Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Stable Learning Algorithm Approaches for ANFIS as an Identifier

M. Aliyari Shoorehdeli, M. Teshnehlab, A. K. Sedigh

Abstract- **This study suggests new learning laws for Adaptive Network based Fuzzy Inference System that is structured on the basis of TSK type III as a system identifier. Stable learning algorithms for consequence parts of TSK type III rules are proposed on the basis of the Lyapunov stability theory and some constraints are obtained. Simulation results are given to validate the results. It is shown that instability will not occur for learning rates in the presence of constraints. The learning rate can be calculated online from the input–output data, and an adaptive learning for the Adaptive Network based Fuzzy Inference System structure can be provided.**

Key words: **Learning Rate, Optimization, TSK fuzzy System, ANFIS, Lyapunov Theory, Identification and Stability Analysis.**

I. Introduction

Fuzzy systems and neural networks are both very popular techniques that have seen increasing interest in recent years. At first glance, they seem to be totally different areas with marginal connections to each other. However, both methodologies belong to the soft computing area. Soft computing includes approaches to human reasoning that try to make use of the human tolerance for incompleteness, uncertainty, imprecision, and fuzziness in a decision making process. Many different structures for Fuzzy neural Networks (FNNs) have been proposed [1]. Among them, Adaptive Network based Fuzzy Inference System (ANFIS) is a neural network based on fuzzy approach, in which the learning procedures are performed by interleaving the optimization of the antecedent and consequent part parameters. In this study, the parameter adaptation procedures for the consequent parameters in ANFIS employ gradient–descent (GD) methods to adjust the membership functions' (MFs') parameters. The consequent parameters are very important and they could lead ANFIS to instability easily, but the antecedent parameters have a little impress on ANFIS instability. The gradient techniques have the advantage of being less computationally expensive for a given size network topology; a factor that becomes significant for larger networks. However, one problem inherent in them is their convergence to local minima and the user set parameters are sensitive to the learning [2]. The stability problem of fuzzy neural network identification is very important in applications.

M. Aliyari Shoorehdeli is with computer department of Science and Research Branch, Islamic Azad University, Hesarak, Poonak, Tehran, Iran.
M. Aliyari Shoorehdeli, M. Teshnehlab and A. K. Sedigh are with department of K. N. Toosi University of Tech Tehran, Iran (emails: m_aliyari@eetd.kntu.ac.ir, teshnehlab@eetd.kntu.ac.ir , Sedigh@kntu.ac.ir).

It is well known that normal identification training algorithms (e.g., gradient descent and least square) are stable in ideal conditions. In the presence of unmodeled dynamics, they might become unstable [3]. The learning procedure of fuzzy neural networks can be regarded as a type of parameter identification.

The backpropagation (BP) learning of the FNN model, on the basis of GD technique is stable, if FNN models can match nonlinear plants exactly [3].

The stability of GD algorithms for fuzzy neural networks using type I and II TSK's rules has been discussed in many studies [3–9], but there is no work on stability of ANFIS as a fuzzy neural network using type III TSK's rules.

Recently, the stability of type III systems has attracted considerable interest in the fuzzy literature [10–14] but they did not focus on ANFIS structure.

Some studies have been made on the stability of fuzzy neural network with TSK type II rules and the popular method is Lyapunov stability theorem [4, 6, 15]. Most of these results require the existence of common quadratic Lyapunov function, [10–13]. Sonbol and Fadali [14] proposed a new method of stability without using Lyapunov theorem for stability of type III fuzzy systems. Nevertheless there are few studies on stability analysis using the convergence of learning algorithm. Yu and Li [3] used input to state stability (ISS) techniques for Mamdani and TSK fuzzy neural networks but the stability of ANFIS as an identifier has not been studied. In this study, the Lyapunov stability approach is applied to system identification via ANFIS as TSK's type III. The GD update rule for consequent is considered. The new stable algorithm with time varying learning rate is applied to ANFIS.

The rest of article is organized as follows: in Section II ANFIS structure and learning algorithms are reviewed. In Section III ANFIS stability analysis are discussed and stability constraints found. Simulation and application of this method to nonlinear identification is presented in Section IV. Section V presents conclusions.

II. The Concept of ANFIS

A. ANFIS Structure

Here, type III ANFIS topology and the learning method used for this neuro-fuzzy network are presented. Both Neural Network and Fuzzy Logic [16] are model-free estimators and share the common ability to deal with the uncertainties and noise. Both of them encode the information in parallel and

distribute architecture in a numerical framework. Hence, it is possible to convert fuzzy logic architecture to a neural network and vice versa. This makes it possible to combine the advantages of neural network and fuzzy logic. A network obtained in this way could be used with excellent training algorithms that neural networks have at their disposal, to obtain the parameters that would not have been possible in fuzzy logic architecture. Moreover, the network obtained in this way would not remain a black box, because this network would have fuzzy logic capabilities to interpret in terms of linguistic variables [17].
The ANFIS combines the two approaches of neural network and fuzzy systems. If both these two intelligent approaches are combined, it will achieve good reasoning in quality and quantity. In other words, both fuzzy reasoning and network calculation will be available simultaneously. For m-dimensional neuro-fuzzy identifiers which implement an m-to-one mapping, the number of fuzzy rules or parameters exponentially increases with the number of input variables. This is the problem known as curse of dimensionality [24]. For this reason, it can be difficult to design and implement a high-dimensional neuro-fuzzy identifier when the number of input variables is large [25].

The ANFIS's network is composed of two parts similar to fuzzy systems. The first part is the antecedent and the second part is the consequent, which are connected to each other by rules. The ANFIS structure can be viewed as a multilayered neural network as shown in Fig. 1. The first layer executes a fuzzification process, the second layer executes the fuzzy AND (product) of the antecedent part of the fuzzy rules, the third layer normalizes the membership functions (MFs), the fourth layer executes the consequent part of the fuzzy rules, and the last layer computes the output of fuzzy system by summing up the outputs of layer four. The feedforward equations of the ANFIS with two inputs and two labels for each input shown in Fig. 1 are as follows:

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \qquad i = 1,2. \tag{1}$$

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \qquad i = 1,2. \tag{2}$$

$$\left.\begin{aligned} f_1 &= p_1 x + q_1 y + r_1 \\ f_2 &= p_2 x + q_2 y + r_2 \end{aligned}\right\} \Rightarrow f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} = \bar{w}_1 f_1 + \bar{w}_2 f_2 \tag{3}$$

To model complex nonlinear systems, the ANFIS model carries out input space partitioning that splits the input space into many local regions, from which simple local models (linear functions or even adjustable coefficients) are employed. The ANFIS uses fuzzy MFs for splitting each input dimension; the input space is covered by MFs with overlapping; that is, several local regions can be activated simultaneously by a single input. As simple local models are adopted in the ANFIS model, the ANFIS has a high ability of approximation that will depend on the resolution of the input space partitioning. Input space partitioning is determined by the number of MFs in the antecedent part of ANFIS. Usually, the MFs are used as bell-shaped with maximum grade equal to 1 and minimum grade equal to zero such as:

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i}\right)^2\right]^{b_i}}, \tag{4}$$

$$\mu_{A_i}(x) = \exp\left\{-\left[\left(\frac{x - c_i}{a_i}\right)^2\right]^{b_i}\right\} \tag{5}$$

Where $\{a_i, b_i, c_i\}$ are the parameters of MFs which affects the shape of MFs.

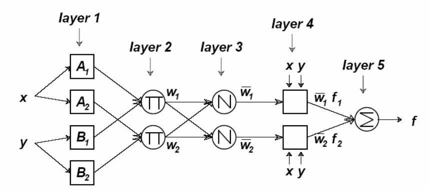

Figure 1: The equivalent structure of ANFIS (type III ANFIS)

B. Learning Algorithms

Subsequent to the development of ANFIS approach, a number of methods have been proposed for learning rules and for obtaining an optimal set of rules. For example, Mascioli et al. [18] have proposed to merge Min-Max and ANFIS model to obtain neuro-fuzzy network and determine an optimal set of fuzzy rules. Jang and Mizutani [19] have presented application of Levenberg-Marquardt method, which is essentially a nonlinear least-squares technique, for learning the ANFIS parameters. Jang [20] has presented a scheme for input selection and Kumar and Garg [17] have used Kohonen's map for training.

Jang [21] introduced four methods to update the parameters of the ANFIS structure, as listed below according to their computation complexities:

1. Gradient decent (GD) only: all parameters are updated by the gradient descent.
2. Gradient decent only and one pass of least mean square error (LSE): the LSE is a technique applied only once at the very beginning to get the initial values of the consequent parameters and then the gradient decent takes over to update all parameters.
3. Gradient decent only and LSE: this is the hybrid learning.
4. Sequential LSE: using extended Kalman filter to update all parameters.

These methods update antecedent parameters by using GD or Kalman filtering. These methods have a high complexity. The other method that can be mentioned here is the use of hybrid optimization method like PSO for antecedent part and GD for consequent part [22]. Chen [2] compares several popular training algorithms in tuning parameters of fuzzy membership functions (MFs). The algorithms compared are GD, Resilient propagation (RPROP), Quickprop (QP), and Levenberg-Marquardt (LM) algorithms. These algorithms are combined with RLSE (Recursive Least Squares Estimate) to improve the efficiency of ANFIS.

III. System Stability Analysis

Suppose an ANFIS with n inputs and for each input there are k_1 , k_2, ... and k_n membership functions, respectively. The feed forward algorithm of ANFIS is:

$$w_i = \mu_{A_1^{i_1}}(x_1) \times \mu_{A_2^{i_2}}(x_2) \times \cdots \times \mu_{A_n^{i_n}}(x_n) \quad (6)$$

where $i_j \in \{1,\ 2\ ,\cdots, k_j\}$, $j = 1,2,...,n$

and if:

$$f_i = \sum_{l=1}^{n} a_l^i x_l + b^i \quad , \quad i = 1,2,\cdots,R \quad (7)$$

then the output of the ANFIS will be:

$$O = \frac{\sum_{i=1}^{R} w_i \ f_i}{\sum_{i=1}^{R} w_i} \quad (8)$$

Now, the feedback algorithm is defined in the following steps. The objective function is defined as:

$$e(k) = y_d(k) - O(k) \quad , \quad E = \frac{e^2(k)}{2} \quad (9)$$

where, $y_d(k)$ and $O(k)$ are the desired and ANFIS outputs, respectively.

Now some matrices and vectors are defined. The input vector is:

$$\vec{X}(k) = [x_1(k),\ x_2(k)\ ,\ \cdots, x_n(k),\ 1] \quad (10)$$

The consequent weights are defined as follows:

$$_1\vec{A} = [a_1^1 \quad a_1^2 \quad \cdots \quad a_1^R]^T \quad , \quad _2\vec{A} = [a_2^1 \quad a_2^2 \quad \cdots \quad a_2^R]^T$$
$$, \ \cdots \ , \quad _n\vec{A} = [a_n^1 \quad a_n^2 \quad \cdots \quad a_n^R]^T$$
$$\vec{B} = [b^1 \quad b^2 \quad \cdots \quad b^R]^T \quad (11)$$

Now, A is defined as follows:

$$A = [_1\vec{A} \quad _2\vec{A} \quad \cdots \quad _n\vec{A} \quad \vec{B}]_{R\times(n+1)} \quad (12)$$

The output in (8) will be:

$$O = \begin{bmatrix} \frac{w_1}{\sum_{i=1}^{R} w_i} & \frac{w_2}{\sum_{i=1}^{R} w_i} & \cdots & \frac{w_R}{\sum_{i=1}^{R} w_i} \end{bmatrix} \times \left(A\vec{X}\right) \quad (13)$$

The antecedent part parameters are defined by two types of parameters; the first type is the means of the MFs:

$$_1\vec{M} = [m_1^1 \quad m_1^2 \quad \cdots \quad m_1^{k_1}]^T \ , \ _2\vec{M} = [m_2^1 \quad m_2^2 \quad \cdots \quad m_2^{k_2}]^T$$
$$, \ \cdots \ , _n\vec{M} = [m_n^1 \quad m_n^2 \quad \cdots \quad m_n^{k_n}]^T$$
$$M = [_1\vec{M}^T \quad _2\vec{M}^T \quad \cdots \quad _n\vec{M}^T] \quad (14)$$

The second type is the standard deviations of the MFs:

$$_1\vec{S} = [s_1^1 \quad s_1^2 \quad \cdots \quad s_1^{k_1}]^T \quad , \quad _2\vec{S} = [s_2^1 \quad s_2^2 \quad \cdots \quad s_2^{k_2}]^T$$
$$, \ \cdots \ , _n\vec{S} = [s_n^1 \quad s_n^2 \quad \cdots \quad s_n^{k_n}]^T \quad (15)$$

and the vector form of (15) is defined as:

$$S = [_1\vec{S}^T \quad _2\vec{S}^T \quad \cdots \quad _n\vec{S}^T] \quad (16)$$

Now a discrete Lyapunov function is defined as follows:

$$V(k) = E(k) = \frac{1}{2} e^2(k) \quad (17)$$

Then, the change of Lyapunov function at each iteration will be:

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2}\left(e^2(k+1) - e^2(k)\right)$$
$$= \frac{1}{2}\left(e(k+1) - e(k)\right)\left(e(k+1) + e(k)\right)$$
$$= \frac{1}{2}\Delta e(k)\left(\Delta e(k) + 2e(k)\right) = \Delta e(k) \ \left(\frac{1}{2}\Delta e(k) + e(k)\right) \quad (18)$$

The change of error caused by the parameters can be approximated by:

$$\Delta e(k) = \underbrace{\left(\frac{\partial e(k)}{\partial S(k)}\right)^T \Delta S(k)}_{\alpha(k)} + \underbrace{\left(\frac{\partial e(k)}{\partial M(k)}^T \Delta M(k)\right)}_{\beta(k)} + \underbrace{tr\left((\frac{\partial e(k)}{\partial A(k)})^T \Delta A(k)\right)}_{\gamma(k)} \quad (19)$$

where, the $tr(.)$ is the trace of matrices. Here we just train the consequent parameters so:

$$\alpha(k) = 0 \ , \ \beta(k) = 0 \quad (20)$$

From equations (18), (19) and (20) the following equation is obtained:

$$\Delta V(k) = \left(\gamma(k)\right) \times \left(\frac{\gamma(k)}{2} + e(k)\right) \quad (21)$$

Now consider (19, 21):

$$\Delta V(k) = \gamma(k)\left(\frac{\gamma(k)}{2} + e(k)\right)$$
$$= tr\left(\left(\frac{\partial e(k)}{\partial A(k)}\right)^T \Delta A(k)\right) \times \left(e(k) + \frac{1}{2} tr\left(\left(\frac{\partial e(k)}{\partial A(k)}\right)^T \Delta A(k)\right)\right) \quad (22)$$

by using the chain rule:

$$\frac{\partial e(k)}{\partial A(k)} = \frac{\partial e(k)}{\partial O(k)} \times \frac{\partial O(k)}{\partial A(k)} = -\frac{\partial O(k)}{\partial A(k)} ,$$
$$\Delta A(k) = \eta_A e(k) \frac{\partial O(k)}{\partial A(k)} \quad (23)$$
$$\left(\left\|\frac{\partial O(k)}{\partial A(k)}\right\|_F\right)^2 = tr\left(\left(\frac{\partial O(k)}{\partial A(k)}\right)^T \left(\frac{\partial O(k)}{\partial A(k)}\right)\right)$$

where η_A is the learning rate that is used to adjust consequent parameters and let $\|\cdot\|_F$ be the Frobenius norm.

Considering (23), (22) can be rewritten as follows:

$$\Delta V(k) = -\eta_A e^2(k)\left(\left\|\frac{\partial O(k)}{\partial A(k)}\right\|_F\right)^2 + \frac{1}{2}\eta_A{}^2 e^2(k)\left(\left\|\frac{\partial O(k)}{\partial A(k)}\right\|_F\right)^4 \quad (24)$$

If

$$D_A(k) = \frac{\partial O(k)}{\partial A(k)} , \quad D_A^{\max} = \max_k \|D_A(k)\|_F \quad (25)$$

then:

$$\Lambda_A = \eta_A \left(\|D_A(k)\|_F\right)^2 - \frac{1}{2}\eta_A{}^2 \left(\|D_A(k)\|_F\right)^4$$
$$= \frac{1}{2}\left(\|D_A(k)\|_F\right)^2 \eta_A \left(2 - \left(\|D_A(k)\|_F\right)^2 \eta_A\right) \quad (26)$$

is obtained:

$$\Lambda_A \geq \frac{1}{2}\left(\|D_A(k)\|_F\right)^2 \eta_A \left(2 - \left(\|D_A(k)\|_F\right)^2 \eta_A\right) \quad (27)$$

For stability condition, change of the Lyapunov function must be less than zero and from (24–27) the following equation is derived:

$$\Delta V(k) < 0 , \quad \Delta V(k) = -\Lambda_A e^2(k) \Rightarrow \Lambda_A > 0$$
$$\Rightarrow 0 < \eta_A < \frac{2}{\left(\|D_A(k)\|_F\right)^2} \quad (28)$$

(28) is an adaptive constraint where, the learning rate stability condition changes in each iteration. Thus, this constraint can be used easily for online training.

From (25 and 28):

$$0 < \eta_A < \frac{2}{\left(D_A^{\max}\right)^2} \quad (29)$$

where equation (29) is a conservative constraint and cannot be calculated in online identification.

The following equation can be written by using the chain rule:

$$\gamma(k) = tr\left(\left(\frac{\partial e(k)}{\partial A(k)}\right)^T \Delta A(k)\right)$$
$$= -\eta_A e(k) tr\left(\left(\frac{\partial O(k)}{\partial A(k)}\right)^T \frac{\partial O(k)}{\partial A(k)}\right) \quad (30)$$
$$= -\eta_A e(k)\left(\left\|\frac{\partial O(k)}{\partial A(k)}\right\|_F\right)^2$$
$$= -\eta_A e(k)\left(\|D_A(k)\|_F\right)^2$$

At each learning step, the learning rate of consequent part is selected to satisfy equation (30) to ensure that our identifier is stable. This adaptive method is performed online during system operation. The simulation result of a system is shown in the next section.

IV. Simulation and results

In this section, the suggested stable learning algorithm from (30) is applied to a function approximation problem and identification of a chaos system. The objective of this section is to show the truth of (30) constraints. In this section, it is shown that if learning rate are chosen from the proposed constraints, the stability of identifier will be guaranteed. Consider using ANFIS with stable training algorithms to a 2-input nonlinear Sinc model equation as:

$$z = \text{sinc}(x, y) = \frac{\sin(x)}{x} \times \frac{\sin(y)}{y} \quad (31)$$

From the grid points of the range $[-10,10]\times[-10,10]$ within the input space of the above equation, 121 training data pairs were first obtained. The ANFIS used here contains 16 rules, with four MFs for each of the inputs and uses 72 data for testing. The initial values for MF's parameters are chosen in such a way that partition input range in equal parts.

The consequent parameters (A) were trained. It means that the $\Delta M = \Delta S = 0$, in this and next example the MF's means are choose normally from input range and variances are fix and equal.

Figures 2, 3, 4, and 5 explicitly show how the obtained constraints provide stability for the whole learning process in ANFIS architecture. It can be seen that the upper boundary is truly calculated and the criterion is very sensitive, hence, a slight change of 0.01% will bring the condition from stability to instability and vice versa. From Figures 5, 6, 7, and 8, it can be concluded that the upper boundaries are obtained true and that (28) is exactly true.

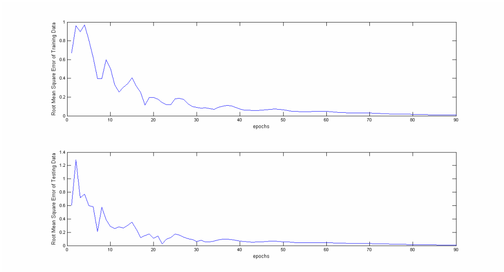

Figure 2: Training error RMS-epochs for the case,

$$\eta_A = \frac{2}{\left(\|D_A(k)\|_F\right)^2}\times 0.99$$

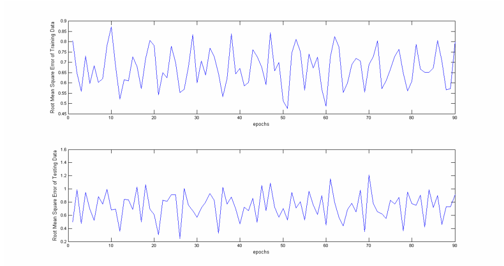

Figure 3: Training error RMS-epochs for the case,

$$\eta_A = \frac{2}{\left(\|D_A(k)\|_F\right)^2}$$

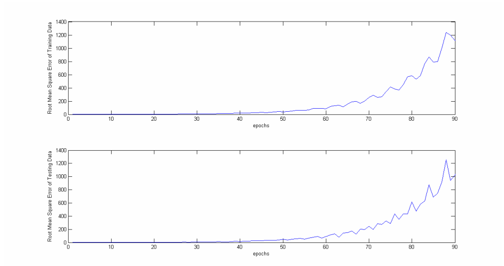

Figure 4: Training error RMS-epochs for the case,

$$\eta_A = \frac{2}{\left(\|D_A(k)\|_F\right)^2}\times 1.001$$

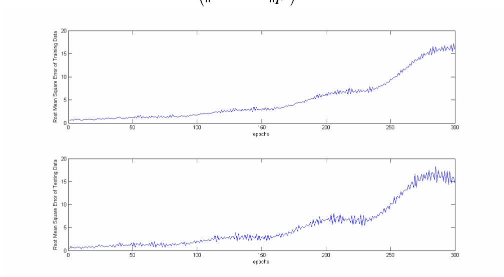

Figure 5: Training error RMS-epochs for the case,

$$\eta_A = \frac{2}{\left(\|D_A(k)\|_F\right)^2}\times 1.0001$$

The second example is to predict future values of a chaotic time series, which is generated by

$$\dot{x} = \frac{0.2x\left(t-\tau\right)}{1+x^{10}\left(t-\tau\right)} - 0.1x\left(t\right) \qquad (32)$$

(32) is also known as the chaotic Mackey-Glass differential delay equation [23]. The initial conditions for $x(0)$ and τ are 1.2 and 17, respectively. Here, 500 data are used, which are similar to the training data (because the test data results will be similar to training data and these constrains will be shown in training data clearly). The ANFIS used here contains 16 rules, four inputs with two MFs for each of the inputs. The inputs and outputs are chosen as follows:

$$\begin{aligned} inputs &= [x(t-18), x(t-12), x(t-6), x(t)] \\ output &= [x(t+6)] \end{aligned} \qquad (58)$$

For this example, Figures 6, 7, 8 and 9 explicitly show how the obtained constraints provide stability for the whole learning process in ANFIS architecture. The upper boundary is truly calculated and the criterion is very sensitive again, hence, a slight change of 0.01% will bring the condition from stability to instability, and vice versa.

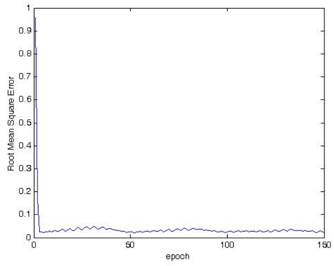

Figure 6: Training error RMS-epochs for the case,

$$\eta_A = \frac{2}{\left(\|D_A(k)\|_F\right)^2}\times 0.99$$

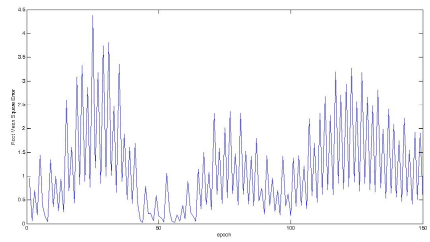

Figure 7: Training error RMS-epochs for the case,

$$\eta_A = \frac{2}{\left(\|D_A(k)\|_F\right)^2}$$

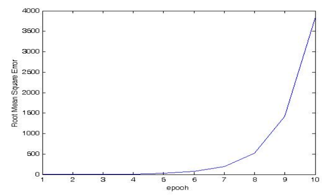

Figure 8: Training error RMS-epochs for the case,

$$\eta_A = \frac{2}{\left(\|D_A(k)\|_F\right)^2}\times 1.001$$

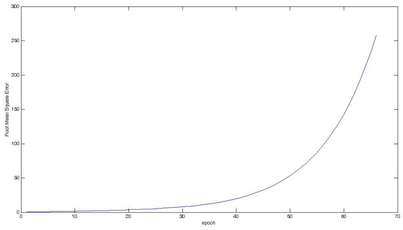

Figure 9: Training error RMS-epochs for the case,

$$\eta_A = \frac{2}{\left(\left\|D_A(k)\right\|_F\right)^2} \times 1.0001$$

V. Conclusion

This study applies Lyapunov stability approaches to ANFIS fuzzy neural networks for the first time and proposes stable learning algorithms that can guarantee the stability during the training process. The proposed algorithms are effective and are tested in several simulations. The main contributions are as follows:

- ✓ The Lyapunov stability approach provides a certain criterion for learning rates of ANFIS consequent structure that guarantees the stability of the algorithm through the learning process.
- ✓ In this study, the effective learning rate in different simulations was obtained.
- ✓ In a future work, it might be possible to find the best learning rate from this range and ensure that the system will be stable in the identification process.

Therefore, using Lyapunov stability approach, stable updating laws for the membership functions, and consequent parameters of ANFIS are proposed, and this will ascertain stability during the training process.

VI. References

[1] Y. Zhou, S. Li, R. Jin, "*A new fuzzy neural network with fast learning algorithm and guaranteed stability for manufacturing process control*", Fuzzy sets and systems, Elsevier, 2001.

[2] M.S. Chen, "*A Comparative Study of Learning methods in Tuning Parameters of Fuzzy Membership Functions*", IEEE conf on Systems, Man, and Cybernetics, Vol.3, 40–44, 1999.

[3] W. Yu and X. Li, "*Fuzzy Identification Using Fuzzy Neural Networks With Stable Learning Algorithms"*, IEEE Trans. on Fuzzy Syst., Vol. 12, No. 3, 2004.

[4] W.C. Kim, S.C. Ahn, W.H. Kwon, "*Stability analysis and stabilization of fuzzy state space models*", Fuzzy Sets and Systems Vol. 71, 131–142, 1995.

[5] Ching-Hung Lee and Ching-Cheng Teng, "*Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks*", IEEE Trans., on Fuzzy Syst., Vol. 8, No. 4, 349–366, 2000.

[6] Yu, X. Li, "*Fuzzy Neural Modeling Using Stable Learning Algorithm*", Proceedings of American Control Conference Denver, Colorado, 4542–4547, 2003.

[7] M. Sugeno, *"On Stability of Fuzzy Systems Expressed by Fuzzy Rules with Singleton Consequents"*, IEEE Trans. on Fuzzy Syst., Vol. 7, No. 2, 1999.

[8] W. Yu and A. Ferreyra, *"System Identification with State-Space Recurrent Fuzzy Neural Networks"*, 43rd IEEE Conference on Decision and Control, 5106–5111, December 14–17, 2004.

[9] W. Yu, X. Li, *"Stable fuzzy Identification using Recurrent Fuzzy Neural Networks"*, Proceeding of IASTED on Neural Networks and Computation Intelligent, Mexico, 2003.

[10] M.A.L. Thathachar and P. Viswanath, "*On the stability of fuzzy systems*," IEEE Trans. Fuzzy Syst., Vol. 5, No. 1, 145–151, 1997.

[11] B. Kosko, "*Global stability of generalized additive fuzzy systems,*" IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., Vol. 28, No. 3, 441–452, Aug. 1998.

[12] J. Joh, Y.H. Chen, R. Langari, "*On the stability of linear Takagi–Sugeno fuzzy models,*" *IEEE Trans. Fuzzy Syst.*, Vol. 6, No. 3, 402– 410, Jun. 1998.

[13] S.M. Guu and C.T. Pang, *"On the asymptotic stability of free fuzzy systems,*" *IEEE Trans. Fuzzy Syst.*, Vol. 7, No. 4, 467–468, Aug. 1999.

[14] A. Sonbol and M.S. Fadali, "*Stability Analysis of Discrete TSK Type II/III Systems,*" IEEE Trans. Fuzzy Syst, accepted for inclusion in a future issue.

[15] M.M. Polycarpou, and P.A. Ioannou, "*Learning and convergence analysis of neural-type structured networks*", IEEE Trans. on Neural Networks, Vol. 3, 39–50, 1992.

[16] 13. R.R.Yager, and L.A. Zadeh, "*Fuzzy Sets Neural Networks, and Soft Computing*", Van Nostrand Reinhold, 1994.

[17] Manish Kumar, and Devendra P. Garg "I*ntelligent Learning of Fuzzy Logic Controllers via Neural Network and Genetic Algorithm*", Proceedings of JUSFA Japan – USA Symposium on Flexible Automation Denver, Colorado, July 19–21, 2004.

[18] F.M. Mascioli, G.M Varazi,. G. Martinelli, "*Constructive Algorithm for Neuro-Fuzzy Networks*", Proceedings of the 6th IEEE International Conference on Fuzzy Systems, Vol. 1, 459 –464, 1997.

[19] Jang, J.-S. R., and Mizutani, E., "*Levenberg-Marquardt Method for ANFIS Learning*", Biennial Conference of the North American Fuzzy Information Processing Society, 87 –91, 1996.

[20] J.-S.R. Jang, , "*Input Selection for ANFIS Learning*", Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, Vol. 2, 1493–1499, 1996.

[21] Jyh-Shing Roger Jang, "*ANFIS: Adaptive-Network-Based Fuzzy Inference System*", *IEEE Trans. Syst., Man and Cybernetics*., Vol. 23, No. 3, 1993.

[22] M. Aliyari Sh., M. Teshnehlab, A. K. Sedigh, "*A Novel Training Algorithm in ANFIS Structure*", American Control Conferences (ACC), 2006.

[23] M.C. Mackey and L. Glass, "*Oscillation and Chaos in Physical control System*" Science, Vol 197, 287–289, July 1977.

[24] G.V.S.Raju, Jun Zhou, Roger A.Kisner*, "Hierarchical fuzzy control"*, Int. J. Control, vol. 54, No. 5, pp. 1201-1216, 1991.

[25] Hao Ying, *"Fuzzy control and modeling: analytical foundations and applications"*, IEEE Press, 2000.