Parameter Identification of A Four-dimensional Chaotic System Using Real-valued Immune Algorithms

Wei-Der Chang[*]
Department of Computer and Communication, Shu-Te University
Kaohsiung 824, Taiwan

Received 23 January 2014

Accepted 22 September 2014

Abstract

This paper presents a new immune algorithm to solve the parameters identification problem of the four-dimensional chaotic system. The proposed algorithm is directly to utilize the real-value operation to fulfill the numerical optimization, which is quite different from the traditional immune algorithm with binary operations. For most cases of engineering applications, this kind of algorithm seems to be more applicable. In order to achieve the real-value manipulation, the modified similarity definition and mutation mechanism have been developed in place of the originals. By the proposed method, the parameter identification of a special nonlinear system, four-dimensional chaotic system, is solved. Finally, simulation results will show the feasibility of the proposed algorithm in the parameter identification of nonlinear chaotic system.

Keywords: Immune Algorithm, Real-valued Manipulation, Parameter Identification, Four-dimensional Chaotic System.

1. Introduction

The immune algorithm is a recent developed algorithm which mimics the immune theory of life science. It is one of heuristic evolutionary algorithms and has been successfully applied to various optimization problems in recent years[1-7]. The immune system is a distributed novel-pattern detection system throughout the body and it has many useful components positioned in the strategic locations. The function of the immune system is to recognize and categorize all cells inside the body as self or non-self cells. An appropriate defensive mechanism is stimulated to cope with non-self cells. It is able to distinguish foreign antigen from own body cells via evolution operations. In the immune responses, the lymphocyte is the main type of immune cell and it possesses some attributes, such as specificity, diversity, memory, and adaptability[5]. It consists of two different types of T and B. The surfaces of both cells have receptor molecules to recognize disease-causing pathogen (antigen). The B-cell receptor molecule is called an antibody. By invasion of pathogens a number of immune cells that recognize these pathogens will proliferate. Some of them become effecter cells (plasma cells) and others are maintained as memory cells[6]. This is so-called clonal selection. The artificial immune algorithm that mimics the original immune system possesses many properties, such as automatic antigen identification, feature extraction, antibody diversification, distributed detection, learning, memory and self-programming, and so on. Hence this algorithm has a great potential in intelligent computation applications[3].

With the advantages mentioned above, many researchers have utilized the immune algorithm to solve several engineering problems, especially for system optimization. In Ref.2, a modified immune clonal selection algorithm has been developed for performing

[*] Corresponding Author: wdchang@stu.edu.tw

This work was partially supported by the Ministry of Science and Technology of Taiwan under grant MOST 103-2221-E-366-004

Co-published by Atlantis Press and Taylor & Francis

the design of stack filter configuration. This algorithm has some features including preventing from prematurity and fast convergence speed. The designed stack filter is then utilized to restore images corrupted by uncorrelated additive noise. In Ref.5, the authors employed the artificial immune algorithm to design the digital filter with the infinite impulse response (IIR) structure. At the beginning, all of filter coefficients must be transformed to the corresponding binary representation. These binary strings are then performed by mechanisms of the immune algorithm such that certain cost function is minimized. Finally, the obtained binary strings are recovered to be real numbers. In Ref.1, the immune algorithm was used to enhance the effectiveness of the distribution automation system (DAS) in the power company. The total cost of customer service outage and investment cost of line switches is minimized by the proposed method such that the placement of switching devices is derived optimally. Similar to the application of Ref.1, the authors used a modified version of real immune algorithm to simultaneous coordinated tuning of power system stabilizers in a power system[8]. Furthermore, for feature selection and classification of ECG signals, the authors proposed a hybrid immune algorithm[9]. Simulation results show that the proposed suppression method is better than the traditional one.

Another issue of this paper focuses on the nonlinear chaotic system. The dynamics of chaos contains lots of natural properties such as extremely sensitive to initial conditions, noise-like dynamic signal and fractal properties of the motion in the phase portrait and so on. Due to these features, the chaotic system is more applicable to some special applications, for example, chaotic synchronization[10], image encryption[11], and signal transmissions[12]. In addition, several control technologies including neural fuzzy control[13], fuzzy control[14], impulsive control[15], and sliding mode control[16] were developed to control the chaotic systems for some purposes like parameter identification and chaos synchronization. Most of chaotic systems always consist of three system states (three dimensions) such as famous Rossler system, Chen system, and Lü system. In recent years, a few new chaotic systems with four-dimensional states were presented[17-19] and they were analyzed and verified by Lyapunov exponents and bifurcation diagrams. In this study, the simulated example is the Lorenz-Stenflo chaotic system. It is a complex chaotic system with four state variables. Until now, the related studies on such a new chaotic system have rarely reported. Because of this reason, this paper will focus on the issue of Lorenz-Stenflo chaotic system and also develop a novel real-valued immune algorithm to solve the parameter estimation problem for this kind of chaotic system. The remainder of this paper is organized as follows. The description for the four-dimensional chaotic system is given in Section 2. In Section 3, a simple introduction to the general binary immune algorithm is presented. Section 4 will show the detail of the proposed real-valued immune algorithm. The full design steps for parameter estimation of such a four-dimensional chaotic system are also given. In Section 5, experiment results are demonstrated by different sets of initial population tests to validate the feasibility of our proposed method. Finally, some simple conclusions are addressed in Section 6.

2. Description of Four-dimensional Chaotic System

Lorenz found the first chaotic attractor in a three-dimensional autonomous system from investigating the atmospheric convection in 1963. This Lorenz system becomes a paradigm of chaos search. Later, other chaos systems were successively proposed such as the Rossler system, Lu system, Chen system, and unified chaotic system and so on. All systems mentioned above fully belong to the type of three-dimensional state differential equations. In this study, we consider a complex chaotic system with four dimensions, i.e., it has four system states and is called the Lorenz-Stenflo system. It is extended from Lorenz system and its nonlinear autonomous state equation is described by[17]

$$\dot{x}_1 = \alpha(x_2 - x_1) + \gamma x_4 \tag{1a}$$

$$\dot{x}_2 = x_1(\lambda - x_3) - x_2 \tag{1b}$$

$$\dot{x}_3 = x_1 x_2 - \beta x_3 \tag{1c}$$

$$\dot{x}_4 = -x_1 - \alpha x_4 \tag{1d}$$

where x_i , $i = 1, 2, 3, 4$, is the state variable of the system, α , β , γ , and λ are the Prandtl number, geometric parameter, rotation number, and Rayleigh number, respectively, and they all are positive constants. It can be seen from Eq. (1) that the four-dimensional chaotic system can be reduced to the Lorenz system exactly if both state x_4 and parameter γ are omitted. Fig. 1 shows the phase portrait of x_1 and x_4 of this autonomous chaotic system and the phase portrait of x_2 and x_3 is displayed in Fig. 2, respectively, under the parameters $\alpha = 1$, $\beta = 0.7$, $\gamma = 1.5$, and $\lambda = 26$. To achieve the parameter identification purpose, it needs to add a control signal u into Eq. (1) for stimulating the system response. Here let the state variable x_1 be the system output and Eq. (1) can be reformulated as

$$\dot{x}_1 = \alpha(x_2 - x_1) + \gamma x_4 \tag{2a}$$

$$\dot{x}_2 = x_1(\lambda - x_3) - x_2 \tag{2b}$$

$$\dot{x}_3 = x_1 x_2 - \beta x_3 + u \tag{2c}$$

$$\dot{x}_4 = -x_1 - \alpha x_4 \tag{2d}$$

$$y = x_1 \tag{2e}$$

where y denotes the system output and is assumed to be available for measurement. The aim of this study is to develop a new method to identify these parameters α, β, γ, and λ according to the input signal u and output signal y. Furthermore, a mathematical model is constructed and it is expected to match the actual chaotic system of Eq. (2) approximately. It is expressed by

$$\dot{\hat{x}}_1 = \theta_1(\hat{x}_2 - \hat{x}_1) + \theta_3 \hat{x}_4 \tag{3a}$$

$$\dot{\hat{x}}_2 = \hat{x}_1(\theta_4 - \hat{x}_3) - \hat{x}_2 \tag{3b}$$

$$\dot{\hat{x}}_3 = \hat{x}_1 \hat{x}_2 - \theta_2 \hat{x}_3 + u \tag{3c}$$

$$\dot{\hat{x}}_4 = -\hat{x}_1 - \theta_1 \hat{x}_4 \tag{3d}$$

$$\hat{y} = \hat{x}_1 \tag{3e}$$

where $\hat{x}_i$, $i = 1, 2, 3, 4$, is the model state variable, the input signal u is the same with that of Eq. (2c), $\hat{y}$ is the model output, and θ_1, θ_2, θ_3, and θ_4 are the model parameters which are expected to approximate the actual parameters α, β, γ, and λ, respectively, by the proposed method.

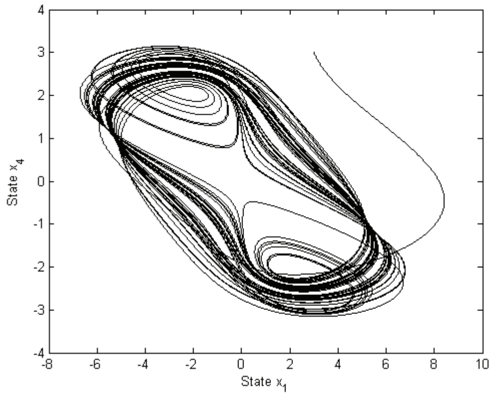

Fig. 1 Phase portrait of x_1 and x_4.

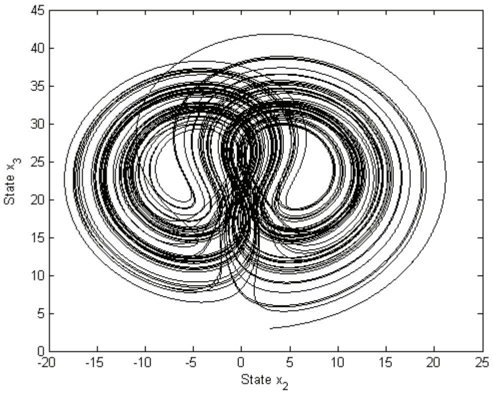

Fig. 2 Phase portrait of x_2 and x_3.

3. Binary Immune Algorithm

The immune algorithm is one of evolutionary computations and it is based on the combination of some properties of natural immune system and engineering application. This algorithm has some advantages for solving optimization problem, for instance, fast convergence, without derivative information regarding the cost function, and a great diversity of feasible space of solution. In Ref.5, Kalinli and Karaboga presented the design steps of the binary immune algorithm as follows.

Step 1. Generate an initial population.
Step 2. Evaluate the stimulation level of each cell.
Step 3. Calculate the similarity of each cell with the highest stimulated cell.
Step 4. Divide population into two sub-populations according to these similarities.
Step 5. Apply somatic mutation to cells of the first sub-population.

Step 6. Calculate the similarity of each cell with other cells.
Step 7. Remove some of the cells stimulated at low level from the population.
Step 8. Generate new cells and insert some of them into the population.
Step 9. If the pre-specific number of generations is attained, then the algorithm stops; otherwise, go to Step 2.

In the beginning of the algorithm, an initial population consisting of many cells is randomly generated. Each of cells is formed with a binary string and it represents a candidate solution of the optimized problem. The cost function corresponding to each cell is then evaluated. Based on the values, all cells of the population need to be arranged in order and the cell with the best cost function is referred to as the antigen. In the meantime, population is further divided into two sub-populations according to the similarity. The similarity of each cell with the assigned antigen is calculated and cells with better matching score by a K-T tournament selection form the first sub-population. Furthermore, a one-point mutation operator is applied to each cell in this sub-population. The operation of mutation is that every bit within the cell is checked and the bit is reversed if a random number chosen from is less than the mutation rate. This resulting cell will survive if its corresponding similarity with antigen is better than that of present one. On the other hand, the similarity of each cell in the second sub-population is also evaluated. Some of them with worse similarity are removed from the population and in the meantime some new random cells are produced to keep the same population size[5].

4. The Proposed Real-valued Immune Algorithm

Following the binary immune algorithm, here we will develop a modified immune algorithm with real-valued operations. Although binary representations are much more straightforward in comparison with real numbers in some cases such as job-shop scheduling, traveling salesman problem, and digital systems in nature. But, a real-valued algorithm is more applicable for most of real engineering problems since the transformation between real numbers and binary strings is not necessary. It can mostly reduce the complexity of computations and prompt the numerical accuracy. Also, such algorithm is more easily implemented to computer programs than the binary algorithm. In the proposed immune algorithm, some important terminologies are described in the following.

Cell Θ: Here the cell Θ is directly constructed by a real-valued vector which contains n elements $\Theta = [\theta_1, \theta_2, \cdots, \theta_n]$, and it is also a candidate solution in the optimization problem. Each element θ_i is constrained by the search interval $[\theta_{i\_\min}, \theta_{i\_\max}]$ during evolutions. If the obtained value is beyond the interval, then the corresponding boundary value is taken to replace this value.

Population: Many cells make up a population and here let the population size be *H*.

Cost function: In the optimization procedure, the performance of each cell is only evaluated by the cost function. To deal with the continuous-time optimization problem, the integral of absolute error (*IAE*) is suitably taken as the cost function and it is expressed by

$$IAE = \int_0^{T_i} |y(t) - \hat{y}(t)| dt = \int_0^{T_i} |e(t)| dt \tag{4}$$

where T_i is the time of integration and e is the error signal between y and $\hat{y}$ defined in Eqs. (2e) and (3e), respectively.

Antigen: According to the value of cost function corresponded to each cell, the one with the best *IAE* is referred to as the antigen and denoted by $\Theta^* = [\theta_1^*, \theta_2^*, \cdots, \theta_n^*]$.

Similarity: Cell's similarity is defined by the Euclidean distance between this cell $\Theta = [\theta_1, \theta_2, \cdots, \theta_n]$ with the antigen $\Theta^* = [\theta_1^*, \theta_2^*, \cdots, \theta_n^*]$, i.e.,

$$S = \sqrt{(\theta_1^* - \theta_1)^2 + (\theta_2^* - \theta_1)^2 + \cdots + (\theta_n^* - \theta_n)^2} \tag{5}$$

The population will be divided into two sub-populations in accordance with the similarities.

Mutation: In the binary immune algorithm, the mutation operation is that an assigned bit is reversed between 0 and 1 when a random number is less than the mutation rate *m*. In our proposed algorithm, the following real-valued mutation is utilized:

$$\theta \leftarrow \theta + N\left(0, k\frac{S_i}{\sum_{i=1}^{H} S_i}\right) \tag{6}$$

where θ is the element of the ith cell Θ_i, $N(\cdot)$ represents a Gaussian distribution function with mean $\mu = 0$ and variance $\sigma^2 = kS_i \Big/ \sum_{i=1}^{H} S_i$, k is a positive constant, and S_i is the similarity of Θ_i.

Stopping criterion: The algorithm is terminated when the number of generations achieves the prescribed maximum allowable number *G*.

The design steps of the proposed real-valued immune algorithm for solving a four-dimensional chaotic parameter identification problem are summarized as follows:

Data: An unknown four-dimensional chaotic system described by Eq. (2), search interval $\lfloor \theta_{i\_\min}, \theta_{i\_\max} \rfloor$, population size *H*, time of integration T_i in Eq. (4), mutation rate *m*, positive constant *k* in Eq. (6), and number of generations *G*.

Goal: Use the proposed immune algorithm to minimize the cost function of Eq. (4) such that the unknown system parameters can be optimally solved.

Step 1. Construct an initial population with *H* cells from the search interval randomly.

Step 2. Calculate the cost function of each cell using Eq. (4) and find out the antigen Θ^* in the population.

Step 3. Evaluate the similarity of each cell with the antigen using Eq. (5).

Step 4. Divide the population into two sub-populations according to these similarities. The first $H/2$ cells with better similarities make up the first sub-population, then the remainder forms the other sub-population.

Step 5. For all cells in the first sub-population, perform the mutation of Eq. (6) and check whether the obtained value is beyond the search interval or not. If the resulting cell is better than the present one, then this new cell replaces it.

Step 6. Remove $H \times 20\%$ cells with worse similarities from the population and randomly generate $H \times 20\%$ new cells into the population.

Step 7. If the prescribed number of generations *G* is attained, then the algorithm stops. Otherwise, go to Step 2.

5. Experiment Results

The parameter identification of Lorenz-Stenflo chaotic system is illustrated in this section. It is expected that four actual chaotic parameters $\alpha = 1.0$, $\beta = 1.5$, $\gamma = 26.0$, and $\lambda = 0.7$ are identified by the proposed method. The software of programming the real-valued immune algorithm is the Version 5.02 of Borland C++. To simulate the continuous four-dimensional Lorenz-Stenflo system, the sampling time is set to 0.001. The control input for stimulating the system response here is the random number generated from the interval $[-1, 1]$. Besides, the system initial values are given by $x_1(0) = x_2(0) = x_3(0) = x_4(0) = 3.0$ for simulations, respectively. The related variable settings used in the proposed algorithm are listed in Tab. 1.

Tab. 1 Related variable settings utilized in the proposed immune algorithm.

Variables	Values
$\lfloor \theta_{i\_\min}, \theta_{i\_\max} \rfloor$	$0 \le \theta_1, \theta_2, \theta_4 \le 2$ and $20 \le \theta_3 \le 30$
Population size *H*	40
Integral time T_i	1.0
Mutation rate *m*	1.0
Positive constant *k*	0.001
Number of generations *G*	8000

Tab. 2 Numerical results for the first five runs (Run 1~Run 5).

	θ_1	θ_2	θ_3	θ_4
Run 1	1.0	1.5	25.9997	0.7
Run 2	0.999	1.5001	26.0028	0.6997
Run 3	0.9994	1.5003	26.0086	0.6994
Run 4	1.0001	1.5	25.9981	0.7002
Run 5	1.0003	1.4997	25.9961	0.7003

In the simulation, 100 different initial sets of populations are provided to validate the feasibility of the propose method. Fig. 3 displays the convergence trajectories of the first five runs with respect to the number of generations. It is clearly shown that all trajectories eventually approach to zero. This means that the parameter optimization can be achieved. The convergences of the estimated chaotic parameters θ_1, θ_2, θ_3, and θ_4 are then shown in Figs 4-7, respectively, and the final numerical results are listed in Tab. 2 for the first five runs. Furthermore, after 100 independent executions the mean and variance are evaluated by 0.999994, 0.699968, 1.499997, 26.00034, and 3.08853×10^{-7}, 3.11×10^{-7}, 8.15×10^{-8}, 4.91×10^{-5}, respectively, for θ_1, θ_2, θ_3, and θ_4. From these figures and numerical results, it is concluded that the proposed method is satisfactory in the parameter identification of four-dimensional Lorenz-Stenflo chaotic system.

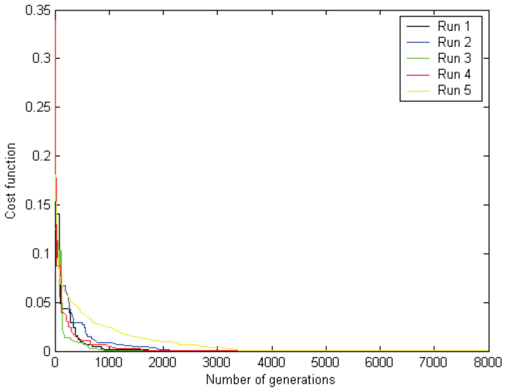

Fig. 3 Convergences of cost function for the first five runs (Run 1~Run 5).

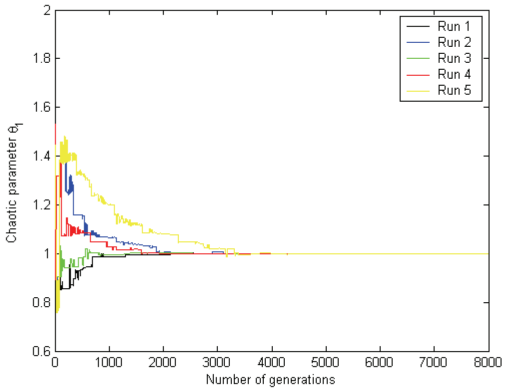

Fig. 4 Convergences of parameter θ_1 for the first runs (Run 1~Run 5).

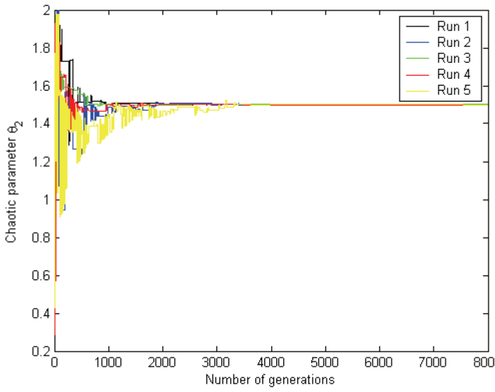

Fig. 5 Convergences of parameter θ_2 for the first five runs (Run 1~Run 5).

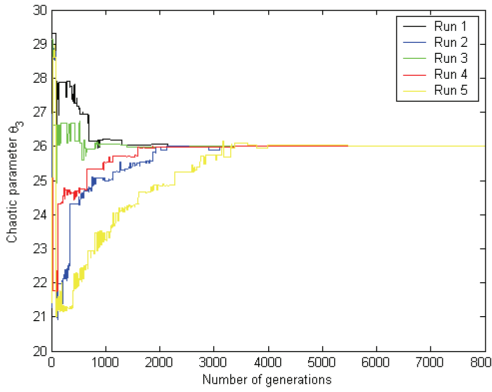

Fig. 6. Convergences of parameter θ_3 for the first five runs (Run 1~Run 5).

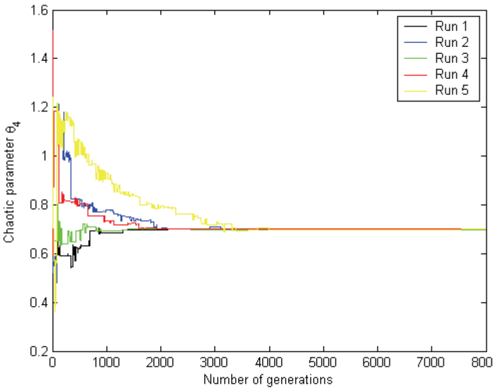

Fig. 7. Convergences of parameter θ_4 for the first five runs (Run 1~Run 5).

6. Conclusions and Future Study

This paper has successfully presented a modified immune algorithm to solve the parameter identification problem for a complex four-dimensional chaotic system. In order to meet the real-valued operations, we have modified the similarity definition and mutation mechanism of the original algorithm. In the proposed method, the unknown chaotic parameters can be directly evolved without extra coding and decoding conversions. Moreover, simulation results also reveal the effectiveness of the proposed real-valued immune algorithm. In the future work, we will attempt to apply the algorithm to other engineering fields such as filter design and controller parameter setting. Besides, the convergence analysis of the proposed method is also developed to further confirm the applicability in real implementations.

References

1. C. S. Chen, C. H. Lin, H. J. Chuang, C. S. Li, M. Y. Huang, C. W. Huang, Optimal placement of line switches for distribution automation systems using immune algorithm, *IEEE Transaction on Power Delivery*, 21 (2006), pp.1209-1217.
2. W. Dong, G. Shi, L. Zhang, Immune memory clonal selection algorithms for design stack filters, *Neurocomputing*, 70 (2007), pp.777-784.
3. Z. Guo, S. Wang, J. Zhuang, A novel immune evolutionary algorithm incorporating chaos optimization, *Pattern Recognition Letters*, 27 (2006), pp.2-8.
4. S. J. Huang, An immune-based optimization method to capacitor placement in a radial distribution system, *IEEE Transaction on Power Delivery*, 15 (2000), pp.744-749.
5. A. Kalinli, N. Karaboga, Artificial immune algorithm for IIR filter design, *Engineering Application of Artificial Intelligence*, 18 (2005), pp.919-929.
6. R. Tavakkoli-Moghaddam, A. Rahimi-Vahed, A. H. Mirzaei, A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness, *Information Science*, 177 (2007), pp.5072-5090.
7. Z. Zhang, Immune optimization algorithm for constrained nonlinear multiobjective optimization problems, *Applied Soft Computing*, 7 (2007), pp.840-857.
8. M. M. Farsangi, S. Kyanzadeh, S. Haidari, H. Nezamabadi-pour, Coordinated control of low-frequency oscillations using real immune algorithm with population management, *Energy Conversion and Management*, 51 (2010), pp.271-276.
9. M. Bereta, T. Burczynski, Comparing binary and real-valued coding in hybrid immune algorithm for feature selection and classification of EGG signals, *Engineering Application of Artificial Intelligence*, 20 (2007), pp.571-585.
10. H. H. Kuo, Y. Y. Hou, J. J. Yan, T. L. Liao, Reliable synchronization of nonlinear chaotic systems, *Mathematics and Computers in Simulation*, 79 (2009), pp.1627-1635.
11. X. Y. Wang, L. Yang, R. Liu, A. Kadir, A chaotic image encryption algorithm based on perceptron model, *Nonlinear Dynamics*, 62 (2010), pp.615-621.
12. J. P. Mangeot, F. Launay, P. Coirault, A multi-user transmission using chaotic carriers, *Communications in Nonlinear Science and Numerical Simulation*, 14 (2009), pp.2709-2719.
13. C. S. Chen, Quadratic optimal neural fuzzy control for synchronization of uncertain chaotic systems, *Expert Systems with Applications*, 36 (2009). pp.11827-11835.
14. J. H. Kim, C. W. Park, E. Kim, M. Park, Adaptive synchronization of T-S fuzzy chaotic systems with unknown parameters, *Chaos, Solitons & Fractals*, 24 (2005), pp.1353-1361.
15. Y. Liu, S. Zhao, T–S fuzzy model-based impulsive control for chaotic systems and its application, *Mathematics and Computers in Simulation*, 81 (2011), pp.2507-2516.
16. M. Heydari, H. Salarieh, M. Behzad, Stochastic chaos synchronization using Unscented Kalman–Bucy Filter and sliding mode control, *Mathematics and Computers in Simulation*, 81 (2011), pp.1770-1784.
17. J. A. Laoye, U. E. Vincent, S. O. Kareem, Chaos control of 4D chaotic systems using recursive backstepping nonlinear controller, *Chaos, Solitons & Fractals*, 39 (2009), pp.356-362.
18. G. Qi, S. Du, G. Chen, Z. Chen, Z. Yuan, On a four-dimensional chaotic system, *Chaos, Solitons & Fractals*, 23 (2005), pp.1671-1682.
19. Z. Wang, X. Shi, Synchronization of a four-dimensional energy resource system via linear control, *Communications in Nonlinear Science and Numerical Simulation*, 16 (2011), pp.463-474.