PID controller and its optimal time domain tuning based on integral performance indices. *J. Eng. Appl. Artif. Intell.* **25**(2), 430–442 (2012)
24. V. Kumar, K.P.S. Rana, J. Kumar, P. Mishra, S.S. Nair, A robust fractional order fuzzy P + fuzzy I + fuzzy D controller for nonlinear and uncertain system. *Int. J. Autom. Comput.* 1–13 (2014)
25. R.H. Mohammed, F. Bendary, K. Elserafi, Trajectory tracking control for robot manipulator using fractional order-fuzzy-PID controller. *Int. J. Comput. Appl.* **134**(15), 22–29 (2016)
26. M. Tajjudin, N. Ishak, M.H.F. Rahiman, R. Adnan, Design of fuzzy fractional-order PI + PD controller, in *12th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*, Malacca City (2016), pp. 253–257
27. R. Singh, A. Kumar, R. Sharma, Fractional order PID control using ant colony optimization, in *1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems*, Delhi (2016), pp. 1–6
28. L. Liu, F. Pan, D. Xue, Variable-order fuzzy fractional PID controller. *ISA Trans.* **55**, 227–233 (2015)
29. M. Dorigo, The ant system optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* **26**(1), 1–13 (1996)
30. K.S. Book Miller, B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equation* (Wiley, New York, 1993)