1-1-2009

Echo-State-Network-Based Real-Time Wind Speed Estimation for Wind Power Generation

Wei Qiao
University of Nebraska–Lincoln, wqiao@engr.unl.edu

Follow this and additional works at: http://digitalcommons.unl.edu/electricalengineeringfacpub

Part of the Electrical and Computer Engineering Commons

http://digitalcommons.unl.edu/electricalengineeringfacpub/136

This Article is brought to you for free and open access by the Electrical Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Faculty Publications from the Department of Electrical Engineering by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.
Echo-State-Network-Based Real-Time Wind Speed Estimation for Wind Power Generation

Wei Qiao

Abstract—Wind turbine generators (WTGs) are usually equipped with one or more well-calibrated anemometers to measure wind speed for system monitoring, control, and protection. The use of these mechanical sensors increases the cost and hardware complexity and reduces the reliability of the WTG system. This paper proposes an echo-state-network (ESN)-based real-time wind speed estimation algorithm for WTG systems. The ESN is designed to provide a nonlinear inverse model of the WTG dynamics, which is used to estimate the wind speed in real time from the measured WTG output electrical power and shaft speed at any turbine blade pitch angle. The estimated wind speed is then used for wind-speed-sensorless control of the WTG system. The proposed algorithm is verified by simulation studies on a 3.6-MW wind turbine equipped with a doubly fed induction generator (DFIG).

I. INTRODUCTION

Wind power is capable of becoming a major contributor to the nation’s and world’s electricity supply over the next three decades. It was estimated [1] by the Global Wind Energy Council (GWEC) that more than 20% of the world’s electricity would be supplied by wind power by 2030. The European Wind Energy Association (EWEA) has set a target to satisfy more than 22% of European electricity demand with wind power by 2030 [2]. In the United States, according to a report [3] by the Department of Energy, it is feasible to supply 20% of the nation’s electricity from wind by 2030.

Electrical power generation from wind is carried out by a wind turbine generator (WTG) system. Control and protection of WTGs require wind speed information, which is usually measured by well-calibrated mechanical sensors such as anemometers. In most WTG systems, one or more anemometers are placed surrounding the wind turbine at the same height as the hub height; the measured wind speed is approximately the average wind speed across the area swept by the rotor blades. In order to maintain the sensors in good operating condition, regular inspection and maintenance are required. Obviously, the use of these sensors increases both the equipment and maintenance costs of the WTG system. Moreover, the sensors are inevitably subject to failure even with careful maintenance. Sensor failure may further cause the failure of the WTG control and electrical systems [4]. According to statistical data reported in [4], sensor failures constitute more than 14% of failures in WTG systems, while more than 40% of failures are linked to the failure of sensors, the control system, or the electrical system. Repairing the failed components requires additional cost and causes significant downtime resulting in a significant loss in electrical energy production. Therefore, the use of sensors reduces the reliability and increases the operational cost of WTG systems. In addition, anemometers are very sensitive to icing. In many locations that have excellent wind resources but long winters, special models of anemometers with electrically heated shaft and cups are required. Consequently, additional sensors and an electrical grid connection are required to measure the temperature and to run the heater, respectively.

The problems incurred in using wind-speed sensors can be solved through real-time wind speed estimation where the wind speed is estimated from other sensor measurements instead of measured directly from sensors. Consequently, the WTG system is controlled and protected without the need of any wind-speed sensors.

In the existing wind speed estimation methods, the wind turbine power characteristics are explicitly represented by a nonlinear function [5] or approximated by a polynomial [6] or an artificial neural network (ANN) [7]-[8]. Based on the measured wind turbine shaft speed and mechanical power, the wind speed can be obtained by calculating the roots of the nonlinear function [5] or polynomial [6] or can be estimated from the ANN [7]-[8]. However, in a WTG system, usually the measured power is the generator electrical power instead of the turbine mechanical power. References [5]-[7] use the measured electrical power directly as the turbine mechanical power for wind speed estimation. However, because of the nonnegligible power losses and complex shaft system dynamics of many WTG systems, particularly the large WTGs with a gearbox, turbine mechanical power is not equivalent to generator electrical power. This problem is solved in [8] in which turbine mechanical power is estimated from generator electrical power by taking into account the losses and shaft dynamics of the system. However, this method requires the knowledge of the characteristics of the WTG system, including wind turbine power characteristics, shaft dynamics, losses, etc. These characteristics may not be accurately obtained due to high complexity, nonlinearity, and uncertainty of the WTG system.

The work in [7] and [8] is based on two types of static ANNs, i.e., a multilayer perceptron neural network.
(MLPNN) [9]-[10] and a radial basis function neural network (RBFNN) [11]-[12], respectively. These two types of ANNs may not be able to provide a good approximation for nonlinear, complex, dynamical systems. Mathematically, recurrent neural networks (RNNs) provide a universal approximator for dynamical systems. Compared to MLPNNs and RBFNNs, the mathematical representation and training of RNNs are much more complicated. The commonly used methods for supervised training of RNNs include back-propagation through time [13], real-time recurrent learning [14]-[15], extended Kalman filtering based algorithms [16]-[17], etc. However, training of RNNs is a complicated procedure and computationally expensive. There is no clear winner among these methods. Mathematical complexity and difficulties in training have prevented practical applications of RNNs.

Recently developed echo state networks (ESNs) [18]-[19] provide an architecture and a novel supervised training approach for RNNs. The ESNs overcome the aforementioned drawbacks of general RNNs, and therefore, are potentially an excellent candidate for approximation of nonlinear dynamical systems. However, still few practical applications of ESNs have been reported.

This paper proposes a novel ESN-based real-time wind speed estimation algorithm for WTG systems. An ESN is developed to approximate the nonlinear dynamical electrical power–wind speed characteristics of the WTG. Based on this dynamical ESN model, the wind speed is estimated directly from the measured WTG output electrical power. The estimated wind speed is then used for real-time control of the WTG system. The effectiveness of the proposed wind speed estimation algorithm is demonstrated by simulation studies in PSCAD/EMTDC [20] for maximum power generation control of a 3.6-MW wind turbine [8], [21] equipped with a doubly fed induction generator (DFIG).

II. WIND TURBINE GENERATOR SYSTEMS

Currently, the majority of the existing WTGs for electrical power production fall into the following three categories: 1) a fixed wind turbine equipped with a squirrel-cage induction generator (SCIG) as shown in Fig. 1, 2) a variable-speed wind turbine equipped with a DFIG as shown in Fig. 2, and 3) a variable-speed wind turbine equipped with a direct-drive synchronous generator (DDSG) as shown in Fig. 3.

In Fig. 1, the wind turbine is coupled to a SCIG through a gearbox. The wind turbine is typically fixed pitch or stall controlled [22]. Since there is no active control action on the generator, the SCIG can only operate within a narrow range of the rotational speed slightly above the synchronous speed from no-load to full load conditions. The almost fixed-speed SCIG has no control possibilities in the electric machine for active and reactive power. Large fluctuations in turbine and generator output power (caused by wind speed variations) stress the gearbox, which suffers relatively high failure rates. The SCIG consumes a large amount of reactive power, and therefore, is usually equipped with some form of switched capacitor banks for power factor correction.

In Fig. 2, the wind turbine is connected to a DFIG through a gearbox. The DFIG is a wound-rotor induction machine. It is connected to a power grid at both stator and rotor terminals. The stator is directly connected to the grid while the rotor is fed through a variable frequency de-link-voltage converter, which consists of a rotor-side converter (RSC) and a grid-side converter (GSC) and usually has a rating of a fraction (25%-30%) of the DFIG nominal power. As a consequence, the WTG can operate in a large range (±25%-30%) of the rotational speed around the synchronous speed, and its active and reactive power can be controlled independently. Typically this type of wind turbine is also equipped with a blade pitch control to limit the extracted power during conditions of high wind speeds.

The DDSG shown in Fig. 3 can have a wound rotor or be excited by permanent magnets [23]. It is typically a multipole low-speed machine that is directly coupled to the turbine and connected to the power grid through a variable frequency de-link-voltage converter. Again, power extracted at high wind speeds can be limited by controlling the pitch of the blades. No gearbox is needed in such a system, but this advantage must be paid for by the disadvantage of a larger power electronic converter and a more complicated and expensive generator.

Fig. 1. Configuration of a fixed-speed SCIG wind turbine connected to a power grid.

Fig. 2. Configuration of a variable-speed DFIG wind turbine connected to a power grid.

Fig. 3. Configuration of a variable-speed DDSG wind turbine connected to a power grid.
III. REAL-TIME WIND SPEED ESTIMATION ALGORITHM

A. The Principle of Wind Speed Estimation

For a WTG system, the electrical power generation from wind energy can be described by the mathematical models in the flow chart in Fig. 4. The conversion of wind energy ($P_w$) to turbine mechanical power ($P_m$) is represented by a wind turbine aerodynamic model. In this model, the turbine mechanical power $P_m$ is represented as a nonlinear function of the wind speed $v_w$, turbine shaft speed $\omega_n$ and blade pitch angle $\beta$. The dynamics of the WTG shaft system is represented by a set of differential equations. Consequently, the mechanical power is transferred from the turbine to the generator ($P_e$). The generator converts mechanical power into electrical power ($P_e$). The losses ($P_{\text{loss}}$) of the system (referred to the generator side) are considered in the model. The output electrical power $P_e$ of the generator is measured. Therefore, if developing an inverse model from $P_e$ to $v_w$, then the wind speed can be estimated from the measured output electrical power. In this paper, such an inverse model is approximated by using an ESN, as shown in Fig. 5. The benefit of using a neural network is that the WTG system can be treated as a black box. This removes the need of developing an explicit mathematical model as shown in Fig. 4 to represent the complex dynamics of the WTG system. In Fig. 5, the black box (i.e., the WTG) represents the nonlinear dynamics from ($v_w$, $\omega_n$, $\beta$) to $P_e$ of the WTG system; the ESN provides an inverse model of such WTG dynamics to estimate the wind speed $\hat{v}_w$ in real time from the measured data ($P_e$, $\omega_n$, $\beta$).

![Fig. 5. Wind speed estimation using an ESN.](image)

B. Echo State Networks

ESNs are a novel approach for designing and supervised training of RNNs. An ESN consists of a large, fixed RNN called dynamical reservoir (DR), which can be excited by suitably presented input and/or fed-back output. The desired output dynamics are then obtained by training the connection weights from the DR to the output only; while all other connection weights are fixed during the training.

The general architecture of an ESN is shown in Fig. 6. It is a discrete-time ANN with $M$ input units (i.e., neurons), $N$ internal units in the DR, and $L$ output units. The activations at the time step $k$ of the input units are $\mathbf{u}(k) = [u_1(k), u_2(k), \ldots, u_M(k)]$, of the internal units of the DR (i.e., the states) are $\mathbf{x}(k) = [x_1(k), x_2(k), \ldots, x_N(k)]$, and of the output units are $\mathbf{y}(k) = [y_1(k), y_2(k), \ldots, y_L(k)]$. The connection weights from the input to the DR are collected in an $N \times M$ matrix $W_m = (w_{mi})$; the internal connection weights of the DR are collected in a $N \times N$ matrix $W = (w_{ij})$; the connection weights from the input and the DR to the output as well as the self-connection weights of the output units are collected in a $L \times (M + N + L)$ matrix $W_{\text{out}} = (w_{ij}^\text{out})$; the feedback connection weights from the output to the DR are collected in a $N \times L$ matrix $W_{\text{back}} = (w_{ij}^\text{back})$. The output weights $W_{\text{out}}$ (shown as the dash-line arrows in Fig. 6) will be learned. The input weights $W_m$, internal weights $W$, and the output feedback weights $W_{\text{back}}$ (shown as the solid-line arrows in Fig. 6) are fixed before learning. The internal weights are typically in a sparse, random connectivity pattern.

The activation of the internal units of the DR is updated by

$$x(k+1) = f(W_m \mathbf{u}(k+1) + W \mathbf{x}(k) + W_{\text{back}} \mathbf{y}(k))$$  (1)

where $f$ is the element-wise activation function of the internal units. The most commonly used activation functions include the sigmoid function and the hyperbolic tangent function, as described by (2) and (3), respectively.

$$f(z) = \frac{1}{1 + e^{-z}}$$  (2)

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = \frac{e^{2z} - 1}{e^{2z} + 1}$$  (3)

The activation of the output units is updated by

$$y(k+1) = f(\mathbf{w}_{\text{out}}^\top [\mathbf{u}(k+1), \mathbf{x}(k+1), \mathbf{y}(k)])$$  (4)
where $f^{\text{out}}$ is the element-wise activation function of the output units. The commonly used forms of $f^{\text{out}}$ include the sigmoid function (2), the hyperbolic tangent function (3), and the identity matrix. In the last case, the output units are simply linear weighted combination of the input, internal, and output units.

**C. Design of the Echo State Network for Real-Time Wind Speed Estimation**

In the proposed wind speed estimation algorithm, an ESN is used to provide an inverse model of the WT's dynamics to estimate the wind speed $\hat{v}_w$, provided that values are known for the WT's output electrical power $P_e$, turbine shaft speed $\omega_s$, and blade pitch angle $\beta$, as shown in Fig. 7. Therefore in this application, the input and output activations of the ESN at the time step $k$ are $u(k) = [P_e(k), \omega_s(k), \beta(k)]$ and $y(k) = \hat{v}_w(k)$, respectively. The activation functions of the internal and output units are both sigmoid functions described by (2). Wind blowing is a highly stochastic process in which the wind speed at the present time step is usually irrelevant to those in previous time steps. Therefore, it is not necessary to use the output feedback term in (1) and (4). Consequently, the activations of the internal and output units in this design can be updated according to the following two equations.

$$x(k+1) = f(W_{\text{in}}) u^T(k+1) + Wx^T(k)$$

$$y(k+1) = f^{\text{out}}(W_{\text{out}} [u(k+1), x(k+1)]]$$

The ESN is trained to learn the nonlinear dynamics from $(P_e, \omega_s, \beta)$ to $v_e$ of the WT system. A key issue prior to training the network $(W_{\text{in}}, W, W_{\text{out}})$ in Fig. 7 is that it must have the property of echo states. According to [19], a trained network $(W_{\text{in}}, W, W_{\text{out}})$ is an ESN if the untrained core $(W_{\text{in}}, W)$ (i.e., the DR) has the echo state property. The echo state property of DR can be defined as follows [19]. Assume the DR starts from two arbitrary states $x(0)$ and $x'(0)$ and is updated according to (5) with the same input sequence in both cases, then it is said to have the property of echo states if the resulting state $x(k)$ and $x'(k)$ converge to each other. In other words, if the network $(W_{\text{in}}, W, W_{\text{out}})$ and the DR have the property of echo states, then the states $x(k)$ of the DR are independent of the initial conditions and will asymptotically depend only on the input history.

The echo state property of the DR is connected to algebraic properties of the weight matrix $W$. A sufficient condition to ensure an ESN is that the matrix $W$ is contractive. In practice it is consistently found that if the spectral radius $|\lambda_{\text{max}}|$ of $W$ is less than unity, the resulting network is an ESN [19].

The weight matrix $W_{\text{in}}$ is irrelevant to the echo state property of the network [19]. In practice $W_{\text{in}}$ is a matrix with randomly generated elements.

Once the weight matrices $W_{\text{in}}$ and $W$ are generated, they are fixed during training. Therefore, training the ESN only involves calculating the output weights $W_{\text{out}}$. This task is usually done by minimizing the following mean-square-error (MSE) over selected training data sets.

$$MSE = \frac{1}{K-I} \sum_{k=I}^{K} \left( f^{\text{out}}(y(k) - W_{\text{out}} [u(k), x(k)]) \right)^2$$

where $u(k) \in U$; $y(k) \in Y$; $U = \{u(k) \mid k = 1, ..., K\}$ and $Y = \{y(k) \mid k = 1, ..., K\}$ are the input and output training data sets of the network; the states $x$ of the DR are updated by (5). The first $I$ data in $U$ and $Y$ are used to die out the initial transient of the DR, and therefore, are not used for training the network. Only the rest $K-I$ data in $U$ and $Y$ are collected for training. Consequently, the following two matrices can be obtained:

$$A = (f^{\text{out}})^{-1} [y(I+1), ..., y(K)]$$

$$B = [B(I+1), ..., B(K)]$$

where $B(k) = [u(k), x(k)]]^T, k = I+1, ..., K$. Then the output weights $W_{\text{out}}$ can be calculated as follows by pseudoinverse method:

$$W_{\text{out}} = AB^\dagger (BB^\dagger)^{-1}$$

There are three key steps to design and train the ESN.

**Step 1**: procure an untrained DR network with fixed $W_{\text{in}}$ and $W$ which has the echo state property. This step involves the following substeps.

1. Randomly generate a sparse internal weight matrix $W_0$ with the weights uniformly distributed over $[-1, 1]$ and a density of $d$.
2. Normalize $W_0$ to a matrix $W_1$ with a unit spectral radius by $W_1 = W_0/|\lambda_{\text{max}}|$, where $|\lambda_{\text{max}}|$ is the spectral radius of $W_0$. The commonly used forms of $f^{\text{out}}$ include the sigmoid function (2), the hyperbolic tangent function (3), and the identity matrix. In the last case, the output units are simply linear weighted combination of the input, internal, and output units.
radius of \( W_0 \).

3. Scale \( W_t \) to \( W = aW_0 \), where \( a < 1 \). Therefore, \( W \) has a spectral radius of \( a \). Then the DR network has the property of echo states, regardless of how \( W_0 \) is chosen.

4. \( W_0 \) is usually randomly generated. In this design, the weights in \( W_0 \) are uniformly distributed over \([-0.1, 0.1]\).

**Step 2:** generate training data sets. This step includes the following substeps.

1. At each time step \( k = 1, \ldots, K \), sample the data of wind speed \( v_\infty(k) \), WTG output electrical power \( P_\infty(k) \), turbine shaft speed \( \omega_t(k) \), and blade pitch angle \( \beta(k) \); collect these data samples into the input and output activation vectors \( \psi(k) = [P_\infty(k), \omega_t(k), \beta(k)] \) and \( y(k) = v_\infty(k) \) of the ESN, respectively.

2. Initialize the DR network states, e.g., to zero state \( x(0) = 0 \).

3. Update the DR network according to (5) by the training data samples for \( k = 0, \ldots, I \). At \( k = 0 \) where \( y(k) \) is not defined use \( y(k) = 0 \).

4. For each time step \( k > I \), collect the input data samples and the network states \( B(k) = [\psi(k), x(k)^T] \) as a new column into a matrix \( B \). Finally, a matrix \( B \) of the size \((M+N)(K-I)\) is obtained.

5. Similarly, for each time step \( k > I \), collect \( A(k) = (f^{(m)})^{-1} y(k) \) as a new column into a matrix \( A \). Finally, a matrix \( A \) of the size \( L(K-I) \) is obtained.

**Step 3:** Calculate the output weight matrix \( W_{out} \) using (10).

Once the output weights are determined, the ESN is then used online by the WTG for real-time wind speed estimation.

### IV. Wind-Speed-Sensorless Control Of A DFIG Wind Turbine

The mechanical power that a wind turbine extracts from wind can be calculated by

\[
P_m = \frac{1}{2} \rho A v_\infty^3 C_p(\lambda, \beta)
\]

where \( \rho \) is the air density in kg/m\(^3\); \( A \) is \( \pi R^2 \) is the area swept by the rotor blades in m\(^2\); \( v_\infty \) is the wind speed in m/s at the hub height; \( C_p \) is the power coefficient, which is a function of both tip speed ratio \( \lambda \) and the blade pitch angle \( \beta \).

The tip speed ratio \( \lambda \) is defined by:

\[
\lambda = \frac{\omega_t R}{v_\infty}
\]

where \( R \) is the wind turbine rotor radius in m. The \( C_p, \lambda, \beta \) characteristics depend on the blade design. At any wind speed within the operating range, there is a unique wind turbine shaft speed \( \omega_t \) to achieve the maximum power coefficient \( C_{pm} \). According to (11), when \( C_p \) is controlled at the maximum value, the wind turbine extracts the maximum mechanical power from the wind. If the wind speed is below the rated value, the WTG operates in the variable-speed mode, and the turbine shaft speed \( \omega_t \) is adjusted optimally at \( \omega_{t,opt} \). This results in an optimal tip speed ratio \( \lambda_{opt} \) such that \( C_{p} \) remains at the \( C_{pm} \) point.

\[
\lambda_{opt} = \frac{\omega_{t,opt} R}{v_\infty}
\]

At this optimal condition, the optimal generator rotor speed \( \omega_{r,opt} \) is proportional to the wind speed, given by:

\[
\omega_{r,opt} = k_w v_\infty
\]

where the coefficient \( k_w \) is determined by:

\[
k_w = \frac{N_i}{N_g} \frac{\lambda_{opt}}{R}
\]

where \( N_i/N_g \) is the gear ratio between the turbine and the generator. In the variable-speed operating mode, the wind turbine pitch control is deactivated. However, if the wind speed increases above the rated value, the pitch control is activated to increase the wind turbine pitch angle to reduce the mechanical power that the turbine extracts from the wind.

The proposed real-time wind speed estimation algorithm is integrated into the control system of a DFIG wind turbine, as shown in Fig. 8. The overall control system consists of two parts: the electrical control of the DFIG and the mechanical control of the wind turbine blade pitch angle. Control of the DFIG is achieved by controlling the variable-frequency ac/dc/ac converter, which includes control of the RSC and control of the GSC. Based on the estimated wind speed \( \hat{v}_w \), the optimal DFIG rotor speed reference \( \omega_{r,opt} \) is determined by (14), which is then used by the RSC controller. The RSC controller regulates the DFIG rotor speed \( \omega_r \) optimally to generate the maximum electrical power from the wind as well as to maintain the DFIG stator terminal voltage \( V \) at the desired value. The control objective of the GSC is to keep the dc-link voltage \( V_{dc} \) of the variable-frequency ac/dc converter constant regardless of the magnitude and direction of the rotor power. The pitch control regulates the wind turbine blade pitch angle \( \beta \), which determines the mechanical power that the turbine extracts from the wind. A detailed description of the DFIG wind turbine control system is given in [8].

The proposed wind speed estimation algorithm removes the need of using mechanical wind speed sensors to measure wind speed. The resulting control system provides a wind-speed-sensorless control for the DFIG wind turbine.

### V. Simulation Results

Simulation studies are carried out for a 3.6-MW DFIG wind turbine [8], [21] to demonstrate the effectiveness of the proposed real-time wind speed estimation algorithm. In this application, the numbers of the input and output units of the ESN are \( M = 3 \) and \( L = 1 \), respectively. Other parameters of the ESN are chosen as follows by trial and error: the number of the internal units in the DR is \( L = 100 \); the density of the weight matrix \( W \) is \( d = 10\% \); the scaling factor is \( \alpha = 0.7 \).
A. Training of the ESN

The DFIG wind turbine in Fig. 8 is operated under various variable wind conditions from low to high wind speeds. The training data sets are therefore generated over a wide system operating range for training the ESN. This ensures that the trained ESN can approximate the dynamics of the WTG over a wide range of system operating conditions. Fig. 9 shows a typical wind speed profile used for generating the training data, in which the wind speeds vary in the range of ±3.5 m/s around the mean value of 12 m/s. The training data are collected with a sampling period of 10 ms when the WTG system in Fig. 8 is operated under these wind speed conditions. The overall collected training data are then used to calculate the output weights of the ESN. The estimated wind speeds by the ESN are shown in Fig. 9 as well, which track the actual wind speeds with good precision. The average tracking error is 0.108 m/s, which is less than 1% of the mean wind speed. The average tracking error $E$ is defined as follows:

$$ E = \frac{1}{N_T} \sum_{k=1}^{N_T} \left| \dot{v}_w(k) - \hat{v}_w(k) \right| $$

where $v_w$ and $\hat{v}_w$ are the actual and estimated wind speeds, respectively; $N_T$ is the total number of data samples.

B. Real-Time Wind Speed Estimation Results

The trained ESN is used by the WTG control system as shown in Fig. 8 for real-time wind speed estimation and control. To demonstrate the effectiveness of the proposed algorithm, a different wind speed profile from that in Fig. 9 is now applied to the WTG. This wind speed profile has not been used for training the ESN. Fig. 10 compares the actual wind speeds and the estimated wind speeds generated by the ESN. The wind speeds vary in the range of ±2 m/s around the mean value of 11 m/s. The estimated wind speeds track the actual wind speeds accurately and the average tracking error is 0.058 m/s, which is less than 0.6% of the mean wind speed.

Fig. 11 compares WTG speed control performance when using the actual wind speeds and the estimated wind speeds. By using the accurately estimated wind speed, the sensorless control strategy regulates the WTG system with the same performance as in the case of using anemometers. The tip speed ratio $\lambda$ calculated by (12) of the wind turbine varies in a small range of [-0.13, 0.12] around the optimal tip speed ratio $\lambda_{opt} = 5.96$, as shown in Fig. 12. This result indicates that the WTG is always controlled at the optimal operating condition to generate the maximum electrical power from the wind.
Electrical power generation from wind is realized by a wind turbine generator (WTG) system. Such a system is usually equipped with one or more anemometers to measure wind speed for system monitoring, control, and protection. The use of these mechanical sensors increases the cost and hardware complexity and reduces the reliability of the WTG system.

This paper has proposed a novel echo-state-network (ESN)-based real-time wind speed estimation algorithm to solve the problem of using anemometers. An ESN was developed to provide a nonlinear inverse model of the WTG dynamics, which was used to estimate the wind speed in real time from the measured WTG output electrical power, shaft speed, and blade pitch angle. The estimated wind speed was then used for wind-speed-sensorless control of the WTG system. This removed the need of mechanical sensor to measure wind speed. The proposed algorithm has been verified by simulation studies on a 3.6-MW wind turbine equipped with a doubly fed induction generator (DFIG). Results have shown that it accurately estimates the wind speed in real time during various wind conditions and the resulting sensorless control strategy provides a precise control for maximum power generation of the WTG system.

VI. CONCLUSION

REFERENCES


