A Neural Model Reference Adaptive Controller Algorithm for Nonlinear Systems

Sabrine Slama[1], Ayachi Errachdi[1], Mohamed Benrejeb[1]

[1]*Automation Research Laboratory, Tunis El Manar University, ENIT, Department of Electrical Engineering, Le Belvédère B.P. 37, Tunis, 1002, Tunisia*
slema.sabrine@gmail.com

***Abstract*— Model reference adaptive control is a viable control method to impose the desired dynamics on plants whose parameters are affected by large uncertainties. In this paper, it is proposed an algorithm of the Model Reference Adaptive Control (MRAC) for nonlinear systems based-on neural networks. The adaptation mechanism depends on the error between the plant output and the reference model output. In the corresponding control architecture, three neural networks are used as a reference model, a plant model and a controller. The approach is to first construct a plant model to identify the model of the controlled process, and then the controller is adaptively trained on-line to generate the control signal to force the process to follow a desired reference model. For illustration and test purposes the proposed algorithm is applied to the control of a non-linear dynamic system.**

Keywords— Model Reference Adaptive Control (MRAC); adaptation mechanism; back–propagation; nonlinear systems; Neural Network (NN) controller; plant identification

I. Introduction

The technology of neural networks represents an appealing alternative for industrial process control because it can learn nonlinear behavior of the process and act as an universal approximator [1].Their ability to learn nonlinear relationships is widely appreciated and is used in many different types of applications, modeling of dynamic systems, signal processing, and control system design being some of the most common [2].

The most popular approaches of neural control are model reference neural adaptive control, neural predictive control, and feedback linearization neural control [1].

Among various adaptive control methods, the model reference adaptive control is the most widely used and it is also relatively easy to implement. The MRAC have been adopted by many researchers in controlling nonlinear systems as in [17, 18, 21, 22 and 25]. The concept of model reference adaptive control is based on selecting an appropriate reference model and adaptation algorithm [8] that tells how the process output ideally should respond to the command signal [23].The adaptation mechanism of the proposed method is detailed. The control strategy used to define the adaptation law is based on the tracking error between the dynamic system output and target output, which is the response of the reference model. Then, tuning of the weights is based on the steepest descent algorithm to minimize the tracking error. Tuning a controller implies setting its adjustable parameters to appropriate values that provide good control performance [20].

The paper is organized as follows. In Section 2, we introduce the structure of the NN MRAC. The identification of the nonlinear system, the controller network structure and learning process are demonstrated in section 3. Simulation results are discussed in Section 4. Section 5 gives the conclusion of this paper.

II. Structure of the Neural Network MRAC

In [11, 12, 13, 14, and 15], the neural networks have been used for both identification and control of dynamical systems. The structure of the NN MRAC method is presented in Fig. 1. The different blocks of this method are the process, its model network, the controller network and the neural network reference model.

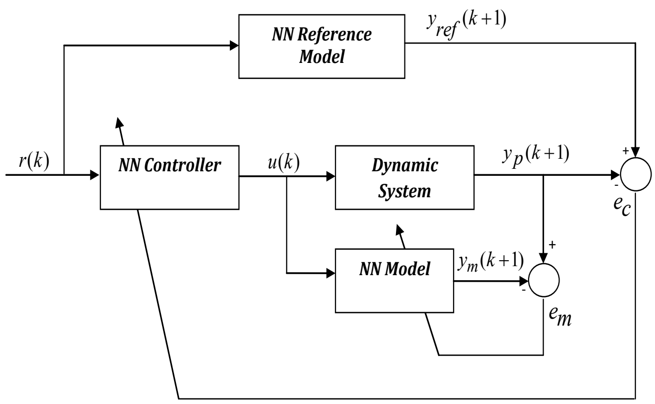

Fig.1. Diagram of NN MRAC structure

where $r(k)$ is the reference input signal, $y_r(k+1)$ is the reference model output, $y_{ref}(k+1)$ is the neural network reference model, $y_p(k+1)$ is the process output, $y_m(k+1)$ is the neural network process model, $u(k)$ is the controller output, $e_c(k)$ is the tracking error and $e_m(k)$ is the identification error. In this structure, the neural network controller provides the appropriate control signal $u(k)$ to the process to maintain the process output $y_p(k+1)$ as close as possible to the desired output $y_{ref}(k+1)$, specified by the neural network reference model. The tracking error $e_c(k)$ between the NN reference model output and the process output is used to adjust the controller parameters. However, the identification error $e_m(k)$

between the NN model and the process output is used to adjust the NN model parameters.

The reference model represents a part of the design in the MRAC strategy. It sets the required performance of the closed loop system because it specifies the rise time, settling time, overshoot and other characteristics. The reference model must be chosen carefully so that the required performance can be achievable by the closed loop system.

III. ALGORITHM DESCRIPTION

A. Modeling using neural networks

The development of models and controllers in real time process control systems is usually based only on input and output data. The process states are often unavailable, and dynamics are introduced to systems using past process input and output information [26-27]. A process sampled at regular time instances can be described by the nonlinear difference equation

$$y_p(k+1)=f_p\left[y_p(k),\dots,y_p(k-n_{y_p}),u(k),\dots,u(k-n_u)\right] \quad (1)$$

with $y_p(k+1)$ and $u(k)$ are respectively the system output and system input, $f_p[.]$ is the process nonlinearity, n_u and n_{yp} are the number of past process input and output respectively.

The neural network used in the nonlinear system identification is a multilayer neural network with Tapped Delay Lines (TDL) of both input and output. The delay elements are used to introduce delayed inputs and outputs that are then fed to a static network as the repressor vector so that the predicted neural network output will follow the target output. The input-output map of NN is shown to match certain types of nonlinear systems. The architecture of the NN is presented in Fig. 2.

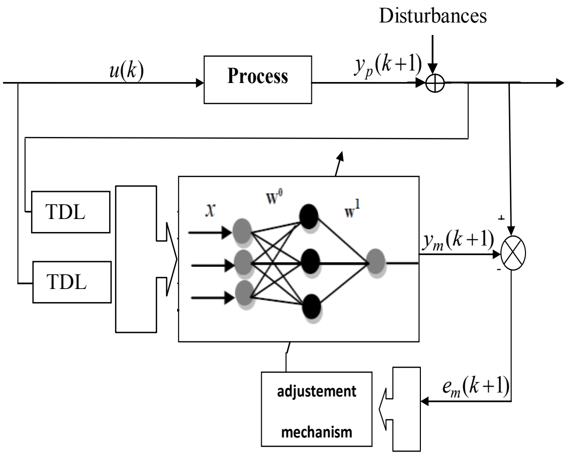

Fig. 2. Principle of the neural network modeling of nonlinear systems

The NN model output is given by the following equation

$$\begin{aligned} y_m(k+1) &= \lambda f_0\left(\sum_{j=1}^{n_2} w_j^1 f_0\left(\sum_{i=1}^{n_1} w_{ij}^0 x_i(k)\right)\right) \\ &= \lambda f_0\left(\sum_{j=1}^{n_2} w_j^1 f_0\left(h_{0j}\right)\right) \end{aligned} \quad (2)$$

or in the compact form

$$y_m(k+1)=\lambda f_0\left[w_j^{1T} F_0(Wx)\right] \quad (3)$$

with $h_{0j}=\sum_{i=1}^{n_1} w_{ij}^0 x_i(k)$, $F_0(Wx)=\left[f_0(h_{0j})\right]_{j=1,\dots,n_2}^T$,

$x=\left[x_i\right]_{i=1,\dots,n_1}^T$, $w^1=\left[w_j^1\right]_{j=1,\dots,n_1}^T$ $w^0=\left[w_{ij}^0\right]_{j=1,\dots,n_1,i=1,\dots,n_2}$,

$F_0'(Wx)$ is the Jacobian matrix of $F_0(Wx)$, its derivative is $F_0'(Wx)=diag\left[f_0'(h_{01}),\dots,f_0'(h_{0n_1})\right]$, n_1 is the neuron number of input layer, n_2 is the neuron number of hidden layer.

Using the gradient descent method, the update of the output weight and the hidden weight of the NN model are respectively

$$w^1(k+1)=w^1(k)+\eta\lambda f_0'(net)F_0(Wx)e_m(k) \quad (4)$$

$$w^0(k+1)=w^0(k)+\eta\lambda f_0'(net)F_0'(Wx)w^1x^Te_m(k) \quad (5)$$

with λ is the regularization coefficient, and η, $0\le\eta\le1$, is the fixed learning rate.

B. The structure of the NN controller

The structure of the NN MRAC presented in the figure 1 is used in this paper. In fact, the output of the neural network controller is given by the expression (6).

$$u(k)=f_r\left[r(k),u(k-1),\dots,u(k-n_f),y_p(k),\dots,y_p(k-n_g)\right] \quad (6)$$

This neural network controller is multi-layer perceptron given by the equation

$$u(k)=\lambda f_1\left(\sum_{j=1}^{n_4} w_{1j}^3 f_1\left(h_{1j}\right)\right) \quad (7)$$

or in the compact form

$$u(k)=\lambda f_1(w^{3T}F_1(W^2x_r))=\lambda f_1(net_2) \quad (8)$$

with $h_{1j}=\sum_{i=1}^{n_3} w_{ij}^2 x_{ri}(k)$, $net_2 = w^{3T} F_1(W^2 x_r)$, $F_1(W^2x_r)=[f_1(h_{1j})]^T_{j=1,\dots,n_4}$, $x_r=[x_{ri}]^T_{i=1,\dots,n_3}$, $w^3=[w_{1j}^3]^T_{j=1,\dots,n_4}$, $w^2=[w_{ij}^2]_{\substack{j=1,\dots,n_4\\ i=1,\dots,n_3}}$, $F_1^{'}(W^2x_r)$ is the Jacobian matrix of $F_1(W^2x_r)$, its derivative is $F_1^{'}(W^2x_r)=diag[f_1^{'}(h_{11}),\dots,f_1^{'}(h_{1n_4})]$,

n_3 is the neuron number of input layer and n_4 is the neuron number of hidden layer.

The update of the weights of the neural network controller is based on the optimizing of the function cost given by:

$$J(k)=\frac{1}{2}(e_c(k))^2=\frac{1}{2}(y_{ref}(k)-y_p(k))^2 \tag{9}$$

Concerning the update of the output weights of the neural network controller which is given by the following equation

$$w^3(k+1)=w^3(k)+\Delta w^3(k)=w^3(k)-\eta_1\frac{\partial J_c(k)}{\partial w^3(k)} \tag{10}$$

with η_1 is the fixed learning rate, $0\le\eta_1\le 1$.

$$\begin{aligned}\frac{\partial J_c(k)}{\partial w^3(k)}&=\frac{\partial\left(\frac{1}{2}e_c(k)^2\right)}{\partial y_p(k)}\frac{\partial y_p(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w^3(k)}\\&=-e_c(k)\frac{\partial y_p(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w^3(k)}\end{aligned} \tag{11}$$

The system Jacobian $\frac{\partial y_p(k)}{\partial u(k)}$ is not available as the true plant parameters are assumed to be unknown. By using the model of the process, we can provide an approximation to the system Jacobian, as $y_p(k)\approx y_m(k)$. Then, the modified back-propagation method can be applied to adjust the neural network controller weights and the expression can be

$$\frac{\partial y_p(k)}{\partial u(k)}\approx\frac{\partial y_m(k)}{\partial u(k)} \tag{12}$$

where $y_m(k)$ is the neural network model of the process given by the equation (2) where $u(k)=x_{r1}(k)$ is the input vector of the neural network controller. The new expression of the neural controller is given as

$$\begin{aligned}y_m(k)&=\lambda f_0\left(\sum_{j=1}^{n_2} w_j^1 f_0\left(\sum_{i=1}^{n_1} w_{ij}^0 u_i(k)\right)\right)\\&=\lambda f_0\left(\sum_{j=1}^{n_2} w_j^1 f_0(h_{0j})\right)\end{aligned} \tag{13}$$

The novel derivate is

$$\frac{\partial y_p(k)}{\partial u(k)}\approx\frac{\partial y_m(k)}{\partial u(k)}=\lambda\sum_{k=1}^{n_2} f_0^{'}(net)\, w_{1k}^1 f_0^{'}(h_{0k})\, w_{k1}^0 \tag{14}$$

In the same, the second dervative is given as below,

$$\frac{\partial u(k)}{\partial w^3(k)}=\lambda_1 f_1^{'}(net_2)\, F_1(W^2x_r) \tag{15}$$

Finally, the update of the output weights of the neural network controller

$$\begin{aligned}&w^3(k+1)=\\&w^3(k)+\eta_1\lambda_1\lambda f_0^{'}(net)\, w^1 F_0^{'T}(W^0x)\, w_1^0 f_1^{'}(net_2)\, F_1(W^2x_r)\, e_c(k)\end{aligned} \tag{16}$$

In the other hand, the update of the hidden weights of the neural network controller is given by the following equation:

$$w^2(k+1)=w^2(k)+\Delta w^2(k)=w^2(k)-\eta_1\frac{\partial J_c(k)}{\partial w^2(k)} \tag{17}$$

with η_1 is the learning rate $0\le\eta_1\le 1$.

$$\begin{aligned}\frac{\partial J_c(k)}{\partial w^2(k)}&=\frac{\partial\left(\frac{1}{2}e_c(k)^2\right)}{\partial y_p(k)}\frac{\partial y_p(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w^2(k)}\\&=-e_c(k)\frac{\partial y_p(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w^2(k)}\end{aligned} \tag{18}$$

The derivative of the second term is given as below,

$$\frac{\partial u(k)}{\partial w^2(k)}=\lambda_1 f_1^{'}(net_2)\, w^3 F_1^{'}(W^2x_r)\, x_r \tag{19}$$

Finally, the update of the hidden weights of the neural network controller

$$\begin{aligned}&w^2(k+1)=w^2(k)+\\&\eta_1\lambda_1\lambda f_0^{'}(net)\, w^1 F_0^{'T}(W^0x)\, w_1^0 f_1^{'}(net_2)\, w^3 F_1^{'}(W^2x_r)\, x_r e_c(k)\end{aligned} \tag{20}$$

The objective of the neural network controller is to find the control law which is given by the equation (7).

The MRAC algorithm was implemented at each time step as summarized below based on the issues previously discussed.

- **Step1**. *Initialize the neural model parameters and the neural network controller parameters,*
- **Step 2**. *Measure the process output* $y_p(k)$
- **Step 3**. *Measure the reference model output* $y_r(k)$
- **Step 4**. *Compute the neural network reference model output* $y_{ref}(k)$.
- **Step 5**. *Compute the control law* $u(k)$ *from the controller input vector.*
- **Step 6**. *Compute the neural network model and test the condition* $\|y_p(k)-y_m(k)\|<\varepsilon_1$ (*where* $\varepsilon_1>0$ *is a given small constant*).
- **Step 7**. *If the condition* $\|y_p(k)-y_m(k)\|<\varepsilon_1$ *is satisfied, test the condition* $\|y_p(k)-y_{ref}(k)\|<\varepsilon_2$ (*where* $\varepsilon_2>0$ *is a given small constant*).
- **Step 8**. *The controller parameters are updated, using the equations (16) and (20), if the condition* $\|y_p(k)-y_{ref}(k)\|<\varepsilon_2$ *is not satisfied.*
- **Step 9.** *Save current and past process information for next sampling instant.*
- **Step 10.** *End.*

The proposed algorithm is applied, in the next section, in tracking nonlinear system output.

IV. SIMULATIONS RESULTS AND DISCUSSION

Consider the nonlinear system described by the input-output model [19].

$$y(k+1)=\frac{y(k)y(k-1)u(k)+u^3(k)+0.5y(k-1)}{1+y^2(k)+y^2(k-1)} \quad (21)$$

with $y(k+1)$ and $u(k)$ are respectively the system output and system input.

The performance of neural network depends on several design parameters such as the type of transfer function the learning rate, the number of input layers, the number of hidden layers and the number of nodes in the hidden layer. In general, identification of these parameters is done by trial and error [24].

In order to testify the efficiency of the proposed model, four performance measures are adopted to evaluate the validity of the model [20], with different neurons number in hidden layer varying from 8 to 30, including root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and Theil's inequality coefficient (TIC).The computational formulas of these performance measures are provided as follows

$$RMSE=\sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\hat{y}(k)-y(k)\right)^2}$$

$$MAE=\frac{1}{N}\sum_{k=1}^{N}\left|\hat{y}(k)-y(k)\right|$$

$$MAPE=\frac{1}{N}\sum_{k=1}^{N}\left|\frac{\hat{y}(k)-y(k)}{y(k)}\right|$$

$$TIC=\frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(\hat{y}(k)-y(k)\right)^2}}{\sqrt{\frac{1}{N}\sum_{k=1}^{N}y(k)^2}+\sqrt{\frac{1}{N}\sum_{k=1}^{N}\hat{y}(k)^2}}$$

Table 1. Variation of performance measures in function of neurons number of hidden layer

Criteria	Number of neurons in the hidden layer					
	8	10	15	20	25	30
RMSE	$1.07\ 10^{-5}$	$1.09\ 10^{-5}$	$1.01\ 10^{-5}$	$1.1\ 10^{-5}$	$1.07\ 10^{-5}$	$1.07\ 10^{-5}$
MAE	$5.04\ 10^{-5}$	$5.08\ 10^{-6}$	$4.65\ 10^{-6}$	$5.15\ 10^{-6}$	$4.92\ 10^{-6}$	$5.02\ 10^{-6}$
MAPE	$3.82\ 10^{-4}$	$1.48\ 10^{-4}$	$9.44\ 10^{-5}$	$2.08\ 10^{-4}$	$1.04\ 10^{-4}$	$1.51\ 10^{-4}$
TIC	$1.07\ 10^{-5}$	$1.08\ 10^{-5}$	$1.01\ 10^{-5}$	$1.1\ 10^{-5}$	$1.07\ 10^{-5}$	$1.07\ 10^{-5}$

The NN model results presented in Table1 were quite satisfactory, the ANN architecture with 15 neurons in hidden layer, sigmoid transfer function in hidden layer and in output layer seems to be the more appropriate for modeling the system.

Fig. 3 presents the neural network model output and the system output. The estimation error $e_m(k)$ is presented by Fig. 4. These figures show that NN identification model can approximate the dynamic system.

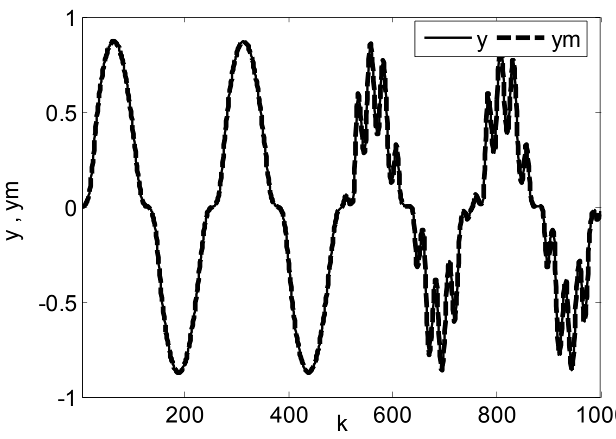

Fig. 3. Responses of the system $y(k)$ and the neural network model $y_m(k)$

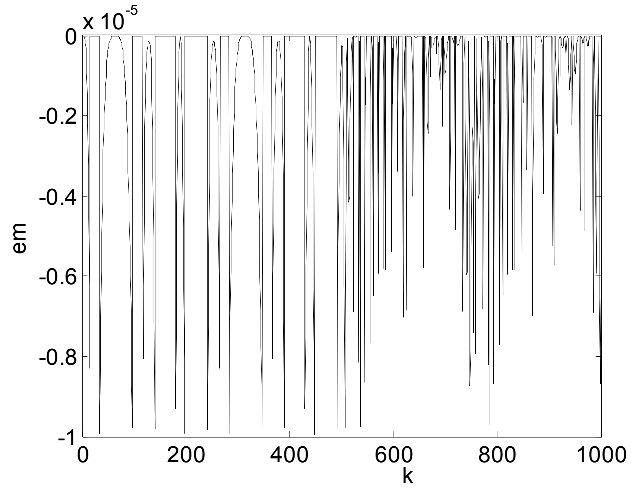

Fig. 4. Evolution of the identification error $e_m(k)$

The reference model transfer function is given by

$$G_m(p) = \frac{K\omega_n^2}{p^2 + 2\omega_n \xi p + \omega_n^2}$$

The MRAC algorithm has been tested over two reference signals inputs $r_1(k)$ and $r_2(k)$. To illustrate, a simulation has been carried out using the following values for the reference model parameters and learning rate, respectively $\omega_n = 3.1623 \text{ rad/s},\ \xi = 0.316,\ \eta_1 = 0.46$.

Fig. 5 presents the neural network reference model $y_{ref}(k)$ and the system output $y(k)$ for a step reference input signal $r_1(k)$ (Case 1). This figure shows that the plant output follows asymptotically the reference model in such a way that the tracking error presented in Fig. 6 converges to zero with the generated control law shown in Fig. 7.

In case 2, a sinusoidal reference input $r_2(k)$ is applied to the NN controller which gives the control law presented in Fig. 11.

$$r_1(k) = \begin{cases} \sin\left(\frac{2\pi k}{250}\right) & k \le 500 \\ \sin\left(\frac{2\pi k}{250}\right) + \sin\left(\frac{2\pi k}{100}\right) & k > 500 \end{cases}$$

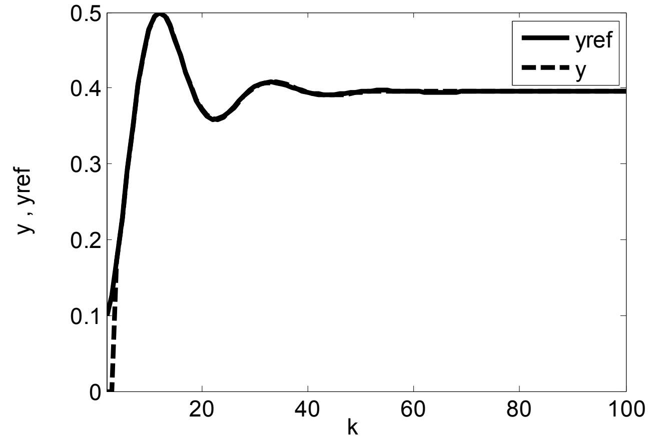

Fig. 5. Responses of the system $y(k)$ and the NN reference model $y_{ref}(k)$ *(case1)*

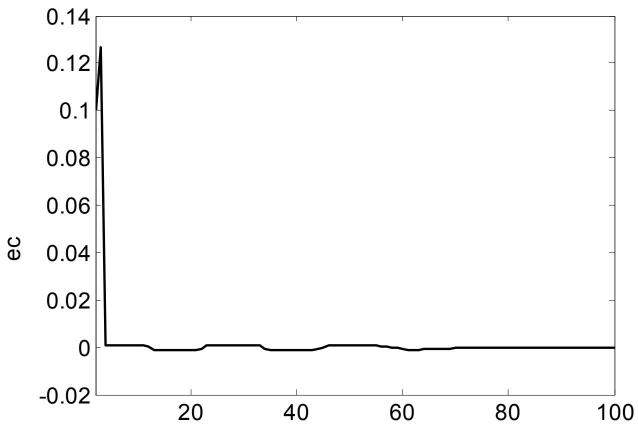

Fig. 6. The tracking error $e_c(k)$

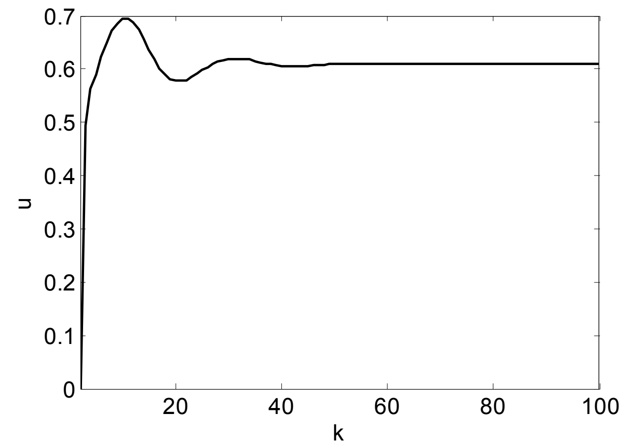

Fig. 7. The control law $u(k)$

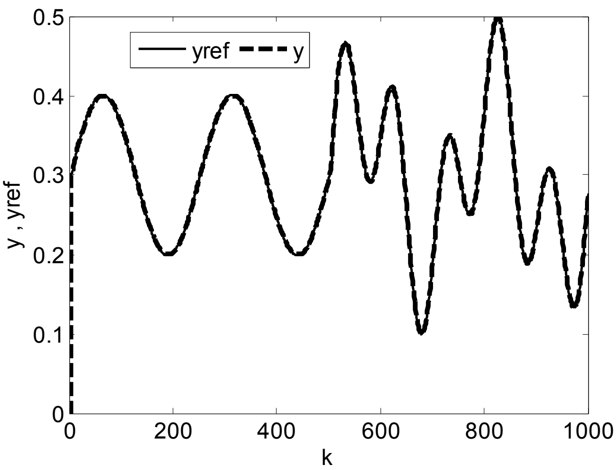

Fig. 8. Responses of the system (y) and the NN reference model (y_{ref}) *(case2)*

We zoom in a portion of graph .Thus we can see clearly in Fig. 9 that the system tract the desired reference model in such a way that the tracking error presented in Fig. 10 converges to zero.

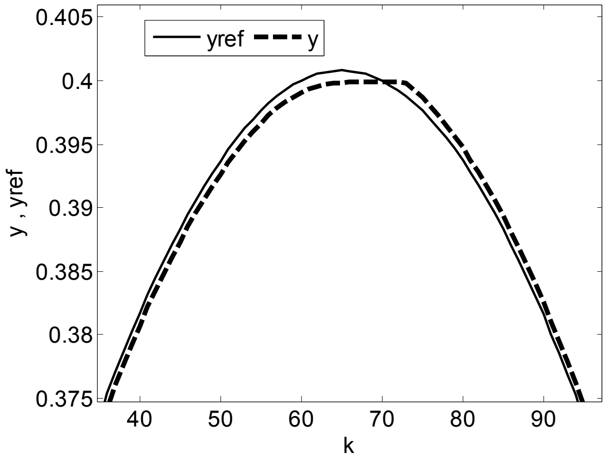

Fig. 9.Portion of graph of figure 5

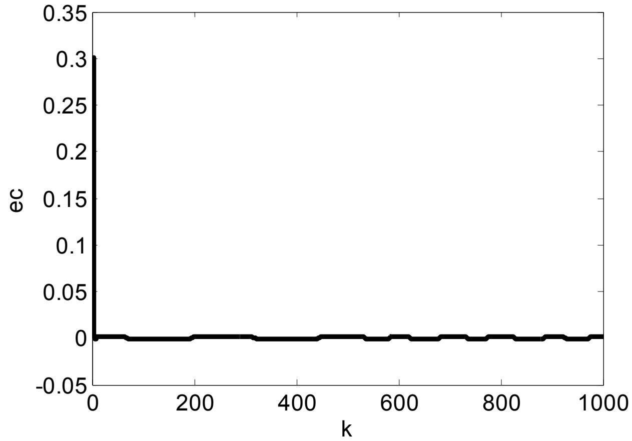

Fig. 10. The tracking error $e_c(k)$

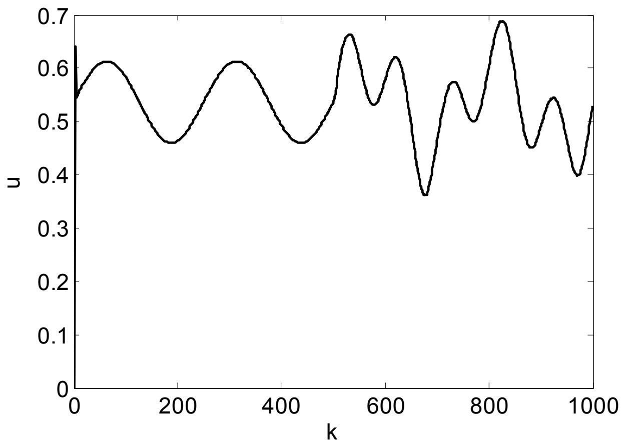

Fig. 11. The control law $u(k)$

A. Effect of disturbances

In this section a noise ξ is added to the output of the plant in order to test the effectiveness of the proposed algorithm.

Fig. 12 presents the neural network reference model $y_{ref}(k)$ and the system output $y_p(k)$. The tracking error $e_c(k)$ is presented in Fig. 13 and the control $u(k)$ is shown in Fig. 14.

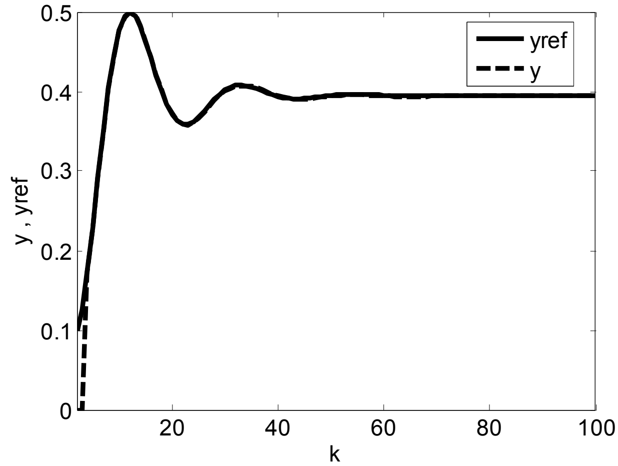

Fig. 12. Responses of the system $y(k)$ and the NN reference model $y_{ref}(k)$ *with noise (case1)*

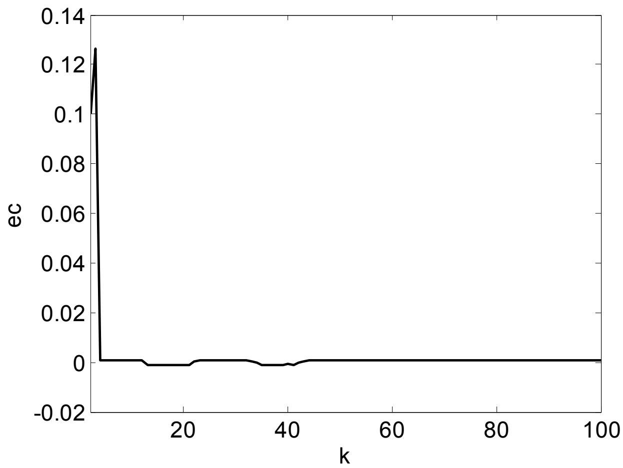

Fig. 13. The tracking error $e_c(k)$ in presence of noise

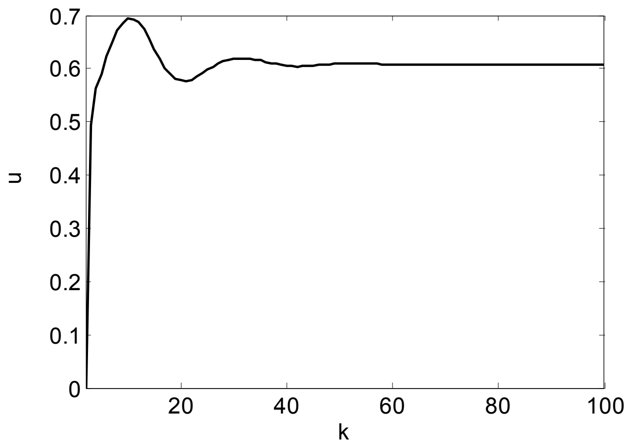

Fig. 14. The control law $u(k)$ in presence of noise

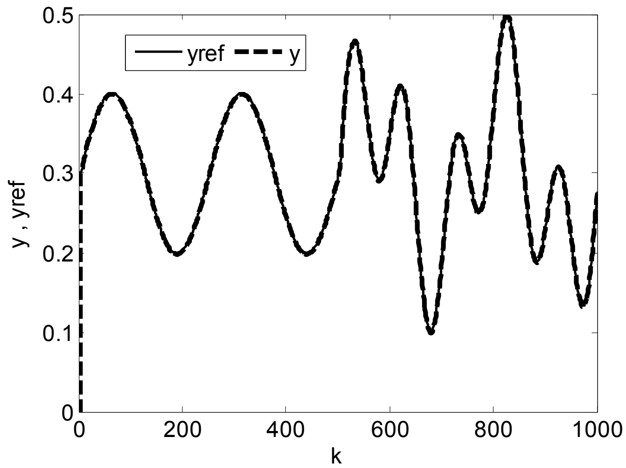

Fig. 15. Responses of the system *y(k)* and the NN reference model $y_{ref}(k)$ *with noise (case 2)*

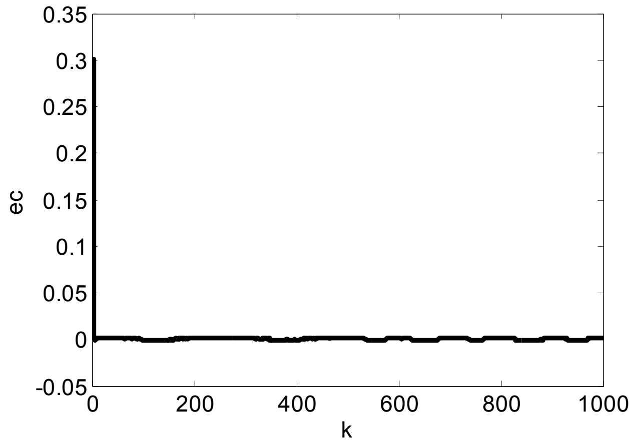

Fig. 16. The tracking error $e_c(k)$ in presence of noise

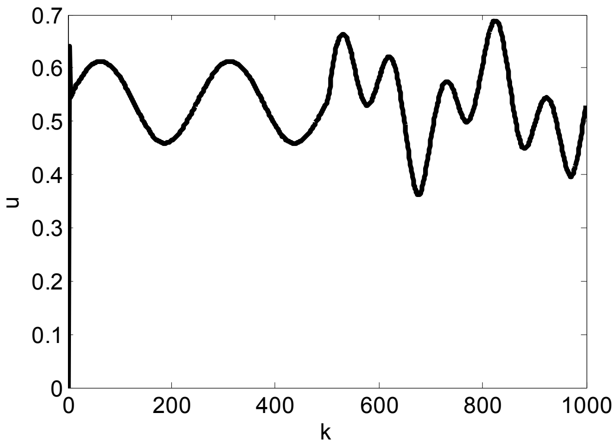

Fig. 17. The control law *u(k)* in presence of noise

The reference model output and that of the nonlinear plant for the reference input signal $r_1(k)$ are presented in Fig. 15. The error between the plant output and the reference model output is showed in Fig. 16. The control law is presented in Fig. 17.

In all figures, it is clear that the plant output follows the reference model output although the added noise. This simulation result shows the efficiency of the proposed algorithm, and its simplicity to treat complex nonlinearity.

Table 2. Mean Square Error performances

Mean Square Control Error (MSE)	Value
MSE (*case 1*)	$4.7452\ 10^{-4}$
MSE Control with added noise (*case 1*)	$4.5309\ 10^{-4}$
MSE Control (*case 2*)	$6.7464\ 10^{-5}$
MSE Control with added noise (*case 2*)	$4.8126\ 10^{-4}$

The control results are presented in Table 2 where the MSE is used as the performance index. From the above Table, it is proved that neural network controller performs well although the presence of noise.

From simulation results, the effect of the proposed method is clear where significant improvement can be seen in terms of tracking performance and by evaluating the control performance using the mean square error where the minimum is attained.

V. CONCLUSION

The current work presents an on-line Model Reference Adaptive Controller algorithm for a nonlinear system.

The control system contains Neural Network model reference, Neural Network controller and adaptation mechanism. This controller is designed in such a way that makes the dynamic system follows the reference model. The proposed approach shows good tracking results demonstrated through simulations. The adequate behavior of the system will depend on factors like model selection and different parameters of neural networks. This paper proposes an on-line dynamic neural network identification method for nonlinear systems, which gives suitable error estimation and it gives information of the convergence of the algorithm of updating the synaptic weights.The proposed technique may be very helpful to design an adaptive control strategy of nonlinear system.

REFERENCES

[1] Popa C., Popa A.,”Improving FCC Plant Performance with Model Reference Adaptive Control Based on Neural Network.” 8th International Conference on Electronics, Computers and Artificial Intelligence, 2016.

[2] Norgard M., Ravn O., PoulsenN.K., and HansenL.K.,”Neural Networks for Modelling and Control of Dynamic Systems”, Springer-Verlag, London, 2000.

[3] Errachdi A., Saad I., and Benrejeb M.,“Model reference adaptive control based-on neural networks for nonlinear time-varying system.”International Journal of Control, 2011.

[4] Jing S.,"Model Reference Adaptive Control, Encyclopedia of Systems and Control", Springer-Verlag London, pp 1-7,2014.

[5] Melin P., Castillo O.,"Soft Computing for Intelligent Control of Nonlinear DynamicalSystems", International Journal of Computational Cognition, Vol. 2, No.1, pp 45–78,March 2004.

[6] PrakashR., AnitaR., "Design of model reference adaptative intelligent controller using neural network for nonlinear systems", Journal of Electrical Engineering,2007

[7] Sung J., Jin B., Yoon H.,"Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural networks via adaptive learning rates", Information Sciences 177,pp 3074–3098, 2007.

[8] Agata N.,"Synthesis of neural network controller with a reference model ,Mechanics and control",Vol 29 No.1, 2010

[9] Priyaranjan M., Binay B.,"To Identify a Torque Controller System Approximating a Neural Network Based on Model Reference Technique", International Journal Of Engineering And Computer Science ISSN:2319-7242 Vol. 5, pp.17833-17838, 2016.

[10] NarendraK.S., ParthasarathyK.,"Identification and control of dynamical systems using neural networks". IEEE Transactions on Neural Networks, Vol. 1 (Issue 1), pp.4-27, 1990.

[11] Gang T., "Multivariable adaptive control", Automatica, Vol. 50 , pp. 2737–2764, 2014.

[12] Leeghim H., Kim D. "Adaptive neural control of spacecraft using control moment gyros". J. Adv. Space Res,2014.

[13] Marco A., Cesar A., Floriberto O., ElsaR, "Indirect hierarchical FCMAC control for the ball and plate system", Vol. 73, Issues 13–15, pp. 2454–2463,2010.

[14] Jianhua Z., Quanmin Z., Xueli W., Yang L., "Generalized indirect adaptive neural networks backstepping control procedure for a class of non-affine nonlinear systems with pure-feedback prototype", Neurocomputing Vol. 121, pp. 131–139, 2013.

[15] HenriquesJ.,GilP.,CardosoA.,CarvalhoP.,DouradoA., "Adaptive neural output regulation control of a solar power plant", Control.", Engineering Practice Vol. 18 ,pp. 1183–1196, 2010.

[16] Xiao-Li L., Chao J., De-Xin L., Da-Wei D., "Nonlinear adaptive control using multiple models and dynamic neural networks.", Neurocomputing Vol. 136, pp. 190–200 ,2014.

[17] PatinoH.D.,LiuD., "Neural network-based model reference adaptive control system". IEEE Trans. Syst. Man Cybern., 30(1),pp. 198- 204, 2000.

[18] HusainH., KhalidM.,YusofR., "Direct model reference adaptive controller based on neural fuzzy techniques for nonlinear dynamical systems". American Journal of Applied Sciences 5(7) ,pp. 769-776, 2008.

[19] Sanjay K.M.,"Adaptive control of nonlinear systems using Neural Networks", PhD thesis,University of Adelaide,Faculty of Engineering,1995.

[20] Yu F. , Yong P., Ningbo C., Daozhi G., Kuandi Z.,"Modeling reference evapotranspiration using extreme learning machine and generalized regression neural network only with temperature data". Computers and Electronics in Agriculture 136,pp.71–78, 2017.

[21] Lam H.K., Leung F., "On design of a switching controller for nonlinear systems with unknown parameters based on a model reference approach" . IEEE Int. Conference on Fuzzy systems, pp.1263-1266, 2001.

[22] Alessandro G.,Umberto M., "Application of a Robust Model Reference Adaptive Control Algorithm to a Nonlinear Automotive Actuator". International Journal of Automation and Computing 11(4), pp. 377-391, National Research Council, Naples, Italy, August 2014.

[23] PirabakaranK., BecerraV.M.,"PID autotuning using Neural Networks and Model Reference Adaptive Control, 15th Triennial World Congress, Barcelona, Spain,2002.

[24] Jun W., Zhang Y., Jacek M.Z., Bao-Liang L., Yin H.,"Advances in Neural Networks", third International Symposium on Neural Networks, Proceedings, Part 3, Chengdu, China, May 28 - June 1, 2006.

[25] Aström K.J., WittenmarkB., "Adaptive control: Second Edition",Lund Institute of technology, 2013.

[26] Errachdi A., Saad I.,Benrejeb M.," On-line identification of multivariable nonlinear system using neural networks". In IEEE CCCA, International Conference on Communications, Computing and Control, Hammamet, 3-5 Mars 2011.

[27] Errachdi A., Saad I.,Benrejeb M. ,"On-line identi¯cation method basedon dynamic neural network. International Review of Automatic Control (IREACO),3(5) : pp.474-479,2010.