07.pdf

The Twenty-First International Symposium on Artificial Life and Robotics 2016 (AROB 21st 2016),
The First International Symposium on BioComplexity 2016 (ISBC 1st 2016),
B-Con Plaza, Beppu, Japan, January 20-22, 2016

Application of Self Organizing Quasi-ARX RBFN for Rotor Speed Tracking Control of a Wind Turbine

Imam Sutrisno[1,2], Mohammad Abu Jamiin[1, 2], Jinglu Hu[1] and Norman Mariun[3]

[1]Graduate School of Information, Production and Systems, Waseda University, Japan
[2]Politeknik Perkapalan Negeri Surabaya, Indonesia
[3]Universiti Putra Malaysia
(Tel: 81-93-692-5271, Fax: 81-93-692-5271)

[1]jinglu@waseda.jp, imams3jpg@moegi.waseda.jp

Abstract: 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 self organizing quasi-linear ARX radial basis function network (SOQARX-RBFN) model. The stability of the closed loop controller is guaranteed by the switching of the linear and nonlinear parts parameters. From the simulation results, it is observed that the proposed controller is effective to track maximum power of WECS.

Keywords: self organizing quasi-ARX-RBFN model, rotor speed tracking control, wind turbine.

1 INTRODUCTION

There has been increased interest in wind energy since it is an environmentally friendly and renewable source of energy. With improved aerodynamic designs and sophisticated power electronic interfaces, wind turbines are capable of supplying a substantial amount of power. However, we can identify the dynamics of the wind energy conversion system (WECS) online by self organizing quasi-linear ARX radial basis function network (SOQARX-RBFN) model. [1][2][3]

In this paper, we propose a SOQARX-RBFNN prediction model to estimate parameters of the input vector, and by applying the minimum variance control law, the estimated parameter is set as the controller parameters with switching law. The linear part parameters are used in the whole time, while the nonlinear part parameters work under switching function. The use of nonlinear parameters can improve the accuracy of control, but sometimes damage the control system, so that only the linear parameters are working. Nonlinear parameters will work until the system recovers. To begin, a SOQARX-RBFN model is used to identify a dynamic system online [4][5]. The network parameters are updated continuously in accordance with the sampling time. The trained network weights of SOQARX-RBFN are used to estimate linear and nonlinear parameters by the next regression input. With a minimum variance controller law, the controller signal is calculated by using estimated linear and nonlinear parameters. The stability of the closed-loop control system is maintained by switching linear and nonlinear parts; switching to the nonlinear part maintains accuracy, while switching to the linear part guarantees stability. The SOQARX-RBFN 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 [1][6][7].

The contributions of this paper are: (1) a SOQARX-RBFN is applied to model and predict WECS dynamics online with emphasis on the search parameters of the input vector; (2) with minimum variance controller law, a controller signal is derived by SOQARX-RBFN prediction model using linear and nonlinear parameters with the switching law.

This paper is organized as follows. Section 2 introduces the SOQARX-RBFN prediction model. Section 3 introduces dynamic modeling of WECS. Section 4 describes numerical simulations to demonstrate the effectiveness of the proposed approach. Finally, Section 5 presents the conclusions.

2 DYNAMIC MODELING OF WECS

A model for the entire WECS can be structured into several interconnected subsystem models as shown in Fig. 1. This system consists of aerodynamics, a drive train, and an induction generator (IG). Finally, there is the actuator subsystem that models pitch servo behavior. The IG turbine is the most important part of this system and hence its reliability must be guaranteed. The possible faults in this system include those of the generator speed sensor and rotational speed of the turbine.

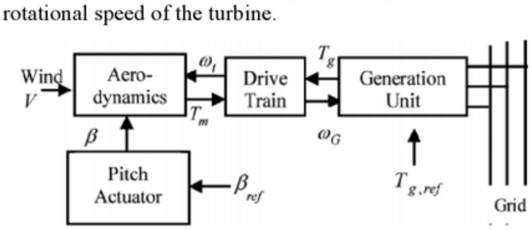

Fig. 1 Structural diagram of wind power system

The power extracted from the wind can be expressed as [2]

$$P_m = 0.5\rho\pi C_p(\gamma,\beta)R^2V^3 \quad (1)$$

where β is constant for fixed pitch wind turbines, is given byλ=$R\omega_t/V$, and $C_p(\lambda, \beta)$is defined by the following relation[3]:

$$C_p(\lambda,\beta) = (0.44 - 0.0167\beta)\sin\left[\frac{\pi(\lambda-3)}{15} - 0.3\beta\right] - 0.00184((\lambda-3)\beta. \quad (2)$$

A typical C_p-λ curve is shown in Fig. 2. It can be seen that the maximum value of C_p ($C_{p(max)}$ =0.45) is achieved forβ=0°and for λ=10.20. This particular value of $\lambda$is defined asthe optimal value of tip speed ratio (λ_{opt}). Normally, a variable speed wind turbine follows $\omega_{opt} = \lambda_{opt}$ V/R.

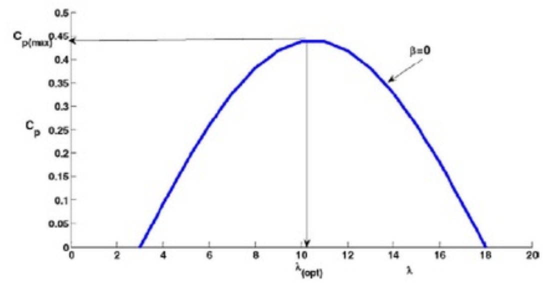

Fig. 2 Power coefficient versus Tip Speed Ratio

The drive train model consists of a low-speed shaft and a high-speed shaft having moments of inertia J_T and J_G and, respectively. Its model is given as

$$\begin{aligned} \dot{\theta}_s(t) &= \omega_t - \omega_G \\ J_T\dot{\omega}_t(t) &= -K_s\theta_s - B_s\omega_t + B_s\omega_G + T_m(\beta, V) \quad (3) \\ J_G\dot{\omega}_G(t) &= K_s\theta_s + B_s\omega_t - B_s\omega_G + T_g(\omega_G, T_{g,\mathrm{ref}}) \end{aligned}$$

where the aerodynamic torque $T_m$is given by [4]

$$T_m = \frac{P_m}{\omega_t} = \frac{\rho\pi Cp(\lambda,\beta)R^3V^2}{2\lambda}. \quad (4)$$

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

$$T_g = B_g\omega_G - T_{g,ref} \,(5)$$

The pitch actuator is modeled as a first-order dynamic system with saturation in the amplitude and derivative of the pitch β as [5][6].

$$\dot{\beta} = \frac{-1}{\tau}\beta + \frac{1}{\tau}\beta_{ref} \quad (6)$$

The control system acts to control 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 [2]:

Turbine and drive train parameters

R=30.30m, K_s=15.66x105N/m, B_s=30.29x102$N.ms/rad$, J_t=83.00x104$kg.m^2$

Generator parameters

B_g=15.99 $N.ms/rad$, J_g=5.9 $kg.m^2$

Pitch actuator

τ=100 ms.

3 SELF ORGANIZING QUASI-ARX-RBFN

To control WECS, the controller is designed in two steps. The first step involves the identification and prediction of WECS by using SOQARX-RBFN model, while the second involves deriving and implementing the control law based prediction model.

3.1 System Identification

Consider a single-input-single-output (SISO) nonlinear time invariant dynamical system with input-output relation as:

$$\begin{aligned} y(t) &= g(\varphi(t)) \\ \varphi(t) &= [y(t-1), \cdots y(t-n), \\ &\quad u(t-d), \cdots, u(t-d-m+1)]^T \end{aligned} \quad (7)$$

where $y(t)$ is the output at the time t (t = 1, 2, ...), $u(t)$ is the input, d is the known integer time delay, $\varphi(t)$ is the regression vector and n, m are the system orders. $g(\cdot)$ is a nonlinear function and at a small region around $\varphi(t) = 0$, it is $C\infty$ continuous. The origin is an equilibrium point, then $g(0) = 0$. As described in [8], a quasi-linear ARX model is constructed in two steps. In the first step, a macro-model is derived, which serves as a useful interface to introduce some properties favorable to specific applications, while the system complexity is embedded in the coefficient vector which is unknown nonlinear function of regression vector. In the second step, a flexible nonlinear nonparametric model such as RBFN is used to parameterize the coefficient vector.

Under the continuous condition, the unknown nonlinear function $g(\varphi(t))$ can be Taylor-expanded on small region around $\varphi(t) = 0$, i.e.,

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

where the prime denotes differentiation with respect to $\varphi(t)$. by introducing a coefficient vector Θ_φ, defined by

$$\Theta_\varphi = [a_{1,t} \cdots a_{n,t} \;\; b_{0,t} \cdots b_{m-1,t}]^T$$

to represent $(g'(0) + \frac{1}{2}\varphi^T(t)g''(0) + \cdots)$, we have a regression expression for the system:

$$y(t) = \varphi^T(t)\Theta_\varphi$$

where the coefficient vector Θ_φ is unknown nonlinear function of $\varphi(t)$.

In order to predict $y(t+d)$ by using the input-output data up to time t, the coefficients $a_{i,t}$ and $b_{j,t}$ should be calculable using the input-output data up to time t. To do so, let us iteratively replace $y(t+l)$ in the expressions of $a_{i,t}$ and $b_{j,t}$ with their predictions:

$$\hat{y}(t+l) = \hat{g}(\hat{\varphi}(t+l)), \quad l = 1, \cdots, d-1 \;\; (d > 1)$$

where $\hat{\varphi}(t+l)$ is $\varphi(t+l)$ whose elements $y(t+k)$, (k = 1, 2, ..., $l-1$) are replaced with their predictions $\hat{y}(t+k)$. Define the new expressions of the coefficients by:

$$\phi(t) = [y(t) \cdots y(t-n+1) \;\; u(t) \cdots u(t-m-d+2)]^T (8)$$

In a similar way to linear case [9], we can derive a predictor expression for the system

$$y(t+d) = \phi^T(t)\Theta_\phi \tag{9}$$

where Θ_ϕ is a coefficient vector defined by

$$\Theta_\phi = [\alpha_{0,t} \cdots \alpha_{n-1,t}\ \beta_{0,t} \cdots \beta_{m+d-2,t}]^T \tag{10}$$

where the coefficients $\alpha_{i,t}, \beta_{j,t}$ are unknown nonlinear functions of $\phi(t)$, and their relations with the coefficients $\tilde{a}_{i,t}$, $\tilde{b}_{j,t}$ are referred to Ref.[9] for the details.

Sometimes it is better to have a differential expression for controller design. For this purpose, let's consider the coefficients $\alpha_{i,t}(i = 0, \cdots, n-1)$ and $\beta_{j,t}(j = 0, \cdots, m+d-2)$ as a summation of two parts: the constant part α_i^l and β_j^l and the nonlinear function part on $\phi(t)$ which are denoted by $\alpha_{i,t} - \alpha_i^l$ and $\beta_{j,t} - \beta_j^l$, respectively. By introducing $\Theta_\phi = \theta + \Theta_\phi^n$, the expression of system in the predictor form Eq.(9) can be described by:

$$y(t+d) = \phi^T(t)\theta + \phi^T(t)\Theta_\phi^n \tag{11}$$

where $\theta = [\alpha_0^l \cdots \alpha_{n-1}^l \beta_0^l \cdots \beta_{m+d-2}^l]^T$ is a constant vector, and $\Theta_\phi^n = [(\alpha_{0,t} - \alpha_0^l) \cdots (\alpha_{n-1,t} - \alpha_{n-1}^l)(\beta_{0,t} - \beta_0^l) \cdots (\beta_{m+d-2,t} - \beta_{m+d-2}^l)]^T$ is a coefficient vector that is unknown nonlinear function of $\phi(t)$.

Applying a d-difference operator, defined by $\Delta = 1 - q^d$, to Eq.(11) and introducing $\psi(t) = \Delta\phi(t)$ and $\Psi^T(t)\tilde{\Theta}_\Psi^n = \Delta\phi^T(t)\Theta_\phi^n$, we obtain a d-difference expression of the system:

$$\Delta y(t+d) = \psi^T(t)\theta + \Psi^T(t)\tilde{\Theta}_\Psi^n \tag{12}$$

where $\Psi(t) = [y(t) \cdots y(t-d-n+1) u(t) \cdots u(t-m-2d+2)]^T$ and $\tilde{\Theta}_\Psi^n$ is unknown nonlinear function of $\Psi(t)$.

If we consider a controller design, it is better to have the prediction model linear of $u(t)$. However, the macro model described by Eq.(12) is a general one which is nonlinear in the variable $u(t)$, because the coefficient vector $\tilde{\Theta}_\Psi^n$ is nonlinear function of $\Psi(t)$ whose elements contain $u(t)$. To solve this problem, an extra variable $x(t)$ is introduced and an unknown nonlinear function $\rho(\xi(t))$ is used to replace the variable $u(t)$ in $\tilde{\Theta}_\Psi^n$. Obviously, in a control system, the reference signal $y^*(t+d)$ can be used as the extra variable $x(t+d)$. Under Assumption 1(v), the function $\rho(\xi(t))$ exists. Define

$$\begin{aligned}\xi(t) = [&y(t) \cdots y(t-n_1)\ x(t+d) \cdots \\ &x(t-n_3+d)\ u(t-1) \cdots u(t-n_2)]^T\end{aligned} \tag{13}$$

including the extra variable $x(t+d)$ as an element. A typical choice for n_1, n_2 and n_3 in $\xi(t)$ is $n_1 = n+d-1, n_2 = m+2d-2$ and $n_3 = 0$. Then we have a d-defference predictor expression linear in $u(t)$, defined by

$$\Delta y(t+d) = \psi^T(t)\theta + \Psi^T(t)\Theta_\xi^n \tag{14}$$

where the coefficient vector $\Theta_\xi^n = \tilde{\Theta}_\Psi^n$ is a unknown nonlinear function of $\xi(t)$, in which the element $u(t)$ in $\Psi(t)$ is replaced by $\rho(\xi(t))$.

The elements of Θ_ξ^n are unknown nonlinear function of $\xi(t)$, which can be parameterized by RBFN or other neural network models. By using a multi-input and multi-output RBFN model, we have

$$\Theta_\xi^n = \sum_{j=1}^{M} \mathbf{w}_j R_j(\xi(t), \Omega_j) \tag{15}$$

where M is the number of RBF nodes, $\mathbf{w}_j = [\omega_{1j}\ \omega_{2j} \cdots \omega_{Nj}]^T$ the weight vector ($N = \dim(\Psi(t))$), and $R_j(\xi(t), \Omega_j)$ the RBF node functions defined by:

$$R_j(\xi(t), \Omega_j) = e^{-\lambda_j \|\xi(t) - Z_j\|^2} \qquad j = 1, 2, \cdots, M \tag{16}$$

where $\Omega_j = \{Z_j, \lambda_j\}$ is the parameter sets of the RBF node functions; Z_j is the center vector of RBF and λ_j are the scaling parameters; $\|\cdot\|^2$ denotes the vector two-norm. Then we can express the quasi-linear ARX RBFN prediction model for Eq.(14) in a form of:

$$\Delta y(t+d) = \psi^T(t)\theta + \sum_{j=1}^{M} \Psi^T(t)\mathbf{w}_j R_j(\xi(t), \Omega_j) \tag{17}$$

Now, introducing the following notations:

$$\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2\ \cdots\ \mathbf{w}_M] \tag{18}$$

$$\aleph(\xi(t)) = [e^{-\lambda_1\|\xi(t)-Z_j\|^2}\ \ldots\ e^{-\lambda_M\|\xi(t)-Z_j\|^2}]^T \tag{19}$$

the quasi-linear ARX RBFN model is further expressed by

$$\begin{aligned}\Delta y(t+d) &= \psi^T(t)\theta + \Psi^T(t)\mathbf{W}\aleph(\xi(t)) \\ &= \psi^T(t)\theta + \Xi^T(t)\Theta\end{aligned} \tag{20}$$

where $\Theta = [w_{11} \cdots w_{n1} \cdots w_{1M} \cdots w_{nM}]^T$, $\Xi(t) = \aleph(\xi(t)) \otimes \Psi(t)$, while the symbol $\otimes$ denotes Kronecker production.

The quasi-linear ARX RBFN prediction model described by Eq.(17) is an accurate model of the system in d-difference form described by Eq.(14). It is linear in the input variable $u(t)$, which is useful for controller design.

The linear parameter vector θ of the linear part of the model is updated as [15]:

$$\hat{\theta}(k) = \hat{\theta}(k-d) + \frac{a(k)\psi(k-d)e_1(k)}{1 + \psi(k-d)^T\psi(k-d)} \tag{21}$$

where $\hat{\theta}(k)$ is the estimate of θ at step k, which also denotes the parameter of a linear model used to approximate the system in d-difference form, and

$$a(k) = \begin{cases} 1 & \texttt{if } |e_1(k)| > 2D \\ 0 & \texttt{otherwise} \end{cases} \tag{22}$$

where $e_1(k)$ denotes the error of the linear model, defined by

$$e_1(k) = \Delta y(k) - \psi(k-d)^T\hat{\theta}(k-d) \tag{23}$$

The linear parameter Θ of nonlinear part of the quasi-linear ARX RBFN model is updated by a least square (LS) algorithm:

$$\hat{\Theta}(k) = \hat{\Theta}(k-d) + \frac{P(k)\Xi(k-d)e_2(k)}{1 + \Xi(k-d)^T P(k)\Xi(k-d)} \tag{24}$$

where $\hat{\Theta}(k)$ is the estimate of Θ at step k. $\hat{\Theta}(0) = \Theta_\circ$ is assigned with an appropriate initial value. $e_2(k)$ is the error of quasi-linear ARX RBFN model, defined by:

$$e_2(k) = \Delta y(k) - \psi(k-d)^T \hat{\theta}(k-d) - \Xi^T(k-d)\hat{\Theta}(k-d) \quad (25)$$

$$P(k) = \frac{P(k-d) - P^T(k-d)\Xi(k-d)^T\Xi(k-d)P(k-d)}{1+\Xi(k-d)^T P(k)\Xi(k-d)} \quad (26)$$

No restriction is made on how the parameters $\hat{\Theta}(k)$ are updated except they always lie inside some pre-defined compact region $\hbar$:

$$\hat{\Theta}(k) \in \hbar \quad \forall k \quad (27)$$

The proposed SOQARX-RBFN model for identification approach is summarized in six steps [7]:

Step 1) Set a proper initial value of the number of RBF node, M, and predetermine the parameter sets of RBF nodes.

Step 2) Estimate the linear parameter vectors of Θ and Θ by using the iterative algorithms. Check the performance of the quasi-linear ARX RBFN model. If it needs further structure optimization, go to Step 3, otherwise stop.

Step 3) Compute the active firing rate Af_k for each RBF node. The RBF nodes with active firing rate larger than an activity threshold will split into multiple RBF nodes. Check the value of Af_k, if there is $Af_k > Af_0$, go to Step 4), otherwise go to Step 5).

Step 4) Carry out RBF node split for the RBF node k when $Af_k > Af_0$ (k=1,2,...,M): the old k-th RBF node will be deleted and new RBF nodes will be inserted. After completing all RBF node splits, update the number of RBF node, M, and the parameter sets of RBF nodes, and go to Step 5.

Step 5) Compute the mutual information between the k-*th* RBF node and the corresponding output nodes, $m(Q_k)$. The RBF nodes with mutual information smaller than a threshold m_0, ($0 < m_0 < 0.05$) will be deleted. Check $m(Q_k)$, for all k, if there is $m(Q_k)<m_0$, go to Step 6). Otherwise, go back to Step 2).

Step 6) Carry out RBF node deletion for the RBF node k when $m(Q_k)<m_0$ (k=1,2,...,M). Delete the k-*th* RBF node as well as all connections between the node Q_k and the corresponding output nodes, and update the remaining RBFN parameters in the following way. Find the RBF node k' which has the minimal Euclidean distance to the k-*th* RBF node. After completing all RBF node deletions, update the number of RBF node, M, and the parameter sets of RBF nodes, and go to Step 2.

3.2 Control Strategy .

A SOQARX-RBFN model is improved to guarantee closed loop stability of control system expressed as

$$\hat{y}(t+d) = (1-\mu_t)\hat{y}_1(t+d) + \mu_t \hat{y}_2(t+d) \quad (28)$$

where

$$\hat{y}_1(t+d) = \psi^T(t)\hat{\theta}(t) + y(t)$$
$$\hat{y}_2(t+d) = \psi^T(t)\hat{\theta}(t) + \sum_{j=1}^{M} \Psi^T(t)\hat{\mathbf{w}}_{\hat{y}}(t) R_j(\xi(t), \hat{\Omega}_j) + y(t) \quad (29)$$

Consider a minimum variance control with the criterion function as follows:

$$\mathbf{M}(t+d) = \frac{1}{2}(y(t+d) - y^*(t+d))^2 \quad (30)$$

where $y^*(t)$ is a known bounded reference output. The optimal control law minimizing (18) is:

$$y(t+d) - y^*(t+d) = 0. \quad (31)$$

Then corresponding to the predictors (28)-(29), we can obtain the following controllers:

$$\mathcal{C}: \psi^T(t)\hat{\theta}(t) + \mu_t \sum_{j=1}^{M} \Psi^T(t)\hat{\mathbf{w}}_j(t) R_j(\xi(t), \hat{\Omega}_j) = y^*(t+d) - y(t) \quad (32)$$

and two others C_1 and C_2 corresponding to the extreme cases of $\mu_t = 0$ and $\mu_t = 1$, respectively

$$\mathcal{C}_1: \psi^T(t)\hat{\theta}(t) = y^*(t+d) - y(t)$$
$$\mathcal{C}_2: \psi^T(t)\hat{\theta}(t) + \sum_{j=1}^{M} \Psi^T(t)\hat{\mathbf{w}}_j(t) R_j(\xi(t), \hat{\Omega}_j) = y^*(t+d) - y(t) \quad (33)$$

4 NUMERICAL SIMULATIONS

Simulations were performed in MATLAB using the nonlinear model provided in Section 2. In this section, simulation compares the results of the proposed algorithm with the previous algorithm [1] subject to parameter uncertainty and disturbance. The simulation data represent a wind turbine with three blades, a horizontal axis, and variable speed. The proposed controller for the WECS is tested for random variation of wind speed as shown in Fig. 3 to demonstrate the effectiveness of the proposed algorithm. To compare the results of the proposed algorithm with the previous algorithm [1], system response is studied for case of small disturbance.

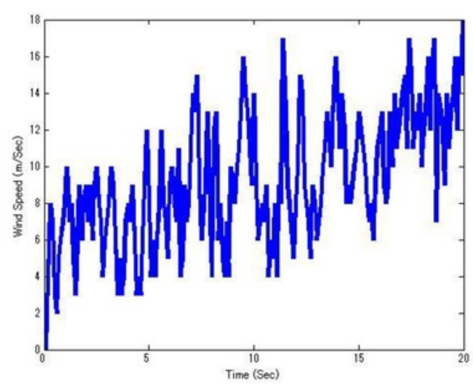

Fig. 3 Wind speed

The rotor speed for the capture of maximum power from the wind turbine is shown in Fig. 4 (solid line). It is clear that the dashed and dotted curves in Fig. 4 which represent the actual rotor speed for proposed algorithm and the previous algorithm, respectively, coincide with the solid curve. Rotor speed tracking errors for the two algorithms are shown in Fig. 5.

To compare the results of the proposed algorithm with those of the previous algorithm [1], we studied the response of the system subject to small disturbance (10% of wind speed). It can be seen that the proposed SOQARX-RBFN controller can fully control the plant throughout the entire range of wind velocities tested, in comparison to the previous algorithm which failed to provide adequate responses.

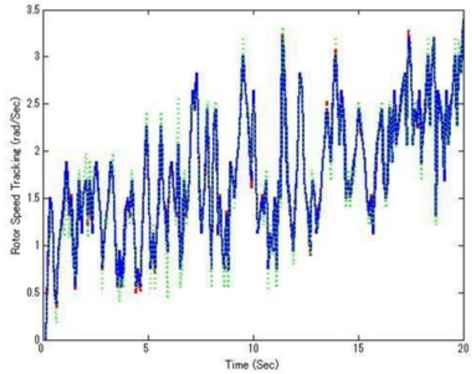

Fig. 4 Rotor speed tracking of the proposed algorithm (dashed line) and the previous algorithm [1] (dotted line)

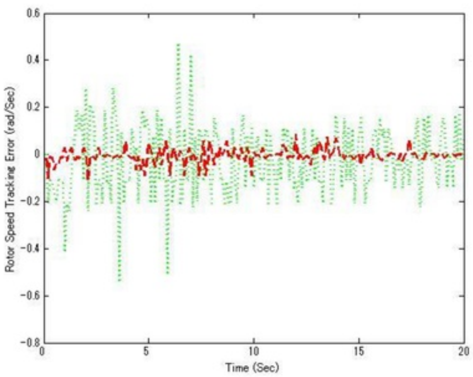

Fig. 5 Rotational speed tracking error of the proposed algorithm (dashed line) and the previous algorithm [1] (dotted line)

Moreover, the rotor speed tracking error is smaller with the proposed algorithm, compared to the previous, as shown in Figs. 5.

5 CONCLUSIONS

This paper presents an adaptive controller using prediction model of SOQARX-RBFN. Based on the result of simulation, a minimum variance controller based on QARXNN prediction model is effective to track MPPT of the WECS. The proposed method is executed step by step as follows: (1) wind speed dynamic model is adopted bases autoregressive moving average model by generating a random signal; (2) the principles of dynamic modeling of WECS is derived with given parameters where the maximum energy that can be extracted of wind power is influenced by wind speed and pitch of the blades; (3) The dynamic of WECS is simulated and identified online using quasi-ARX neural network model. The next input regression vector is the input for an embedded system of MLPNN to estimate parameters that is 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: 1) the successful development of nonlinear dynamics of WECS modeling bases wind speed dynamic of ARMA model with generating a random signal; 2) the successful application of the SOQARX-RBFN prediction model to predict WECS online; 3) the successful application of switching controller based SOQARX-RBFN to track MPPT of the WECS.

REFERENCES

[1] Mohammad Abu Jami'in, Imam Sutrisno and Jinglu Hu, Maximum Power Tracking Control for a Wind Energy Conversion System Based on a Quasi-ARX Neural Network Model, IEEJ Transactions On Electrical And Electronic Engineering, Vol.10, No.4, July, 2015

[2] E. Kamal, M. Koutb, A. A. Sobaih, and S. Kaddah, Maximum power control of hybrid wind-diesel-storage system, Adv. Fuzzy Syst., vol. 8, pp. 1–10, Jan. 2008.

[3] S. Karimi, A. Gaillard, P. Poure, and S. Saadate, Current sensor fault tolerant control for WECS with DFIG, IEEE Trans. Ind. Electron., vol. 56, no. 11, pp. 4660–4670, Nov. 2009.

[4] V. Galdi, A. Piccolo, and P. Siano, Exploiting maximum energy from variable speed wind power generation system by using an adaptive Takagi-Sugeno-Kang fuzzy model, Energy Convers. Manage., vol.50, no. 2, pp. 413–421, Feb. 2009.

[5] E. Muhando, T. Senjyu, A. Yona, H. Kinjo, and T. Funabashi, Disturbance rejection by dual pitch control and self-tuning regulator for wind turbine generator parametric uncertainty compensation, IET Control Theory Appl.,vol. 1, pp. 1431–1440, 2007.

[6] M. Soliman, O. Malik, and D. Westwick, Multiple model multiple-input multiple-output predictive control for variable speed variable pitch wind energy conversion systems, IET Renew. Power Gener., vol. 5(2), pp. 124–136,2011.

[7] I. Sutrisno, M.A. Jami'in, J. Hu and M.H. Marhaban: Self-Organizing Quasi-linear ARX RBFN Modeling for Identification and Control of Nonlinear Systems, in Proceedings of The SICE Annual Conference 2015 (SICE2015) (Hangzhou China), pp. 788-793, 2015.

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

[9] J.Hu, K.Kumamaru and K.Hirasawa: A Quasi-ARMAX Approach to the Modeling of Nonlinear Systems, International Journal of Control, Vol.74, No.18, pp.1754-1766, 2001.

07.pdf

ORIGINALITY REPORT

84%

SIMILARITY INDEX

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

★Jami'in, Mohammad Abu, Imam Sutrisno, and Jinglu Hu. "Maximum power tracking control for a wind energy conversion system based on a quasi-ARX neural network model : Maximum Power Tracking Control for a Wind Energy Conversion", IEEJ Transactions on Electrical and Electronic Engineering, 2015.
Crossref

30%

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	OFF		