

## Research Article

# Multiple Input Delays Estimation Using an Artificial Bee Colony Algorithm

**Wei-Der Chang**

*Department of Computer and Communication, Shu-Te University, Kaohsiung 824, Taiwan*

Correspondence should be addressed to Wei-Der Chang; [wdchang@stu.edu.tw](mailto:wdchang@stu.edu.tw)

Received 6 June 2013; Accepted 15 August 2013

Academic Editor: Chang-Hua Lien

Copyright © 2013 Wei-Der Chang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper focuses on the time delay estimation of the system described in the form of discrete-time state equation with multiple input delays. To estimate the input delays, a new evolutionary computation called the artificial bee colony (ABC) algorithm is utilized. This algorithm is originally motivated from the social behaviors of honeybee organization, and it has been proven to be a powerful means for solving the optimized problem. Based on the proposed algorithm, the unknown system input delays can be further solved by minimizing a quadratic cost function of the system. Two illustrative examples are provided to verify the potential of the presented method in the time delay estimation. Some simulations containing different initial condition examinations and appearance of noises are further given. Numerical results show that the proposed method can do well in the multiple inputs delay estimation of discrete-time state equations.

## 1. Introduction

Among numerous evolutionary computations developed in recent years, artificial bee colony (ABC) algorithm has attracted many researchers' considerable attentions and also been applied in a variety of engineering optimization problems. It was initially developed by Karaboga and spread out by his research teams [1–4], and it is a population-based optimization algorithm like other evolutionary computations. Thus, this algorithm is in parallel to seek for the global or near global solution of the optimized problem, not a single direction search such as the gradient descent method. The ABC algorithm is motivated from the physical bee swarm behavior and possesses a powerful searching capacity on optimizations. In the real honeybee colony, there are three different types of foragers which are the employed bees, onlookers, and scouts [1]. The employed bee is a bee that is going to the food source visited by itself previously, the onlooker is a bee that is waiting on the dance area for making decision to select some food source, and the scout bee means that a bee is carrying out random search for food sources. Furthermore, each employed bee goes to one food source, and, therefore, this implies that the number of employed bees is identical with the number of food sources. An employed bee will

become a scout when her food source is exhausted by herself and onlooker bees.

In [1], Karaboga and Basturk have shown that the proposed algorithm is superior to some evolutionary algorithms, for example, genetic algorithm, particle swarm optimization, particle swarm inspired evolutionary algorithm, and so on. A large number of engineering applications combined with the ABC algorithm have been developed in recent years [5–11]. In [5], a nonlinear chemical CSTR process system was controlled by using proportional-integral-derivative (PID) controller from the stable equilibrium point to unstable equilibrium point. The design method for tuning PID control gains is the ABC algorithm. In [7], the authors proposed an optimal design method for the reactive power flow in which the active power loss was minimized by the ABC approach. In machining operations, selection of cutting parameters is an essential task for reducing the cost of products. Yildiz developed an optimization approach by using artificial bee colony algorithm for optimal selection of cutting parameters in multipass turning operations. Simulation results revealed that the ABC algorithm outperforms other optimization techniques [8]. Besides, Huang and Liu presented the ABC algorithm to enhance the fault section estimation performance in power systems [9]. The proposed method can

reduce the possibility of inaccurate diagnosis and find the accurate fault section among voluminous alarms.

Another issue considered in this paper is concerned with the time delay estimation. Time delay problem often occurs in many physical situations such as sensor array, sonar, medical imaging, and communications because of the inherent system dynamics and instrumentations. Especially for the control engineering and system identification, the time delay information is of importance. It is attractive and practical to directly estimate these delays according to the measured data rather than making an attempt to obtain them from a physical understanding of systems [12]. To cope with time delay estimation, many techniques were developed in the literature [12–17]. In [14], the authors proposed a novel time delay estimation approach by sliding discrete Fourier transform, and the maximum magnitude of the spectrum and its corresponding phase offset were utilized. It was shown that the proposed method outperforms the standard correlation-based method. Shaltaf then developed a Neurofuzzy-based approach combined with discrete cosine transform (DCT) coefficients for time delay estimation problem [15]. However, a design method for system time delay estimations based on using ABC optimization algorithm seems not to be reported to date. Thus, this paper will focus on the issue of multiple time delays estimation and develop an ABC-based approach to solve such a delay estimation problem. It is requested to define a quadratic cost function of the system at the beginning; the developed ABC algorithm then minimizes this function by means of certain mechanisms, and the actual time delays can be estimated correctly. Besides, the effect of initial condition of the algorithm on the estimation performance is also considered. The rest of this paper is organized as follows. In Section 2, the system formulation for multiple input time delays of discrete-time state equations is addressed. The concept and evolutionary computation of the ABC algorithm is firstly introduced in Section 3, and the design steps combined with ABC for such a time delay estimation is proposed. In Section 4, two illustrative examples are given to confirm the potential of the developed method, and some examinations including different independent runs and appearance of noises are further performed. Finally, Section 5 gives some simple conclusions.

## 2. System Formulations

In this section, the following system that is expressed by the discrete-time state equation with multiple input delays is considered [16]:

$$x(k+1) = Ax(k) + \sum_{i=1}^p B_i u_i(k - \tau_i), \quad (1)$$

where  $x(k) = [x_1(k), x_2(k), \dots, x_n(k)] \in R^n$  is the state variable vector of the system at time  $k$ ,  $u_i(k) \in R$ , for  $i = 1, 2, \dots, p$ , represents the  $i$ th system input,  $A \in R^{n \times n}$  is the coefficient matrix of the system and it is here assumed to be known,  $B_i \in R^n$ , for  $i = 1, 2, \dots, p$ , is also a known coefficient vector, and  $\tau_1, \tau_2, \dots, \tau_p$  denote the unknown integer time delays which will be estimated. Moreover, it is assumed that

the  $n$  system state variables in (1) are observed at each time  $k$ . In order to estimate these input delays accurately, the following mathematical model for (1) is given:

$$\hat{x}(k+1) = A\hat{x}(k) + \sum_{i=1}^p B_i u_i(k - \hat{\tau}_i), \quad (2)$$

where  $\hat{x}(k) = [\hat{x}_1(k), \hat{x}_2(k), \dots, \hat{x}_n(k)] \in R^n$  is the state variable vector of this estimated model at time  $k$ ,  $u_i(k) \in R$  is the same as one of (1),  $\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_p$  represent the estimates of actual input delays  $\tau_1, \tau_2, \dots, \tau_p$ , and they should be integer constants as well. The ABC algorithm is utilized such that the estimated values can approximate the actual ones. Before solving the estimation problem, a proper cost function of the system needs to be defined. In this study, a sum of squared error of system states is used over a given time frame as follows

$$\begin{aligned} J(\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_p) &= \sum_{k=0}^{N-1} [x(k) - \hat{x}(k)] [x(k) - \hat{x}(k)]^T \\ &= \sum_{k=0}^{N-1} e(k) \cdot e^T(k) \\ &= \sum_{k=0}^{N-1} \sum_{i=1}^p e_i^2(k), \end{aligned} \quad (3)$$

where  $e(k) = x(k) - \hat{x}(k) = [e_1(k), e_2(k), \dots, e_n(k)] \in R^n$  is the error vector of the system between  $x(k)$  and  $\hat{x}(k)$  and  $N$  is the sampling number. It can be seen that better estimation results are obtained by minimizing (3). For simplicity, these estimates of input time delays are collected to form a parameter vector as

$$\Theta = [\theta_1, \theta_2, \dots, \theta_p] = [\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_p]. \quad (4)$$

In the viewpoint of ABC algorithm, the cost function of (3) and the parameter vector of (4) are referred to as the nectar amount and the food source, respectively. Moreover, a food source represents a possible solution of the optimized problem and a large number of food sources then forms the so-called population. From (3), it is clearly seen that the food sources greatly affect the cost function value. Our purpose is to evolve these food sources successively according to some mechanisms of ABC algorithm such that the best food source (i.e., this food source is with the lowest cost function value) is obtained. Next section further introduces the detailed ABC algorithm and its designing steps on time delay estimation.

## 3. ABC Algorithm and Its Application to Time Delay Estimation

In recent years, Karaboga and his teams have developed a large number of literatures regarding the ABC algorithm and its application to diverse engineering fields such as cluster [3], parameter optimization [18], and symbolic regression [19]. They already showed that the ABC algorithm is an efficient means for solving the optimization problems and is superior

to other evolutionary computations like GA, PSO, and DE algorithms. The foragers of honeybee society consist of the following behavior models [1, 5]. Firstly, a set of food source positions are chosen at random by bees, and the nectar amount related to each food source is calculated. These honeybees then come back to the hive to share the nectar information of sources with other bees that are waiting on the dance area. Secondly, each of employed bees, after sharing the information, still goes to the neighborhood of the original food source which was visited by herself previously and then selects a new food source by visual information. Thirdly, the onlooker prefers a food source area depending on the nectar information shared by employed bees on the dance area. While the nectar amount of a food source increases, the probability of choosing such a food source by an onlooker increases as well. The onlooker bee then flies to arrive at the selected area and seeks for a new food source around this area. Visual information is based on the comparison of food source positions. Lastly, a scout bee will randomly find out a new food source to take the place of the previous one when its nectar is abandoned by the honeybee. The artificial bee colony is developed by mimicking such model of foraging bees.

Following the above descriptions, some central components of the ABC algorithm need to be explained in detail [1, 5].

**3.1. Food Source.** In the ABC algorithm, the notation for the  $i$ th food source is denoted by  $\Theta_i$ . Each of food sources represents a possible candidate solution to the optimized problem, and note that the number of food sources is the same with that of employed bees. As can be seen from (4), for time delay estimation problem, the  $i$ th food source can be further expressed by  $\Theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{ip}]$  consisting of  $p$  elements.

**3.2. Population Size.** A population is composed of numerous food sources, and here it is assumed that the size of population is  $H$ . Thus, for each food source position, it can be denoted by  $\theta_{ij}$  for  $i = 1, 2, \dots, H$  and  $j = 1, 2, \dots, p$ .

**3.3. Probability Related to Food Source.** In the ABC algorithm, a probability value needs to be evaluated for each of food sources. This probability is concerned with its cost function as defined in (3) and determines whether an onlooker bee chooses a new food source or not. The probability related to the  $i$ th food source  $\Theta_i$  is defined by

$$P_r(\Theta_i) = \frac{J(\Theta_i)}{\sum_{k=1}^H J(\Theta_k)}, \quad \text{for } i = 1, 2, \dots, H, \quad (5)$$

where  $J(\Theta_i)$  is the cost function evaluated by  $\Theta_i$  (the nectar amount of the  $i$ th food source). From (5), it can be realized that such a probability value is proportional to the cost function of every food source.

**3.4. New Food Source Position.** A new food source position can be generated around the present one  $\theta_{ij}$  in accordance with the following equation:

$$\theta_{ij}^{\text{new}} \leftarrow \theta_{ij} + \phi_{ij}(\theta_{ij} - \theta_{mj}), \quad (6)$$

for  $i = 1, 2, \dots, H, \quad j = 1, 2, \dots, p, \quad m \neq i,$

where  $\theta_{ij}^{\text{new}}$  is the new position after updating,  $\phi_{ij} \in [-2, 2]$  is a uniformly random number, which greatly dominates the making of a new food source position around  $\theta_{ij}$ , and  $m \in \{1, 2, \dots, H\}$  is a random integer constant as well but  $m \neq i$ . Equation (6) represents a comparison of neighbor food positions visually by foraging honeybees, and the ABC algorithm uses (6) to achieve the optimization for a given problem. Moreover, it should be noticed for time delay estimation problem that the estimates  $\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_p$  of the mathematical model equation (2) are requested to be integer constants; therefore, it is necessary to take the integer part of every food source position to evaluate the mathematical model and the cost function of each food source after executing (6) and for initial positions.

**3.5. Abandoned Food Source.** In the real-world situation, the food source may be further replaced by a new food source when its nectar has been abandoned by honeybees. In the ABC algorithm, therefore, if the cost function derived from some food source cannot be improved through an assigned number of generations (here called *limit*), such a food source is abandoned. After that, a new food source instead of the abandoned one will be produced from the original interval.

**3.6. Termination Condition.** This algorithm stops if the number of generations achieves the maximum allowable number  $G$  which is assigned by the designer.

Designing steps for solving the time delay estimation problem based on ABC algorithm can be summarized in the following.

**Data.** A discrete-time state equation with some unknown input delays described by (1), a mathematical model of (2) for modeling (1), control input  $u_i(k) \in R$  for  $i = 1, 2, \dots, p$ , the sampling number  $N$ , population size  $H$ , parameter *limit*, and number of generations  $G$ .

**Goal.** Accurately estimate the real input delays  $\tau_1, \tau_2, \dots, \tau_p$  of (1) by using the proposed ABC algorithm to minimize the cost function defined in (3).

**Step 1.** In the beginning, a population consisting of  $H$  food sources is randomly established from the interval  $[0, 1]$ .

**Step 2.** If an assigned number of generations  $G$  is achieved, then the algorithm stops.

**Step 3.** Calculate the cost function  $J(\Theta_i)$  of every food source  $\Theta_i$  by using (3).

**Step 4.** Evaluate every probability  $P_r(\Theta_i)$  of  $\Theta_i$  by using (5).

**Step 5.** Generate a random number from  $[0, 1]$ , and if the probability  $P_r(\Theta_i)$  is less than this generated number, then a new food source  $\Theta_i^{\text{new}} = [\theta_{i1}^{\text{new}}, \theta_{i2}^{\text{new}}, \dots, \theta_{ip}^{\text{new}}]$  is derived by means of (6).

**Step 6.** Further calculate  $J(\Theta_i^{\text{new}})$ , and let this new food source  $\Theta_i^{\text{new}}$  replace the original  $\Theta_i$  if  $J(\Theta_i^{\text{new}}) < J(\Theta_i)$ ;

otherwise, the present food source  $\Theta_i$  still remains, and the new one is discarded.

*Step 7.* When the number of generations is equal to the number, *limit*, check every  $\Theta_i$  whether its positions have been modified or not. If not yet, this food source is removed, and correspondingly a new food source is generated from the initial interval  $[0, 1]$ .

*Step 8.* Go back to Step 2.

#### 4. Illustrative Examples

This section will show the applicability of the ABC-based approach to the time delay estimation by using two illustrative examples. For all the following simulations, the related parameter setting for the ABC algorithm is given by the population size  $H = 40$ , *limit* = 20, number of generations  $G = 100$ , and sampling number  $N = 200$  in (3).

*Example 1.* Let us consider the first example which is a discrete-time state equation with three state variables and two system inputs described by [16]

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} &= \begin{bmatrix} 0.05 & 1 & 0 \\ -0.25 & 0.05 & 0 \\ 0.025 & 0.025 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \\ &+ \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix} u_1(k-6) + \begin{bmatrix} 1 \\ 0.25 \\ 0.5 \end{bmatrix} u_2(k-4), \end{aligned} \quad (7)$$

where  $\tau_1 = 6$  and  $\tau_2 = 4$  are the real input time delays and will be estimated. Correspondingly, a mathematical model for (7) is constructed by

$$\begin{aligned} \begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \\ \hat{x}_3(k+1) \end{bmatrix} &= \begin{bmatrix} 0.05 & 1 & 0 \\ -0.25 & 0.05 & 0 \\ 0.025 & 0.025 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \\ \hat{x}_3(k) \end{bmatrix} \\ &+ \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix} u_1(k-\hat{\tau}_1) + \begin{bmatrix} 1 \\ 0.25 \\ 0.5 \end{bmatrix} u_2(k-\hat{\tau}_2), \end{aligned} \quad (8)$$

where  $\hat{\tau}_1$  and  $\hat{\tau}_2$  denote the estimates of  $\tau_1$  and  $\tau_2$ , respectively. It is assumed that the initial values are given by  $x_1(0) = 0$ ,  $x_2(0) = 0$ , and  $x_3(0) = 0$  for the system and  $\hat{x}_1(0) = 0$ ,  $\hat{x}_2(0) = 0$ , and  $\hat{x}_3(0) = 0$  for the model. In addition, system inputs  $u_1$  and  $u_2$  are two uniformly random numbers generated from the interval  $[-1, 1]$ . After executing the proposed ABC algorithm with these data, simulation results are listed in Table 1 and displayed in Figure 1, respectively. In Table 1, it is clearly shown that the estimates  $\hat{\tau}_1$  and  $\hat{\tau}_2$  are all exactly the same with  $\tau_1 = 6$  and  $\tau_2 = 4$  from ten runs with different initial conditions of the algorithm. Convergence trajectories of  $\tau_1$  and  $\tau_2$  with respect to number of generations is shown in Figure 1 only for Run 1. After performing about 25 iterations, the actual time delays can be correctly estimated.

TABLE 1: Estimation results for Example 1.

	$\hat{\tau}_1$	$\hat{\tau}_2$
Run 1	6	4
Run 2	6	4
Run 3	6	4
Run 4	6	4
Run 5	6	4
Run 6	6	4
Run 7	6	4
Run 8	6	4
Run 9	6	4
Run 10	6	4

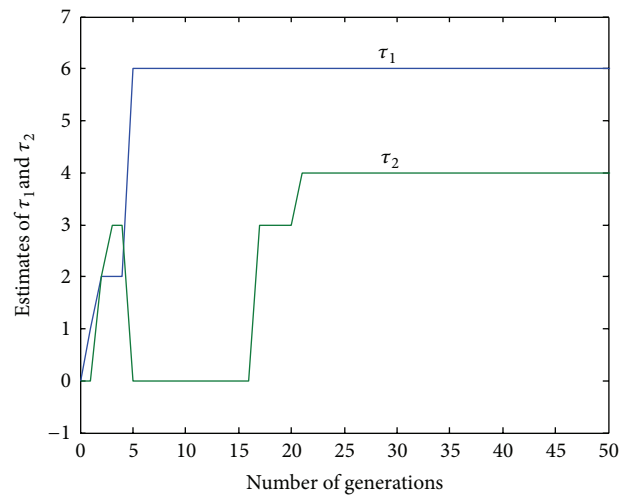


FIGURE 1: Estimates of  $\tau_1$  and  $\tau_2$  for Run 1 of Example 1.

*Example 2.* In the second example, a discrete-time state equation with two state variables and three system inputs is considered and can be expressed by

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} -0.1 & -0.01 \\ 0.05 & -0.2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \end{bmatrix} u_1(k-3) \\ &+ \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix} u_2(k-4) + \begin{bmatrix} -2 \\ 1 \end{bmatrix} u_3(k-1), \end{aligned} \quad (9)$$

where three input delays  $\tau_1 = 3$ ,  $\tau_2 = 4$ , and  $\tau_3 = 1$  are given, respectively. As the same as Example 1, three system inputs are all generated from  $[-1, 1]$  randomly to excite the responses, and all initial values are assumed to be zero. However, in this case the effect of measurement noises on estimation performance is involved. It is assumed that two Gaussian signals  $N_1(0, 0.1)$  and  $N_2(0, 0.1)$  shown in Figure 2 are incorporated into the system states  $x_1$  and  $x_2$ , respectively, as measurement noises. Also, ten different runs are illustrated for examining the robustness of the developed algorithm. Table 2 lists the estimation results for Example 2, and all simulation results from any independent run reveal

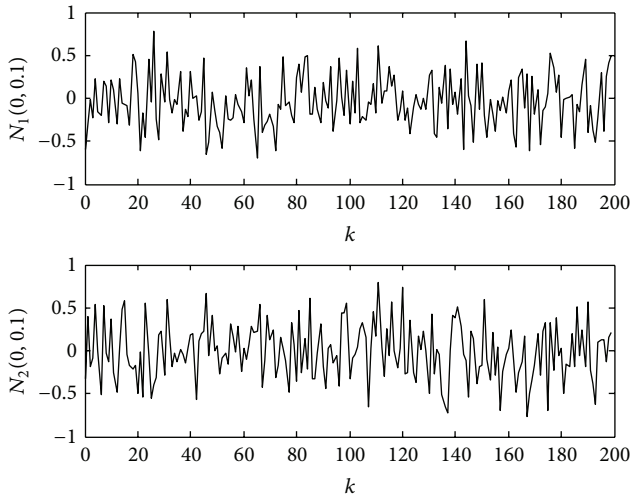


FIGURE 2: Gaussian noises  $N_1(0, 0.1)$  and  $N_2(0, 0.1)$  for Example 2.

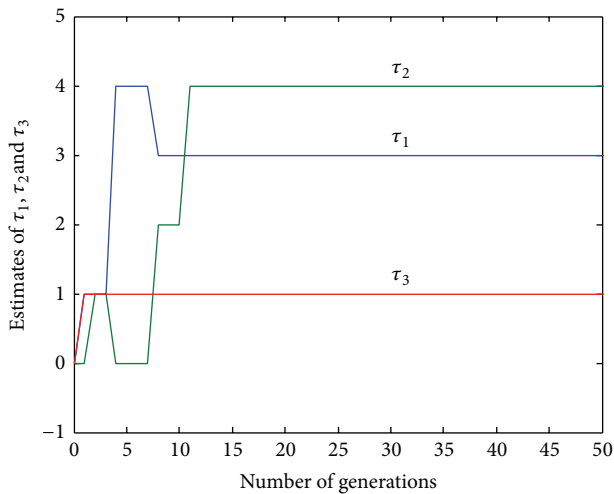


FIGURE 3: Estimates of  $\tau_1$ ,  $\tau_2$ , and  $\tau_3$  for Run 1 of Example 2.

that the input time delays can be correctly estimated even though measured noises appear. Figure 3 then displays the convergence of time delay estimations for Run 1. As can be seen from the figure, the estimates all approach the actual values after some generations.

### 5. Conclusions

This paper has successfully developed an estimation approach for the discrete-time state equation with multiple input time delays. The estimation method is based on the artificial bee colony algorithm which is one of new evolutionary computations. By mimicking the social behaviors of honeybees, such an algorithm can be used to solve the system optimization problem. In this paper, the detailed descriptions for ABC algorithm have been presented, and the design steps combined with the proposed ABC algorithm have also been given for multiple input time delay estimations. It is concluded form

TABLE 2: Estimation results for Example 2.

	$\hat{\tau}_1$	$\hat{\tau}_2$	$\hat{\tau}_3$
Run 1	3	4	1
Run 2	3	4	1
Run 3	3	4	1
Run 4	3	4	1
Run 5	3	4	1
Run 6	3	4	1
Run 7	3	4	1
Run 8	3	4	1
Run 9	3	4	1
Run 10	3	4	1

two illustrative examples that the ABC-based approach has a satisfactory outcome for solving the time delay estimation problem.

### References

- [1] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [2] N. Karaboga, "A new design method based on artificial bee colony algorithm for digital IIR filters," *Journal of the Franklin Institute*, vol. 346, no. 4, pp. 328–348, 2009.
- [3] D. Karaboga and C. Ozturk, "A novel clustering approach: artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 652–657, 2011.
- [4] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, 2012.
- [5] W. D. Chang, "Nonlinear CSTR control system design using an artificial bee colony algorithm," *Simulation Modelling Practice and Theory*, vol. 31, pp. 1–9, 2013.
- [6] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [7] K. Ayan and U. Kiliç, "Artificial bee colony algorithm solution for optimal reactive power flow," *Applied Soft Computing Journal*, vol. 12, no. 5, pp. 1477–1482, 2012.
- [8] A. R. Yildiz, "Optimization of cutting parameters in multi-pass turning using artificial bee colony-based approach," *Information Sciences*, vol. 220, pp. 399–407, 2013.
- [9] S. J. Huang and X. Z. Liu, "Application of artificial bee colony-based optimization for fault section estimation in power systems," *Electrical Power and Energy Systems*, vol. 44, no. 1, pp. 210–218, 2013.
- [10] A. Khorsandi, S. H. Hosseini, and A. Ghazanfari, "Modified artificial bee colony algorithm based on fuzzy multi-objective technique for optimal power flow problem," *Electrical Power Systems Research*, vol. 95, pp. 206–213, 2013.
- [11] A. R. Yildiz, "A new hybrid artificial bee colony algorithm for robust optimal design and manufacturing," *Applied Soft Computing*, vol. 13, no. 5, pp. 2906–2912, 2013.
- [12] S. Selvanathan and A. K. Tangirala, "Time-delay estimation in multivariate systems using Hilbert transform relation and

- partial coherence functions,” *Chemical Engineering Science*, vol. 65, no. 2, pp. 660–674, 2010.
- [13] B. Ni, D. Xiao, and S. L. Shah, “Time delay estimation for MIMO dynamical systems—with time-frequency domain analysis,” *Journal of Process Control*, vol. 20, no. 1, pp. 83–94, 2010.
- [14] S. Assous and L. Linnett, “High resolution time delay estimation using sliding discrete Fourier transform,” *Digital Signal Processing*, vol. 22, no. 5, pp. 820–827, 2012.
- [15] S. Shaltaf, “Neuro-fuzzy based time-delay estimation using DCT coefficients,” *ISA Transactions*, vol. 46, no. 1, pp. 21–30, 2007.
- [16] R. Waschburger and R. K. H. Galvao, “Time delay estimation in discrete-time state-space models,” *Signal Processing*, vol. 93, no. 4, pp. 904–912, 2013.
- [17] B. S. Chen and J. C. Hung, “A global estimation for multichannel time-delay and signal parameters via genetic algorithm,” *Signal Processing*, vol. 81, no. 5, pp. 1061–1067, 2001.
- [18] B. Akay and D. Karaboga, “A modified artificial bee colony algorithm for real-parameter optimization,” *Information Sciences*, vol. 192, pp. 120–142, 2012.
- [19] D. Karaboga, C. Ozturk, N. Karaboga, and B. Gorkemli, “Artificial bee colony programming for symbolic regression,” *Information Sciences*, vol. 209, pp. 1–15, 2012.