Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 11 – 16
ISBN 978-83-60810-22-4
ISSN 1896-7094

A New RBF Network Based Sliding-Mode Control of Nonlinear Systems

Selami Beyhan
Department of Electrical and Electronics Engineering
Ege University, Izmir-TURKEY
selami.beyhan@ege.edu.tr

Musa Alcı
Department of Electrical and Electronics Engineering
Ege University, Izmir-TURKEY
musa.alci@ege.edu.tr

***Abstract*—In this paper, a novel radial basis function (RBF) neural network is proposed and applied successively for online stable identification and control of nonlinear discrete-time systems. The proposed RBF network is a one hidden layer neural network (NN) with its all parameters being adaptable. The RBF network parameters are optimized by gradient descent method with stable learning rate whose stable convergence behavior is proved by Lyapunov stability approach. The parameter update is succeeded by a new strategy adapted from Levenberg-Marquardth (LM) method. The aim of construction of the proposed RBF network is to combine power of the networks which have different mapping abilities. These networks are auto-regressive exogenous input model, nonlinear static NN model and nonlinear dynamic NN model. To apply the model to control of the nonlinear systems, a known sliding mode control is applied to generate input of the system. From simulations; it is sown that the proposed network is an alternative model for identification and control of nonlinear systems with accurate results.**

***Keywords*—RBF network, stable learning rate, sliding mode control, online system identification and control.**

I. Introduction

System identification is an important associative field for the control theory. A mathematical model of the system or artificial intelligent model which has same input-output characteristic with model is necessary to analyze and control the system. Modeling which is based on physical laws forms a mathematical model for the system. But for the identification, there is no need to use previous knowledge and physical structure of the system, so they are known as the black-box identification process [1]. The input-output characteristic of the nonlinear systems is changing naturally with high noise disturbance and its time varying behavior. So there is need to employ fine identifier models. The most applied methods in identification and control of the nonlinear systems are neural networks (NNs) and fuzzy logic systems (FLSs). The known supports of these methods are their ability to learn and good performance for the approximation of the nonlinear functions. These methods does perform highly nonlinear static mapping. However, for linear or less nonlinear systems, these nonlinear models are not well suited and so there is resulted less accurate of identification. To extract dynamics we need to use combinatorial models of linear and nonlinear models. In the development of the neural networks, new structures are introduced in everyday. New static and dynamic types of NNs and local and global recurrences and other mixed structures are developed numerously to get better identification [2]. The available information about the system is used in two ways for the identification. One way is the off-line procedure and the other way is online procedure. Off-line working is performed by collecting a batch of data to train the network and then this trained network is used for new data to obtain new responses. However, in online procedure, data is simultaneously used to get current output and to optimize the model parameters at the same time. To work online is actually a challenging task for strongly nonlinear systems and frequently considered important than off-line work. To capture the change in operating conditions and noise disturbances is also important task of the identification. In this work, online identification is aimed that so there is no explicit learning phase needed. In other words, the network is utilized for learning-while-functioning task, instead of learning then functioning.

II. Radial Basis Function Neural Networks

Radial basis function neural networks (RBFNNs) are the one of the different functionalized type of NNs with high approximation and regularization capability [3]. The RBFs are preferred as the basic structure of neural networks because of their good local specialization and global generalization ability [4]. The design of a RBFN in its most basic form consists of three separate layers. The first layer is the input layer. The second layer is the hidden layer and it is structured with high dimension to provide better approximation. The last layer gives the output of the network. There exists nonlinear transformation between the input layer and hidden layer. However, from hidden layer to the output layer it is linear transformation [2]. There are used some radial basis functions as functions of RBFNNs such that Gaussian RBFs, multiquadratic RBFs, inverse multiquadratic RBFs, thin plate splines RBFs, cubic splines RBFs, linear splines RBFs. However, Gaussian RBFs are employed frequently, since it is bounded, strictly positive and continuous on $\Re^n$ [2]. Moreover, they are known with noise suppression properties [5]. So in this study, Gaussian RBFs are utilized in the network.

$$R_i(x) = \exp\left(-\frac{\| x - u_i \|^2}{2\sigma_i^2}\right) \tag{1}$$

x is the input vector, u_i is the center and σ_i is the standard deviation of the Gaussian function, respectively. The optimization of the network for the adaptation of centers and standard deviation provides better approximation and interpolation capability as compared to the sigmoid functions [3]. To optimize the RBFNN parameters, there were used some different methods such that online gradient descent in [6], fast orthogonal search in [7], recursive orthogonal least squares in [8], Extended Kalman filter in [9], ant colony optimization in [10], respectively. In this study, the proposed network is optimized by gradient descent method with a new stable learning rate.

III. Proposed RBF network

The modeling scheme with two inputs and one output model is represented in Fig.1. After the realization of the construction, the network can be designed with different number of inputs and outputs. The proposed RBF network is constructed by two parts. First part is auto-regressive exogenous inputs (ARX) part. There exist past values of input-output terms. The second part is the static NN part and then dynamic recurrent NN part which are excited with same ARX terms. In the Fig.1, the ARX inputs are u_{k-1}, y_{k-1} and y_{k-2}. However, u_{k-1} and y_{k-1} inputs are used to excite the static and dynamic parts of the RBF network. These are known NN models; however they are not introduced together previously. Using the suitable optimization, the network is seen as a well alternative model for modeling. The dynamic NN part is called "Block-Diagonal Neural Network" in [11]. Because of the construction we also called the proposed network as "Mixed Structured RBF Network". The network has general properties as followings.

A. The Utilities of the Network

One Hidden Layer: The network is constructed especially, one hidden layer to reduce the complexity. So it is more applicable for the online identification for linear and nonlinear estimation models.

ARX part: To extract the linear dynamics of the system and to capture the change in system by past input-output terms, this part important for the approximation.

Static Part: The general neural network construction by two inputs is exemplified by this part. It is important and necessary for nonlinear static mapping.

Dynamic Part: The local recurrence of this block-diagonal part brings to the network a dynamic mapping. To extract internal dynamics of the nonlinear system, there are used local recurrences. So there exists good approximation.

RBF Functions: It is stated above that the importance of the RBF functions for approximation and interpolation is known superior properties of RBFs. Also it is empirically seen here that this same network is constructed with sigmoid functions and it has resulted poor approximation when compared to the RBFs. The general output formula of the network is given by,

$$\hat{y}(k+1)=\sum_{i=1}^{nd}\alpha_i u(k-i)+\sum_{j=1}^{np}\beta_j y(k-j)+ \sum_{k=1}^{nk}\gamma_k f\left(\frac{\|x_k^1-c_k\|^2}{2\sigma_k^2}\right)+\sum_{l=1}^{nr}\xi_l f\left(\frac{\|x_l^2-c_l\|^2}{2\sigma_l^2}\right) \quad (2)$$

where the *nd* and *np* are selected delays of the inputs and outputs. Also *nk* and *nr* are the number of the RBFs used for static and dynamic parts of the network, respectively.

α_i, β_j, γ_k and ξ_l are the output layer weights. In the network, they are seen as w_1 to w_7.

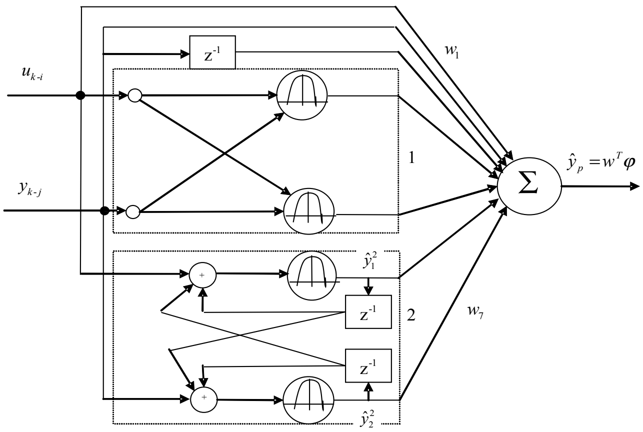

Figure 1. The proposed RBF network

The RBF function inputs are as follows;

for the static part which is indexed as 1,

$$\begin{aligned} x_1^1 &= w_{1,1}^1 u_{k-i} + w_{2,1}^1 y_{k-j} \\ x_2^1 &= w_{1,2}^1 u_{k-i} + w_{2,2}^1 y_{k-j} \end{aligned} \quad (3)$$

for dynamic part which is indexed as 2,

$$\begin{aligned} x_1^2 &= w_{1,1}^2 u_{k-i} + \hat{y}_1^2(k-1) + \hat{y}_2^2(k-1) \\ x_2^2 &= w_{2,2}^2 u_{k-i} + \hat{y}_1^2(k-1) + \hat{y}_2^2(k-1) \end{aligned} \quad (4)$$

where superscript 2 represents block 2, i.e. $w_{1,1}^2$ is weight from input 1 to the first RBF of block 2.

IV. Stable Learning Methods

The stability of the identification is as much important as the controller stability. To extract the dynamics of the system with well optimized parameters in stable sense is the basic necessity for identification. Identification stability is first related to the optimization method convergence to a local or global minimum. The convergence does not consider about the parameters magnitude or other characteristics. Therefore, when training the NNs and FLSs, the convergence of the error to the minimum does not show the stability of the system except some other stability conditions. Therefore, there has to be a law for stability in optimizing the parameters. This law is derived by the Lyapunov stability, Input-to-state stability, Bounded-Input Bounded-Output stability, passivity approach and etc. The other important thing is the time-variance of the learning rate. Before, some previously determined, constant stability guaranteed learning rates are used in algorithms. But the rate is chosen heuristically, so it does not provide the good convergence in the algorithm. However, if it is determined by online current knowledge of the inputs, the change in the parameters depends on the current change in the dynamics of the network. The convergence of the gradient descent with descent lemma [12] and the constraint in the learning rate with Lipschitz constant [13] and also Widrow-Hoff Learning [13] are the first important works about the learning rate selection. Some of the recent works about stable learning rate have been used for fuzzy neural network [14-16] such that,

$$\eta_k = \frac{\mu}{(1+ \| J(x(k)) \|^2)} \;;\; 0 < \mu \le 1 \tag{5}$$

where J(.) is the jacobian of the inputs. The author is in his another work [17] enhanced this learning rate as

$$\eta_k = \frac{\mu}{1+ \| J(x(k)) \|^2 + \frac{c}{k}} \tag{6}$$

where $0 < \mu \quad 1$ and $c>0$. The new term in denominator is used to regulate convergence speed. The other work [18] determines the learning rate by Lyapunov Stability approach as

$$\eta_k = \mu \frac{\| e \|^2}{(1+ \| J_p^T e \|^2 + \varepsilon)} \;;\; 0 < \mu \le 1 \tag{7}$$

where the e is error, J_p (.) is jacobian and ε is convergence speed regulator term, respectively. In that work, it is also proved that the local minimum is avoided in optimization. It is noticed that stability condition could also be satisfied without learning rate adaptation. Some of the other methods are about using structurally stable models [11,19] and different optimization algorithms combinations [20,21].

A. *Learning Algorithm*

The network is optimized by gradient descent algorithm [5] with stable learning rate.

$$\hat{W}(t+1) = \hat{W}(t) - \eta_t \frac{\partial E(t)}{\partial w(t)} \tag{8}$$

where $\hat{W}(t)$ is current parameter vector, E(t) is the quadratic cost function, i.e. $E(t)=0.5e^2(t)$. The employed learning rate η_t is selected as

$$\eta_k = \frac{1}{\varepsilon_k + \| J(x(k)) \|^2} \tag{9}$$

and μ is a constant as 1 in (5). However, ε_k parameter is adapted in the modeling of the system as LM method. So that the identification takes part a few iterations before control input is generated. The LM optimization [22] of the weights is

$$w_{n+1} = w_n - (\mu I + J_n^T J_n)^{-1} J_n^T e_n \tag{10}$$

where J_n is the jacobian of cost function w.r.t. the parameters, I(nxn) identity matrix, and μ is adaptation parameter for the convergence. The ε_k parameter is updated as follows.

If $E(k+1) >= E(k)$

Set : $\varepsilon_k = \varepsilon_k \delta$

Else if $E(k+1) < E(k)$

Set: $\varepsilon_k = \varepsilon_k / \delta$

End

Where E(k) is the cost value and δ is the user defined small constant as $0<\delta<1$. This parameter is updated up to the cost decrease is succeeded. Here, it is limited with 5 inner iterations and then control input is generated. After some iteration, because of correct parameters are obtained, there needs only one inner iteration. Its change does affect η_k and it has time-varying behavior for a certain decrease in cost. It is plotted in the simulations.

B. *Control Problem*

One of the models of nonlinear systems representations is,

$$y_p(k+1) = f[y_p(k),...,y_p(k-np)] \;\; g(y_p(k),...,y_p(k-np))u(k) \tag{11}$$

where *np* and *nd* delays of the inputs, and f(.) and g(.) are nonlinear functions of inputs. The modeling is applied to approximate the nonlinear parts f(.) and of the system. For the sake of simplicity, the model is used here to have direct input force as

$$y_p(k+1) = f(y_p(k),...,y_p(k-np)) + u(k) + d(k) \tag{12}$$

where f(k) is the nonlinear part of the system, u(k) is the control input and d(k) is the unknown but bounded disturbance such that $\|d(k)\| < d_m$. The sum of the f(k) and d(k), $fd(k) = f(k) + d(k)$ gives the unknown nonlinear part of the system to be identified. Using the proposed RBF network to identify the $f_d(k)$, the control effort u(k) is generated by a controller.

IV. Sliding Mode Control

Sliding mode control is frequently used for nonlinear systems control. Stability, reaching condition and chattering phenomena are known important difficulties [23]. For mathematically known models it is used directly to track the reference signals. However, for unknown models or much noised systems there is need to use an identifier to modeling the system and then sliding mode control is used to generate control input. Before stated above that the sliding mode is used with fuzzy systems [5] and neural networks [24, 25]. The introduction of sliding mode control with neural networks is taken from these works.

The state space representation of the *n*-th order discrete-time plant is given by,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ . \\ x_{n-1}(k+1) \end{bmatrix} = \begin{bmatrix} 0\ 1\ 0 \ldots\ldots 0 \\ 0\ 0\ 1 \ldots\ldots 0 \\ . \\ . \\ 0\ 0\ 0 \ldots\ldots 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ . \\ . \\ x_{n-1}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ x_n(k) \end{bmatrix} \tag{13}$$

and

$$x_n(k+1) = f(x(k)) + u(k) + d(k) \tag{14}$$

where $\bar{x}(k) = [x_1(k)\ x_2(k) \ldots x_n(k)]^T$ is the state vector and $f: \mathbb{R}^N \to \mathbb{R}$ is the unknown nonlinear function. Bar shows the vector parameter. Let us define the sliding surface s(k) as

$$s(k) = \bar{c}[\bar{x}(k) - \bar{x}_d(k)] = \bar{c}\ \bar{e}(k) \tag{15}$$

where $\bar{x}_d(k)$ desired value of the $\bar{x}(k)$ at time k and $\bar{c} = [c_1 \ldots c_n]$ is a constant vector and all roots must keep in the unit disk. Then s(k+1) is

$$\begin{aligned} s(k+1) &= c_1 e_1(k+1) + c_2 e_2(k+1) + \ldots \\ &\ldots + c_{n-1} e_{n-1}(k+1) + c_n e_n(k+1) \end{aligned} \tag{16}$$

Substitute the (13), (14) and (15) into (16) and sliding surface will become,

$$\begin{aligned} s(k+1) &= c_1 e_2(k) + \ldots\ldots\ldots\ldots + c_{n-1} e_n(k) + \\ &+ c_n[f_d(k) + u(k) - x_{nd}(k+1)] \end{aligned} \tag{17}$$

For sliding surface, the reaching condition is defined as

$$s(k+1) = -v s(k) \tag{18}$$

where the v parameter is defined in $0 < v < 1$. From (18) and (17) it is derived that,

$$\begin{aligned} -v s(k) &= c_1 e_2(k) + \ldots\ldots\ldots\ldots + c_{n-1} e_n(k) + \\ &+ c_n[f_d(k) + u(k) - x_{nd}(k+1)] \end{aligned} \tag{19}$$

the control input can be found as follows.

$$\begin{aligned} u(k) &= x_{nd}(k+1) - f_d(k) \\ &- \frac{1}{c_n}[c_1 e_2(k) + \ldots + c_{n-1} e_n(k) + v s(k)] \end{aligned} \tag{20}$$

In (20) the unknown part is the $f_d(k)$ and it is approximated as $\hat{f}_d(k)$ with an identifier. Here, its approximate value is found by the proposed RBF network at each time index k. The control input is produced online and used to track the $x_{nd}(k)$. It is seen that from (20), to produce the fine control input for the control system, the approximated value $\hat{f}_d(k)$ has crucial importance.

V. Simulation Study

In simulations, first a nonlinear system is identified by the proposed network and compared with previous works. Second, a nonlinear functioned system is identified and controlled by proposed methods and results are shown by figures.

A) Identification Simulation

In this part, Box-Jenkins gas furnace system identification is used to compare model capability before to employ for control.

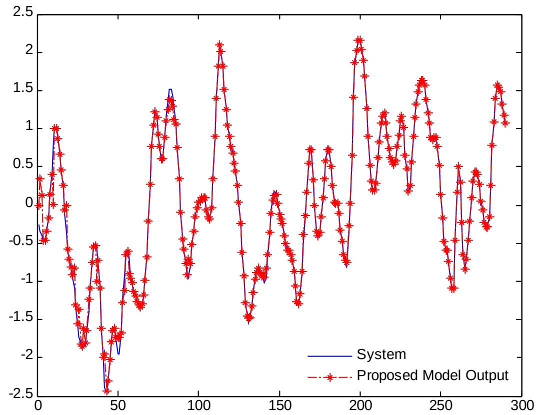

Figure 2. Figure 2. Box-Jenkins Data Identification

Here, because of online identification all data comes sample by sample and all of 290 samples of data is used for identification. The identification of Box-Jenkins system is shown in Fig.2. The resulting MSE is 7.9e-3 and that is a good result for online identification for this system.

B) Identification and Control Simulation

In this part of the simulation, identification and control is succeeded in the same time. The selected a two dimensional system [26] is;

$$x_1(k+1) = x_2(k)$$

$$x_2(k+1) = f(x(k)) + u(k) + d(k) \tag{21}$$

where

$$f(x(k)) = \frac{x_2(k)(x_1(k)+1)}{1 + x_1(k)^2 + x_2(k)^2} \tag{22}$$

The unknown disturbance is defined as a shot noise and continuous disturbance such that,

$$d(k) = \begin{cases} -1.5 & \text{if} \quad k = 0.45N \\ 2 & \text{if} \quad 0.75N < k <= N \end{cases} \tag{23}$$

where N is the number of total samples.

The controller is produced to force the state $x_2(k)$ to the reference signal is,

$$x_{2d}(k) = \begin{cases} sign(\sin(\frac{2\pi k}{200})) & \text{if}\ 0 < k <= 0.5N \\ \sin(\frac{2\pi k}{100}) & \text{if}\ 0.5N < k <= N \end{cases} \tag{24}$$

The parameters of the controller are selected as $c_1 = 0.1$, $c_2 = 0.2$ and $\nu = 0.2$ .The identifier model is selected as in Fig.1 and its inputs are $f_d(k-1)$ and $x_1(k-1)$. In Fig.3 the tracking result of the simulation is represented and it is seen that the even the disturbances occur in the system, it is controlled well to track the reference signal. In Fig.4 the network identification and modeling errors are plotted. From this figure, it is resulted that the proposed RBF model is modeling accurately and robust to the disturbances.

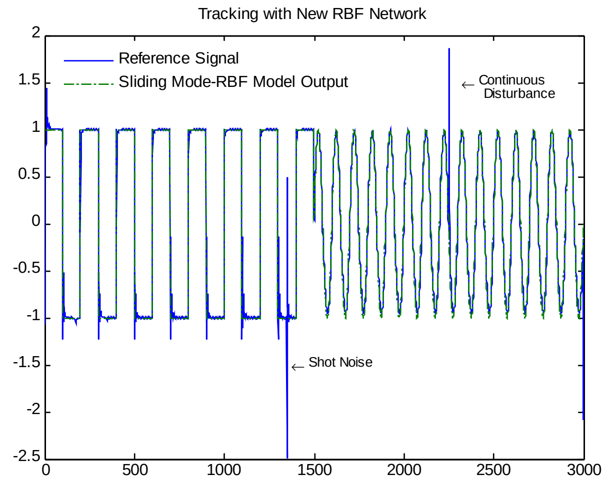

Figure 3. Figure 3. Tracking by the proposed model

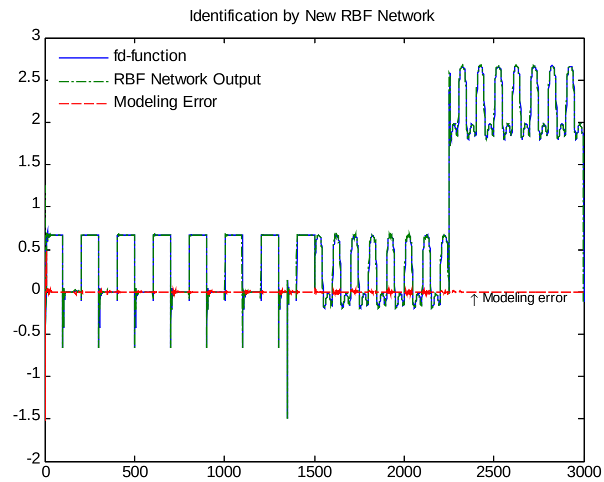

Figure 4. RBF Network Identification

In the simulation, identification and control mean square errors are 4.96e-4 and 17.9e-3 respectively. In Fig.5 the resulting sliding function and controlling errors are shown and the resulted sliding function is always close to the zero. Because of sharp transitions in square wave and disturbances, there exist large errors and sliding function values. However, the system quickly adapts itself to track the reference signal closely. In Fig.6 the resulting learning rates are represented. There is seen that the new strategy changes its value when abrupt changes occur. In the algorithm, there is used a limit as one for the value of the learning rate. However, from the Fig.6, it is seen that the values never exceeds the one.

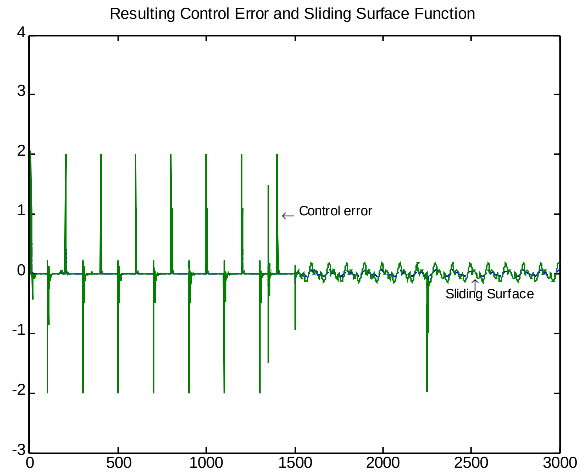

Figure 1. Sliding Function and Control Errors

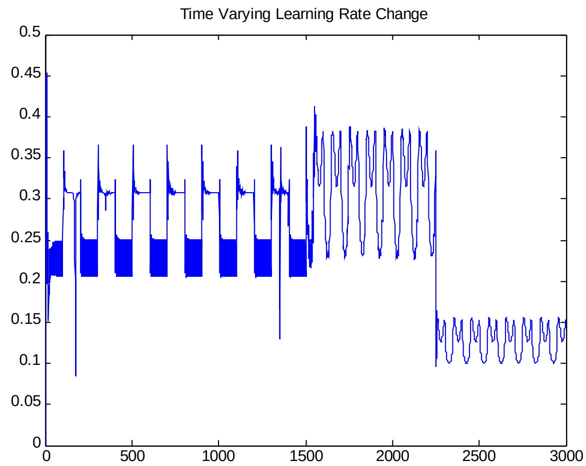

Figure 5. Figure 6. Learning Rate Change

There is a drawback which is the initialization of network parameters in modeling. To avoid unstable values, weight parameters are selected as 0.5, centers are selected as 2, and standard values are selected as 1.

VI. Conclusion

A new RBF network model is introduced for online system identification and control successively. By trial-and-error approach, to determine the constant learning rate for the networks is time consuming, difficult, and it does not guarantee stable on-line learning. Because, small learning rate causes smooth but slow convergences and it results large errors in the beginning. However, the large learning rate leads faster convergence and but large oscillating behavior, as well instability, so it results large errors in adaptation. So the learning rate must be chosen by current system information and satisfy the stability. In the study, a stable learning rate is utilized and it is convergence behave is also increased by Levenberg-Marquardth strategy. The parameter ε can be selected constant, but it is not well suited for the instant noises or continuous noises. As a result of this approach and the new RBF network the identification is performed well. It is first proved by Box-Jenkins gas furnace system and second, to approximate a nonlinear functioned model. Also sliding mode control is defined with its parameters which are selected to satisfy the stability condition. Even the shot noise and continuous noises are disturbed the nonlinear system, the proposed model forces to model to track reference signal correctly.

References

[1] Ljung L., *System Identification Theory for the User,* 1999, Prentice Hall PTR.

[2] Madan M. Gupta, Liang Jin, Noriyasu Homma, *Static and Dynamic Neural Networks From Fundamentals to Advanced Theory,* 2003, John Wiley&Sons Inc.

[3] Simon Haykin, Neural *Networks a Comprehensive Foundation,* 1998, Prentice Hall PTR.

[4] Li Yan, N. Sundararajan, P. Saratchandran, *Nonlinear System Identification Using Lyapunov Based Fully Tuned Dynamic RBF Networks,* Neural Processing Letters 12: 291-303, 2000.

[5] L-X. Wang, *A course in fuzzy systems and control,* USA, Prentice-Hall Inc. ISBN 0-13-593005-7, 1997.

[6] Shaohua Tan,Jianbin Hao, Joos Vandewalle, *Identification of Nonlinear Discrete-Time Multivariable Dynamical Systems by RBF Neural Networks,* 0-7803-1901-X/94$ 4.00 0,1994, IEEE

[7] W. Ahmed, D. M. Hummels and M. T. Musavi, *Application of Fast Orthogonal Search for the Design of RBFNN,* 0-7803-2570-2/95 $4.00 01995 IEEE.

[8] D. L. Yu, J. B. Gomm and D. Williams, *A Recursive Orthogonal Least Squares Algorithm for Training RBF Networks,* Neural Processing Letters 5: 167–176, 1997.

[9] Dan Simon, *Training radial basis neural networks with the extended Kalman Filter,* Neurocomputing 48 (2002) 455–475.

[10] Man Chun-tao, Li Xiao-xia, Zhang Li-yong, *Radial Basis Function Neural Network Based on Ant Colony Optimization,* 0-7695-3073-7/07 $25.00 © 2007 IEEE, DOI 10.1109/CIS.Workshops.2007.226

[11] P. A. Mastorocostas, J. B. Theocharis, *A Stable Learning Algorithm for Block-Diagonal Recurrent Neural Networks: Application to the Analysis of Lung Sounds,* IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, April 2006 Vol. 36, No. 2.

[12] D. P. Bertsekas, *Nonlinear Programming,* Athena Scientific, Belmont, Massachusttes, ISBN 1-886529-14-0, 1995.

[13] C. J. Principe., N. R. Euliano, W. C. Lefebvre, *Neural and Adaptive Systems,* John Wiley&Sons, 1999.

[14] Yu Wen, Xiaoou Li, 2004, *Fuzzy Identification Using Fuzzy Neural Networks With Stable Learning Algorithms,* IEEE Transactions On Fuzzy Systems, Vol. 12, No. 3, June 2004

[15] W. Yu, *State-Space Recurrent Fuzzy Neural Networks for Nonlinear System Identification,* DOI 10.1007/s11063-005-1523-4, Neural Processing Letters ,2005, 22:391–404.

[16] Wen Yu, M. A. Moreno-Armendariz, F. O. Rodriguez, 2007, *System identification using hierarchical fuzzy neural networks with stable learning algorithm,* Journal of Intelligent & Fuzzy Systems 18 171-183 IOS Press

[17] W. Yu, X. Li, *Online fuzzy modeling with structure and parameter learning,* Expert Systems with Applications 36, 2009, p7484–7492

[18] L. Behera, Swagat Kumar, Awhan Patnaik, *On Adaptive Learning Rate That Guarantees Convergence* in Feedforward Networks, IEEE Transactions On Neural Networks, Vol. 17, No. 5, September 2006

[19] F. Abdollahi, H. A. Talebi, R. V. Patel, *Stable Identification of Nonlinear Systems Using Neural Networks: Theory and Experiments,* IEEE/ASME Transactions On Mechatronics, 2006, Vol. 11, No. 4.

[20] M. A. Shoorehdeli, et al., *Identification using ANFIS with intelligent hybrid stable learning algorithm approaches,* Neural Comput. & Applic. 2008, DOI 10.1007/s00521-007-0168-9

[21] Y. Zhou, et al, *A new fuzzy neural network with fast learning algorithm and guaranteed stability for manufacturing process control,* Fuzzy Sets and Systems 132, 2002, p 201–216.

[22] L. E. Scales, *Introduction to nonlinear optimization,* Springer-Verlag New York Inc. ISBN 0-387-91252-5, 1985.

[23] Gao,W., Wang, Y., Homaifa, A.: *Discrete-Time Variable Structure Control Systems.* IEEE Transactions on Industrial Electronics Vol. 42 No. 2 April 1995

[24] S. Jagannathan, *Control of a Class of Nonlinear Discrete-Time Systems Using Multilayer Neural Networks,* IEEE Transactions On Neural Networks, Vol. 12, NO. 5, September 2001

[25] C.-H. Tsai, H-Y Chung , F-M Yu, *Neuro-Sliding Mode Control With Its Applications to Seesaw Systems,* IEEE Transactions On Neural Networks, Vol. 15, No. 1, January 2004 , Journal of Intelligent & Fuzzy Systems 18 ,(2007), 171-183

[26] K. S. Narendra, K. Parthasarathy, *Identification and control of dynamical systems using neural networks,* IEEE Trans. Neural Netw. 1 (1) (1990) 4–27.