The Twentieth International Symposium on Artificial Life and Robotics 2015 (AROB 20th 2015),
B-Con Plaza, Beppu, Japan, January 21-23, 2015

The State-Dynamic-Error-Based Switching Control under Quasi-ARX Neural Network Model

Mohammad Abu Jami'in (1,2); Imam Sutrisno (1,2); Jinglu Hu (1)

(1) Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan
(2) Politeknik Perkapalan Negeri Surabaya, Surabaya, Indonesia
Email: mohammad@ruri.waseda.jp; imam@moegi.waseda.jp; jinglu@waseda.jp

***Abstract*—In our previous research, an error-based switching control has already used for controlling nonlinear system. However, the switching function is not work efficiently, because it is difficult to obtain more information from error vector to determine the stability of the control system. Hence unnecessary switching to linear controller will be longer and more often that causes the accuracy of the control system become poor. In this paper, a new switching rule based on Lyapunov stability theorem is proposed which is derived from the state dependent parameter estimation (SDPE). Not only error but also one up to *p*-th differential error will be available as the switching variable. Thus the proposed control method is able to keep the stability and improve the accuracy of the control system. A numerical simulation reveal that the proposed control gives satisfactory tracking and disturbances rejection performances. Experimental results demonstrate its effectiveness.**

I. Introduction

Nonlinear system with the parameter always uncertain has more than one stability region [1]. For those problems, with only nonlinear controller cannot guarantee the bounded of the input-output closed-loop control. To relax nonlinear control and to guarantee bounded controller, two controllers are used with switching mechanism. Thought, by using the linearization technique, nonlinear system is linear to the input controller, therefore the control law can be derived [2], [3], [4], [5], [6]. With a different approach of linearization technique, a nonlinear system can also be expressed as a linear relationship between nonlinear coefficient and its input vector called as quasi-linear ARX model. Thus the control law can be derived by a simple linear inverse utilizing nonlinear coefficient. In this paper, the controller strategy under a quasi-linear ARX model is used to control nonlinear system. Fortunately, quasi-linear model is comprised with the two parts parameter of the linear and nonlinear. It can be used to facilitate the controller design based on the estimated parameter of the linear and nonlinear parts combined with switching mechanism.

A quasi-ARX neural networks (QARXNN) model is a nonlinear model. It can be simplified as a linear correlation between the input vector and its nonlinear coefficient. An embedded system is a sub-system to give a coefficient for each element of the input vector. The coefficient called as state dependent parameter estimation (SDPE) that consists of two parts: linear parameter and nonlinear parameter. The linear parameter estimator (LPE) is performed by least square error (LSE) algorithm, which is set as bias vector for the output nodes of nonlinear multi layers Parceptron neural network (MLPNN) performed by gradient-descent-based learning algorithm [7], [8], [9]. In view of a nonlinear system is modeled under a quasi-linear autoregressive (quasi-ARX) model, nonlinear nature is placed on to the coefficients of the autoregressive (AR) or autoregressive moving average (ARMA). If the system is linear than SDPE will converge at the fixed value, whereas if the system is nonlinear then SDPE is a variable that will change at any time [10]. Because QARXNN prediction model consists of two parts parameter. It is easy to design switching-based control utilizing its parameters. Therefore, the controller is comprised of a linear robust adaptive controller (LRAC), a nonlinear robust adaptive controller (NRAC), and a switching mechanism. NRAC controller is designed based on SDPE that is the output of MLPNN, whereas LRAC is designed based on linear part parameter estimator (LPE) with the LSE algorithm.

A switching control based on error convergence index uses the error in the linear and nonlinear parts as switching variable. It is difficult to obtain a lot of information based on error vector to determine the stability of nonlinear systems. It only able to detect the condition of the system tends to be stable or unstable by the next controller input. Thus unnecessary switching to linear controller will be longer and more often resulting poor controller performance. Therefore, in this paper, a different switching mechanism is proposed based on the state of dynamics tracking error so that much information can be obtained. The switching based on convergence index of error is a function that uses the error space as variable, whereas the proposed switching control uses the state space of error dynamic. By using Lyapunov theorem, the stability of the closed-loop system is derived based on the state of dynamic tracking error resulting the switching mechanism can work more effective and efficient. Moreover, the switching formula use SDPE as switching variable make it easy to be used because it is already available from prediction model.

By using the proposed switching law, the controller is comprised of a linear robust adaptive controller (LRAC), a nonlinear robust adaptive controller (NRAC), and a switching mechanism. NRAC controller is designed based on both of linear and nonlinear parts of SDPE, whereas LRAC is designed by using only linear part parameter estimator. To begin, a QARXNN model is used to identify a dynamic system online. The network parameters are updated continuously in accordance with the sampling time. The trained network weights of QARXNN is used to estimate SDPE by the next regression input. With the estimated parameters of the linear and nonlinear parts, the dynamics tracking error is derived. The stability of the overall system is then verified by Lyapunov theorem so that ultimately bounded tracking is accomplished.

The advantages of the use proposed controller as follows: 1) a simplified quasi-ARX neural network model presented by state dependent parameter estimation (SDPE) is used to derive the controller formulation to deal with its computational complexity. 2) The control law can be derived easily from model prediction based on linearization technique that the system is linear to the input controller. The SDPE is used to parameterize the input vector. Hence, the control law is derived by utilizing the transformation by its linear inverse. 3) Lyapunov-based switching control is performed to guarantee the closed-loop stability using the state dependent parameter (SDPE). It improve controller accuracy by reducing unnecessary switching to linear controller.

II. Quasi-ARX Neural Networks Model

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

$$y(t) \quad = \quad g(\phi(t)) \qquad (1)$$

where $g(\cdot)$ denotes a nonlinear function, $y(t) \in R$ denotes the system output and $t = 1, 2, \cdots$ denotes the sampling of time. The quasi-linear ARX model is nonlinear model derived by performing Taylor expansion series of nonlinear system. The model predictor is made linear the input variable $u(t)$ presented as [7], [11]:

$$y(t) \quad = \quad \phi^T(t)\aleph(\xi(t)) \qquad (2)$$

where $\phi(t) = [y(t-1)\cdots y(t-n_y)\ u(t-1)\cdots u(t-n_u)]^T$ and $\aleph(\xi(t)) = [a_{(1,t)}\cdots a_{(n_y,t)}\, b_{(1,t)}\cdots b_{(n_u,t)}]^T$, $\xi(t) = [y(t-1)\cdots y(t-n_y)u(t-2)\cdots u(t-n_u)v(t)]^T$ are the input vector, the coefficients of the input vector, and the the input of embedded system of MLPNN, respectively. $v(t)$ is a virtual input and the reference set point can be set as $v(t)$. $\phi(t) \in R^{n=n_u+n_y}$ where n_u and n_y denote the orders of time delay in the input and the orders of time delay in the output, respectively.

The SDPE consists of the linear parameters (LP) θ and nonlinear parameter (NP) $\delta(\xi(t))$. LP and NP are executed using the least-squares error (LSE) algorithm and gradient descent based MIMO-MLPNN, respectively. The SDPE is the sum of LP and NP for the nonlinear robust adaptive controller (NRAC) while for LRAC only LP is used.

$$\aleph(\xi(t), \Omega) = \delta(\xi(t)) + \theta \qquad (3)$$

where $\Omega = \{W_1, W_2, \theta\}$ is the network parameters that consists of the weights of the hidden layer W_1, the weights of output layer W_2, and the bias vector of the output nodes θ. Nonlinear part executed by MLPNN is described by

$$\delta(\xi(t), W) = W_2 \Gamma W_1(\xi(t)). \qquad (4)$$

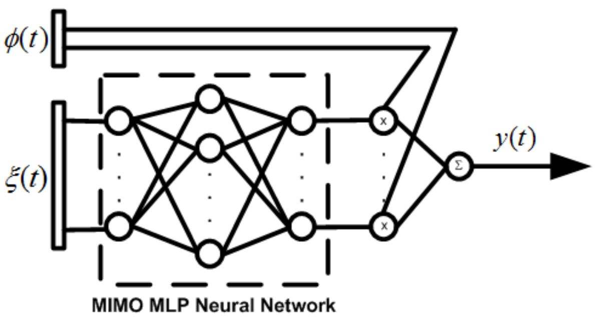

Fig. 1. A QARXNN with an embedded MIMO-MLPNN .

where Γ is the diagonal nonlinear operator with the identical sigmoidal elements on hidden nodes and $W = \{W_1, W_2\}$. The detail algorithm of the quasi-ARX neural network model can be found in [7], [10], [12]. Fig. 1 shows a QARXNN model with a MLPNN is an embedded system. In our main theoretical result, the following assumption are made.

A1. The pairs of the input-output of training data are bounded.
A2. The coefficients of the regression vector $\aleph(\xi(t))$ are bounded.
A3. There exists optimal weights of the regression coefficient $\aleph^*(\xi(t))$.

III. Control Strategy

A model in (2) is simplified and can be rewritten in the form of the relationship between the input vector and its coefficients as follows:

$$\begin{aligned} y(t) \quad = \quad & \hat{a}_{(1,t)}y(t-1) + \hat{a}_{(2,t)}y(t-2) + \\ & \hat{a}_{(n_y,t)}y(t-n_y) + \hat{b}_{(1,t)}u(t-1) + \\ & \hat{b}_{(2,t)}u(t-2) + \hat{b}_{(n_u,t)}u(t-n_u). \qquad (5) \end{aligned}$$

where $\hat{\aleph}(\xi(t)) = [\hat{a}_{(1,t)}\cdots\hat{a}_{(n_y,t)}\,\hat{b}_{(1,t)}\cdots\hat{b}_{(n_u,t)}]^T$ is the coefficient of the input vector called as SDPE. Define the tracking error vector as follows:

$$\begin{aligned} E(t) &= (e(t), \dot{e}(t), \ddot{e}(t), \cdots, \dot{e}_{p-1}(t)), \\ e(t) &= y(t) - y^d(t), \\ \dot{e}(t) &= \frac{\partial e(t)}{\partial t} = (e(t) - e(t-1))/\Delta t, \\ &\vdots\ , \\ \dot{e}_{p-1}(t) &= (e(t-p+2) - e(t-p+1))/\Delta t, \qquad (6) \end{aligned}$$

The tracking error vector is derived by using QARXNN prediction model such as:

$$\begin{aligned} \dot{e}(t) &= \frac{\partial e(t)}{\partial t} = (e(t) - e(t-1))/\Delta t, \\ &= ((y(t) - y^d(t)) - (y(t-1) - y^d(t-1)))/\Delta t \\ &= (\triangle y(t) - \triangle y^d(t))/\Delta t \simeq \dot{y}(t) - \dot{y}^d(t), \qquad (7) \end{aligned}$$

where the notation of $\triangle y(t)$ denotes $y(t) - y(t-1)$. The closed-loop system of the tracking error vector dynamics is described as:

$$\begin{array}{rcl} \dot{y}_p(t) & = & \dot{y}_p^d(t) + K^T E(t) \\ \dot{y}_p(t) - \dot{y}_p^d(t) & = & -k_p\dot{e}_{p-1}(t) - k_{p-1}\dot{e}_{p-2}(t) - \cdots - k_1 e(t) \\ \dot{e}_p & = & -k_p\dot{e}_{p-1}(t) - k_{p-1}\dot{e}_{p-2}(t) - \cdots - k_1 e(t) \\ 0 & = & \dot{e}_p + k_p\dot{e}_{p-1}(t) + k_{p-1}\dot{e}_{p-2}(t) + \cdots + k_1 e(t) \end{array} \quad (8)$$

where $K = [k_p, k_{p-1} \cdots k_1] \in R^p$, $k_i(i = 1, \cdots, p)$ are positive constants, y^d is the reference input trajectory, and p is the degree of tracking error derivative.

To derive the control signal, model in (5) can be rewritten as

$$\begin{array}{rcl} u(t-1) & = & \frac{1}{\hat{b}_{1,t}}(y(t) + g(t)) \quad (9) \\ g(t) & = & -\hat{a}_{(1,t)}y(t-1) - \hat{a}_{(2,t)}y(t-2) \\ & & -\hat{a}_{(n_y,t)}y(t-n_y) - \hat{b}_{(2,t)}u(t-2) \\ & & -\hat{b}_{(n_u,t)}u(t-n_u)). \quad (10) \end{array}$$

If model in (2) is simplified with (5), satisfies the input-output mapping of the system, assumption **A1.**,**A2.** and **A3.** are fulfilled, then the output at time $(t+d)$ can be predicted. Equation (2) is regressed at time $(t+d)$ to calculate the output at d step ahead prediction described as,

$$y(t+d) \;=\; \phi^T(t+d)\hat{\aleph}(\xi(t+d)) \quad (11)$$

where $\hat{\aleph}(\xi(t+d)) = [\hat{a}_{(1,t+d)} \cdots \hat{a}_{(n_y,t+d)}\hat{b}_{(1,t+d)} \cdots \hat{b}_{(n_u,t+d)}]^T$ is the coefficient of the input vector, $\phi(t+d) = [y(t+d-1)\, y(t+d-2) \cdots y(t+d-n_y)u(t+d-1)u(t+d-2) \cdots u(t+d-n_u)]^T$ is the input vector at d step ahead prediction, and $\xi(t+d) = [y(t+d-1)y(t+d-2) \cdots y(t+d-n_y)u(t+d-2)u(t+d-3) \cdots u(t+d-n_u-1)v(t+d)]^T$. For online step ahead prediction d is equal to one. From (11), we have

$$\begin{array}{rcl} u(t) & = & \frac{1}{\hat{b}_{1,t+1}}(y(t+1) + g(t+1)) \quad (12) \\ g(t+1) & = & -\hat{a}_{(1,t+1)}y(t) - \hat{a}_{(2,t+1)}y(t-1) \\ & & - \cdots - \hat{a}_{(n_y,t+1)}y(t-n_y+1) - \hat{b}_{(2,t+1)}u(t-1) \\ & & - \cdots - \hat{b}_{(n_u,t+1)}u(t-n_u+1)). \quad (13) \end{array}$$

where $u(t)$ is a control signal corresponding to the networks output of $y(t)$ and $u^*(t)$ is the desired control signal which correspond to y^d (y^d denotes the desired control output). However, in cases of nonlinear systems, the parameters are always uncertain that may cause instability to the system.

To maintain the accuracy of the closed-loop control and stability, the switching line is used between the linear and nonlinear parameters. SDPE can be divided by linear and nonlinear parts, thus the control law can be defined as:

$$u(t) \;=\; \chi(t)u_n + (1 - \chi(t))u_l(t). \quad (14)$$

where u_l is a control signal calculated by linear robust control by using linear part parameters of $\hat{\theta}$ and u_n is a control signal by nonlinear robust control using nonlinear part parameters of $\hat{\aleph}(\cdot)$. $\chi(t)$ is a switching line which $\chi(t) = 1$ denotes nonlinear robust control and $\chi(t) = 0$ denotes linear robust control. By (12) and (14), we define a controller estimation error as

$$\begin{array}{rcl} u(t) - u^*(t) & = & \frac{1}{\hat{b}_{1,t+1}}(y(t+1) + \hat{g}(t+1)) - \\ & & \frac{1}{\hat{b}_{1,t+1}}(y^d(t+1) + g(t+1)) \\ & = & \frac{1}{\hat{b}_{1,t+1}}(y(t+1) - y^d(t+1) + \\ & & \hat{g}(t+1) - g(t+1)) \\ U(t) & = & \frac{1}{\hat{b}_{1,t+1}}(e(t+1) + G) \end{array} \quad (15)$$

where $U = u(\cdot) - u^*(\cdot)$,$G = \hat{g}(\cdot) - g(\cdot)$, $\hat{g}(\cdot)$ is calculated using the output of prediction model. The error tracking can be obtained by

$$\begin{array}{rcl} e(t+1) & = & y(t+1) - y^d(t+1) = \hat{b}_{1,t+1}U(t) - G(t+1) \\ \dot{e}(t+1) & = & e(t+1) - e(t) \\ & = & \hat{b}_{1,t+1}U(t) - \hat{b}_{1,t}U(t-1) - G(t+1) + G(t) \\ \ddot{e}(t+1) & = & \dot{e}(t+1) - \dot{e}(t) \\ & = & \hat{b}_{1,t+1}U(t) - 2\hat{b}_{1,t}U(t-1) \\ & & +\hat{b}_{1,t-1}U(t-2) - G(t+1) + 2G(t) - G(t-1) \\ \dot{e}_3(t+1) & = & \dot{e}_2(t+1) - \dot{e}_2(t) \\ & = & \hat{b}_{1,t+1}U(t) - 3\hat{b}_{1,t}U(t-1) \\ & & +3\hat{b}_{1,t-1}U(t-2) - \hat{b}_{1,t-2}U(t-3) \\ & & -G(t+1) + 3G(t) - 3G(t-1) + G(t-2) \\ \dot{e}_4(t+1) & = & \dot{e}_3(t+1) - \dot{e}_3(t) \\ & = & \hat{b}_{1,t+1}U(t) - 4\hat{b}_{1,t}U(t-1) \\ & & +6\hat{b}_{1,t-1}U(t-2) - 4\hat{b}_{1,t-2}U(t-3) \\ & & +\hat{b}_{1,t-3}U(t-4) - G(t+1) + 4G(t) \\ & & -6G(t-1) + 4G(t-2) - G(t-3) \end{array} \quad (16)$$

Through (8) and (16), the dynamic tracking error can be stated as follows:

$$\dot{E} \;=\; AE + BU + G. \quad (17)$$

where

$$A = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 \\ -k_p & -k_{p-1} & \cdots & -k_1 \end{pmatrix},$$

$$B = \begin{pmatrix} \hat{b}_{1,t+1} & 0 & 0 & 0 \\ \hat{b}_{1,t+1} & -\hat{b}_{1,t} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ c_1\hat{b}_{1,t+3-p} & -c_2\hat{b}_{1,t+2-p} & \cdots & (-1)^p c_{p+1}\hat{b}_{1,t+2-p} \end{pmatrix},$$

$$U = \begin{pmatrix} U(t) \\ U(t-1) \\ \cdots \\ U(t-p+1) \end{pmatrix}, and$$

$$G = \begin{pmatrix} -G(t+1) \\ -G(t+1)+G(t) \\ \vdots \\ -c_1G(t+3-p)+\cdots+(-1)^{p+1}c_{p+1}G(t+2-p) \end{pmatrix}$$

where A is a nonsingular matrix, c_n are binomial series coefficients such as $\begin{pmatrix} p \\ r \end{pmatrix} = \frac{p!}{r!\,(p-r)!}, 0 \leq r \leq p$.

By (8) and (17), we can calculate K such that the roots of the characteristic equation (17) can be chosen strictly in such a way that the poles lie in the left half of the complex plane. This will ensure $\lim_{t\to\infty} e(t) = 0$. A minimum approximation control error can be defined as

$$\varepsilon = u^* - u(E|\aleph^*(\cdot)). \tag{18}$$

The controller objective is to maintain stability and accuracy of the closed-loop system by considering ε such that

$$\aleph^*(\cdot) = \arg\min_{\aleph(\cdot)\in R} [\sup_{E\in R} |U|],$$

where $\aleph^*(\cdot)$ is an optimal network weight that achieves the minimum approximation error obtained through network learning. If the system dynamic in (17) is a bounded by ($|U| < \varepsilon$), then there are will be a positive real number of ε. By introducing ε in(17)

$$\dot{E} = AE + B(U(E|\aleph(\cdot)) - U(E|\aleph^*(\cdot)) - \varepsilon) + G. \tag{19}$$

Consider a Lyapunov function stated as

$$V(t) = \frac{1}{2}E^T PE \tag{20}$$

where P is a symmetric positive definite matrix. Since $V(t)$ was selected to be positive definite, $\dot{V}(t)$ has to be negative semidefinite in order to make the system is uniformly stable. Therefore, we require $\dot{V}(t) = -\dot{E}^T QE$ to be a negative semidefinite that implies $V(t) \leq V(0)$. A negative semidefinite matrix Q is stated as

$$Q = -(A^T P + PA) \tag{21}$$

Theorem 1: Suppose a dynamic tracking error is described by

$$\dot{E} = f(E,t) \tag{22}$$

where $f(0,t) = 0$ for all t. If there exists scalar function $V(E,t)$ having continuous first partial derivative and satisfying the condition:

1) $V(E,t)$ is a positive definite
2) $\dot{V}(E,t)$ is a negative semidefinite,

then the equilibrium state at the origin is uniformly stable.

For the system of (19), an equilibrium state E_e is defined as $f(E,t) = 0, \forall t$. For nonlinear systems, there are one or more E_e. We denote a spherical region of radius r about an equilibrium state as $\parallel E - E_e \parallel \leq r$ as the Euclidean norm defined by

$$\parallel E - E_e \parallel = ((E_1 - E_{1e})^2 + \cdots + (E_p - E_{pe})^2)^{\frac{1}{2}}. \tag{23}$$

Let $S(\gamma)$ consist of all point such that $\parallel E - E_e \parallel \leq \gamma$ where $\gamma \geq \varepsilon$. The time derivative of Lyapunov function along any trajectory is

$$\begin{aligned}
\dot{V}(t) &= \frac{1}{2}\dot{E}^T PE + \frac{1}{2}E^T P\dot{E}. \\
&= \frac{1}{2}(AE + B(U(E|\aleph(\cdot)) - U(E|\aleph^*(\cdot)) - \varepsilon) + G)^T PE + \\
&\quad \frac{1}{2}E^T P(AE + B(U(E|\aleph(\cdot)) - U(E|\aleph^*(\cdot)) - \varepsilon) + G) \\
&= \frac{1}{2}(E^T A^T PE + E^T PAE) + \frac{1}{2}(B(\tilde{U} - \varepsilon) + G)^T PE + \\
&\quad \frac{1}{2}E^T P(B(\tilde{U} - \varepsilon) + G) \\
&= -\frac{1}{2}(E^T QE) + \frac{1}{2}((B(\tilde{U} - \varepsilon) + G)^T PE \\
&\quad +E^T P(B(\tilde{U} - \varepsilon) + G)) \\
&= -\frac{1}{2}(E^T QE) + (B(\tilde{U} - \varepsilon) + G)^T PE \\
&= -\frac{1}{2}(E^T QE) + (\tilde{U} - \varepsilon)^T B^T PE + G^T PE
\end{aligned} \tag{24}$$

where $\tilde{U} = U(E|\aleph(\cdot)) - U(E|\aleph^*(\cdot))$.

Theorem 2: Using prediction model of (2),control law given in (14), by a positive constant ε such that

$$\rho \leq -\frac{1}{2}(E^T QE) + (\tilde{U} - \varepsilon)^T B^T PE + G^T PE, \rho \leq 0 \tag{25}$$

then $\lim_{t\to\infty} E(t) = 0$, $E(t) \to 0$ at $t \to \infty$, the tracking error e will converge to zero.

According the Lyapunov theory, by $\dot{V}(t) \leq 0$, it implies that E is bounded by a positive constant ε that satisfies (25). From the convergence analysis based on Lyapunov theorem, the following can be conclude:

1) $\dot{V}(t)$ is actually total derivative of $V(t)$ with respect to t along solution of the system. By $\dot{V}(t) \leq 0$ implies $V(t)$ is decreasing function of t. By (25) with a positive constant ε, the closed-loop error trajectory (20) is a definite positif and non-increasing and by (19), $\dot{E}$ is also bounded. As a result, the QARXNN based adaptive control is stable and uniformly bounded. Therefore, $\lim_{t\to\infty} E(t) = 0$, $E(t) \to 0$ at $t \to \infty$, the tracking error of closed-loop system e will converge to zero.
2) For linear robust control, A is a nonsingular matrix then there exists one equilibrium state. Therefore $\dot{V}(t) \leq -\frac{1}{2}(E^T QE), \forall t$, that implies $\lim_{t\to\infty} E(t) = 0$, $E(t) \to 0$ at $t \to \infty$, the tracking error of closed-loop system e will converge to zero for all time.

According **Theorem 2:**, a switching line is used to change control action between linear part controller and nonlinear part controller. The proposed model only with linear parameters has to work until the use of nonlinear parameters does not damage the stability of closed loop system. Therefore, the controller with using linear parameters $\hat{\theta}$ will work all the time, but the nonlinear parameters $\hat{\aleph}(\xi(t))$ will work under the switching sequence. The control law (14) works under the switching line

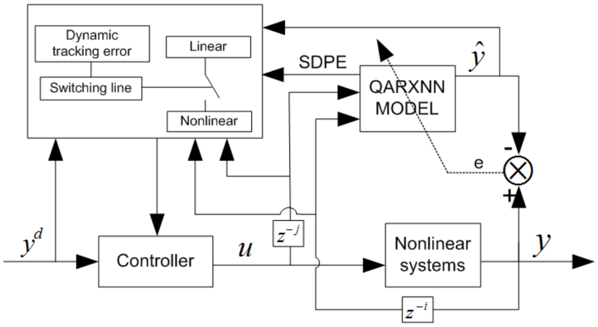

Fig. 2. Nonlinear adaptive predictive controller based on QARXNN prediction model. $i = 1, \cdots, n_y, j = 1, \cdots, n_u$

as follows:

$$\chi(t) = \begin{cases} 1, & \text{if } \rho \leq 0 \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

For the system (2), a nonlinear predictive controller based QARXNN model contains a feedback controller, QARXNN predictive model, and a switching mechanism shown by Fig. 2. Here, the feedback controller is performed based on the dynamics tracking error (19) with Lyapunov stability theorem (20), (24). By using QARXNN prediction model with two parameters of linear and nonlinear, two controllers also can be obtained with switching mechanism of (12), (14),(25),and (26). In the following, the design algorithm of the proposed control law can be summarized as follows.

Step 1.Identify the system under QARXNN model described in Section 2.

Step 2.Find the estimated parameter of SDPE by using simplified QARXNN prediction model

Step 3.By using SDPE, calculate dynamics tracking error shown by dynamic matrix of A in (17); and by introducing ε, find a new state of dynamics tracking error in (19) to obtain stability region with specific ε ($\varepsilon \leq \gamma$).

Step 4.Check the stability of NRAC controller by satisfying (25) and switching line of (26).

Step 5.Calculate controller signal using (12), two controller can be obtained by using the linear and nonlinear parts parameters of SDPE via switching mechanism in (14),(25),and (26).

Step 6.Goto **Step 1.**.

IV. Simulation Result

In this section, an illustrative example is provided to demonstrate the performance of the proposed controller.

Example: A nonlinear discrete-time dynamical system mixed with external disturbances given by [13] and [14] is observed. The system model is stated as follows:

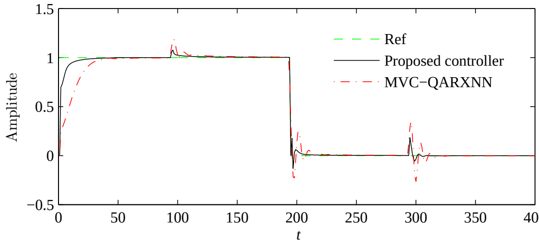

Fig. 3. Output responses (under external disturbances)

$$\begin{aligned} y(t) &= 0.9722y(t-1) + 0.378u(t-1) - 0.1295u(t-2) \\ &- 0.3103y(t-1)u(t-1) - 0.04228y^2(t-2) \\ &+ 0.1663y(t-2)u(t-2) - 0.03259y^2(t-1)y(t-2) \\ &- 0.3513y^2(t-1)u(t-2) \\ &+ 0.3084y(t-1)y(t-2)u(t-2) \\ &+ 0.1087y(t-2)u(t-1)u(t-2) + \omega(t). \end{aligned} \tag{27}$$

The reference input and the external disturbances $\omega(t)$ are given by

$$y^d(t) = \begin{cases} 1, & 0 < t \leq 200 \\ 0, & 200 < t \leq 400 \end{cases} \tag{28}$$

$$\omega(t) = \begin{cases} 0, & 0 < t \leq 100 \\ 0.05, & 100 < t \leq 300 \\ 0.2, & 300 < t \leq 400 \end{cases} \tag{29}$$

From the system model (27), an embedded system MLPNN of QARXNN is constructed with three layer neural network. The input vector of $\phi(t)$ is specified by $\phi(t) = [y(t-1)\ y(t-2)\ u(t-1)\ u(t-2)]^T$ with $n_u = 2$ and $n_y = 2$, the number of input nodes, hidden nodes, and output nodes is also the same as $n = n_u + n_y$. Constant learning rate of BP algorithm is selected by $\eta_{bp} = 0.1$ and gain of adaptive tracking control based on QARXNN model are given by: $\gamma = 0.02$, $p = 2$, $Q = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$. To test the robust characteristics of the proposed controller, this example is performed in which the system is mixed with external disturbances. Fig. 3, Fig. 4, Fig. 5, and Fig. 6 show the output responses, the control signals, the tracking errors, and the switching sequences of the proposed controller compared with our previous work of minimum variance controller baased on QARXNN (MVC-QARXNN). With the output response and error shown by Fig. 3 and Fig. 5, it indicates that the proposed controller can adapt the external disturbance mixed in nonlinear system. The details of comparison are summarized in Table I. As can be seen, the performance of the proposed controller is better than those that can be obtained with the other controllers.

From the simulation results, Fig. 6 shows that the amount of time switching to linear controller is less using the proposed controller compared to the use of MVC-QARXNN that uses the error-based switching control. It can conclude that the proposed switching technique is more effective. The more time

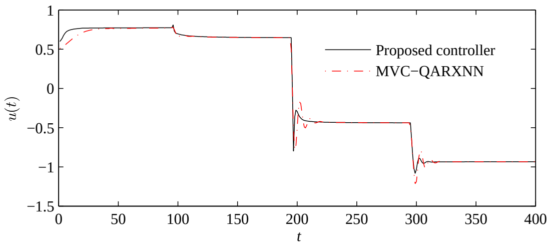

Fig. 4. Control signals

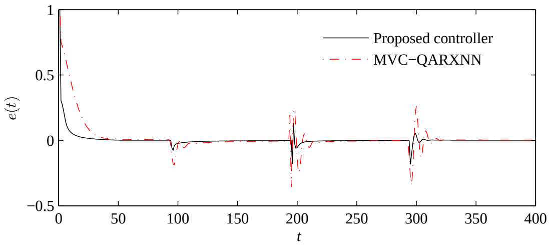

Fig. 5. Tracking error

switch to nonlinear controller, the accuracy of the control system will be increased as well. The use of MVC-QARXNN-based control switch to linear control longer and more often.

V. Discussion and Conclusion

This paper presents an adaptive controller based on QARXNN. The emphasis that will discuss in this paper is the improvement of the controller accuracy by reducing the unnecessary switching to linear controller. First, the principles of quasi-ARX prediction model is derived. The SDPE is executed using an embedded system of MLPNN for nonlinear part and using LSE algorithm for linear part. The linear parameter estimation is then set as bias vector for output nodes of MLPNN. Second, the tracking error of closed-loop system is derived via SDPE, the stability analysis of closed-loop system is performed using Lyapunov stability theory. Third, the switching line between linear and nonlinear parts of SDPE is conducted to guarantee boundedness and convergence of error. The main contributions of this study are: (1) the successful development of a quasi-ARX neural network based adaptive control. It has two advantages, reducing unnecessary switching to linear controller to improve accuracy and simplify the switching formula by utilizing SDPE as switching variable. (2) the successful application of QARXNN based adaptive predictive control for nonlinear system mixed with noise.

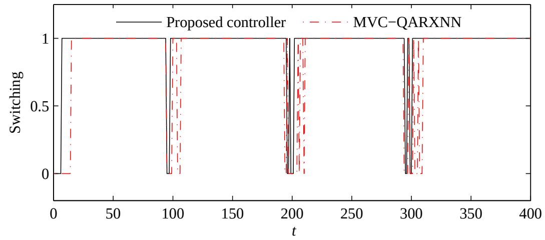

Fig. 6. Switching sequence (0: linear; 1: nonlinear)

TABLE I
Simulation results of the control systems

Controllers	Network Parameters	RMS error
Proposed Controller	36	0.0602
SPC [14]	24	0.0866 *
Fuzzy-based GPC [14]	24	0.1192 *
MVC-QARXNN [11]	36	0.1271
GPC [14]	0	0.1649 *

* The results are listed in the original papers.

Acknowledgment

The first author would like to thank to Indonesian Government Scholarship Directorate General of Higher Education (DGHE) (Beasiswa Luar Negeri DIKTI - Kementerian Pendidikan dan Kebudayaan Republik Indonesia) for the scholarship and Politeknik Perkapalan Negeri Surabaya (Shipbuilding Institute of Polytechnic Surabaya).

References

[1] T. Chai, Y. Zhang, H. Wang, C. Y. Su, and J. Sun, "Data-based virtual unmodeled dynamics driven multivariable nonlinear adaptive switching control," *IEEE Transactions on Neural Networks*, vol. 22(12), pp. 2154–2172, 2011.

[2] F. J. Lin, R. J. Wai, and H. P. Chen, "A hybrid computed torque controller using fuzzy neural network for motor-quick-return servo mechanism," *IEEE/ASME Trans. on Mechatronics*, vol. 6(1), pp. 75–89, 2001.

[3] C.-L. Hwang and C.-Y. Kuo, "A stable adaptive fuzzy sliding-mode control for affine nonlinear systems with application to four-bar linkage systems," *IEEE Trans. on Fuzzy Systems*, vol. 9(2), pp. 868–879, 2001.

[4] C. M. Lin and C. F. Hsu, "Neural-network-based adaptive control for induction servomotor drive system," *IEEE Trans. Ind. Electron.*, vol. 49(1), pp. 238–252, 2002.

[5] R.-J. Wai, "Hybrid fuzzy neural-network control for nonlinear motor-toggle servomechanism," *IEEE Trans. on Control Systems Tech.*, vol. 10(4), pp. 519–532, 2002.

[6] K. Nam, "Stabilization of feedback linearizable systems using a radial basis function network," *IEEE Trans. on Automatic Control*, vol. 44(5), pp. 1026–1031, 1999.

[7] J. Hu, K. Kumamaru, and K. Hirasawa, "A Quasi-ARMAX approach to modelling of non-linear systems," *Int. J. Control*, vol. 74(18), pp. 1754–1766, 2001.

[8] Y. Cheng, L. Wang, and J. Hu, "Quasi-ARX wavelet network for SVR based nonlinear system identification," *Nonlinear Theory and its Applications (NOLTA), IEICE*, vol. 2(2), pp. 165–179, 2011.

[9] M. A. Jami'in, I. Sutrisno, and J. Hu, "Lyapunov learning algorithm for quasi-ARX neural network to identification of nonlinear dynamical system," in *Proc. IEEE International Conference on Systems, Man, and Cybernetics (Seoul)*, 2012, pp. 3141–3146.

[10] M. A. Jami'in, I. Sutrisno, and J. Hu, "Deep searching for parameter estimation of the linear time invariant (LTI) system by using quasi-ARX neural network," in *Proc. IEEE International Joint Conference on Neural Network (Dallas)*, 2013, pp. 2759–2762.

[11] L. Wang, 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(4), pp. 246–252, 2010.

[12] J. Hu and K. Hirasawa, "A method for applying multilayer perceptrons to control of nonlinear systems," in *Proc. 9th International Conference on Neural Informassion Processing (Singapure)*, 2002.

[13] K. R. Sales and S. A. Billings, "Self-tuning control of nonlinear armax model," *Int. J. Control*, vol. 51, pp. 753–769, 1990.

[14] C.-H. Lu, "Design and application of stable predictive controller using recurrent wavelet neural networks," *IEEE Trans. on Industrial Electronics*, vol. 56(9), pp. 3733–3742, 2009.