ICIC Express Letters
Volume **10**, Number **9**, September **2016**
 ISSN 1881-803X
pp. 2291–2296

A FOPID CONTROLLER DESIGN FOR CHAOS SYNCHRONIZATION BY ANT COLONY OPTIMIZATION

Kuo-Nan Yu, Chin-Tsung Hsieh, Her-Terng Yau*
Yi-Hung Lin and Siao-Ren Jhang

Department of Electrical Engineering
National Chin-Yi University of Technology
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan
*Corresponding author: htyau@ncut.edu.tw

Received February 2016; accepted May 2016

Abstract. *This paper discusses the use of a Fractional Order Proportional Integral Derivative (FOPID) controller to carry out the master-slave chaos synchronization control. In a normal integer order PID (IOPID) controller, a significant number of main types of gain constant combinations exist. In FOPID, the orders of the integrator and the differentiator are not integers, requiring the use of an intelligent algorithm to obtain optimal values after matching. The intelligent algorithm used here was Ant Colony Optimization (ACO), and in comparison to the integer order PID controller, the number of parameter values required was increased to five, making the optimization process more complicated. Nevertheless, from the simulation result, it can be seen that with the same objective function of Integrated Absolute Error (IAE), the FOPID calculated using ACO in the master-slave synchronous system process allows optimal parameters to be found with fewer iterations compared to the integer order PID controllers.*
Keywords: FOPID system, Ant colony optimization, Master-slave synchronous system

1. **Introduction.** The discovery and utilization of the chaotic synchronous system is a recent event [1] and has been found to be of great benefit in numerous applications in particular in physics, engineering and communications security systems. Chaotic synchronization was first proposed by Pecora and Carroll [2]. A normal chaotic system synchronization method typically uses a nonlinear system controller as a master-slave chaotic system to drive the slave system and synchronize it with the master system to achieve chaotic synchronization. In general, a PID controller is the simplest and easiest method used for synchronization. However, the question of how to determine the optimal system parameters of the PID controller is an important issue. In addition, dynamic characteristics of non-integer order exist in the chaotic system. In recent years, a lot of systems that originally used integer order have been studied again for their differences (apart from the integer order) because the fractional order system has drawn a great deal of attention. Therefore, if the PID controller of non-integer order is taken into consideration, there is an opportunity to describe the non-integer order characteristics in the control of a chaotic synchronous system more appropriately. Accordingly, this paper discusses the differences and pros/cons of an integer order PID (IOPID) and a fractional order PID (FOPID) [3].

In a PID control system, common algorithms used to find optimal parameters include: Particle Swarm Optimization (PSO) [4], the Genetic Algorithm (GA) [5] and the Evolutionary Programming (EP) method [6]. Particle swarm optimization is often used in a PID controller. The most important difference between the FOPID controller and the PID controller is that the PID has three variables, whereas the FOPID requires the finding of five variables. Consequently, particle swarm optimization often leads to a situation where an optimal solution is sought within a local range. The genetic algorithm can preserve an optimal solution while the remaining genes pair, mutate and duplicate to form other genes. This algorithm involves the "chance" of finding an optimal solution, but not every pairing will lead to the required optimal solution. The Evolutionary Programming method is similar to the genetic algorithm in that there is much uncertainty in the method used to obtain an optimal solution. To overcome these difficulties we have used Ant Colony Optimization to find the optimal parameters of a chaotic synchronous system. From the results, it shows that the ACO method has faster rate of convergence than the GA method in Reference [6].

The organization of this paper is as follows. Section 2 addresses the system description and problem formulation of this study. Section 3 describes the Ant Colony Optimization. In Section 4, the result of the experiments is presented to show the effectiveness of the proposed method. Finally, the conclusions are presented in Section 5.

2. **System Description and Problem Formulation.** Consider the description of two chaotic master-slave systems [6] of Single Input and Single Output (SISO), and their definitions are as shown in Equations (1) and (2):

$$\begin{cases} \dot{x}_m(t) = F(t, x_m) \\ y_m(t) = Ax_m \end{cases} \tag{1}$$

$$\begin{cases} \dot{x}_s(t) = F(t, x_s) + Bu(t) \\ y_s(t) = Ax_s \end{cases} \tag{2}$$

where $x_m(t) = [x_{m1}, x_{m2}, \ldots, x_{mn}] \in R^n$ and $x_s(t) = [x_{s1}, x_{s2}, \ldots, x_{sn}] \in R^n$ are the state vectors of master and slave systems, respectively. $F : R \times R^n \to R^n$ is a given nonlinear function. $y_m(t) \in R$ and $y_s(t) \in R$ are the outputs of the master and slave systems, respectively. $B \in R^{n\times 1}$ and $A \in R^{1\times n}$ are the system matrices. $u(t) \in R$ is the control input and added in the slave system (2) to achieve synchronization between master and slave systems. Generally many chaotic systems can be expressed by (1).

Set the synchronization error state to: $e_1 = x_{m1} - x_{s1}$, $e_2 = x_{m2} - x_{s2}$, $\ldots$, $e_n = x_{mn} - x_{sn}$, the purpose being to design a more effective FOPID controller using Ant Colony Optimization, and $u(t)$ can achieve synchronization with Equations (1) and (2), as shown in Equation (3):

$$\lim_{t\to\infty} \| x_m(t) - x_s(t) \| \to 0 \tag{3}$$

The input $u(t)$ is a function of the output error signal $y_e = y_m - y_s$, and the relationship between the input $y_e(\cdot)$ and output $u(\cdot)$ of the PID controller is defined as shown in Equation (4):

$$u(t) = k_p y_e(t) + k_i \int_0^t y_e(\tau)d\tau + k_d \frac{d}{dt} y_e(t) \tag{4}$$

Next, taking the Laplace Transform of Equation (4), we can obtain Equation (5) as shown below:

$$G(s) = k_p + \frac{k_i}{s} + k_d s \tag{5}$$

The expression for FOPID is as shown in Equation (6):

$$G(s) = k_p + \frac{k_i}{s^\alpha} + k_d s^\beta \tag{6}$$

The range of α, β in the aforementioned equation is $0 < \alpha, \beta < 1$. Here, the IAE is used for the objective function (referred to as "OF") as shown in Equation (7):

$$OF = IAE = \sum_{k=1}^{k_f} \| E(k) \| \tag{7}$$

The Riemann-Liouville [7] definition is used here, and the normal fractional order general equation can be expressed by Equation (8) as shown below:

$$D_t^\alpha f(t) \approx \frac{\gamma(m+1)}{\gamma(m+1-\alpha)} t^{m-\alpha} \tag{8}$$

wherein $\gamma()$ represents γ function (gamma function), α is an order of derived function, and m is a positive integer; when the value is $0 < \alpha < 1$, it represents a physical phenomenon of the fractional order. Because the Matlab® program was used here, the affirmative value of the fractional order cannot be obtained; consequently, the approximate solution of Ousaloup as shown in Equation (9) is used [8]:

$$s^q = k \prod_{n=1}^{N} \frac{1 + \frac{s}{\omega_{zn}}}{1 + \frac{s}{\omega_{pn}}} \quad q > 0 \tag{9}$$

3. **Ant Colony Optimization.** It has been shown that Ant Colony Optimization can effectively process a great number of very complicated problems [9]. We used Ant Colony Optimization to obtain the minimum value of IAE. During the marching process of m number of ants, the probability of choosing the next path is as shown in Equation (10) as $p_{ij}^n(t)$:

$$p_{ij}^n(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \times (\eta_{ij})^\beta}{\sum_{u \in J_n(i)} (\tau_{iu}(t))^\alpha \times (\eta_{iu})^\beta}, & \text{if } j \in J_n(i) \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

α is the pheromone parameter; β is the reciprocal of distance; $p_{ij}^n(t)$ refers to the probability of the n^{th} ant at the t^{th} iteration from node i to node j; η_{ij} is the reciprocal of the distance between node i and node j; $\tau_{ij}(t)$ refers to the concentration of pheromone at the t^{th} iteration from node i to node j; $J_n(i)$ refers to the set of nodes that the n^{th} ant at node j has not yet passed. The greater the value of $(\tau_{ij}(t))^\alpha \times (\eta_{ij})^\beta$, the greater the probability of selection; after the completion of each calculation, the pheromone concentration must always be updated, as shown in Equations (11) and (12):

$$\tau_{ij}(t) = (1-\delta) \times \tau_{ij}(t) + \sum_{n=1}^{m} \Delta\tau_{ij}^n, \tag{11}$$

$$\Delta\tau_{ij}^n = \begin{cases} \frac{Q}{A_n}, & \text{if } n^{\text{th}} \text{ ant uses edge } ij \text{ in its tour} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

δ is the pheromone evaporation coefficient. Q is the parameter affecting the pheromone. A_n is the solution obtained by the n^{th} ant; after the completion of one iteration, the pheromone for paths that the ants have not passed by is zeroed; when all the ants travel on one identical path, this will be the shortest distance and also the optimal solution; when the number of iteration reaches a certain quantity or reaches a certain number of optimal solutions, the optimization algorithm can be ended. The flowchart of this algorithm is shown in Figure 1.

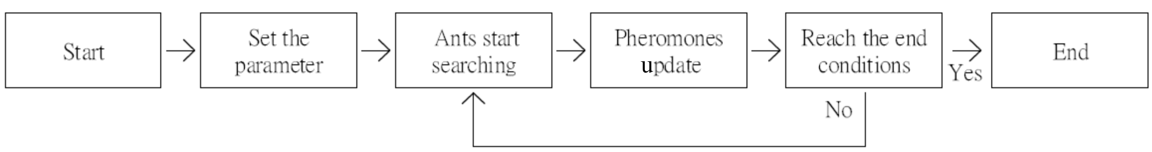


FIGURE 1. Ant colony optimization flowchart

4. **Result of the Experiments.** Here, the Sprott [6] chaotic circuit is referenced for simulation, and the system equations are (13) and (14).

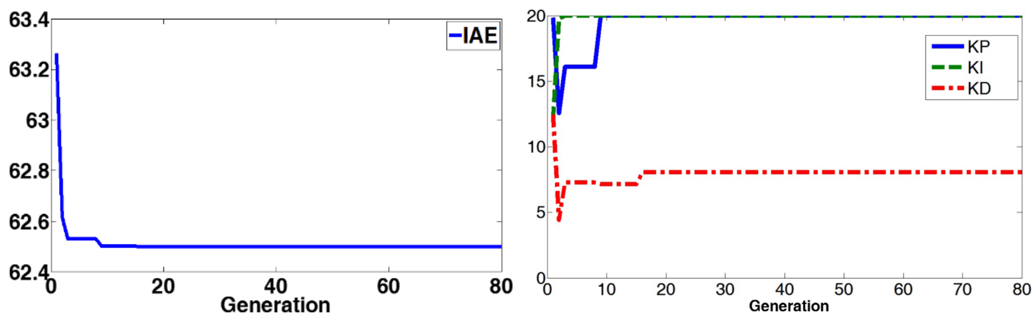
Master:

$$\begin{aligned}\dot{x}_{m1} &= x_{m2}\\ \dot{x}_{m2} &= x_{m3}\\ \dot{x}_{m3} &= -1.2x_{m1} - x_{m2} - 0.6x_{m3} + 2\cdot \mathrm{sign}(x_{m1})\end{aligned} \tag{13}$$

Slave:

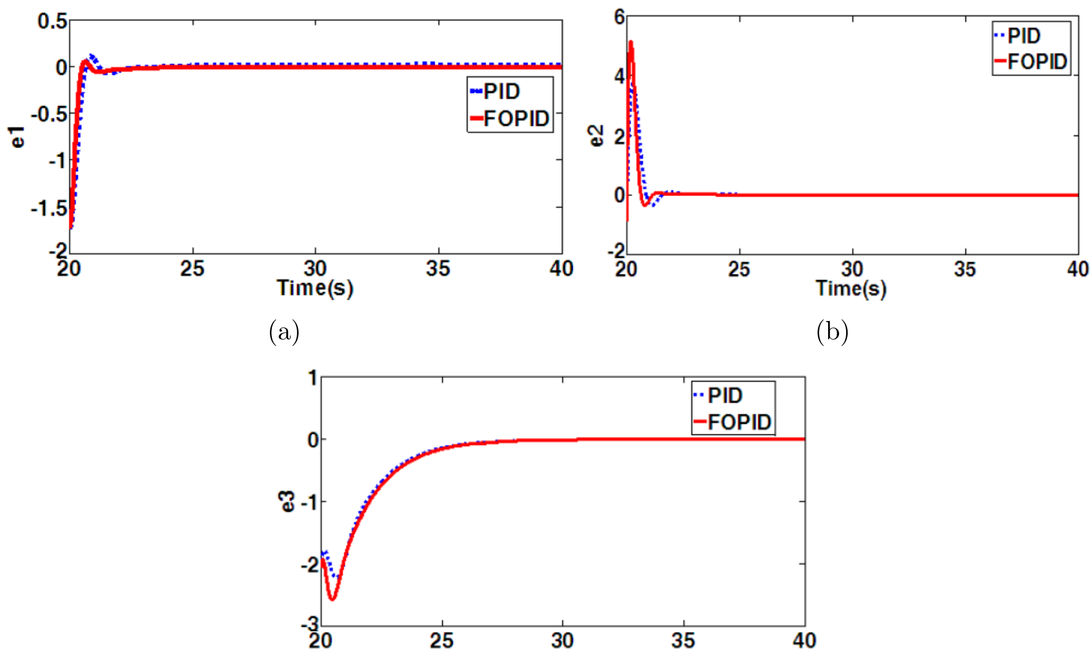
$$\begin{aligned}\dot{x}_{s1} &= x_{s2}\\ \dot{x}_{s2} &= x_{s3} + u(t)\\ \dot{x}_{s3} &= -1.2x_{s1} - x_{s2} - 0.6x_{s3} + 2\cdot \mathrm{sign}(x_{s1})\end{aligned} \tag{14}$$

In the original experiment, the EP method was used to obtain the IAE value, and the optimal solution was found after about 180 iterations. Here, the Ant Colony Optimization is used with the utilization of FOPID to calculate the IAE value, and the optimal solution was found after about 20 iterations as shown in Figure 3. In each iteration, the optimal KP, KI and KD values are found; the changes of optimal values for each iteration are shown in Figure 2.



(a) (b)

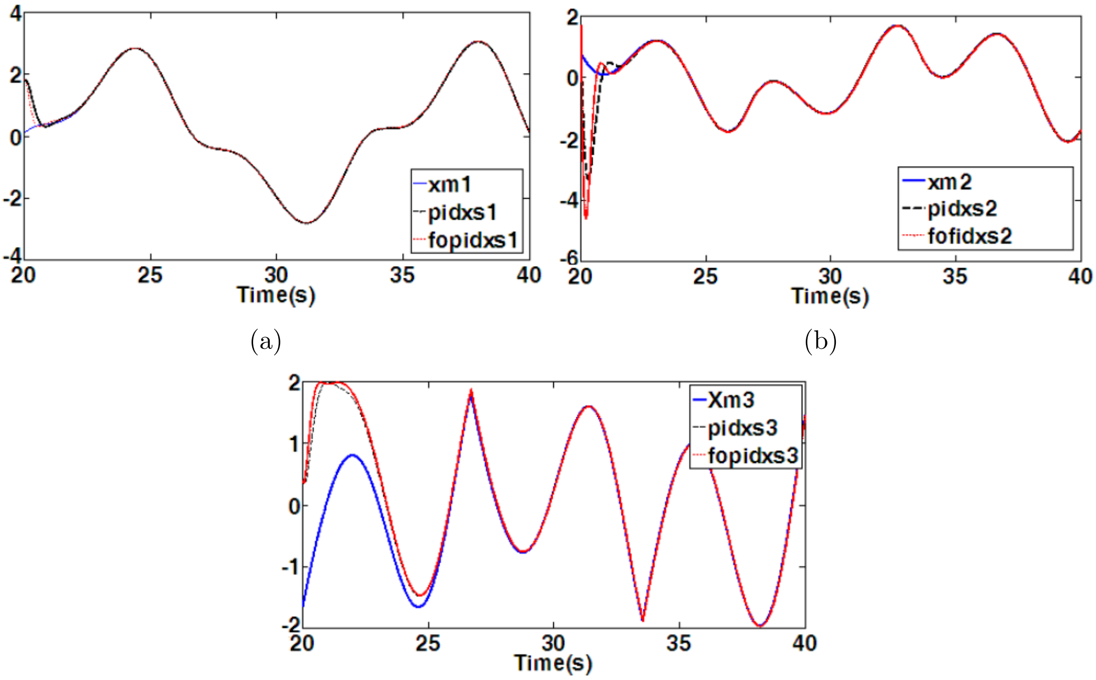
FIGURE 2. IAE values change (a) and KP, KI and KD values change (b) with the iteration number



(a) (b)

(c)

FIGURE 3. Error drawings for PID (dotted line) and FOPID (solid line)



(a) (b) (c)

Figure 4. Master-slave synchronization

Figure 3 shows the error state dynamic responses obtained from an integer order PID and a FOPID. The control input is active at $t = 20$ sec. It can be seen from the error state e1, that when control is introduced in the FOPID, the error is smaller than that seen with the PID. With regard to the error state e2, the starting FOPID has a greater rising time; however, the descending speed and stability are faster than those in the PID. When control is introduced into e3, the amount of descending of the FOPID is greater than that in the PID; however, the rising speed is faster than that of the PID; the curves show no significant differences.

The control input is active at $t = 20$ sec in Figure 3. In Figure 3(a) it can be seen that the stability of a FOPID controller is higher than that of a PID controller. From Figure 3(b) it can be seen that the rising speed of a FOPID controller is faster than that in a PID controller. Figure 3(c) then shows that there are no obvious differences between the two, as shown in Figure 4. The controller also is implemented with the same method as Reference [6].

5. **Conclusions.** The pros and cons of an integer order and a fractional order PID have been discussed in detail. Ant Colony Optimization was used to obtain the solution for system optimization in such a way that observation of the changing pheromone concentration allowed the optimal parameters to be found. These were used in models and in the EP algorithm to compare the error values and IAE value changes. Our final conclusion was that the Ant Colony Optimization algorithm was faster, and the error value control by a FOPID controller was more reliable as well as faster than that of a PID controller. This is a clear advantage in the realization of real-time control. In the future, this result can be applied in the topics of secure communication, etc.

REFERENCES

[1] Z. Alam, L. Yuan and Q. Yang, Chaos and combination synchronization of a new fractional-order system with two stable node-foci, *IEEE/CAA Journal of Automatica Sinica*, vol.3, pp.157-164, 2016.

[2] L. M. Pecora and T. L. Carroll, Synchronization in chaotic systems, *Physical Review Letters*, vol.64, pp.821-824, 1990.

[3] A. Djari, T. Bouden and A. Boulkroune, Design of fractional-order PID controller (FOPID) for a class of fractional-order MIMO systems using a particle swarm optimization (PSO) approach, *The 3rd International Conference on Systems and Control*, pp.1055-1060, 2013.

[4] W.-D. Chang, PID control for chaotic synchronization using particle swarm optimization, *Chaos, Solitons & Fractals*, vol.39, pp.910-917, 2009.

[5] C. Ou and W. Lin, Comparison between PSO and GA for parameters optimization of PID controller, *International Conference on Mechatronics and Automation*, pp.2471-2475, 2006.

[6] H.-T. Yau, Y.-C. Pu and S. C. Li, An FPGA-based PID controller design for chaos synchronization by evolutionary programming, *Discrete Dynamics in Nature and Society*, vol.2011, 2011.

[7] H. Y. Jia, Z. Q. Chen and G. Y. Qi, Chaotic characteristics analysis and circuit implementation for a fractional-order system, *IEEE Trans. Circuits and Systems I: Regular Papers*, vol.61, pp.845-853, 2014.

[8] B. Vinagre, I. Podlubny, A. Hernandez and V. Feliu, Some approximations of fractional order operators used in control theory and applications, *Fractional Calculus and Applied Analysis*, vol.3, pp.231-248, 2000.

[9] M. Dorigo, M. Birattari and T. Stutzle, Ant colony optimization, *IEEE Computational Intelligence Magazine*, vol.1, pp.28-39, 2006.