ABCM Symposium Series in Mechatronics - Vol. 4 - pp.270-277

NONLINEAR IDENTIFICATION USING NEURAL NETWORK COMBINED WITH TRAINING BASED ON PARTICLE SWARM OPTIMIZATON

Fábio Alessandro Guerra, guerra@lactec.org.br
André Eugênio Lazzaretti, lazzaretti@lactec.org.br
Institute of Technology for Development (LACTEC), Eletroeletronics Department (DPEE), Electrical System Division (DVSE).
Centro Politécnico da UFPR, BR 116, Km 98, Jardim das Américas, Zip code 81531-980, Curitiba, PR, Brazil.

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

Cesar Augusto Tacla, tacla@cpgei.ct.utfpr.edu.br
Federal University of Technological - Paraná (UTFPR),
Graduate Program in Eletrical Engineering and Industrial Informatics (CPGEI).
Av. 7 de Setembro, 3165, Centro, Zip code 80230-901, Curitiba, PR, Brazil.

Abstract. *Most processes in industry are characterized by nonlinear and time-varying behavior. In this context, the identification of mathematical models typically nonlinear systems is vital in many fields of engineering. A variety of system identification techniques are applied to the modeling of processes dynamics. Recently, the identification of nonlinear systems by artificial neural networks has been successfully applied in many applications. In this paper, an original approach based on radial basis function neural network (RBF-NN) with a training method based on particle swarm optimization (PSO) is proposed as an alternative solution. RBF-NN is considered as a good candidate for the prediction problems due to its rapid learning capacity and, therefore, has been applied successfully to nonlinear time series modeling and nonlinear identification. On the other hand, PSO was inspired by the choreography of a bird flock and can be seen as a distributed behavior algorithm that performs multidimensional search. The RBF-NN model is trained and validated based on the experimental data of a nonlinear process. Finally, simulation results from the performance analysis of RBF-NN are presented and discussed.*

Keywords*: nonlinear identification, radial basis function neural networks, nonlinear processes, particle swarm optimization.*

1. INTRODUCTION

The mathematical description of dynamic systems is not a simple task in which basic principles may be used. For complex systems, modeling using basic laws to determine the dynamic behavior of a system 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 them (Coelho and Guerra, 2002).

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 it is necessary the development of techniques that model such behavior. A particular area of nonlinear system identification is the chaotic modeling. Several researches have approached problem in classification, analysis, comprehension and control chaotic systems (Alligood *et al.*, 1996; Ioh *et al.*, 2001).

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. 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, 2000; Pei and He, 1999; Huang and Loh, 2001; Lian and Liu, 2000).

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

The reminder of the paper is organized as follows. In section 2, the heating system process is presented. In section 3, the one-step-ahead prediction for system identification with RBF-NN with a training method based on PSO is discussed. The simulation results are presented in section 4. The conclusions and future works are discussed in section 5.

2. HEATING SYSTEM PROCESS

The identification case study evaluated in this paper is a heating process. The system is an experiment with single-input-single-output heating system. The input drives a 300 Watt Halogen lamp, suspended several inches above a thin steel plate. The output is a thermocouple measurement taken from the back of the plate. The sampling interval is 2 seconds and number of samples is 801. Figures 1 and 2 illustrated the input drive voltage and the output temperature in Celsius degrees of this case study. The database used was the *DaISy: Database for the Identification of Systems* (De Moor, 2009).

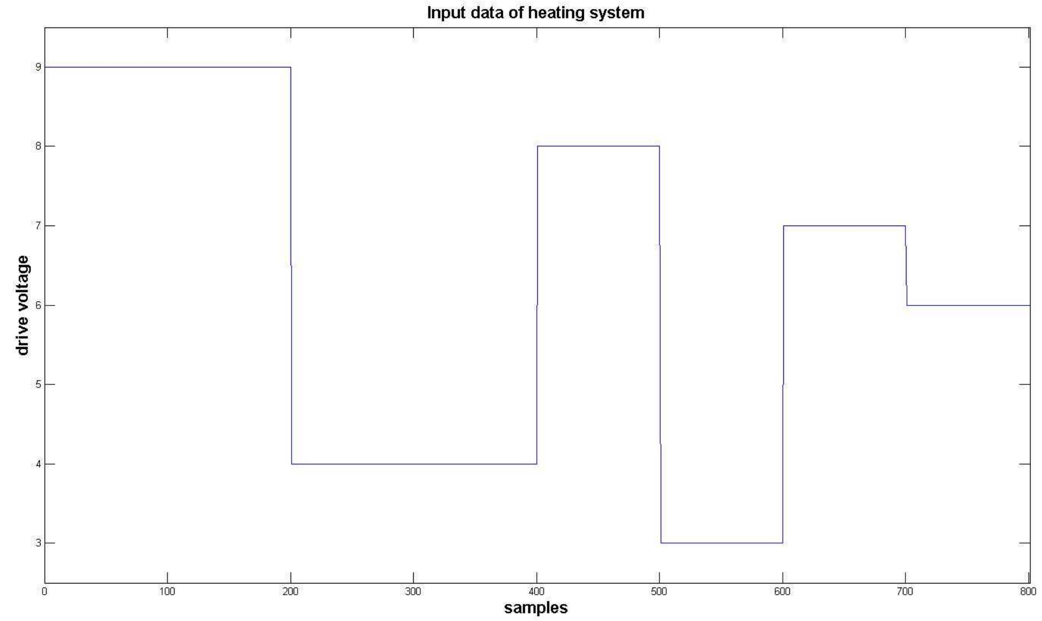

Figure 1. Input drive voltage

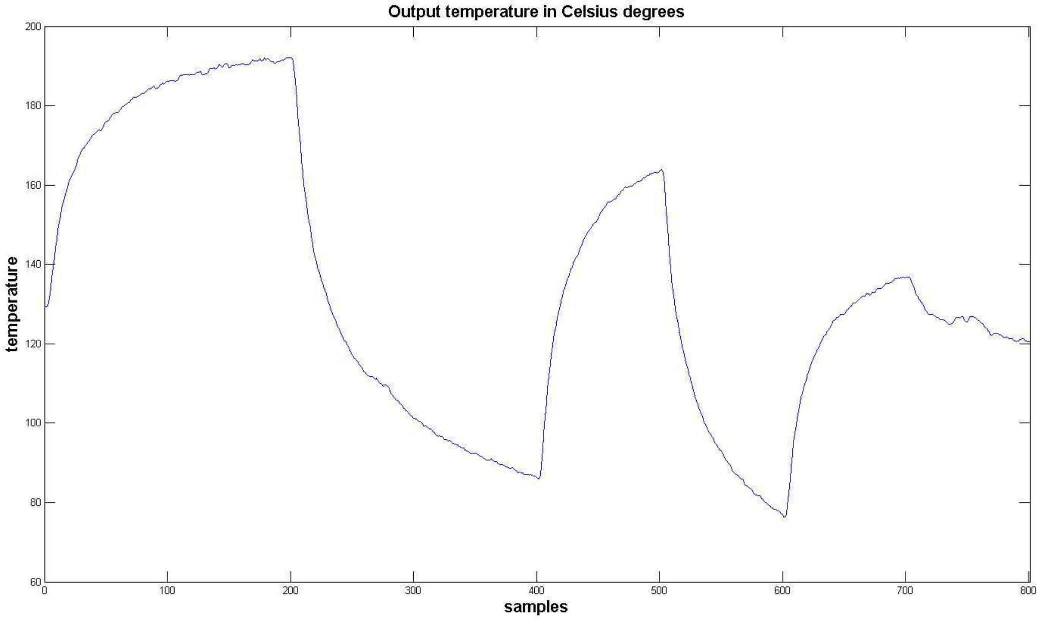

Figure 2. Output temperature in Celsius degrees

3. SYSTEM IDENTIFICATION AND RBF-NN WITH A TRAINING METHOD BASED ON PSO

The processes identification is a procedure to identify a model of an unknown process, for intentions of forecast and/or understanding of the dynamic behavior of the process. A model describes reality in some way, and system identification is the theory of how mathematical models for dynamical systems are constructed form observed data. Typically, a parameterized set of models, a model structure, is hypothesized and data is used to find the best model within this set according to some criterion. The choice of model structure is guided by prior knowledge or assumptions about the system which generated the data. When little prior knowledge is available it is common to use a black-box model. A black-box model is a standard flexible structure and it can be used to approximate a large variety of different systems (Sjöberg, 1995). Neural network models have proven to be successful non-linear black-box model structures in many applications (Hong and Chen, 2009; Huang and Du, 2008; Li *et al.*, 2009; Pappala *et al.*, 2009).

3.1. Fundamentals of RBF-NN

The neural networks consist in elements of processing highly interconnected called neurons. Each neuron has inputs and one output. The output of each neuron is determined as a nonlinear function of weighed sum of the inputs, however more complex mathematical operations could be included. The neurons are interconnected through weights, which are adjusted during the period of training.

Among the excellent characteristics of the neural networks there are: parallel processing, learning, associative and distributed memory. These characteristics are inspired in the biological neural networks (Bortman and Aladjem, 2009). Accordingly, RBF-NN is widely used in identification of nonlinear systems. The key problems of RBF-NN are the following: determining centers and widths of radial basic function, the number of hidden nodes, weights between hidden layer and output layer and the parameters of hidden layer are optimized locally, not globally (Chen *et al.*, 2007; Hong and Chen, 2009).

There are several representations for nonlinear system modeling. In this application we have chosen RBF-NN. RN-RBF design can be seen as a curve adjustment problem (function approximation problem) in a high dimensionality space. The radial basis function (or activation function) used in RBF-NN is Gaussian type as illustrated in Eq. (1). The estimated output is shown in Eq. (2). Fig. 1 shows the general structure of RBF-NN (Chen *et al.*, 2007; Huang and Wang, 2007) given by

$$f(x) = e^{-(\frac{x_i - c_j}{\sigma_j})} \tag{1}$$

where x_i is the input vector, c_j is the activation function center (Gaussian), and σ_j is the standard deviation.

$$\hat{y}(t) = \sum_{i=1}^{nc} w_m k_m \tag{2}$$

where *nc* is the clusters quantity (neurons), w_m are the weights, and k_m is the hidden layer output.

The clustering method used in this application by the RBF-NN for classification problems is fuzzy *c-means* algorithm (FCM), which was developed by Dunn (1973) and improved by Bezdek (1981). This algorithm is frequently used for standards recognition and is based on minimize of the objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \quad ,1 \le m < \infty \tag{3}$$

where u_{ij} is the degree of the set x_j in the group j, x_i is the element i of the measured data, c_j is the center of the group j, the parameter m is a weight that determines the degree to which partial members of a cluster affect the clustering result and $\|.\|$ is the norm between measured data and the center. The update of u_{ij} and c_j is given by:

$$\mu_{ij}(x) = \frac{1}{\sum_{k=1}^{C} \left(\frac{\| x_i - c_j \|^2}{\| x_i - c_k \|^2} \right)^{\frac{2}{m-1}}} \tag{4}$$

and

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} (u_{ij}^m)} . \tag{5}$$

3.2. Particle Swarm Optimization Algorithm

The PSO is responsible for optimizing the centers obtained by *c-means* and also optimizing the Gaussian spreads. PSO has a population with random positions, each of these particles has 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 (Chen *et al.*, 2007; Huang and Wang, 2007).

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 these equations:

$$V_{id}^{t+1} = W \cdot V_{id}^{t} + c_1 \cdot rand_1 \cdot \left(P_{id} - X_{id}^{t}\right) + c_2 \cdot rand_2 \cdot \left(P_{gd} - X_{id}^{t}\right) \tag{6}$$

$$X_{id}^{t+1} = X_{id}^{t} + V_{id}^{t+1} \cdot \Delta t \tag{7}$$

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 generated with uniform distribution in range [0,1], used to maintain the population diversity. Equation (6) is used to update each particle’s speed, for it calculation the speed in last iteration, multiplied by an inertial weight (Lin *et al.*, 2006).

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*). The Equation (7) represents the update position of a particle, according with its previous position and its actual speed, considering $\Delta t = 1$. One of the main reasons for the PSO attractiveness is the need to adjust few parameters (Xie *et al.*, 2002).

Constants c_1 and c_2 are positive 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). However, recent literature informs that the choice may be even better if the cognition parameter is 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 makes possible a global search, on 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 the adoption of co-evolution, meta-optimization of fuzzy systems (Xiao and Wang, 2006; Zhan *et al.*, 2009).

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. (8) is adopted, for W update, where t_{max} is the maximum number of iterations and t is the actual iteration (Shi and Eberhart, 2002), where

$$W = W_{\max} - \frac{W_{\max} - W_{\min}}{t_{\max}} t . \tag{8}$$

The linear optimization method to make 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. (9) given by

$$w_m = \left(\left(k^T k\right)^{-1} k^T\right) y(t) \tag{9}$$

where *y(t)* is the desired output.

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 $\hat{y}(t)$, is realized by Eq. (10).

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i(t) - \hat{y}_i(t))^2}{\sum_{i=1}^{n} (y_i(t) - \bar{y})^2}, \tag{10}$$

where *n* is the number of measured samples of the process output. In this context, 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. SIMULATION RESULTS

In Table 1, the identification results of heating system using a RBF-NN using *c-means* for clustering and optimized by pseudo-inverse and PSO with concepts of one-step-ahead prediction are presented. In the estimation phase (training of RBF-NN) 400 samples were used, and in the validation phase 401 different samples were used.

For the results, were performed 10 simulations with different numbers of delayed inputs (*Nu*), delayed outputs (*Ny*) and centers. Table 1 contains 5 simulations using *Nu* equal to 2, *Ny* equal to 1 and the number of centers was simulated with 2, 3, 4, 5 and 6. Table 1 presents 5 simulations using *Nu* equal to 2, *Ny* equal to 1 and the number of centers was simulated with 2, 3, 4, 5 and 6. Table 2 also contains 5 simulations, but changing the *Ny* to 2. The results obtained for these simulations are the R^2*_est* and R^2*_val* (estimation and validation phases).

On the Table 1, the best result was found in simulation 4, and the Fig. 3 illustrated the real and estimated output graphic of the heating system. And on Table 2, the best result was found in simulation 9, and the Fig. 4 illustrated the real and estimated output graphic. Every simulation was included at least one delayed output to the RBF-NN, therefore improving the results. The reason for this improvement is that the RBF-NN obtains more information about the nonlinear dynamic, improving one-step-ahead identification. But special care must be taken, because when the number of *Nu* and *Ny* increases, the complexity of the model increases too.

Table 1. Results with different numbers of centers using RBF-NN with a training method based on PSO (inputs of RBF-NN are two delayed inputs and one delayed output)

Number of simulation	*Nu*	*Ny*	Number of centers	R^2*_est*	R^2*_val*
1	2	1	2	0.8599	0.7106
2	2	1	3	0.9999	0.9997
3	2	1	4	0.9999	0.9998
4	2	1	5	0.9999	0.9998
5	2	1	6	0.9999	0.9997

Table 2. Gaussian centers of the best simulation (simulation 4).

Cluster	center c_j of *u*(*t*-2)	*u*(*t*-1)	*y*(*t*-1)
1	1.0000	0.9997	0.9707
2	0.4439	0.4436	0.6165
3	0.4441	0.4440	0.4883
4	0.9998	0.9988	0.8053
5	0.4450	0.4429	0.8668

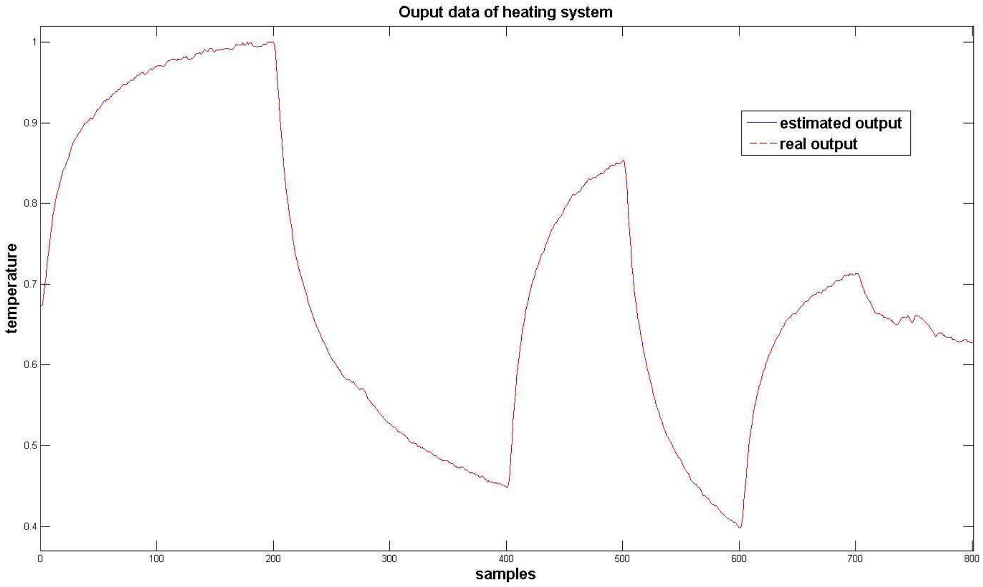

Figure 3. Output data of heating system with estimated and real output (simulation 4)

Table 3. Results with different numbers of centers using RBF-NN with a training method based on PSO (inputs of RBF-NN are two delayed inputs and one delayed output)

Number of simulation	*Nu*	*Ny*	Number of centers	R^2*_est*	R^2*_val*
6	2	2	2	0.8976	0.8162
7	2	2	3	0.9998	0.9996
8	2	2	4	0.9998	0.9992
9	2	2	5	0.9999	0.9996
10	2	2	6	0.9998	0.9468

Table 4. Gaussian centers of the best simulation (simulation 9).

Cluster	center c_j of $u(t\text{-}2)$	$u(t\text{-}1)$	$y(t\text{-}2)$	$y(t\text{-}1)$
1	1.0000	0.9997	0.9716	0.9722
2	0.4473	0.4449	0.8661	0.8519
3	0.4448	0.4446	0.6159	0.6132
4	09998	0.9987	0.8125	0.8187
5	0.4446	0.4445	0.4882	0.4873

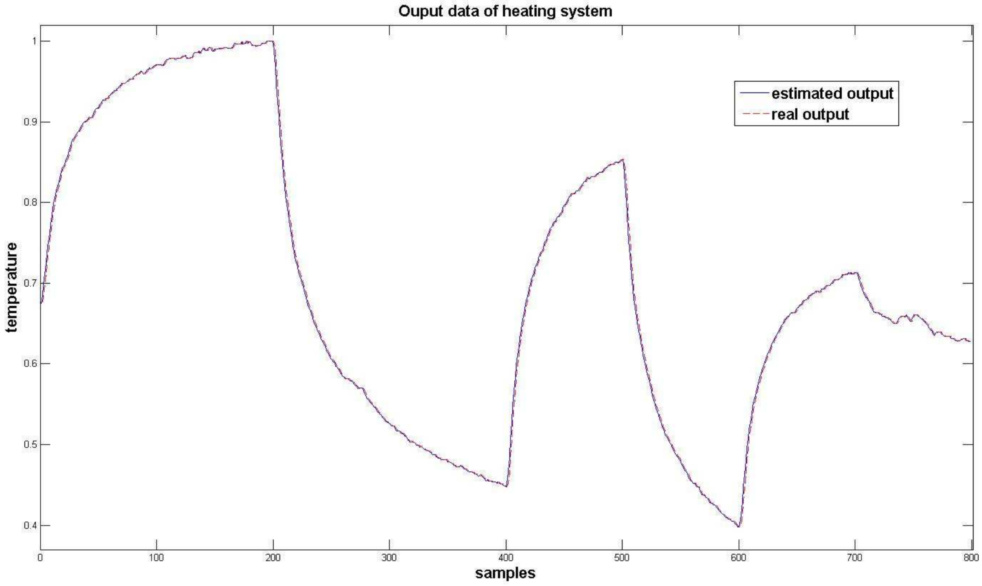

Figure 4. Output data of heating system with estimated and real output (simulation 9)

5. CONCLUSIONS AND FUTURE RESEARCH

This paper presented a methodology including RBF-NN, the clustering algorithm *c-means* and optimization by PSO. The tested case study was a heating system which had a drive voltage as its input, and the temperature as output. For obtaining the results all the methods were described and put under context.

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

The *c-means* algorithm is sensitive to the earlier choices of the cluster, demanding a proper initialization to obtain correct results. Using an algorithm to make those choices can solve the problem, initializing with the centers close to the final centers, making sure that the number of iterations will be reduced.

Finally, the obtained results were considered satisfactory, showing that the present methodology can achieve the identification of the analyzed nonlinear system. The results could be observed on graphics and tables, where the multiple correlation coefficient was presented on estimation and validation phase. Therefore, the methodology proved that it can be applied to other type of systems, such as chaotic system or even multivariable systems.

6. REFERENCES

Alligood, K. T., Sauer, T. D. and Yorke, J. A.,1996, "Chaos: An Introduction to Dynamical Systems", Springer, Heidelberg, Germany.

Bezdek J.C., 1981, "Pattern Eecognition with Fuzzy Objective Function Algorithms", Ed. Plenum Press, NY, USA.

Bortman, M. and Aladjem, M., 2009, "A Growing and Pruning Method for Radial Basis Function Networks", IEEE Transactions on Neural Networks (accepted to future publication).

Chen, S., Mei, T., Luol, M. and Yang, X., 2007, "Identification of Nonlinear System Based on a New Hybrid Gradient-Based PSO Algorithm", Proceedings of the 2007 International Conference on Information Acquisition, Korea, pp 265-268.

Coelho L. S. and Guerra, F. A., 2002, "Rede Neural de Base Radial Aplicada à Identificação de um Processo Experimental Multivariável", Anais do XIV Congresso Brasileiro de Automática, Vol. 1, Natal, Brazil, pp. 1-6 (in Portuguese).

De Moor B.L.R. (ed.), "DaISy: Database for the Identification of Systems, Department of Electrical Engineering", ESAT/SISTA, K.U.Leuven, Belgium, URL: http://homes.esat.kuleuven.be/~smc/daisy/, 13/04/2009. [Used dataset: Heating system, section Thermic Systems, code number [99-001]] .

Dunn J.C., 1973, "A fuzzy relative of the ISODATA process and it's use in detecting compact well-separated clusters", Journal of Cybernetics, Vol. 3, pp. 32-57.

Gudise V.G. and Venayagamoorthy G.K., 2003, “Evolving digital circuits using particle swarm”, Proc of the Int Joint Conf on Neural Networks, Portland, USA, pp. 468-472.

Haykin, S.,2000, “Redes Neurais: Princípios e Aplicações”, Ed. Bookman, Porto Alegre, Brazil.

Hong, X. and Chen, S., 2009, “A New RBF Neural Network With Boundary Value Constraints”, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 39, No. 1, pp 298-303.

Huang, C.-M. and Wang, F.-L., 2007, “An RBF Network With OLS and EPSO Algorithms for Real-Time Power Dispatch”, IEEE Transactions on Power Systems, Vol. 22, No. 1, pp. 96-104.

Huang, D.S.and Du, J.X., 2008, "A Constructive Hybrid Structure Optimization Methodology for Radial Basis Probabilistic Neural Networks", IEEE Transactions on Neural Network, Vol. 19, No. 12, pp 2099-2115.

Ioh, M.; Yang, T. and 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.

Li, K., Peng, J.-X. and Bai, E.-W., 2009, “Two-Stage Mixed Discrete–Continuous Identification of Radial Basis Function (RBF) Neural Models for Nonlinear Systems”, IEEE Transactions on Circuits and Systems, Vol. 56, No. 3, pp 630-643.

Lian, K. W. and 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.

Lin, C.-L., Hsieh, S.-T., Sun, T.-Y. and Liu, C.-C., 2006, “Cluster Distance Factor Searching by Particle Swarm Optimization for Self-Growing Radial Basis Function Neural Network”, International Joint Conference on Neural Networks, pp. 4825-4830.

Lo, J.T.-H., 1998, “Multilayer Perceptrons and Radial Basis Functions are Universal Robust Approximators”, International Joint Conference on Neural Networks, Anchorage, AK, Vol. 2, pp. 1311-1314.

Mcloone, S.; Brown, M. D.; Irwin, G. and 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. and Parthasarathy, K.,1990, “Identification and Control of Dynamical Systems Using Neural Networks”, IEEE Transactions on Neural Networks, Vol. 1, No. 1, pp. 4-27.

Pappala, V.S., Erlich, I. and Rohrig, K., 2009, “A Stochastic Model for the Optimal Operation of a Wind-Thermal Power System”, IEEE Transactions on Power Systems, Vol. 24, No. 2, pp 940-950.

Parsopoulos, K.E. and 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. and He, Z., 1999, “Identification of Dynamical Noise Levels in Chaotic Systems and Application to Cardiac Dynamics Analysis”, International Joint Conference on Neural Networks, IJCNN, Vol. 5, Washington, DC, USA, pp. 3680-3684.

Schaible, B., Xie, H. and 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. and Eberhart, R.C., 1998, “Parameter Selection in Particle Swarm Optimizer”, Proceedings of 7th Annual Conference on Evolutionary Programming, Lecture Notes in Computer Science, Springer, pp. 591-601.

Shi, Y. and Eberhart, R.C., 2002, “Fuzzy Adaptive Particle Swarm Optimization”, Proceedings of the Congress on Evolutionary Computation, Vol. 1, pp. 101-106.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Deylon, B., Glorennec, P.-Y., Hjalmarsson, H. and Juditsky, A., “Nonlinear Black-Box Modeling in System Identification: a Unified Overview”, Automatica, Vol. 31, No. 12, pp. 1691-1724, 1995.

Xiao, J.-M. and Wang, X.-H., 2006, “Nonlinear Neural Network Predictive Control for Power Unit using Particle Swarm Optimization”, Proceedings of the 5th International Conference on Machine Learning and Cybernetics, Dalian, China, pp. 2851-2856.

Xie, X., Zhang, W. and Yang, Z., 2002, “A Dissipative Particle Swarm Optimization”, Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, HI, USA, Vol. 2, pp. 1456-1461.

Zhan, Z.-H,. Zhang, J. Li, Y. and Chung, H. S.-H., 2009, “Adaptive Particle Swarm Optimization”, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 39, No. 1, pp 1-20.

7. RESPONSIBILITY NOTICE

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