Implementation of Lyapunov Learning Algorithm for Fuzzy Switching Adaptive Controller Modeled Under Quasi-ARX Neural Network

Imam Sutrisno
Graduate School of Information Production and Systems
Waseda University
Kitakyushu 808-0135 Japan and
Politeknik Perkapalan Negeri Surabaya
Email: imams3jpg@moegi.waseda.jp

Mohammad Abu Jami'in
Graduate School of Information Production and Systems
Waseda University
Kitakyushu 808-0135 Japan and
Politeknik Perkapalan Negeri Surabaya
Email: mohammad@ruri.waseda.jp

Jinglu Hu
Graduate School of Information Production and Systems
Waseda University
Kitakyushu 808-0135 Japan
Email: jinglu@waseda.jp

***Abstract*—This paper presents a fuzzy adaptive controller applied to a non linear system modeled under a Quasi-linear ARX Neural Network, with stability proof by using the Lyapunov approach. This work exploits the new idea to use Lyapunov function to train multi-input multi-output neural network on the core-part sub-model. The proposed controller is designed between a linear controller and non linear controller based on the characteristic of fuzzy switching algorithm. The improving performances of the Lyapunov learning algorithm are stable in the learning process, fast convergence of error, and able to increase the accuracy of the controller.**

***Index Terms*—Lyapunov Learning Algorithm, Fuzzy Switching Adaptive Controller, Quasi-ARX Neural Network.**

I. Introduction

In the previous work a quasi-ARX Neural Network (QARXNN) model with a switching mechanism studied for system adaptive control as in Refs. [1][2] to simplify the identification for control, which is a combination of a linear part followed by a 0/1 switching part. It can satisfy the stability and performance requirements by using only one model. Nevertheless, there are still some aspects that need to be improved upon the control method based on the QARXNN model. One is the 0/1 hard switching method is not extremely smooth; the second is the parameters of QARXNN model to be adjusted online are highly nonlinear, which deteriorates since the adaptability of a control system. In the controller design, Lyapunov function can be used to ensure bounded ness in control trajectory [3], Lyapunov based on the controller can make the closed-loop system globally [4]. Lyapunov also can be used to estimate the asymptotic region in controller design by genetic algorithm [5], Lyapunov method of linear programming to be used to analyze system stability and control synthesis of the state feedback controller [6].

The rest of this paper organized as follows: Section II derives a QARXNN prediction model. Section III introduces the Lyapunov learning algorithm for QARXNN model. Section IV constructs a fuzzy switching adaptive control system based on the QARXNN using Lyapunov learning algorithm predictors and analyzes the stability of the control system. Section V presents simulation results to demonstrate its performance. Finally, Section VI concludes the paper.

II. Quasi-ARX NN Prediction Model

A. Problem Formulation

Consider a single-input single-output (SISO) time invariant whose input-output relationship described by,

$$y(t) = g(\phi(t)) + e(t) \tag{1}$$

$$\phi(t) = [y(t-1) \cdots y(t-n_y)\; u(t-1) \cdots u(t-n_u)]^T$$

where, $u(t) \in R, y(t) \in R, e(t) \in R, t = 1, 2, \cdots$. $g(\cdot) : R$ is the unknown continuous function describing the nonlinear dynamical systems. $\phi(t) \in R^{n=n_u+n_y}$ is the regression vector composed delays of the input-output data. The number of input variables n equaled to the sum of n_u and n_y. The noise into the system $e(t)$ is the uniform random input added on the unknown function input-output of the system.

Assumption 1: (i) The input and output of training data bounded and $g(\phi(t))$ is the unknown continuous function. (ii) The system has a globally uniformly asymptotically stable zero dynamics. Through Taylor's expansion series, $g(\phi(t))$ can be described around the small region of $\phi(t) = 0$,

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

output of the system (2) can be decomposed into $y_0 = g(0)$ then by the output of the system (2) the regression of the system (1) can be described into (3),

$$y(t) = y_0 + \phi(t)^T\theta(\phi(t)) + e(t) \tag{3}$$

The output of the system (3) can be simplified as a matrix equation as,

$$y(t|\phi(t)) = \Phi^T\psi \tag{4}$$

where, $\psi = [y_0 \theta(\phi(t))]$, $\Phi = [1 \phi^T(t)]^T$, ψ called as nonlinear parameter of core-part and Φ called as an input variable space of macro-part.

B. Prediction Model

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

$$\frac{y(t)}{u(t)} = \frac{b_{(1,t)}q^{-1} + \cdots + b_{(n_u,t)}q^{-n_u}}{1 + a_{(1,t)}q^{-1} + a_{(2,t)}q^{-2} + \cdots + a_{(n_y,t)}q^{-ny}} \quad (5)$$

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

The coefficient of $a_{i,t}$ and $b_{i,t}$ are nonlinear functions coefficient of a regression vector of $\theta(\phi(t))$. The polynomial equation ARX model (5) has the same properties with (3), so (5) can be represented into matrix form by (4).

Regression of the system (1) also using Taylor's expansion series. The output of the system has a linear series function between input spaces $\phi(t)$ to the unknown nonlinear function of the input space $\theta(\phi(t))$.

The output of model in (3) up to $y_p(t+d)$, where d is the time delay. Suppose that the output of model satisfies at the unknown nonlinear coefficient of model $\theta^*(\phi(t))$ so the prediction output $y_p(t+d)$ of the model can be calculated at the time $(t+d)$.

For a system described by (1), the output predictor at d step ahead prediction $y_p(t+d|t, \phi(t))$ satisfies if $y(t)$ satisfies in (3). The form (5) can be rewritten into,

$$\begin{aligned} y(t) = {} & b_{(1,t)}u(t-1) + \cdots + b_{(n_u,t)}u(t-n_u) \\ & - a_{(1,t)}y(t-1) - \cdots - a_{(n_y,t)}y(t-n_y) \end{aligned} \quad (6)$$

The output (6) can be rewritten as a regression output prediction at time delay d described by (7),

$$\begin{aligned} y_p(t+d) = {} & b_{(1,t)}u(t+d-1) + \cdots + b_{(n_u,t)}u(t-n_u+d) \\ & - a_{(1,t)}y(t-1+d) - \cdots - a_{(n_y,t)}y(t-n_y+d) \end{aligned} \quad (7)$$

Based on (4) and (7),the output predictor $y_p(t+d)$ can be expressed into matrix equation as,

$$y_p(t+d|t, \xi(t)) = \Phi_p^T \psi_p \quad (8)$$

where, $\Phi_p = [1 \; y(t+d-1) \; y(t+d-2) \cdots y(t+d-n_y) \; u(t+d-1) \; u(t+d-2) \cdots u(t+d-n_u)]$ and $\psi_p = [\, a_{(1,t)} \cdots a_{(n_y,t)} \; b_{(1,t)} \cdots b_{(n_u,t)}]$.

The input of MLP core-part can be expressed as $\xi = [y(t+d-1) \, y(t+d-2) \cdots y(t+d-n_y) \, u(t+d-2) \cdots u(t+d-n_u) \;\; \chi(t+d)]$. The $\chi(t+d)$ is the virtual input of core-part, the input reference of the system can be used as virtual input of core-part.

The prediction output can be realized by QARXNN. The input vector is Φ_p, and the input of core-part is ξ, and the output of core-part are ψ_p. The output of core-part is the nonlinear coefficient $\aleph$, and the input of core-part ξ is the input-output composed with time delay.

The number of the input vector dimension of ξ is equal to n, and the number of hidden layer is m, and the number of the output layer is $n+1$. The QARXNN incorporating neural network can be expressed as,

$$y_p(t+d|t, \xi(t)) = \Phi_p^T \aleph(\xi(t), \Omega) \quad (9)$$

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

where $\Omega = \{W_1, W_2, \theta\}, W_1 \in R^{n\text{x}m}, W_2 \in R^{M\text{x}(n+1)}$ are the weights matrix of the first layer and second layer. $\theta \in R^{(n+1)\text{x}1}$ is the bias vector of output nodes, and Γ is the diagonal nonlinear operator with similar sigmoidal elements on hidden nodes.

By substitute (9) and (10), the prediction output of QARXN-N model can be expressed by,

$$y_p(t+d|t, \xi(t)) = \Phi_p^T W_2 \Gamma W_1 \xi(t) + \Phi_p^T \theta \quad (11)$$

Define the switching criterion function as follows,

$$\begin{aligned} J_i(t) = {} & \sum_{l=k}^{t} \frac{a_i(l)(\|e_i(l)\|^2 - 4\Delta^2)}{2(1 + a_i(l)\Psi(l-k)^T P_i(l-k-1)\Psi(l-k))} \\ & + c * \sum_{l=t-N+1}^{t} (1 - a_i(l)\, \|e_i(l)\|^2), \quad i = 1, 2 \end{aligned} \quad (12)$$

where N is an integer, and $c \geq 0$ is a predefined constant.

Now, allow the expression of switching law $\chi(t)$ based on the switching criterion function,

$$\chi(t) = \begin{cases} 1, & J_1(t) > J_2(t); \\ 0, & \texttt{otherwise} \end{cases} \quad (13)$$

By comparing $J_1(t)$ and $J_2(t)$, decides when the nonlinear part abandoned. If $J_1(t) > J_2(t)$ the nonlinear part added, else only use linear part to identify.

III. Lyapunov Learning Algorithm for QARXNN Prediction Model

A. MIMO MLP Neural Network

The prediction output satisfied if only the output of core-part can be satisfied. The core-part of QARXNN is MIMO MLP neural network. Consider that the MLP core-part QARXNN have n input node, m hidden node, so the output must has $n+1$ node. The input vector is $\xi(t) = [\xi_1 \; \xi_2 \; \cdots \; \xi_n]$, and the output of hidden node is $h = [h_1 \, h_2 \cdots h_m]$, the output of MLP core-part is $s = [s_1 \, s_2 \cdots s_{n+1}]$.

The output at hidden node with nonlinear function in the hidden layer can be expressed as,

$$h_j = f_j\left(\sum_{i=1}^{n} \sum_{j=1}^{m} \xi_i W_{1(i,j)}\right) = f_j(z) \quad (14)$$

where f_j is the nonlinear sigmoidal function at hidden nodes.

$$f(z) = \frac{1 - e^{-z}}{1 + e^{-z}} \quad (15)$$

The output of nonlinear MLP core-part can be expressed as,

$$\aleph(\xi(t), \Omega) = \sum_{j=1}^{m} \sum_{r=1}^{n+1} h_j W_{2(j,r)} + \theta \quad (16)$$

where, $W_{1(i,j)}, W_{2(j,r)}, \theta$ are the weight matrix in the first layer, the weight matrix in the second layer, and the bias vector of the output node. The output of core-part $\aleph(;)$ is the MLP output with n input nodes of $\xi(t)$. Consider there are two sub-models in QARXNN, those are linear part or macro-part, and nonlinear part or core-part. The output of two sub-models expressed by,

$$SM1 \quad z_l = \Phi_p^T \theta \tag{17}$$

$$SM2 \quad z_n = \Phi_p^T W_2 \Gamma(W_1 \xi(t)) \tag{18}$$

The output guidance of two sub-models are expressed by,

$$z_l = y_p(t+d|t,\xi(t)) - \Phi_p^T W_2 \Gamma(W_1 \xi(t)) \tag{19}$$

$$z_n = y_p(t+d|t,\xi(t)) - \Phi_p^T \theta \tag{20}$$

B. Lyapunov Learning Algorithm

The learning algorithm based on Lyapunov function described as follows,

1) Set $\theta = 0$, and small initial values of W_1 and W_2, set $k = 1$.
2) Calculate z_l, then estimate θ by model SM1 using LSE (Least Square Error) method.
3) Calculate error, $z_n = \aleph(\Omega^*, \xi(t))$,

$$e(k) = \aleph(\Omega^*, \xi(t)) - \aleph(\Omega, \xi(t)) \tag{21}$$

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

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, if $V(k) > 0$ and $V(k) < 0$, the error output will converge to zero at time goes to infinity.

$$\lim_{k \to \infty} e(k) = 0$$

6) Stop if pre-specified condition is met, otherwise goto step 2. set k=k+1.

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

$$V(k) = \beta^k e^2(k) \tag{22}$$

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

$$\begin{aligned}
\Delta V(k) &= V(k) - V(k-1) \\
&= \beta^k e^2(k) - \beta^{k-1} e^2(k-1) \\
&= \beta^k (\aleph(\Omega^*, \xi(t)) - \aleph(\Omega, \xi(t)))^2 - \beta^{k-1} e^2(k-1) \\
&= \beta^k (\aleph(\Omega^*, \xi(t)) - \sum_{j=1}^{m} \sum_{r=1}^{m} W_{2(j,r)} h_j(t))^2 - \beta^{k-1} e^2(k-1) \\
&= \beta^k (\aleph(\Omega^*, \xi(t)) - \sum_{j=1}^{m} \sum_{r=1}^{m} W_{2(j,r)} f_j (\sum_{i=1}^{n} \sum_{j=1}^{m} W_{1(i,j)} \xi_i))^2 \\
&\quad - \beta^{k-1} e^2(k-1) \\
&= \beta^k (\aleph(\Omega^*, \xi(t)) - \sum_{j=1}^{m} \sum_{r=1}^{m} W_{2(j,r)} f_j (\sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{m\xi_i} f_j^{-1}(R)))^2 \\
&\quad - \beta^{k-1} e^2(k-1) \\
&= -(\beta^{k-1} - 1) e^2(k-1) < 0, \ if \ \beta > 1
\end{aligned} \tag{23}$$

IV. Fuzzy Switching Controller And Its Stability

A. Controller Design

The controller design includes two stage; the first stage for identifying QARXNN prediction model; and the next stage for deriving control law. The identified QARXNN prediction model from previous parts, described by,

$$\hat{y}(t+d \mid t, \xi(t)) = \Psi^T(t)\hat{\theta} + \chi(t)\Psi^T(t).\hat{W}_2 \Gamma(\hat{W}_1 \xi(t) + \hat{B}) \tag{24}$$

where $\widehat{\theta}, \hat{W}_1, \hat{W}_2$ and $\hat{B}$ used for controller design. Consider a minimum variance control with the criterion function as follows,

$$M(t+1) = \left[\frac{1}{2}(y(t+d) - y^*(t+d))^2 + \frac{\lambda}{2} u(t)^2\right] \tag{25}$$

where λ is weighting factor for the control input. The controllers can achieve by solving,

$$\frac{\partial M(t+1)}{\partial u_i} = 0 \quad i = 1, 2 \tag{26}$$

Two controllers can be derived based on QARXNN prediction model by solving (26),

$$\begin{aligned}
C_1 : u_l(t) \quad &= \quad \frac{\hat{b}_0^1}{\hat{b}_0^1 \hat{b}_0^1 + \lambda}((\hat{b}_0^1 - \hat{b}^1(q^{-1})q)u(t-1) \\
&+ \quad y^*(t+1) - \hat{a}^1(q^{-1})y(t))
\end{aligned} \tag{27}$$

$$\begin{aligned}
C_2 : u_n(t) \quad &= \quad \frac{\hat{\beta}_{0,t}}{\hat{\beta}_{0,t}^2 + \lambda}((\hat{\beta}_{0,t} - \hat{\beta}(q^{-1})q)u(t-1) \\
&+ \quad y^*(t+1) - \hat{\alpha}(q^{-1}, \xi(t))y(t))
\end{aligned} \tag{28}$$

where
$\hat{a}(q^{-1}) = \hat{a}_0 + \hat{a}_1 q^{-1} + \cdots + \hat{a}_{n-1} q^{-n+1}$;
$\hat{b}(q^{-1}) = \hat{b}_0 + \hat{b}_1 q^{-1} + \cdots + \hat{b}_{m+d-2} q^{-m-d+2}$;
$\hat{\alpha}(q^{-1}, \xi(t)) = \hat{\alpha}_{0,t} + \hat{\alpha}_{1,t} q^{-1} + \cdots + \hat{\alpha}_{n-1,t} q^{-n+1}$;
$\hat{\beta}(q^{-1}, \xi(t)) = \hat{\beta}_{0,t} + \hat{\beta}_{1,t} q^{-1} + \cdots + \hat{\beta}_{m+d-2} q^{-m-d+2}$;
The coefficients can be gotten as follows,
$[\hat{\alpha_0} \cdots \hat{\alpha}_{n-1} \hat{\beta_0} \cdots \hat{\beta}_{m+d-2}] = \hat{\theta}$
$[\hat{\alpha_{0,t}} \cdots \hat{\alpha}_{n-1,t} \hat{\beta_{0,t}} \cdots \hat{\beta}_{m+d-2,t}] = \hat{\theta} + \hat{W}_2 \Gamma(\hat{W}_1 \xi(t) + \hat{B})$
Switching control based on two or more controllers researched as in Refs. [2][9]. An integer switching law was introducing into control model just like the function $\xi(t)$ in the prediction model. It means that the linear and nonlinear controllers alternately used. This paper used a fuzzy membership functions $v(t)$ based on the criterion function $J_1(t)$ and $J_2(t)$,

$$v(t) = \begin{cases} 0, & x(t) > K; \\ x(t), & k \leq x(t) \leq K; \\ 1, & x(t) < k \end{cases} \tag{29}$$

where $x(t) = \frac{J_1(t)}{J_1(t)+J_2(t)}$. K and k are constants which satisfy $k \in (0, 0.5), K \in (0.5, 1)$. Now, a fuzzy switching controller [7][8] obtained based on the fuzzy membership functions $v(t)$,

$$C : u(t) = (1 - v(t))u_l(t) + v(t)u_n(t) \tag{30}$$

The switching law $\xi(t)$ firstly calculated from input and output signals and model errors, then used into the identified model. The fuzzy switching law $v(t)$ calculated from input and output signals and model errors, then is used into the control model. The proposed controller has four distinguishing features,

1) it is linear for the variable synthesized in control systems;
2) its parameters have explicit meanings;
3) it is only one predictor which combines a switching algorithm
4) it has a fuzzy switching mechanism not a simple 0 or 1 switching

B. Stability Analysis

Give the stability analysis of the proposed nonlinear controller system as follow, **Theorem:** For the system (1) with adaptive fuzzy switching controller (30), all the input and output signals in the closed-loop system are bounded. Moreover, the tracking error of the system can converge on zero when a properly neural network is determined. *Proof :* Firstly, similar to Refs.[2][7], it can get,

$$\lim_{N\to\infty} \sum_{t=1}^{N} \frac{a(t)^2(e_1(t)^2 - 4\Delta)}{2(1 + \Psi(t-d)^T\Psi(t-d))} < \infty, \tag{31}$$

and

$$\lim_{N\to\infty} \frac{a(t)^2(e_1(t)^2 - 4\Delta)}{2(1 + \Psi(t-d)^T\Psi(t-d))} \to 0 \tag{32}$$

Along with **Assumption 1** (ii) $e_1(t)$ is bounded. By (12) and (31), the second term of $J_1(t)$ always bounded. $J_2(t)$ has two cases, (i) $J_2(t)$ is bounded, so the model error $e(t)$ bounded and satisfies Eq.(32) (ii) $J_2(t)$ is unbounded. Since (1) $J_1(t)$ is bounded. So there exist a constant t_0 such that $v(t) = 1, \forall t > t_0$. The model also has bounded error $e(t)$.

From above inequalities, the input and output of the closed-loop switching control system bounded. The linear part always bounded. If a proper nonlinear part is chosen and the accurate parameters is adjusted, the model error $e_2(t)$ can converge on zero. It also exists a constant T_0 satisfy $v(t) = 0, \forall t > T_0$. Then the tracking error of the model can converge on zero.

V. Control Simulations

The system considered is a nonlinear one governed by,

$$y(t) = g[y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)] + v(t) \tag{33}$$

where $g(.)$ is the nonlinear function with a disturbance,

$$\begin{aligned} g[x_1, x_2, x_3, x_4, x_5] &= p_t \frac{x_1x_2x_3x_5(x_3-1) + x_4}{1 + x_2^2 + x_3^2} \\ &+ q_t \mathtt{ln}(1 + 0.2x_4) \end{aligned} \tag{34}$$

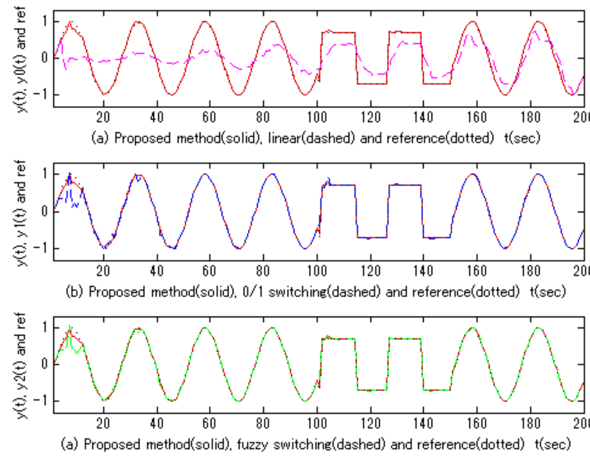

Fig. 1. (a) Simulation result of proposed method compare with linear control and (b) Simulation result of proposed method compare with 0/1 switching (c) Simulation result of proposed method compare with fuzzy switching Ref.[8]

The desired output in this example is a piecewise function:

$$y^*(t) = \begin{cases} 0.6y^*(t-1) & \\ +r(t-1), & t \in [1, 100] \cup [151, 200] \\ 0.7\mathtt{sign}(0.4493 & \\ y^*(t-1) & \\ +0.57r(t-1), & t \in [101, 150] \end{cases} \tag{35}$$

where $r(t) = \sin(2\pi t/25)$. The performance of network is measured using $RMSE$ root mean square error index expressed as,

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N}(y_p(t) - y(t))^2}{N}} \tag{36}$$

where $t = 1, 2, \cdots, N$. The other performance of Quasi-ARX neural network model for prediction is measured by NPE (Normalized Prediction Error) index stated as,

$$NPE(t+d) = \sqrt{\frac{\sum_{t=1}^{N}(y_p(t+d) - y(t+d))^2}{\sum_{t=1}^{N}(y(t+d))^2}} \text{x}100\% \tag{37}$$

In Fig. 1, the black dotted line is the desired output, the red solid line denotes the proposed method output $y(t)$. In Fig. 1(a), magenta dashed line shows the linear control output $y_0(t)$. Obviously, the control output with the proposed method is nearly consistent with the desired output at most of the time and the linear control output have bad performance. In the Fig. 1(b), blue dashed line shows the 0/1 switching control output $y_1(t)$. Obviously the proposed control output is almost coincidence with the desired output. It also can be found that the 0/1 switching control results have some wobble at the last half time. In Fig. 1(c), green dashed line shows the fuzzy switching control output $y_2(t)$. Obviously the proposed control method can do better than fuzzy switching controller.

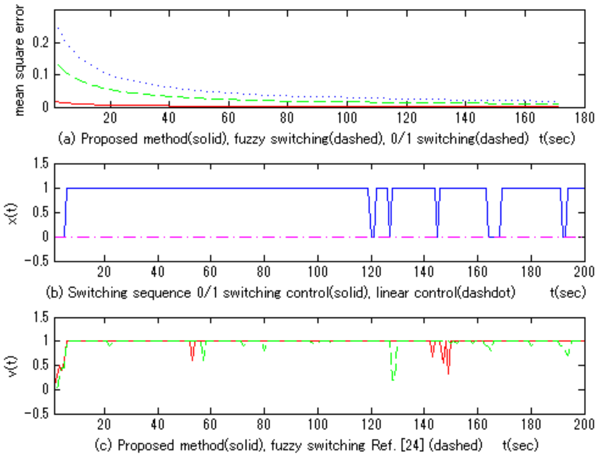

Fig. 2. (a) Convergence characteristics of the errors (b) Switching sequence (c) Fuzzy switching sequence

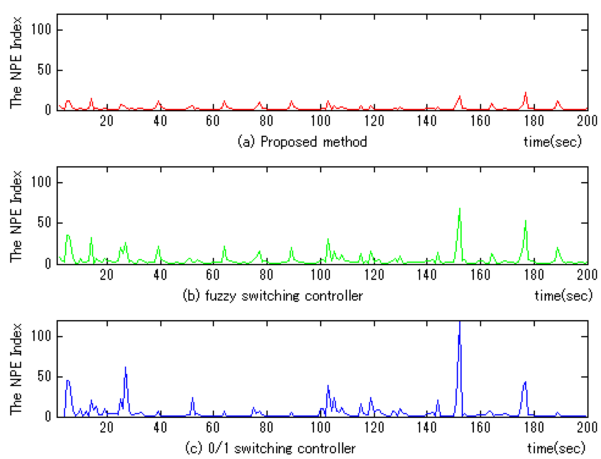

Fig. 3. The NPE Index Performance using Lyapunov learning algorithm

The similar conclusion also can be gotten from convergence characteristic of the errors is shown in Fig. 2(a). The switching sequence is presented in Fig. 2(b) and fuzzy switching sequence is presented in Fig. 2(c).

Table I gives the errors of three methods; the proposed control method gets a better accuracy and the error of the proposed method is smaller than the other methods.

In Fig. 3, the NPE index near to constant value, the fluctuation in a small range of NPE value is from uncertainty function of random PRBS signal when the system are tested by deterministic input. The NPE index prediction are always constant in independence trials, it proof that the Lyapunov learning algorithm make stable in system model prediction.

VI. Conclusion

This study has successfully demonstrated the effectiveness of the proposed fuzzy switching adaptive controller using Lyapunov learning algorithm based on QARXNN prediction model. First, the principles of QARXNN prediction model was derived. Then, the network structure and theoretical bases of proposed method has been adopted to adapt the Lyapunov learning algorithm to replace the traditional trial-and-error method. Finally, the control performance of the proposed method based on QARXNN prediction model has been confirmed by experimental result.

The main contributions of this study are:(1)the successful development of an improved fuzzy switching controller;(2) the successful adoption of a Lyapunov learning algorithm;(3)the successful application of the fuzzy switching controller based on QARXNN prediction model to control nonlinear system with robust control performance.

TABLE I
Comparison results of the errors.

Method	RMSE	Variances	Accuracy
Proposed control	0.0109	0.023	95.78
Fuzzy switching control	0.0147	0.047	94.55
0/1 switching control	0.0201	0.082	91.41
linear control	0.0240	0.105	81.23

Acknowledgment

This research has been supported by Directorate General of Higher Education, Ministry of National Education Indonesia and Shipbuilding Institute of Polytechnic Surabaya (SHIPS).

References

[1] L. Wang L, Y. Cheng and 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.

[2] L. Wang, Y. Cheng and J. Hu, "Adaptive control for nonlinear systems based on quasi-ARX neural network," *Proceedings of World Congress on Nature and Biologically Inspired Computing (NaBIC2009) (Coimbatore)*, pp. 1548-1551, 2009.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] L. Wang, Y. Cheng and J. Hu, "Quasi-arx neural network and its application to adaptive control of nonlinear systems," *in Proceedings of The Fifteenth International Symposium on Artificial Life and Robotics (AROB 15th 10) (Oita)*, pp. 577-580, February 2010.

[8] L. Wang, Y. Cheng and J. Hu, "Nonlinear adaptive control using a fuzzy switching mechanism based on improved quasi-arx neural network," *in Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN2010) (Barcelona)*, pp. 1-7, 2010.

[9] I. Sutrisno, M.A. Jami'in and J. Hu, "Neural predictive controller of nonlinear systems based on quasi-ARX neural network," *in Proceedings of the 2012 International Conference on Automation and Computing (ICAC 18th 2012) (Loughborough UK)*, pp. 83-88, 2012.