IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING
IEEJ Trans 2016; **11**: 83–90
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI:10.1002/tee.22191

Paper

Quasi-ARX Neural Network Based Adaptive Predictive Control for Nonlinear Systems

Mohammad Abu Jami'in[*,**], Non-member
Jinglu Hu[*a], Member
Mohd Hamiruce Marhaban[***], Non-member
Imam Sutrisno[*,**], Non-member
Norman Bin Mariun[***], Non-member

In this paper, a new switching mechanism is proposed based on the state of dynamic tracking error so that more information will be provided −not only the error but also a one up to pth differential error will be available as the switching variable. The switching index is based on the Lyapunov stability theory. Thus the switching mechanism can work more effectively and efficiently. A simplified quasi-ARX neural-network (QARXNN) model presented by a state-dependent parameter estimation (SDPE) is used to derive the controller formulation to deal with its computational complexity. The switching works inside the model by utilizing the linear and nonlinear parts of an SDPE. First, a QARXNN is used as an estimator to estimate an SDPE. Second, by using SDPE, the state of dynamic tracking error is calculated to derive the switching index. Additionally, the switching formula can use an SDPE as the switching variable more easily. Finally, numerical simulations reveal that the proposed control gives satisfactory tracking and disturbance-rejection performances. Experimental results demonstrate its effectiveness.

Keywords: Quasi-ARX neural network, SDPE, dynamic tracking error, switching control, Lyapunov stability

Received 3 November 2014; Revised 18 March 2015

1. Introduction

If the dynamic model of a controlled system can be known exactly, then the ideal control can be calculated to obtain the desired reference trajectory. Generally, a linear system mixed with noise or nonlinear system results in uncertainty of the parameters of the system. It becomes difficult to solve these problems because the ideal control is unobtainable. Hence, to resolve these problems, a conventional linear and robust control has been adopted to consider the robustness and performance accuracy. However, by increasing the robustness, the control accuracy will be reduced. To maintain the control accuracy, nonlinear models such as the neural network (NN) and fuzzy models were used as identifiers for the controller design because they are generally applicable to systems with mathematically poor models [1,2]. An NN-based adaptive control is performed using analysis theory such as stability, robustness, and control accuracy. However, the major disadvantage is the lack of a systematic design of the control methodology [3,4].

To facilitate the controller design methodology, the technique of using a feedback linearization of a nonlinear system was proposed by using a nonlinear model [1,3,5–7]. In this paper, a quasi-ARX neural-network (QARXNN) model is proposed as an identifier. A QARXNN model is a nonlinear model that describes the system modeling by a linear relationship between the nonlinear coefficients and the regression vector. An NN is used to parameterize the regression vector, with the output being state-dependent parameter estimation (SDPE). Thus, the control law is derived directly by utilizing the transformation of its linear inverse.

It is a key point to guarantee the stability and improve the control accuracy for designing a control system. The use of a nonlinear controller improves the control accuracy. However, it is difficult to use only a nonlinear controller to guarantee the stability of the closed-loop controller because of the uncertainty of a nonlinear system [8,9]. To improve the tracking control performance, Zhang *et al.* [8] proposed a switching mechanism to guarantee the stability and to improve the control accuracy. The switching and tuning framework has been established for the adaptive control design with multiple models [8,9]. Two linear and nonlinear models are used with a switching mechanism [8]: (i) a linear controller-driven model with self-tuning parameters, and (ii) an estimator based on an adaptive network-based fuzzy-inference system (ANFIS) for unmodeled dynamics to design a nonlinear controller. Two linear and nonlinear estimators work under a switching mechanism, and the linear adaptive control is always stable to ensure the boundedness of the input and output of the closed-loop system, while the nonlinear controller improves the control accuracy.

The switching index between the linear and nonlinear controller was proposed by some researchers [2,8,10,11]. The switching rule is based on the convergence index of error, which is a function of the estimation error of the linear and nonlinear adaptive controllers. It works by comparing the convergence index of the linear and nonlinear parts, which activates the switching mechanism to switch on the controller with the smallest minimum index. However, with such a switching mechanism, it is difficult to obtain more information from the error vector to determine the stability of

[a] Correspondence to: Jinglu Hu. E-mail: jinglu@waseda.jp

[*] Graduate School of Information Production and Systems, Waseda University, Kitakyushu 808-0135, Japan
[**] Politeknik Perkapalan Negeri Surabaya, Jalan Teknik Kimia, Kampus ITS Sukolilo, Surabaya 60111, Indonesia
[**] CAPER, Center of Advanced Power and Energy Research, Universiti Putra Malaysia, Serdang 43400, Malaysia

the closed-loop control system. Therefore, such a switching rule is not efficient, because unnecessary switching to the linear controller will be longer and more frequent. Thus, the accuracy of the control system becomes poor.

In this paper, a new switching mechanism is proposed based on the state of dynamic tracking error so that more information will be provided, and not only error but also one up to pth differential error will be available as the switching variable. The switching mechanism is derived based on Lyapunov stability theorem by utilizing the state parameter of dynamic tracking error obtained from the prediction model. Therefore, the switching mechanism becomes more effective and efficient. Moreover, the proposed switching formula can use the parameter of prediction model presented by SDPE as variables of the switching condition criterion.

A QARXNN model can be simplified as a linear correlation between the input vector and its coefficients. An embedded system is a subsystem used to parameterize the regression or the input vector. A nonlinear model such as a feedforward NN, a neuro-fuzzy network, or a wavelet network can be used as an embedded system. The output is the coefficient of the regression vector called an SDPE [12–14]. The difference between using an NN as an embedded system for QARXNN models and the others is that the bias vector of the output nodes of an NN is from the estimated parameters of the linear estimator. With one prediction model of the QARXNN, we have two estimators: linear and nonlinear. A linear parameter estimator (LPE) is estimated using a least-square error (LSE) algorithm. It is set as the bias vector for the output nodes of an NN. The switching mechanism works to select the linear or nonlinear estimators based on the proposed switching condition index. The controller is derived from the parameter of the selected estimator. The switching condition index is used only to check the stability condition of a nonlinear controller based on the Lyapunov Stability Theorem and to switch to the linear controller if it is not stable, and vice versa.

By using the proposed switching law, the controller comprises a linear-robust-adaptive controller (LRAC), a nonlinear-robust-adaptive controller (NRAC), and a switching mechanism. An NRAC controller is designed based on a nonlinear estimator, whereas an LRAC is designed by using a linear estimator. At the beginning, 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 are used to estimate an SDPE by the next regression input. From the estimated parameters of the linear and nonlinear parts, the dynamic tracking error is derived. The stability of the overall system is then verified by the Lyapunov theorem so that ultimately bounded tracking is accomplished.

The main contributions of this paper are summarized as follows. (i) A new controller based on a simplified QARXNN predictive model is constructed to deal with its computational complexity for controlling a nonlinear system mixed with external disturbances. (ii) A new switching rule based on the Lyapunov stability theorem utilizing the state parameters of dynamic tracking error is proposed so that the controller can work more effectively and efficiently. (iii) Simulation results are given to demonstrate the effectiveness of the proposed approach.

2. Quasi-ARX Neural Network Model

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

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

where $g(\cdot)$, $\phi(t) = [y(t-1) \cdots y(t-n_y)\ u(t-1) \cdots u(t-n_u)]^{\mathrm{T}}$, $y(t) \in R$ is the unknown nonlinear function, regression or input vector, and system output, and $t = 1, 2, \cdots$ denotes the sampling of time. By using a Taylor expansion series and system dynamics, the nonlinear system (1) can be presented as a linear correlation between a nonlinear coefficient (Taylor coefficient) and its regression or input vector, described as follows [2,12,14]:

$$y(t) = \phi^{\mathrm{T}}(t)\aleph(\xi(t)). \tag{2}$$

where $\aleph(\xi(t)) = [a_{(1,t)} \cdots a_{(n_y,t)}\ b_{(1,t)} \cdots b_{(n_u,t)}]^{\mathrm{T}}$ denotes the output of an embedded submodel to parameterize the regression vector. $\xi(t) = [y(t-1) \cdots y(t-n_y)\ u(t-2) \cdots u(t-n_u)\ \nu(t)]^{\mathrm{T}}$ and $\nu(t)$ are the input of an embedded system injected into a QARXNN model and a virtual input, respectively. Incorporated into an NN selected as an embedded system, a QARXNN model is rewritten as

$$\begin{aligned} y(t) &= \phi^{\mathrm{T}}(t)\aleph(\xi(t)) \\ \aleph(\xi(t), \Omega) &= W_2 \Gamma W_1(\xi(t)) + \theta \qquad (3) \\ &= \delta(\xi(t)) + \theta \qquad (4) \end{aligned}$$

where $\Omega = \{W_1, W_2, \theta\}$ are the network parameters, and Γ is the diagonal nonlinear operator with identical sigmoidal elements on hidden nodes.

In (3) and (4), we define the system model with two linear θ and nonlinear $\delta(\xi(t))$ parameters. θ is a bias vector of the output nodes of an NN. The difference with the other NN is that θ is the linear parameter, i.e. estimated based on the linear estimator, that uses an LSE algorithm. The coefficient $\aleph(\xi(t))$ of an NN is composed hierarchically based on the following identification scheme: First, the system is estimated under a linear model using a least-square error (LSE) algorithm. Second, θ is set as the bias vector of the output nodes of an NN, which is an embedded system of a QARXNN model to parameterize the regression vector. The linear parameter (LP) is estimated by using the LSE algorithm with the output predictor, described by the following:

$$\begin{aligned} y_L(t) &= a_{(L,1)}y(t-1) + a_{(L,2)}y(t-2) \\ &\quad + a_{(L,n_y)}y(t-n_y) + b_{(L,1)}u(t-1) \\ &\quad + b_{(L,2)}u(t-2) + b_{(L,n_u)}u(t-n_u). \\ y_L(t) &= \phi^{\mathrm{T}}(t)\theta \qquad (5) \end{aligned}$$

where $\theta = [a_{(L,1)} \cdots a_{(L,n_y)}\ b_{(L,1)} \cdots b_{(L,n_u)}]^{\mathrm{T}}$ is the linear-parameter estimation, which is set as a bias vector for MLPNN. By incorporating $\aleph(\xi(t))$ to ensure the stability and control accuracy, we divide the linear and nonlinear parts of the SDPE equipped with a switching mechanism. When performing a switching mechanism, two linear and nonlinear estimators will be available: (i) a linear estimator with the estimated parameter θ, and (ii) a nonlinear estimator with the estimated parameter $\aleph(\xi(t))$. The model in (2) can be rewritten as follows:

$$\begin{aligned} y(t) &= \phi^{\mathrm{T}}(t)(\delta(\xi(t)) + \theta) \\ &= \phi^{\mathrm{T}}(t)\delta(\xi(t)) + \phi^{\mathrm{T}}(t)\theta \qquad (6) \end{aligned}$$

The details of the algorithm of the QARXNN model can be found in Refs [12,15,16]. A QARXNN model with an MLPNN set as an embedded system is shown in Fig. 1. In our main theoretical result, the following assumptions are made:

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

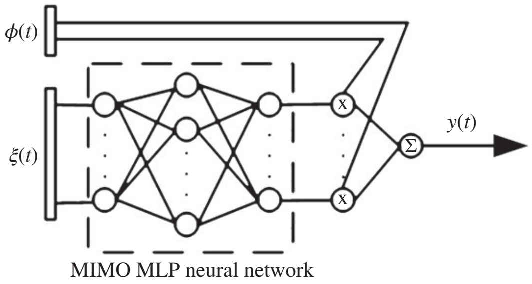

Fig. 1. Quasi-ARX neural network model with an embedded system of neural network

3. Control Strategy

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

$$\begin{aligned} y(t) &= \hat{a}_{(1,t)}y(t-1) + \hat{a}_{(2,t)}y(t-2) \\ &\quad + \hat{a}_{(n_y,t)}y(t-n_y) + \hat{b}_{(1,t)}u(t-1) \\ &\quad + \hat{b}_{(2,t)}u(t-2) + \hat{b}_{(n_u,t)}u(t-n_u). \end{aligned} \tag{7}$$

where $\hat{\aleph}(\xi(t)) = [\hat{a}_{(1,t)} \cdots \hat{a}_{(n_y,t)}\ \hat{b}_{(1,t)} \cdots \hat{b}_{(n_u,t)}]^{\mathrm{T}}$ is a state-dependent parameter estimation. To derive the control signal, the model in (7) can be rewritten as

$$u(t-1) = \frac{1}{\hat{b}_{1,t}}(y(t) + g(t)) \tag{8}$$

$$\begin{aligned} g(t) &= -\hat{a}_{(1,t)}y(t-1) - \hat{a}_{(2,t)}y(t-2) \\ &\quad -\hat{a}_{(n_y,t)}y(t-n_y) - \hat{b}_{(2,t)}u(t-2) \\ &\quad -\hat{b}_{(n_u,t)}u(t-n_u)). \end{aligned} \tag{9}$$

If the model in (2) is rewritten by (7), it satisfies the input–output mapping of the system, and the assumptions **A1–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 steps ahead of the prediction, described as follows:

$$y(t+d) = \phi^T(t+d)\hat{\aleph}(\xi(t+d)) \tag{10}$$

where $\hat{\aleph}(\xi(t+d)) = [\hat{a}_{(1,t+d)} \cdots \hat{a}_{(n_y,t+d)}\ \hat{b}_{(1,t+d)} \cdots \hat{b}_{(n_u,t+d)}]^{\mathrm{T}}$ 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 steps ahead of the prediction, and $\xi(t+d) = [y(t+d-1)\ y(t+d-2) \cdots y(t+d-n_y)\ u(t+d-2)\ u(t+d-3) \cdots u(t+d-n_u-1)\ v(t+d)]^{\mathrm{T}}$. The online step ahead of the prediction, d, is equal to 1. From (10), we have the following:

$$u(t) = \frac{1}{\hat{b}_{1,t+1}}(y(t+1) + g(t+1)) \tag{11}$$

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

where $u(t)$ is a control signal corresponding to a nonlinear estimator $\hat{\aleph}(\xi(t))$. For the control signal calculated by using a linear predictor, $\hat{\aleph}(\xi(t))$ is replaced with $\hat{\theta}$.

By using a nonlinear estimator, the control accuracy can be maintained. However, it is difficult for the control signal calculated based on a nonlinear estimator to guarantee the stability of the closed-loop controller. Therefore, a linear estimator is used to keep the closed-loop stability [2,8,10]. Thus, the switching line is introduced between the linear part θ and the nonlinear part $\delta(\xi(t))$ of an SDPE, described as follows:

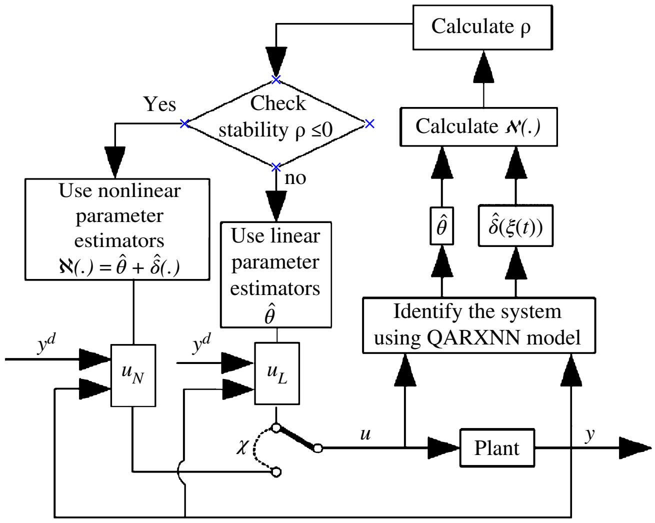

Fig. 2. Switching mechanism using the linear and nonlinear parameter estimators

$$\hat{\aleph}(\xi(t)) = \hat{\theta} + \chi(t)\hat{\delta} \tag{13}$$

$$u(t) = \chi(t)u_n + (1-\chi(t))u_l(t) \tag{14}$$

where u_l is a control signal calculated by the linear robust control that uses the parameters of the linear estimator $\hat{\theta}$, and u_n is a control signal from the nonlinear robust control that uses the parameters of the nonlinear estimator by summing $\hat{\theta}$ and $\hat{\delta}(\xi(t))$. $\chi(t)$ is a switching line, with $\chi(t) = 1$ denoting nonlinear robust control and $\chi(t) = 0$ denoting linear robust control, which is shown by Fig. 2.

4. Switching Condition

The use of a nonlinear estimator-based control can improve the control accuracy, but it is difficult to ensure closed-loop stability. The use of a linear estimator-based control can ensure the closed-loop stability, but it is low in accuracy. To improve the overall control performance, a switching condition is used to monitor the stability of the closed-loop system at all times when using a nonlinear controller. Therefore, the analysis of the switching conditions is placed on the use of the nonlinear controller. This proposed switching rule is based on the stability of the dynamic tracking error, defined as follows:

$$\begin{aligned} E(t) &= (e(t), \dot{e}(t), \ddot{e}(t), \cdots, \dot{e}_{p-1}(t)), \\ e(t) &= y(t) - y^d(t), \\ \dot{e}(t) &= \frac{\partial e(t)}{\partial t} = (e(t) - e(t-1))/\Delta t, \\ &\vdots\ , \\ \dot{e}_{p-1}(t) &= (e(t-p+2) - e(t-p+1))/\Delta t, \end{aligned} \tag{15}$$

where $y^d(t)$ is the reference input trajectory. The tracking error vector is described as follows:

$$\begin{aligned} \dot{e}(t) &= \frac{\partial e(t)}{\partial t} = (e(t) - e(t-1))/\Delta t, \\ &= ((y(t) - y^d(t)) - (y(t-1) - y^d(t-1)))/\Delta t \\ &= (\triangle y(t) - \triangle y^d(t))/\Delta t \simeq \dot{y}(t) - \dot{y}^d(t), \end{aligned} \tag{16}$$

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 follows:

$$\begin{aligned}
\dot{y}_p(t) &= \dot{y}_p^d(t) + K^{\mathrm{T}} E(t) \\
\dot{y}_p(t) - \dot{y}_p^d(t) &= -k_p \dot{e}_{p-1}(t) - k_{p-1} \dot{e}_{p-2}(t) - \cdots - k_1 e(t) \\
\dot{e}_p &= -k_p \dot{e}_{p-1}(t) - k_{p-1} \dot{e}_{p-2}(t) - \cdots - k_1 e(t) \\
0 &= \dot{e}_p + k_p \dot{e}_{p-1}(t) + k_{p-1} \dot{e}_{p-2}(t) + \cdots + k_1 e(t)
\end{aligned} \tag{17}$$

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

We define a nonlinear controller-estimation error as

$$\begin{aligned}
u(t) - u^*(t) &= \frac{1}{\hat{b}_{1,t+1}} (y(t+1) + \hat{g}(t+1)) \\
&\quad - \frac{1}{\hat{b}_{1,t+1}} (y^d(t+1) + g(t+1)) \\
&= \frac{1}{\hat{b}_{1,t+1}} (y(t+1) - y^d(t+1) \\
&\quad + \hat{g}(t+1) - g(t+1)) \\
U(t) &= \frac{1}{\hat{b}_{1,t+1}} (e(t+1) + G)
\end{aligned} \tag{18}$$

where $U = u(\cdot) - u^*(\cdot)$, $G = \hat{g}(\cdot) - g(\cdot)$, $\hat{g}(\cdot)$ are calculated using nonlinear predictor of QARXNN model. The error tracking can be obtained as follows:

$$\begin{aligned}
e(t+1) &= y(t+1) - y^d(t+1) = \hat{b}_{1,t+1} U(t) - G(t+1) \\
\dot{e}(t+1) &= e(t+1) - e(t) \\
&= \hat{b}_{1,t+1} U(t) - \hat{b}_{1,t} U(t-1) - G(t+1) + G(t) \\
\ddot{e}(t+1) &= \dot{e}(t+1) - \dot{e}(t) \\
&= \hat{b}_{1,t+1} U(t) - 2\hat{b}_{1,t} U(t-1) \\
&\quad + \hat{b}_{1,t-1} U(t-2) - G(t+1) + 2G(t) - G(t-1) \\
\dot{e}_3(t+1) &= \dot{e}_2(t+1) - \dot{e}_2(t) \\
&= \hat{b}_{1,t+1} U(t) - 3\hat{b}_{1,t} U(t-1) \\
&\quad + 3\hat{b}_{1,t-1} U(t-2) - \hat{b}_{1,t-2} U(t-3) \\
&\quad - G(t+1) + 3G(t) - 3G(t-1) + G(t-2) \\
\dot{e}_4(t+1) &= \dot{e}_3(t+1) - \dot{e}_3(t) \\
&= \hat{b}_{1,t+1} U(t) - 4\hat{b}_{1,t} U(t-1) \\
&\quad + 6\hat{b}_{1,t-1} U(t-2) - 4\hat{b}_{1,t-2} U(t-3) \\
&\quad + \hat{b}_{1,t-3} U(t-4) - G(t+1) + 4G(t) \\
&\quad - 6G(t-1) + 4G(t-2) - G(t-3).
\end{aligned} \tag{19}$$

Using (17) and (19), the dynamic tracking error can be stated as follows:

$$\dot{E} = AE + BU + G \tag{20}$$

where

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

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

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

$$G = \begin{pmatrix} -G(t+1) \\ -G(t+1) + G(t) \\ \vdots \\ -c_1 G(t+3-p) + \cdots + (-1)^{p+1} c_{p+1} G(t+2-p) \end{pmatrix}$$ where A is a nonsingular matrix and c_n is is a binomial series coefficient such as $\binom{p}{r} = \frac{p!}{r!(p-r)!}, 0 \le r \le p$.

Using (17) and (20), we can calculate K such that the roots of the characteristic equation (20) 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 follows:

$$\varepsilon = |u^* - u(E|\aleph^*(\cdot))|. \tag{21}$$

The controller's objective is to maintain the 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 (20) is a bounded by $(|U| < \varepsilon)$, then there are will be a positive real number of ε. By introducing ε in (20), it will be as follows:

$$\dot{E} = AE + B(U(E|\aleph(\cdot)) - U(E|\aleph^*(\cdot)) - \varepsilon) + G. \tag{22}$$

Consider a Lyapunov function

$$V(t) = \frac{1}{2} E^T P E \tag{23}$$

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 to make the system uniformly stable. Therefore, we require $\dot{V}(t) = -\dot{E}^T Q E$ to be negative semidefinite, which implies $V(t) \le V(0)$. The negative semidefinite matrix Q is given by

$$Q = -(A^T P + PA) \tag{24}$$

Theorem 1: Suppose a dynamic tracking error is described by the following:

$$\dot{E} = f(E, t) \tag{25}$$

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

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

then the equilibrium state at the origin is uniformly stable. To prove it, we note any trajectory of E such that

$$V(E, t) = V(E, 0) + \int_0^t \dot{V}(E, \tau) d\tau. \tag{26}$$

$\dot{V}(E, t)$ is negative semidefinite; hence, $V(E, t)$ is nonincreasing along the corresponding trajectory.

For the system (22), 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 $\| E - E_e \| \leq r$ and the Euclidean norm defined by

$$\| E - E_e \| = ((E_1 - E_{1e})^2 + \cdots + (E_p - E_{pe})^2)^{\frac{1}{2}}. \quad (27)$$

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

$$\begin{aligned}
\dot{V}(t) &= \frac{1}{2}\dot{E}^{\mathrm{T}}PE + \frac{1}{2}E^{T}P\dot{E} \\
&= \frac{1}{2}(AE + B(U(E|\aleph(\cdot)) - U(E|\aleph^{*}(\cdot)) - \varepsilon) + G)^{\mathrm{T}}PE \\
&+ \frac{1}{2}E^{\mathrm{T}}P(AE + B(U(E|\aleph(\cdot)) - U(E|\aleph^{*}(\cdot)) - \varepsilon) + G) \\
&= \frac{1}{2}(E^{\mathrm{T}}A^{\mathrm{T}}PE + E^{\mathrm{T}}PAE) + \frac{1}{2}(B(\tilde{U} - \varepsilon) + G)^{\mathrm{T}}PE \\
&+ \frac{1}{2}E^{T}P(B(\tilde{U} - \varepsilon) + G) \\
&= -\frac{1}{2}(E^{T}QE) + \frac{1}{2}((B(\tilde{U} - \varepsilon) + G)^{\mathrm{T}}PE \\
&+ E^{\mathrm{T}}P(B(\tilde{U} - \varepsilon) + G)) \\
&= -\frac{1}{2}(E^{\mathrm{T}}QE) + (B(\tilde{U} - \varepsilon) + G)^{\mathrm{T}}PE \\
&= -\frac{1}{2}(E^{\mathrm{T}}QE) + (\tilde{U} - \varepsilon)^{T}B^{\mathrm{T}}PE + G^{\mathrm{T}}PE
\end{aligned} \quad (28)$$

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

Theorem 2: Using the prediction model (2), the control law given in (14), with the use of nonlinear parameter $\hat{\aleph}(.)$ and a positive constant ε, the switching condition is defined as

$$\rho \leq -\frac{1}{2}(E^{\mathrm{T}}QE) + (\tilde{U} - \varepsilon)^{\mathrm{T}}B^{T}PE + G^{\mathrm{T}}PE \quad (29)$$

where ρ is a switching condition that is obtained from the time derivative of a Lyapunov function. Therefore, $\lim_{t\to\infty} E(t) = 0$, $E(t) \to 0$ at $t \to \infty$, and the tracking error e will converge to zero.

The switching logic is based on the condition of guaranteeing the stability of the closed-loop controller. The control signal calculated by using nonlinear parameters sometimes breaks the stability of the closed-loop controller $\rho > 0$. We cannot control the unstable system. However, the use of a linear and robust adaptive control (LRAC) is always stable during the whole time. Therefore, to guarantee the stability of the closed-loop controller, an LRAC is used only when the use of a nonlinear and robust adaptive control destroys the stability of the closed-loop controller. According to the Lyapunov theory, the system is stable if the time derivative of the Lyapunov function is negative semidefinite, $\rho \leq 0$.

By $\dot{V}(t) \leq 0$, it implies that E is bounded by a positive constant ε that satisfies (29). From the convergence analysis based on the Lyapunov theorem, the following can be concluded:

1. $\dot{V}(t)$ is actually a total derivative of $V(t)$ with respect to t along the solution of the system. By $\dot{V}(t) \leq 0$, it implies $V(t)$ is a decreasing function of t. By (29) with a positive constant ε, the closed-loop error trajectory of (23) is positive definite and nonincreasing, and by (22), $\dot{E}$ 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 $t \to \infty$, and the tracking error of the closed-loop system e will converge to zero.
2. For linear robust control, A is a nonsingular matrix, and then there exists one equilibrium state. Therefore, $\dot{V}(t) \leq -\frac{1}{2}(E^{\mathrm{T}}QE), \forall t$ implies $\lim_{t\to\infty} E(t) = 0$, $E(t) \to 0$ at $t \to \infty$, and the tracking error of closed-loop system e will converge to zero for all time.

According to Theorem 2, a switching line is used to change control action between linear and nonlinear controllers. The proposed model with only linear parameters has to work until the use of nonlinear parameters does not damage the stability of closed-loop system. Therefore, the controller 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 as follows:

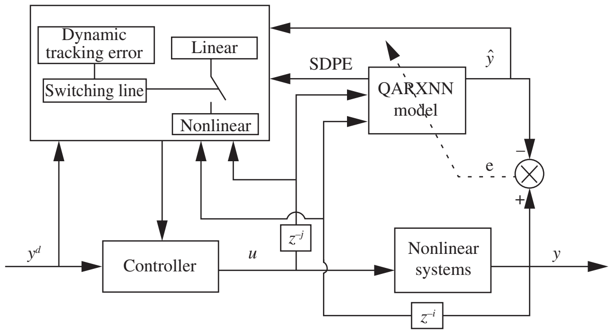

Fig. 3. Nonlinear adaptive predictive controller based on QARXNN prediction model. $i = 1, \cdots, n_y, j = 1, \cdots, n_u$

$$\chi(t) = \begin{cases} 1, & \text{if } \rho \leq 0 \\ 0, & \text{otherwise} \end{cases} \quad (30)$$

For the system (2), a nonlinear predictive controller based on the QARXNN model contains a feedback controller, a QARXNN predictive model, and a switching mechanism, as shown by Fig. 3. Here, the feedback controller performs based on the dynamic tracking error (22) with the Lyapunov stability theorem of (26) and (29). By using a QARXNN prediction model with the two linear and nonlinear estimators (5) and (2), two controllers perform with the switching mechanism of (14).

The switching mechanism selects the use of either the linear or nonlinear controller based on the index of ρ in (29) and as presented in Fig. 2. By $\rho > 0$, the closed-loop system is unstable using the nonlinear controller. The switching mechanism switches to using the linear controller and resets the nonlinear part $\delta(\xi(t))$ of an SDPE. In the following, the design algorithm of the proposed control law can be summarized as follows:

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

Step 2. Find the estimated parameter of an SDPE using the embedded system of the QARXNN prediction model.

Step 3. By using an SDPE, calculate the dynamic tracking error shown by the dynamic matrix of A in (20); and by introducing ε, find a new state of dynamic tracking error in (22) to obtain stability region with a specific ε ($\varepsilon \leq \gamma$).

Step 4. Check the stability of the NRAC controller by satisfying (29) and switching line of (30).

Step 5. Calculate the controller signal using (11); two controllers can be obtained by using the linear and nonlinear parts parameters of an SDPE via the switching mechanism in (14), (29), and (30).

Step 6. Go to **Step 1.**

5. Simulation Results

In this section, two illustrative examples are provided to demonstrate the performance of the proposed APC-QARXNN controller. The examples also show the effect of set-point changes and external disturbances on the control systems employing the proposed controller.

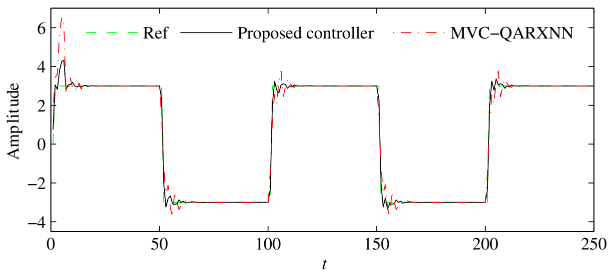

Fig. 4. Output responses

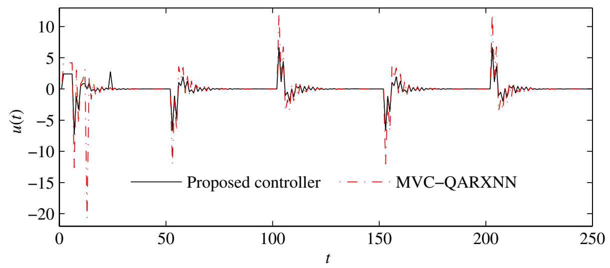

Fig. 5. Control signals

Example 1: Consider the control of a nonlinear discrete-time dynamical system with unstable zero dynamics given by [8,9]. The system model is described as follows:

$$\begin{aligned} y(t) \;&=\; 2.6y(t-1) - 1.2y(t-2) + u(t-1) \\ &+ 1.5u(t-2) + 0.5y(t-1)\mathtt{sin}(u(t-1) \\ &+ u(t-2) + y(t-1) + y(t-2)) \end{aligned} \quad (31)$$

The objective is to make the system output $y(t)$ track a reference input (desired output trajectory) $y^d(t)$ specified by the following:

$$y^d(t) = 3sign(sin(\pi t/50)), \quad 0 < t \leq 250. \quad (32)$$

From the system model (31), an embedded system MLPNN of QARXNN is constructed with a three-layer neural network. The input vector of $\phi(t)$ is specified by the following: $\phi(t) = [y(t-1)\; y(t-2)\; u(t-1)\; u(t-2)]^T$ and $n_u = 2$ and $n_y = 2$. The number of input nodes, hidden nodes, and output nodes is also the same as $n = n_u + n_y$. The constant learning rate of BP algorithm is selected by $\eta_{bp} = 0.1$, and the gain of adaptive tracking control based on the QARXNN model is given by $\gamma = 0.02$, $p = 2$, and $Q = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$. The output responses, the control signals, the tracking errors, and the switching sequences of the proposed controller compared with the MVC-QARXNN are shown in Figs 4–7. To evaluate the performance of the control system, one defines the root-mean-square (RMS) error as follows:

$$\text{RMS} = \sqrt{\frac{\sum_{t=1}^{N}(y(t) - y^d(t))^2}{N}} \quad (33)$$

where $y^d(t)$, $y(t)$, $t = 1, 2, \cdots, N$ are the desired output, the output of controlled system, and the time sampling, respectively, and N is the length of the input-output of controlled system. Figure 7 shows the switching sequence, where $\chi(t) = 1$ denotes the use of NRAC and $\chi(t) = 0$ denotes the use of an LRAC. In Table I, the RMS value of the proposed control system is less than that of the MVC-QARXNN-based control. As we can see, by using the same prediction model, the performance of the proposed controller is significantly better.

As we can see from Fig. 7, the use of nonlinear control u_n ($\chi(t) = 1$) is used almost everywhere in time t. However, linear control u_l is still used to ensure the closed-loop stability. The use only nonlinear control is difficult to ensure the close-loop control stability due to the uncertainty of the nonlinear system. We cannot control the unstable system. The result of using only u_n is shown in Fig. 8.

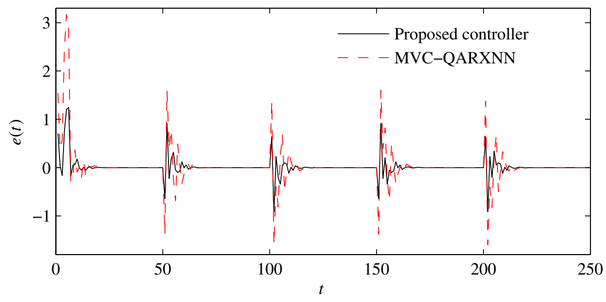

Fig. 6. Tracking error

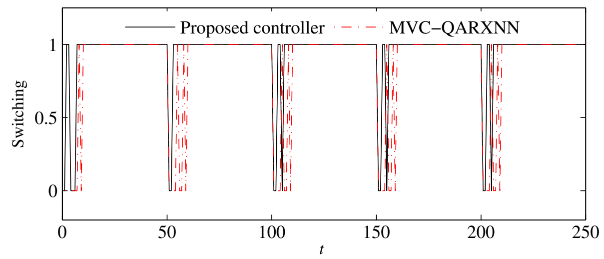

Fig. 7. Switching sequence (0: linear; 1: nonlinear)

Table I. Simulation results of the control systems

Controllers	Network parameters	RMS error
Proposed controller	36	0.201
MVC-QARXNN [2]	36	0.456

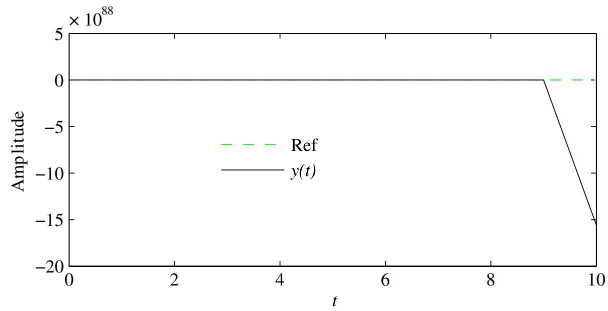

Fig. 8. Response of the closed-loop system by using only nonlinear control

Example 2: To further illustrate the applicability of APC-QARXNN proposed in this paper, a nonlinear discrete-time dynamical system mixed with external disturbances given by [17,18] is observed. The system model is stated as follows:

$$\begin{aligned} y(t) \;&=\; 0.9722y(t-1) + 0.378u(t-1) - 0.1295u(t-2) \\ &- 0.3103y(t-1)u(t-1) - 0.04228y^2(t-2) \\ &+ 0.1663y(t-2)u(t-2) - 0.03259y^2(t-1)y(t-2) \\ &- 0.3513y^2(t-1)u(t-2) \\ &+ 0.3084y(t-1)y(t-2)u(t-2) \\ &+ 0.1087y(t-2)u(t-1)u(t-2) + \omega(t). \end{aligned} \quad (34)$$

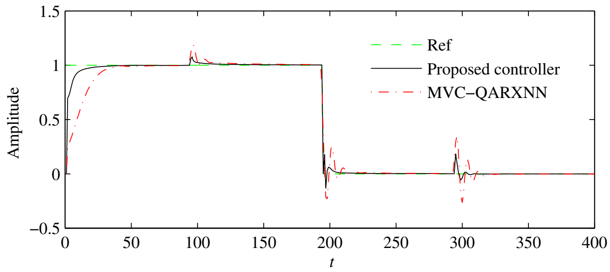

Fig. 9. Output responses (under external disturbances)

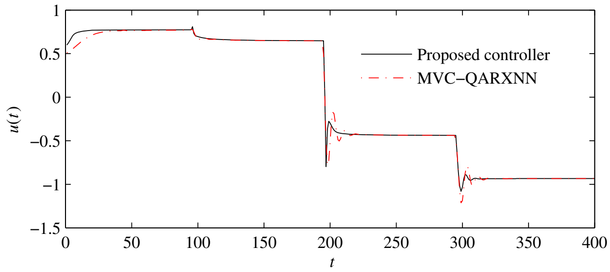

Fig. 10. Control signals

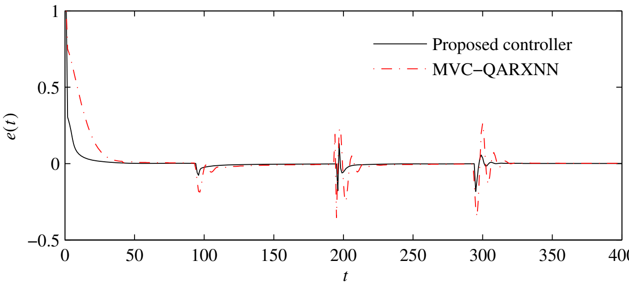

Fig. 11. Tracking error

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

$$y^d(t) = \begin{cases} 1, & 0 < t \leq 200 \\ 0, & 200 < t \leq 400 \end{cases} \quad (35)$$

$$\omega(t) = \begin{cases} 0, & 0 < t \leq 100 \\ 0.05, & 100 < t \leq 300 \\ 0.2, & 300 < t \leq 400 \end{cases} \quad (36)$$

The input variables of the QARXNN model and the gain of the adaptive tracking control are the same as in Example 1. To test the robust characteristics of the proposed controller, this example is performed in which the system is mixed with external disturbances. Figures 9–12 show the output responses, the control signals, the tracking errors, and the switching sequences of the proposed controller compared with the MVC-QARXNN based control. With the output response and error shown in Figs 9 and 11, we see that the proposed controller can adapt the external disturbance mixed in nonlinear system. The details of the comparison are summarized in Table II. As can be seen, the performance of the proposed controller is better than that of the other controllers.

From the simulation results, Figs 7 and 12 show that the amount of time switching to the linear controller is less by using the proposed controller compared to the previous one of MVC-QARXNN-based control. It can be concluded that the proposed switching technique is more effective. With more time to switch to the nonlinear controller, the accuracy of the control system will be increased as well. The use of MVC-QARXNN-based

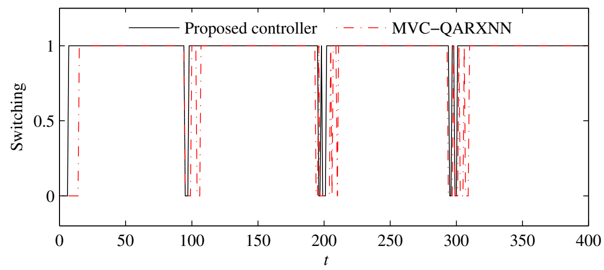

Fig. 12. Switching sequence (0: linear; 1: nonlinear)

Table II. Simulation results of the control systems

Controllers	Network parameters	*RMS error*
Proposed controller	36	0.0602
SPC [18]	24	0.0866[a]
Fuzzy-based GPC [18]	24	0.1192[a]
MVC-QARXNN [2]	36	0.1271
GPC [18]	0	0.1649[a]

[a]The results are listed in the original papers.

control switches to linear control longer and more often. This is because the switching based on error vector does not provide much information to determine the stability of nonlinear systems. As for switching techniques based on the state of dynamic error, it is possible to get more information about the stability of the closed-loop nonlinear controller. Therefore, the controller performance can be increased.

6. Discussion and Conclusion

This paper introduced an adaptive controller based on the prediction model of the quasi-ARX neural network (APC-QARXNN). The difference from our previous approach of MVC-QARXNN was shown by its controller strategy and switching rule. The switching based on an index-convergence error was not efficient, because it caused unnecessary switching to the linear controller. Therefore, with APC-QARXNN, a new switching rule based on the Lyapunov Stability Theorem was proposed by utilizing the state parameters of the dynamic tracking error so that the controller could work more effectively and efficiently. The advantages of using an APC-QARXNN are as follows: (i) A simplified QARXNN model presented by a state-dependent parameter estimation (SDPE) is used to derive the controller formulation to deal with its computational complexity. (ii) The control law can be derived easily from the model prediction based on the linearization technique, where 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. (iii) A Lyapunov stability-based switching control is performed to guarantee the closed-loop stability using the SDPE. The proposed switching rule improves the controller's accuracy by reducing unnecessary switching to the linear controller. The major contribution of this paper is the development of the QARXNN-based adaptive control with a new switching mechanism applied for a nonlinear system mixed with external disturbances. Finally, two numerical simulation results confirmed the theoretical analysis.

Acknowledgment

The first author would like to acknowledge an Indonesian Government scholarship from the Directorate General of Higher Education (DGHE) (Beasiswa Luar Negeri DIKTI - Kementerian

Pendidikan dan Kebudayaan Republik Indonesia) and Politeknik Perkapalan Negeri Surabaya (Shipbuilding Institute of Polytechnic Surabaya).

References

(1) Lin CM, Hsu CF. Neural-network-based adaptive control for induction servomotor drive system. *IEEE Transactions on Industrial Electronics* 2002; **49**(**1**):238–252.
(2) Wang L, Cheng Y, Hu J. A quasi-ARX neural network with switching mechanism to adaptive control of nonlinear systems. *SICE Journal of Control, Measurement, and System Integration* 2010; **3**(**4**): 246–252.
(3) Hwang C-L, Kuo C-Y. A stable adaptive fuzzy sliding-mode control for affine nonlinear systems with application to four-bar linkage systems. *IEEE Transactions on Fuzzy Systems* 2001; **9**(**2**): 868–879.
(4) Kamal E, Aitouche A, Ghorbani R, Bayart M. Robust nonlinear control of wind energy conversion systems. *International Journal of Electrical Power & Energy Systems* 2013; **44**:202–209.
(5) Lin FJ, Wai RJ, Chen HP. A hybrid computed torque controller using fuzzy neural network for motor-quick-return servo mechanism. *IEEE/ASME Transactions on Mechatronics* 2001; **6**(**1**):75–89.
(6) Wai R-J. Hybrid fuzzy neural-network control for nonlinear motor-toggle servomechanism. *IEEE Transactions on Control Systems Technology* 2002; **10**(**4**):519–532.
(7) Nam K. Stabilization of feedback linearizable systems using a radial basis function network. *IEEE Transactions on Automatic Control* 1999; **44**(**5**):1026–1031.
(8) Zhang Y, Chai T, Wang H, Fu J, Zhang L, Wang Y. An adaptive generalized predictive control method for nonlinear systems based on ANFIS and multiple models. *IEEE Transactions on Fuzzy Systems* 2010; **18**(**6**):1070–1081.
(9) Chen L, Narendra KS. Nonlinear adaptive control using neural networks and multiple models. *Automatica* 2001; **37**:1245–1255.
(10) Jami'in MA, Sutrisno I, Hu J. Nonlinear adaptive control for wind energy conversion systems based on quasi-ARX neural networks model. Procedings of the International MultiConference of Engineers and Computer Scientists 2014, March 12-14, 2014, Hong Kong, 2014.
(11) Chai T, Zhang Y, Wang H, Su CY, Sun J. Data-based virtual unmodeled dynamics driven multivariable nonlinear adaptive switching control. *IEEE Transactions on Neural Networks* 2011; **22**(**12**):2154–2172.
(12) Hu J, Kumamaru K, Hirasawa K. A Quasi-ARMAX approach to modelling of non-linear systems. *International Journal of Control* 2001; **74**(**18**):1754–1766.
(13) Cheng Y, Wang L, Hu J. Quasi-ARX wavelet network for SVR based nonlinear system identification. *Nonlinear Theory and its Applications (NOLTA), IEICE* 2011; **2**(**2**):165–179.
(14) Jami'in MA, Sutrisno I, Hu J. Lyapunov learning algorithm for quasi-ARX neural network to identification of nonlinear dynamical system. Proc. IEEE International Conference on Systems, Man, and Cybernetics, 14-17 October 2012, Seoul, 2012; 3141–3146.
(15) Jami'in MA, Sutrisno I, Hu J. Deep searching for parameter estimation of the linear time invariant (LTI) system by using quasi-ARX neural network. Proceedings of the 2013 IEEE International Joint Conference on Neural Network (IJCNN), 4-9 2013, Dallas, TX, 2013; 2759–2762.
(16) Hu J, Hirasawa K. A method for applying multilayer perceptrons to control of nonlinear systems. Proceedings of the 9th International Conference on Neural Informassion Processing, 18-22 November 2002, Singapore, 2002.
(17) Sales KR, Billings SA. Self-tuning control of nonlinear ARMAX models. *International Journal of Control* 1990; **51**:753–769.
(18) Lu C-H. Design and application of stable predictive controller using recurrent wavelet neural networks. *IEEE Transactions on Industrial Electronics* 2009; **56**(**9**):3733–3742.

Mohammad Abu Jami'in (Non-member) received the B.E. degree in Marine Engineering and the M.E. degree in Control System Engineering from Institut Teknologi Sepuluh Nopember (ITS) Surabaya, Indonesia, in 2000 and 2008, respectively. He is a Lecturer with Politeknik Perkapalan Negeri Surabaya, and is currently pursuing the Ph.D. degree at the Graduate School of Information, Production and Systems, Waseda University, Japan. His research interests include artificial intelligence and its applications.

Jinglu HU (Member) received the M.S. degree in Electronic Engineering from Zhongshan University, China, in 1986, and the Ph.D. degree in Computer Science and Engineering from Kyushu Institute of Technology, Japan, in 1997. From 1986 to 1993, he worked with Zhongshan University, where he was a Research Associate and then a Lecturer. From 1997 to 2003, he worked as a Research Associate at Kyushu University. From 2003 to 2008, he worked as an Associate Professor and since April 2008 has been a Professor with the Graduate School of Information, Production and Systems, Waseda University. His research interests include neurocomputing, evolutionary computation and their applications. Prof.Hu is a member of the IEEE, SICE, and IEICE.

Mohd Hamiruce Marhaban (Non-member) received the B.Eng. degree in Electrical and Electronic Engineering from the University of Salford, UK, in 1998, and the Ph.D. degree in Electronic Engineering from the University of Surrey, UK, in 2003. He is an Associate Professor with the Department of Electrical and Electronic Engineering, Universiti Putra Malaysia. His research interests include intelligent control system and computer vision.

Imam Sutrisno (Non-member) obtained the bachelor's and master's degrees in Control System Engineering from the Electrical Engineering Department, Institute of Technology 10th Nopember Surabaya, Indonesia, in 1998, and 2005, respectively. He is a Lecturer with Politeknik Perkapalan Negeri Surabaya, and is currently pursuing the Ph.D. degree at the Graduate School of Information, Production and Systems, Waseda University, Japan. His research interests include system identification and control systems design.

Norman Bin Mariun (Non-member) received the B.S. degree in Electrical and Electronic Engineering from The University of Nottingham, Nottingham, U.K., in 1980, the M.Sc. degree from North Carolina State University, Raleigh, USA, in 1983, and the Ph.D. degree from the University of Bradford, Bradford, U.K., in 1998. He is a Professor in Electrical Engineering with the Faculty of Engineering, Universiti Putra Malaysia, Serdang, Malaysia, where he is the Director of the Research Management Centre. His current research interest include power system quality and energy management, power electronics applications in power system and electrical drives, modeling and testing of power semiconductor devices, and application of multimedia in engineering education. Prof. Mariun is a Fellow of The Institution of Engineers, Malaysia, and a Registered Professional Engineer.