ABCM Symposium Series in Mechatronics - Vol. 4 - pp.288-296

COMBINED APPROACH OF RBF NEURAL NEWORK, GENETIC ALGORITHM AND LOCAL SEARCH AND ITS APPLICATION IN IDENTIFICATION OF A NONLINEAR PROCESS

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

Hugo Vieira Neto, hvieir@utfpr.edu.br
Federal University of Technology - Paraná (UTFPR), Eletrical Engineering and Industrial Information Technology Graduate Program (CPGEI).
Av. 7 de Setembro, 3165, Centro, Zip code 80230-901, 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.

Abstract. *The identification of nonlinear systems by artificial neural networks has been successfully applied in many applications. In this context, the radial basis function neural network (RBF-NN) is a powerful approach for nonlinear identification. A RBF neural network has an input layer, a hidden layer and an output layer. The neurons in the hidden layer contain Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the neuron. In this paper, a combined approach including RBF-NN neural network with training based on genetic algorithm (GA) and local search is presented. During the identification procedure, the GA with local search aims to optimize the parameters of RBF-NN and the optimum values are regarded as the initial values of the RBF-NN parameters. The validity and accuracy of identification of RBF-NN model are tested by simulations, whose results reveal that it is feasible to establish a good model for a nonlinear process of pH neutralization.*

Keywords*: genetic algorithm, nonlinear identification, radial basis function neural networks, nonlinear process, local search, optimization.*

1. INTRODUCTION

A model describes real systems in some way, and system identification is the theory of how mathematical models for dynamical systems are constructed from 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 guide 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.

One common assumption in system identification is that the unknown system is linear. This is never true in real applications, but often it is a good approximation. Linear system theory is very well developed and there exist many results which can be applied to the obtained linear model. One reason why system identification is much harder for non-linear models than for linear models concerns the choice of model structure. For non-linear models there are many more alternatives than for linear models.

Neural network models have proven to be successful non-linear black-box model structures in many applications and they have attracted a growing interest in the past 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).

The main objective of this paper is to present an optimization approach for nonlinear identification using RBF-NN of pH neutralization process. The RBF-NN uses the *k-means* clustering algorithm, and is optimized by pseudo-inverse and GA (Yu *et al.*, 2004; Zhang and Baia, 2005; Zuo and Liu, 2004).

The reminder of this paper is organized as follows. In section 2, the pH neutralization process is presented. In section 3, the one-step-ahead prediction for system identification with RBF-NN with a method training based on GA

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.288-296

and local search is discussed. The simulation results are presented in section 4. The conclusions and future works are discussed in section 5.

2. DESCRIPTION OF CASE STUDY

The identification case study boarded in the paper is the nonlinear dynamic system of a pH neutralization in a constant volume stirring tank. The system is an experiment is a double-input single-output neutralization process. The input one is acid solution flow and input two is the base solution flow. The output is a pH of the solution in the tank. The sampling interval is 10 seconds and number of samples is 2001. The volume of the tank is 1100 liters and the concentration of the acid solution (HAC) and base solution (NaOH) are 0.0032 Mol/l and 0.05 Mol/l, respectively McAvoy *et al*., 1972). Fig. 1 illustrates the inputs and output of the neutralization process. The used database is in *DaISy: Database for the Identification of Systems* (De Moor, 2009).

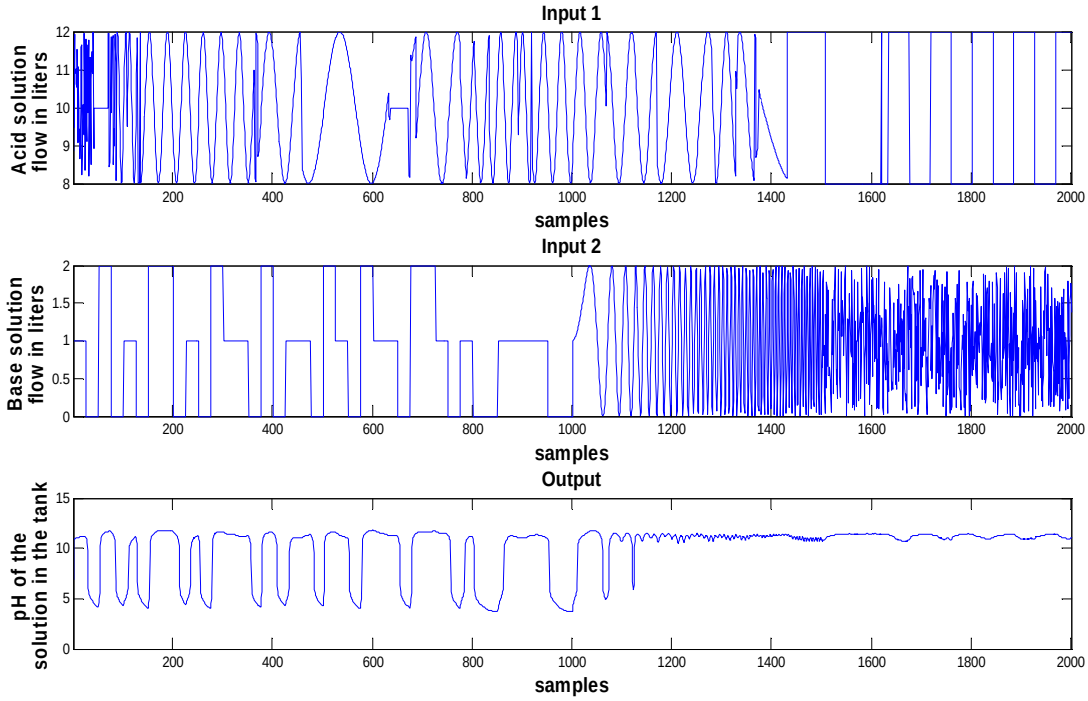

Figure 1. Inputs and output of case study

3. NEURAL NETWORKS AND GENETIC ALGORITHM FOR SYSTEM IDENTIFICATION

System identification is a process that requires the modeler involvement (Chen *et al*., 1990; Ljung, 1997). The designer must analyze which system's variables are relevant for the modeling, and if the chosen structure model is adequate, otherwise, he must take the necessary decisions to solve the problem (Cassini and Aguirre, 1999). 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 modeling with chaotic behavior. 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).

3.1. Radial Basis Function Neural Network

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).

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.288-296

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. 2 shows the general structure of RBF-NN.

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

where:

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

and

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

where:

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

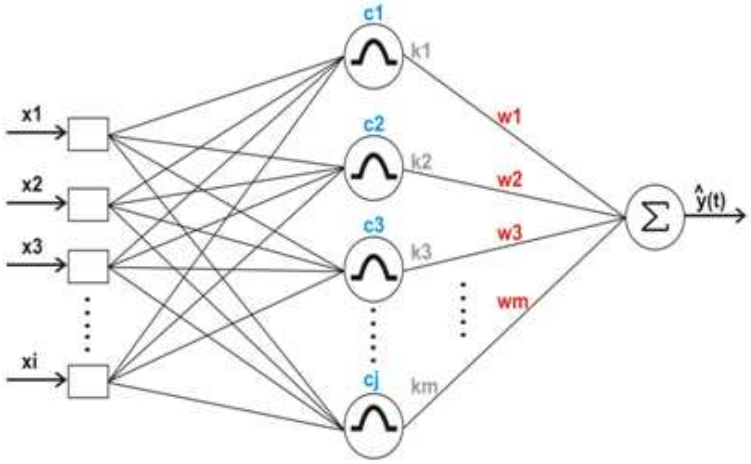

Figure 2. General structure of RBF-NN

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

$$\| x_i - c_{j^*} \| = \min_j \| x_i - c_j \| \tag{3}$$

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

$$c_j = \frac{1}{m_j} \sum_{x_i \in j} x_i \tag{4}$$

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.

3.2. Genetic Algorithm and Local Search

A Genetic Algorithm emulates evolutionary biological theories based on Natural Selection Theory. GA can be useful to solve optimization problems (Ribeiro-Filho and Treleaven, 1994). It manipulates a population of individuals, to preserve the individuals if good genetic characteristics, which can represent more promising areas of the search

space, in the optimization point of view. The basic idea of the GA is to generate an initial population formed by a random group of individuals which represents possible solutions for the problem. During the evolutionary process of the GA, this population is evaluated, and for every individual a parameter (named fitness) is given, reflecting its adaptation ability for the problem. A percentage of the more adapted individuals are maintained, and the others eliminated, through the selection method. The ones which are maintained by the selection can have its characteristics modified by genetic operators as mutation and recombination, generating descendents for the next generation. This process is repeated until a set of satisfactory solutions is found (Goldberg, 1989).

The operation of the GA is summarized in the following steps:

Step 1: Randomly generate an initial population;
Step 2: Compute and save the fitness for each individual in the current population;
Step 3: Select the best individuals;
Step 4: Genetic manipulation to create the new population of strings.

The main feature of the GA is that it performs a global search for the solution in the search space. Aiming to supplement this feature, a local search technique was used in this work (Yun, 2006; Oh and Lee, 2002). The method chosen for the local search uses the simplex search method (Nelder and Mead, 1965). This is a direct search method that does not use numerical or analytic gradients. It basically finds the minimum of a scalar function of several variables, starting at an initial estimate. A simplex is the geometrical figure in *n* dimensions consisting of *n+1* vertices (in two-space, is a triangle; in three-space is a pyramid). The Simplex Algorithm for minimization takes such a set of *n+1* points and attempts to move them into a minimum. The simplex formed from the points should be non-degenerate, it should have a non-zero volume.

The simplex method comprises the following steps:

Step 1: Find an initial basic feasible solution;
Step 2: Verify that the current solution is optimal. If so, stopping;
Step 3: Determine the non-basic variable that should enter in the base;
Step 4: Determine the basic variable that is to leave the base;
Step 5: Find a new basic feasible solution and return to **Step 2**.

In this context, the basic variables are the solution to the problem. The simplex method has been implemented at the end of the genetic algorithm, refining the local search phase of the proposed methodology.

3.3 Other Aspects

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. (5) given by

$$w_m = \left(\left(k^T k\right)^{-1} k^T\right) y(t) \tag{5}$$

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, Eq. (6) , between real output *y(t)* and the estimated output $\hat{y}(t)$.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i(t) - \hat{y}_i(t))^2}{\sum_{i=1}^{n} (y_i(t) - \bar{y})^2} \tag{6}$$

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. SIMULATIONS AND RESULTS

In Table 1, the pH process system identification results using a radial basis function neural network using *k-means* for clustering and optimized by pseudo-inverse and GA with or without Local Search concepts of one-step-ahead prediction are presented. In the estimation phase (training of RBF-NN) 1000 samples were used, and in the validation phase 1001 different samples were used.

For the results, were performed 10 simulations with different numbers of centers and search method (with and without local search). Table 1 presents 5 simulations using 2 delayed inputs (*Nu*), 2 delayed outputs (*Ny*) and the

number of centers was simulated with 2, 3, 5, 8 and 10. The Table 3 contains 5 simulations using *Nu* equal to 2, *Ny* equal to 2 and the number of centers was simulated with 2, 3, 5, 8 and 10. 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 5, and the Fig. 3 illustrated the real and estimated output graphic of the pH neutralization system. And on the Table 3, the best result was found in simulation 10, and the Fig. 5 illustrated the real and estimated output graphic. The figures 4 and 6 represent the sample percent error for best solutions using GA without local search and GA with local search, respectively.

Table 1. Experimental results with different numbers of centers (input layer of RBF-NN with two delayed inputs and two delayed outputs) using RBF-NN with a training method based on GA.

Simulation	N_u	N_y	Number of centers	R^2 *(est)*	R^2 *(val)*
1	2	2	2	0.9199	0.8843
2	2	2	3	0.9338	0.9222
3	2	2	5	0.9528	0.9436
4	2	2	8	0.9603	0.9587
5	2	2	10	0.9661	0.9593

Table 2. Gaussian centers of the best simulation (simulation 5).

Cluster	Best spreads (simulation 5)
1	0.8173
2	0.0153
3	0.8792
4	0.2004
5	0.6610
6	0.9701
7	0.8025
8	0.9852
9	0.2664
10	0.0484

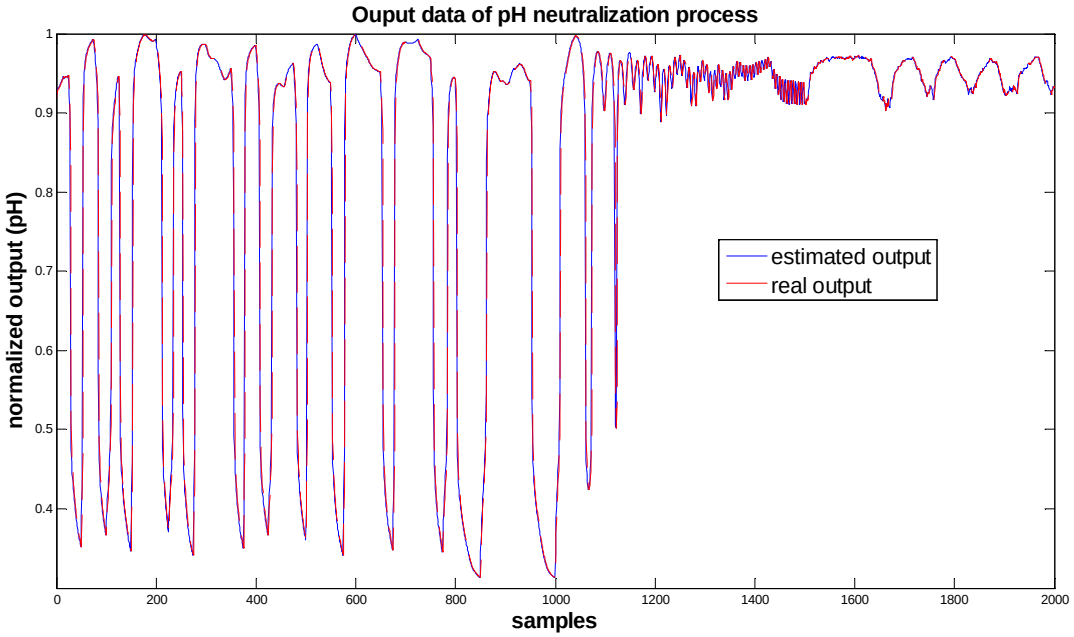

Figure 3. Output data of pH neutralization system with estimated and real output (simulation 5)

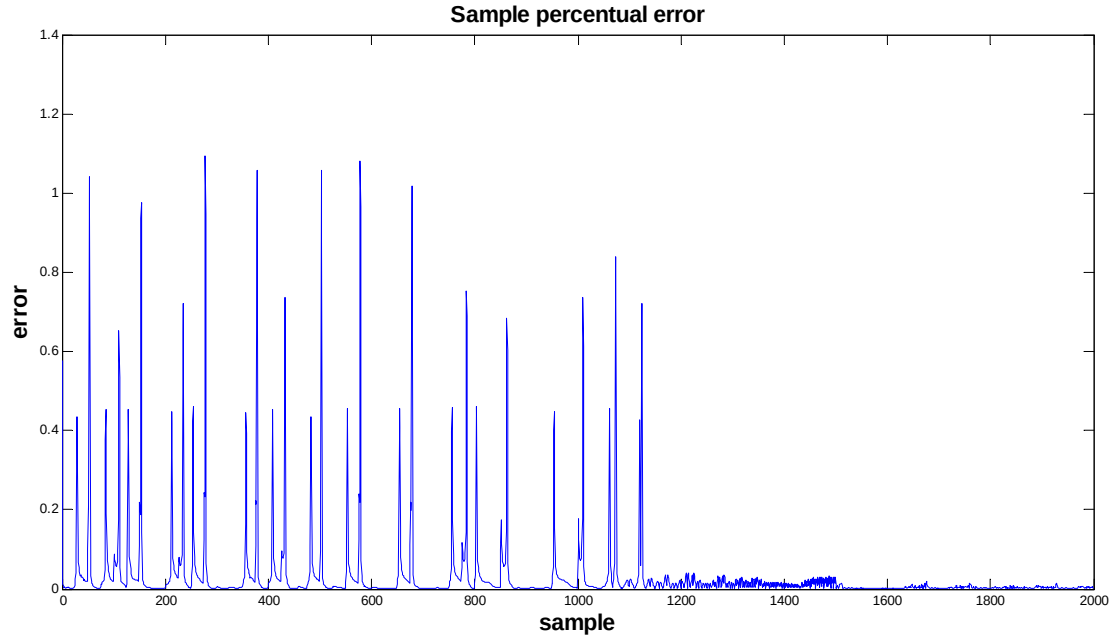

Figure 4. Percent Error (simulation 5)

Table 3. Experimental results with different numbers of centers (input layer of RBF-NN with two delayed inputs and two delayed outputs) using RBF-NN with a training method based on GA and local search.

Simulation	N_u	N_y	Number of centers	R^2 *(est)*	R^2 *(val)*
6	2	2	2	0.9233	0.8915
7	2	2	3	0.9416	0.9185
8	2	2	5	0.9626	0.9556
9	2	2	8	0.9620	0.9545
10	2	2	10	0.9620	0.9597

ABCM Symposium Series in Mechatronics - Vol. 4 - pp.288-296

Table 4. Gaussian centers of the best simulation (simulation 10).

Cluster	Best spreads (simulation 10)
1	0.6732
2	0.1070
3	1.0044
4	1.2472
5	0.0001
6	0.0882
7	0.0207
8	0.2515
9	0.9598
10	0.6321

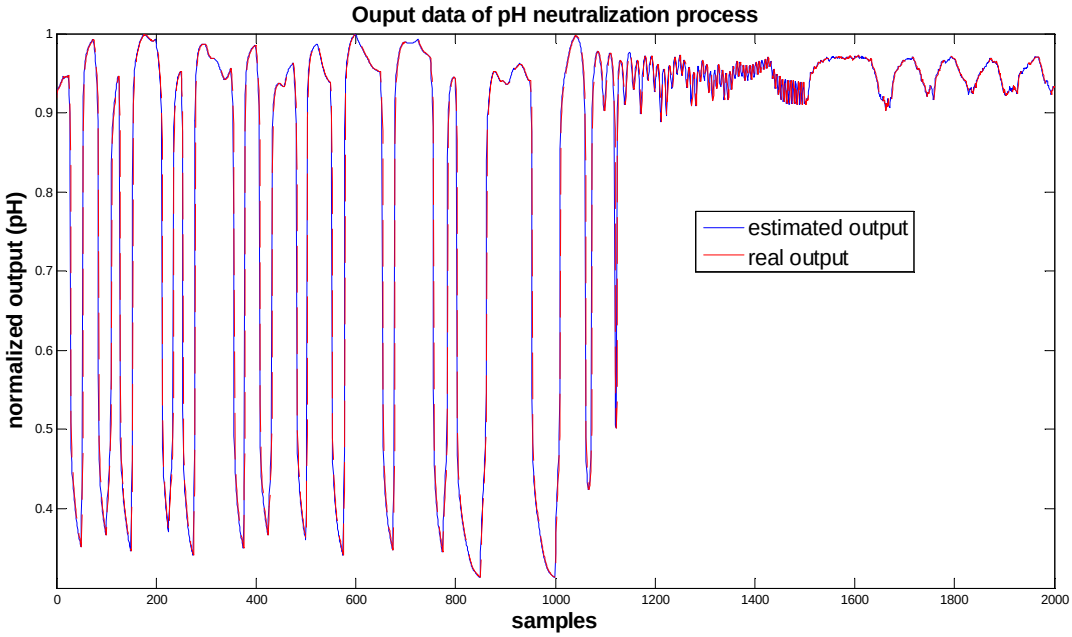

Figure 5. Output data of pH neutralization system with estimated and real output (simulation 10)

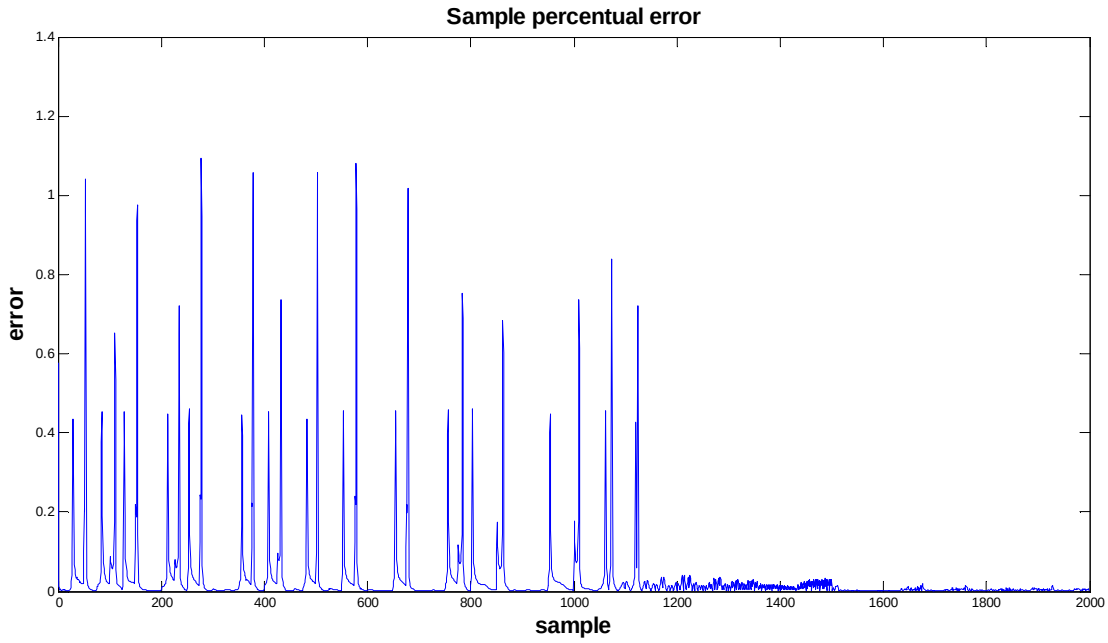

Figure 6. Percent Error (simulation 10)

5. CONCLUSION AND FUTURE RESEARCH

This paper demonstrated a combined approach of RBF-NN, GA and Local Search and its application in one-step-ahead identification of a pH neutralization process. The centers of the RBF-NN are obtained by the *k-means* clustering method, the weights by the pseudo-inverse liner optimization and the Gaussian spreads by GA. The Genetic Algorithm is developed and compared with or without a local search.

The obtained results show that the application of this proposed methodology is viable regarding R^2. Some aspects must be considered on the process. First, *k-means* clustering method is comparatively simple and allows establish initial conditions that can improve its performance. A disadvantage of this method is the need to inform the number of groups (centers) that the algorithm will use. Regarding the GA, it is very versatile in the methodology, mainly because it has the possibility of hybridization with other methods, like local search. The local search is responsible for an improvement on performance of GA, which characteristic is the global search.

For future researches, a clustering method that obtains automatically the number of centers can be developed. Also other optimization algorithms can be applied, such as Ant Colony Optimization.

Finally, is important to mention that the obtained results depend exclusively of the researcher, since the N_u and N_y values are selected *a priori* and are associated with his experience. Besides, the values act directly on the system's complexity.

6. REFERENCES

Cassini, C. C.; Aguirre, L. A., 1999, "Uma Introdução à Identificação de Sistemas Não-Lineares, Modelagem, Análise e Controle de Sistemas Não-Lineares", Editora da UFMG (in Portuguese).

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.

De Moor B.L.R. (ed.), 2009, 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/*

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

Haykin, S., 2000, "Redes Neurais: Princípios e Aplicações", Ed. Bookman, Porto Alegre, Brazil (in Portuguese).

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.

Lo, J.T.-H., 1998,"Multilayer Perceptrons and Radial Basis Functions are Universal Robust Approximators", International Joint Conference on Neural Networks, Anchorage, AK, USA, 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.

McAvoy, T.J., Hsu, E. and Lowenthal, S., 1972, "Dynamics of pH in Controlled Stirred Tank Reactor", Industrial & Engineering Chemistry Process Design Development, vol. 11, pp. 71-78.

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.

Nelder, J. A., Mead, R., 1965, "A Simplex Method for Function Minimization", Computer Journal, Vol. 7, pp. 308-313.

Oh, I. S., Lee, B. R., 2002, "Local Search-Embedded Genetic Algorithms for Feature Selection", Proceedings of 16th International Conference on Pattern Recognition, Volume 2.

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.

Ribeiro-Filho, J. and Treleaven, P. C., 1994, "Genetic-Algorithm Programming Environments", IEEE Computer, Vol. 27, No. 6, pp. 28-43.

Schaible, B., Xie, H. and Lee, Y. C., 1997, "Fuzzy Logic Models for Ranking Process Effects", IEEE Transactions on Fuzzy Systems, Vol. 5, pp. 545-556.

Yu, H., Zhu, Y., Shi, Y., 2004, "RBF Networks Trained by Genetic Algorithm Applied in Active Control of Noise and Vibration", Acoustical Science and Technology, Vol. 25, No. 1, pp.109-111

Yun, Y., 2006, "Hybrid Genetic Algorithm with Adaptive Local Search Scheme", Source Computers and Industrial Engineering, Vol. 51, No. 1, pp. 128–141.

Zhang, L. and Baia, Y. F., 2005, "Genetic Algorithm-trained Radial Basis Function Neural Networks for Modeling Photovoltaic Panels", Engineering Applications of Artificial Intelligence, Vol. 18, No. 7, pp. 833-844.

Zuo, G. and Liu, W., 2004, "Genetic algorithm Based RBF Neural Network for Voice Conversion", Intelligent Control and Automation, Vol. 5, pp. 4215-4218.

7. RESPONSIBILITY NOTICE

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