RESEARCH ARTICLE OPEN ACCESS

Convergence Analysis of Adaptive Recurrent Neural Network

Hong Li*, Ali Setoodehnia**
*(Department of Computer Systems Technology, New York City College of Technology, New York, USA)
** (Department of Electronics Engineering Technology, ECPI University, USA)

ABSTRACT
This paper presents analysis of a modified Feed Forward Multilayer Perceptron (FMP) by inserting an ARMA (Auto Regressive Moving Average) model at each neuron (processor node) with the Backp ropagation learning algorithm. The stability analysis is presented to establish the convergence theory of the Back propagation algorithm based on the Lyapunov function. Furthermore, the analysis extends the Back propagation learning rule by introducing the adaptive learning factors. A range of possible learning factors is derived from the stability analysis. Performance of such network learning with adaptive learning factors is presented and demonstrates that the adaptive learning factor enhance the performance of training while avoiding oscillation phenomenon.
***Keywords*–**Adaptive learning, Back propagation, Neural networks, Stability analysis

I. INTRODUCTION

In the last few decades, Artificial Neural Networks (ANNs) have been successfully applied to many real world applicationsand proven to be useful in various tasks of modeling nonlinear systems, such as signal processing, pattern recognition, optimization, weather forecasting, to name a few. ANN is a set of processing elements (neurons or perceptrons) with a specific topology of weighted =interconnections between these elements and a learning law for updating the weights of interconnection between two neurons. To respond to the increased demand of system identification and forecasting with large set of data, many different ANNs structures and learning rules, supervised, or unsupervised, have been proposed to meet various needs as robustness and stability. The FMP networks have been shown to obtain successful results in system identification and control [1, 2]. The Lyapunov function approach was used to obtain stability analysis of Backpropagation training algorithm of such network in [3]. The major drawback of the FMP is that it requires large number of input datafor training to achieve sufficient performance. Recurrent neural networks have been shown successful in identification of time varying systems along with the stability analysis in [4, 5]. However, the training process can be very sensitive to initial conditions such as number of neurons, the number of layers, and value of weights, and learning factors which are often chosen by trial and error. This paper presents a modified FMP architecture which inserts a dynamic filtering capability, ARMA local feedback at each neuron in the FMP structure. The Backpropagation algorithm is used for learning – weight adjusting. Stability analysis will be derived using Lyapunov function. It turns out that the learning factor must be within a range of values in order to guarantee the convergence of the algorithm. In the simulation, instead of selecting a learning factor by trial and error, authors define an adaptive learning factor which satisfies the convergence condition and adjust connection weight accordingly. The simulation results are presented to demonstrate the performance.

II. BASIC PRINCIPLE OF ARMA-LFMP NETWORK

An identification problem can be outlined as it is shown in Figure 1. A set of data is collected from the PLANT: input data and corresponding output data observed, or measured as target output of the identification problem. The set is often called "training set". A neural network model with parameters, called weights, is designed to simulate the PLANT. When the output from the neural network is calculated, an error between the network output and the target output is generated. The learning process of neural network is to modify the network to minimize the error.

Consider a system with ***n*** inputs $X = \{X_1, \dots ,$ and ***m*** output unitsY = $\{Y_1, \dots ,$. In a typical neuron, called perceptron, the output Y is expressed as:

$$Y = F(Z) = \frac{1 - e^{-\theta Z}}{1 + e^{-\theta Z}}; \quad Z = \sum_{i=1}^{n} W \qquad (1)$$

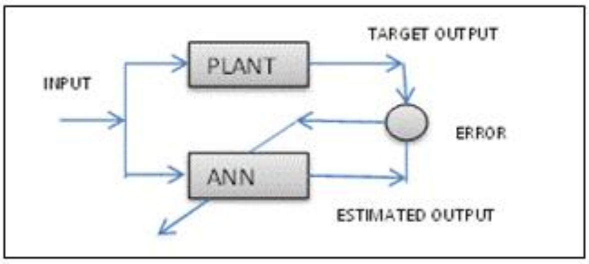

Fig.1 Outline of ANN identification problem

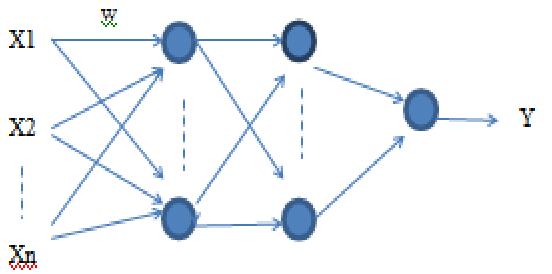

Fig.2Feedforward multi-layer network

where is called connection weight from input ; F is a nonlinear sigmoid function with a constant number θ, call it slope.

A FMP network combines number of neurons, called nodes, feed forward to next layer of nodes, illustrated in Figure 2. Suppose is the number of nodes in the *l*th layer, each output from the*l*-1th layer will be used as input for the next layer, that can be expressed as:

$$X_j^l = F(Z_j); Z_j = \sum_{i=1}^{N_{l-1}} W_{i,j}^l X_i^{l-1}; \; j = 1, 2, \ldots N_l \qquad (2)$$

, where $l = 1, ..; L$ is the number of layers in the network and l is the connection weight from the *ith* node in *l*-1 layer to the *jth* node in *l* layer.

In this structure in Figure 3, an ARMA model is inserted at each node in the Local feedback FMP network. The outputs from ARMA model are used as inputs to the FMP neural network node. The output at the *j*-th node in the *l*-th layer with an ARMA model is expressed as:

$$X_j^l(p) = F\left(Z_j^l(p)\right) \qquad (3)$$

and

$$Z_j^l(p) = \sum_{i=1}^{N_{l-1}} X_{ij}^{*l}(p) + \sum_{d=1}^{Dv} v_{jd}^l X_j^l(p - \qquad (4)$$

where

$$X_{ij}^{*l}(p) = \sum_{d=1}^{DA} a_{ijd}^l X_{ij}^{*l}(p-d) + \sum_{d=1}^{DB} b_{ijd}^l X_i^{l-1}(p-d) \qquad (5)$$

$j = 1, \ldots N_l; p = 1, ..$ andT is the number of patterns of the data set. X represents the output of nonlinear sigmoid function for the hidden layer and output layer, and is also used as an input to ARMA model; represents the output of ARMA model; a and b are connection weights of the ARMA, ν is weight of local feedback at each node and DA and DB are number of delays for AR and MA processes respectively.

The back-propagation algorithm has become the standard algorithm used for training feed-forward multilayer perceptrons. It is a generalized the Least Mean Square algorithm that minimizes the mean squared error between the target output and the network output with respect to the weights. The algorithm looks for the minimum of the error

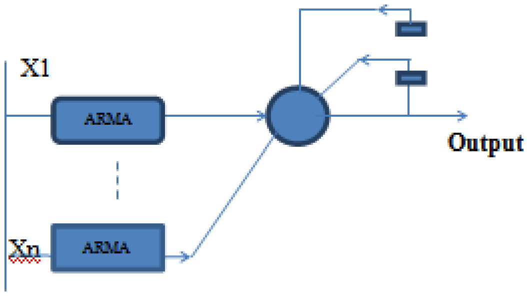

Fig. 3 One node of ARMA-LFMP model

function in the weight space using the method of gradient descent. The combination of weights which minimizes the error function is considered to be a solution of the learning problem. A proof of the Backpropagation algorithm was presented in [6] based on a graphical approach in which the algorithm reduces to a graph labeling problem.

The total error E of the network over all training set is defined as

$$E = \frac{1}{T} \sum_{k=1}^{N_L} \sum_{p=1}^{T} e_k^2(p) \qquad (6)$$

where e_k^2 is the error associated with pth pattern at the kth node of output layer,

$$e_k^2(p) = (d_k(p) - Y_k^L(p))^2 \; (7)$$

where d_k is the target at kth node and Y_k^L is the output of network at the kth node.

The network connection weights a_{ijd}^l, b_{ijd}^l, and v between neurons i in layer l-1 and neuron j in Layer l (l = 1, …, L) are updated iteratively by the Gradient Descent Rule

$$\Delta a_{ijd}^l = -\mu \frac{\partial E}{\partial a_{ijd}^l} \qquad (8)$$

$$\Delta b_{ijd}^l = -\mu \frac{\partial E}{\partial b_{ijd}^l} \qquad (9)$$

$$\Delta v_{jd}^l = -\mu \frac{\partial E}{\partial v_{jd}^l} \; (10)$$

where μ is the learning factor. Substituting (7) into (8), (9) and (10), the above updating equation can be expressed as follow:

$$\Delta a_{ijd}^l = -\frac{2\mu}{T} \sum_{k=1}^{N_L} \sum_{p=1}^{T} e_k^l(p) \frac{\partial X_k^l(p)}{\partial a_{ijd}^l} \qquad (8)$$

$$\Delta b_{ijd}^l = -\frac{2\mu}{T} \sum_{k=1}^{N_L} \sum_{p=1}^{T} e_k^l(p) \frac{\partial X_k^l(p)}{\partial b_{ijd}^l} \qquad (9)$$

$$\Delta v_{jd}^l = -\frac{2\mu}{T} \sum_{k=1}^{N_L} \sum_{p=1}^{T} e_k^l(p) \frac{\partial X_k^l(p)}{\partial v_{jd}^l} \; (10)$$

The rate of change of an output from k-th node of *l*-th layer with respect to connection weights a, b,and ν in *l*th layer can be expressed as:

$$\frac{\partial x_k^l(p)}{\partial v_{jd}^l} = \begin{cases} F'() \left[X_k^l(p) + \sum_{t=1}^{DV} v_{kj}^l \frac{\partial x_k^l(p-t)}{\partial v_{jd}^l}\right] & k = j \\ 0 & k \neq i \end{cases} \quad (11)$$

$$\frac{\partial x_k^l(p)}{\partial a_{ijd}^l} = \begin{cases} F'(Z_j^l(p)) \left[\frac{\partial x_{ij}^{*l}(p)}{\partial a_{sjd}^l} + \sum_{t=1}^{DV} v_{kj}^l \frac{\partial x_k^l(p-t)}{\partial a_{ijd}^l}\right] & \\ 0 & k \neq j \end{cases} \quad (12)$$

Where

$$\frac{\partial x_{ij}^{*l}(p)}{\partial a_{ijd}^l} = X_{ij}^{*l}(p-d) + \sum_{t=1}^{DA} a_{ijt}^l \frac{\partial x_{ij}^{*l}(}{\partial a_i^l} \quad (13)$$

$$\frac{\partial x_k^l(p)}{\partial b_{ijd}^l} = \begin{cases} F'(\ ) \left[\frac{\partial x_{ij}^{*l}(p)}{\partial b_{ijd}^l} + \sum_{t=1}^{DV} v_{jd}^l \frac{\partial x_k^l(p-t)}{\partial b_{ijd}^l}\right] & \\ 0 & k \neq i \end{cases} \quad (14)$$

where

$$\frac{\partial x_{ij}^{*l}(p)}{\partial b_{ijd}^l} = \sum_{t=1}^{DA} a_{ijd}^l \frac{\partial x_{ij}^{*l}(p-t)}{\partial b_{ijd}^l} + X_j^{l-1}(p- \quad (15)$$

Further calculation leads to the expressions of $\frac{\partial x}{\partial a}$, $\frac{\partial x}{\partial b}$, and $\frac{\partial x}{\partial v}$, rate of change of an output from *k-th* node of *l-th*layer with respect to connection weights a, b, and v in layer *l-n* for n < l:

$$\frac{\partial x_k^l(p)}{\partial a_{ijd}^{l-n}} = F'\left(Z_j^l(p)\right)\left[\sum_{t=1}^{N_{l-1}} \frac{\partial x_{tk}^{*l}(p)}{\partial a_{ijd}^{l-n}} + \sum_{s=1}^{DV} v_{kj}^l \frac{\partial x_k^l(p-s)}{\partial a_{ijd}^{l-n}}\right] \quad (16)$$

$$\frac{\partial x_k^l(p)}{\partial b_{ijd}^{l-n}} = F'\left(Z_j^l(p)\right)\left[\sum_{t=1}^{N_{l-1}} \frac{\partial x_{tk}^{*l}(p)}{\partial b_{ijd}^{l-n}} + \sum_{s=1}^{DV} v_{kj}^l \frac{\partial x_k^l(p-s)}{\partial b_{ijd}^{l-n}}\right] \quad (17)$$

$$\frac{\partial x_k^l(p)}{\partial v_{jd}^{l-n}} = F'\left(Z_j^l(p)\right)\left[\sum_{t=1}^{N_{l-1}} \frac{\partial x_{tk}^{*l}(p)}{\partial v_{jd}^{l-n}} + \sum_{s=1}^{DV} v_{kj}^l \frac{\partial x_k^l(p-s)}{\partial v_{jd}^{l-n}}\right] \quad (18)$$

where

$$\frac{\partial x_{tk}^{*l}(p)}{\partial a_{ijd}^{l-n}} = \sum_{s=1}^{DA} a_{tkj}^l \frac{\partial x_{tk}^{*l}(p-s)}{\partial a_{ijd}^{l-n}} + \sum_{s=0}^{DB} b_{tks}^l \frac{\partial x_t^{l-1}(p-s)}{\partial a_{ijd}^{l-n}} \quad (19)$$

$$\frac{\partial x_{tk}^{*l}(p)}{\partial b_{ijd}^{l-n}} = \sum_{s=1}^{DA} a_{tkj}^l \frac{\partial x_{tk}^{*l}(p-s)}{\partial b_{ijd}^{l-n}} + \sum_{s=0}^{DB} b_{tks}^l \frac{\partial x_t^{l-1}(p-s)}{\partial b_{ijd}^{l-n}} \quad (20)$$

$$\frac{\partial x_{tk}^{*l}(p)}{\partial v_{jd}^{l-n}} = \sum_{s=1}^{DA} a_{tkj}^l \frac{\partial x_{tk}^{*l}(p-s)}{\partial v_{jd}^{l-n}} + \sum_{s=0}^{DB} b_{tks}^l \frac{\partial x_t^{l-1}(p-s)}{\partial v_{jd}^{l-n}} \quad (21)$$

III. STABILITY ANAYLSIS

Stability for nonlinear systems refers to the stability of a particular solution. There may be one solution which is stable and another which is not stable. There are no inclusive general concepts of stability for nonlinear systems. The behavior of a system may depend drastically on the inputs and the disturbances. However, Lyapunov developed a theory to examine the stability of nonlinear systems.

The definition of Lyapunov function and Lyapunov theorem are quoted below [7]:

Definition 1 (Lyapunov function): A scalar function V(x) is a Lyapunov function for the system

$$x(t+1) = f\left(x(t)\right), \quad (22)$$

if the following conditions hold:

1. $V(0) = 0$ and $V(x)$is continuous in x
2. $V(x)$ is positive definite, that is,$V(x) \geq 0$with$V(x) = 0$only if$x = 0$
3. $\Delta V(x) = V(f(x(t)) - V(x(t))$ is negative definite, that is, $V(f(x(t)) - V(x(t)) \leq 0$ with$\Delta V(x) = 0$only if$x = 0$;

Theorem 1(Lyapunov Theorem): The solution for the system given by (11) is asymptotically stable if there exists a Lyapunov function in x.

The stability of the learning process in an identification approach leads to a better modeling and a guaranteed reached performance. According to Lyapunov theorem, the determination of stability depends on the selection and verification of a positive definite function. For the systems defined in (3) – (5), assume that the Backpropagation learning rule is applied and the error function and weights updating rule are defined in (6) – (10), then define

$$V(t) = \frac{1}{N_L T}\sum_{j=1}^{N_L}\sum_{p=1}^{T} e_k^2(p) \quad (23)$$

The proof is given in the following theorem that V(t) satisfies the Lyapunov condition.

Theorem 2:Assume that the nonlinear function F( ) is continuous and differentiable, the ARMA-LFMP is defined in (3)-(5), rewrite the learning rule (8) - (10) in weights vector form as:

$$\Delta A^l = -\frac{2\mu}{N_L T}\sum_{k=1}^{N_l}\sum_{p=1}^{T} e_k^l(p)\frac{\partial X_k^l(p)}{\partial A^l} \quad (24)$$

$$\Delta B^l = -\frac{2\mu}{N_L T}\sum_{k=1}^{N_l}\sum_{p=1}^{T} e_k^l(p)\frac{\partial x}{\partial} \quad (25)$$

$$\Delta v^l = -\frac{2\mu}{N_L T}\sum_{k=1}^{N_L}\sum_{p=1}^{T} e_k^l(p)\frac{\partial x}{\partial} \quad (26)$$

where , A^l, *and* are weight vectors in *lth* layer, then the system is stable under the condition:

$$\mu < \frac{TN_L}{6\sum_{k=1}^{N_L}\sum_{p=1}^{T}\sum_{l=1}^{L}\left[\left\|\frac{\partial Y_{pj}}{\partial A^l}\right\|^2 + \left\|\frac{\partial Y_{pj}}{\partial B^l}\right\|^2 + \left\|\frac{\partial Y_{pj}}{\partial v^l}\right\|^2\right]} \quad (27)$$

Proof: Assume that the Lyapunov function is defined in (23), calculation of ΔV leads to:

$$\Delta V(t) = V(t+1) - V(t)$$

$$= \frac{1}{N_L T}\sum_{j=1}^{N_L}\sum_{p=1}^{T}\left[e_{pj}^2(t+1) - e_{pj}^2(t)\right]$$

$$= \frac{1}{N_L T}\sum_{j=1}^{N_L}\sum_{p=1}^{T}\left[\left(e_{pj}(t) + \Delta e_{pj}(t)\right)^2 - e_{pj}^2(t)\right]$$

$$= \frac{1}{N_L T}\sum_{j=1}^{N_L}\sum_{p=1}^{T}\left[2e_{pj}(t)\Delta e_{pj}(t) + \left(\Delta e_{pj}(t)\right)^2\right] \quad (28)$$

Apply the first order Taylor expansion of with respect to weight vectors ,

$$\Delta e_{pj}(t) = \sum_{l=1}^{L}\left[\frac{\partial e_{pj}}{\partial A^l}\cdot\Delta A^l + \frac{\partial e_{pj}}{\partial B^l}\cdot\Delta B^l + \frac{\partial e_{pj}}{\partial v^l}\cdot\Delta v^l\right] \quad (29)$$

Substitute (24), (25), and (26) into (29),

$$\Delta e_{pj}(t) = \frac{-2\mu}{N_L T}\sum_{l=1}^{L}\left[\frac{\partial e_{pj}}{\partial A^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial A^l} + \frac{\partial e_{pj}}{\partial B^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial B^l} + \frac{\partial e_{pj}}{\partial v^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial v^l}\right] \quad (30)$$

Then, substitute the (30) into (28),

$$\Delta V(t) = \frac{-2\mu}{N_L T}\sum_{j=1}^{N_L}\sum_{p=1}^{T} e_{pj}\sum_{l=1}^{L}\left[\frac{\partial e_{pj}}{\partial A^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial A^l} + \frac{\partial e_{pj}}{\partial B^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial B^l} + \frac{\partial e_{pj}}{\partial v^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial v^l}\right] + \frac{4\mu^2}{(N_L T)^2}\sum_{j=1}^{N_L}\sum_{p=1}^{T}\left[\sum_{l=1}^{L}\left(\frac{\partial e_{pj}}{\partial A^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial A^l} + \frac{\partial e_{pj}}{\partial B^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial B^l} + \frac{\partial e_{pj}}{\partial v^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial v^l}\right)\right]^2 \quad (31)$$

Apply the algebraic rule $(a+b+c)^2 \le 3(a^2+b^2+$ into the second term of (31)

$$\Delta V(t) \le \frac{-2\mu}{N_L T}\sum_{j=1}^{N_L}\sum_{l=1}^{L}\left[\left(\sum_{p=1}^{T} e_{pj}\frac{\partial e_{pj}}{\partial A^l}\right)^2 + \left(\sum_{p=1}^{T} e_{pj}\frac{\partial e_{pj}}{\partial B^l}\right)^2 + \left(\sum_{p=1}^{T} e_{pj}\frac{\partial e_{pj}}{\partial v^l}\right)^2\right] + \frac{12\mu^2}{(N_L T)^2}\sum_{j=1}^{N_L}\sum_{p=1}^{T}\sum_{l=1}^{L}\left[\left(\frac{\partial e_{pj}}{\partial A^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial A^l}\right)^2 + \left(\frac{\partial e_{pj}}{\partial B^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial B^l}\right)^2 + \left(\frac{\partial e_{pj}}{\partial v^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial v^l}\right)^2\right] \quad (32)$$

For simplicity, let

$$\alpha = \sum_{j=1}^{N_L}\sum_{l=1}^{L}\left[\left(\sum_{p=1}^{T} e_{pj}\frac{\partial e_{pj}}{\partial A^l}\right)^2 + \left(\sum_{p=1}^{T} e_{pj}\frac{\partial e_{pj}}{\partial B^l}\right)^2 + \left(\sum_{p=1}^{T} e_{pj}\frac{\partial e_{pj}}{\partial v^l}\right)^2\right] \quad (33)$$

Then apply it into (32) and consider that

$\left(\frac{\partial e_{pj}}{\partial A^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial A^l}\right)^2$,
$\left(\frac{\partial e_{pj}}{\partial B^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial B^l}\right)^2$, and
$\left(\frac{\partial e_{pj}}{\partial v^l}\cdot\sum_{s=1}^{T} e_{sj}\frac{\partial e_{sj}}{\partial v^l}\right)^2$, then

$$\Delta V(t) \le \frac{-2\mu\alpha}{N_L T} + \frac{12\mu^2}{(N_L T)^2}\sum_{j=1}^{N_L}\sum_{l=1}^{L}\sum_{p=1}^{T}\left[\left|\frac{\partial e_{pj}}{\partial A^l}\right|^2\alpha + \left|\frac{\partial e_{pj}}{\partial B^l}\right|^2\alpha + \left|\frac{\partial e_{pj}}{\partial v^l}\right|^2\alpha\right] \quad (34)$$

To ensure that the function V satisfies the Lyapunov condition,

Let the right-hand side of (34) be less than zero, and consider that

$\left\|\frac{\partial e_{pj}}{\partial A^l}\right\| = \left\|\frac{\partial Y}{\partial}\right.$, $\left\|\frac{\partial e_{pj}}{\partial B^l}\right\| = \left\|\frac{\partial Y}{\partial}\right.$, and $\left\|\frac{\partial e_{pj}}{\partial v^l}\right\| = \left\|\frac{\partial Y}{\partial}\right.$

then we obtain the condition

$$\mu < \frac{N_L T}{6\sum_{j=1}^{N_L}\sum_{l=1}^{L}\sum_{p=1}^{T}\left[\left|\frac{\partial Y_{pj}}{\partial A^l}\right|^2 + \left|\frac{\partial Y_{pj}}{\partial B^l}\right|^2 + \left|\frac{\partial Y}{\partial v}\right.\right.} \quad (35)$$

Therefore, the ARMA-LFMP system defined in (3)-(5) is stable when the learning factor in Backpropagation learning rule described in (8) –(10) satisfies the condition (35).

For purpose of simplifying the simulation, instead of calculating all , , and for l = 1, … L; j = 1, … , the following corollary will identify an upper bound of

$\sum_{j=1}^{N_L}\sum_{l=1}^{L}\sum_{p=1}^{T}\left[\left|\frac{\partial Y_{pj}}{\partial A^l}\right|^2 + \left|\frac{\partial Y_{pj}}{\partial B^l}\right|^2 + \left|\frac{\partial Y_p}{\partial v^l}\right.\right.$, then replace the denominator of the (35) by the upper bound that provides a more restrictive but an easier calculated condition.

Consider the infinite norm notation for any vector $X = \{x_1, x_2, \dots,$ that $\|X\| = \max_{1<i<n}|$, for simplicity, we use notation | in this paper representing $\|X\|$)

First, for the output layer *L*, apply infinite norm in (11) and the notation $|v_j^l| = \sum_{d=1}^{Dv}$, calculation leads to

$$\left\|\frac{\partial Y}{\partial v^L}\right\| \le \frac{\theta}{2}\left[1 + |v_j^l|\left\|\frac{\partial Y}{\partial v^L}\right\|\right]$$

and then

$$\left\|\frac{\partial Y}{\partial v^L}\right\| \le \frac{\theta}{(2-\theta} \quad (36)$$

From (5),

$$\|X_{ij}^{*l}\| \le \sum_{d=1}^{DA} a_{ijd}^l\,\|X_{ij}^{*l}\| + \sum_{d=1}^{DB} b_{ijd}^l$$

$$\|X_{ij}^{*l}\| \le \frac{\sum_{d=1}^{DB} b_{i}^{l}}{(1-\sum_{d=1}^{DA} a} \quad (37)$$

let $|A_{ij}| = \sum_{d=1}^{DA} i$, $|B_{ij}| = \sum_{d=1}^{DA} i$, and $\beta = \frac{|B_i}{(1-|}$ then from (13)

$$\left\|\frac{\partial x_{ij}^{*L}}{\partial a_{ijd}^{L}}\right\| \le \beta + |A_{ij}| \left\|\frac{\partial x}{\partial a}\right\|,$$

$$\left\|\frac{\partial x_{ij}^{*L}}{\partial a_{ijd}^{L}}\right\| \le \frac{\beta}{(1-|A_{ij}|)} = \frac{|B_i}{(1-|A} \quad (38)$$

Apply(38) into (12)

$$\left\|\frac{\partial X_j^L}{\partial a_{ijd}^L}\right\| \le \frac{\theta}{2}\left[\frac{\beta}{(1-|A_{ij}|)} + |v_j| \left\|\frac{\partial X_j^L}{\partial a_{ijd}^L}\right\|\right]$$

$$\left\|\frac{\partial x_j^L}{\partial a_{ijd}^L}\right\| \le \frac{\theta\beta}{(2-\theta|v_j|)(1-|} \quad (39)$$

From (15),

$$\left\|\frac{\partial X_{ij}^{*L}}{\partial b_{ijd}^L}\right\| \le |A_{ij}| \left\|\frac{\partial X_{ij}^L}{\partial b_{ijd}^L}\right\| + 1$$

$$\left\|\frac{\partial x_{ij}^{*L}}{\partial b_{ijd}^L}\right\| \le {}^{1}\!/(1-|A \quad (40)$$

Apply(40) into (14)

$$\left\|\frac{\partial X_j^L}{\partial b_{ijd}^L}\right\| \le \frac{\theta}{2}\left[\frac{1}{(1-|A_{ij}|)} + |v_j| \left\|\frac{\partial X_j^L}{\partial b_{ijd}^L}\right\|\right]$$

$$\left\|\frac{\partial x_j^L}{\partial b_{ijd}^L}\right\| \le \frac{\theta}{(2-\theta|v_j|)(1-|} \quad (41)$$

Without loss of generality, for change of rate of layer l with respect to *l-n*,we derive for n = 1 first. From (19)

$$\left\|\frac{\partial x_{tk}^{*l}}{\partial a_{ijd}^{l-1}}\right\| \le |A_{ij}| \left\|\frac{\partial x_{tk}^{*l}}{\partial a_{ijd}^{l-1}}\right\| + \frac{\theta\beta}{(2-\theta|v_j|)(1-|} \quad (42)$$

$$\left\|\frac{\partial x_{tk}^{*l}}{\partial a_{ijd}^{l-1}}\right\| \le \frac{\theta\beta}{(2-\theta|v_j|)(1-|A} \quad (43)$$

Apply (43) into (16),

$$\left\|\frac{\partial X_k^l}{\partial a_{ijd}^{l-1}}\right\| \le \frac{\theta}{2}\left[\sum_{t=1}^{N_{l-1}} \frac{\theta\beta}{(2-\theta|v_j|)(1-|A_{ij}|)^2} + |v_j| \left\|\frac{\partial X_j^l}{\partial a_{ijd}^{l-1}}\right\|\right]$$

$$\left\|\frac{\partial x_k^l}{\partial a_{ijd}^{l-1}}\right\| \le \frac{\theta^2\beta N_{l-1}}{(2-\theta|v_j|)^2(1-|A} \quad (44)$$

Similarly, apply norm in (20), and then apply into (17),

$$\left\|\frac{\partial x_k^l}{\partial b_{ijd}^{l-1}}\right\| \le \frac{\theta|B_{ij}|N_{l-1}}{(2-\theta|v_j|)(1-|A_{ij}^l|)(1-|A_i^l} \quad (45)$$

Apply norm in (21), and then apply into (18),

$$\left\|\frac{\partial x_k^l}{\partial v_{jd}^{l-1}}\right\| \le \frac{\theta^2|B_{ij}|N_{l-1}}{(2-\theta|v_j^l|)(2-\theta|v_j^{l-1}|)(1-|A_{ij}^l|)(1-|A_i^l} \quad (46)$$

Inductively, for any layer L-k< L, let

$$\Delta = \prod_{i=0}^{k}(2-\theta|v_j^{L-i}|)(1-|A_{ij}^{L}|,$$

$$\left\|\frac{\partial Y_k^L}{\partial v_{jd}^{L-k}}\right\| \le \frac{\theta^{k+1}|B_{ij}|^k}{\Delta} \quad (47)$$

$$\left\|\frac{\partial Y_k^L}{\partial a_{jd}^{L-k}}\right\| \le \frac{\beta\,\theta^{k+1}|B_{ij}|^k}{\Delta} \quad (48)$$

$$\left\|\frac{\partial Y_k^L}{\partial b_{jd}^{L-k}}\right\| \le \frac{\theta^{k+1}|B_{ij}|^k}{\Delta} \quad (49)$$

Apply (47), (48) and (49) into condition (35), we obtain a more restrictive condition as follows:

$$\mu \le \frac{N_L}{6\sum_{k=1}^{L}\sum_{j=1}^{N_L}\frac{(2+\beta)\theta^{k+1}|B_{ij}|^k}{\Delta}} \quad (50)$$

Corollary 1: The ARMA-LFMP system converges if the following conditions are satisfied:

$$2-\theta|v_j| > 0, \quad (51)$$

$$(1-|A_{ij}|) > 0, \quad (52)$$

and $\mu \le \frac{N_L}{6\sum_{k=1}^{L}\sum_{j=1}^{N_L}\frac{(2+\beta)\theta^{k+1}|B_{ij}|^k}{\Delta}}$.

IV. SIMULATION

In this section, an example of chaotic system known as Henon system is considered to demonstrate the effectiveness of developed methods in this paper. The **Hénon map** is a discrete-time dynamical system described as:

$$X(k+1) = 1 - a\,X^2(k) + Y(k)$$
$$Y(k+1) = bX(k)$$

It is one of the most studied examples of dynamical systems that exhibit chaotic behavior. The Hénon map takes a point (*x(k)*, *y(k)*) in the plane and maps it to a new point.The map depends on two parameters, *a* and *b*, which for the **classical Hénon map** have values of *a* = 1.4 and *b* = 0.3. For the classical values the Hénon map is chaotic.

In this simulation, consider the system

$$X(k+1) = 1 - 1.4\,X^2(k) + Y(k)$$
$$Y(k+1) = 0.3X(k)$$

and a three-layer neural network structure was selected for two inputs and two output with number of nodes as 5, 5 and 2 in layer 1, 2 and 3 respectively. 100 patterns of data were generated and used for learning. After number of trial and error attempts, with slope set as 0.6 and learning factor set as constant .01, and random generated initial weights, the system reached to absolute error 0.0899999 after 2177311 iterations.

With adaptive learning factor, it took 373317 number of iteration to reach the same threshold of 0.089999. It is also observed that the error decreases steadily while the adaptive learning factor is applied and the oscillation of error was observed while a predefined constant learning factor is applied.

The Figure 4 demonstrated 100 patterns of data generated from Henon system comparing with simulated data from the neural network described above.

The adaptive learning factor guarantees that the errors will steadily decrease. The drawback is that it increased calculation since an updated learning factor needs to be calculated at every weights update based on Backpropagation algorithm. Applying the constant learning factor avoids the calculation, but a proper constant learning factor has to be identified through trial and error. In the Figure 5, the plot demonstrated comparison of

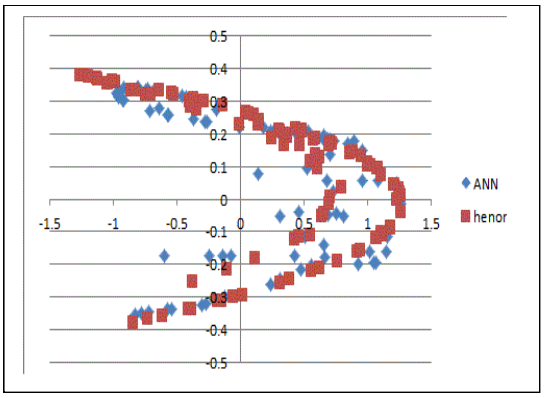

Figure 4. ANN Simulation of Henon System

errors from learning process with constant learning factor and the adaptive learning factor. The constant learning factor was selected with value of 0.1. To compare the effectiveness of adaptive learning method, the 0.1 was used as initial learning factor in the adaptive method. Points for the plot were taken from the errors of every 1000th of iteration.

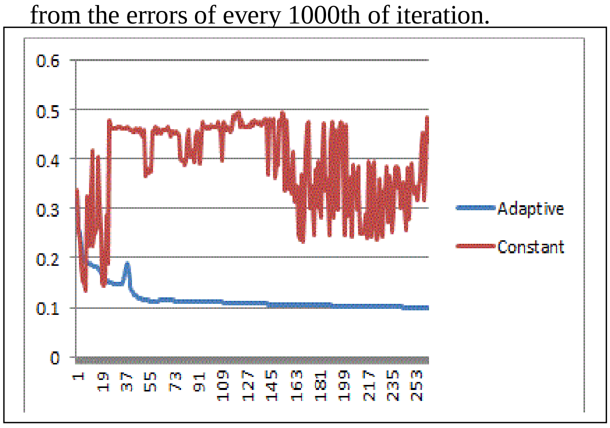

Figure 5. Comparison of errors from learning with adaptive and constant learning factor

REFERENCES

[1] R. Raeisi & A. Kaur, *Artificial Neural Networks for Realization and Verification of Digital Logic Circuits. International Journal of Modern Engineering, Spring/Summer 12(2), 5-9, 2012.*

[2] W. Yu, A. S. Poznyak, and X. Li, *Multilayer dynamic neural networks for no-linear system on-line identification,International Journal of Control, 7(18). 1858-1864, 2001.*

[3] T. Korkobi, M. Djemel, and M. Ctourou, *Stability analysis of neural netorks-based system identification, Modeling and Simulation in Engineering, Volume 2008, Article ID 343940*

[4] E. Zhu. *Stability Analysis of Recurrent Neural Networks with Random Delay and Markovian Switching, Journal of Inequalities and Applications 2010*

[5] H. Akça, R. Alassar, and V. Covachev, *Stability of neural networks with time varying delays in the presence of impulses, Advances in Dynamical Systems and Applications 1(1), pp. 1-15, 2006*

[6] R. Rojas, *Neural Networks*, Springer-Verlag, 1996

[7] Sontag, E.D., *Mathematical Control Theory*, Springer-Verlag, New York

[8] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.