Intelligent Control and Automation, 2011, 2, 167-175
doi:10.4236/ica.2011.23020 Published Online August 2011 (http://www.SciRP.org/journal/ica)

Neural Modeling of Multivariable Nonlinear Stochastic System. Variable Learning Rate Case

Ayachi Errachdi, Ihsen Saad, Mohamed Benrejeb
Unit of Research LARA Automatic, *le Belvedere*, *Tunisia*
E-mail: *errachdi_ayachi@yahoo.fr*, *ihsen.saad@enit.rnu.tn*, *mohamed.benrejeb@enit.rnu.tn*
Received December 23, 2010; *revised January* 20, 2011; *accepted January* 27, 2011

Abstract

The objective of this paper is to develop a variable learning rate for neural modeling of multivariable nonlinear stochastic system. The corresponding parameter is obtained by gradient descent method optimization. The effectiveness of the suggested algorithm applied to the identification of behavior of two nonlinear stochastic systems is demonstrated by simulation experiments.

Keywords: Neural Networks, Multivariable System, Stochastic, Learning Rate, Modeling

1. Introduction

The Neural Networks (NN) was well used in modeling of nonlinear systems because of its ability of learning, its generalization and its approximation [1-4]. Indeed, this approach provides an effective solution for wide classes of nonlinear systems which are not known or only partial state information is available [5].

Identification is the process of determining the dynamic model of a system from measurements inputs/ outputs [6]. Often, the measured output system is tainted noise. This is due either to the effect of disturbances acting at different parts of the process, either to measurement noise. Therefore these noises may introduce errors in the identification. The stochastic model is a solution to overcome this problem [7]. In this paper, a multivariable nonlinear stochastic system is our interest.

Among the parameters of the NN model, the learning rate (η) has an important role in training phase. In this phase several tests are taken account to find the suitable fixed value. For instance, this parameter can slow down this phase of training [8,9] if it is small. However, if this parameter is large, the training phase is occurring quickly and it becomes unstable [8,9]. To overcome this problem, an adaptive learning rate was asked in [8,9]. This solution is applied in training algorithm of a nonlinear single-variable system [8] and in multivariable nonlinear system [9]. In this paper, a variable learning rate of neural network is developed in order to model a multivariable nonlinear stochastic system. Different cases of signal ratio to noise (SNR) are taken account to show the influence of the noise in identification and the stability of training phase.

This paper is organized as follows. In second section, a multivariable system modeling by neural networks is presented. In third section, the fixed learning rate method is showed. The simulation of the multivariable stochastic systems by NN method using fixed learning rate is detailed in the fourth section. The development of the variable learning rate and results simulations are presented in fifth section. Conclusions are given in sixth section.

2. Multivariable System Modeling by Neural Networks

To find the neural model of such nonlinear systems, some stages must be respected [10]. Firstly the input variables are standardized and centered. Then, the structure of the model is chosen. Finally, the synaptic weights are estimated and the obtained model must be validated. In this context, different algorithms are interested of the synaptic weights estimation. For instance, the gradient descent algorithm [11], the conjugate gradient algorithm [11], the one step secant [11], the Levenberg-Marquardt method [11] and resilient Backpropagation algorithm [11] are developed and confirmed their effectiveness in training. In this paper, the gradient descent algorithm is our interest.

On the basis of the input and output relation of a system, the above nonlinear system can be expressed by a NARMA (Nonlinear Auto-Regressive Moving Average) model [12], that is given by the Equation (1). The architecture of the RNN is presented in **Figure 1**.

$$y_i(k+1) = f_p\left[y_i(k),\cdots,y_i(k-n_2+1),\ u_i(k),\cdots,u_i(k-n_1+1)\right] \quad (1)$$

The output of the l^{th} hidden node is given by the following equation:

$$h_l = \sum_{j=1}^{N_1} p_{lj} v_j,\ l = 1,\cdots,N_2 \quad (2)$$

The i^{th} neural output is given by the following equation:

$$\begin{aligned} o_i(k+1) &= \lambda f\left(\sum_{l=1}^{N_2} f\left(\sum_{j=1}^{N_1} p_{lj} v_j\right) z_{il}\right) \\ &= \lambda f\left(\sum_{l=1}^{N_2} f(h_l) z_{il}\right),\quad i = 1,\cdots,ns \end{aligned} \quad (3)$$

Finally, the compact form is defined as:

$$\begin{aligned} O(k+1) &= \begin{bmatrix} o_1(k+1) \\ \vdots \\ o_{ns}(k+1) \end{bmatrix} \\ &= \lambda f\left[\begin{bmatrix} f(h_1) & \dots & f(h_{N_2}) \\ \vdots & \ddots & \vdots \\ f(h_1) & \dots & f(h_{N_2}) \end{bmatrix}\begin{bmatrix} z_{11} & \dots & z_{ns1} \\ \vdots & \ddots & \vdots \\ z_{1N_2} & \dots & z_{nsN_2} \end{bmatrix}\right] \\ &= \lambda f\left[Z^T(k) F\left[P(k)v(k)\right]\right] \end{aligned} \quad (4)$$

where

$$\begin{bmatrix} f(h_1) \\ \vdots \\ f(h_{N_2}) \end{bmatrix} = f\left(\begin{bmatrix} p_{11} & \dots & p_{1N_1} \\ \vdots & \ddots & \vdots \\ p_{N_2 1} & \dots & p_{N_2 N_1} \end{bmatrix}\begin{bmatrix} v_1(k) \\ \vdots \\ v_{N_1}(k) \end{bmatrix}\right) = F[Pv]^T$$

The principle of neural modeling of the multivariable stochastic system is showing in **Figure 1**.

To show the influence of disturbances on modeling, a noise signal $b_i(k)$ is added to the output system. Different cases of Signal Noise Ratio (SNR_i) are taken. This (SNR_i) measures the correspondence between the system output and the estimated output, the equation of SNR_i is as follows:

$$SNR_i = \frac{\frac{1}{N}\sum_{k=0}^{N}\left(y_i(k)-\bar{y}_i\right)}{\frac{1}{N}\sum_{k=0}^{N}\left(b_i(k)-\bar{b}_i\right)} \quad (5)$$

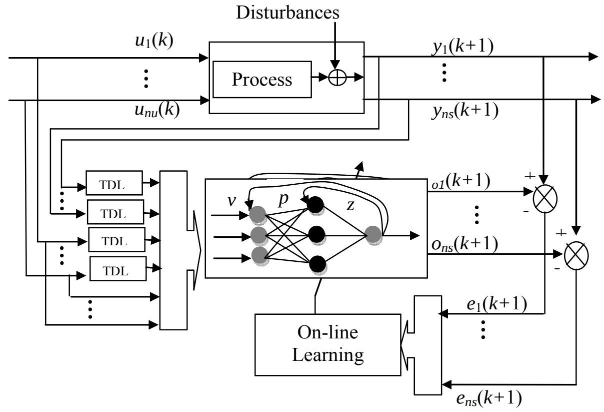

Figure 1. Principle of the neural modeling of the multivariable stochastic system.

The accuracy of correlations relative to the measured values is finding by various statistical means. The criteria exploited in this study were the Relative Error (RE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) [11] given by :

$$RE = E\left[\left\{\left(y_i - o_i\right)^2 / y_i^2\right\}^{1/2}\right] \quad (6)$$

$$MAPE(e_i) = \frac{100}{N}\sum_{k=0}^{N}\left|y_i(k) - o_i(k)\right| \quad (7)$$

3. Fixed Learning Rate Method

The neural system modeling is the research of parameters (weights) model. The search of these weights is the subjects of different works [1-6,8-13]. The gradient descent method is one among different methods which was well applied on neural identification for single-variable system [8] and for multivariable system [9]. In this paper, the same principle is suggested to be applied on neural identification of the multivariable stochastic systems. Indeed, the i^{th} criterion is minimized as follows:

$$J_i(k) = \frac{1}{2}\left(e_i(k)\right)^2 = \frac{1}{2}\left[y_i(k) - o_i(k)\right]^2 \quad (8)$$

By application of the GD method, the theory of [1] is used; we find then [9]:

- For the variation of the synaptic weights of the hidden layer towards the output layer with $(i = 1,\cdots,ns)$.

$$\Delta z_{il} = -\eta_i \frac{\partial J_i(k)}{\partial z_{il}(k)} = \eta_i \frac{\partial o_i(k)}{\partial z_{il}(k)} e_i(k) \quad (9)$$

The compact form (4) is used here, so we find

$$\begin{aligned} \Delta z_{il} &= \lambda \eta_i f'(h_l) \frac{\partial\left(z_{il}^T F(Pv)\right)}{\partial z_{il}(k)} e_i(k) \\ &= \lambda \eta_i f'(h_l) F(Pv) e_i(k) \end{aligned} \quad (10)$$

Finally, the synaptic weights of the hidden layer towards the output layer can be written in the following way:

$$z_{il}(k) = z_{il}(k-1) + \lambda\eta_i f'(h_l) F(Pv) e_i(k) \quad (11)$$

- For the variation of the synaptic weights of the input layer towards the hidden layer.

$$\begin{aligned}\Delta p_{lj} &= -\eta_i \frac{\partial J_i(k)}{\partial p_{lj}(k)} \\ &= \eta_i \frac{\partial o_i(k)}{\partial p_{lj}(k)} e_i(k) \\ &= \lambda\eta_i f'(h_l) \frac{\partial\left(z_{il}^T F(Pv)\right)}{\partial p_{lj}(k)} e_i(k) \\ &= \lambda\eta_i f'(h_l) F'(Pv) z_{il} v^T e_i(k)\end{aligned} \quad (12)$$

Finally, the synaptic weights of input layer towards the hidden layer can be written in the following way:

$$p_{lj}(k) = p_{lj}(k-1) + \lambda\eta_i f'(h_l) F'(Pv) z_{il}(k) e_i(k) \quad (13)$$

In these expressions, η_i is a positive constant value [8,9] which represents the learning rate $(0 < \eta_i \le 1)$ and $F'(Pv)$ represents Jacobian matrix of $F(Pv)$.

$$F'(Pv) = diag\left[f'\left(\sum_{j=1}^{N_1} p_{lj} v_j\right)\right]^T_{l=1,\cdots,N_2} \quad (14)$$

$$f'\left(\sum_{j=1}^{N_1} p_{lj} v_j\right) = \left.\frac{\partial f\left(\sum_{j=1}^{N_1} p_{lj} v_j\right)}{\partial\left(\sum_{j=1}^{N_1} p_{lj} v_j\right)}\right|_{l=1,\cdots,N_2} \quad (15)$$

4. Simulation of Multivariable Nonlinear Stochastic Systems $(SNR = 5)$

In this section, two types of multivariable nonlinear stochastic systems with 2 dimensions $(nu = 2, ns = 2)$ are presented with $(SNR = 5)$. The system (S_1) [8] and (S_2) [14] are defined respectively by the following equations:

$$(S_1)\begin{cases} y_1(k+1) = 0.3y_1(k) + 0.6y_1(k-1) \\ \qquad + 0.6\sin(\pi u_1(k)) \\ \qquad + 0.3\sin(3\pi u_1(k)) \\ \qquad + 0.1\sin(5\pi u_1(k)) + b_1(k) \\ y_2(k+1) = 0.3y_2(k) + 0.6y_2(k-1) \\ \qquad + 0.8\sin(2y_2(k)) \\ \qquad + 1.2u_1(k) + b_2(k) \end{cases} \quad (16)$$

$$(S_2)\begin{cases} y_1(k+1) = \dfrac{0.8y_1^3(k) + u_1^2(k)u_2(k)}{2 + y_2^2(k)} \\ \qquad + b_1(k) \\ y_2(k+1) = \dfrac{y_1(k) - y_1(k)y_2(k) + (u_1(k) + 0.8)}{2 + y_2^2(k)} \\ \qquad + b_2(k) \end{cases} \quad (17)$$

with b_1 and b_2 are a random signals, or u_1 and u_2 are the input signals of the systems considered defined by:

$$u_1(k) = \sin\left(\frac{2\pi k}{250}\right) \quad (18)$$

and

$$u_2(k) = \sin\left(\frac{2\pi k}{25}\right) \quad (19)$$

The input signal u_1 and u_2 are presented in **Figure 2**.

4.1. Simulation Results of System (S1)

A dynamic NN is used to simulate a multivariable nonlinear stochastic system (S1) $(SNR = 5)$. In **Figure 3**, the evolution of the process output and the NN output of the system (S1) is presented. The estimation error between these two outputs is presented in **Figure 4**.

The obtained results, present that for a fixed learning rate $\eta_1 = 0.32$, the NN output o_1 follows the measured output y_1 with an error of prediction $e_1 = 0.0720$ and that o_2 follows the measured output y_2 with an error of prediction $e_2 = 0.0601$ whose learning rate is $\eta_2 = 0.27$.

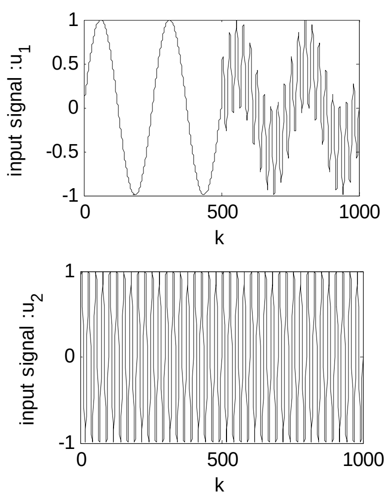

Figure 2. Input signals of the multivariable nonlinear stochastic system.

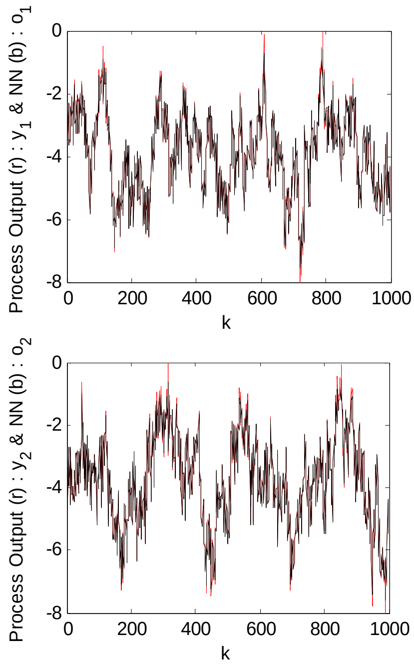

Figure 3. Output of process and NN of system (S1) using a fixed learning rate.

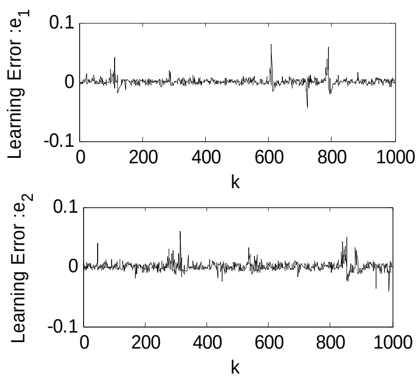

Figure 4. Learning error between the output of process and NN.

If this system has not an added noise, the error of prediction is $e_1 = 0.0384$ and $e_2 = 0.0375$ [9].

4.2. Simulation Results of System (S2)

A dynamic NN is used to simulate a multivariable nonlinear stochastic system (S2) $(SNR = 5)$. In **Figure 5**, the evolution of the process output and the NN output of the system (S2) is presented. The estimation error between these two outputs is presented in **Figure 6**.

The obtained results showing in **Figure 5**, present that for a fixed learning rate $\eta_1 = 0.3$, the NN output o_1 follows the measured output y_1 with an error of prediction $e_1 = 0.0650$ and that o_2 follows the meas- ured output y_2 with an error of prediction $e_2 = 0.0670$

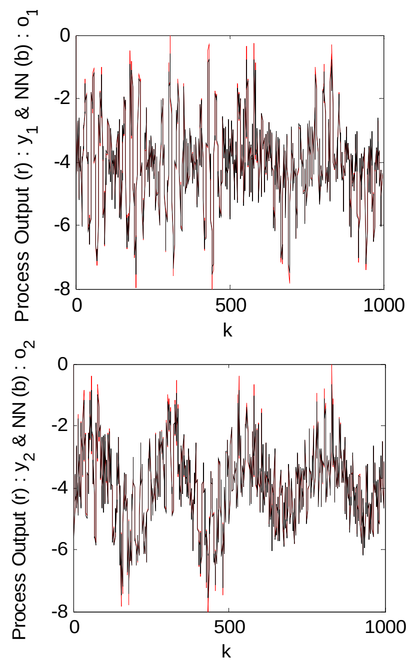

Figure 5. Output of process and NN of system (S2) using fixed learning rate.

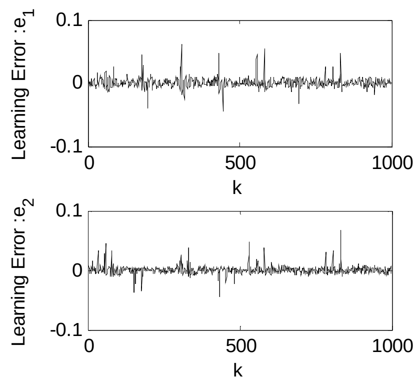

Figure 6. Learning error between the output of process and NN.

whose learning rate is $\eta_2 = 0.25$.

However, if $b_1 = 0$ and $b_2 = 0$, the error of prediction is $e_1 = 0.0531$ and $e_2 = 0.0471$ [9].

Table 1 shows the obtained results of each statistical indicator in the system (S1) and (S2) in the case of fixed learning rate.

Three cases of $\text{SNR}(5, 10 \text{ and } 20)$ are taken to show the influence of disturbances modeling. The obtained results are presented in **Table 2** for the first system and in table 3 for the second system.

In both **Tables 2** and **3**, when the SNR increases the $mse(e_i)$ decrease, it is due under the presence of disturbances in the system.

In this section, the simulation of the two systems (S1 and S2) is carried out using a fixed learning rate. To find

Table 1. Values of different statistical indicators.

SNR = 5%		RE	MAPE
S1	e_1	4.370e – 4	0.0437
	e_2	4.106e – 4	0.0211
S2	e_1	2.616e – 4	0.0262
	e_2	3.542e – 4	0.0354

Table 2. Different cases of SNR.

SNR	5%	10%	20%
$mse(e_1)$	7.611e – 5	6.679e – 5	5.648e – 5
$mse(e_2)$	7.458e – 5	6.643e – 5	4.205e – 5

Table 3. Different cases of SNR.

SNR	5%	10%	20%
$mse(e_1)$	8.698e – 5	7.705e – 5	5.947e – 5
$mse(e_2)$	8.688e – 5	7.562e – 5	5.278e – 5

the suitable learning rate it is necessary to carry out several tests by keeping the condition that $(0 < \eta_i \le 1)$. This research of the learning rate can slow down the phase of training. To cure this disadvantage and in order to accelerate the phase of training, a variable learning rate is used and a fast algorithm will be developed.

5. The Proposed Fast Algorithm

The need for using a variable learning rate is to have a fast training [8-9,15-18]. To answer this condition, the difference of the i^{th} estimation error at $(k+1)$ and at (k) is calculated [8,9].

$$e_i(k+1) - e_i(k) = y_i(k+1) - o_i(k+1) - y_i(k) + o_i(k) \tag{20}$$

We suppose that

$$\Delta y_i(k+1) = y_i(k+1) - y_i(k) \quad \text{and} \quad \Delta o_i(k+1) = o_i(k+1) - o_i(k) \tag{21}$$

by application of [8,9]

$$\left\|\Delta y_i(k+1)\right\| \square \left\|\Delta o_i(k+1)\right\| \tag{22}$$

then the Equation (20) can be

$$\begin{aligned} e_i(k+1) - e_i(k) &\approx -\Delta o_i(k+1) \\ &= \eta_i \frac{\partial o_i(k)}{\partial p_{lj}(k)} e_i(k) = -\lambda \Delta f(h_l) \\ &= -\lambda f'(h_l)\left[F^T(Pv)\Delta z_{il} + z_{il}^T F'(Pv)\Delta P_{lj} v\right] \end{aligned} \tag{23}$$

We introduce (10) and (12),

$$\begin{aligned} &e_i(k+1) - e_i(k) \\ &= -\lambda f'(h_l)\left[F^T(Pv)\lambda\eta_i f'(h_l)F(Pv)e_i(k) \right. \\ &\left. + z_{il}^T F'(Pv)\lambda\eta_i f'(h_l)F'(Pv)z_{il}v^T e_i(k)v\right] \end{aligned} \tag{24}$$

so we find

$$\begin{aligned} e_i(k+1) - e_i(k) &= -\eta_i\lambda^2 f'^2(h_l)\left[F^T(Pv)F(Pv) \right. \\ &\left. + z_{il}^T F'(Pv)F'(Pv)z_{il}v^T v\right]e_i(k) \\ &= -\eta_i \xi_i(k) e_i(k) \end{aligned} \tag{25}$$

with

$$\begin{aligned} \xi_i(k) = \lambda^2 f'^2(h_l)&\left[F^T(Pv)F(Pv) \right. \\ &\left. + z_{il}^T F'(Pv)F'(Pv)z_{il}v^T v\right] \end{aligned} \tag{26}$$

at $(k+1)$ the i^{th} estimation error is

$$e_i(k+1) = \left[1 - \eta_i \xi_i(k)\right]e_i(k) \tag{27}$$

To ensure the convergence of the i^{th} estimation error, *i.e.*, $\lim_{k\to\infty} e_i(k) = 0$, the condition $\left|1 - \eta_i \xi_i(k)\right| < 1$ has to be satisfied [8,9]. This condition implies $0 < \eta_i < 2\xi_i^{-1}(k)$. It is clear that the upper range of the learning rate (η_i) is variable because $\xi_i(k)$ depends on v, z_{il} and p_{lj}. The fastest learning occurs when the learning rate is:

$$\begin{aligned} \eta_i = \xi_i^{-1}(k) = 1\Big/\Big(\lambda^2 f'^2(h_l)&\left[F^T(Pv)F(Pv) \right. \\ &\left. + z_{il}^T F'(Pv)F'(Pv)z_{il}v^T v\right]\Big) \end{aligned} \tag{28}$$

Note that this selection of η_i implies $e_i(k+1) = \left[1 - \eta_i \xi_i(k)\right]e_i(k) = 0$. It's certain that the learning process cannot finish instantly because of the approximation which is caused by the finite sampling time contrary to the theory which is proved that it can be happen if infinitely fast sampling can occur.

Using the obtained variable learning rate η_i, the synaptic weights Δz_{il} and Δp_{lj} will be respectively. it's certain that the learning process cannot finish instantly because of the approximation caused by the finite sampling time contrary to the theorie which proved that it can be happen if infinitely fast sampling can occur it's certain that the learning process cannot finish instantly because of the approximation caused by the finite sampling time contrary to the theorie which proved that it can be happen if infinitely fast sampling can occur it's certain that the learning process cannot finish instantly because of the approximation caused by the finite sampling time contrary to the theorie which proved that it can be happen if infinitely fast sampling can occur it's certain that the learning process cannot finish instantly because of the approximation caused by the finite sampling time contrary to the theorie which proved that it can be happen if infinitely fast sampling can occur.

$$\Delta z_{il} = \eta_i \lambda f'(h_l) F(Pv) e_i(k) = \frac{F(Pv) e_i(k)}{\lambda f'(h_l)\left[F^T(Pv)F(Pv) + z_{il}^T F'(Pv) F'(Pv) z_{il} v^T v\right]} \quad (29)$$

$$\Delta p_{lj} = \eta_i \lambda f'(h_l) F'(Pv) z_{il} v^T e_i(k) = \frac{F'(Pv) z_{il} v^T e_i(k)}{\lambda f'(h_l)\left[F^T(Pv)F(Pv) + z_{il}^T F'(Pv) F'(Pv) z_{il} v^T v\right]} \quad (30)$$

Finally, $z_{il}(k)$ and p_{lj} can be:

$$z_{il}(k) = z_{il}(k-1) + \frac{F(Pv) e_i(k)}{\lambda f'(h_l)\left[F^T(Pv)F(Pv) + z_{il}^T F'(Pv) F'(Pv) z_{il} v^T v\right]} \quad (31)$$

$$p_{lj}(k) = p_{lj}(k-1) + \frac{F'(Pv) z_{il} v^T e_i(k)}{\lambda f'(h_l)\left[F^T(Pv)F(Pv) + z_{il}^T F'(Pv) F'(Pv) z_{il} v^T v\right]} \quad (32)$$

5.1. Simulation Results of System (S1) ($SNR = 5$)

In this section, the obtained variable learning rate (η_1, η_2) are applied. In **Figure 7**, the evolution of the process output and the NN output of the system (S1) is presented. The error estimation between these two outputs is presented in **Figure 8**.

The obtained results present that for a variable learning rate, the neural output o_1 follows the measured output y_1 with an error of prediction $e_1 = 0.0634$ and that o_2 follows the measured output y_2 with an error of prediction $e_2 = 0.0588$. However, if $b_1 = 0$ and $b_2 = 0$, the error of prediction is $e_1 = 0.0175$ and $e_2 = 0.0369$ [9].

5.2. Simulation Results of System (S2) ($SNR = 5$)

The evolution of the process output and the NN output of the system (S2) is presented in **Figure 9**. The error between these two outputs is presented in **Figure 10**. The evolution of the squared error in two cases; fixed and variable learning rates is presented in **Figures 11** and **12**.

The obtained results, concerning system (S2), present that for a variable learning rate, the neural output o_1 follows the measured output y_1 with an error of predicttion $e_1 = 0.0539$ and that o_2 follows the measured output y_2 with an error of prediction $e_2 = 0.0668$.

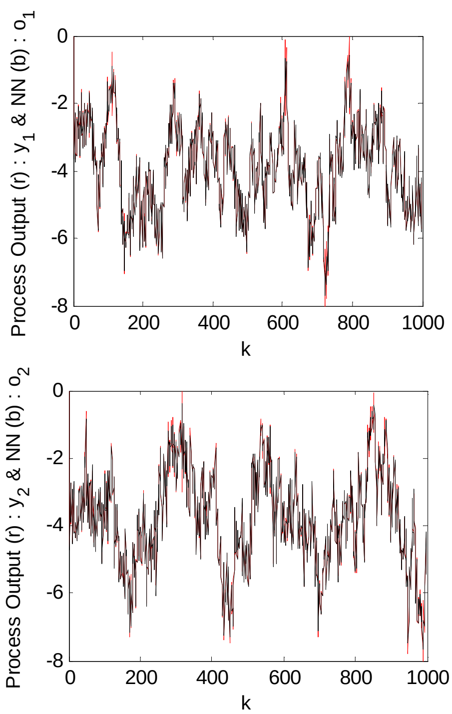

Figure 7. Output of process and NN of system (S1) using a variable learning rate.

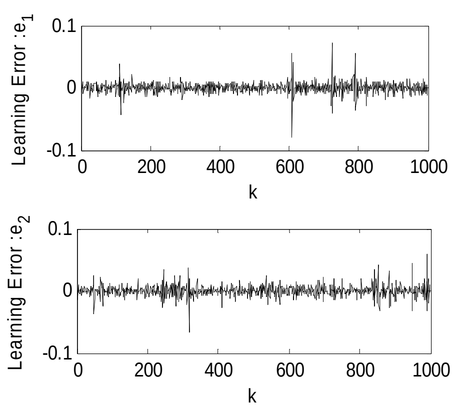

Figure 8. Learning error between the output of process and NN.

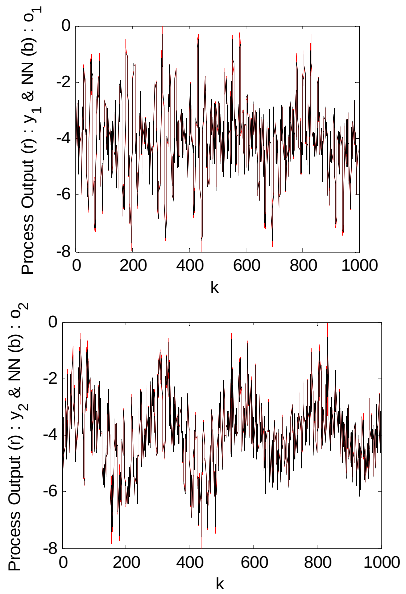

Figure 9. Output of process and NN of system (S2) using a variable learning rate.

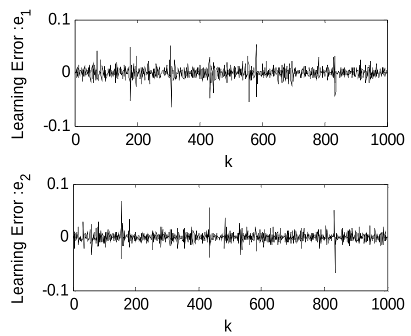

Figure 10. Learning error between the output of process and NN.

However, if $b_1 = 0$ and $b_2 = 0$, the error of prediction is $e_1 = 0.0292$ and $e_2 = 0.0166$ [9].

The obtained results presented in **Figures 11** and **12** showing that, when a variable learning rate is used, the convergence of the squared error is very faster than a fixed learning rate is used.

Table 4 shows the obtained results of each statistical indicator in the system (S1) and (S2) in the case of variable learning rate.

We took three cases of $SNR(5,10 \text{ and } 20)$ to show the influence of disturbances modeling. The obtained results are presented in **Table 5** for the first system and in **Table 6** for the second system. In both tables, when the SNR increases the $mse(e_i)$ decrease, it is due under the presence of disturbances in the system.

The obtained values $mse(e_i)$ in **Tables 5** and **6** are lower compared to $mse(e_i)$ which are calculated in **Tables 2** and **3**, that explains the variable rate adjusts with

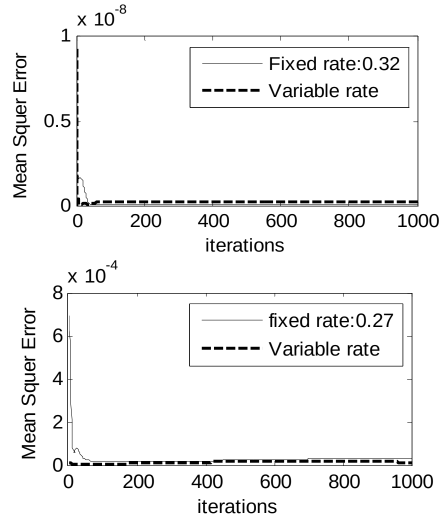

Figure 11. Evolution of the mean squared error of (S1).

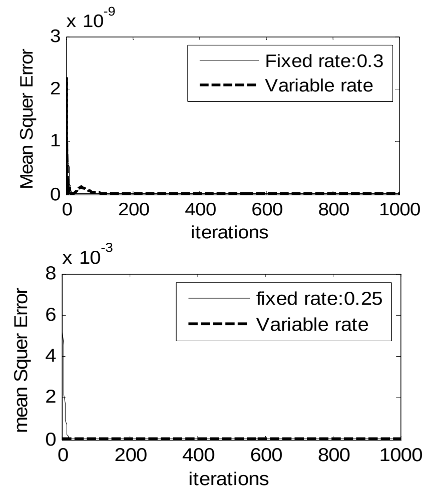

Figure 12. Evolution of the mean squared error of (S2).

Table 4. Values of different statistical indicators.

SNR = 5%		RE	MAPE
S1	e_1	3.256e – 4	0.0326
	e_2	3.793e – 4	0.0379
S2	e_1	6.453e – 4	0.0645
	e_2	6.236e – 4	0.0624

Table 5. Different cases of SNR.

SNR	5%	10%	20%
$mse(e_1)$	5.906e – 5	5.152e – 5	3.932e – 5
$mse(e_2)$	6.501e – 5	5.552e – 5	4.310e – 5

Table 6. Different cases of SNR.

SNR	5%	10%	20%
$mse(e_1)$	7.402e – 5	6.368e – 5	6.334e – 5
$mse(e_2)$	7.863e – 5	4.601e – 5	4.537e – 5

changes in examples.

6. Conclusions

In this paper, a variable learning rate for neural modeling of multivariable nonlinear stochastic system is suggested. This parameter can slow down the training phase when it is chosen as small, and can be unstable when it is chosen as large. To avoid this step, a variable learning rate method is developed and it is applied in identification of nonlinear stochastic system. The advantages of the proposed algorithm are firstly the simplicity to apply it in a multi-input multi-output nonlinear system. Secondly, the gain of the training time is remarked and the result quality is noticed. Besides, this algorithm is a manner to avoid the search for such fixed training rate which presents a disadvantage at the level the phase of training. In contrary, the variable learning rate algorithm does not require any experimentation for the selection of an appropriate value of the learning rate. The proposed algorithm can be applied in real time process modeling. Different cases of SNR are discussed to test the developed method and it showed that the obtained results using a variable learning rate is very satisfy than when the fixed learning rate was used.

7. References

[1] K. Kara, "Application des Réseaux de Neurones à l'identification des Systèmes Non Linéaire," Thesis, Constantine University, 1995.

[2] S. R. Chu, R. Shoureshi and N. Tenorio "Neural Networks for System Identification," *IEEE Control System Magazine*, Vol. 10, No. 3, 1990, pp. 31-35.

[3] S. Chen and S. A. Billings, "Neural Networks for Nonlinear System Modeling and Identification," *Inernational Journal of Control*, Vol. 56, No. 2, 1992, pp. 319-346. doi:10.1080/00207179208934317

[4] N. N. Karabutov, "Structures, Fields and Methods of Iden- tification of Nonlinear Static Systems in the Conditions of Uncertainty," *Intelligent Control and Automation* (*ICA*), Vol. 1, No. 1, 2010, pp. 1-59.

[5] D. C. Psichogios and L. H. Ungar, "Direct and Indirect Model-Based Control Using Artificial Neural Networks," *Industrial and Engineering Chemistry Research*, Vol. 30, No. 12, 1991, pp. 25-64. doi:10.1021/ie00060a009

[6] A. Errachdi, I. Saad and M. Benrejeb, "On-Line Identifycation Method Based on Dynamic Neural Network," *International Review of Automatic Control*, Vol. 3, No. 5, 2010, pp. 474-479.

[7] A. M. Subramaniam, A. Manju and M. J. Nigam, "A Novel Stochastic Algorithm Using Pythagorean Means for Minimization," *Intelligent Control and Automation*, Vol. 1, No. 1, 2010, pp. 82-89. doi:10.4236/ica.2010.12009

[8] D. Sha and B. Bajic, "On-Line Adaptive Learning Rate BP Algorithm for MLP and Application to an Identification Problem," *Journal of Applied Computer Science,* Vol. 7, No. 2, 1999, pp. 67-82.

[9] A. Errachdi, I. Saad and M. Benrejeb, "Neural Modelling of Multivariable Nonlinear System. Variable Learning Rate Case," 18*th Mediterranean Conference on Control and Automation*, Marrakech, 2010, pp. 557-562.

[10] P. Borne, M. Benrejeb and J. Haggege, "Les Réseaux de Neurones. Présentation et Application," Editions Technip, Paris, 2007.

[11] S. Chabaa, A. Zeroual and J. Antari, "Identification and Prediction of Internet Traffic Using Artificial Neural Net-Works," *Journal of Intelligent Learning Systems & Applications*, Vol. 2, No. 1, 2010, pp. 147-155.

[12] M. Korenberg, S. A. Billings, Y. P. Liu and P. J. Mcllroy, "Orthogonal Parameter Estimation Algorithm for Nonlinear Stochastic Systems," *International Journal of Control*, Vol. 48, No. 1, 1988, pp. 346-354. doi:10.1080/00207178808906169

[13] 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*, Anaheim, 2010, pp. 1-13.

[14] D. Sha, "A New Neural Networks Based Adaptive Model Predictive Control for Unknown Multiple Variable No-Linear systems," *International Journal of Advanced Mechatronic Systems*, Vol. 1, No. 2, 2008, pp. 146-155. doi:10.1504/IJAMECHS.2008.022013

[15] R. P. Brent, "Fast Training Algorithms for Multilayer Neural Nets," *IEEE Transactions on Neural Networks*, Vol. 2, No. 3, 1991, pp. 346-354. doi:10.1109/72.97911

[16] R. A. Jacobs, "Increase Rates of Convergence through Learning Rate Adaptation," *IEEE Transactions on Neural Networks*, Vol. 1, No. 4, 1988, pp. 295-307.

[17] D. C. Park, M. A. El-Sharkawi and R. J. Marks, "An Adaptively Trained Neural Network," *IEEE Transactions on Neural Networks*, Vol. 2, No. 3, 1991, pp. 334-345. doi:10.1109/72.97910

[18] P. Saratchandran, "Dynamic Programming Approach to Optimal Weight Selection in Multilayer Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 2, No. 4, 1991, pp. 465-467. doi:10.1109/72.88167

Nomenclature

y_i : vector of process output, its average value $\overline{y}_i$,

u_i : vector of process input,

f_p : unknown function of process,

n_1 : input delay,

n_2 : output delay, $n_1 \le n_2$,

U : input of the process, $U = \left[u_1(k) \cdots u_n(k)\right]^T$,

Y : output of the process, $Y = \left[y_1(k) \cdots y_{ns}(k)\right]^T$,

o_i : vector of RNN output,

O : output of the RNN model, $O = \left[o_1 \cdots o_{ns}\right]^T$,

N_1 : number of nodes of input layer,

N_2 : number of nodes of hidden layer,

p_{lj} : synaptic weights of the input layer towards the hidden layer, $P = \left[p_{lj}\right]$ with $l = 1, \cdots, N_2$ and $j = 1, \cdots, N_1$,

v : input vector of the RNN model,

$$v = \left[v_1 \cdots v_{N_1}\right]$$
$$= \left[u_i(k) \cdots u_i(k - n_1 + 1) y_i(k) , \cdots y_i(k - n_2 + 1)\right]$$

ns : number of nodes of output layer,

z_{il} : synaptic weights of hidden layer towards the output layer, $Z = \left[z_{il}\right]$ with $l = 1, \cdots, N_2$ and $i = 1, \cdots, ns$,

η_i : learning rate, $0 < \eta_i \le 1$,

λ : a scaling coefficient used to expand the range of RNN output, $0 < \lambda \le 1$,

f : activation function, $f(h_l)$ is the output of the l^{th} node,

$e_i(k)$: error between the i^{th} measured process output and the i^{th} measured RNN output,
$e_i(k) = y_i((k) - o_i(k)$,

E : vector of error, $E = \left[e_1(k) \cdots e_{ns}(k)\right]^T$,

N : number of observations,

TDL : Tapped Delay Line block,

h_l : l^{th} output of neuron of hidden layer,

$$F\left[Pv\right] = \left[f(h_1) \cdots f\left(h_{N_2}\right)\right]^T ,$$

$$F'\left[Pv\right] = diag\left[f'(h_1) \cdots f'\left(h_{N_2}\right)\right]^T ,$$

$b_i(k)$: noise of measurement of symmetric terminal δ , $b_i(k) \in \left[-\delta, +\delta\right]$,

$\overline{b_i}$: noise average value.