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

COMBINED APPROACH OF RBF NEURAL NETWORKS, GENETIC ALGORITHMS AND LOCAL SEARCH AND ITS APPLICATION TO THE IDENTIFICATION OF A NON-LINEAR 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 Systems Division (DVSE), Electronic Systems Division (DVEL)
Centro Politécnico da UFPR, BR 116, km 98, Jardim das Américas, 81531-980, Curitiba, Paraná, Brazil.

Hugo Vieira Neto, hvieir@utfpr.edu.br
Federal University of Technology - Paraná (UTFPR), Post-graduation Program in Eletrical Engineering and Industrial Informatics (CPGEI)
Avenida Sete de Setembro, 3165, Centro, 80230-901, Curitiba, Paraná, Brazil.

Leandro dos Santos Coelho, leandro.coelho@pucpr.br
Pontifical Catholic University of Paraná (PUCPR), Post-graduation Program in Production and Systems Engineering (PPGEPS), Automation and Systems Laboratory (LAS)
Rua Imaculada Conceição, 1155, 80215-901, Curitiba, Paraná, Brazil.

Abstract. *The identification of non-linear 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 non-linear system identification. An 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 decrease exponentially according to the square of the distance to the center of the neuron. In this paper, a combined approach including an RBF-NN neural network with training based on a genetic algorithm (GA) and local search is presented. During the identification procedure, the GA guided by local search aims to optimize the parameters of the RBF-NN and the optimal values are regarded as the initial values of the RBF-NN parameters. The validity and accuracy of system identification using the RBF-NN model are tested by simulations, whose results reveal that it is feasible to establish a good model for a non-linear process of pH neutralization.*

Keywords*: genetic algorithms, non-linear system identification, radial basis function neural networks, non-linear process, local search, optimization.*

1. INTRODUCTION

Models are simplified abstractions that describe relevant features of real systems, and system identification is the theory that allows the construction of mathematical models for dynamical systems 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 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 common assumption in system identification is that the unknown system is linear. This is never true in real applications, but is often a good approximation. Linear system theory is very well developed and there are many methods which can be applied to obtain a linear model. However, system identification is much harder for non-linear models than for linear models and one of the main reasons for that concerns the choice of model structure. For non-linear models there are many more alternatives than for linear models.

Neural network models have been 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 the functionality of biological neural networks and may learn complex functional relations through a limited number of training data. Neural networks may serve as black-box models of non-linear 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 architecture consists of multiple simple processing elements, called neurons, interconnections among them and the weights attributed to these interconnections. The relevant information of such approach is stored in the weights (Haykin, 2000; Pei and He, 1999).

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

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 using an RBF-NN with a training method based on GA and local search is discussed. The simulation results are discussed in section 4. Finally, conclusions and future work are presented in section 5.

2. CASE STUDY

The system identification case study in this paper refers to the non-linear dynamic system of pH neutralization in a constant volume stirring tank, which is a chemical experiment involving a double-input single-output process. Input one is the acid solution flow, input two is the base solution flow and the output is the pH measurement of the solution in the tank. The sampling interval is 10 seconds and the number of samples is 2001. The volume of the tank is 1100 liters and the concentrations of the acid solution (HAC) and base solution (NaOH) are 0.0032 Mol/l and 0.05 Mol/l, respectively. Figure 1 illustrates the behavior of the inputs and the output of the neutralization process. The used data is available in the *DaISy: Database for the Identification of Systems* (De Moor B.L.R. (ed.)).

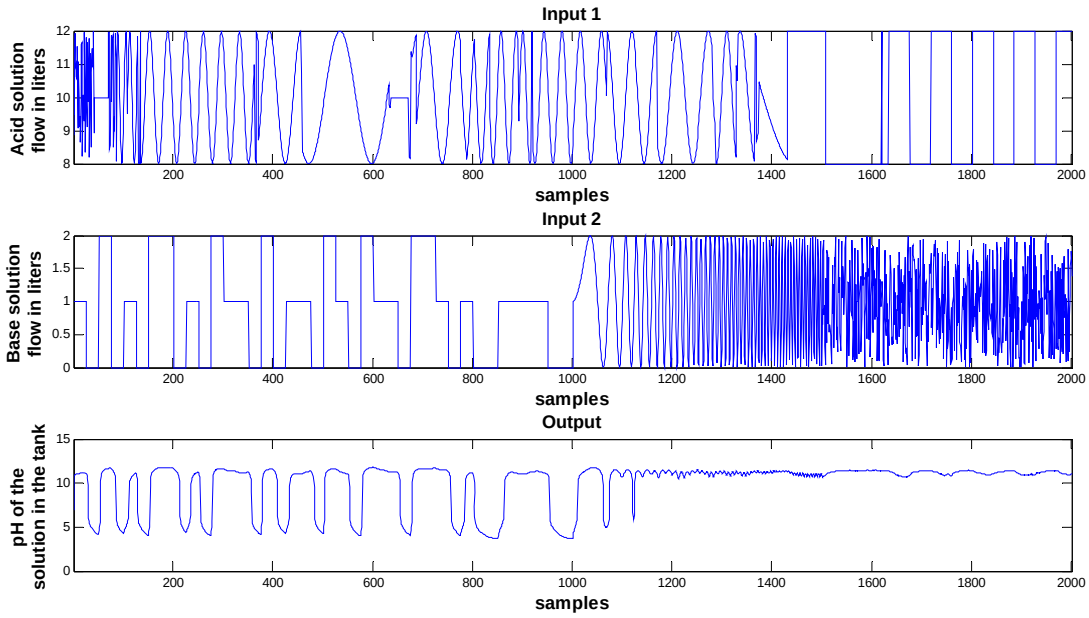

Figure 1. Case Study Inputs and Output

3. NEURAL NETWORKS AND GENETIC ALGORITHM FOR SYSTEM IDENTIFICATION

System identification is a process that requires involvement of the modeler (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, the designer must take the necessary decisions to solve the problem (Aguirre, 2000). The following steps may be quoted in the identification system process: (i) experimentation; (ii) non-linear detection; (iii) structure model determination; (iv) parameter estimation; and (v) validation phase.

There are several representations for non-linear system modeling with chaotic behavior and for this application we chose an RBF-NN. The design of a neural network in this context can be understood as a curve fit problem (function approximation) in a high dimensional space. For this task, RBF-NN learning is equivalent to finding a surface in a multidimensional space that better fits the training data set, using statistical measurements as the criteria for best fit (Chen *et al.*, 1990; Jang and Sun, 1993).

3.1. Radial Basis Function Neural Networks

RBF neural networks are flexible tools that are suitable for modeling dynamic environments. They have the ability to learn complex patterns and tendencies present in data quickly and also adapt to changes. Such characteristics make them adequate to temporal series prediction, especially those ruled by non-linear or non-stationary processes (Lo, 1998).

The radial basis function (or activation function) used in RBF-NN neurons is usually Gaussian, as illustrated in Eq. (1). The estimated output of the network is shown in Eq. (2) and the general structure of an RBF-NN in Fig. 2.

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

$$f(x) = e^{-\left(\frac{x_i - c_j}{\sigma_j}\right)^2}, \quad (1)$$

where:

x_i: input vector;
c_j: activation function center;
σ_j: standard deviation.

$$\hat{y}(t) = \sum_{i=1}^{n} w_i k_i, \quad (2)$$

where:

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

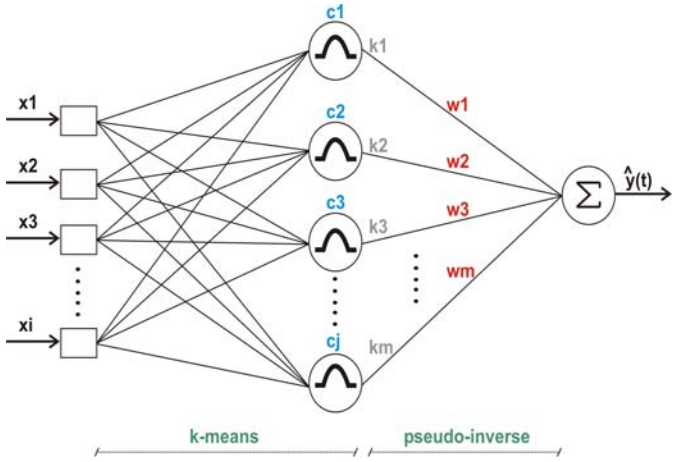

Figure 2. General structure of an RBF-NN

The clustering method used in this paper for determining RBF-NN centers is *k-means*. The implementation follows these steps:

Step 1: Initialize functions centers.
Adjust functions initial centers to the first training data.
Step 2: Group all data within each function center.
Each input data (x_i) belongs to a cluster j^*, where the following holds:

$$\| x_i - c_{j^*} \| = \min_j \| x_i - c_j \| \quad (3)$$

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

$$c_j = \frac{1}{m_j} \sum_{x_i \in j} x_i \quad (4)$$

where m_j is the number of data samples of cluster *j*.
Step 4: Repeat step 2, until there is no more changes in each cluster.

3.2. Genetic Algorithms and Local Search

Genetic algorithms emulate biological evolutionary theories based on the Natural Selection Theory. This sort of algorithm intends to solve optimization problems (Ribeiro Filho *et al.*, 1994) by manipulating a population of individuals and preserve the ones with good genetic characteristics, which represent the more promising areas of the search space, in the optimization point of view. The basic idea of a 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, every individual is assessed by a fitness parameter, which reflects their capability to solve the problem. A percentage of the more capable individuals are maintained, and the remaining eliminated through the selection method. The individuals maintained by the selection can have their characteristics modified by genetic operators such 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 a GA is summarized in the steps below:

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: Compute genetic operations to generate the new population.

The main feature of GA is that a global search for the solution in the search space is performed, but in order to enhance this feature, a local search technique (Yun, 2006; Oh *et al.*, 2002) was used in this work. The method chosen for the local search uses the simplex search method (Nelder and Mead, 1965), a direct search method that finds the minimum of a scalar function of several variables, starting at an initial estimate and without resorting to numerical or analytic gradients. A simplex is the geometrical figure in n dimensions consisting of $n+1$ vertices (in two-dimensional space, it is a triangle; in three- dimensional space it is a pyramid). The simplex algorithm for minimization takes this set of $n+1$ points and attempts to move them into a minimum. The simplex formed should be non-degenerate, i.e. 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 whether the current solution is optimal – if so, stop.
Step 3: Determine the non-basic variable that should enter in the base.
Step 4: Determine the basic variable that is about to leave the base.
Step 5: Find a new basic feasible solution and return to step 2.

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 approach.

3.3 Other Aspects

The pseudo-inverse is the linear optimization method selected to make RBF-NN parameters linear in this work. The update of each weight for training the RBF-NN using this variation of the least mean squares method is done by Eq. (5).

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

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 in Eq. (6), between the actual 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, it 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 Tab. 1, the pH neutralization 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.

We performed ten simulations with different numbers of centers and search methods (with and without local search). Table 1 contains five simulations using two delayed inputs (*Nu*), two delayed outputs (*Ny*) and the number of centers was simulated using 2, 3, 5, 8 and 10. Table 3 contains five simulations using $Nu = 2$, $Ny = 2$ and the number of centers was simulated using 2, 3, 5, 8 and 10. The results obtained for these simulations are the R^2 *(est)* and R^2 *(val)*, corresponding to the estimation and validation phases, respectively.

In Table 1, the best result was found in the fifth simulation – Fig. 3 illustrates the real and estimated output plots of the pH neutralization system. The RBF centers of the best simulation (simulation 5) are shown in Table 2. In Table 3, the best result was found in the tenth simulation – Fig. 5 illustrates the real and estimated output plots and Table 4 the RBF centers. Figs. 4 and 6 present the sample percentage error for the best solutions using GA without local search and GA with local search, respectively.

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

Table 1. Experimental results with different numbers of centers, two delayed inputs and two delayed outputs using an 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. RBF 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

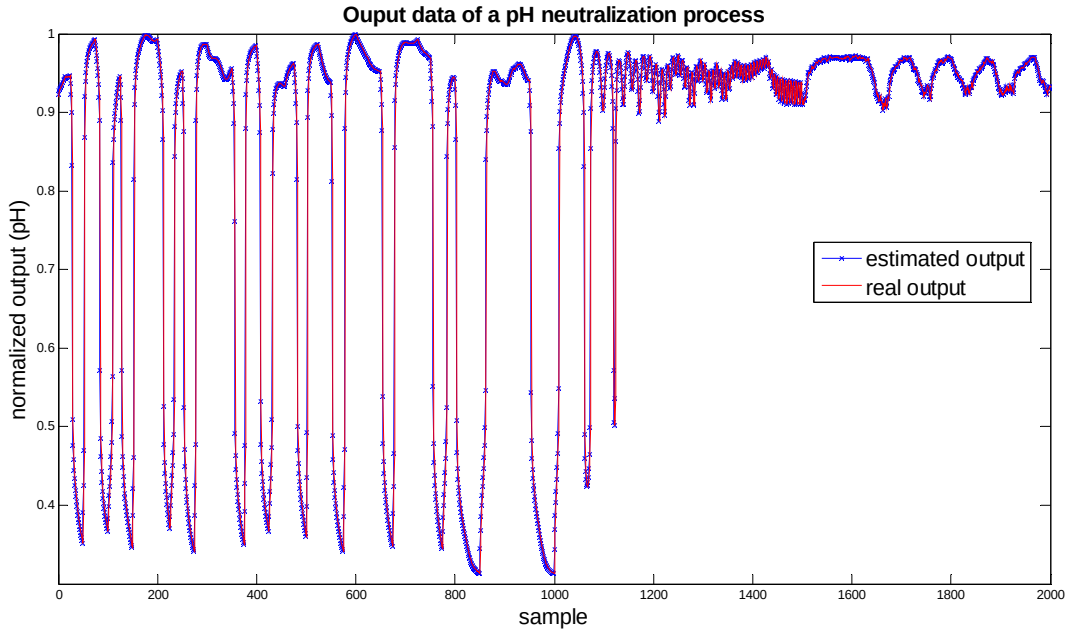

Figure 3. Output data of pH neutralization system with estimated and actual output (simulation 5)

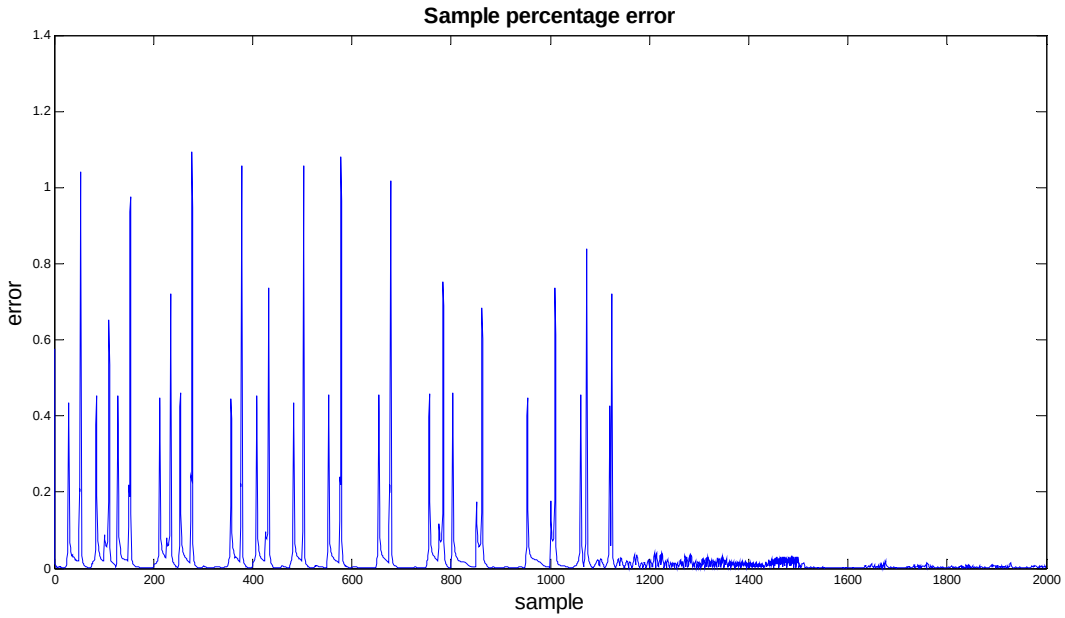

Figure 4. Percentage Error (simulation 5)

Table 3. Experimental results with different numbers of centers, two delayed inputs and two delayed outputs using an 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

Table 4. RBF 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

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

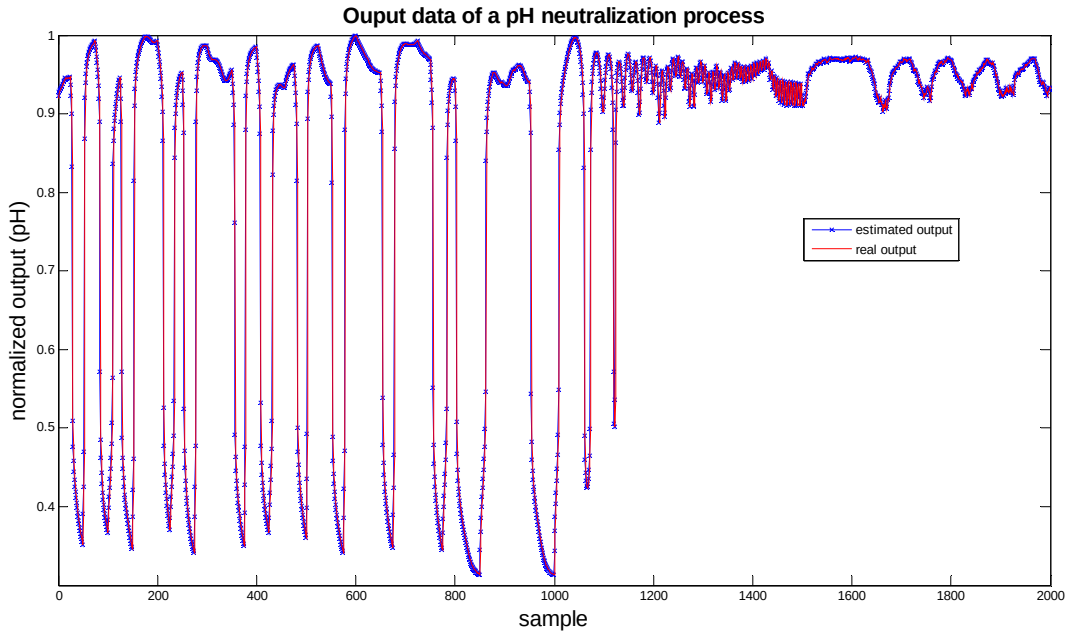

Figure 5. Output data of pH neutralization system with estimated and actual output (simulation 10)

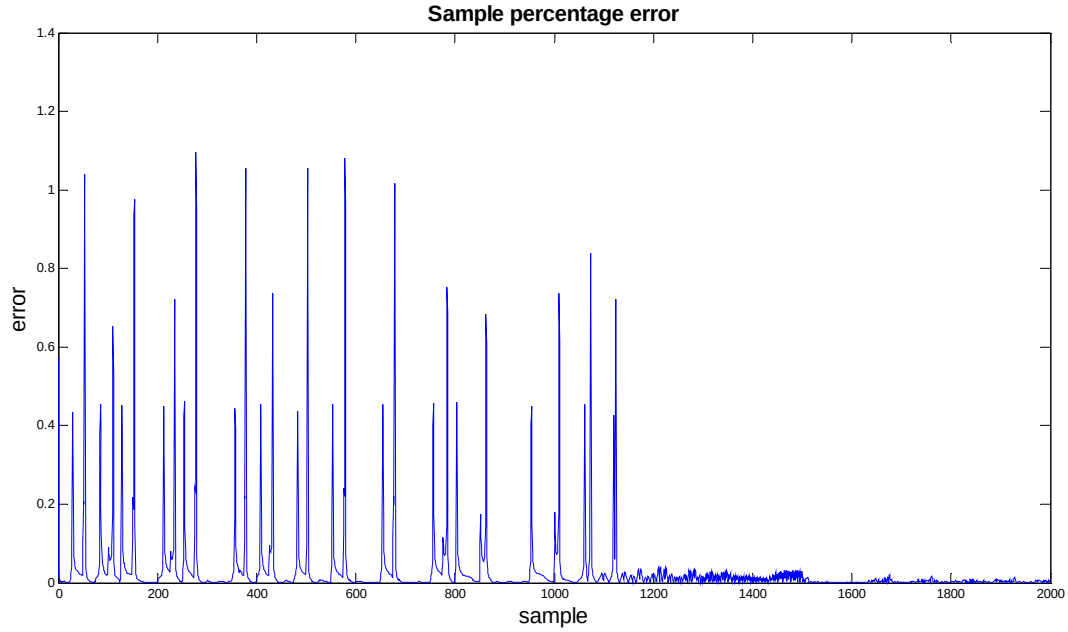

Figure 6. Percentage Error (simulation 10)

The comparison between simulation 5 in Table 1 and simulation 3 in Table 10 shows that the introduction of a local search stage provides a small improvement in the assessment of the validation via the R^2 statistic. The use of local search is associated with the nature of the problem, and in some cases may not express a significant improvement in the process of optimization as a whole, since a global search can provide solutions very close to the optimal solution for the assessed search space.

5. CONCLUSIONS 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 were obtained by the *k-means* clustering method, the weights by the pseudo-inverse linear optimization and the spreads of the Gaussian functions by the GA. The performance of the GA was compared with and without local search.

The results obtained show that the application of the proposed method is viable regarding the R^2 statistic, but some aspects still must be considered in the process. First, the *k-means* clustering method is relatively simple and allows establishing initial conditions that can improve performance, but a disadvantage of this method is the need to establish *a priori* the number of groups (centers) that the algorithm will use. Regarding the GA, it is considered to be very versatile, mainly because it offers the possibility of hybridization with other methods, like local search. Local search is responsible for an improvement on performance of the GA, whose main characteristic is to perform global searches.

For future research, we envisage the development of a clustering method that is able to obtain the number of RBF centers automatically. Also other optimization algorithms can be applied, such as Ant Colony Optimization.

Finally, it is important to mention that the results obtained depend heavily on the researcher's choices for parameters, since N_u and N_y values are selected *a priori* and are associated with personal experience – these values reflect directly on the resulting system's complexity.

6. REFERENCES

Aguirre, L. A., 2000, "Introdução à Identificação de Sistemas: Técnicas Lineares e Não-lineares Aplicadas a Sistemas Reais", 2nd ed., Editora UFMG, Belo Horizonte.

Chen, S.; Billings, S. A.; Cowan, C. F. N.; Grant, P. M., 1990, "Practical identification of NARMAX models using radial basis functions", International Journal of Control, Vol. 52, No. 6, pp. 1327-1350.

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

Haykin, S., 2000, "Redes Neurais: Princípios e Prática", Bookman, Porto Alegre, Brazil.

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", Neural Networks Proceedings, IEEE World Congress on Computational Intelligence, Vol. 2, pp. 1311-1314.

McLoone, S.; Brown, M. D.; Irwin, G. W. and Lightbody, G., 1998, "A hybrid linear/non-linear 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.

Nelder, J. A.; Mead, R., 1965, "A simplex method for function minimization", Computer Journal, Vol. 7, No. 4, pp. 308-313.

Oh, I.-S.; Lee, J.-S.; Moon, B.-R., 2002, "Local search-embedded genetic algorithms for feature selection", Proceedings of the 16th International Conference on Pattern Recognition, Vol. 2.

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, pp. 3680-3684.

Ribeiro Filho, J. L.; Treleaven, P. C.; Alippi, C., 1994, "Genetic-algorithm programming environments", Computer, Vol. 27, No. 6, pp. 28-43.

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.

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", Computers and Industrial Engineering, Vol. 51 , No. 1, pp. 128–141.

Zhang, L., Bai, 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., Liu, W., 2004, "Genetic algorithm based RBF neural network for voice conversion", Proceedings of the 5th World Congress on 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.