

A Comparative Study of Nonlinear Time-Varying Process Modeling Techniques: Application to Chemical Reactor

Errachdi Ayachi, Saad Ihsen, Benrejeb Mohamed

LARA Automatique, Tunis, Tunisia.

Email: errachdi_ayachi@yahoo.fr, ihsen.saad@enit.rnu.tn, mohamed.benrejeb@enit.rnu.tn

Received December 27th, 2010; revised May 12th, 2011; accepted May 22nd, 2011

ABSTRACT

This paper proposes the design and a comparative study of two nonlinear systems modeling techniques. These two approaches are developed to address a class of nonlinear systems with time-varying parameter. The first is a Radial Basis Function (RBF) neural networks and the second is a Multi Layer Perceptron (MLP). The MLP model consists of an input layer, an output layer and usually one or more hidden layers. However, training MLP network based on back propagation learning is computationally expensive. In this paper, an RBF network is called. The parameters of the RBF model are optimized by two methods: the Gradient Descent (GD) method and Genetic Algorithms (GA). However, the MLP model is optimized by the Gradient Descent method. The performance of both models are evaluated first by using a numerical simulation and second by handling a chemical process known as the Continuous Stirred Tank Reactor CSTR. It has been shown that in both validation operations the results were successful. The optimized RBF model by Genetic Algorithms gave the best results.

Keywords: Nonlinear Systems; Time-Varying Systems; Multi Layer Perceptron; Radial Basis Function; Gradient Descent; Genetic Algorithms; Optimization

1. Introduction

Neural networks are widely used in the characterization of nonlinear systems [1-7], time-varying time-delay nonlinear systems [8] and they are applied in various applications [9-12].

The system may be with invariant parameters or time-varying parameters. The variation of some system may be such as; system with slow time-varying parametric uncertainties [13,14], with arbitrarily rapid time-varying parameters in a known compact set [15], with rapid time-varying parameters which converge asymptotically to constants [16], and with unknown parameters with arbitrarily fast and nonvanishing variations [17].

Using MLP architecture depends on various parameters, for instance the number of hidden layers, the number of neurons in each hidden layers, the activation function and the learning rate. These parameters present a difficulty to find the suitable architecture of the MLP.

A renewed interest in Radial Basis Function (RBF) neural network has been found in recent years in various application areas such as modeling and control [1-2], pattern recognition [18] identifying malfunctions of dynamical systems in the case of the frequency multiplier [19], and in the case of jump phenomenon [20], determining the optimal choice of machine tools [21], pre-

dicting 2D structure of proteins [22], classification [23, 24], solving systems of equations [25], analysis of the interaction of multi-input multi-output [26] and modeling of robots [27].

Using an RBF leads to a general model structure is less complex than that produced by an MLP network. The computational complexity induced by their learning is less than that induced by learning the MLP networks. The RBF network performance depends, to a choice of activation function [1], the number of hidden neurons and synaptic weights. By the time-varying nature of parameters the RBF methods are not applicable. However, the RBF is well used in invariant-system. The MLP is used in estimation of time-varying time-delay nonlinear system [17]. In this work, we investigate the possibility of extending the well conventional methods to model a nonlinear system in presence of time-varying parameters.

Several methods such as iterative methods [28-32] with the gradient descent method and evolutionary algorithms [32] that genetic algorithms are used in this paper to optimize the structure and determine the parameters of the RBF model. On the other hand, the gradient descent method is used to optimize the MLP network [33].

This paper focuses on the optimization of radial basis functions architecture, and compares it to the MLP ar-

chitecture. The proposed algorithms are applied to time-varying nonlinear systems. The RBF using genetic algorithms gave the best results.

This paper is organized as follows. Nonlinear system modeling by MLP and RBF network is presented at the second and the third section. A comparative study between the MLP and RBF model, applied to two examples of nonlinear systems is presented in the forth section. Conclusions are given in the fifth section.

2. Nonlinear System Modeling by MLP

Modeling a nonlinear system from its input-output can be for several models. Among these models, the NARMA (Nonlinear Auto-Regressive Moving Average) [19] is used; its expression is given by the following equation:

$$\begin{aligned} & y(k+1) \\ &= f(y(k), \dots, y(k-N_y+1), u(k), \dots, u(k-N_u+1)) \end{aligned} \quad (1)$$

where f is a nonlinear mapping, $u(k)$ and $y(k)$ are the input and output vector, N_u and N_y are the maximum input and output lags, respectively. In this paper, the coefficients of the model (1) depend on time.

The used MLP in this paper is to describe the nonlinear system (1). The objective of the modeling is to obtain an MLP that its output follows the output of the system.

2.1. Structure of MLP

The multilayer perceptron network consists of an input layer, an output layer and usually one or more hidden layers. **Figure 1** shows the architecture of MLP network employed for modeling a nonlinear system. It has an input layer of n_0 neurons, L hidden layers, each hidden layer contains n_i neurons and one neuron in the only output layer. The sigmoid activation function, s , is used.

For the nonlinear system (1), if no knowledge about the structure of the nonlinearity of the system is available such system is considered as a “black box” system modeling.

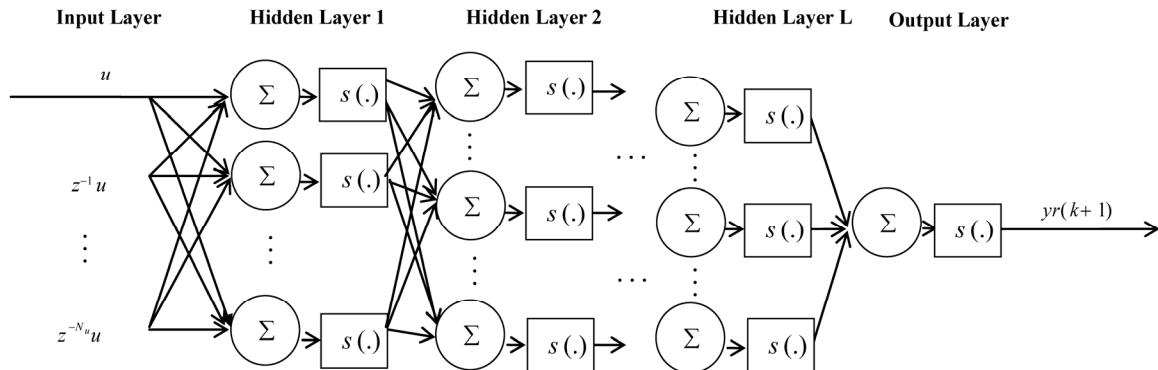


Figure 1. Multilayer perceptron feedforward neural network (MLP).

The output of MLP model $yr(k+1)$ is given by the following equation:

$$yr(k+1) = \lambda s \left(\sum_{l=1}^{n_L} w_{ll}^{(L+1)} \cdots s \left(\sum_{p=1}^{n_2} w_{qp}^{(3)} s \left(\sum_{j=1}^{n_1} w_{pj}^{(2)} s \left(\sum_{i=1}^{n_0} w_{ji}^{(1)} x_i \right) \right) \right) \right) \quad (2)$$

where s is a sigmoid activation function, $yr(k)$ is the output vector of MLP. $w_{ll}^{(L+1)}$, $w_{qp}^{(3)}$, $w_{pj}^{(2)}$ and $w_{ji}^{(1)}$ are the synaptic weights of MLP. x_i is the input vector of MLP.

2.2. Optimization of MLP

Among the optimization methods of MLP, the gradient descent method is used in this paper. Optimization of the MLP is to minimize the mean square error E .

$$E(k) = \frac{1}{2} \sum_{k=0}^N (e(k))^2 = \frac{1}{2} \sum_{k=0}^N [y(k) - yr(k)]^2 \quad (3)$$

where E is a function cost.

In this section, 2 hidden layers are taken into account with a single input layer and one output layer, the result of optimization is given by Equations (4) to (10):

$$\begin{aligned} w_{qp}^{(3)}(k+1) &= w_{qp}^{(3)}(k) \\ &+ \eta \delta^{(3)}(k) s \left(\sum_{j=1}^{n_1} w_{pj}^{(2)}(k) s \left(\sum_{i=1}^{n_0} w_{ji}^{(1)}(k) x_i \right) \right) \end{aligned} \quad (4)$$

$$w_{pj}^{(2)}(k+1) = w_{pj}^{(2)}(k) + \eta \delta^{(2)}(k) s \left(\sum_{i=1}^{n_0} w_{ji}^{(1)}(k) x_i \right) \quad (5)$$

$$w_{ji}^{(1)}(k+1) = w_{ji}^{(1)}(k) + \eta \delta^{(1)}(k) x_i \quad (6)$$

with:

$$\begin{aligned} \delta^{(3)}(k) &= e(k) s' \left(\sum_{p=1}^{n_2} w_{qp}^{(3)}(k) s \sum_{i=1}^{n_0} w_{ji}^{(1)}(k) x_i \right) \\ &\quad \left(\sum_{j=1}^{n_1} w_{pj}^{(2)}(k) s \left(\sum_{i=1}^{n_0} w_{ji}^{(1)}(k) x_i \right) \right) \end{aligned} \quad (7)$$

$$\delta^{(2)}(k) = e(k) s' \left(\sum_{p=1}^{n_1} w_{pj}^{(2)}(k) s \left(\sum_{i=1}^{n_0} w_{ji}^{(1)}(k) x_i \right) \right) \quad (8)$$

$$\delta^{(1)}(k) = e(k) s' \left(\sum_{i=1}^{n_0} w_{ji}^{(1)}(k) x_i \right) \quad (9)$$

$$s'(.) = s(.) (1 - s(.)) \quad (10)$$

3. Nonlinear System Modeling by RBF

As we did with the MLP model, the RBF is used to describe the nonlinear system (1).

3.1. Structure of RBF

The RBF consists of only three layers; an input layer, an output layer and usually one hidden layers contains a hidden radial basis function. The RBF model calculates a linear combination of radial basis functions as is given by the following equation:

$$ym(k+1) = \sum_{j=1}^{n_1} \phi_j v_j \quad (11)$$

where $ym(k)$ is the output vector of RBF. v_j is the synaptic weights of RBF and ϕ_j is a Gaussian activation function:

$$\phi_j = \exp \left[-\frac{(\|x - \mu_j\|)^2}{2\sigma_j^2} \right] \quad (12)$$

3.2. Optimization Methods of RBF

Compared to the MLP, the RBF contains a very small number of parameters. The purpose of optimizing RBF is to determine μ_j , σ_j and v_j by minimizing the function cost E .

$$\begin{aligned} E &= \frac{1}{2} \sum_{k=1}^N (e(k))^2 = \frac{1}{2} \sum_{k=1}^N (y(k) - ym(k))^2 \\ &= \frac{1}{2} \sum_{k=1}^N \left(y(k) - \sum_{j=1}^{n_1} \phi_j v_j \right)^2 \end{aligned} \quad (13)$$

In order to find the minimum of μ_j , σ_j and v_j two strategies are proposed in the literature for finding the minimum of E . The first is based on supervised methods or algorithms using direct time-consuming calculation to determine the minimum of E . The second adopts a hybrid scheme (less costly in computation time) to determine the minimum of E . Solving these problems can be by various methods such as iterative methods (the Gradient Descent method) and evolutionary algorithms (Genetic Algorithm).

3.2.1. Optimization of RBF Using Gradient Descent

The principle of the GD method is applied to optimize the parameters of the RBF model. It uses the rules of delta:

$$\Delta v_j(k) = \eta_1 \frac{\partial E}{\partial v_j} \quad (14)$$

$$\Delta \sigma_j(k) = \eta_2 \frac{\partial E}{\partial \sigma_j} \quad (15)$$

$$\Delta \mu_j(k) = \eta_3 \frac{\partial E}{\partial \mu_j} \quad (16)$$

Hence the calculation of partial derivatives introduced by the following equations:

$$\frac{\partial E}{\partial v_j} = - \left[y(k) - \sum_{j=1}^{n_1} \phi_j v_j \right] \phi_j \quad (17)$$

$$\frac{\partial E}{\partial \sigma_j} = -v_j \frac{\|x - \mu_j\|^2}{\sigma_j^3} \left[y(k) - \sum_{j=1}^{n_1} \phi_j v_j \right] \phi_j \quad (18)$$

$$\frac{\partial E}{\partial \mu_j} = -v_j \frac{x - \mu_j}{\sigma_j^2} \left[y(k) - \sum_{j=1}^{n_1} \phi_j v_j \right] \phi_j \quad (19)$$

finally, we obtain

$$v_j(k+1) = v_j(k) + \eta_1 \frac{\partial E}{\partial v_j} \quad (20)$$

$$\sigma_j(k+1) = \sigma_j(k) + \eta_2 \frac{\partial E}{\partial \sigma_j} \quad (21)$$

$$\mu_j(k+1) = \mu_j(k) + \eta_3 \frac{\partial E}{\partial \mu_j} \quad (22)$$

The learning rate η_i satisfies the following condition:

$$0 \leq \eta_i \leq 1 \quad i = 1, 2, 3 \quad (23)$$

E is nonlinear in the parameters, which calls for finding the minimum, the use of an iterative algorithm that requires an arbitrary initialization of RBF network parameters and a suitable choice of η_i . To maximize the chances of finding the global minimum of E , several initialization parameters and therefore more training is needed, which increases the computing time.

Another method is to optimize separately the parameters of the hidden layer (the centers μ and the widths σ) by genetic algorithms and the synaptic weights between the hidden layer and output layer by the gradient descent method.

3.2.2. Optimization of RBF Using Genetic Algorithms

The genetic algorithm is an optimization algorithm based on techniques derived from genetics and natural evolu-

tion: crossover, mutation, selection. The GA is often used for optimization of RBF [34-38].

In this paper, the GA is used in order to optimize separately the parameters of the hidden layer (the centers μ and the widths σ) of the RBF model.

To find suitable parameters, five elements of GA are called:

- A population is generated randomly. The population size is chosen to achieve a compromise between computation time and solution quality.
- The evaluation of each individual is performed by an evaluation function called fitness function. This function represents the only link between the physical problem and GA. In this paper, the used fitness function is given by the following equation.

$$g = x_1^2 - x_2^2 - \sin(0.9x_1) \quad (24)$$

with x_1 and x_2 are respectively the μ and the σ which are used also in the following equation:

$$E = \frac{1}{2} \sum_{k=1}^N \left(y(k) - \sum_{j=1}^{n_1} \exp \left[-\frac{\|x - \mu_j\|^2}{2\sigma_j^2} \right] v_j \right)^2 \quad (25)$$

- Once the evaluation of generation is realized, it makes a selection from the fitness function. In this paper, the tournament selection is used.
- The crossover operator is designed to enrich the diversity of the population by manipulating the genes of individuals existing in the population. In the other hand, the mutation operator involves the inversion of a bit in a chromosome. The mutation that mathematically guarantees the global optimum can be reached.
- The stopping criterion indicates that the solution is sufficiently approximate the optimum. In this paper, the maximum number of generations is chosen as stopping criterion.

Figure 2 shows the organization of GA to find the minimal parameters of the hidden layer.

The obtained parameters (μ, σ) by GA are used also in the following equation:

$$ym(k+1) = \sum_{j=1}^{n_1} \exp \left[-\frac{\|x - \mu_j\|^2}{2\sigma_j^2} \right] \quad (26)$$

and the synaptic weights are calculated using the gradient descend method:

$$v_j(k+1) = v_j(k) + \eta_l \left[y(k) - \sum_{j=1}^{n_1} \phi_j v_j \right] \phi_j \quad (27)$$

4. Comparative Study of Models

The effectiveness of the suggested methods applied to

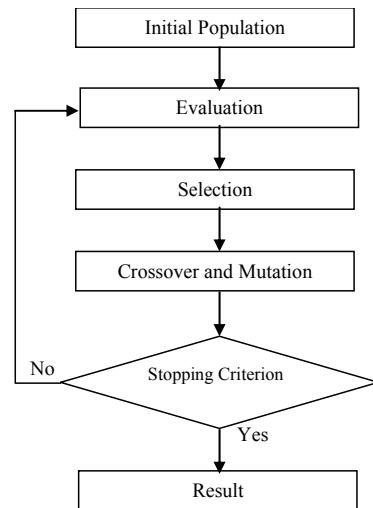


Figure 2. Organization of genetic algorithm.

the identification of behavior of two nonlinear time-varying systems are demonstrated by simulation experiments.

The performance of MLP and RBF models are evaluated by Normalized root Mean Square Error between the system output and the model output, denoted $NMSE$.

$$NMSE_{MLP} = \frac{\sum_{k=1}^N (y(k) - yr(k))^2}{\sum_{k=1}^N (y(k))^2} \quad (28)$$

and

$$NMSE_{RBF} = \frac{\sum_{k=1}^N (y(k) - ym(k))^2}{\sum_{k=1}^N (y(k))^2} \quad (29)$$

4.1. Nonlinear Time Varying-System

We consider the nonlinear time-varying system described by input-output model:

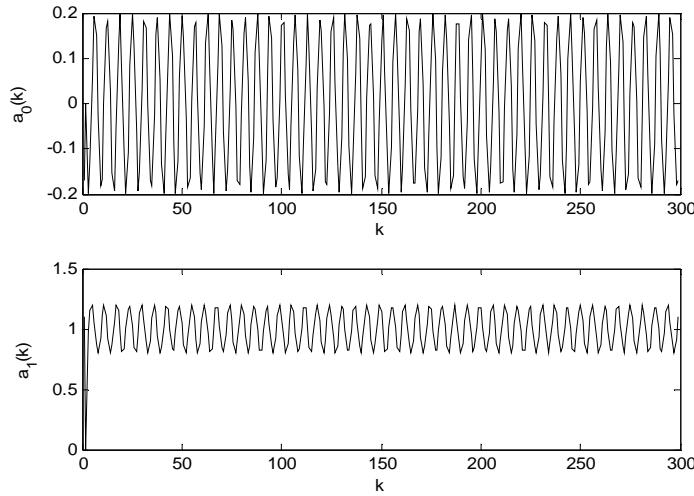
$$\begin{aligned} & y(k+1) \\ &= \frac{y(k) y(k-1) y(k-2) u(k-1)(y(k-2)-1) + u(k)}{1 + a_0(k) y^2(k-1) + a_1(k) y^2(k-2)} \end{aligned} \quad (30)$$

with:

$$\begin{aligned} a_0(k) &= 1 + 0.2 \cos(k) \\ a_1(k) &= 1 - 0.2 \sin(k) \end{aligned} \quad (31)$$

The trajectory of $a_0(k)$ and $a_1(k)$ are given in **Figure 3**.

The input $u(k)$ is sinusoidal signal and it is defined by the following equation:

Figure 3. $a_0(k)$ and $a_1(k)$ trajectories.

$$u(k) = 0.325 \sin(0.95k) \cos(0.55k) \quad (32)$$

In **Figure 4**, the time-varying system responses, the MLP model and the optimized RBF model by the GD method are presented. In this simulation figure, the MLP parameters are $n_0 = 6$, $n_1 = 15$, $n_2 = 20$ and $\eta = 0.44$. The obtained $NMSE_{MLP}$ is $2.78 \times 10^{-3}\%$. However, the RBF parameters are $\eta_1 = 0.34$, $\eta_2 = 0.01$ and $\eta_3 = 0.002$. The obtained $NMSE_{RBF}$ is $3.9812 \times 10^{-3}\%$.

In **Figure 5**, the time-varying system responses, the MLP model and the optimized RBF model by GA are illustrated. In this simulation the same MLP parameters are taken, while the RBF parameters are $\eta_1 = 0.172$, $Pc = 0.7$, $Pm = 0.1$ and $T = 100$. The obtained $NMSE_{RBF}$ is $2.02 \times 10^{-3}\%$.

In these two **Figures (4 and 5)**, it is clear that the responses of MLP and RBF models follow the system response although the variation of parameters.

In one hand, the obtained MLP model is found by several tests of parameters and of learning. The large number of MLP parameter increases the difficulty of its use. However, the simplicity of RBF makes modeling is simple and takes much less training time.

In **Figure 4**, the optimized RBF model by the gradient descent method depends on an expensive time of training and depends on different learning rate ($\eta_1 = 0.34$, $\eta_2 = 0.01$ and $\eta_3 = 0.002$) while, in **Figure 5**, the parameter of RBF model (μ and σ) are finding separately by the GA and the synaptic weights (v) by the DG method, hence the model is faster than the previous.

Noise Effect of Time-Varying Nonlinear System

To validate the quality of the proposed algorithm, an added white noise is used. The influence of the noise of modeling, the Signal Noise Ratio (SNR) is used. The

Figures 6(a)-(c) present the $NMSE$ evolution of different SNR for the both models.

$$SNR = \frac{\sum_{k=1}^N (y(k) - \bar{y})^2}{\sum_{k=1}^N (\zeta(k) - \bar{\zeta})^2} \quad (33)$$

where \bar{y} and $\bar{\zeta}$ are respectively the output average value and noise average value.

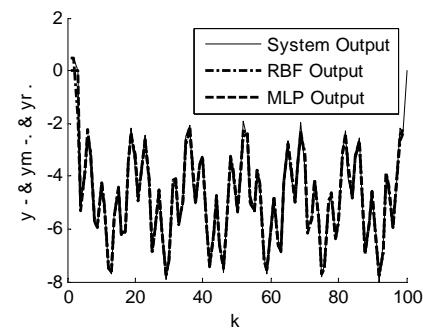


Figure 4. The responses of time-varying system, MLP model and the optimised RBF model by GD.

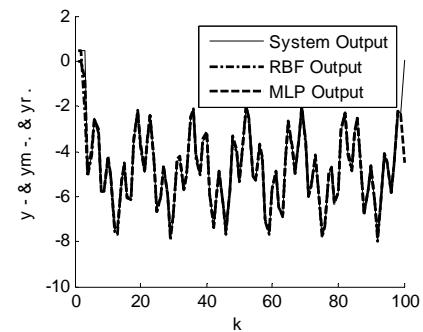


Figure 5. The responses of time-varying system, MLP model and the optimised RBF model by GA.

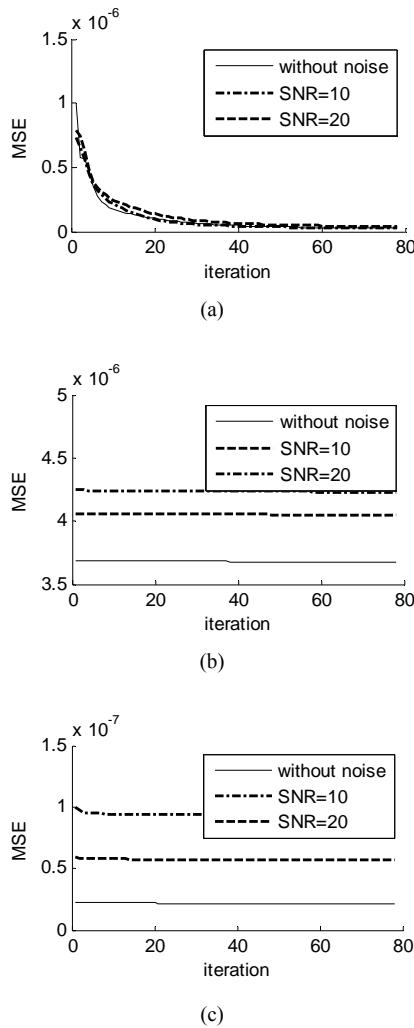


Figure 6. Evolution of MSE of different SNR: (a) MLP; (b) Optimized RBF model with GD method; (c) Optimized RBF model with GA.

In these three **Figures 6(a)-(c)**, we remark firstly the error goes down when the SNR value goes high, then the lowest MSE is obtained when the GA is used (**Figure 6(c)**). Finally, in these all figures we see that the responses of MLP and RBF models follow the time-varying system response despite of the variation of parameters and an added noise.

4.2. Chemical Reactor

To test the effectiveness of the MLP and RBF models we test them on a Continuous Stirred Tank Reactor, CSTR, which is a type of slowly time-varying nonlinear system used for the conduct of the chemical reactions [39-41]. However, the input-output are used in discrete time. A diagram of the reactor is given in the **Figure 6**. The physical equations describing the process are (34) and (35):

$$\frac{dh(t)}{dt} = w_1(t) + w_2(t) - 0.2\sqrt{h(t)} \quad (34)$$

$$\begin{aligned} \frac{dC_b(t)}{dt} = & (C_{b1} - C_b) \frac{w_1(t)}{h(t)} + (C_{b2} - C_b) \frac{w_2(t)}{h(t)} \\ & - \frac{k_1 C_b(t)}{(1+k_2 C_b(t))^2} \end{aligned} \quad (35)$$

where $h(t)$ is the height of the mixture in the reactor, $w_1(t)$ (respectively $w_2(t)$) is the feed of reactant 1 (respectively reactant 2) and C_{b1} (respectively C_{b2}) is the concentration of reactant 1 (respectively reactant 2). w is the feed product of reaction and its concentration is C_b . k_1 , k_2 , w_2 and C_{b2} are consumption reactant rate. They are assumed to be constant. The temperature in the reactor is assumed constant and equal to the ambient temperature. The feed of reactant w_1 and the concentration C_{b1} are the input of the process however C_b represents its output. A diagram of the reactor is given in the **Figure 7**.

For the purpose of the simulations the CSTR model of the reactor provided with Simulink-Matlab is used.

In **Figure 8**, the responses of the chemical reactor, the model that produced by MLP and optimized RBF model by the gradient descent method are presented. In **Figure 9**, the responses of the chemical reactor, the MLP model and the optimized RBF model by genetic algorithms are illustrated.

In **Figures 8 and 9**, the responses of the optimized RBF model by the GD method and MLP model follow the response of chemical reactor. Indeed, in **Figure 8**, the MLP model is carried out with 5 neurons in input layer, 25 neurons in first hidden layer, 22 neurons in hidden layer and the learning rate equal to 0.4. However, the RBF model depends only 3 layers, the second and the only hidden layer contains 11 neurons. The used parameters in the GD method are $\eta_1 = 0.37$, $\eta_2 = 0.003$ and $\eta_3 = 0.02$. In contrary, in **Figure 9**, the parameter of

w_1 : Feed of reactant 1

C_{b1} : Concentration of reactant 1

w_2 : Feed of reactant 2

C_{b2} : Concentration of reactant 2

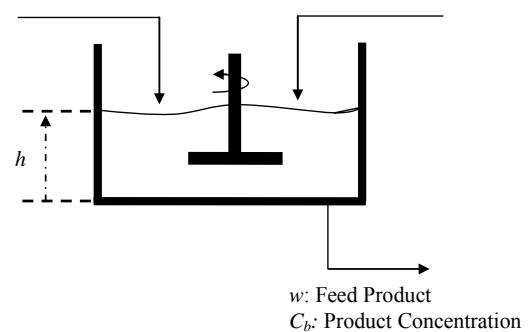


Figure 7. Chemical reactor diagram.

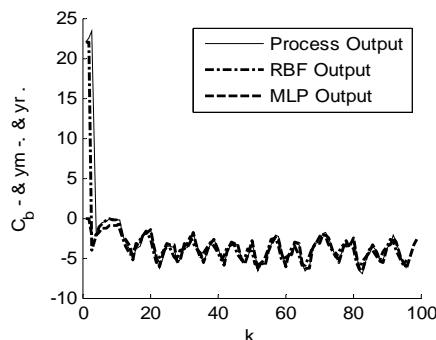


Figure 8. The responses of process, MLP model and the optimized RBF model by GD.

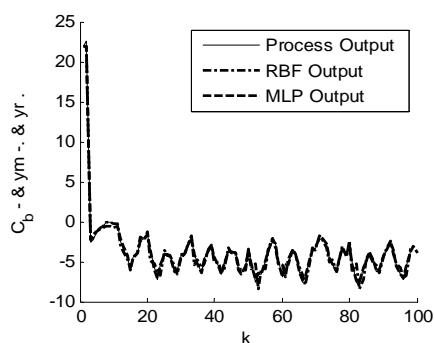


Figure 9. The responses of process, MLP model and the optimized RBF model by GA.

RBF model are optimized separately (μ and σ) using GA and (v) using gradient descent. These parameters are $\eta_1 = 0.11$, $P_c = 0.8$, $P_m = 0.2$ and $T = 100$.

The NMSE is given in the table below:

-	MLP	RBF + GD	RBF + GA
NMSE (%)	1.45e-003	5.97e-002	2.50e-002

From this table, the NMSE computed in the RBF model optimized by the genetic algorithms method is lower than that found by applying the gradient descent method which proves that the evolutionary algorithms give good accuracy for modeling methods of dynamical systems.

5. Conclusion

This paper has dealt with the study and the comparison of two systems modeling techniques the multilayer network model and the radial basis function neural network model. These two approaches are applied in a class of nonlinear system with time-varying parameters. It has been shown that the MLP architecture depends on various parameters and of course a much training time. However, the RBF model depends on the synaptic weights, center and width of its function. In this paper, the RBF

model is optimized by gradient descent method and genetic algorithms. Each optimized RBF models are compared with multilayer perceptron. Mean square error is carried out to evaluate performance of both models and the influence of an additive noise on the identification qualities. These models have been tested for modeling of chemical reactor and results are successful. The RBF model optimized by genetic algorithms showed good performance compared to that optimized by gradient descent method.

REFERENCES

- [1] V. T. S. Elanayar and C. S. Yung, "Radial Basis Function Neural Network for Approximation and Estimation of Nonlinear Stochastic Dynamic," *IEEE Transactions on Neural Networks*, Vol. 5, No. 4, 1994, pp. 594-603. [doi:10.1109/72.298229](https://doi.org/10.1109/72.298229)
- [2] P. Borne, M. Benrejeb and J. Haggege, "Les Réseaux de Neurones. Présentation et Applications," Editions Ophrys, Paris, 2007.
- [3] S. Chabaa, A. Zeroual and J. Antari, "Identification and Prediction of Internet Traffic Using Artificial Neural Networks," *Journal of Intelligent Learning Systems & Applications*, Vol. 2, No. 3, 2010, pp. 147-155.
- [4] A. Errachdi, I. Saad and M. Benrejeb, "On-Line Identification Method Based on Dynamic Neural Network," *International Review of Automatic Control*, Vol. 3, No. 5, 2010, pp. 474-479.
- [5] H. Vijay and D. K. Chaturvedi, "Parameters Estimation of an Electric Fan Using ANN," *Journal of Intelligent Learning Systems & Applications*, Vol. 2, No. 1, 2010, pp. 43-48.
- [6] A. Mishra and Zaheeruddin, "Design of Hybrid Fuzzy Neural Network for Function Approximation," *Journal of Intelligent Learning Systems & Applications*, Vol. 2, No. 2, 2010, pp. 97-109.
- [7] A. Errachdi, I. Saad and M. Benrejeb, "Internal Model Control for Nonlinear Time-Varying System Using Neural Networks," *11th International Conference on Sciences and Techniques of Automatic Control & Computer Engineering*, Monastir, 19-21 December 2010, pp. 1-13.
- [8] Y. Tan, "Time-Varying Time-Delay Estimation for Nonlinear Systems Using Neural Networks," *International Journal of Applied Mathematics and Computer Science*, Vol. 14, No. 1, 2004, pp. 63-68.
- [9] R. Sollacher and H. Gao, "Towards Real-World Applications of Online Learning Spiral Recurrent Neural Networks," *Journal of Intelligent Learning Systems & Applications*, Vol. 1, No. 1, 2009, pp. 1-27.
- [10] S. H. Ling, "A New Neural Network Structure: Node-to-Node-Link Neural Network," *Journal of Intelligent Learning Systems & Applications*, Vol. 2, No. 1, 2010, pp. 1-11.
- [11] J. C. Garcia Infante, J. Jesús Medel Juárez and J. C. Sánchez García, "Evolutive Neural Net Fuzzy Filtering: Basic Description," *Journal of Intelligent Learning Sys-*

- tems & Applications*, Vol. 2, No. 1, 2010, pp. 12-18.
- [12] K. N. Sujatha and K. Vaisakh, "Implementation of Adaptive Neuro Fuzzy Inference System in Speed Control of Induction Motor Drives," *Journal of Intelligent Learning Systems & Applications*, Vol. 2, No. 2, 2010, pp. 110-118.
- [13] R. H. Middleton and G. C. Goodwin, "Adaptive Control of Time-Varying Linear Systems," *IEEE Transactions on Automatic Control*, Vol. 33, No. 2, 1988, pp. 150-155. [doi:10.1109/9.382](https://doi.org/10.1109/9.382)
- [14] F. Giri, M. Saad, J. M. Dion and L. Dugard, "Pole Placement Direct Adaptive Control for Time-Varying Ill-Modeled Plants," *IEEE Transactions on Automatic Control*, Vol. 35, 1990, pp. 723-726. [doi:10.1109/9.53553](https://doi.org/10.1109/9.53553)
- [15] K. S. Tsakalis and P. A. Ioannou, "Linear Time-Varying Systems: Control and Adaptation," Prentice-Hall, Upper Saddle River, 1993.
- [16] R. Marino and P. Tomei, "Adaptive Control of Linear Time-Varying Systems," *Automatica*, Vol. 39, No. 4, 2003, pp. 651-659. [doi:10.1016/S0005-1098\(02\)00287-X](https://doi.org/10.1016/S0005-1098(02)00287-X)
- [17] R.-H. Chi, S.-L. Sui and Z.-S. Hou, "A New Discrete-Time Adaptive ILC for Nonlinear Systems with Time-Varying Parametric Uncertainties," *ACTA Automatica Sinica*, Vol. 34, No. 7, 2008, pp. 805-808. [doi:10.3724/SP.J.1004.2008.00805](https://doi.org/10.3724/SP.J.1004.2008.00805)
- [18] M. R. Berthold, "A Time Delay Radial Basis Function Network for Phoneme Recognition," IEEE Berthold Intel Corporation, Santa Clara, 1994, pp. 4470-4472.
- [19] M. Zarouan, J. Haggège and M. Benrejeb, "Anomalies de Fonctionnement de Systèmes Dynamiques et Modélisation par Réseaux de Neurones des Non-Linearités: Cas de la Démultiplication de Fréquence," *Premières Journées Scientifiques des Jeunes Chercheurs en Génie Électrique et Informatique*, 23-24 March 2001, Sousse-Tunisie, pp. 123-128.
- [20] J. Haggège, M. Zarouan and M. Benrejeb, "Anomalies de Fonctionnement et Modélisation de systèmes Dynamiques par réseaux de Neurones: Cas du Phénomène de Saut," *Premières Journées Scientifiques des Jeunes Chercheurs en Génie Électrique et Informatique*, 23-24 March 2001, Sousse-Tunisie, pp. 134-138.
- [21] S. Allali and M. Benrejeb, "Application des Réseaux de Neurones Multicouches pour le Choix Optimal des Machines Outils," *4 ème Conférence Internationale JTEA*, Hammamet, 12-14 March 2006.
- [22] H. Bouziane, B. Messabih and A. Chouarfia, "Prédiction de la Structure 2D des Protéines par les Réseaux de Neurones," *Communications of the IBIMA*, Vol. 6, 2008, pp. 201-207.
- [23] J. Park and I. Sandberg, "Universal Approximation Using Radial-Basis-Function Networks," *Neural Computation*, Vol. 3, No. 2, 1991, pp. 246-257. [doi:10.1162/neco.1991.3.2.246](https://doi.org/10.1162/neco.1991.3.2.246)
- [24] A. Sifaoui, A. Abdelkrim, S. Alouane and M. Benrejeb, "On new RBF Neural Network Construction Algorithm for Classification," *SIC*, Vol. 18, No. 2, 2009, pp. 103-110.
- [25] A. Golbabai, M. Mammadov and S. Seifollahi, "Solving a System of Nonlinear Integral Equations by an RBF Network," *Computer and Mathematics with Applications*, Vol. 57, No. 10, 2009, pp. 1651-165. [doi:10.1016/j.camwa.2009.03.038](https://doi.org/10.1016/j.camwa.2009.03.038)
- [26] X. Chengying and C. S. Yung, "Interaction Analysis for MIMO Nonlinear Systems Based on a Fuzzy Basis Function Network Model," *Fuzzy Sets and Systems*, Vol. 158, No. 18, 2007, pp. 2013-2025. [doi:10.1016/j.fss.2007.02.012](https://doi.org/10.1016/j.fss.2007.02.012)
- [27] S. S. Chiddarwar and N. R. Babu, "Comparison of RBF and MLP Neural Networks to Solve Inverse Kinematic Problem for 6R Serial Robot by a Fusion Approach," *Engineering Applications of Artificial Intelligence*, Vol. 23, No. 7, 2010, pp. 1-10.
- [28] R. P. Brent, "Fast Training Algorithms for Multilayer Neural Net," *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, 1991, pp. 989-993.
- [29] M. T. Hagan and M. Menhaj, "Training Feedforward Networks with Marquardt Algorithm," *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, 1994, pp. 989-993. [doi:10.1109/72.329697](https://doi.org/10.1109/72.329697)
- [30] R. A. Jacobs, "Increased Rates of Convergence through Learning Rate Adaptation," *Neural Networks*, Vol. 1, No. 4, 1988, pp. 295-307. [doi:10.1016/0893-6080\(88\)90003-2](https://doi.org/10.1016/0893-6080(88)90003-2)
- [31] D. C. Park., M. A. El Sharkawi and R. J. Marks II, "An Adaptively Trained Neural Network," *IEEE Transactions on Neural Networks*, Vol. 2, No. 3, 1991, pp. 334-345. [doi:10.1109/72.97910](https://doi.org/10.1109/72.97910)
- [32] M. Schoenauer and Z. Michalewicz, "Evolutionary Computation," *Control and Cybernetics*, Vol. 26, No. 3, 1997, pp. 307-338.
- [33] A. Errachdi, I. Saad and M. Benrejeb, "Neural Modeling of Multivariable Nonlinear System. Variable Learning Rate Case," *Intelligent Control and Automation*, Vol. 2, No. 3, 2011, pp. 165-175.
- [34] W. L. Cheol and C. S. Yung, "Growing Radial Basis Function Networks Using Genetic Algorithm and Orthogonalization," *International Journal of Innovative Computing, Information and Control ICIC International*, Vol. 5, No. 11A, 2009, pp. 3933-3948.
- [35] G. Lin, H. De-Shuang and Z. Wenbo, "Combining Genetic Optimization with Hybrid Learning Algorithm for Radial Basis Function Neural Networks," *Electronic Letters*, Vol. 39, No. 22, 2003, pp. 1600-1601.
- [36] T. Renato and O. M. J. Luiz, "Selection of Radial Basis Functions via Genetic Algorithms in Pattern Recognition Problems," *10th Brazilian Symposium on Neural Networks*, Salvador, 26-30 October 2008, pp. 71-176.
- [37] M. L. Huang and Y. H. Hung, "Combining Radial Basis Function Neural Network and Genetic Algorithm to Improve HDD Driver IC Chip Scale Package Assembly Yield," *Expert Systems with Applications*, Vol. 34, No. 1, 2008, pp. 588-595. [doi:10.1016/j.eswa.2006.09.030](https://doi.org/10.1016/j.eswa.2006.09.030)
- [38] R. Niranjan and G. Ranjan, "Filter Design Using Radial Basis Function Neural Network and Genetic Algorithm for Improved Operational Health Monitoring," *Applied Soft Computing*, Vol. 6, No. 2, 2006, pp. 154-169. [doi:10.1016/j.asoc.2004.11.002](https://doi.org/10.1016/j.asoc.2004.11.002)

- [39] C. Junghui and H. Tien-Chih, "Applying Neural Networks to On-Line Updated PID Controllers for Nonlinear Process Control," *Journal of Process Control*, Vol. 14, No. 2, 2004, pp. 211-230.
[doi:10.1016/S0959-1524\(03\)00039-8](https://doi.org/10.1016/S0959-1524(03)00039-8)
- [40] J. Van Grop, J. Schoukens and R. Pintelon, "Learning Neural Networks with Noise Inputs Using the Errors in Variables Approach," *IEEE Transactions on Neural Networks*, Vol. 11, No. 2, 2000, pp. 402-414.
[doi:10.1109/72.839010](https://doi.org/10.1109/72.839010)
- [41] H. Demuth, M. Beale and M. Hagan, "Neural Network Toolbox 5," User's Guide, The MathWorks, Natick, 2007.

Nomenclature

$y(k)$: process output,
 $u(k)$: process input,
 f : unknown function,
 N_y : output delay,
 N_u : input delay, $N_u \leq N_y$,
 $yr(k)$: output of MLP,
 x : input vector of (MLP or RBF),
 L : number of hidden layer,
 n_0 : number of nodes of input layer,
 n_i : number of nodes of hidden layer,
 n : number of nodes of output layer,
 w : synaptic weights of MLP,
 s : activation function,

η : learning rate,
 λ : regularization coefficient,
 $ym(k)$: output of RBF,
 ϕ_j : hidden radial basis function,
 μ_j : hidden center,
 σ_j : hidden width,
 v_j : synaptic weights of RBF,
 Pc : crossover probability,
 Pm : mutation probability,
 T : size generation,
 ζ : white noise,
 N : number of observations,
SNR: Signal Noise Ratio.