# Fractional order ant colony control with genetic algorithm assisted initialisation

## Ambreesh Kumar

ECE Division,
Mewar University, India
Email: kumar_amb@hotmail.com

## Varun Upadhyaya, Ayush Singh, Paras Pandey and Rajneesh Sharma*

Department of Instrumentation and Control Engineering,
NSUT,
Sector-3, Dwarka, New Delhi 110075, India
Email: varunupa95@gmail.com
Email: ayush4021@gmail.com
Email: paraspandey14@gmail.com
Email: rajneesh496@gmail.com
*Corresponding author

**Abstract:** The task of parameter initialisation of an ant colony optimisation (ACO) has gained much attention in recent years. For the systems using ACO-based control, the technique used was generally hit and trial. However, in order to be able to obtain better and faster response, along with better convergence, for control of fractional order (FO) systems, it became imperative to formulate some approach. In this paper, we have used genetic algorithm (GA) to initialise the ACO parameters for a systematic design of ACO-based fractional order controllers. The GA-based ACO fractional order PID controller is developed by minimisation of a multi-objective function using a nested GA technique. The effectiveness of the method used is verified using seven FO systems. The results are compared with the controllers based on ACO and GA. The proposed GA-based ACO controller yields reasonably better performance as compare to the existing techniques with a slight weakness of higher computational complexity. This limitation can be easily overcome by use of high performance machines.

**Keywords:** fractional order system; ant colony optimisation; ACO; genetic algorithm; GA-based ACO.

**Reference** to this paper should be made as follows: Kumar, A., Upadhyaya, V., Singh, A., Pandey, P. and Sharma, R. (2021) 'Fractional order ant colony control with genetic algorithm assisted initialisation', *Int. J. Swarm Intelligence*, Vol. 6, No. 1, pp.77–90.

**Biographical notes:** Ambreesh Kumar received his BE degree in Electronics Engineering in 1992 from Nagpur University and MTech in Engineering Systems in 1994 from Faculty of Engineering, Dayalbagh Educational Institute (DEI). He is pursuing his PhD from Mewar University, Chittorgarh. He is presently working in a private company as an engineer. His areas of interest are control and signal processing.

Varun Upadhyaya received his BE degree in Instrumentation and Control Engineering from NSIT in 2018 and his MTech degree in VLSI Design Tools and Technology from IIT Delhi in 2020. He is currently an Analogue Design Engineer with the Analogue Power Products Group at Texas Instruments India, where he works on the design of high performance low-dropout linear voltage regulators (LDOs).

Ayush Singh received his BE degree in Instrumentation and Control engineering from NSIT in 2018. He is a Senior Engineer at Samsung Semiconductor India R&D. For the past two years, he worked on the design and verification of Samsung's flagship mobile system on chip focusing on the sensor hub sub-system and on-chip communications.

Paras Pandey received his BE degree in Instrumentation and Control engineering from NSIT in 2018. He is currently working in the marketing team at analogue devices.

Rajneesh Sharma obtained his BE in Electrical Engineering, ME in Control and Instrumentation from Delhi College of Engineering, Delhi, and PhD in Electrical Engineering from Indian Institute of Technology, Delhi in 1993, 1999, and 2007 respectively. Thereafter, he carried out Post Doctoral Research at Institute for Systems and Robotics, IST, Lisbon, Portugal for one year. His current research interests include Markov game based control, reinforcement learning, soft computing tools and their applications to control. Currently, he is a Professor with the Instrumentation and Control division at the Netaji Subhas University of Technology, New Delhi, India.

This paper is a revised and expanded version of a paper entitled 'Genetic algorithm based optimisation of ant colony controller for fractional order systems' presented at ICCI 2017, Indian Institute of Technology, Kanpur, India, 6–8 December 2017.

## 1 Introduction

One of the most reliable controller for dynamical systems is the proportional-integral-derivative. Recently, researchers have toyed with several soft computing approaches that were earlier used in the conventional PID control to increase effectiveness and reliability of PID controllers, e.g., neural networks have been used for online nonlinear PID controller design in Kumar et al. (2016). Rastogi et al. (2011) proposed a fuzzy logic PID controller for stabilising electronic circuits. The fuzzy PID controller is compared with a Zeigler-Nichols (ZN) tuning-based conventional PID controller. Jayachitra and Vinodha (2014), propose a continuous stirred tank reactor (CSTR) process controller wherein genetic algorithm (GA) has been used GA for adjusting the PID control parameters.

Fractional order (FO) PID controller design attempts to tune two additional FO parameters which are additional to the three parameters used in the conventional PID controller making FO control more complex but more effective. Generalisation of PID control to fractional order systems gives fract order PID (FOPID) control (Vinagre and Monje, 2012). Fract order PID control is nearer to actual systems as all physical systems are inherently fractional in reality. Thus, fractional calculus is used to model a system more precisely. Fract order PID controller's transfer function (PIλDμ) is:

$$G_{FOPID}(x) = \frac{U_{cont}(x)}{I_{error}(x)} = K_p + K_i x^{\lambda} + K_d x^{-\mu} \tag{1}$$

In equation (1), we notice that $\lambda$ and $\mu$ are the two other parameters in the fract order setup as compared to just three in the PID control. This makes the controller design more versatile than conventional PID one.

Several authors attempted to incorporate evolutionary algorithms in FOPID, e.g., a fractional fuzzy FOFPID controller is envisaged in Mishra et al. (2015), for distillation column. Basu et al. (2017) designed a FOPID controller for heating furnace using different optimisation techniques whereas, Zhang and Li (2011) proposed genetic algorithm-based tuning of fractional-order PID controller. Noureddine et al. (2013) deliberated tuning of fuzzy fractional order PID sliding-mode controller using PSO algorithm for non-linear systems. Zukhri and Paputungan (2013) suggested hybrid optimisation algorithm based on genetic algorithm and ant colony optimisation. Presented controller belongs to the Takagi Sugeno type with non-integer values of differentiation and integration operators on a fuzzy setup. In yet another application the authors of Ismayil et al. (2015), automated generation control of thermal power plants with genetic algorithm assisted fract order PID (GAFOPID) has been implemented. The controller employs GA and fuzzy logic for finding parameters of a PI controller. Ramezanian and Balochian (2013) propose optimisation based on particle swarms for adjusting the parameters of a PID controller for application on automatic voltage regulator system. The authors compare PSO-FOPID controller to FOPID/PID control and show that their control formalism has superior robustness and model uncertainties handling capability. Singh et al. (2016) propose an ant colony assisted fractional fuzzy PID approach and give simulation results on several fract-order plants. In yet another proposal, a GA assisted ACO control has been depicted in Suribabu and Chiranjeevi (2016) for optimising dynamical characteristics of automated voltage regulation systems.

Researchers proposed GA (Zalzala and Fleming, 1997) as an evolutionary technique inspired by Darwinian theory with three main tenets: reproduction, crossover and mutation. Over the years, GA has proven its mettle by providing reliable solutions to a wide array of problems. Another conspicuous evolutionary approach is the ANT colony algorithm developed in Dorigo et al. (1999) which is motivated by the approach taken by ants in locating a food source and optimising the path to it. The ants do this by releasing and reinforcing a pheromone along the food path. Most optimal path is repeatedly sprayed over by other ants till a more optimal one is found (if any). The algorithm that emulates this ant behaviour is called ant colony optimisation (ACO). In our current work, we have attempted to combine these two approaches together, i.e., GA is used to tune the ACO controller. The novelty of the method lies in the fact that:

1 the initial parameters of ACO have not been chosen randomly; GA has been used to tune these parameters

2 the optimisation approach used in the paper is a multi-objective optimisation, i.e., instead of optimising only one parameter, four time-domain specification parameters are optimised simultaneously.

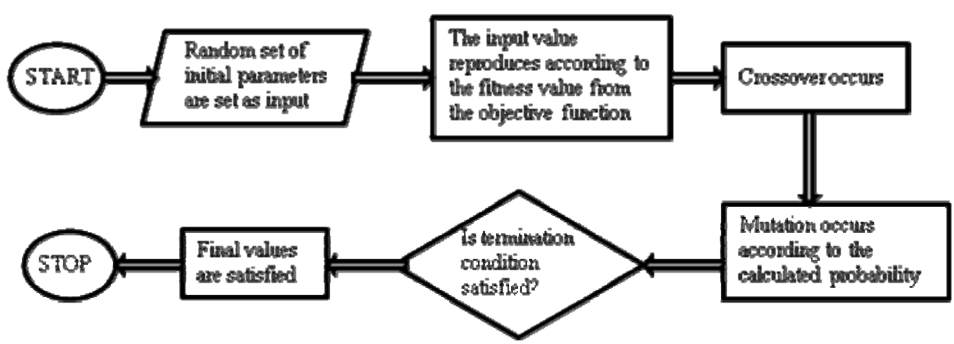
Choice of initial parameters of ACO algorithm, i.e., $\alpha$, $\beta$ and $\rho$ (to be explained in Section 3), *apriori* is a critical task. Incorrect choice increases the computational time unnecessarily. Tuning these parameters using GA reduced both time and complexity of the task. Also, by using the multi-objective optimisation, better results were obtained as compared to the GA and ACO algorithm results used individually. This motivated us to use the proposed approach.

We give a concise overview of the GA and ACO techniques in next two sections, respectively. A detail of our GA tuned ANT controller approach along with the discussion of the multi-objective function is considered in Section 4. In Section 5, the simulation results of the GA-based ACO approach applied to the seven different fract-order plants are presented and their comparison with ACO and GA-based FOPID controllers also deliberated. Conclusions are given in Section 6.

## 2 Genetic algorithm

It is an algorithm with roots in the evolutionary biology and genetics. In other words, it mimics the way humans and other species have evolved into better and fitter ones over time. Starting point is a set of candidate solutions to a problem at hand which are completely random. From this random raw set of individuals called population, fitter set of individuals evolve. The main criteria for generating better population is an evaluation function called the fitness function. Main processes involved in producing better set of individuals are selection, reproduction, crossover and mutation. This is similar to evolution of species on this planet over time. Fitness function is used to evaluate quality of a candidate solution to a problem and is primarily problem dependent. At each stage, the off springs are evaluated vis. a vis. parents to see whether we get a better population set. It is a stochastic approach meaning that there holds no guarantee that only the highest scoring individuals will form the next population; some low scoring candidates also form part of the solution set. This procedure helps maintain diversity in the resulting solution and avoids solution getting stuck in local minima. In literature, we find roulette wheel and tournament selection as the two primary methods for selecting next population set.

**Figure 1** GA procedure in nutshell




GA terminates on two counts: when a satisfactory level of performance is reached in terms of a fitness value or a specified maximum number of generations have been produced. The algorithm acclimatises with the environment by producing an alternate set of population by means of reproduction. Fitter set of off springs serve as parents for the next generation. When a specified number of iterations have taken place, mutation takes place. Mutation seeks to bring in genetic material that may have been left out. Figure 1 gives a pictorial presentation of the GA process.

## 3 Ant colony approach

Herein, we take a snapshot of the ant colony algorithm presented in Dorigo et al. (1999) and discuss the design of a FOPID controller via the ACO technique, i.e., varying the ant variables $\alpha$, $\beta$ and $\rho$. ACO is an algorithm which belongs to the umbrella techniques collectively called the meta heuristic search techniques. The algorithm searches for the best path between the sets of nodes and mimics the ants. The ants have a unique way of identifying the most economical path from their origin to the source. Although individual ants do not have much intelligence, yet their collective effort leads them to the most optimal path to the food.

Ants search for food in a completely random manner, initially. But this randomness changes when a food source is identified by the ant. This particular ant then takes a small bit of food back to the colony and while doing so sprays a pheromone on the path it takes to the colony. Other ants lock on to this pheromone trail and also carry food from the source to the colony and in doing so reinforce the pheromone trail by the first ant. This reinforcement increases the pheromone strength along the path. These pheromones gradually evaporate over time so a particular path needs to be reinforced time and again to remain a viable one. Most of the time ants stick to the path marked by pheromone but sometimes an ant takes a somewhat different path or an exploratory one. This facilitates search of other optimal path by other ants. If the new path is better than the previous one the earlier path is discarded and the new one is reinforced by the pheromones.

ACO technique tries to mimic this ant behaviour by attempting to optimise a cost function. In our current work, we have employed ACO technique to optimise a multiple object function which includes time domain specifications and the intergral time absolute error (ITAE) index. The ACO variables $\alpha$, $\beta$ and $\rho$ chart out the efficiency of the ACO approach. $\alpha$ and β are the parameters defined by the user while $\rho$(ro) is evaporation rate of the pheromone. We detail our implementation of the ACO approach for FO plant control:

1 Initialising the search

   A nodal matrix containing possible parameters of *KP*, *KD*, *KI*, *lb*(lambda) and *mu* is created wherein the values are uniformly distributed. Next, a matrix specified node to node transition probabilities is created and has the size same as the nodal matrix. The probability of visiting a particular node is same, initially. Second step is creation of a pheromone matrix (Dorigo et al., 1999) which lists accumulation and evaporation of pheromone concentration. In addition, an ant matrix is also created to count the nodes traversed by the $m^{\text{th}}$ ant after every iteration.

2 Probability matrix update

   The transition probability, i.e., the probability of transition of ant from node $p$ to $q$ is calculated by

$$P_{mn}^{S}(t) = \frac{[\tau_{mn}(t)]^{\alpha}[\eta_{mn}]^{\beta}}{\sum_{m,n \in T^{S}}[\tau_{mn}(t)]^{\alpha}[\eta_{mn}]^{\beta}}; \text{ if } m, n \in T^{S} \tag{2}$$

where $\tau_{mn}$ = pheromone parameter; $\eta_{mn} = (1 / d_{mn})$ is a heuristic parameter; $d_{mn}$ is the distance of node $m$ from node $n$. $\tau_{mn}$ is based on the ant's behaviour that have got good solution. The decision of subsequent ants is characterised by parameters $\alpha \geq 0$ and $\beta \geq 0$ and depends on pheromone strength and heuristic values. The path taken by an ant $A$ is given by $T^{A}$. Pheromone strength released on a particular path is:

$$\Delta\tau_{pa}^{A} = \begin{cases} \dfrac{L^{\min}}{L^{A}} & \text{if } p, q \in T^{A} \\ 0 & \text{else} \end{cases} \tag{3}$$

Best solution and objective function generated by ants is given by $L^{A}$ and $L^{\min}$, respectively. The probability matrix update and the value of $\Delta\tau_{mn}^{A}$ is redefined after every iteration and reflects the route taken by the $m^{\text{th}}$ ant. Strength of pheromone is updated as:

$$\sum \Delta\tau^{j} = \Delta\tau^{j=1} = \frac{\xi f_{top}}{f_{bottom}} \tag{4}$$

Evaporation pheromone rate is updated as:

$$\tau_{mn}(t) = \rho\tau_{mn}(t-1) + \sum_{A=1}^{N} \Delta\tau_{mn}^{A}(t) \tag{5}$$

Number of ants are denoted by $N$ and evaporation rate by $\rho$ $(0 < \rho < 1)$.

## 4 GA-based ant technique

In this work, we attempt to use GA to optimally tune the ACO algorithm parameters for designing the FOPID controllers. ACO algorithm's performance depends on these ant parameters. In literature, we find no mechanism for initialising the ACO parameters, they are set randomly. This work is aimed at removing this randomised approach to initial ANT parameter selection. We use GA as the optimising technique over the population of possible ANT parameters $\alpha$, $\beta$ and $\rho$. Better individuals are selected in GA using an evaluative function as outlined in the previous section. This is done by using the conventional GA processes of selection, crossover and mutation. After a specific number of GA iterations, we get optimal initial ACO parameters.

1 Initialise: First step is generation of a random population relating to values of ACO parameters.

2 Calculating cost functions: Next step involves calculation of cost function with these set of ACO algorithm parameters values extracted from the population.

3 GA processes of selection, crossover, mutation: Based on the cost function values, we extract best four $\alpha$, $\beta$ and $\rho$ values. GA is employed wherein rest of the population is mated with the random population. GA is done with parameters: No. of chromosomes = 12, size of population is 8, mutation rate of 10% and iteration count is 50.

4 Looping: We repeat the steps 2 through 3 for a fixed iteration count.

5 Result lock: Once the best set of ACO parameters are attained, the iterations are terminated.

### *4.1 Multi-objective functional*

Generally, a effectiveness of a controller is validated using the cost function intergral square of error (ISE) or ITAE. A recent trend is to club one or more of such indices termed as a multi-objective functional mentioned in Meng and Xue (2009). This performance measure facilitates simultaneous satisfaction of a number of performance parameters. For FOPID controller, we are concerned with optimisation of some of these performance parameters. In this work, we design a multi-objective functional which forms a weighted sum of these indices:

$$C = w_1 I_1 + w_2 I_2 + w_3 I_3 + w_4 I_4 \tag{6}$$

where $I_1$ refers to settling time, $I_2$ refers to rise time, $I_3$ refers to peak overshoot and $I_4$ is ITAE with being weights relating to the performance index.

## 5 Results

An 8 GB RAM Intel Core i5 processor with speed of 1.6 GHz has been used for simulating our GA tuned ANT based FOPID controller. The proposed controller performance is perused on seven unrelated fract-order systems and compared with both the ant colony and GA approaches dealt one at a time. The parameters (GA) are: population number = 8, number of iterations (limiting) = 50; chromosome number = 12; percentage of mutation = 10%. FOPID parameters are constrained with values of $K_P$, $K_i$ and $K_d$ in the range [0, 10); and $\lambda$ and $\mu$ in the range [0.1).

GA based ACO approach is simulated with the parameters as follows: number of iterations for GA = 5, iterations for ACO = 50; ants = 10; nodes = 100 (parameter tuned on100 × 5 nodal matrix). ACO parameters being $0 \le \alpha < 2$; $0 \le \beta < 1$; $0 \le \rho < 2$. The systems on which we test our GA-based ACO controllers are:

$$\text{Sys. 1}: \quad G_1(s) = \frac{5}{x^{2.3} + 1.3x^{0.9} + 1.25} \tag{7}$$

$$\text{Sys. 2}: \quad G_2(x) = \frac{1}{0.8x^{2.2} + 0.5x^{0.9} + 1} \tag{8}$$

$$\text{Sys. 3}: \quad G_3(x) = \frac{x+1}{10x^{3.2} + 185x^{2.5} + 288x^{0.7} + 1} \tag{9}$$

$$\text{Sys. 4}: \quad G_4(x) = \frac{5x^{0.5} + 2}{x^{3.3} + 3.1x^{2.6} + 2.89x^{1.9} + 2.5x^{1.4} + 1.2} \tag{10}$$

$$\text{Sys. 5}: \quad G_5(x) = \frac{1}{x^{2.2} + 3.2x^{1.4} + 2.4x^{0.9} + 1} \tag{11}$$

$$\text{Sys. 6}: \quad G_6(x) = \frac{1}{x^{2.8} + 3.6x^{1.5} + 1} \tag{12}$$

$$\text{Sys. 7}: \quad G_7(x) = \frac{1}{x^{1.6} + 8.8x^{1.2} + 1} \tag{13}$$

The time response of various controllers for step input for plant 1 is depicted in Figure 2. We notice that our proposed approach yields best time domain response. Another advantage is better ITAE value of 2.6697 (Table 1) achieved by our GA based ACO controller in comparison with other approaches.

**Figure 2** Time domain output for step input of Sys. 1: G A, A C O, GA-ACO

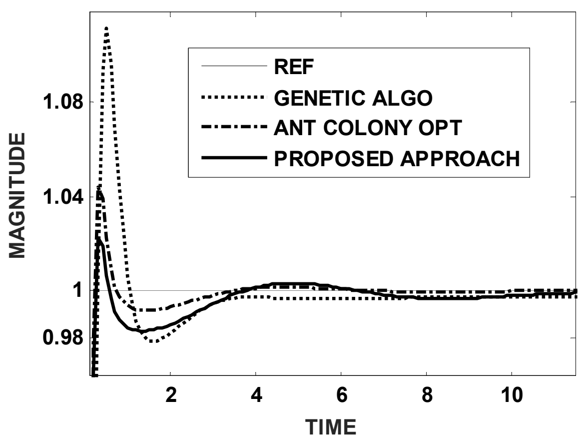



**Figure 3** Time domain output for step input of Sys. 2: G A, A C O, GA-ACO

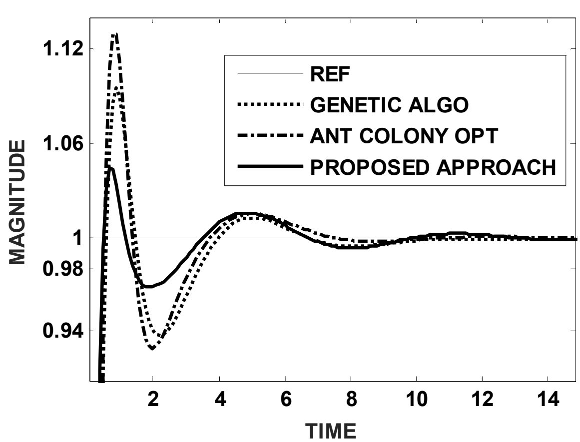


**Figure 4** Time domain output for step input of Sys. 3: G A, A C O, GA-ACO

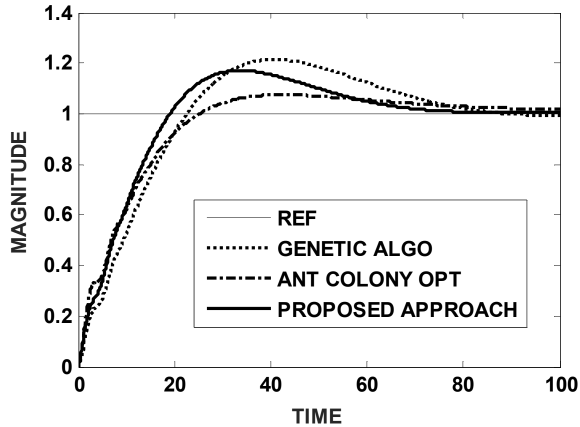



**Figure 5** Time domain output for step input of Sys. 4: G A, A C O, GA-ACO

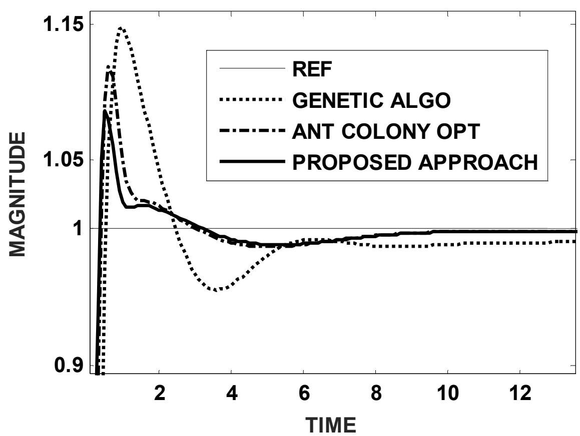



**Figure 6** Time domain output for step input of Sys. 5: G A, A C O, GA-ACO

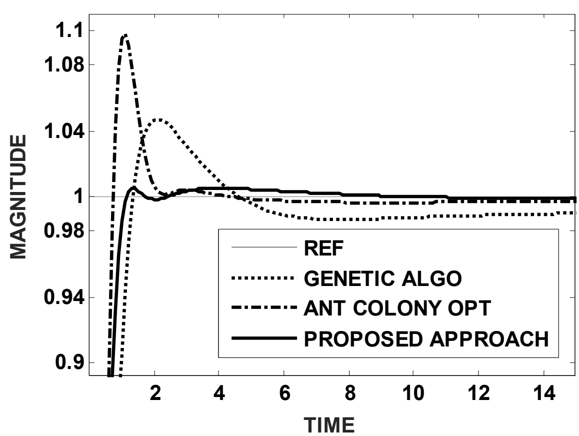


**Figure 7** Time domain output for step input of Sys. 6: G A, A C O, GA-ACO

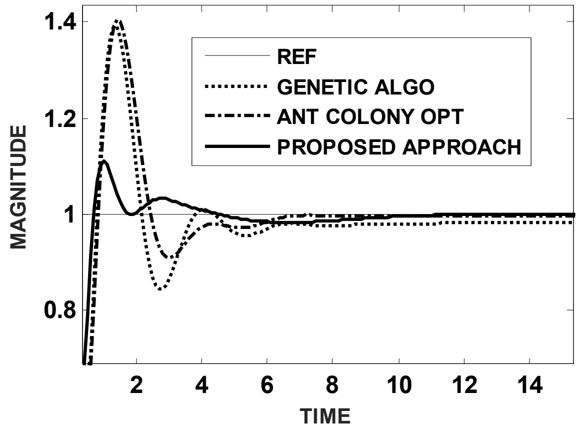
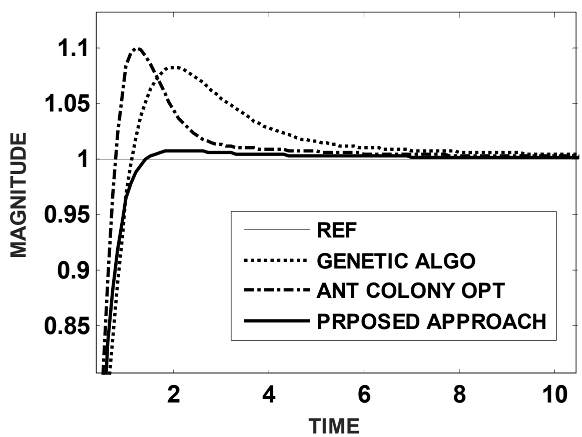



**Figure 8** Time domain output for step input of Sys. 7: G A, A C O, GA-ACO




To evaluate efficacy of the approach used in the paper, simulations are performed on six other fract order systems. The parameters for simulating the plants are listed in Table 1. From the simulation results depicted in Figure 3, we see that our GA tunes ACO approach achieves best time response specifications against other techniques. This trend of superior results is repeated in case of five other plants as well (Figure 4 to Figure 8).

A comparison of time response parameters for all the plants is listed in Table 1. The GA-based ACO technique achieves best transient as well as steady state parameters. This can be judged by looking at the parameter values listed in Table 1.

**Table 1** Time response comparison of GA-based ACO with other FOPID controllers

| *Technique* | *FOPID Parameters* | *Response* | *ITAE* |
|---|---|---|---|
| | *Sys. 1* | | |
| Genetic algorithm | PC = 1.76, IC = 8.52, DC = 3.64; lbd = 0.64, mu = 0.96 | r-t = 0.224, s-t = 1.849, Overshoot% = 11.15 | 94.313 |
| Ant colony optim. | PC = 6.8, I C =9.57, DC = 9.26; lbd = 0.9098, mu = 0.9752 | r-t = 0.158, s-t = 0.517, Overshoot% = 4.408 | 3.634 |
| Proposed approach | al = 1.03, bt = 0.01, ro = 0.18 PC = 0.07, IC = 8.51, DC = 9.81; lbd = 0.7136, mu = 0.9881 | r-t = 0.162, s-t = 0.372, Overshoot% = 2.240 | 2.669 |
| | *Sys. 2* | | |
| Genetic algorithm | PC = 0.87, IC = 5.59, DC = 4.71; lbd = 0.91, mu = 0.83 | r-t = 0.439, s-t = 3.379, Overshoot% = 9.504 | 9.379 |
| Ant colony optim. | PC = 1.53, IC = 5.75, DC = 5.3 lbd = 0.9871, mu = 0.7611 | r-t = 0.404, s-t = 3.115, Overshoot% = 13.003 | 7.042 |
| Proposed approach | al = 0.69, bt = 0.17, ro = 0.93; PC = 1.41, IC = 7.41, DC = 6.92; lbd = 0.9732, mu = 0.91676 | r-t = 0.356, s-t = 2.690, Overshoot% = 4.447 | 6.271 |
| | *Sys. 3* | | |
| Genetic algorithm | PC = 8.070, IC = 4.830, DC = 4.750 lbd = 0.870, mu = 0.1500 | r-t = 17.854, s-t = 80.286, Overshoot% = 21.468 | 8376.82 |
| Ant colony optim. | PC = 8.26, IC = 9.76, DC = 8.57 lbd = 0.6422, mu = 0.0397 | r-t = 17.996, s-t = 92.172, Overshoot% = 7.502 | 2464.5 |
| Proposed approach | al = 1.71, bt = 0.03, ro = 1.71 PC = 9.91, IC = 8.34, DC = 6.7 lbd = 0.784, mu = 0.814 | r-t = 14.895, s-t = 69.542, Overshoot% = 17.161 | 2408.2 |
| | *Sys. 4* | | |
| Genetic algorithm | PC = 1.01, IC = 6.21, DC = 4.05 lbd = 0.59, mu = 0.99 | r-t = 0.395, s-t = 4.994, Overshoot% = 14.693 | 282.7 |
| Ant colony optim. | PC = 5.46, IC = 4.6, DC = 6.85 lbd = 0.9682, mu = 0.94351 | r-t = 0.293, s-t = 1.597, Overshoot% = 11.897 | 19.74 |
| Proposed approach | al = 0.58 bt = 0.02 ro = 0.02 PC = 5.78, IC = 5.78, DC = 9.26 lbd = 0.9098, mu = 0.97523 | r-t = 0.255, s-t = 0.995, Overshoot% = 8.611 | 17.087 |
| | *Sys. 5* | | |
| Genetic algorithm | PC = 2.55, IC = 5.75, DC = 6.39 lbd = 0.67, mu = 0.54 | r-t = 0.870, s-t = 3.507, Overshoot% = 4.697 | 44.222 |
| Ant colony optim. | PC = 9.60, IC = 6.45, DC = 9.64 lbd = 0.856, mu = 0.4202 | r-t = 0.518, s-t = 1.763, Overshoot% = 9.818 | 42.646 |
| Proposed approach | al = 1.3, bt = 0.34, ro = 1.86 PC = 9.17, IC = 3.72, DC = 7.84 lbd = 0.963, mu = 0.609 | r-t = 0.655, s-t = 0.971, Overshoot% = 0.543 | 11.983 |

Notes: Notation: PC = KP, IC = KI, DC = KD.
r-t, s-t, Overshoot% - specifications of rise time, settling time, percentage overshoot.

**Table 1** Time response comparison of GA based ACO with other FOPID controllers (continued)

| *Technique* | *FOPID Parameters* | *Response* | *ITAE* |
|---|---|---|---|
| | | *Sys. 6* | |
| Genetic algorithm | PC = 0.2302, IC = 9.4394, DC = 6.5566 lbd = 0.49610, mu = 0.6627 | r-t = 0.83, s-t = 6.5, Overshoot% = 28.68 | 74.594 |
| Ant colony optim. | PC = 1.97200, IC = 7.3974, DC = 6.5566 lbd = 0.8512 , mu = 0.6151 | r-t = 0.795, s-t = 6.1, PO% = 28.00 | 21.680 |
| Proposed approach | al = 0.010, bt = 0.010, ro = 0.010 PC = 5.9159, IC = 4.7748, DC = 9.0390 lbd = 0.9424, mu = 0.9484 | r-t = 0.690, s-t = 6.00, Overshoot% = 9.91 | 11.289 |
| | | *Sys. 7* | |
| Genetic algorithm | PC = 7.9900, IC = 4.8300, DC = 3.4700 lbd = 0.4700, mu = 0.7900 | r-t = 0.784, s-t = 4.532, Overshoot% = 8.262 | 15.476 |
| Ant colony optim. | PC = 4.7347, IC = 6.5766, DC = 7.6977 lbd = 0.1866 , mu = 0.0437 | r-t = 0.547, s-t = 2.528, Overshoot% = 10.02 | 9.063 |
| Proposed approach | al = 0.8100, bt = 0.3700, ro = 1.7200 PC = 5.9159, IC = 4.7748, DC = 9.0390 lbd = 0.9424, mu = 0.9484 | r-t = 0.703, s-t = 1.121, Overshoot% = 0.783 | 5.413 |

Notes: Notation: PC = KP, IC = KI, DC = KD.
r-t, s-t, Overshoot% - specifications of rise time, settling time, percentage overshoot.

From Table 1, it is observed that:

1 In Sys. 1, the peak overshoot has reduced considerably and the settling time has also reduced. However, the rise time is comparable with the ACO approach.

2 In Sys. 2, all the three parameters have decreased considerably when the proposed approach is applied.

3 In Sys. 3, though the peak overshoot is not the lowest, but settling and rise times are significantly less.

4 In Sys. 4, again the proposed approach yields best results.

5 In Sys. 5, the peak overshoot has diminished substantially, whereas the other two parameters have also decreased.

6 In Sys. 6, again there is a dramatic decrease in peak overshoot, whereas, the settling and rise times are also comparable.

7 In Sys. 7, there is a drastic reduction in peak overshoot, but at the compromise of rise time, although the settling times for the three methods are comparable.

Another very important performance measure is the computational time comparison of controllers. We give computational complexity of various controllers in Table 2. As expected, we see that our GA-based ACO has highest computational time. This result obtained is as expected based on our technique wherein GA is nested with ACO. Even then our approach holds promise:

a it is a futuristic approach as it does away with the trial and error procedure adopted in the ANT colony approach

b has the best transient response amongst all the controllers

c produces best steady state performance as well. In today's age of fast processing computers which are available at cheap prices, the increased computational burden would not pose a major hurdle.

**Table 2** Controller comparison: computational complexity

| *Technique* | *Computation time (sec)* |
|---|---|
| GA | 5.32 |
| ACO | 7.92 |
| GA based ACO | 20.53 |

## 6 Conclusions

GA based ACO technique as outlined in this work is a research leap in the designing of ANT colony based controller design philosophy. We have removed the trial and error procedure used so far in the ACO technique with a mathematically oriented GA based optimisation. The GA initialised ACO parameters lead to faster convergence to target values as evidenced by superior transient response and steady state response of our proposed GA tunes ACO approach. To showcase feasibility of our approach we simulated the technique on seven different fract order systems and compared its performance against two other contemporary evolutionary algorithm based approaches. One small though not negligible drawback of our technique is relatively large computational complexity.

## References

Basu, A., Mohanty, S. and Sharma, R. (2017) 'Designing of FOPID controller for heating furnace using different optimization techniques', in *ISCO 2017: Proceedings of the 11th IEEE International Conference on Intelligent Systems and Control*, pp.318–322.

Dorigo, M., Caro, G.D. and Gambardella, L.M. (1999) 'Ant algorithms for discrete optimization', *Artificial Life*, Vol. 5, No. 2, pp.137–172.

Ismayil, C., Kumar, R.S. and Sindhu, T.K. (2015) 'Optimal fractional order PID controller for automatic generation control of two-area power systems', *Electrical Energy Systems*, Vol. 25, No. 12, pp.3329–3348, Wiley.

Jayachitra, A. and Vinodha, R. (2014) 'Genetic algorithm based PID controller tuning approach for continuous stirred tank reactor', *Advances in Artificial Intelligence*, Hindawi, Article ID 791230.

Kumar, R., Srivastava, S. and Gupta, J.R.P. (2016) 'Artificial neural network based PID controller for online control of dynamical systems', in *ICPEICES 2016: Proceedings of the1st International Conference on Power Electronics, Intelligent Control and Energy Systems*, Delhi, pp.1–6.

Meng, L. and Xue, D. (2009) 'Design of an optimal fractional-order PID controller using multi-objective GA optimization', in *CDC 2009: Proceedings of the Chinese Control and Decision Conference*, Guilin, pp.3849–3853.

Mishra, P., Kumar V. and Rana, K.P.S. (2015) 'A fractional order fuzzy PID controller for binary distillation column control', *Expert Systems with Applications*, Vol. 42, No. 22, pp.8533–8549, Elsevier.

Noureddine, B., Djamel, B. and Boudjema, F. (2013) 'Tuning fuzzy fractional order PID sliding-mode controller using PSO algorithm for nonlinear systems', in *Proceedings of the 3rd International Conference on Systems and Control*, Algeria, 978-1-4799-0275-0/13/$31.00 ©2013 IEEE.

Ramezanian, S.H. and Balochian, S. (2013) 'Design of optimal fractional-order PID controllers using particle swarm optimization algorithm for automatic voltage regulator (AVR) system', *Journal of Control, Automation and Electrical Systems*, Vol. 24, No. 5, pp.601–611.

Rastogi, M., Arora, A. and Hote, Y.V. (2011) 'Design of fuzzy logic based PID controller for an unstable system', *Computer Networks and Intelligent Computing*, Vol. 157, No. 3, pp.566–571, Springer.

Singh, R., Kumar A. and Sharma, R. (2016) 'Fractional order PID control using ant colony optimization', in *ICPEICES 2016: Proceedings of the 1st International Conference on Power Electronics, Intelligent Control and Energy Systems*, Delhi, pp.1780–1786.

Suribabu, A.G. and Chiranjeevi, B.T. (2016) 'Implementation of fractional order PID controller for an AVR system using GA and ACO optimization techniques', *IFAC*, Vol. 49, No. 1, pp.456–461, Elsevier.

Vinagre, B.M. and Monje, C.A. (2012) *Fractional-Order PID, Chapter 15, PID Control in the Third Millennium, Advances in Industrial Control*, Springer, London.

Zalzala, A.M.S. and Fleming, P.J. (1997) 'Genetic algorithm in engineering systems', *J. IET*, London, ISBN: 0852969023.

Zhang, Y. and Li, J. (2011) *Fractional-Order PID Controller Tuning Based on Genetic Algorithm*, pp.764–767, 978-1-61284-109-0/11/$26.00 © 2011 IEEE.

Zukhri, Z. and Paputungan, I.V. (2013) 'A hybrid optimization algorithm based on genetic algorithm and ant colony optimization', *International Journal Of Artificial Intelligence & Applications (IJAIA)*, September, Vol. 4, No. 5, pp.63–75, DOI: 10.5121/Ijaia.2013.4505.