IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING
IEEJ Trans 2015; **10**: 368–375
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI:10.1002/tee.22095

Paper

Maximum Power Tracking Control for a Wind Energy Conversion System Based on a Quasi-ARX Neural Network Model

Mohammad Abu Jami'in[*,**], Non-member
Imam Sutrisno[*,**], Non-member
Jinglu Hu[*a]

By itself, a wind turbine is already a fairly complex system with highly nonlinear dynamics. Changes in wind speed can affect the dynamic parameters of wind turbines, thus rendering the parameters uncertain. However, we can identify the dynamics of the wind energy conversion system (WECS) online by a quasi-ARX neural network (QARXNN) model. A QARXNN presents a problem in searching for the coefficients of the regression vector (input vector). A multilayer perceptron neural network (MLPNN) is an embedded system that provides the unknown parameters used to parameterize the input vector. Fascinatingly, the coefficients of the input vector from prediction model can be set as controller parameters directly. The stability of the closed-loop controller is guaranteed by the switching of the linear and nonlinear parts of the parameters. The dynamic of WECS is derived with given parameters, and then a wind speed signal created by a random model is fed to the system causing uncertainty parameters and reducing the power that can be absorbed from wind. By using a minimum variance controller, the maximum power is tracked from WECS. From the simulation results, it is observed that the proposed controller is effective in tracking the maximum power of WECS. © 2015 Institute of Electrical Engineers of Japan. Published by John Wiley & Sons, Inc.

Keywords: wind energy conversion system (WECS), quasi-ARX neural network, nonlinear parameter estimation, switching controller

Received 27 February 2014; Revised 23 July 2014

1. Introduction

Concerns about both environmental damage caused by burning of fossil fuels and extreme fluctuation in oil prices have aroused worldwide interest in using alternative energy sources [1,2]. Among these sources, wind energy is a very promising and abundantly available alternative. Clean renewable energy sources (i.e. wind, photovoltaic (PV), and fuel cells) do not cause global warming, and according to the estimates by the European Wind Energy Association, tapping only 10% of available wind power can sufficiently meet the world's total electricity needs. Given the current technological advances in the field of power electronics, variable speed drives, and wind turbines, the cost of wind power can approach that of maintaining fossil-fuel power plants. The United States and Germany are world leaders in terms of using wind energy's installed capacity (25 GW), while in terms of the use of wind energy, Denmark reports the highest percentage, which is 20% of the total energy use. Though wind energy currently only accounts for 1% of all generated electricity, this figure is expected to rise to 20% by 2030 [2].

The control system plays an important role in extracting the maximum energy from wind. Maximum power point tracking (MPPT) control operates by varying the speed of the generator so that the turbine can operate at the point of maximum aerodynamic efficiency and, in turn, obtain maximum power extracted from wind [3]. The amount of power that can be converted depends on the MPPT accuracy, which is highly influenced by the accuracy of the control system. The controller is intended to maximize the power output of the turbine, regardless of the type of generator used.

The issues involved in the design of a nonlinear controller for MPPT concern the robustness of parameter uncertainty since the parameters are always subject to change according to the change of wind speed [4]. In conventional linear controls for robustness, increasing robustness can reduce the accuracy of the control system [5,6]. Several approaches have been developed to solve the control problems associated with wind energy. An adaptive neural network model-based estimator has been developed to estimate uncertain aerodynamics online, after which a tracking control law can be derived based on Lyapunov stability analysis [7]. The wind energy conversion system (WECS) can also be modeled as a linear system by using a linear parameter estimator (LPE), in which the controller uses a hybrid control unit of the linear quadratic Gaussian (LQG) and neurocontroller (NC) performed by a feed-forward radial basis function (RBF) model [8]. However, as we know, fuzzy systems, neural networks, and neurofuzzy systems are black-box models. The stability analysis of these models is difficult, and parameter tuning is generally a time-consuming process due to their nonlinear and multiparametric nature [9,10]. Among other drawbacks, LQG controllers designed with an LPE estimator also require additional estimation schemes using neurocontrol for systems with parameter uncertainties. In this paper, we propose a quasi-ARX neural network (QARXNN) prediction model to estimate the parameters of the input vector, and by performing the minimum variance control law, the estimated parameters are set as the controller parameters with the switching law [6].

With QARXNN, we assume that WECS is a nonlinear system in which nonlinearity is placed on the parameter estimates. The estimated parameters have a linear relationship with the regression vector, which makes it easy to derive the control law from the proposed model. The linear parts of the parameters are used the whole time, while the nonlinear parts of the parameters work under switching function. The use of nonlinear parameters can

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

[*] Graduate School of Information Production and Systems, Waseda University, Hibikino 2-7, Wakamatsu-ku, Kitakyushu-shi, Fukuoka 808-0135, Japan

[**] Politeknik Perkapalan Negeri Surabaya, Jalan Teknik Kimia Kampus ITS Sukolilo, Surabaya, Jawa Timur 60111, Indonesia

improve the accuracy of control, but sometimes damage the control system, so that only the linear parameters work. Nonlinear parameters will work until the system recovers. To begin with, 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 the linear and nonlinear parameters by the next regression input. With a minimum variance controller law, the controller signal is calculated by using the estimated linear and nonlinear parameters. The stability of the closed-loop control system is maintained by switching the linear and nonlinear parts; switching to the nonlinear part maintains accuracy, while switching to the linear part guarantees stability. Though a QARXNN model is built by using a neural network, the parameters obtained can be simplified in the form of regression coefficients. Control signals can be constructed directly by using regression coefficients.

By performing Taylor series expansions, a nonlinear system develops a linear correlation between the regression vector and its coefficients [6]. The coefficients serve as the parameters of the input vector called kernel functions, which can be executed by using a multi-input multi-output (MIMO) model. These also can be executed by neurofuzzy, wavelet, radial basis function, and multilayer perceptron neural network (MLPNN) [11]. The accuracy, stability, and the speed of convergence can be improved with Lyapunov training [12]. The QARXNN can also be used to identify the linear system with more accurate results than those achieved by using the technique of recursive least squares error identification [13]. The contributions of this paper are as follows: (i) the modeling of WECS with uncertainty parameters is derived by wind speed signal under an autoregressive moving average (ARMA) model with random process; (ii) a QARXNN is applied to model and predict WECS dynamics online with emphasis on the search parameters of the input vector; (iii) with the minimum variance controller law, a controller signal is derived by QARXNN prediction model using linear and nonlinear parameters with the switching.

2. Dynamic Modeling of WECS System

2.1. Wind speed modeling Electrical energy generated by wind power plants in any region heavily depends on the characteristics of the wind in that region. There are many ways to model the wind speed as a basis to evaluate the power systems. One way uses the ARMA model by generating a random signal denoted as $ARMA(p,q)$, in which p refers to the order of autoregressive signals and q to the order of the moving average. The ARMA model created for the Swift Current site in Saskatchewan, Canada, based on data for the period from 1996 to 2003 appears in the following [14]:

$$\begin{aligned} s(t) &= 1.1772s(t-1) + 0.1001s(t-2) - 0.3572s(t-3) \\ &\quad +0.0379s(t-4) + v(t) - 0.5030v(t-1) \\ &\quad -0.2924v(t-2) + 0.1317v(t-3) \\ &\quad v(t) \in (0, 052476^2). \end{aligned} \tag{1}$$

The simulated wind speed at hour t, designated as $V(t)$, can be calculated as follows:

$$V(t) = \mu(t) + \sigma(t)s(t) \tag{2}$$

where $\mu(t)$ is the mean observed wind speed per hour, and $\sigma(t)$ is the standard deviation of the observed wind speed per hour.

2.2. Dynamic modeling of WECS The power captured by a wind turbine is given by

$$P_m = 0.5\rho\pi Cp(\lambda,\beta)R^2V^3 \tag{3}$$

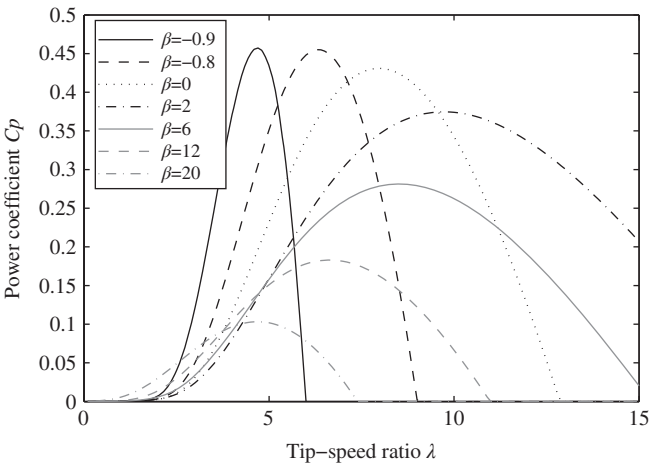

Fig. 1. Power coefficient versus tip-speed ratio for various blade pitches β

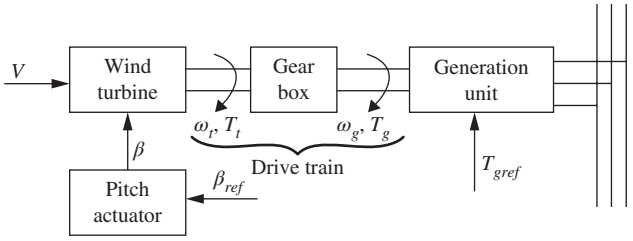

Fig. 2. Structural diagram of WECS systems

where ρ is the air density (typically 1.25 kg/m^3), R is radius of blades (m), $Cp(\lambda,\beta)$ is the wind turbine's power coefficient, and V is the wind speed (m/s). The coefficient $Cp(\lambda,\beta)$ depends on the pitch of the blades β (in degrees) and the tip-speed ratio λ. Tip-speed ratio is defined as the ratio of the linear velocity of the blade tip to the wind speed described as follows:

$$\lambda = \frac{\omega_t R}{V} \tag{4}$$

where ω_t is the wind turbine shaft speed (rad/s).

The relation between Cp and λ for a three-blade, horizontal-axis wind turbine for various blade pitches β is illustrated in Fig. 1. The curves have been obtained by plotting (5), which is commonly used in wind turbine simulators [15]:

$$Cp(\lambda,\beta) = 0.5176\left(\frac{116}{\lambda_i} - 0.4\beta - 5\right)e^{-21/\lambda_i} + 0.0068\lambda \tag{5}$$

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.008\beta} - \frac{0.035}{\beta^3 + 1}. \tag{6}$$

Figure 2 shows the subsystems interconnected to WECS, which consists of the wind turbine, the drive train, and the generation unit.

The objective of the proposed control is to maximize the power that the turbine extracts, which can be achieved if Cp is maximized. To maximize Cp, λ must be kept constant at its optimum value regardless of the wind speed. Figure 3 illustrates the steady-state power−speed characteristics (i.e. solid curves) and the maximum power point curve (i.e. dashed curve) attained for each wind speed at a pitch angle of 0^o. The aerodynamic torque on the wind turbine rotor can be obtained by using the following relations:

$$T_m = \frac{P_m}{\omega_t} = \frac{\rho\pi Cp(\lambda,\beta)R^3V^2}{2\lambda} \tag{7}$$

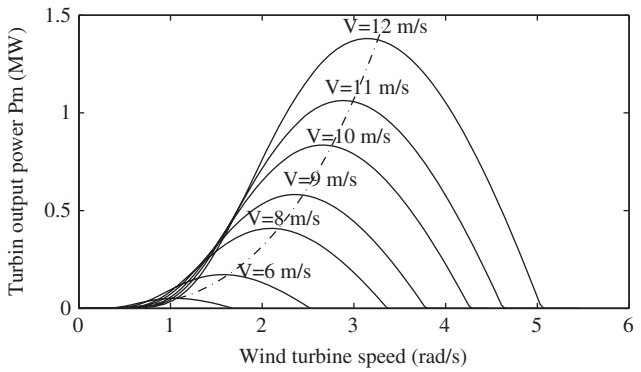

Fig. 3. Power–speed characteristics of wind turbines for various wind speeds at a pitch of 0^o.

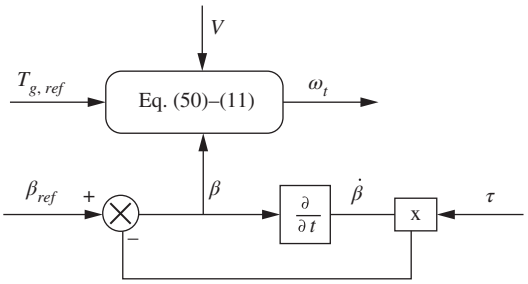

Fig. 4. Block diagram of nonlinear dynamic of the WECS

The proposed MPPT technique seeks to retrieve the optimal rotor speed ω_t (i.e. the speed corresponding to the maximum generated power) for any instantaneous value of the wind speed. Note that in Fig. 2 the external inputs of the dynamic WECS are the set points of generator torque $T_{g,ref}$, the desired pitch β_{ref}, and wind speed signal V. The outputs of WECS can be measured as presented by the turbine rotor speed ω_t. The desired pitch is the optimum pitch obtained from an aerodynamic turbine characteristic with the maximum power coefficient, which can be interpreted as the setting of the pitch position that extracts the maximum power from wind. The wind speed signals are fluctuating, and can be assumed as a disturbance signal affecting the uncertainty parameters of the WECS dynamics. The WECS dynamic can be described as follows:

$$\dot{\theta} = \omega_t - \omega_g \tag{8}$$

$$J_g \dot{\omega}_g(t) = K_s\theta + B_s\omega_t - B_s\omega_g + T_g(\omega_g, T_{g,ref}) \tag{9}$$

$$J_t \dot{\omega}_t(t) = -K_s\theta - B_s\omega_t + B_s\omega_g + T_m(\beta, V) \tag{10}$$

The generator torque T_g is a nonlinear function, with the generator speed ω_g and the reference electromagnetic torque $T_{g,ref}$ as variables. The generator usually operates in the linear region of its torque characteristics, which can be approximated into a linear form as

$$T_g = B_g\omega_g - T_{g,ref} \tag{11}$$

The pitch actuator is modeled as a first-order dynamic system with saturation in the amplitude and derivative of the pitch β as [8,15]

$$\dot{\beta} = \frac{-1}{\tau}\beta + \frac{1}{\tau}\beta_{ref} \tag{12}$$

Figure 4 shows the dynamic of the WECS model described in (3)–(12).

The control system acts to control the blade pitch position in order to maximize the power extracted from the wind, with the reference electromagnetic torque $T_{g,ref}$ set as constant. The system parameters are given as follows [4]:
Turbine and drive train parameters
$R = 30.30$ m, $K_s = 15.66 \times 10^5$ N/m, $Bs = 30.29 \times 10^2$ N.ms/rad, $J_t = 83.00 \times 10^4$ kg.m^2
Generator parameters
$B_g = 15.99$ N.ms/rad, $J_g = 5.9$ kg.m^2
Pitch actuator
$\tau = 100$ ms.

3. Control Strategy

To control the WECS, the controller is designed in two steps. The first step involves the identification and prediction of WECS by using the QARXNN model, while the second involves deriving and implementing the control law-based prediction model. Figure 5 shows an adaptive controller based on the QARXNN model. The turbine speed is operated at the MPPT point by controlling the pitch blade position, with the generator torque assumed to be constant.

3.1. System identification Through performing Taylor series expansions [6,12], the nonlinear continuous function can be presented as

$$y(t) = y_0 + \phi(t)^T \aleph(\phi(t)) \tag{13}$$

where $\aleph(\phi(t)) = [a_{(1,t)} \cdots a_{(n_y,t)}\, b_{(1,t)} \cdots b_{(n_u,t)}]^T$ are the Taylor coefficients; $\phi(t) \in R^{n=n_u+n_y}$ denotes the input vector with elements $\phi(t) = [-y(t-1) \cdots - y(t-n_y)\, u(t-1) \cdots u(t-n_u)]^T$; and n_u and n_y represent the orders of time delay in the input–output data. $\aleph(\phi(t)) \in R^{n=n_u+n_y}$ denotes a kernel function that is used to give the coefficients of the input vector. In our main theory, the following assumption are made:

Assumption 1. The pairs of input and output of the training data are bounded.

Assumption 2. The input and output of the nonlinear function $\aleph(\phi(t))$ are bounded.

By performing Taylor series expansion, we develop the nonlinear system presented as a linear correlation between the input vector and its coefficients. If the system modeling represents a plant that is a linear system, the coefficients obtained will be constant; otherwise, if the system modeling represents a plant that is a nonlinear system, the obtained coefficients will be a function of time [13]. A QARXNN model puts nonlinear function into the coefficients of input vector as follows:

$$\begin{aligned} y(t, \phi(t)) = {} & b_{(1,t)}u(t-1) + \cdots + b_{(n_u,t)}u(t-n_u) \\ & - a_{(1,t)}y(t-1) - \cdots - a_{(n_y,t)}y(t-n_y) \end{aligned} \tag{14}$$

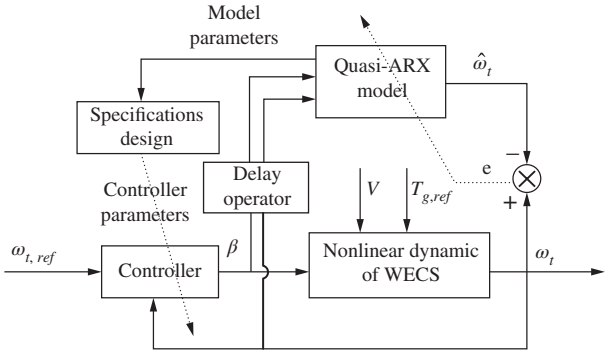

Fig. 5. MPPT controller of WECS based on the QARXNN prediction model

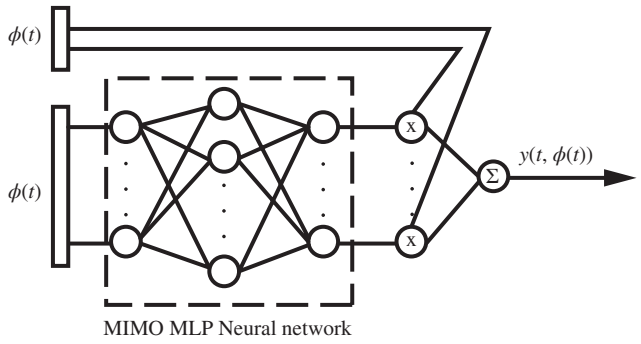

Fig. 6. Quasi-ARX neural network with MLP network as embedded systems

Figure 6 illustrates the scheme of system identification and prediction by QARXNN model.

The input vector is used as the input for an embedded system constructed by MLPNN with a three-layer neural network. The number of input layers, hidden layers, and output layers is the same and is equal to n. A QARXNN model incorporating neural network can be expressed as

$$y(t, \phi(t)) = \phi(t)^T \aleph(\phi(t)) \tag{15}$$

$$\aleph(\phi(t)) = W_2 \Gamma W_1(\phi(t) + B) + \theta \tag{16}$$

where $\Omega = \{W_1, W_2, B, \theta\}$. $W_1 \in R^{mxn}, W_2 \in R^{nxm}, B \in R^{mx1}$ are the weights matrix in the first layer, second layer, and bias vector of hidden nodes. The linear parameters of $\theta \in R^{nx1}$ is searched by applying the least square error (LSE) algorithm, which is set as a bias vector of the output nodes for an embedded system of MLPNN. The symbol Γ is the diagonal nonlinear operator with identical sigmoidal elements on the hidden nodes.

The learning algorithms of the QARXNN model executed in the two steps are: (i) the LSE algorithm is used to update θ and then is set as bias vector for MLPNN; (ii) perform the backpropagation (BP) error algorithm for an embedded MLPNN. By using the two algorithms, we introduce two submodels incorporating a linear submodel and a nonlinear submodel as follows: $z_l(k) = y(t, \phi(t)) - \phi(t)^T [W_2(k) \Gamma W_1(k)(\phi(t) + B(k))]$ and $z_n(k) = y(t, \phi(t)) - \phi(t)^T \theta(k)$, where k denotes a sequence learning number. The learning algorithms for a QARXNN is executed step by step as follows:

1. set $k = 0$ for initial conditions, $\theta(k) = 0$; and small initial values to $W_1(k)$, $W_2(k)$, and $B(k)$, then set $k = 1$, where k is the learning number.
2. calculate $z_l(k)$, then estimate $\theta(k)$ for by using the LSE algorithm.
3. calculate $z_n(k)$, then estimate $W_1(k)$, $W_2(k)$, and $B(k)$. It is realized by using the well-known BP algorithm.
4. use (16) to update $\aleph(\phi(t))$.
5. stop if prespecified conditions are met; otherwise go to Step 2, and repeat the estimation of $\theta(k)$, $W_1(k)$, $W_2(k)$, and $B(k)$, set $k = k + 1$.

If the model in (15) is sufficient to model the input–output training data of the system, **Assumption 1** and **Assumption 2** are fulfilled, and then the output at $(t + d)$ can be predicted. To obtain the predicted output, (15) is regressed at time $(t + d)$ described as [6,13]

$$\begin{aligned} y(t+d) &= \hat{b}_{(1,t+d)} u(t+d-1) + \cdots + \\ &\hat{b}_{(n_u,t+d)} u(t-n_u+d) - \hat{a}_{(1,t+d)} y(t-1+d) \\ &- \cdots - \hat{a}_{(n_y,t+d)} y(t-n_y+d) \end{aligned} \tag{17}$$

where $\phi(t+d) = [y(t+d-1)\, y(t+d-2) \cdots y(t+d-n_y)\, u(t+d-1)\, u(t+d-2) \cdots u(t+d-n_u)]^T$ is d step ahead of the input vector. $\hat{\aleph}(\phi(t+d))$ and $[\hat{a}_{(1,t+d)} \cdots \hat{a}_{(n_y,t+d)}\, \hat{b}_{(1,t+d)} \cdots \hat{b}_{(n_u,t+d)}]^T$ are the estimated parameters of the nonlinear part by MLPNN and its elements, respectively. For online step-ahead prediction, d is equal to 1. The estimated coefficient of the input vector is calculated by using an embedded MLPNN with the next input vector of $d = 1$ by $\phi(t+1) = [y(t)\, y(t-1) \cdots y(t+1-n_y)\, u(t)\, u(t-1) \cdots u(t+1-n_u)]^T$. With the next regression vector, the estimated coefficients are calculated by

$$\hat{\aleph}(\phi(t+1)) = W_2 \Gamma W_1(\phi(t+1) + B) + \hat{\theta} \tag{18}$$

By considering (18), we can see that the QARXNN model consists of the estimated linear part of the parameter $\hat{\theta}$ and the estimated nonlinear part of parameter $\hat{\aleph}(\phi(t+1))$.

3.2. Controller design A QARXNN model is improved to guarantee closed-loop stability of the control system expressed as

$$y(t, \phi(t)) = \phi(t)^T \hat{\aleph}(\phi(t), \chi(t)) \tag{19}$$

$$\hat{\aleph}(\phi(t), \chi(t)) = \chi(t) W_2 \Gamma W_1(\phi(t) + B) + \hat{\theta} \tag{20}$$

where $W_2 \Gamma W_1(\phi(t) + B) + \hat{\theta}$ is the estimated parameter of the nonlinear part, and $\hat{\theta}$ denotes the estimated parameter of θ in the linear part that is used as a bias vector in the output nodes. Obviously, by introducing the switching function $\chi(t)$, the improved QARXNN model is different from the conventional QARXNN model. When $\chi(t) = 1$, it is a nonlinear prediction model, which can ensure prediction accuracy. And when $\chi(t) = 0$, it is a linear prediction model, which can ensure control stability [5].

The linear part error and the nonlinear part error, respectively, are defined as follows:

$$e_1(t) = y(t) - \phi(t)^T \hat{\theta} \tag{21}$$

$$\begin{aligned} e_2(t) &= y(t) - \phi(t)^T \hat{\aleph}(\phi(t)) \\ &= y(t) - \phi(t)^T \hat{\theta} \\ &\quad - \phi(t)^T W_2 \Gamma W_1(\phi(t) + B) \end{aligned} \tag{22}$$

Then θ is updated as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{a(t)\phi(t-1)e_1(t)}{1 + \phi(t-1)^T \phi(t-1)} \tag{23}$$

$$a(t) = \begin{cases} 1, & \text{if } \|e_1(t)\| > 2\triangle \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

Similar to those in Refs [5,16,17], the switching criterion function is described as follows:

$$\begin{aligned} J_i(t) &= \sum_{l=1}^{t} \frac{a(l)(\|e_i(l)\|^2 - 4\triangle^2)}{2(1 + \phi(l-1)^T \phi(l-1))} \\ &\quad + c \sum_{l=t-N+1}^{t} (1 - a(l)) \|e_i(l)\|^2, i = 1, 2 \end{aligned} \tag{25}$$

$$\chi(t) = \begin{cases} 1, & \text{if } J_1(t) > J_2(t) \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

where i=1 denotes the linear adaptive minimum variance controller, and i=2 denotes the nonlinear adaptive minimum variance controller. The value of $\triangle$ is determined by the designer, where $\triangle \le \phi(t)\hat{\aleph}(\phi(t))$, N is a positive integer, and $c \ge 0$ is a predefined constant. Switching theory and stability analysis of the closed-loop system can be studied more clearly in Refs [5,16–19].

A minimum variance controller is proposed, which defined as follows:

$$M(t+1) = \left(\frac{1}{2}(y(t+d) - y^*(t+d))^2 + \frac{\lambda}{2}u(t)^2\right) \quad (27)$$

where λ is the weight of the control input, d is a differential operator which equals 1 for online step-ahead prediction. The controller can be obtained by solving

$$\frac{\partial M(t+1)}{\partial u} = 0 \quad (28)$$

A QARXNN is used to model WECS online. The controller signal calculated by solving (28) is difficult due to nonlinearity and the multiparametric model. Fortunately, a QARXNN model can be simplified by a linear correlation between the input vector and its coefficients. The controller is linear with respect to the input variable $u(t)$. Therefore, a controller is derived from simplified QARXNN model [5,6]. By adapting the input vector, the control law is modified as follows:

$$u(t) = \frac{\hat{b}_1(t)}{\hat{b}_1^2(t) + \lambda}(\hat{b}_1(t) - \hat{b}(q^{-1}, \phi(t))q)u(t-1) + y^*(t+1) - \hat{a}(q^{-1}, \phi(t))y(t)) \quad (29)$$

The controller is designed in two steps: (i) identifying the system using the QARXNN model, and (ii) designing a controller by using the parameter estimation that has been done in the first step.
Theorem: For a system expressed in (19) using an adaptive minimum variance controller, (29) and all the input–output signals in the closed-loop system are bounded. Moreover, the tracking error of the system can converge to zero when a proper neural network is determined.

Proof: First, the model error $e_i(t)$ is defined as

$$e_i(t) = y(t) - \phi(t)^T\hat{\aleph}(\phi(t), \chi(t)) = y(t) - \phi(t)^T\hat{\theta} - \chi(t)\phi(t)^T W_2\Gamma W_1(\phi(t) + B) \quad (30)$$

Then subtracting θ_0 from both sides of (23), we get

$$\tilde{\theta} = \tilde{\theta}(t-1) - \frac{a(t)\phi(t-1)(\phi(t-1)^T\tilde{\theta}(t-1) - \delta(t))}{1 + \phi(t-1)^T\phi(t-1)} \quad (31)$$

where $\tilde{\theta} = \hat{\theta}(t) - \theta_0$ and $\delta(t) = y(t) - \phi(t)^T\theta\hat{(t)}$.

Consider the function

$$V(t) = \left\|\tilde{\theta}(t)\right\|^2 \quad (32)$$

Then, noting that $a(t)$=0 or 1, and combined with (23) and (24), we can get

$$\begin{aligned} V(t) &= V(t-1) - \frac{2a(t)(e_i(t) - \delta(t))e_i(t)}{1 + \phi(t-1)^T\phi(t-1)} + \frac{a(t)^2\phi(t-1)^T\phi(t-1)e_i(t)^2}{(1 + \phi(t-1)^T\phi(t-1))^2} \\ &\le V(t-1) + \frac{a(t)(2e_i(t)\delta(t))}{1 + \phi(t-1)^T\phi(t-1)} - \frac{a(t)e_i(t)^2}{1 + \phi(t-1)^T\phi(t-1)} \end{aligned} \quad (33)$$

from $2ab \le Ca^2 + b^2/C, \forall C$, the following inequality holds:

$$\begin{aligned} V(t) &\le V(t-1) + \frac{a(t)(e_i(t)^2/2 + 2\delta(t)^2)}{1 + \phi(t-1)^T\phi(t-1)} - \frac{a(t)e_i(t)^2}{1 + \phi(t-1)^T\phi(t-1)} \\ &\le V(t-1) + \frac{2a(t)\triangle^2}{1 + \phi(t-1)^T\phi(t-1)} - \frac{1}{2}\frac{a(t)e_i(t)^2}{1 + \phi(t-1)^T\phi(t-1)} \end{aligned} \quad (34)$$

$V(t)$ is a nonincreasing sequence bounded by zero. Moreover

$$\lim_{N\to\infty}\sum_{t=1}^{N}\frac{a(t)(e_i(t)^2 - 4\triangle^2)}{2(1 + \phi(t-1)^T\phi(t-1))} < \infty, \quad (35)$$

and

$$\lim_{N\to\infty}\frac{a(t)(e_i(t)^2 - 4\triangle^2)}{2(1 + \phi(t-1)^T\phi(t-1))} \to 0 \quad (36)$$

The stability of the closed system via the switching technique for the adaptive minimum variance controller can be described as follows: $J_1(t)$ is always bounded by (24) and (35). $J_2(t)$ has two cases:

1. $J_2(t)$ is bounded; so the model error $e(t)$ is bounded and satisfies (36).
2. $J_2(t)$ is unbounded, since $J_1(t)$ is bounded. So there exists a constant t_0 such that $\chi(t) = 0, \forall t > t_0$. The model also has bounded error $e(t)$.

From above inequalities, the input and output of the closed-loop switching control system are bounded. The linear control system is always bounded. If a proper nonlinear model is chosen and the accurate parameters is adjusted, the nonlinear control error $e_2(t)$ can converge to zero. The model only with linear parameters has to work until the use of nonlinear parameters does not disturb the stability of closed-loop system. Therefore, the controller using linear parameters $\hat{\theta}$ will work all the time, but with the nonlinear parameters $\hat{\aleph}(\phi(t))$ it will work under the switching sequence.

4. Simulation and Results

The proposed MPPT control strategy is applied to arrange the pitch of blade β in order to track the angular velocities of a turbine operating at the MPPT point. Wind speed is generated by an ARMA model shown by Fig. 7; the mean observed wind speed of $\mu(t)$ = 12 m/s and the standard deviation (SD) of the observed wind speed of $\sigma(t)$ = 1.5. The simulation results are shown in detail in Figs 7–13. In order to obtain the maximum output of power from a wind turbine generator system, it is necessary to drive the wind turbine at an optimal rotor speed for a particular wind speed.

An embedded system is constructed by MLPNN with a three-layer neural network. The input vector $\phi(t)$ is selected by $\phi(t) = [y(t-1)\, y(t-2)\, y(t-3)\, y(t-4)\, u(t-1)\, u(t-2)\, u(t-3)]^T$ with $n = 7$ equal to the sum of $n_u = 3$ and $n_y = 4$. The number

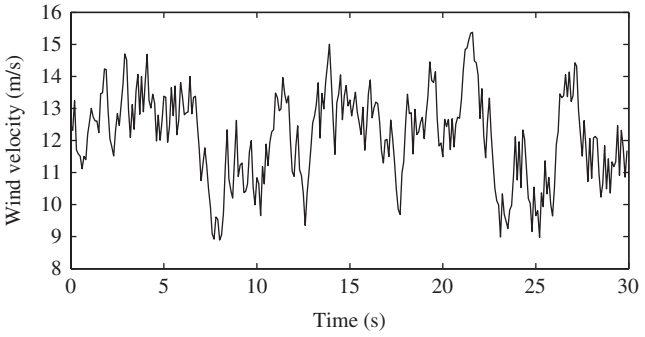

Fig. 7. Wind speed

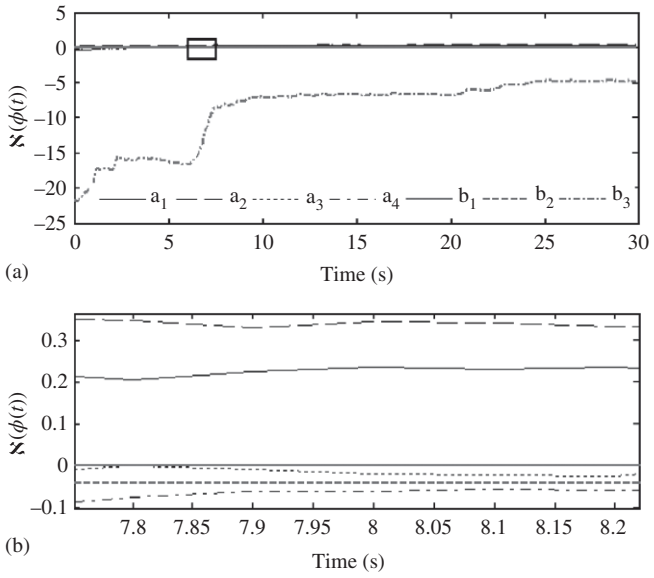

Fig. 8. Parameters of the input vector $\hat{\aleph}(\phi(t))$

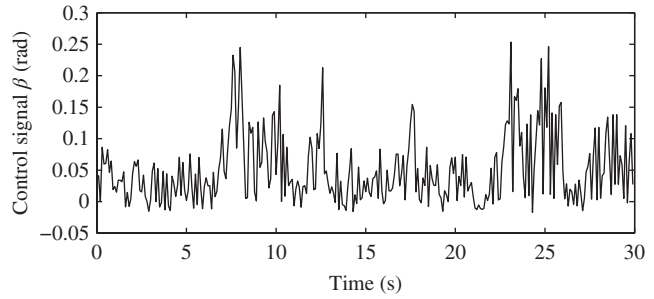

Fig. 9. Control signal

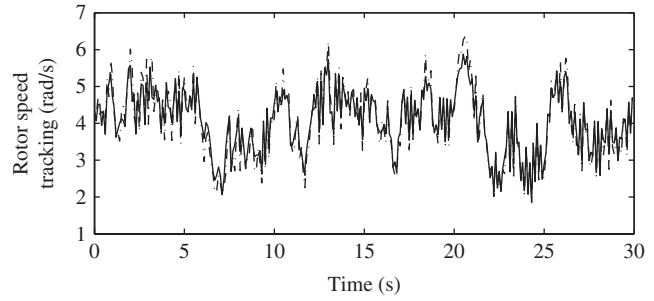

Fig. 10. Trajectory of ω_t of the minimum variance controller with switching-based quasi-ARX model

of input nodes, hidden nodes, and output nodes is also the same as n. The parameter of the switching criterion c=1.2 and N=3. By using TS fuzzy model with parameter uncertainties [4], the parameter (stiffness, damping, and moments of inertia) of the WECS is considered as the uncertainty parameter. It is presented by a nominal parameter (fixed) and an uncertainty parameter (noise). In this paper, the WECS system is modeled under the QAXNN model. The parameter consist of two parts: (i) nonlinear parameter $W_2\Gamma W_1(\phi(t)+B)$, which can be regarded as the uncertainty parameter executed by MLPNN, and (ii) the linear parameter θ. Linear relationship between the parameter and the input vector makes it easy to derive the control law. A switching mechanism is used to maintain system stability. By performing system identification, the parameter of the input vector $\aleph(\phi(t))$ can be estimated. As we can see, the WECS is a nonlinear system influenced by

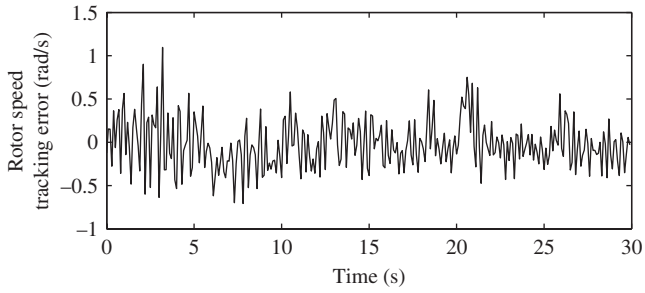

Fig. 11. Tracking error of turbine angular velocity

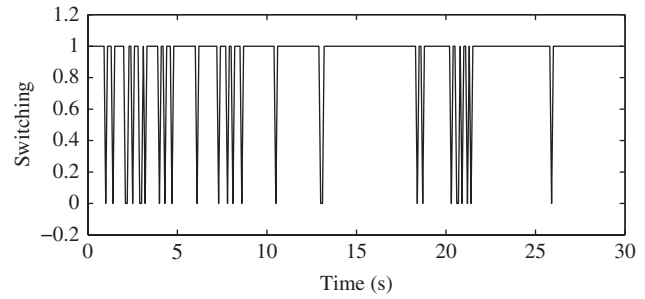

Fig. 12. Switching sequance

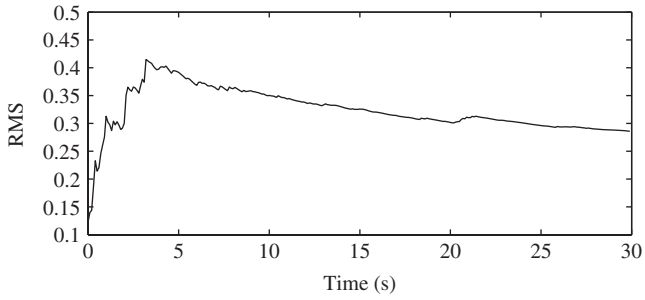

Fig. 13. RMS error versus time

wind speed fluctuation, thus rendering the parameters uncertain or a time function, as shown by Fig. 8. By using the technique of a minimum variance controller, the control signal is calculated by using $\aleph(\phi(t))$ set as controller parameter with the switching law. For better contrast, the snapshot of the part marked in Fig. 8(a) is shown enlarged in Fig. 8(b).

The control signal shown in Fig. 9 is fed to the WECS system to track the wind turbine rotor speed, and the result is shown in Fig. 10. The dot-dash line denotes the speed references ω_t at the MPPT operating point, and solid line denotes the output of the proposed method. The tracking error of the turbine rotor speed is shown in Fig. 11. The switching function between nonlinear and linear parts to ensure closed-loop stability and to improve the control accuracy is shown in Fig. 12. The performance of the proposed controller is also measured by the root-mean-squared (RMS) error versus time, as shown in Fig. 13.

$$RMS = \sqrt{\frac{\sum_{t=1}^{N}(y^*(t)-y(t))^2}{t}} \tag{37}$$

where $y^*(t)$ denotes the reference signal, and $y(t)$ denotes the output of the controlled plant. Figure 14 illustrates the WECS response at the MPPT operating point for the initial conditions $t=0s, V=12.48m/s$, power of the MPPT aerodynamic tracking $1.45MW$, $\beta=0^0$, and angular velocity $\omega_t=4.12rad/s$. When the wind speed decreases or increases, the rotor speed of turbine also

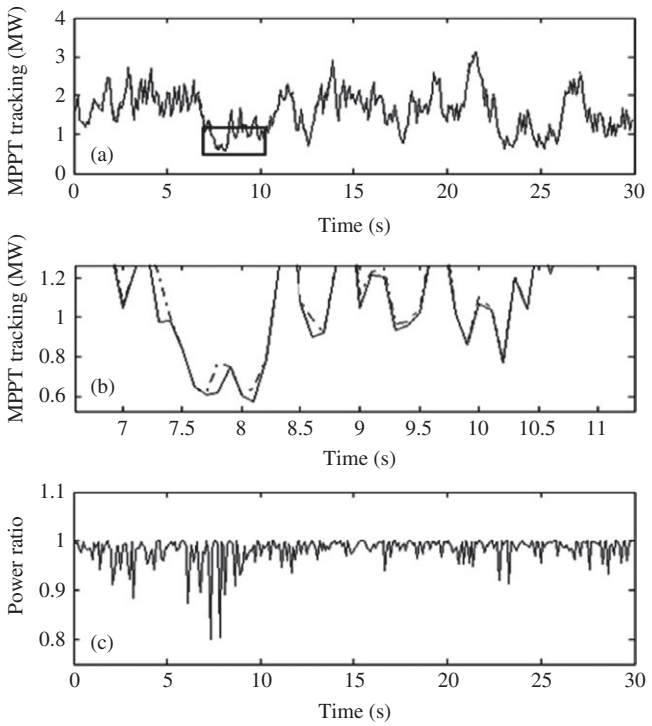

Fig. 14. Tracking MPPT power compared to aerodynamic power. (a) MPPT power tracking. (b) Snapshot of the part marked in (a) enlarged. (c) Ratio between the tracked MPPT power and the aerodynamic power

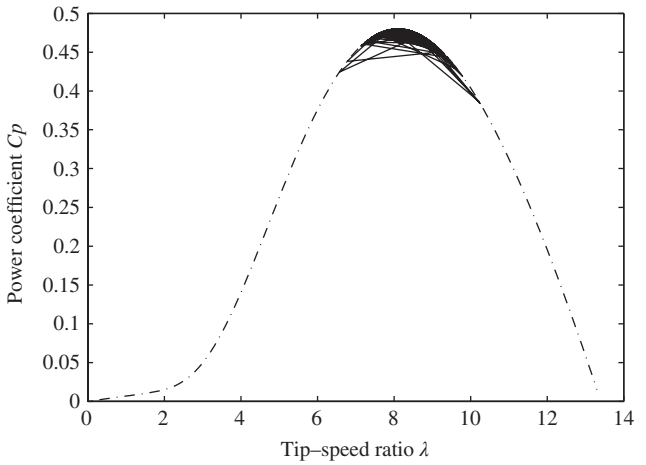

Fig. 15. Tracking *Cp* compared to the characteristic of the turbine

should change in order to keep maximum power of WECS by controlling the blade pitch ratio β.

If the rotor operates at speed references ω_t, then the maximum power coefficient can be achieved. Thus the turbine generator output is maximum. The result of MPPT power tracking is shown in Fig. 14. The tracked MPPT power using the proposed controller is compared to the aerodynamic power in Fig. 14(a). The dot-dashed line shows the aerodynamic power, and dashed line shows the tracked MPPT power. To make better contrast, the snapshot part marked in Fig. 14(a) is enlarged in Fig. 14(b). The accuracy of the tracked MPPT power is shown in Fig. 14(c) presented by the ratio of the tracked MPPT power and the aerodynamic power. A ratio of 1 indicates an accuracy of 100%. The performance of tracking control also can be measured, as shown in Figs 15 and 16. The turbine's peak performance is achieved with $Cp = 0.48$. The dot-dashed line shows the *Cp* characteristics of the turbine,

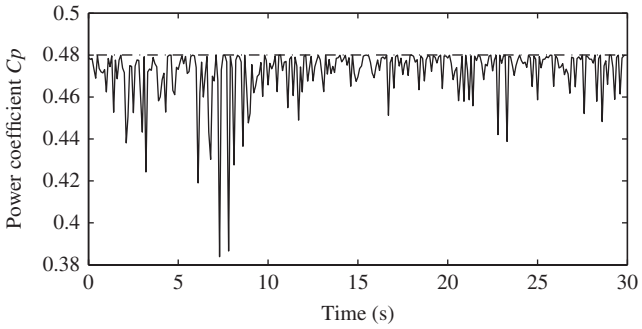

Fig. 16. Tracking *Cp* versus time

and the dashed line shows the result of the tracked *Cp*. These can also be given by *Cp* versus time in Fig. 16.

In Ref. [4], the resolution of the pitch actuator is 0.1 s. Therefore, a sampling time of 0.1 s is used for tracking the MPPT control simulation; so it seems to be high-speed control. However, to reduce the high-speed control, the sampling time can be changed and adapted to the real application and to the characteristic of the existing turbine components.

5. Conclusion

This paper presented an adaptive controller using the prediction model QARXNN. Based on the result of simulation, a minimum variance controller based on the QARXNN prediction model was shown to be effective in tracking the MPPT of the WECS. The proposed method is executed step by step as follows: (i) The wind speed dynamic model is adopted based on the ARMA model by generating a random signal. (ii) The principles of dynamic modeling of WECS are derived with given parameters, where the maximum energy that can be extracted from wind is influenced by the wind speed and pitch of the blades. (iii) The dynamics of WECS is simulated and identified online using the QARXNN model. The next input regression vector is the input for an embedded system of MLPNN to estimate the parameters that are used directly as controller parameters. The controller works under switching law to guarantee closed-loop stability. Finally, the control performance has been confirmed by a simulation and experimental results. The main contributions of this study are as follws: (i) the successful development of nonlinear dynamics of WECS modelling bases wind speed dynamic of ARMA model with generating a random signal; (ii) the successful application of the QARXNN prediction model to predict WECS online; and (iii) the successful application of switching controller based on QARXNN to track MPPT of the WECS.

Acknowledgments

This research was partly supported by the Indonesian Government Scholarship with Directorate General of Higher Education, Ministry of National Education, (Beasiswa Luar Negeri DIKTI Kementrian Pendidikan dan Kebudayaan Republik Indonesia) and the Shipbuilding Institute of Polytechnic Surabaya (Politeknik Perkapalan Negeri Surabaya PPNS).

References

(1) Rahman S, de Castro A. Environmental impacts of electricity generation: a global perspective. *IEEE Transcations on Energy Conversion* 1995; **10**(**2**):307–314.
(2) Bose B. Global warming: energy, environmental pollution and the impact of power electronics. *IEEE Industrial Electronics Magazine* 2010; **4**(**1**):6–17.

(3) Mesemanolis A, Mademlis C, Kioskeridis I. High-efficiency control for a wind energy conversion system with induction generator. *IEEE Transcations on Energy Conversion* 2012; **27**(**4**):958–967.
(4) Kamal E, Aitouche A, Ghorbani R, Bayart M. Robust fuzzy fault-tolerant control of wind energy conversion systems subject to sensor faults. *IEEE Transctions on Sustainable Energy* 2012; **3**(2):231–241.
(5) 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.
(6) Hu J, Kumamaru K, Hirasawa K. A quasi-ARMAX approach to modelling of non-linear systems. *Internationa Journal of Control* 2001; **74**(**18**):1754–1766.
(7) She Y, She X, Baran ME. Universal tracking control of wind conversion system for purpose of maximum power acquisition under hierarchical control structure. *IEEE Transcations on Energy Conversion* 2011; **26**(**3**):766–775.
(8) Muhando E, Senjyu T, Yona A, Kinjo H, Funabashi T. Disturbance rejection by dual pitch control and self-tuning regulator for wind turbine generator parametric uncertainty compensation. *IET Control Theory & Applications* 2007; **1**:1431–1440.
(9) Cheng Y, Wang L, Hu J. Identification of quasi-ARX neurofuzzy model with an SVR and GA approach. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences* 2012; **E.95-A**(**5**):876–883.
(10) 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.
(11) 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.
(12) Jami'in MA, Sutrisno I, Hu J. Lyapunov learning algorithm for quasi-ARX neural network to identification of nonlinear dynamical system. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Seoul, 2012; 3141–3146.
(13) 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 IEEE International Joint Conference on Neural Network, Dallas, TX, 2013; 2759–2762.
(14) Billinton R, Karki R, YiGao, Huang D, Hu P, Wangdee W. Adequacy assessment considerations in wind integrated power systems. *IEEE Transactions on Power Systems* 2012; **27**(**4**):2297–2305.
(15) Soliman M, Malik O, Westwick D. Multiple model multiple-input multiple-output predictive control for variable speed variable pitch wind energy conversion systems. *IET Renewable Power Generation* 2011; **5**(**2**):124–136.
(16) Chen L, Narendra KS. Nonlinear adaptive control using neural networks and multiple models. *Automatica* 2001; **37**:1245–1255.
(17) 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.
(18) Fu Y, Chai T. Nonlinear multivariable adaptive control using multiple models and neural networks. *Automatica* 2007; **43**:1101–1110.
(19) 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.

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 currently a Lecturer with the Politeknik Perkapalan Negeri Surabaya. He is also 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. Mr Jami'in is a student member of the IEEE.

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 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. Mr Sutrisno is a student member of the IEEE.

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 with 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, IEEJ, SICE, and IEICE.