CHAOTIC SYNCHRONIZATION BASED ON PID CONTROL AND A NORMATIVE PARTICLE SWARM OPTIMIZATION APPROACH

Leandro dos Santos Coelho, leandro.coelho@pucpr.br
Pontifical Catholic University of Paraná, PUCPR
Industrial and Systems Engineering Graduate Program, LAS/PPGEPS
Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Paraná, Brazil

Rafael Bartnik Grebogi, rafagrebogi@gmail.com
Pontifical Catholic University of Paraná, PUCPR
Undergraduate in Mechatronic Engineering
Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Paraná, Brazil

Diego Luis de Andrade Bernert, dbernert@gmail.com
Pontifical Catholic University of Paraná, PUCPR
Industrial and Systems Engineering Graduate Program, LAS/PPGEPS
Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Paraná, Brazil

Abstract. *Recently, the investigation of synchronization and control problem for discrete chaotic systems has attracted much has stimulated a wide range of research activity, including both theoretical studies and practical applications. This paper deals with the tuning of a proportional-integral-derivative (PID) controller based on an improved particle swarm optimization (PSO) method based on normative knowledge (NPSO), a source of knowledge in cultural algorithms, for synchronization of two identical discrete chaotic systems subject to different initial conditions. PSO is a kind of a population-based algorithm and is motivated by the simulation of social behavior instead of the survival of the fittest individual (solutions of an optimization problem). It is a population-based evolutionary algorithm. Similar to the other population-based evolutionary algorithms, PSO is initialized with a population of random solutions. Unlike the most of the evolutionary algorithms, each potential solution (individual) in PSO is also associated with a randomized velocity, and the potential solutions, called particles, are then "flown" through the problem space. Numerical simulations are given to show the effectiveness of the proposed synchronization method using NPSO.*

Keywords*: optimization, swarm intelligence, particle swarm optimization, chaotic synchronization, chaotic systems, nonlinear dynamics, cultural algorithms.*

1. INTRODUCTION

Chaos is a special feature of parametric nonlinear dynamical systems (Strogatz, 2000; Peitgen *et al*., 2004). The theory of chaotic dynamics has a deep impact on our understanding of Nature. The strength of this theory comes from its generality, in that it is not limited to a particular equation or scientific domain. It should be viewed as a conceptual framework with which one can capture properties of systems with complicated behavior. Obviously, such a general framework cannot describe a system down to its most intricate details, but it is a useful and important guideline on how a certain kind of complex systems may be understood and analyzed (Collet and Eckmann, 2006).

Chaos, an apparently disordered behavior that is nonetheless deterministic, is a universal phenomenon that occurs in many nonlinear systems. It is featured by highly unstable motion of deterministic systems in a bounded region of the phase space. High instability means that the distance of two nearby orbits increases exponentially with time, which is a result of the extreme sensitivity of chaotic systems to the initial conditions. The Lyapunov exponents quantify this property (Lu *et al*., 2005).

During recent years, the problem of chaos control and synchronization have received considerable attentions of many researchers due to its great potential in technological applications, leading to the development of many methods.

Recent interest has peaked by the pioneering work of Pecora and Carroll (1990) showing that a drive signal from a chaotic system could be used to synchronize a second chaotic system. A basic configuration for chaos synchronization is the master-slave (drive-response) pattern, where the response chaotic system must track the drive chaotic trajectory.

Among various control methods, control techniques based on proportional-integral-derivative (PID) design using meta-heuristics, such as evolutionary programming (Hung *et al*., 2008), differential evolution (Liu *et al*., 2007), harmony search (Coelho and Bernert, 2009a), tribes optimization (Coelho and Bernert, 2009b), and particle swarm optimization (Chang, 2009) have been validated for chaotic synchronization.

In this paper, the tuning of a PID controller based on an improved particle swarm optimization approach (PSO) algorithm based on normative knowledge (NPSO) for synchronization of two identical discrete chaotic systems subject to different initial conditions. The normative knowledge, a feature of cultural algorithms (Reynolds, 1994), is useful to improve the convergence of the classical PSO.

The PSO originally developed by Kennedy and Eberhart (1995) and Eberhart and Kennedy (1995) is a population-based swarm algorithm. Similarly to genetic algorithms (Goldberg, 1989), an evolutionary algorithm approach, PSO is an optimization tool based on a population, where each member is seen as a particle, and each particle is a potential solution to the problem under analysis. Each particle in PSO has a randomized velocity associated to it, which moves through the space of the problem. However, unlike genetic algorithms, PSO does not have operators, such as crossover and mutation. PSO does not implement the survival of the fittest individuals; rather, it implements the simulation of social behavior.

The remainder of this paper is organized as follows. In section 2, a description of a discrete PID controller and the case study using chaotic systems are formulated. Section 3 presents the fundamentals of PSO and NPSO approaches. Section 4 presents the simulation results of PID's tuning and chaotic synchronization. Lastly, the conclusion is provided in section 5.

2. DESCRIPTION OF PROBLEM

2.1. PID controller

As modeled in this paper, the transfer function of PID controller is described by the following equation in the continuous *s*-domain (Laplace operator)

$$G_{PID}(s) = P + I + D = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d \cdot s \tag{1}$$

$$\text{or } G_{PID}(s) = K_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\right), \tag{2}$$

where $U(s)$ and $E(s)$ are the control (controller output) and tracking error signals in *s*-domain, respectively; K_p is the proportional gain, K_i is the integration gain, and K_d is the derivative gain. T_i is the integral action time or reset time and T_d is referred to as the derivation action time or rate time.

In this context, the output of the PID controller in time domain is given by (Coelho and Bernert, 2009)

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(\tau) d\tau + K_d . \frac{de(t)}{dt}, \tag{3}$$

where $u(t)$ and $e(t)$ are the control and tracking error signals in time domain, respectively. Using trapezoidal approximations for equation (3) to obtain the discrete control law, we have (Coelho and Bernert, 2009)

$$u(k) = u(k-1) + K_p \cdot [e(k) - e(k-1)] + K_i \cdot \frac{T_s}{2} \cdot [e(k) - e(k-1)] + K_d . \frac{T_s}{2} \cdot [e(k) - 2e(k-1) + e(k-2)], \tag{4}$$

where T_s is the sampling period.

Over the past 60 years, several methods for determining PID controller parameters have been developed. Some employ information about open-loop step response, for example, the Coon-Cohen reaction curve method; other methods use knowledge of the Nyquist curve, e.g., the Ziegler-Nichols frequency-response method (Hang *et al.*, 2002; Cominos and Munro, 2002; O'Dwyer, 2006). Recently, many optimization techniques have been employed to improve the PID controller performance. In the present work, in order to find the controller parameters of PID, the PSO and NPSO approaches were used.

2.2 Nonlinear discrete chaotic system

In this study, two identical delayed discrete chaotic systems are considered to be synchronized using the proposed PID control. The master system is given by the following difference equation (Peng, 2004; Peng *et al.*, 2004):

$$x(k+1) = x(k) - \frac{\delta}{m} x(k-m) + \frac{\varepsilon}{m} x^3(k-m), \tag{5}$$

where δ and ε are positive constants, m is the delay term, and x is the master state. The delayed discrete system admits decaying, oscillatory, and chaotic behavior relying on settings of system parameters. On the other hand, the corresponding slave system is described by

$$y(k+1) = y(k) - \frac{\delta}{m} y(k-m) + \frac{\varepsilon}{m} y^3(k-m) + u(k)\,, \tag{6}$$

where y is the slave state and u is the external control force that adopts the PID control of equation (4). For two identical delayed discrete chaotic systems (5) and (6) without control u, the state trajectories of these two chaotic systems will quickly separate each other if their initial conditions are not the same. However, the state trajectories can approach synchronization for any initial conditions if an appropriate controller is utilized. Hence the purpose of this paper is to apply the PSO and NPSO approaches to find out the optimal PID control gains such that chaos synchronization for two identical delayed discrete chaotic systems is achieved.

For simplicity, the cost function F used in the study is defined as

$$F = \sum_{k=1}^{N} \left| x(k) - y(k) \right| = \sum_{k=1}^{N} \left| e(k) \right|\,, \tag{7}$$

where $e(k)$ is the error signal between the master and slave states and N is the total number of sampling. The optimization problem involves finding $g^* = [K_p^*,\ K_i^*,\ K_d^*]$ such that the F performance index of the system is minimized.

3. FUNDAMENTALS OF PARTICLE SWARM OPTIMIZATION

This section describes the proposed NPSO algorithm. First, a brief overview of the classical PSO is provided, and finally the proposed NPSO algorithm is discussed.

3.1. PSO algorithm

The proposal of PSO algorithm was put forward by several scientists who developed computational simulations of the movement of organisms such as flocks of birds and schools of fish. Such simulations were heavily based on manipulating the distances between individuals, i.e., the synchrony of the behavior of the swarm was seen as an effort to keep an optimal distance between them.

In theory, at least, individuals of a swarm may benefit from the prior discoveries and experiences of all the members of a swarm when foraging (Kennedy and Eberhart, 1995). The fundamental point of developing PSO is a hypothesis in which the exchange of information among creatures of the same species offers some sort of evolutionary advantage.

Similarly to other population-based algorithms, PSO exploits a population of search points to probe the search space. Each individual in particle swarm, referred to as a ‘particle’, represents a potential solution. Each particle utilizes two important kinds of information in decision process. The first one is their own experience; that is, they have tried the choices and know which state has been better so far, and they know how good it was. The second one is other particle’s experiences; that is, they have knowledge of how the other agents around them have performed.

Each particle in PSO keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far. This value is called *pbest* (*p*ersonal *best*). Another “best” value that is tracked by the global version of the particle swarm optimizer is the overall best value and its location obtained so far by any particle in the population. This location is called *gbest* (*g*lobal *best*).

Each particle moves its position in search domain and updates its velocity according to its own flying experience and neighbor’s flying experience toward its *pbest* and *gbest* locations (global version of PSO). Acceleration is weighted by random terms, with separate random numbers being generated for acceleration toward *pbest* and *gbest* locations, respectively.

The procedure for implementing the global version of PSO is given by the following steps (Coelho, 2009):

Step 1: *Initialization of swarm positions and velocities*: Initialize a population (array) of particles with random positions and velocities in the n dimensional problem space using uniform probability distribution function.

Step 2: *Evaluation of particle’s fitness*: Evaluate each particle’s fitness value.

Step 3: *Comparison to pbest (personal best)*: Compare each particle’s fitness with the particle’s *pbest*. If the current value is better than *pbest,* then set the *pbest* value equal to the current value and the *pbest* location equal to the current location in n-dimensional space.

Step 4: *Comparison to gbest (global best)*: Compare the fitness with the population’s overall previous best. If the current value is better than *gbest*, then reset *gbest* to the current particle’s array index and value.

Step 5: *Updating of each particle's velocity and position*: Change the velocity, v_i, and position of the particle, x_i, according to equations (8) and (9):

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot ud \cdot [p_i(t) - x_i(t)] + c_2 \cdot Ud \cdot [p_g(t) - x_i(t)] \tag{8}$$

$$x_i(t+1) = x_i(t) + \Delta t \cdot v_i(t+1) \tag{9}$$

where w is the inertia weight; i=1,2,…,N indicates the number of particles of population (swarm); t=1,2,…t_{max}, indicates the iterations; $v_i = [v_{i1}, v_{i2}, ..., v_{in}]^T$ stands for the velocity of the i-th particle, $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T$ stands for the position of the i-th particle of population, and $p_i = [p_{i1}, p_{i2}, ..., p_{in}]^T$ represents the best previous position of the i-th particle. Positive constants c_1 and c_2 are the cognitive and social components, respectively, which are the acceleration constants responsible for varying the particle velocity towards *pbest* and *gbest*, respectively. Index g represents the index of the best particle among all the particles in the swarm. Variables ud and Ud are two random functions in the range [0,1]. Equation (9) represents the position update, according to its previous position and its velocity, considering $\Delta t = 1$.

Step 6: *Repeating the evolutionary cycle*: Return to **Step 2** until a stop criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

3.2. Proposed NPSO algorithm

Cultural algorithms are evolutionary techniques based on some theories proposed in sociology and archaeology to model cultural evolution, which extract information from the domain of the problem during the evolutionary process itself. In this context, a cultural algorithm can incorporate domain knowledge obtained during the evolutionary process to render the search process more efficient. The application of cultural algorithms in PSO is an alternative strategy to improve the convergence performance and local search (Reynolds, 1994).

Cultural algorithms consist of three components. First, there is a population component (or population space) that contains the population to be evolved and the mechanisms for its evaluation, reproduction and modification. Second, there is a belief space that represents the bias that has been acquired by the population during its problem-solving process. In this work, the proposed NPSO approach employs a belief space with a normative knowledge source.

The normative knowledge contains the intervals for decision variables (individuals) where good solutions have been found, in order to move new solutions towards those intervals. The l_i and u_i are the lower and upper bounds, respectively, for the i-th decision variable, and L_i and U_i are the values of the fitness function. More details fo implementation of NPSO are presented in Coelho and Alotto (2008).

In this work, the modification of equation (8) (classical PSO) proceeds as follows in the pseudo code of NPSO, where j=1,…,n are decision variables of each dimension:

```
% Flag: boolean variable (0-false; 1–true)
Flag = 0;
If x_ij(t) < l_ij then
        v_i(t+1) = w·v_i(t) + c_1·ud·[p_i(t) − x_i(t)] + c_2·Ud·[p_g(t) − x_i(t)]
        Flag = 1;
End if
If x_ij(t) > u_ij then
        v_i(t+1) = w·v_i(t) − c_1·ud·[p_i(t) − x_i(t)] − c_2·Ud·[p_g(t) − x_i(t)]
         Flag = 1;
End if
```

4. SIMULATION RESULTS

In this section, we illustrate the synchronization PID controller design for the above two systems given by equations (5) and (6) with different initial value conditions. The parameters δ = 3.6, ε = 1, m = 10 and initial conditions x_i = 0.2 and y_i = -0.2 (i = -m,-m + 1,...,0) are used in this example. We solve the optimization problem with N = 190, k_f = 80, T_s=1 s, and b = 0.05.

In this section, to verify and demonstrate the effectiveness of the proposed method, we discuss the simulation results of synchronization between the master and slave states for the classical PSO and the NPSO approaches.

Each optimization method was implemented in Matlab (MathWorks). All the programs were run on a 3.2 GHz Pentium IV processor with 2 GB of random access memory. In each case study, 50 independent runs were made for each of the optimization methods involving 50 different initial trial solutions for each optimization method. In this paper, the PSO approaches are adopted using 2,000 cost function evaluations in each run. The lower and upper bounds of the search space used in optimization methods were $(K_p, K_i, K_d) \in [0,3]$.

In this case study, the population size was 20 particles, number maximum of generations (stopping criterion) was 100, $c_1 = c_2 = 2.05$, and w is a linearly decreasing function during the evolutionary cycle of 0.9 to 0.4 in PSO and NPSO approaches.

Optimization results of PSO and NPSO approaches in PID controller tuning for chaotic synchronization are presented in Table 1. It can be seen from Table 1 that with the same preset maximum number of generations, NPSO obtained better mean and minimum F values than classical PSO. Furthermore, the convergence curve presented in Figure 1 showed that the NPSO approach presented faster convergence that the PSO in minimization of F. Table 2 presents the best results of the PID controller gains obtained using PSO and NPSO in tested example. Figure 2 shows the state responses of the master and slave systems using the resulting PID controller gains obtained by NPSO.

Table 1. Convergence results for synchronization task of chaotic system in 50 runs.

Optimization Method	Minimum F	Mean F	Maximum F	Standard Deviation of F
PSO	2.3693	2.5255	2.7379	0.1411
NPSO	2.3143	2.3143	2.3144	$1.0734 \cdot 10^{-5}$

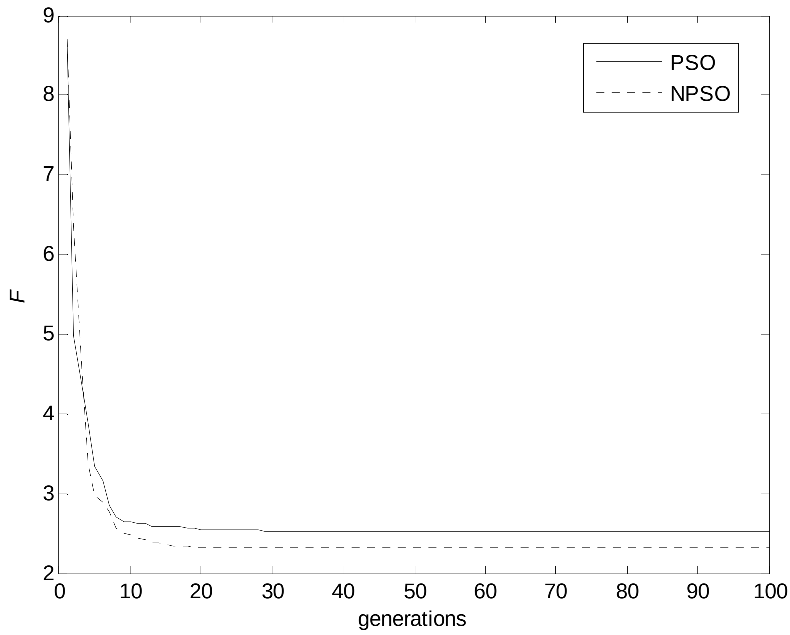

Figure 1. Convergence (mean of best F) of PSO and NPSO in 50 runs.

Table 2. Best results of PID controller gains and performance data using optimization method in 50 runs.

Parameter	PSO	NPSO
K_p	0.6079	0.6211
K_i	0.1095	0.1234
K_d	0.3073	0.3238
Mean of error signal	0.0106	0.0095
Variance of control signal	$4.0933 \cdot 10^{-4}$	$4.6186 \cdot 10^{-4}$
F	2.3693	2.3143

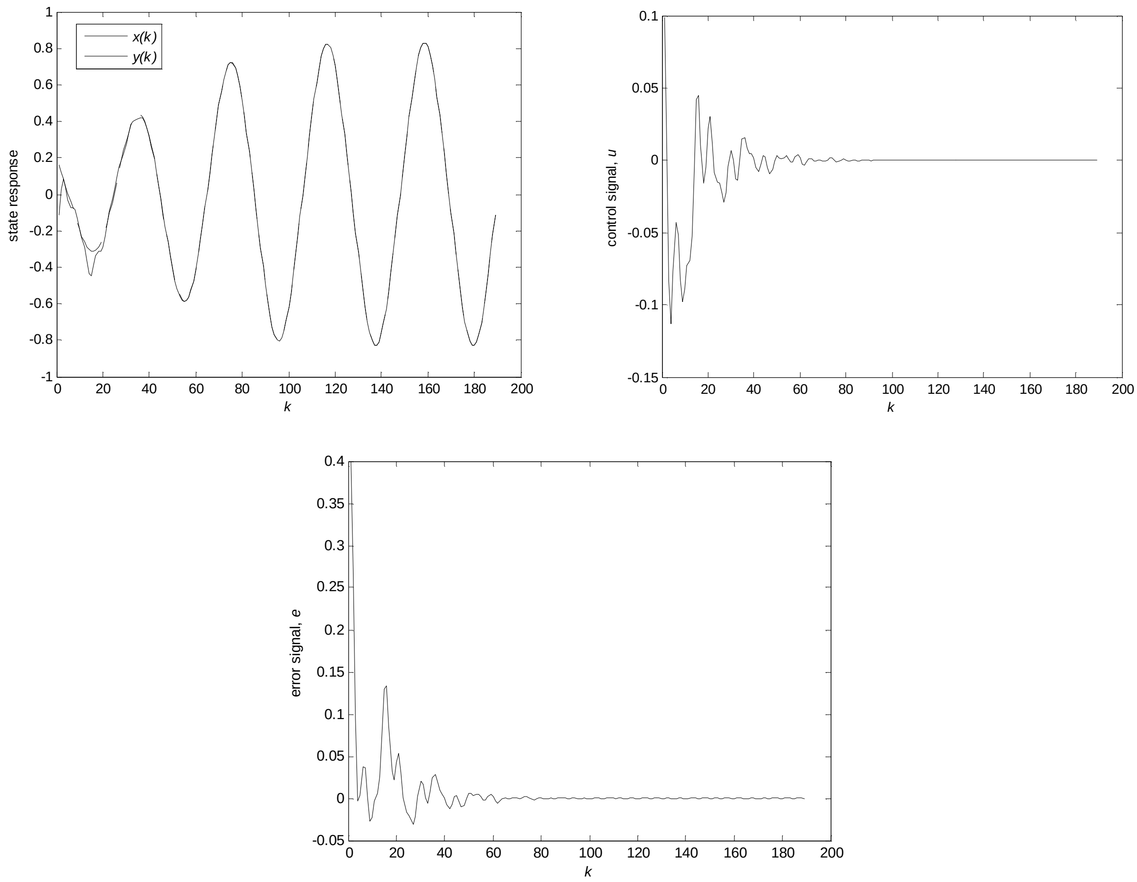

Figure 2. Best result using NPSO.

5. CONCLUSION

In recent years the investigation of chaos synchronization has attracted much attention in the light of potential applications in engineering field.

The contribution of this paper is to investigate the synchronization of two identical discrete chaotic systems subject to different initial conditions by designing a PID controller based on PSO and NPSO approaches. The proposed control scheme is easily implemented. A numerical example has been presented to verify the validity of the developed control scheme. The results have shown that the proposed controller is successful when applied to chaotic synchronization.

As an extension of this paper, we may also incorporate the dynamic parameters updating techniques into the NPSO method to make it more promising in applications of PID tuning in multivariable systems.

6. ACKNOWLEDGEMENTS

This work was supported by the National Council of Scientific and Technologic Development of Brazil — CNPq — under Grant 309646/2006-5/PQ and by the Institutional Program of Scientific Initiation Grants from the Pontifical Catholic University of Paraná — PIBIC/PUCPR.

7. REFERENCES

Chang, W.D., 2009, “PID Control for Chaotic Synchronization Using Particle Swarm Optimization”, Chaos, Solitons & Fractals, Vol. 39, No. 2, pp. 910-917.

Coelho, L. S., 2009, “An Efficient Particle Swarm Approach for Mixed-Integer Programming in Reliability-Redundancy Optimization”, Reliability Engineering & System Safety, Vol. 94, No. 4, pp. 830-837.

Coelho, L.S. and Alotto, P., 2008, “Particle Swarm Optimization Combined with Normative Knowledge Applied to Loney’s Solenoid Design”, Proceedings of 10th International Workshop on Optimization and Inverse Problems in Eletromagnetism, Ilmenau, Germany.

Coelho, L.S. and Bernert, D.L.A., 2009a, "An Improved Harmony Search Algorithm for Synchronization of Discrete-time Chaotic Systems", Chaos, Solitons & Fractals, Vol. 41, No. 5, pp. 2526-2532.

Coelho, L.S. and Bernert, D.L.A., 2009b, "PID Control Design for Chaotic Synchronization Using a Tribes Optimization Approach", Chaos, Solitons & Fractals, Vol. 42, No. 1, pp. 634-640.

Collet, P. and Eckmann, J.-P., 2006, Concepts and Results in Chaotic Dynamics: A Short Course, Springer, Heidelberg, Germany.

Cominos, P. and Munro, N., 2002, "PID Controllers: Recent Tuning Methods and Design to Specification", IEE Proc.-Control Theory and Applications, Vol. 149, No. 1, pp. 46-53.

Eberhart, R.C. and Kennedy, J.F., 1995, "A New Optimizer Using Particle Swarm Theory", Proceedings of International Symposium on Micro Machine and Human Science, Japan, pp. 39-43.

Goldberg, D.E., 1989, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison Wesley: Reading, MA, USA.

Hang, C.C., Åström, K.J. and Wang, Q.G., 2002, "Relay Feedback Auto-tuning of Process Controllers — A Tutorial Review", Journal of Process Control, Vol. 12, No. 1, pp. 143-162.

Hung, M.L., Lin, J.S., Yan, J.J. and Liao, T.L., 2008, "Optimal PID Control Design for Synchronization of Delayed Discrete Chaotic Systems", Chaos, Solitons & Fractals, Vol. 35, No. 4, pp. 781-785.

Kennedy, J.F. and Eberhart, R.C., 1995, "Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942-1948.

Liu, B., Wang, L., Jin, Y.H., Huang, D.X. and Tang, F., 2007, "Control and Synchronization of Chaotic Systems by Differential Evolution Algorithm", Chaos, Solitons and Fractals, Vol. 34, No. 2, pp. 412-419.

Lu, Z., Shieh, L.-S., Chen, G. and Coleman, N.P., 2005, "Simplex sliding mode control for nonlinear uncertain systems via chaos optimization", Chaos, Solitons and Fractals, Vol. 23, pp. 747-755.

O'Dwyer, 2006, "Handbook of PI and PID Controller Tuning Rules", 2nd edition, London, UK, Springer-Verlag.

Pecora L.M. and Carroll, T.L., 1990, "Synchronization in Chaotic Systems", Physical Review Letters, Vol. 64, No. 8, pp. 821-824.

Peitgen, H.O., Jürgens, H., Saupe, D., 2004, "Chaos and Fractals: New Frontiers of Science", 2nd edition. Springer, New York, NY, USA.

Peng, M., 2004, "Bifurcation and Chaotic Behavior in the Euler Method for a Kaplan-Yorke Prototype Delay Model", Chaos, Solitons & Fractals, Vol. 20, No. 3, pp. 489-496.

Peng, M., Bai, E.W. and Lonngren, K.E., 2004, "On the Synchronization of Delay Discrete Models", Chaos, Solitons & Fractals, Vol. 22, No. 3, pp. 573-576.

Reynolds, R.G., 1994, "An Introduction to Cultural Algorithms", Proceedings of 3rd Annual Conference on Evolutionary Programming, A.V. Sebald and L.J. Fogel (eds.), World Scientific, River Edge, NJ, USA, pp. 131-139.

Strogatz, S.H., 2000, "Nonlinear Dynamics and Chaos", Massachussetts, Perseus Publishing.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.