PARTICLE SWARM OPTIMIZATION APPROACH FOR MULTI-STEP-AHEAD PREDICTION USING RADIAL BASIS FUNCTION NEURAL NETWORK

Cezar A. Sierakowski [1], Fábio A. Guerra[2], and Leandro dos S. Coelho[1]

[1] *Pontifical Catholic University of Parana, PUCPR/PPGEPS/LAS*
Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, PR, Brazil
E-mail: cezar.sierakowski@pucpr.br; leandro.coelho@pucpr.br

[2] *Artificial Intelligence Division*
TECPAR – Parana Institute of Technology
St. Algacyr Munhoz Mader, 3775, Zip code 81350-010, Curitiba, PR, Brazil
E-mail: guerra@tecpar.br

Abstract: An alternative approach, between much others, for mathematical representation of dynamics systems with complex or chaotic behaviour, is a radial basis function neural network using *k*-means for clustering and optimized by pseudo-inverse and particle swarm optimisation. This paper presents the implementation and study to identify a dynamic system, with nonlinear and chaotic behaviour, called *Rössler's* circuit, with concepts of multi-step-ahead prediction. *Copyright © 2005 IFAC*

Keywords: Neural network, clustering, chaotic circuit, Rössler's circuit, multivariable system identification, particle swarm optimization, nonlinear systems.

1. INTRODUCTION

The mathematical description of dynamic systems it's not a simple task in which basic principles may be used. For complex systems, modelling using basic laws to determine the dynamic behaviour of such systems is not always possible. An interesting alternative to solve such problems would be an experimental for systems identification. A model based in an input-output system must be found, seeking a relation between these.

In real life, most systems are nonlinear and the use of linear models is limited, because they cannot represent the system dynamics, such as its hysteresis, amplitude dependency, bifurcations or chaos (Ivankhnenko, 1971). This characteristics describes a nonlinear system and is necessary the development of techniques that model such behaviour. A particular area of nonlinear system identification is the chaotic modelling. A high number of experiments for classification, analysis, comprehension and control chaotic systems exists (Alligood *et al.*, 1996; Ioh *et al.*, 2001; Thamilmaran *et al.*, 2000).

Nonlinear systems identification is normally a difficult task. When the system is dissipative, to develop a model through experimental data became a challenge due to its nature. Although the system outputs are limited, the chaotic behaviour is essentially unstable, with asymptotic behaviour, producing strange attractors. Besides, the chaotic systems show a huge sensitivity to initial conditions. When two or more trajectories diverge and become non-correlated, a limit is imposed to make predictions and makes harder to determine if the identified model is equivalent to the tested system (Huang and Loh, 2001; Lian and Liu, 2000).

The use of neural networks to nonlinear identification problems has attracted some attention in recent years. Neural networks are originally inspired by biologic neural networks' functionality that may learn complex functional relations through a limited number of training data. Neural networks may serve as black-box models of nonlinear multivariable dynamic systems and may be trained using input-output data, observed from the system (Mcloone, *et al.*, 1998; Narendra and Parthasarathy, 1990). The usual neural network consists of multiple simple processing elements, called neurons, interconnections among them and the weights attributed to the interconnections. The relevant information of such methodology is stored in the weights. (Haykin, 1994; Pei and He, 1999).

The main objective of this paper is to present an optimization approach for nonlinear identification using radial basis function neural network (RBF-NN) of Rössler's chaotic electronic system. The RBF-NN uses the k-means clustering algorithm, and is optimized by pseudo-inverse and particle swarm optimization (PSO).

This paper is organised as follows. In section 2, the Rössler's system is presented. In section 3, the multi-step-ahead prediction with RBF-NN is discussed. The simulation results are presented in section 4. The conclusions and future works are discussed in section 5.

2. RÖSSLER'S CIRCUITS

The identification case study boarded in the paper is the nonlinear dynamic system with chaotic behaviour of Rössler (1976). The german scientist O. Rössler proposed a chaotic attractor composed by nonlinear differential equations. The Rössler's equations are:

$$\frac{dx}{dt} = -y - z \tag{1}$$

$$\frac{dy}{dt} = x + ay \tag{2}$$

$$\frac{dz}{dt} = b + (x - c)z \tag{3}$$

These equations show chaotic behaviour for (a, b, c) = (0.36; 0.40; 4.50). This behaviour is illustrated in Fig. 1.

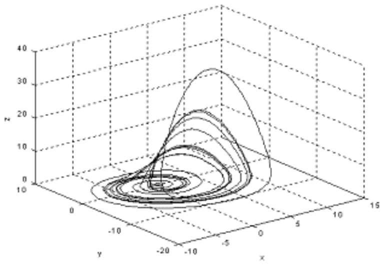

Fig. 1. Rossler's attractor system for (a, b, c) = (0.36, 0.40, 4.50).

A pair of electronic oscillators is used to simulate Rössler's chaotic oscillation, a study is developed in (Taherion and Lai, 2000). The schematic diagram of two unidirectionally coupled Rössler's chaotic circuits is presented in Fig. 2.

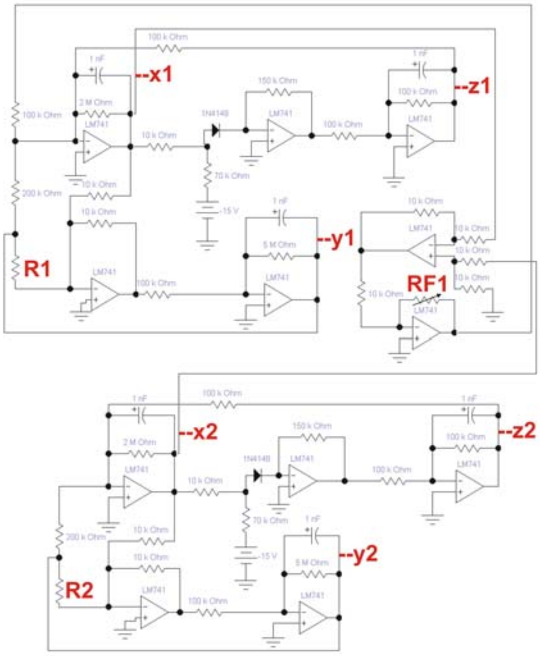

Fig. 2. Two coupled Rössler's chaotic circuits diagram.

A pair of unidirectionally coupled oscillators can be described, mathematically, as follows:

$$\frac{dx}{dt} = f(x) \tag{4}$$

$$\frac{dy}{dt} = g(x, y) \tag{5}$$

where x and y are the sets of dynamical variables of the two oscillators. The unidirectional coupling scheme is equivalent to the master-slave type of coupling because there is no influence, say, from y to x. The unidirectionally coupling scheme is actually quite representative of coupled nonlinear oscillators in general, because there always exists a mathematical change of coordinates to transform a pair of mutually coupled (bidirectionally coupled) oscillators into a pair of unidirectionally coupled ones, at least locally near the state of synchronization (Rössler, 1976).

For the unidirectionally coupling scheme in our experiment, the differential equations describing the circuit are:

$$\frac{dx_1}{dt} = -\gamma x_1 - \alpha y_1 - z_1 + \varepsilon(x_2 - x_1) \tag{6}$$

$$\frac{dy_1}{dt} = \beta x_1 - a_1 y_1 \tag{7}$$

$$\frac{dz_1}{dt} = g(x_1) - z_1 \tag{8}$$

$$\frac{dx_2}{dt} = -\gamma x_2 - \alpha y_2 - z_2 \qquad (9)$$

$$\frac{dy_2}{dt} = -\beta x_2 + a_2 y_2 \qquad (10)$$

$$\frac{dz_2}{dt} = g(x_2) - z_2 \qquad (11)$$

where $g(x) = 0$ if $x \leq 3$, $g(x) = \mu(x - 3)$ if $x > 3$. The parameters in Eq. above are as follows: $\alpha = 0.5$, $\beta = 1$, $\gamma = 0.05$, $a_1 = 0.113$, $a_2 = 0.129$, $\varepsilon = 0.015$, and $\mu = 15$. The uncertainties in these parameters are about 5%. The resistors R_1 and R_2 in the circuit are chosen to be 75 kΩ and 67 kΩ, respectively, to ensure a systematic parameter mismatch between the two Rössler circuits. This difference corresponds to approximately 10% difference in the parameters a_1 and a_2.

3. MULTI-STEP-AHEAD WITH RBF-NN

System identification is a process that requires the modeller involvement (Chen, *et al.*, 1990; Ljung, 1997). The designer must analyse which system's variables are relevant for the modelling, and if the chosen structure model is adequate, otherwise, he must take the necessary decisions to solve the problem. The following steps may be quoted in the identification system process: (i) experimentation; (ii) nonlinear detection; (iii) structure model determination; (iv) parameters estimation phase; and (v) validation phase.

There are several representations for nonlinear system modelling with chaotic behaviour. In this application is chosen RBF-NN. This neural network project can be seen as a curve adjustment problem (function approximation problem) in a high dimensionality space. For this, the RBF-NN learning is equivalent to find a surface in a multidimensional space that better fit the training data set, where the criteria for best fit is measured in statistic (Chen *et al.*, 1990; Jang and Sun, 1993).

The RBF-NN is a flexible tool in dynamic environment. They have the ability to quickly learn complex patterns and tendency present in data and quickly adapt to changes. Such characteristics make them adequate to temporal series prediction, especially those ruled by linear processes and/or non stationary (Lo, 1998).

The radial basis function (or activation function) used in RBF-NN is Gaussian type as illustrated in Eq. 12. The estimated output is shown in Eq. 13. The Fig. 3 shows the general structure of RBF-NN.

$$f(x) = e^{-(\frac{x_i - c_j}{\sigma_j})} \qquad (12)$$

where:

x_i: inputs vector;
c_j: activation function center (Gaussian);
σ_j: standard deviation.

$$\hat{y}(t) = \sum_{i=1}^{n} w_m k_m \qquad (13)$$

where:

n: clusters quantity (neurons);
w_m: weights;
k_m: hidden layer output.

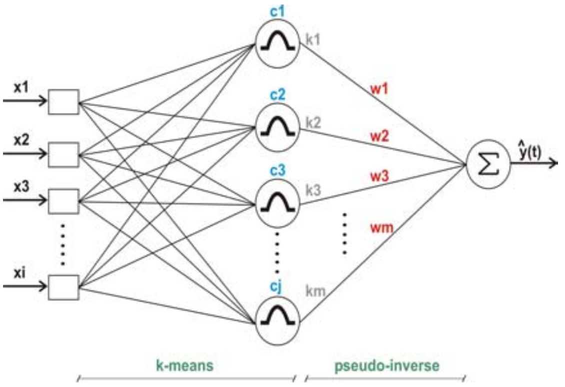

Fig. 3. General structure of RBF-NN.

The clustering method used in this application by the RBF-NN for classification problems is k-means. Its implementation follows the steps:

Step 1: Initialize functions centers.
Adjust the function initial centers to the first training data.

Step 2: Group all data with each function center.
Each input data (x_i) belongs to a cluster j^*, where:

$$\| x_i - c_{j^*} \| = \min_j \| x_i - c_j \| \qquad (14)$$

Step 3: Find each function center.
For each c_j:

$$c_j = \frac{1}{m_j} \sum_{x_i \in j} x_i \qquad (15)$$

where m_j is the number of data of cluster j.

Step 4: Repeat step 2, until there is no more changes in each cluster.

The PSO is responsible for optimizing the centers obtained by k-means and also optimizing the Gaussian spreads, PSO has a population with random positions, each of these particles have a velocity , and the particles "fly" around the search space. The particles store their best position in their memory (*pbest*) and also the fitness in this point.

The best *pbest* of all swarm is denominated as the best global position (*gbest*) (Gudise and Venayagamoorthy, 2003). The basic concept of PSO is to accelerate particles toward *pbest* and *gbest*, weighted by an acceleration factor at each time step. Mathematically, a particle follows the following equations.

$$V_{id}^{t+1} = W * V_{id}^{t} + c_1 * rand_1 * \left(P_{id} - X_{id}^{t}\right) + c_2 * rand_2 * \left(P_{gd} - X_{id}^{t}\right) \quad (16)$$

$$X_{id}^{t+1} = X_{id}^{t} + V_{id}^{t+1}\Delta t \quad (17)$$

where $\Delta t = 1$, t represents the actual iteration and *t+1* represents the next iteration V_{id} and X_{id} represents the particles' velocity and position, $rand_1$ and $rand_2$ are random number between [0,1], used to maintain the population diversity.

Eq. 16 is used to update each particle's speed, for its calculus the speed in last iteration, multiplied by an inertial weight. The second factor is composed by a cognition part, the basis is the difference between the actual position of the particle and the best position it has achieved in history (*pbest*).

The last factor is composed by a social component; the calculus basis is the particle actual position and the best position achieved by any particle in the algorithm execution (*gbest*).

Eq. 17 represents the position update of a particle, according with its previous position and its actual speed, considering $\Delta t = 1$.

One of the main reasons for the PSO attractive is the need to adjust few parameters (Xie *et al.*, 2002).

Constants c_1 and c_2 are positive constants denominated cognition and social components, respectively. These are the acceleration constants, varying the speed of the particle toward *pbest* and *gbest*, according to past experience.

Constants c_1 and c_2 are not critical factors to algorithm convergence. However, a fine tuning of such values may cause a faster convergence. Values of c_1 and c_2 are assumed as 2.0, according to Gaing (1994). But recent researches inform that the choice may be even better if the cognition parameter higher than a social parameter, inside the limits $c_1 + c_2 \leq 4$ (Parsopoulos and Vrahatis, 2002).

The use of *W*, called inertial weight is proposed by Shi and Eberhart (1998), this parameter is responsible for a dynamic adjustment of the particle speed, so, it's responsible for balancing the research performed by the algorithm between a local and a global one, making possible that the algorithm converges in a smaller number of iterations. A higher value of inertial weight make possible a global search, by the other side, a small value takes the algorithm into a local search.

Through a dynamical adjustment of the inertial weight, it's possible to dynamically adjust the search capability. Once the PSO search process is nonlinear and complex, it is hard, if not impossible, to mathematically model the search capability to dynamically adjust the inertial weight, so, a fixed or a linearly decaying inertial weight may be adopted. Other alternatives for dynamical adjustment or W are adoption of co-evolution, meta-optimization or adoption of fuzzy systems.

Application of a high value of inertial weight at the start and decaying until a small value through the PSO execution causes the algorithm to own global search characteristics at the start and local search characteristics in the end of the execution. The value of *W* decaying from a maximum value of 0.9 towards a minimum value of 0.4 through the execution is a good call. When adopting linearly decaying inertial weights, normally Eq. 18 is adopted, for *W* update, where $iter_{max}$ is the maximum number of iterations and *iter* is the actual iteration (Shi and Eberhart, 2002).

$$W = W_{\max} - \frac{W_{\max} - W_{\min}}{iter_{\max}} * iter \quad (18)$$

The linear optimization method to making the parameters of RBF-NN linear, in this application, is the pseudo-inverse. The update of each weight for training RBF-NN using this derivation of least mean squares is realized by Eq. 19.

$$w_m = \left(\left(k^T k\right)^{-1} k^T\right) x(t) \quad (19)$$

where *x*(*t*) is the desired output. The error calculus is realized by:

$$e(t) = \hat{y}(t) - y(t) \quad (20)$$

The performance criteria evaluated for the dynamic system to be identified is the multiple correlation coefficient, R^2, between real output *y(t)* and the estimated output $\hat{y}(t)$.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i(t) - \hat{y}_i(t))^2}{\sum_{i=1}^{n} (y_i(t) - \bar{y})^2} \quad (21)$$

where *n* is the number of measured samples of the process output.

When the value of R^2 is equal to 1.0, indicates an exact fit of the model to the process' measured data. The value of R^2 between 0.9 and 1.0 is considered enough for practical applications, in control systems (Schaible *et al.*, 1997).

4. RESULTS

In Table 1, the Rössler's system identification results using a radial basis function neural network using k-means for clustering and optimized by pseudo-inverse and PSO with concepts of multi-step-ahead prediction are presented.

In the estimation phase (training of RBF-NN) 10000 samples were used, and in the validation phase 10000 different samples were used.

The results were obtained using 3 clusters for a multi-step-ahead prediction of coordinate $x(t)$ of Rössler system. In this case, it's used the addition of a white noise varying between [-0.1, 0.1] for the coordinate $x(t)$. The tables 1 to 3 are formed with the number of steps ahead in the rows and the presence or not of noise in the columns. Table 1 presents the maximum of R^2 obtained by the simulations; Table 2 presents the mean of R^2 and Table 3 presents the standard deviation of R^2.

Table 1. Maximum of R^2 obtained for the simulations

	Without noise	With noise
50	0.9931	0.9398
60	0.9928	0.9101
70	0.9922	**0.9109**
80	0.9843	0.8776
90	0.9807	0.8511
100	**0.9428**	0.7649

Table 2. Mean of R^2 obtained for the simulations

	Without noise	With noise
50	0.9305	0.9188
60	0.9619	0.8932
70	0.9276	0.8685
80	0.8614	0.8517
90	0.8719	0.8209
100	0.7838	0.7022

Table 3. Standard deviation of R^2 obtained for the simulations

	Without noise	With noise
50	0.0350	0.0119
60	0.0530	0.0119
70	0.0761	0.0286
80	0.0943	0.0418
90	0.0982	0.0488
100	0.0967	0.0358

Through Table 1 it's possible to observe that the RBF-NN without noise is capable to predict efficiently until 100 steps-ahead. However,, for a system with Gaussian noise, the RBF-NN is capable to predict only 70 steps-ahead. The figures about these predictions are illustrated in Figs. 4, 5, 6 and 7. These figures present only the results for the validation phase of RBF-NN.

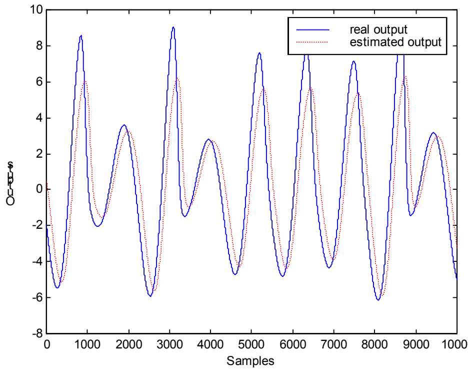

Figure 4. Output signal using RBF-NN with *k*-means without noise and 100 steps-ahead.

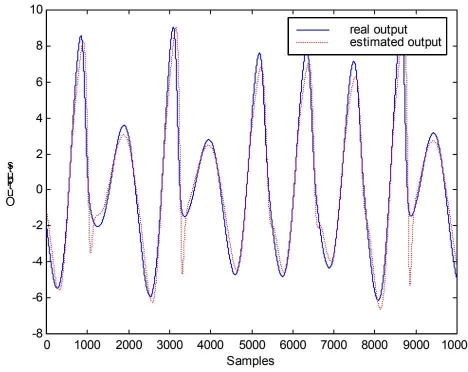

Figure 5. Output signal using RBF-NN with *k*-means and PSO without noise and 100 steps-ahead.

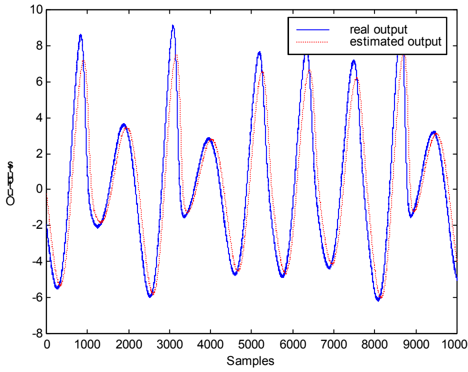

Figure 6. Output signal using RBF-NN with k-means with noise and 70 steps-ahead.

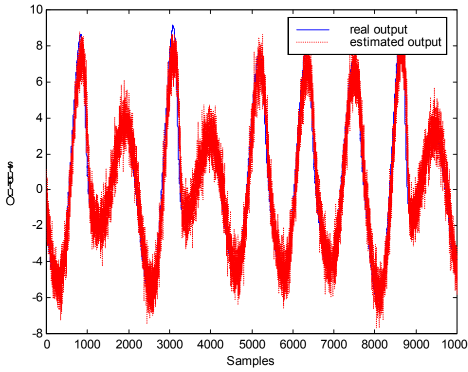

Figure 7. Output signal using RBF-NN with *k*-means and PSO with noise and 70 steps-ahead.

5. CONCLUSIONS

The case study of this paper intended to identification and validation of a Rössler's circuit dynamic behaviour using RBF-NN. In this paper the RBF-NN to identification are presented and the methodology presented result with high precision when dealing with the approximation of Rössler's circuit nonlinear chaotic behaviour.

The preliminary presented results show that RBF-NN can be a powerful tool to predict temporal series and to study complex and chaotic behaviour. It's possible to realize that the use of PSO in optimizing the centers generated by k-means has considerably increased the results, increasing the robustness of RBF-NN.

Including noise it's possible to observe that the applied methodology suffers a decrease in its' results, however, the results, for 70 steps-ahead, may be considered acceptable, because R^2 is above 0.9.

6. REFERENCES

Alligood, K. T.; Sauer, T. D.; Yorke, J. A. (1996). *Chaos: an introduction to dynamical systems*, Springer, London.

Chen, S.; Billings, S. A.; Cowan, C. F.; Grant, P. M. (1990). Practical identification of NARMAX models using radial basis functions, *International Journal of Control*, vol. 52, no. 6, pp. 1327-1350.

Gaing, Z. (1994). A particle swarm optimization approach for optimum design of PID controller in AVR system, *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 384-391.

Gudise, V. G.; Venayagamoorthy, G. K. (2003). Evolving digital circuits using particle swarm, *Proceedings of INSS-IEEE International Joint Conference on Neural Networks*, Portland, PO, USA, vol. 1, pp. 468-472.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*, Macmillan Publishing Company.

Huang, C. -C.; Loh, C. -H. (2001). Nonlinear identification of dynamic systems using neural network, C*omputer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 1, pp. 28-41.

Ioh, M.; Yang, T.; Chua, L. O. (2001). Conditions for impulsive synchronization of chaotic and hyperchaotic systems, *International Journal of Bifurcation and Chaos*, vol. 11, no. 2, pp. 551-560.

Ivankhnenko, A. G. (1971). Polynomial theory of complex systems, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 1, no. 4, pp. 364-378.

Jang, J. -S. R.; Sun, C. -T. (1993). Functional equivalence between radial basis function networks and fuzzy inference systems, *IEEE Transactions on Neural Networks*, vol. 4, no. 1, pp. 156-159.

Lian, K. W.; Liu, P. (2000). Synchronization with message embedded for generalized Lorenz chaotic circuits and its error analysis, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 9, pp. 1418-1430.

Ljung, L. (1987). *System identification: theory for the use.* Prentice-Hall: New York.

Lo, J. T. -H. (1998). Multilayer perceptrons and radial basis functions are universal robust approximators. *Internaitonal Joint Conference on Neural Networks*, Anchorage, AK, USA, pp. 1311-1314.

Mcloone, S.; Brown, M. D.; Irwin, G.; Lightbody, A. (1998). A hybrid linear/nonlinear training algorithm for feedforward neural networks, *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 669-684.

Narendra, K.; Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks, *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4-27.

Parsopoulos, K. E.; Vrahatis, M. N. (2002). Recent approaches to global optimization problems through particle swarm optimization, *Natural Computing*, vol. 1, no. 2-3, pp. 235-306.

Pei, W.; Yang, L.; He, Z. (1999). Identification of dynamical noise levels in chaotic systems and application to cardiac dynamics analysis. *International Joint Conference on Neural Networks,* vol. 5, Washington, DC, USA, pp. 3680-3684.

Rössler, O. E. (1976). An equation for continuous chaos, *Physical Letters*, vol 35A, pp. 397-398.

Schaible, B.; Xie, H.; Lee, Y. C. (1997). Fuzzy logic models for ranking process effects, *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 4, pp. 545-556.

Shi, Y.; Eberhart, R. C. (2002). Fuzzy adaptive particle swarm optimization, *Proceedings of the 2002 Congress on Evolutionary Computation*, Honolulu, Hawaii, USA, vol. 1, pp. 101-106.

Shi, Y.; Eberhart, R. C. (1998). Parameter selection in particle swarm optimizer, *Proceedings Seventh Annual Conference on Evolutionary Programming*, V.W. Porto, N. Saravan, D. Waagen, and A.E. Eiben (eds.). Berlin: Springer-Verlag, pp. 591-601.

Taherion, S.; Lai, Y. -C. (2000). Experimental observation of lag synchronization in coupled chaotic systems, *International Journal of Bifurcation and Chaos*, vol. 10, no. 11, pp. 2587-2594.

Thamilmaran, M.; Lakshmanan; Murali, K. (2000). Rich variety of bifurcations and chaos in a variant of Murali-Lakshmanan Chua circuit, *International Journal of Bifurcation and Chaos*, vol. 10, no. 7, pp. 1781-1785.

Xie, X.; Zhang, W.; Yang, Z. (2002) A dissipative particle swarm optimization, *Proceedings of the 2002 Congress on Evolutionary Computation*, Honolulu, Hawaii, USA, vol. 2, pp. 1456-1461.