2012 IEEE International Conference on Systems, Man, and Cybernetics
October 14-17, 2012, COEX, Seoul, Korea

Lyapunov Learning Algorithm for Quasi-ARX Neural Network to Identification of Nonlinear Dynamical System

Mohammad Abu Jami`in, Imam Sutrisno and Jinglu Hu
Graduate School of Information Production and Systems
Waseda University
Hibikino 2-7, Wakamatsu-ku, Kitakyushu-shi, Fukuoka, Japan
e-mail: mohammad@ruri.waseda.jp, imams3jpg@moegi.waseda.jp, jinglu@waseda.jp

***Abstract*— In this note, we present the modeling of nonlinear dynamical systems with Quasi-ARX neural network using Lyapunov algorithm in learning process. This work exploits the idea on learning algorithm in nonlinear kernel part of Quasi-ARX model to improve stability and fast convergence of error. The proposed algorithm is then employed to model and predict a classical nonlinear system with input dead zone and nonlinear dynamic systems, exhibiting the effectiveness of proposed algorithm. Based on the result of simulation, the proposed algorithm can make the error in process learning become fast convergence, ultimately bounded, and the error distributed uniformly.**

Keywords-Neural network; Quasi-ARX model; nonlinear; modeling; Lyapunov; stability.

I. INTRODUCTION

The modeling of the nonlinear system become very popular and much interest in system control community in the recent years. It is important and an indispensable to be applied in controller design based on predictive output of nonlinear systems modeling [1,2]. Many approaches have been proposed and developed to identify and predict of nonlinear dynamical systems using neural network in different model structure and algorithm.

Consider the fact that the system identifications are always applied in certain application, and a number of them using nonlinear black box-models. As we know, nonlinear black-box model have been criticized not user friendly since they neglect some good properties such as linear structure and simplicity [1, 3, 4]. Especially, the linear structure is very useful and more favorable to certain applications such as nonlinear system control and fault diagnosis [5, 6].

Because of simplicity in structure and more favorable using linear properties in nonlinear model system, for this case, we use Quasi-ARX model for mapping nonlinear dynamical systems. The form of Quasi-ARX model equation can be achieved through Taylor expansion series. The Quasi-ARX model is a linear model with nonlinear coefficient. The coefficient reflected nonlinear form parameter to input vector, so the Quasi-ARX model can be divided into two sub models. Linear part or macro part reflected the correlation of nonlinear parameter to input vector, and kernel part reflected to nonlinear parameter [1,3-7].

Many learning algorithm to get best model have been used to train kernel of nonlinear part because the general problem of Quasi-ARX model be focused on nonlinear kernel part. The macro part with linear parameter characteristic generally uses the method of gradient learning least square error algorithm [1, 5, 7]. The kernel part with nonlinear form characteristic has specific structure, so it may be formed using the other structure like neural network and the structure of neural network can be trained with different method and algorithm.

The structure of neural network reflected of kernel part can be built by MLP structure [1, 5, 7]. The learning algorithm of MLP structure have been proposed using back propagation error [1 ,5 ,7], switching mechanism between linear and nonlinear part [8], MLP with wavelet neuron trained by support vector regression SVR and orthogonal least square (SVR-OLS) [4]. The purpose of kernel part learning is to get nonlinear parameter in vector input which it can make process modeling stable, accurate and fast convergence.

In this case, the structure of kernel part of Quasi-ARX model is MLP structure with Lyapunov learning algorithm. The Lyapunov learning algorithm offers two important advantages, those are fast convergence in learning process and the model error ultimately bounded [2]. The Lyapunov learning algorithm able to avoid local minima and achieve global minima, so it can increase accuracy of model [9]. The Lyapunov function be used to track convergence of error, so the output of tracking error can then asymptotically converge to zero [10], it can be used to analyze stability on identification process [11, 12], and it can be used as switching stability condition with multiple stability criteria [13].

Lyapunov function combining back-stepping approach can be used as adaptive controller to guarantee semi-global boundedness applied in nonlinear time delay [14], Lyapunov function be used as barrier function to ensure boundedness in control trajectory [15, 16], Lyapunov based on controller can make closed loop system globally stable [17]. Lyapunov method to be used to estimate asymptotic stable region in controller design by genetic algorithm [18], Lyapunov method with linear programming to be used to analyze stability and control synthesis of state feedback controller [19].

Aiming at developing a fast, accurate, and convergent in parameter optimization algorithm, Lyapunov function is

978-1-4673-1714-6/12/$31.00 ©2012 IEEE

applied as algorithm in Quasi-ARX neural network with MLP kernel to update its weight. The proposed method only has a few number of learning and the modeling error characteristic bounded and has uniform distribution. The output weight of parameter in kernel part of MLP can be convergent to the optimal weights, and it has near to zero variance of error output by 20 independent trials. Through the theoretical analysis and simulation experiments the model performance offer fast learning, stable bounded in error characteristic, and error distributed uniformly proved by variance error close to zero value.

II. Problem Description and Formulation

A. Modeling Description

Consider a single input single output (SISO) black box time invariant whose input output relation described by,

$$y(t) = g(\varphi(t)) + e(t)$$
$$\varphi(t) = [y(t-1),\ldots, y(t-n_y), u(t-1),\ldots, u(t-n_u)]^T \tag{1}$$

where $u(t) \in R$, $y(t) \in R$, $e(t) \in R$ are the system output, the system input, and stochastic noise of zero mean at time t (t = 1,2,3,…) respectively. $g(\cdot):R^{n=nu+ny} \rightarrow R$ is unknown continuous function (black-box) describing the dynamics of system under study, and $\varphi(t) \in R$ is the regression vector composed delays of the input – output data. The number of input variables n is equals to the sum of n_u and n_y. The noise of the system $e(t)$ added on the unknown function input output of the system.

THEOREM 1. The output of system has a linear series function between input spaces $\varphi(t)$ to the unknown nonlinear function of input space $\theta(\varphi(t))$.

Proof:

Through using Taylor expansion series, nonlinear continuous function $g(\cdot)$ can be described into Taylor equation,

$$y(t) = g(0) + g'(0)\varphi(t) + \frac{1}{2}\varphi^T(t)g''(0)\varphi(t) + \ldots + e(t) \tag{2}$$

the (2), can be decomposed into,

$$y(0) = g(0)$$

$$\theta(\varphi(t)) = \left(g'(0) + \frac{1}{2}\varphi^T(t)g''(0) + \ldots \right)^T$$
$$= [a_{1,t} \ldots a_{n,t} b_{0,t} \ldots b_{m-1,t}]^T \tag{3}$$

where the coefficient $a_{i,t}=a_i(\varphi(t))$ $(i=1,\ldots,n_y)$ is the vector regression in output, and coefficient $b_{j,t}=b_j(\varphi(t))$ $(j=1\ldots,n_u-1)$ is the vector regression in input. The (2) can be simplified as,

$$y(t) = y(0) + \varphi^T(t)\theta(\varphi(t)) + e(t) \tag{4}$$

The (4), can be rewrite in matrix equation as,

$$y(t) = \Phi^T \Psi \tag{5}$$

where,

$\psi = [y(0)\ \theta(\varphi(t))]$, and $\Phi = [1\ \varphi(t)]$.

ψ called as nonlinear parameter of kernel part, and Φ called as input variable space of macro part.

B. Quasi-ARX model

The form ARX model with nonlinear time function at its coefficient is depicted in two polynomials of numerator and denumerator as,

$$\frac{y(t+d,\phi(t))}{u(t+d,\phi(t))} = \frac{A(q^{-1},\phi(t))}{B(q^{-1},\phi(t))} = \frac{b_{0,t} + b_{1,t}q^{-1} + \ldots b_{m-1,t}q^{-m}}{1 - a_{1,t}q^{-1} - \ldots - a_{n,t}q^{-n}} \tag{6}$$

the operator q^{-k} represent to $q^{-k}y(t) = y(t-k)$, and

$$\phi(t) = [y(t),\ldots y(t-n_y+1)u(t),\ldots,u(t-n_u-d+2)].$$

The coefficient of $a_{i,t}$ and $b_{i,t}$ are nonlinear functions of a regression vector $\phi(t)$. By using Taylor expansion and other mathematical transformations, it is easy to show that the system (1) can be represented by an ARX macro-model,

$$y(t,\phi(t-d)).A(q^{-1},\phi(t-d)) =$$
$$g(0) + B(q^{-1},\phi(t-d))q^{-d}u(t) + e(t)$$

where $y(t,\phi(t-d))$ is the model output, for which $A(q^{-1},\phi(t))$ and $B(q^{-1},\phi(t))$ are commutable, e.g., $q^{-1}.A(q^{-1},\phi(t)) = A(q^{-1},\phi(t)).q^{-1}$, see [1] for more details.

THEOREM 2. For a system described by (7), the output predictor at one step ahead prediction $y_p(t+d|t,\phi(t))$, if y(t) by (5) satisfies.

We can see that (1) and (7) have similar characteristic in ARX equation model form. Aiming to get prediction output, the (5) can be rewrite as ARX macro model predictor as,

$$y_p(t+d) = \Phi_p^T \psi_p \tag{7}$$

where,

$\Phi_p = [1\ \phi(t)]$, and $\psi_p = [\,y_e\, \alpha_{0,t}, \ldots, \alpha_{n,t}\, \beta_{0,t}, \ldots, \beta_{m,t}]$

$n = n_y-1$, $m = n_u + d - 2$.

As a predictor the next input vector will be known, and then we estimate the next output of nonlinear system, so it is important to include the next input called virtual element to become input of MLP kernel part. With including of virtual element $x(t+d)$, the input vector of MLP kernel part is showed by (8).

$$\xi(t) = [y(t),\ldots y(t-n)u(t-1),\ldots,u(t-m)x(t+d)] \tag{8}$$

where, $\psi_p = \aleph(\xi(t),\Omega)$.

The (7) represent the macro model, show it has correlation

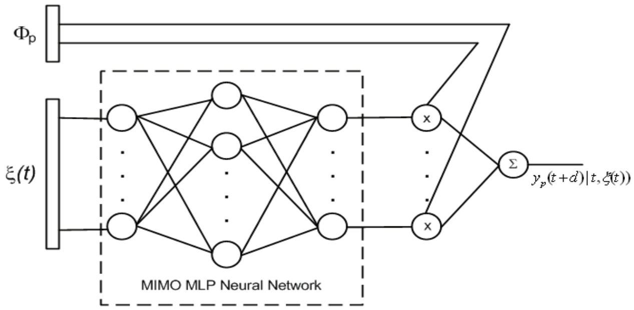

Figure 1. The Quasi-ARX prediction model with kernel part MLP neural network

between the input variable space vector Φp and the output of kernel part ψ_p, and the (8) as input space of kernel part $\xi(t)$. The Quasi-ARX with neural network structure using MLP kernel which represent (7) and (8) is showed by Fig. 1. The parameter of ψ_p are realized by MLP neural network MIMO, the Quasi-ARX prediction model is expressed by,

$$y_p(t+d \mid t, \xi(t)) = \Phi_p{}^T \aleph(\xi(t), \Omega) \tag{9}$$

$$\aleph(\xi(t), \Omega) = W_2 \Gamma W_1 \xi(t) + \theta \tag{10}$$

where, $\Omega = \{W_1, W_2, \theta\}$, $W_1 \in R^{Mxn}$, $W_2 \in R^{(n+1)xM}$ are the weight matrices of the first and second layer. $\theta \in R^{(n+1)x1}$ are the bias vector of output nodes, and Γ is the diagonal nonlinear operator with identical sigmoidal elements on hidden nodes. Substitute (9) and (10), (7) can be expressed by,

$$y_p(t+d \mid t, \xi(t)) = \Phi_p{}^T . W_2 \Gamma W_1(\xi(t)) + \Phi_p{}^T \theta \tag{11}$$

the equation (11) can be realized into Quasi-ARX neural network with MLP structure of kernel part.

C. Learning Algorithm

The prediction output will be satisfied if only the output of kernel part can be satisfied. The kernel of nonlinear part Quasi-ARX is MIMO MLP structure. Consider that the MLP kernel Quasi-ARX have n input node, m hidden node, so the output must have n+1 node. The input vector is $\xi(t)=[\xi_1, \xi_2, \ldots, \xi_n]$, the output of hidden node $h = [h_1\ h_2\ \ldots\ h_m]$, the output of MLP kernel $s = [s_1\ s_2\ \ldots\ s_{n+1}]$. The output at hidden node with nonlinear function in hidden layer can be expressed as,

$$h_j = f_j\left(\sum_{j=1}^{m} \sum_{i=1}^{n} \xi_i W_{1(i,j)}\right) = f_j(z) \tag{12}$$

where, f_j is the nonlinear sigmoid function at hidden node.

$$f(z) = \frac{1-e^{-z}}{1+e^{-z}} \tag{13}$$

The output of node with linear function, can be expressed as,

$$\aleph(\xi(t), \Omega) = \sum_{j=1}^{m} \sum_{r=1}^{n+1} h_j W_{2(j,r)} + \theta \tag{14}$$

where, $W_{1(i,j)}, W_{2(j,r)}, \theta$ are the weight matrices in first, second layer, and the bias vector of output node, $\aleph(\cdot,\cdot)$ is a MLP network with n input nodes.

Consider there are two sub models in Quasi-ARX neural network, those are linear part, and nonlinear kernel part the output of two sub model expressed by,

$$\text{SM1} \quad z_l = \Phi_p{}^T \theta \tag{15}$$

$$\text{SM2} \quad z_n = \Phi_p{}^T W_2 \Gamma(W_1 \xi(t)) \tag{16}$$

The output guidance of two sub models are expressed by,

$$z_l = y_p(t+d \mid t, \xi(t)) - \Phi_p{}^T W_2 \Gamma(W_1 \xi(t)) \tag{17}$$

$$z_n = y_p(t+d \mid t, \xi(t)) - \Phi_p{}^T \theta \tag{18}$$

There are two learning algorithm to update model parameter. The first is the linear part sub model will be identified using square error algorithm and the second is the nonlinear sub model or kernel part will be identified using Lyapunov function. There are assumptions made in our proposed algorithm.

Assumption 1. The pairs of the input and output training data are bounded, and the input is generated from random vector PRBS (pseudo random binary sequence)

Assumption 2. The Weight matrices and bias vector are bounded.

Assumption 3. *g(.)* at (1) is the continuous function.

Suppose that the assumption 2 is valid, $[W_1^*\ W_2^*\ \ \theta^*] = \Omega^*$ is the optimal solution, so $\aleph(\Omega^*, \xi(t))$= satisfies (11). We can update the model parameter $[W_1\ W_2\ \theta] = \Omega$. The learning algorithm based Lyapunov function are described as:

1) Set $\theta = 0$, and small initial values of $W_1\ W_2\ \theta$. k=1.

2) Calculate z_l, then estimate θ by model SM1 using LSE method.

3) Calculate error, $z_n = \aleph(\Omega^*, \xi(t))$

$$e(k) = \aleph(\Omega^*, \xi(t)) - \aleph(\Omega, \xi(t)) \tag{19}$$

where, k = the sequence of learning number, $e(k) = [e_1, e_2, \ldots e_{n+1}]$, $\aleph(\Omega, \xi(t)) = [\aleph_1, \aleph_2, \ldots, \aleph_{n+1}]$, $\aleph(\Omega^*, \xi(t)) = [\aleph_1^*, \aleph_2^*, \ldots, \aleph_{n+1}^*]$.

4) Choose Lyapunov function candidate, the candidate function is stated as $V(k) = f(e(k))$, where $V(k)=0$ only if $e(k)=0$, $V(k)>0$ only if $e(k) \neq 0$.

5) Update the weights of MLP neural network from output layer to input layer based on $\Delta V(k) = V(k) - V(k-1) < 0$. According the Lyapunov theory [2, 17], if $V(k)>0$ and $\Delta V(k)<0$, the error output will converge to zero at time goes to infinity.

$$\lim_{k \to \sim} e(k) = 0, \quad if\ \Delta V(k) < 0 \tag{20}$$

6) Stop if pre - specified condition is met, otherwise go to

step 2), $k=k+1$.

The weight matrices in first layer and second layer can be calculate based on Lyapunov function candidate are expressed as,

$$V(k) = \beta^k e^2(k) \tag{21}$$

where, β is positive constant and $\beta > 1$, k is the k sequence of learning number. The derivative of Lyapunov function is proportional to $\Delta V(k)$.

$$\begin{aligned}
\Delta V(k) &= V(k) - V(k-1) \\
&= \beta^k e^2(k) - \beta^{k-1} e^2(k-1) \\
&= \beta^k \left(\aleph(\Omega, \xi(t)) - \aleph(\Omega^*, \xi(t))\right)^2 - \beta^{k-1} e^2(k-1) \\
&= \beta^k \left(\sum_{r=1}^{n+1}\sum_{j=1}^{m} W_{2(j,r)} h_j(t) - \aleph(\Omega^*, \xi(t))\right)^2 - \beta^{k-1} e^2(k-1) \\
&= \beta^k \left(\sum_{r=1}^{n+1}\sum_{j=1}^{m} W_{2(j,r)} f_j\left(\sum_{j=1}^{m}\sum_{i=1}^{n} W_{1(i,j)} \xi_i\right) - \aleph(\Omega^*, \xi(t))\right)^2 \\
&\quad - \beta^{k-1} e^2(k-1) \\
&= \beta^k \left(\sum_{r=1}^{n+1}\sum_{j=1}^{m} W_{2(j,r)} f_j\left(\sum_{j=1}^{m}\sum_{i=1}^{n} \frac{1}{m\xi_i} f_j^{-1}(R)\right) - \aleph(\Omega^*, \xi(t))\right)^2 \\
&\quad - \beta^{k-1} e^2(k-1) \\
&= -(\beta^{k-1} - 1) e^2(k-1) < 0, \quad if \quad \beta > 1
\end{aligned} \tag{22}$$

where, $$R = \frac{\beta^{-k/2} e(k-1) + \aleph\left(\Omega^*, \xi(t)\right)}{m W_{2(j,r)}}$$

THEOREM 3. Based on Lyapunov function candidate stated as $V(k) = P.e^2(k)$ and P>1. if $V(k) > 0$, and $\Delta V(k) < 0$, where $\Delta V(k) = V(k)-V(k-1)$ for k=1, 2,… , the error e(k) will converge to zero at k goes to infinity.

LEMMA 1. For a Lyapunov function candidate $V(k) = \beta^k e^2(k)$, the error $e(k)$ will convergence at value,

$$e(k) = \beta^{-(\frac{1+k}{4})k} e(0) \tag{23}$$

proof:

$$\begin{aligned}
&\Delta V(k) = V(k) - V(k-1) \\
&-(\beta^{k-1} - 1) e^2(k-1) = \beta^k e^2(k) - \beta^{k-1} e^2(k-1) \\
&e(k) = \beta^{-k/2} e(k-1)
\end{aligned} \tag{24}$$

$$\begin{aligned}
&e(1) = \beta^{-1/2} e(0) \\
&e(2) = \beta^{-1} e(1) = \beta^{-(3/2)} e(0) = \beta^{-(1+2)2/4} e(0) \\
&e(3) = \beta^{-3/2} e(2) = \beta^{-12/4} e(0) = \beta^{-(1+3)3/4} e(0) \\
&\vdots \\
&e(k) = \beta^{-(1+k)k/4} e(0)
\end{aligned}$$

Based on LEMMA 1 and THEOREM 3, the weight of MLP kernel will be updated. Based on (14), we can inverse the function to get the second layer weight,

$$W_{2(j,r)} = \frac{\beta^{-k/2} e_r + \aleph_r^*}{m h_j(k-1)} \tag{25}$$

$$W_{1(i,j)} = f_j^{-1}\left(\sum_{r=1}^{n+1} \frac{\beta^{-k/2} e_r + \aleph_r^*}{m W_{2(j,r)}}\right) \Big/ m\xi_i \tag{26}$$

where, $h_j \neq 0$, $W_{2(j,r)} \neq 0$, and $\xi_i \neq 0$. The equation (25) and (26) can be modified to prevent singularities due the zero value incorporated of $h_j \neq 0$, $W_{2(j,r)} \neq 0$, $\xi_i \neq 0$. The modified update weight described by,

$$W_{1(j,i)} = f_j^{-1}\left(\sum_{r=1}^{n+1} \frac{\beta^{-k/2} e_r + \aleph_r^*}{m W_{2(j,r)} + \lambda_2}\right) \Big/ \left(m\xi_i + \lambda_3\right) \tag{27}$$

$$W_{2(j,r)} = \frac{\aleph_r^* - \aleph_r}{h_j} = \frac{\beta^{-k/2} e_r + \aleph_r^*}{m h_j(k-1) + \lambda_1} \tag{28}$$

III. PLANT UNDER STUDIES

The performance of the Quasi-ARX neural network with Lyapunov learning algorithm is verified and tested to the nonlinear dynamical system. Those are the nonlinear dynamic system and the nonlinear system with input dead zone modeling identification. The performance of network is measured using RMS root mean square error index expressed as,

$$RMS = \sqrt{1/N \sum_{t=1}^{N} \left(y^*(t) - y(t)\right)^2} \tag{29}$$

where k =1,2,…,N. The another performance measured with the average RMS (ARMS) in the independent trial,

$$ARMS = (1/NT) \sum_{j=1}^{NT} RMS(j) \tag{30}$$

where, N is the number of data pairs (training data), $y(k)$ and $y^*(k)$ reflecting the k[th] calculated output and system output, NT is the number of independent test trial.

A. The Plant of a Nonlinear Dynamic System

Consider the nonlinear system dynamic system is stated as,

$$y(t) = f(y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)) + e(t)$$

Where,

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1, x_2, x_3, x_5(x_3 - 1) + x_4}{1 + x_2^2 + x_3^2} \quad (31)$$

B. The Plant of Nonlinear with Input Dead Zone

Nonlinear system with nonlinear characteristic dead zone in input system stated as,

$$g(q^{-1}) = \frac{0.7q^{-1} - 0.68q^{-2}}{1 - 1.72q^{-1} + 0.74q^{-2}} \quad (32)$$

$$z(t) = \begin{cases} u(t) - 0.4375 \cdot A & if \quad u(t) > 0.5 \cdot A \\ 0.0625 \cdot sign(u(t)) \cdot u^2(t) & if \quad |u(t)| \le 2 \\ u(t) + 0.4375 \cdot A & if \quad u(t) < -2 \end{cases} \quad (33)$$

where, $A = 4$. The block diagram of the system represented (32) and (33) is showed on Fig. 2.

IV. Result of System Identification

For both of plant A and B, e(t) is a Gaussian noise e(t) $\in$ [0:0.05]. The system is then performed with PRBS random signal sequence [-1.0;1.0]. The system identification are tested with 20 independence trial to measure performance. The performances of Quasi-ARX model are measured with the index of performance for 500 pairs data training. Those performances index are RMS index, standard deviation, minimum error, maximum error, and variance in the 20 independent trials.

Fig. 3 up to Fig. 5 are the result of identification with performance index RMS is 0.066 with standard deviation 0.0029, best minimum error 0.0557, maximum error 0.0725, and variance error 8.26 10^{-6}, The convergence error index is reached only five learning process. The index performance measured with different condition with changing the parameter modeling of independent trial, the number of input vector (n_u=[1, 2], n_y =[2, 4]), the number of learning NL [100, 200, 500, 900], and the number hidden layer n_h [4, 10, 20].

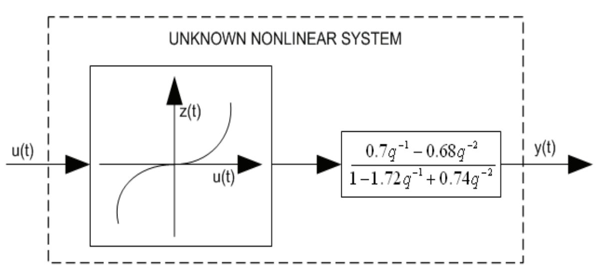

Figure 2. A unknown nonlinear system with dead zone in the input

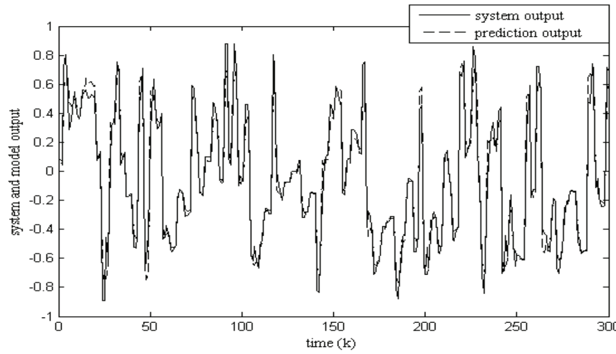

Figure 3. The Quasi-ARX output prediction vs system in 300 samples for plant of nonlinear dynamical system

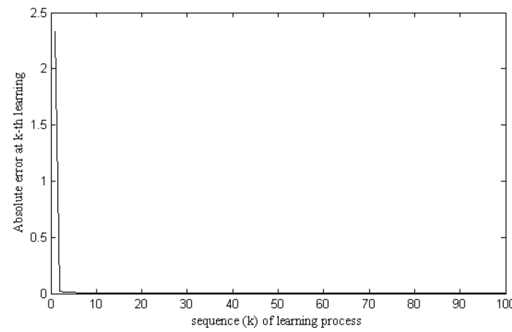

Figure 4. The absolute error sequence at k[th] learning

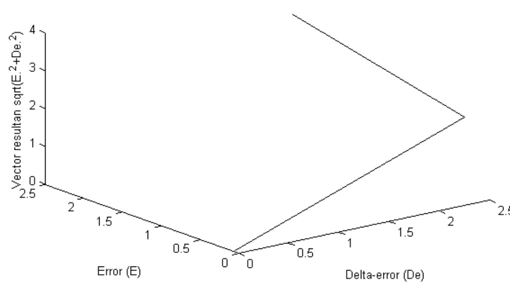

Figure 5. The error vs delta – error characteristic on learning process using Lyapunov function algorithm (RMS = 0.0561)

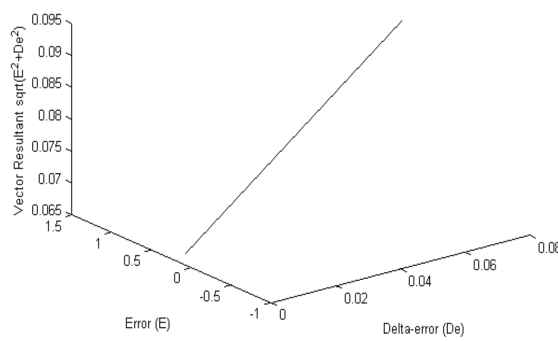

Figure 6. The error vs delta – error characteristic on learning process using back propagation algorithm (RMS = 0.0665)

TABLE I. THE PERFORMANCE INDEX OF THE RESULT OF IDENTIFICATION

	The Average Performance Quasi-ARX Neural Network Based on Lyapunov Learning Algorithm (L) and Back Propagation Algorithm (BP) in 20 Independent Trials x 8 Conditions was Changed in Parameter of (n_u, n_y, N, n_h, NL)			
	Plant A-(L)	***Plant A-(BP)***	***Plant B-(L)***	***Plant B-(BP)***
ARMS	0.066	0.109	0.1214	1.272
Var.	8.26×10^{-6}	3.45×10^{-4}	4.429×10^{-5}	0.0484
SD	0.0029	0.0136	0.006	0.183
Min. RMS	0.0557	0.0567	0.1114	0.163
Max. RMS	0.0725	0.385	0.143	3.489

Var. = variance, SD = Deviation Standard.

The results of the identification plant A and plant B is compared with the results of identification Quasi-ARX neural network with back propagation learning algorithm. The significance improvement can be reached showed in Tab. 1. Based on the result of identification Lyapunov learning algorithm offer the fast convergence error, stable bounded, and it has uniform distribution of error proved by performing method and algorithm in different trials and conditions.

ACKNOWLEDGMENT

This research is sponsored and supported by Indonesian Government Scholarship (Beasiswa Luar Negeri DIKTI - Kementerian Pendidikan dan Kebudayaan Republik Indonesia). The first author is the lecturer of PPNS (Politeknik Perkapalan Negeri Surabaya, Indonesia) who have official task to study in Waseda University, Japan.

REFERENCES

[1] J. Hu, K. Kumamaru, and K. Hirasawa, "A Quasi ARMAX approach to modeling of nonlinear systems", International Journal of Control, Vol. 74, No. 18, pp. 1754-1756, 2001.

[2] H. G. Han, and J. F. Qiao, "Adaptive Computation Algorithm for RBF Neural Network", IEEE Transaction on Neural Networks and Learning Systems, Vol. 23, No. 2, pp. 342-347, 2012.

[3] J. Hu, K. Kumamaru, K. Inoue, and K. Hirasawa, "A hybrid Quasi-ARMAX modeling scheme for identification of nonlinear systems," Transections of the Society of Instrument and Control Engineers, vol. 34, No. 8, pp. 997–985, 1998.

[4] Y. Cheng, L. Wang, and J. Hu, "Quasi – ARX Wavelet Networks for SVR Based Nonlinear System Identification", Nonlinear Theory and Its applications, Vol. 2, No. 2, pp. 167-179, 2011.

[5] J. Hu and K. Hirasawa, "A method for applying neural networks to control of nonlinear systems,"Neural Information Processing: Research and Development, Springer, Berlin, GERMANY, pp. 351–369, 2004.

[6] J. Hu, K. Kumamara, K. Inoue, and K. Hirasawa, "KDI-Based robust fault detection in presence of nonlinear undermodeling," Transections of the Society of Instrument and Control Engineers, vol. 35, No. 2, pp. 200–207, 1999.

[7] J. Hu, K. Hirasawa, and K. Kumamaru, "A Quasi-ARX Model Incorporating Neural Networks for Control of Nonlinear Systems", International Proc. of The 15th IFAC World Congress, Barcelona, July, 2002.

[8] L.Wang, Y. Cheng, J. Hu, "A Quasi-ARX Neural Network with Switching Mechanism to Adaptive Control of Nonlinear Systems", SICE Journal of Control, Measurement, and System Integration, vol. 3, no.4, pp. 246–252, 2010.

[9] L. Behera, S Kumar, and A. Patnaik, "On Adaptive Learning Rate that Guarantees Convergence in Feedforward Networks", IEEE Transaction on Neural Networks, Vol. 17, No. 5, pp. 1116-1125, 2006.

[10] Z. Man, H.R. Wu, S. Liu, and X. Yu, "A New Adaptive Backpropagation Algorithm Based on Lyapunov Stability Theory for Neural Networks", IEEE Transaction on Neural Networks , Vol. 17, No. 6, pp. 1580-1591, 2006.

[11] V.M. Beccera, F.R. Garces, S.J. Nasuto, and W. Holderbaum, "An efficient Parameterization of Dynamic Neural Networks for Nonlinear System Identification", IEEE Transaction on Neural Networks , Vol. 16, No. 4, pp. 983-988, 2005.

[12] F. Abdollahi, H.A. Talebi, and R.V. Patel, "Stable Identification of Nonlinear Systems Using Neural Networks: Theory and Experiments", IEEE/ASME Transaction on Mechatronics, Vol. 11, No. 4, pp. 488-495, 2006.

[13] G. Zhai, I. Matsune, J. Imae, and T. Kobayashi, "A Note on Multiple Lyapunov Functions and Stability Condition for Switched and Hybrid Systems", 16th IEEE International Conference on Control Applications Part of IEEE Multi-conference on Systems and Control, pp. 226-231, 2007.

[14] M. Wang, B. Chen, and P. Shi, "Adaptive Neural Control for a Class of Perturbed Strict-Feedback Nonlinear Time-Delay Systems", IEEE Transaction on Systems, Man, and Cybernetics, Vol. 38, No. 3, pp. 721-730, 2008.

[15] B. Ren, S.S. Ge, K.P. Tee, T.H. Lee, "Adaptive Neural Network Control for Output Feedback Nonlinear Systems Using a Barrier Lyapunov Function", IEEE Transaction on Neural Networks, Vol. 21, No. 8, pp. 1339-1345, 2010.

[16] K. P. Tee, S.S. Ge, E.H. Tay, "Barrier Lyapunov Function for the control of output-constrained nonlinear systems", Automatica, V. 45, pp. 918-927, 2009.

[17] C. Meza, D. Biel, D. Jeltsema, and J.M.A. Scherpen, "Lyapunov-Based Control Schema for Single-Phase Grid-Connected PV Central Inverters", IEEE Transaction on Control System Technology, Vol. 20, No. 2, pp. 520-529, 2012.

[18] B.P. Loop, S.D. Sudhoff, S.H. Zak, and E.L. Zivi, "Estimating Regions of Asymptotic Stability of Power Electronics Systems Using Genetic Algorithms", IEEE Transaction on Control Systems Technology, Vol. 18, No. 5, pp. 1011-1022, 2010.

[19] U. Vaidya, P.G. Mehta, and U.V. Shanbhag, "Nonlinear Stabilization via Control Lyapunov Measure", IEEE Transaction on Automatic Control, Vol. 55, No. 6, pp. 1314-1328, 2010.