Robust Adaptive Control of Maximum Power Point Tracking for Wind Power System

PENG CHEN\textsuperscript{1}, DEZHI HAN\textsuperscript{1}, (Member, IEEE), AND KUAN-CHING LI\textsuperscript{2}, (Senior Member, IEEE)

\textsuperscript{1}Department of Computer Science and Technology, Shanghai Maritime University, Shanghai 200120, China
\textsuperscript{2}Department of Computer Science and Information Engineering (CSIE), Providence University, Taichung 43301, Taiwan

Corresponding authors: Dezhi Han (dzhan@shmtu.edu.cn) and Kuan-Ching Li (kuancili@pu.edu.tw)

This work was supported by the National Natural Science Foundation of China under Grant 61672338 and Grant 61873160.

ABSTRACT A novel data-driven robust approximate optimal Maximum Power Point Tracking (MPPT) control method is proposed for the wind power generation system by using the adaptive dynamic programming (ADP) algorithm. First, a data-driven model is established by a recurrent neural network (NN) to reconstruct the wind power system dynamics using available input-output data. Then, in the design of the controller, based on the obtained data-driven model, the ADP algorithm is utilized to design the approximate optimal tracking controller, which consists of the steady-state controller and the optimal feedback controller. Ulteriorly, developing a robustifying term to compensate for the NN approximation errors introduced by implementing the ADP method. Based on the Lyapunov approach, it proves the stability of the designed model and controller to show that the proposed controller guarantees the system power asymptotically tracking the maximum power. Finally, the simulation results demonstrate that the control method stabilizes the tip speed ratio near the optimal value when the wind speed is lower than the rated wind speed. Moreover, the tracking response speed of the proposed method is fast, which enhances the stability and robustness of the system.

INDEX TERMS Adaptive dynamic programming, maximum power point tracking, neural network, wind power system.

I. INTRODUCTION With the rapid development of the social economy, traditional energy sources are gradually depleted. Therefore, the development of sustainable renewable energy is imminent. As a widely distributed clean and renewable energy, wind energy has received universal attention in the world and has delightful application prospects. To protect the environment and slow down the energy depletion of the world, the installed capacity of wind power continues to grow worldwide [1].

The wind power generation system cannot convert all the captured wind energy into electrical energy. The wind energy utilization coefficient is an index indicating the conversion efficiency of wind energy into electrical energy. The wind energy utilization coefficient varies with wind speed, so the maximum power point of its power output changes with wind speed. Since the randomness of wind speed and the nonlinearity of the wind power generation system, its output characteristics are greatly affected by the external environment, it is difficult to capture the maximum power point. The conversion efficiency of the wind power generation system is low, and the equipment is expensive, it is necessary to improve the utilization rate of wind energy and increase economic benefits through the maximum power point tracking (MPPT) control [2].

There are many control strategies to achieve MPPT, mainly include the tip speed ratio method, power curve control method, three-point comparison method, mountain climbing method, etc. [3]. Researchers at home and abroad have also proposed a large number of control methods for the nonlinear characteristics of wind power generation systems. In [4], the researchers present a fuzzy PID compound controller, which combines fuzzy control and traditional PID control technology to achieve the maximum power output of the generator, but its control accuracy is not high. Reference [5] proposed a fuzzy neural network controller that combines neural network and fuzzy logic control. It uses indirect vector control and reactive power control technology to control the power transmission between the machine and the power grid, but control rules of fuzzy logic are complex, and its
design process is based on experience. When it is difficult to obtain an accurate wind turbine power curve, a method that combines sliding mode control and mountain climbing search algorithm is proposed in [6]. This method can track the optimal power curve without measuring wind speed, but it complicates the modeling process and its requirements for parameter selection and adjustment are high.

Adaptive dynamic programming (ADP) is an effective intelligent control algorithm, which plays an important role in searching for optimal control solution [7]. Compared with the intelligent control method mentioned above, adaptive control [8] can make the controller have the ability to automatically adjust the learning process when the environment or the system changes, which is very useful for the wind power generation system with many uncertain disturbances. As an effective method of approximating mapped input-output data, it has applied a neural network in many nonlinear control systems [9]–[11]. The neural network can adjust the network weight on-line adaptively without pre-training, which can guarantee the stability and performance of the closed-loop system. Through the above analysis, we can see it that the application of adaptive neural network control to uncertain nonlinear dynamic systems such as wind power generation systems has relative significance. Therefore, in this paper, we propose a data-driven robust adaptive control (DRAC) method for maximum power tracking of the wind power generation system, which changes the wind energy utilization coefficient by controlling the rotation speed of the wind turbine to keep the wind energy utilization coefficient at the maximum value, so as to guarantee the system to asymptotically track the maximum power when the wind speed is under the rated wind speed. Then, by adding a robust compensation term to improve the robustness of the system to the disturbance, reduce the uncertainty of on-line estimation, and speed up the convergence speed of the system.

A. RESEARCH CONTRIBUTIONS

1) Based on the nonlinear equations of wind power generation and adaptive dynamic programming algorithm, combined with recurrent neural network (RNN), we propose a data-driven robust adaptive controller method for maximum power tracking of the wind power generation system. The conclusion and prospects are provided in section VI.

2) Combine adaptive dynamic programming, actor-critic structure [12] and neural network to design the adaptive controller. The adaptive dynamic programming algorithm is used to approximate the optimal feedback control input value so that the control input error converges to a tiny range, which proves the tracking effectiveness of the controller.

3) we design a robust compensation term to make the control input error converge to zero, reduce the influence of uncertain disturbance terms on the tracking effect of the wind power generation system, and enhance the robustness and performance of the controller.

B. PAPER STRUCTURE

The rest of this paper is organized as follows. In section II, some preliminary knowledge and preparation for the subsequent controller design are introduced. In section III, we introduce the construction of the data-driven model of the wind power generation system and its stability proof. The design of the robust adaptive controller and its stability analysis is introduced in detail in section IV. In section V, the simulation experiment in MATLAB is given. Finally, the conclusion and prospects are provided in section VI.

II. PROBLEM FORMULATION AND PRELIMINARIES

In this section, we briefly introduce the wind power system model, the principle of maximum power tracking control strategy based on the tip speed ratio method, and the related knowledge involved in the control method proposed in this paper.

A. WIND POWER GENERATION SYSTEM MODEL AND MAXIMUM POWER TRACKING CONTROL

According to the aerodynamic characteristics of the wind power generation system [13], [14], the wind energy captured by the wind turbine is the active power output \( P_m \) of the pneumatic machinery, which is defined as:

\[
P_m = \frac{1}{2} \rho \pi R^2 v^3 C_p \tag{1}
\]

where \( \rho \), \( R \), \( v \) and \( C_p \) denote the air density, blade radius, wind speed and wind energy utilization coefficient, respectively.

\( \lambda \) is the tip speed ratio, which can be measured by the ratio of the blade tip circumferential speed to the wind speed, which can be defined by (2):

\[
\lambda = \frac{\omega R}{v} \tag{2}
\]

where \( \omega \) is the angular velocity of wind rotor.

\( C_p \) is a nonlinear function of tip speed ratio and pitch angle, and its mathematical model can be obtained according to the method of numerical approximation calculation [15], [16]. The empirical expression equation is

\[
C_p(\lambda, \beta) = 0.5176 \left( \frac{116}{\Lambda} - 0.4\beta - 5 \right) e^{\left( -\frac{\lambda}{3} \right)} + 0.0068\lambda
\]

where \( \beta \) is the pitch angle; \( \Lambda \) is the intermediate variable. It can obtain the curve of the wind energy utilization coefficient \( C_p(\lambda, \beta) \) function according to (3)-(4), as shown in Fig.1.

According to (1), when the wind speed is stable, the active power obtained by the wind turbine is proportional to the wind energy utilization coefficient \( C_p \). The higher the wind...
energy utilization coefficient is, the higher the wind energy conversion efficiency will be, and the greater the power of the wind turbine will be. Then, further observation of Fig.1 shows that within a certain range, the smaller the pitch angle, the greater the wind energy utilization coefficient; when the pitch angle is fixed, there is an optimal tip speed ratio $\lambda_{opt}$ to maximize the wind energy utilization coefficient. Therefore, when the wind speed is less than the rated wind speed, fixed the pitch angle $\beta = 0$. By controlling and adjusting the wind turbine speed to the optimal angular speed $\omega_{opt}$, $\lambda$ can reach the optimal tip speed ratio $\lambda_{opt}$, and the wind energy utilization coefficient can reach the maximum $C_{p_{max}}$, to obtain the maximum output power. In this way, the maximum power point tracking (MPPT) control of the wind power generation system can be realized. It shows the schematic diagram of the control strategy principle in Fig.2.

Recruent neural network (RNN) is a type of neural network that takes sequence data as input and establishes a full connection between neurons between layers [18]. RNN has the characteristics of memorability, parameter sharing, and Turing completeness, so it has certain advantages in learning nonlinear features of sequences [19]. Therefore, based on the RNN model, this paper learns the nonlinear characteristics of the wind power generation system and constructs its data-driven model.

### B. RECURRENT NEURAL NETWORK

Fig.3 shows the structure diagram of the RNN. It relates the output at each moment to the input at that moment and the output at the previous moment.

$$X_t = \{X_0, X_1, \ldots, X_t\}$$ is the given time series, the length is $\tau$. The evolution direction of this sequence is called “time-step’’. $h_t$ is the system state of the recurrent neural network, $s_t$ is the internal state, which is related to the system state.

$$s_t = UX_t + Wh_{t-1} + b$$  \hspace{1cm} (7)

$$h_t = \sigma (s_t) = \sigma (UX_t + Wh_{t-1} + b)$$  \hspace{1cm} (8)

where $U$ and $W$ are the weights of the neural network, and $b$ is the bias; $\sigma$ represents the activation function. The output function of the current hidden layer $o_t$ is defined as follows:

$$o_t = Vh_t + c$$  \hspace{1cm} (9)

where $V$ and $c$ are weight coefficients. The weight update can be iterated using the gradient descent algorithm until the required parameters are obtained.

However, due to the activation function, with multi-parameters, the problem of gradient disappearance and gradient explosion is prone to occur in the learning process of recurrent neural networks, so it is necessary to take corresponding measures to solve such problems [18], [19].

### C. ADAPTIVE DYNAMIC PROGRAMMING

Adaptive dynamic programming (ADP) is a novel approach for approximately solving nonlinear optimal control problems [20], [21]. It effectively overcomes the difficulty of solving the Hamilton-Jacobi-Bellman (HJB) equation and the “dimensional disaster” problem in dynamic programming (DP) [22]–[23]. The HJB equation is a partial differential equation, and its solution refers to a real-valued function that satisfies the minimum value of the performance function of a specific dynamic system.

Taking a continuous system as an example, the corresponding HJB equation of the system $F [x (t), u (t)]$ is derived as

$$\frac{d}{dt} V(x(t)) = \min_{u(t)} [F(x(t), u(t)) + \nabla V(x(t)) \cdot \nabla u(t)]$$

The schematic diagram of the RNN.

---

**FIGURE 1.** The curve of the wind energy utilization coefficient $C_p - \lambda$.

**FIGURE 2.** The schematic diagram of the control strategy principle.

**FIGURE 3.** The structure diagram of the RNN.
shown in equations (10)-(12) [24].

$$V(x(t), t) = \int_{t}^{\infty} r(x(\tau), u(\tau)) d\tau$$  \(10\)

$$V^{*}(x(t), t) = \min_{u}[V(x(t), t)]$$  \(11\)

$$\dot{V}^{*} + \min_{u} \left\{ \nabla V^{*} \cdot F[x(t), u(t)] + r(x(t), u(t)) \right\} = 0$$  \(12\)

where \(V(x(t), t)\) is the system performance function; \(V^{*}(x(t), t)\) is the minimum value of \(V(x(t), t)\); \(r(x(t), u(t))\) is the utility function related to the system state \(x(t)\) and the control input \(u(t)\); \(\dot{V}^{*}\) is the derivative of \(V^{*}(x(t), t)\) with respect to time \(t\); \(\nabla V^{*}\) is the gradient of \(V^{*}\) with respect to state \(x\). According to Bellman optimal principle, the goal of optimal control is to find an admissible control \(u(t)\) that minimizes the performance function and satisfies (12).

By observing equations (10)-(12), we know that the solution of the HJB equation is difficult. Therefore, the ADP algorithm uses a function structure to approximate the solution of the HJB equation, and then uses the offline iteration or online update method to obtain the approximate optimal control solution.

The Dual heuristic dynamic programming [25] (DHP), one of the basic structures of ADP, is shown in Fig. 4.

In Fig. 4, it uses a control network to map the relationship between the system state and the control input, and the control input is output; the output of the critic network is used to estimate the gradient of the performance function \(V(x(t), t)\). In this paper, the ADP algorithm is applied to the MPPT control strategy of the wind power generation system. The difference in the angular speed of the rotor rotation is used as the input, and the generator torque control value is the output. The specific steps are explained in the Algorithm 1.

### III. DATA-DRIVEN MODEL OF THE WIND POWER GENERATION SYSTEM

This section discusses the establishment of a data-driven model of wind power generation systems based on recurrent neural networks. The existing state values and control values of the system are used as input-output data of the neural network. The neural network uses these data for learning and training and then gets the trained model. Finally, its stability is analyzed.

#### A. ESTABLISHMENT OF DATA-DRIVEN MODEL

First, we rewrite the simplified dynamic equation of wind power transmission system (5) to the standard affine form:

$$\dot{\Omega} = G(\omega, v) + gu$$  \(13\)

where \(u = T_e\) is the input of the network, \(G(\omega, v)\) and \(g\) is:

$$G(\omega, v) = \frac{1}{2} \rho \pi R^2 \frac{3 C_2}{\omega} J_r + N^2 J_g$$  \(14\)

$$g = -\frac{N}{J_r + N^2 J_g}$$  \(15\)

Then, the data-driven model of wind power generation system is constructed by on-line approximation based on recurrent neural network, rewrite (13) as RNN [26]:

$$\dot{\omega}(t) = A^\omega \omega(t) + B^\omega f(\omega(t)) + A^\nu \nu(t) + B^\nu \nu(t)$$

$$+ B^\mu \mu(t) + gu(t) + \varepsilon(t)$$  \(16\)

where suppose \(\varepsilon(t)\) is a bounded neural network approximation error term; \(A^\omega, B^\omega, A^\nu, \text{ and } B^\nu\) are unknown ideal weights, and \(g\) is a constant, which can be obtained by (15).

The activation function \(f(\cdot) = tanh(\cdot)\) in (16) is a monotonically increasing function that satisfies:

$$0 \leq f(x) - f(y) \leq k(x - y)$$  \(17\)

for any \(x, y \in \mathbb{R}\) and \(x \geq y, k > 0\).

Based on (16), the data-driven model of wind power generation system can be constructed as:

$$\dot{\omega}(t) = \hat{A}(t) \dot{\omega}(t) + \hat{B}(t) f(\dot{\omega}(t)) + \hat{A}_u(t) \nu(t) + \hat{B}_u(t) + gu(t) - d(t)$$  \(18\)

where \(d(t)\) defined as:

$$d(t) = Se_m(t) + \frac{\hat{k}(t)e_m(t)}{\hat{e}_m(t) + \eta}$$  \(19\)
where $\hat{\omega} (t)$ is the estimate of $\omega (t)$; $\hat{A} (t)$, $\hat{B} (t)$, $\hat{A}_u (t)$ and $\hat{B}_u (t)$ are the estimate of unknown ideal weights at $t$; $e_m (t) = \omega (t) - \hat{\omega} (t)$ is the modeling error of $\omega (t); S \in \mathbb{R}$ is a given design term; $\hat{k} (t) \in \mathbb{R}$ is an additional tunable parameter, the role is to increase the flexibility of neurons and improve the fitting ability; $\eta > 1$ is a constant.

It can make the following assumptions [27]:

Assumption 1: The upper bound of $\varepsilon (t)$ is defined by a function related to $e_m (t)$ in the following form:

$$e(t)^2 \leq e_M(t) = \kappa^2 e_m^2(t) \tag{20}$$

where $\kappa^2$ is the target of bounded constant.

B. STABILITY ANALYSIS OF DATA-DRIVEN MODEL

From (16) and (18), the dynamic equation of modeling error $e_m (t)$ is

$$
e_m (t) = A^* e_m (t) + \hat{A} (t) \hat{\omega} (t) + B^* \hat{f} (e_m (t)) + \hat{B}_u (t) v (t) + \hat{\kappa} (t) e_m (t) + \kappa^* e_m^2 (t) + \frac{\kappa^* e_m^2 (t)}{e_m^2 (t) + \eta} \tag{21}$$

where $\hat{A} (t) = A^* - \hat{A} (t)$, $\hat{B} (t) = B^* - \hat{B} (t)$, $\hat{A}_u (t) = A^*_u - \hat{A}_u (t)$, $\hat{B}_u (t) = B^*_u - \hat{B}_u (t)$, $\hat{f} (e_m (t)) = f (\omega (t)) - f (\hat{\omega} (t))$, $\hat{\kappa} (t) = \kappa^* - \hat{\kappa} (t)$.

Theorem 1: Update the weights and parameters of RNN according to the rules of equations (22)-(26). When $t \to \infty$, the modeling error $e_m (t)$ gradually converges to zero [28].

The update rules of the RNN weight and an additional tunable parameter $\hat{\kappa}$ are formulated as follows:

$$
\hat{A} (t) = \gamma_1 \hat{\omega} (t) e_m (t) \tag{22}
$$

$$
\hat{B} (t) = \gamma_2 f (\hat{\omega} (t)) e_m (t) \tag{23}
$$

$$
\hat{A}_u (t) = \gamma_3 v (t) e_m (t) \tag{24}
$$

$$
\hat{B}_u (t) = \gamma_4 e_m (t) \tag{25}
$$

$$
\hat{\kappa} (t) = -\gamma_5 \frac{e_m^2 (t)}{e_m^2 (t) + \eta} \tag{26}
$$

where $\gamma_i$ is a positive number, $i = 1, 2, 3, 4, 5$. To prevent gradient explosion during weight update, the weight matrix is normalized after each iteration.

Proof: According to the Lyapunov Theorem, the Lyapunov function is constructed as follows:

$$J (t) = J_1 (t) + J_2 (t) \tag{27}$$

where

$$J_1 (t) = \frac{1}{2} e_m^2 (t)$$

$$J_2 (t) = \frac{1}{2} \left( \hat{A} (t) \gamma_1^{-1} \hat{A} (t) + \hat{B} (t) \gamma_2^{-1} \hat{B} (t) + \hat{A}_u (t) \gamma_3^{-1} \hat{A}_u (t) + \hat{B}_u (t) \gamma_4^{-1} \hat{B}_u (t) \right) + \frac{1}{2} \hat{\kappa} (t) \gamma_5^{-1} \hat{\kappa} (t)$$

Then calculate the derivative of $J (t)$ along the trajectory of $e_m (t)$ (21) with respect to time:

$$J_1 (t) = A^* e_m^2 (t) + e_m (t) \hat{A} (t) \hat{\omega} (t) + e_m (t) B^* \hat{f} (e_m (t)) + e_m (t) \hat{B}_u (t) v (t) + e_m (t) \hat{\kappa} (t) e_m (t) + \kappa^* e_m^2 (t) \frac{\kappa^* e_m^2 (t)}{e_m^2 (t) + \eta} \tag{28}$$

From (17), we can get:

$$e_m (t) B^* \hat{f} (e_m (t)) \leq \frac{1}{2} B^2 \hat{f}^2 (e_m (t)) + \frac{1}{2} \kappa^2 e_m^2 (t) \tag{29}$$

According to Assumption 1, we have

$$e_m (t) \varepsilon (t) \leq \frac{1}{2} e_m^2 (t) + \frac{1}{2} \kappa^* e_m^2 (t) \tag{30}$$

Therefore (28) can be rewritten as

$$J_1 (t) \leq A^* e_m^2 (t) + e_m (t) \hat{A} (t) \hat{\omega} (t) + \frac{1}{2} e_m (t) B^2 \hat{f}^2 (e_m (t)) + e_m (t) \hat{A}_u v (t) + \left( \frac{1}{2} + \frac{1}{2} \kappa^* + \frac{1}{2} \kappa^2 \right) \frac{e_m^2 (t)}{e_m^2 (t) + \eta} + \frac{1}{2} \hat{\kappa} (t) e_m (t) + \kappa^* \frac{e_m^2 (t)}{e_m^2 (t) + \eta} \tag{31}$$

Computing the time derivative of $J_2 (t)$

$$J_2 (t) = \left( \hat{A} (t) \gamma_1^{-1} \hat{A} (t) + \hat{B} (t) \gamma_2^{-1} \hat{B} (t) + \hat{A}_u (t) \gamma_3^{-1} \hat{A}_u (t) + \hat{B}_u (t) \gamma_4^{-1} \hat{B}_u (t) \right) + \frac{1}{2} \hat{\kappa} (t) \gamma_5^{-1} \hat{\kappa} (t) \tag{32}$$

Combining (31) with (32), we can obtain

$$J (t) \leq A^* e_m^2 (t) + \frac{1}{2} B^2 \hat{f}^2 (e_m (t)) + \kappa^* \frac{e_m^2 (t)}{e_m^2 (t) + \eta} + \left( \frac{1}{2} + \frac{1}{2} \kappa^* + \frac{1}{2} \kappa^2 + S \right) \frac{e_m^2 (t)}{e_m^2 (t) + \eta} \leq Fe_m^2 (t) + S \tag{33}$$

where $F = A^* + \frac{1}{2} (B^2 + 1 + 3 \kappa^* + \kappa^2) + S$. $S$ is selected to make $F < 0$. Therefore, it can be concluded that $\dot{J} (t) < 0$. Since $J (t) > 0$, it follows from second method of Lyapunov that $e_m (t) \to \infty$ as $t \to \infty$.

This completes the proof.

According to the results of Theorem 1, since $e_m (t) \to \infty$ as $t \to \infty$, $\hat{A} (t)$, $\hat{B} (t)$, $\hat{A}_u (t)$, and $\hat{B}_u (t)$ all tend to be constant matrices, which are denoted as $A$, $B$, $A_u$, and $B_u$, respectively.

Consequently, the driven-data model of the wind power system can be obtained as

$$\hat{\omega} (t) = A \omega (t) + B f (\omega (t)) + A_u v (t) + B_u + g u (t) \tag{34}$$
The design of the adaptive controller will be based on the data-driven model (34).

IV. DESIGN OF ADAPTIVE CONTROLLER AND ROBUST TERM

A. DESIGN AND ANALYSIS

According to the dynamic characteristics of the wind power generation system, when the wind speed is under the rated wind speed, to obtain $\gamma_{\text{max}}$, $\beta$ can be fixed at 0°. $\omega$ can be adjusted by controlling $T_e$, so that it can track to the optimal angular velocity of the wind rotor $\omega_{\text{opt}}$ to achieve maximum power point tracking (MPPT) control. $\omega_{\text{opt}}$ can be regarded as the desired angular velocity of the wind rotor $\omega_d$ in MPPT control, which defined as

$$
\omega_{\text{opt}} = \frac{\lambda_{\text{opt}} V}{R}
$$

Combining (34) and (35), assume that the desired trajectory of $\omega_d$ (t) has the following form:

$$
\dot{\omega}_d (t) = A \omega_d(t) + B f (\omega_d(t)) + A u(t) + B u_d(t)
$$

where $u_d(t)$ is the desired control input, which is $T_e$.

By subtracting (34) and (36), we can get the dynamic equation of the error system, which can be formulated as

$$
\dot{e}(t) = A e(t) + B f_e(t) + g u_e(t)
$$

where $e(t) = \omega(t) - \omega_d(t)$, $f_e(t) = f(\omega(t)) - f(\omega_d(t))$, $u_e(t) = u(t) - u_d(t)$. It can be seen that the controller $u(t)$ is composed of two parts, namely, the steady state controller $u_c(t)$ and the error feedback controller $u_e(t)$.

The objective of the optimal feedback controller design is to find the optimal manner in the instantaneous state and stabilize the state tracking error dynamics. In the following, for the convenience of description, $e(t)$, $u_d(t)$, $u_c(t)$, $u_e(t)$, $f_e(t)$ can be abbreviated as $e$, $u_d$, $u_c$, $u$, and $f_e$, respectively.

According to the optimal control theory and (10), the infinite horizon performance function of the wind power generation system is transformed into

$$
V(e) = \int_{t_0}^{\infty} r(e(t), u_e(e(t))) dt
$$

where $r(e, u_c) = Q e^2 + R_c u_c^2$ is the utility function, $R_c$ and $Q$ are positive definite constants. The optimal feedback controller is to find the optimal feedback admissible control $u^*_e$, so that stabilize the state tracking error dynamics of the rotor angular velocity and make the performance function $V(e)$ to obtain a minimum [29]. Then, define the Hamiltonian function as

$$
H(e, u_c, V_e) = V_e(e + B f_e + g u_e) + Q e^2 + R_c u_c^2
$$

where $V_e = \frac{\partial V}{\partial e}$. $u^*_e$ needs to satisfy $\frac{\partial H(e, u_c, V_e)}{\partial u_c} = 0$, we can obtain $u^*_e$ as

$$
u^*_e = -\frac{1}{2} R_c^{-1} g V_e^*
$$

where $V^*_e = \frac{\partial V^*(e)}{\partial e}$, $V^*(e)$ is the minimum value of $V(e)$.

The steady state controller $u_d$ can be obtained by solving (36). In summary, the complete optimal control input is $u^* = u_d + u^*_c$.

Based on the above analysis, it applies the ADP Algorithm to the design of an optimal feedback controller. The specific data flow of the controller using the actor-critic structure is shown in Fig. 5. The action network approximates $u^*_c$ and the critic network approximates the performance index $V(e)$. The main concrete steps are as follows:

1) Calculate the optimal angular velocity of the wind rotor $\omega_{\text{opt}}(t)$, which is $\omega_d(t)$. Then subtract $\omega_d(t)$ and $\omega_{\text{opt}}(t)$ to obtain the error $e(t)$;

2) The action network takes $e(t)$ as input, and calculates $u_c(t)$ as output of the action network;

3) The critic network takes $e(t)$ as input, and calculates the performance function value $V(e)$ and the derivative $V_e$ with respect to $e$;

4) The critic network uses $r(e, u_c)$ and $u_c(t)$ for learning and training, and iteratively updates the weight parameters of the critic network, $V(e)$, and $V_e$;

5) When the critic network updating, the action network uses the updated $V_e$ obtained in step 4 for learning and training, and iteratively updates the weight parameters of the action network and $u_c(t)$;

6) When the objective function value of the action network and the critic network is less than the threshold value or the iteration number reaches the maximum value, the final $u_c(t)$ is output. The steady-state control input $u_d(t)$ and robust compensation term $u_r(t)$ are added to obtain the overall control input $u(t)$. Repeat Step 1 to 6.

The optimal feedback controller design using the ADP method, which is implemented by employing the critic network and action network. The design of the neural network is given in part B and part C.

B. CRITIC NETWORK DESIGN

In the design of the critic network, we use a neural network to approximate the performance index $V(e)$, which is defined
as follows:

$$V(e) = W_e^T \vartheta_e(e) + e_c$$  \hspace{1cm} (41)$$

where $W_e$ is the unknown ideal constant weights of the critic network and $\vartheta_e(e) : \mathbb{R} \rightarrow \mathbb{R}^{N_e}$ is called the critic network activation function vector, $N_e$ is the number of neurons in the hidden layer, and $e_c$ is the critic network bounded approximation error.

The derivative of the performance index function $V(e)$ with respect to $e$ is

$$V_e = \nabla \vartheta_e^T W_e + \nabla e_c$$  \hspace{1cm} (42)$$

where $\nabla \vartheta_e \triangleq \partial \vartheta_e(e)/\partial e$ and $\nabla e_c \triangleq \partial e_c/\partial e$.

Define the estimate of $W_e$ as $\hat{W}_e$, then we get the estimate of $V(e)$ as

$$\hat{V}(e) = \hat{W}_e^T \vartheta_e(e)$$  \hspace{1cm} (43)$$

Combining (39), the approximate Hamiltonian function can be derived as follow:

$$H(e, u_e, \hat{W}_e) = \hat{W}_e^T \nabla \vartheta_e(e) + Q e^2 + R_c u_e^2$$  \hspace{1cm} (44)$$

Define the objective function of the critic network as

$$E_c(\hat{W}_e) = \frac{1}{2} e_c^2$$  \hspace{1cm} (45)$$

The weight update law for the critic network is a gradient descent algorithm, and the update rule is defined as:

$$\dot{\hat{W}}_e = -l_c \sigma_c (\sigma_e^T \hat{W}_e + Q e^2 + R_c u_e^2)$$  \hspace{1cm} (46)$$

where $l_c > 0$ is the adaptive learning rate of the critic network, $\sigma_c = \frac{\sigma}{\sigma_M + 1}$, $\sigma = \nabla \vartheta_e(e + Bf_e + g u_e)$. From the definition of $\sigma_c$, we know that there exists a positive constant $\sigma_M > 1$ to make $\| \sigma_c \| \leq \sigma_M$.

For a control policy $u_e$ that satisfies $\frac{\partial H(e, u_e, V_e)}{\partial u_e} = 0$, the Hamiltonian function (44) can be rewritten as

$$H(e, u_e, W_e) = W_e^T \nabla \vartheta_e(e + Bf_e + g u_e) + Q e^2 + R_c u_e^2$$  \hspace{1cm} (47)$$

where $\epsilon_{HJB}$ the residual error due to the neural network approximation.

Define the weight estimation error of critic network to be $\hat{W}_e = W_e - W_e$. Rewriting (46) by using (47), we have

$$\dot{\hat{W}}_e = -l_c \sigma_c (\sigma_e^T \hat{W}_e + \epsilon_{HJB})$$  \hspace{1cm} (48)$$

C. ACTION NETWORK DESIGN

Similar to the design of the critic network, the action network is used to approximate $u_e$, which is defined as follows:

$$u_e = W_a^T \vartheta_a(e) + e_a$$  \hspace{1cm} (49)$$

where $W_a$ is the unknown ideal constant weights of the action network and $\vartheta_a(e) : \mathbb{R} \rightarrow \mathbb{R}^{N_a}$ is called the action network activation function vector, $N_a$ is the number of neurons in the hidden layer, and $e_a$ is the action network bounded approximation error.

Define the estimate of $W_a$ as $\hat{W}_a$, the actual output can be expressed as

$$\hat{u}_e = \hat{W}_a^T \vartheta_a(e)$$  \hspace{1cm} (50)$$

The feedback error signal used for tuning action NN is defined to be the difference between the optimal feedback control input $u_e^*$ and the actual output of the action network $\hat{u}_e$, and combined with control input minimizing (43) as

$$e_a = \hat{W}_a^T \vartheta_a + \frac{1}{2} R_c^{-1} g \nabla \vartheta_a^T \hat{W}_e$$  \hspace{1cm} (51)$$

Define the objective function of the action network as

$$E_a(\hat{W}_a) = \frac{1}{2} e_a^2$$  \hspace{1cm} (52)$$

The ultimate objective of the action network is to make $E_a(\hat{W}_a)$ as zero, on which the weight update rules of the action network are defined. The weight update law for the action network is a gradient descent algorithm, which is given by

$$\dot{\hat{W}}_a = -l_a \vartheta_a \left( \hat{W}_a^T \vartheta_a + \frac{1}{2} R_c^{-1} g \nabla \vartheta_a^T \hat{W}_e \right)$$  \hspace{1cm} (53)$$

where $l_a > 0$ is the adaptive learning rate of the action network.

Combining (40), (41), and (49), we can have

$$W_a^T \vartheta_a(e) + e_a + \frac{1}{2} R_c^{-1} g \nabla \vartheta_a^T \hat{W}_e + \frac{1}{2} R_c^{-1} g \nabla e_a = 0$$  \hspace{1cm} (54)$$

Define the weight estimation error of critic network to be $\hat{W}_a = \hat{W}_a - W_a$. Combining (53) with (54), we have

$$\dot{\hat{W}}_a = -l_a \vartheta_a \left( \hat{W}_a^T \vartheta_a + \frac{1}{2} R_c^{-1} g \nabla \vartheta_a^T \hat{W}_e + e_{ca} \right)$$  \hspace{1cm} (55)$$

where $e_{ca} = -(e_a + \frac{1}{2} R_c^{-1} g \nabla e_a)/2$.

It is important to note that to ensure $\hat{W}_a$ and $\hat{W}_e$ converge to $W_a$ and $W_e$ respectively, the tracking error must be persistently excited sufficiently under the persistent excitation condition [29]. Further, the persistent excitation condition ensures $\| \sigma_c \| \geq \sigma_{cm}$ and $\| \vartheta_a \| \geq \vartheta_{am}$, with $\sigma_{cm}$ and $\vartheta_{am}$ being positive constants.

D. CONTROLLER STABILITY ANALYSIS

After the design and analysis of the controller, the final overall control input is written as

$$u = u_d + \hat{u}_e$$  \hspace{1cm} (56)$$
Substituting (50) into (37), (37) is rewritten as

\[ \dot{e}(t) = Ae(t) + Bf_e(t) + g\hat{W}_a^T \hat{\phi}_a \]  

(57)

Combining with (49), subtracting and adding \( W_a^T \hat{\phi}_a(e) \) to (57), (57) is rewritten as

\[ \dot{e}(t) = Ae(t) + Bf_e(t) + g\hat{W}_a^T \hat{\phi}_a + gu_e - g\varepsilon_a \]  

(58)

In the following, the stability analysis is performed. First, make the following assumptions [28], [30]:

**Assumption 2:**

1. \( W_c \) and \( W_a \) are upper bounded so that \( \| W_c \| \leq W_{cM} \) and \( \| W_a \| \leq W_{aM} \), respectively;
2. \( \varepsilon_c \) and \( \varepsilon_a \) are upper bounded so that \( \| \varepsilon_c \| \leq \varepsilon_{cM} \) and \( \| \varepsilon_a \| \leq \varepsilon_{aM} \), respectively;
3. \( \vartheta_l(\cdot) \) and \( \vartheta_a(\cdot) \) are upper bounded so that \( \| \vartheta_l(\cdot) \| \leq \vartheta_{lM} \) and \( \| \vartheta_a(\cdot) \| \leq \vartheta_{aM} \), respectively;
4. The gradients of the critic network approximation error and the activation function vector are upper bounded so that \( \| V\varepsilon_c \| \leq \varepsilon_{cM} \) and \( \| V\vartheta_l \| \leq \vartheta_{LM} \), respectively; And the residual error is upper bounded so that \( \| \varepsilon_{HJB} \| \leq \varepsilon_{HJB} \).

Based on the current problem settings and controller design, the above assumptions can be reasonably satisfied. After making assumptions, we can start to prove stability. Choose the Lyapunov function as follows:

\[ L(t) = \frac{1}{2l_c} \text{tr} \left[ \dot{\hat{W}}_c^T \hat{W}_c \right] + \frac{l_c}{2l_a} \text{tr} \left[ \dot{W}_a^T \hat{W}_a \right] + V(e) \]  

(59)

According to Assumptions 1 and 2, combine with (38), (48) and (55), the time derivative of \( L(t) \) along the trajectory of the error system (58) is calculated as

\[ \dot{L}(t) = \frac{1}{l_c} \text{tr} \left[ \dot{W}_c^T \dot{\hat{W}}_c \right] + \frac{l_c}{l_a} \text{tr} \left[ \dot{W}_a^T \dot{\hat{W}}_a \right] + V_e \]

\[ = \frac{1}{l_c} \text{tr} \left[ \dot{W}_c^T \left[ -l_c \varepsilon_c^2 (\sigma_c^T \hat{W}_c + \varepsilon_{HJB}) \right] \right] \]

\[ + \frac{l_c}{l_a} \text{tr} \left[ \dot{W}_a^T \left[ -l_a \vartheta_a^2 \left( \hat{W}_a^T \vartheta_a \right) + \frac{1}{2} R_c^{-1} g \nabla \varepsilon_c^2 \hat{W}_c + ec_a \right] \right] \]

\[ - Qe^2 - R_c u_e^2 \]

\[ \leq - \left( l_c^2 \vartheta_{aM}^2 - \frac{3}{4} l_a \vartheta_{aM}^2 \right) \| \hat{W}_a \|^2 \]

\[ - Qe^2 - R_c u_e^2 - \left( \frac{1}{2} \| \varepsilon_{HJB} \| \right) \| \hat{W}_c \|^2 \]

\[ + \frac{1}{2l_c} \varepsilon_{HJB}^2 + \frac{l_c}{2l_a} \varepsilon_{ca}^2 \]  

(60)

Then according to Assumptions and (61), we have

\[ \frac{1}{2l_c} \varepsilon_{HJB}^2 + \frac{l_c}{2l_a} \varepsilon_{ca}^2 \leq D_M \]  

(62)

If \( l_c \), \( l_a \) satisfy the following conditions:

\[ l_c < \frac{4l_a \sigma_{cm}^2}{2l_a \sigma_{cm}^2 + R_c^{-1} \| g \varepsilon_{dM}^2 \|} \]  

(63)

\[ l_a < \frac{4l_a \sigma_{am}^2}{3l_a \sigma_{am}^2} \]  

(64)

and given that the inequalities

\[ \| e \| \leq \sqrt{D_M Q} \leq b_e \]  

(65)

or

\[ \| \hat{W}_c \| > \frac{D_M}{Q_1} \leq b_{\hat{W}_c} \]  

(66)

or

\[ \| \hat{W}_a \| > \frac{D_M}{Q_1} \leq b_{\hat{W}_a} \]  

(67)

hold, then \( \dot{L}(t) < 0 \). According to Lyapunov theory [31], the tracking error \( e \) and the weight estimation errors \( e_c \) and \( e_a \) of the neural network are uniformly ultimately bounded (UUB). Also, it is possible to prove \( \| u - u^* \| \leq e_u \), where \( \| u - u^* \| \leq e_a \).

**E. ADDITIONAL ROBUSTIFYING TERM DESIGN**

Due to the approximation errors \( \varepsilon_c \) and \( \varepsilon_a \) of the neural network, the tracking error does not gradually converge to zero. Therefore, for improving the tracking performance, we add an additional robustifying term on the basis of the above controller design such that tracking error converges to zero asymptotically.

According to (60), the additional robust compensation term can be constructed in the form

\[ u_r = \frac{K_r e}{e^2 + \theta} \]  

(68)

where \( \theta \) is a positive constant, \( K_r > K_{rmin} \) is a designed parameter. The value of \( K_{rmin} \) satisfies the following inequality:

\[ K_{rmin} \geq \frac{D_M \left( e^2 + \theta \right)}{2l_a l_c g e^2} \]  

(69)

Then combining (56), the overall control input is given as

\[ u_{ad} = u_d + \hat{u}_e - u_r \]  

(70)

Applying (70) into (34) and using (35), the new error system is obtained as

\[ \dot{e}(t) = Ae(t) + Bf_e(t) + g\hat{W}_a^T \hat{\phi}_a + gu_e - g\varepsilon_a - gu_r \]  

(71)
The stability proof of the controller with robust terms is similar to the stability proof in section III. Choose the same Lyapunov function, according to (68) and (69), we can get

$$\dot{L}(t) \leq -\left(\frac{3}{4}l_{2a}^{2} - \frac{3}{4}l_{4a}\right)\left\|\tilde{W}_{a}\right\|^{2} - Qe^{2} - R_{c}\dot{u}_{e}^{2} - \left(\sigma_{cm} - \frac{l_{c}}{2}\sigma_{cm}\right)\left\|\tilde{W}_{c}\right\|^{2}$$

(72)

Equation (72) and (60) select the same parameters, we have \(\dot{L}(t) \leq 0\). Because the variables on the right-hand of (71) are bounded, \(\dot{e}\) is also bounded. From (72), we have

$$\dot{L}(t) \leq -Qe^{2}$$

(73)

Integrating both sides of (73) and then organize according to the mean value theorem, we have

$$\int_{0}^{\infty} e^{2}dt \leq Q^{-1}(L(0) - L(\infty))$$

(74)

since the right side of (71) is bounded, \(\|e\| \in L_{2}\). According to Barbalat’s lemma [31], we have \(\lim_{t \to \infty} \|e\| = 0\). Similarly, we can prove that \(\lim_{t \to \infty} \|\tilde{W}_{c}\| = 0\), \(\lim_{t \to \infty} \|\tilde{W}_{a}\| = 0\), and \(\|u_{ad} - u^{e}\| \leq \varepsilon_{aM}\).

V. SIMULATION

A. ENVIRONMENT AND PARAMETERS

The simulation environment of the experiment is shown in Table 1.

| TABLE 1. Experimental environment. |
|-----------------------------------|
| Hardware | CPU | Intel(R) Core(TM) i7-7700 @3.60GHz |
| Memory | 8GB RAM |
| Software | Operating System | Window 10 |
| Simulation Software | MATLAB 2018a |

We simulate the wind power generation system in MATLAB and record the system state data and control data during the simulation [32]. After that, the data is pre-processed, and use the pre-processed data as input-output data for the data-driven model learning. Finally, we get the completed data-driven model of the wind power generation system. This paper uses the parameters in Table 2 to simulate the wind power generation system. Based on the data model, we use the tip speed ratio method as the MPPT control strategy for wind power generation systems. Then, carry on a simulation experiment to the DRAC control method.

B. EXPERIMENT

The simulation uses the four-component combination wind speed mathematical model to simulate the actual wind speed change signal [33], [34], and it shows its related parameters in Table 3.

We use the parameter values in Table 3 to simulate the wind speed of the wind farm, and the variation curve of the wind speed simulation signal is shown in Fig. 6. It considers various conditions such as basic wind, gust, gradient wind, and random wind, and it conforms to the distribution law of actual wind speed.

| TABLE 2. Parameters for wind power generation system. |
|-------------------------------------------------------|
| Symbol | Quantity     | Value  |
|--------|--------------|--------|
| R      | blade radius | 40.25 m |
| \(\rho\) | air density of wind farm | 1.25 kg/m³ |
| \(V_{w}\) | rated wind speed | 12.5 m/s |
| \(J_{r}\) | inertia of the rotor | 4.9 \times 10^{6} kg \cdot m² |
| \(J_{g}\) | inertia of the generator | 107.87 kg \cdot m² |
| N      | transmission ratio | 104 |
| \(T_{e}\) | rated electromagnetic torque of generator | 9000 N \cdot m |
| \(\lambda_{opt}\) | optimum tip speed ratio | 8.1 |
| \(\omega_{rate}\) | rated angular velocity of the wind rotor | 2.52 r/s |

| TABLE 3. Parameters for wind speed model. |
|------------------------------------------|
| Symbol | Value     | Symbol | Value |
|-------|-----------|--------|--------|
| \(v_{0}\) | 7 m/s     | \(T_{1g}\) | 13s |
| \(G_{max}\) | 6 m/s     | \(T_{2g}\) | 62s |
| \(R_{max}\) | 3 m/s     | \(T_{1r}\) | 30s |
| \(\nu_{max}\) | 1 m/s     | \(T_{2r}\) | 130s |
| \(\tau\) | 150s      | \(T_{r}\) | 50s |

We simulate the wind power generation system in MATLAB and record the system state data and control data during the simulation [32]. After that, the data is pre-processed, and use the pre-processed data as input-output data for the data-driven model learning. Finally, we get the completed data-driven model of the wind power generation system. This paper uses the parameters in Table 2 to simulate the wind power generation system. Based on the data model, we use the tip speed ratio method as the MPPT control strategy for wind power generation systems. Then, carry on a simulation experiment to the DRAC control method.

B. EXPERIMENT

The simulation uses the four-component combination wind speed mathematical model to simulate the actual wind speed change signal [33], [34], and it shows its related parameters in Table 3.

We use the parameter values in Table 3 to simulate the wind speed of the wind farm, and the variation curve of the wind speed simulation signal is shown in Fig. 6. It considers various conditions such as basic wind, gust, gradient wind, and random wind, and it conforms to the distribution law of actual wind speed.

FIGURE 6. The variation curve of wind speed simulation signal.
defined as

\[ Q_{PSO} = \int_0^\infty t |e(t)| \, dt. \]  (75)

When \( Q_{PSO} \leq 0.5 \), the algorithm stops iterating. Through experimental simulation, the tracking error variation curve of the angular velocity of the rotor of the DRAC in a sinusoidal wind speed environment with an average wind speed of 7 m/s is shown in Fig.7. We can observe from Fig.7 that the tracking error can converge, and the convergence speed is fast. At the same time, it can also be verified that the controller stability proof is correct. The angular velocity tracking error variation curve of DRAC, PID, and PSOPID under the actual combined wind speed environment can be obtained, and the comparison curve is shown in Fig.8.

By observing Fig.7 and 8, we can find that the DRAC control method has a similar error convergence rate in the sinusoidal wind speed and the combined actual wind speed, and it is not more difficult to converge due to the increased randomness and variability of wind speed. It shows that the control effect and tracking reaction speed of this control method has no obvious relationship with the change rule of wind speed, and it has strong robustness. In Fig.8, we can observe that although PSOPID has a smaller variance in the tracking error of the angular velocity of the rotor compared to the traditional PID, the error is still larger than that of DRAC, and the tracking error of DRAC is much smaller than PID and PSOPID.

The maximum power tracking curves of different algorithms are shown in Fig.9. To observe the curve more clearly, Fig.9 is partially enlarged, as shown in Fig.10. From Fig.9 and Fig.10, we can observe that the power variation curve of the DRAC control method almost coincides with the maximum power curve, which proves that the tracking effect is the best. At the turning point where the wind speed changes suddenly, the maximum power tracking error of DRAC is small, indicating that the control response speed is fast. Compared with DRAC, the tracking effect of PSOPID is poor, especially at the turning point, the control response speed is slow, but the tracking effect is better than the traditional PID method.
TABLE 5. The name of all quantities used in article.

| Symbol | Quantity | Symbol | Quantity |
|--------|----------|--------|----------|
| $P_m$  | active power output of wind turbine | $A^*, B^*, A_u^*, B_u^*$ | ideal weights of the data-driven model |
| $P_{opt}$ | the maximum power of wind turbine | $A, B, A_u, B_u$ | constants of the data-driven model |
| $C_{p_{max}}$ | the maximum of energy utilization coefficient | $\hat{A}, \hat{B}, \hat{A}_u, \hat{B}_u$ | estimate of $A^*, B^*, A_u^*, B_u^*$ |
| $C_p$ | wind energy utilization coefficient | $g, \eta$ | constant |
| $\nu$ | wind speed | $f(\cdot)$ | activation function |
| $\rho$ | air density of wind farm | $e_m$ | modeling error of $\omega$ |
| $R$ | blade radius | $\kappa(t) \in \mathbb{R}$ | tunable parameter |
| $\lambda$ | tip speed ratio | $\kappa^*$ | bounded constant |
| $\lambda_{opt}$ | optimal tip speed ratio | $J$ | Lyapunov function |
| $\omega$ | angular velocity of wind rotor | $\varepsilon$ | approximation error term |
| $\omega_d$ | desired angular velocity of wind rotor | $\gamma_i$ | positive number, $i = 1,2,3,4,5$ |
| $\beta$ | pitch angle | $J_g$ | inertia of the generator |
| $J_r$ | inertia of the rotor | $N$ | transmission ratio |
| $\varnothing_c$ | critic network activation function vector | $W_c$ | constant weights of the critic network |
| $\varnothing_a$ | action network activation function vector | $W_a$ | constant weights of the action network |
| $\varepsilon_c$ | critic network bounded approximation error | $\varepsilon_{HJB}$ | residual error |
| $\varepsilon_a$ | action network bounded approximation error | $T_e$ | electromagnetic torque of generator |
| $l_c$ | adaptive learning rate of the critic network. | $T_a$ | torque of pneumatic machinery |
| $l_a$ | adaptive learning rate of the action network. | $t$ | simulation time |
| $x$ | system state | $\mathcal{T}$ | simulation cycle |
| $V$ | system performance function | $u_d$ | steady state control |
| $V^*$ | minimum value of $V$ | $u_e$ | error feedback control |
| $\dot{V}^*$ | derivative of $V^*$ with respect to $t$ | $r$ | utility function related to $x$ |
| $\nabla V^*$ | derivative of $V^*$ with respect to $x$ | $H$ | Hamiltonian function |
| $u$ | overall control | $R_c, Q$ | positive definite constants |

Through the above analysis, we can know that all three methods can effectively track the maximum power, but the DRAC control method proposed in this paper has the smallest error and fast-tracking response speed, is relatively more stable, and can track the maximum power better.

With the rapid development of computer performance, the computing power of computers is becoming more and more powerful. The algorithmic complexity of the control algorithm is no longer the primary consideration, but the control performance of the algorithm is the primary concern. Therefore, by using the aerodynamic efficiency performance index $\xi_{aero}$ to compare the performance of the DRAC method with other control methods [14]. $\xi_{aero}$ reflects the effect of tracking the maximum power, and its expression is shown in (76).

\[
\xi_{aero} = \frac{\int_{t_i}^{t_f} P_{opt} dt}{\int_{t_i}^{t_f} P_m dt} \tag{76}
\]

where $P_{opt} = \frac{1}{2} \rho \pi R^2 v^3 C_{p_{\text{max}}}$.
The $\xi_{\text{zero}}$ of the DRAC method, PID method, PSOPID method, NSSFE method [14], NDSFE method [14], and NNC method [36] are shown in Table 4.

Comparing the data in Table 4, it can be observed that the maximum power tracking effect of DRAC is improved by nearly 13% compared with PID, and it is improved by nearly 1.5% compared with PSOPID, which is more than 10% better than NSSFE. The tracking effect of DRAC is compared with NDSFE also increased by about 7%. Although DRAC and NNC have similar effects, their $\xi_{\text{zero}}$ values are close to 100%, which indicates that their wind energy capture optimization efficiency is high, which is conducive to the improvement of wind energy utilization and can effectively track the maximum power.

In this section, it can draw the following conclusions through simulation experiments: The control method proposed in this paper can achieve the desired effect in an environment with random variation of wind speed, has strong robustness, and can better track the maximum power.

VI. CONCLUSION

In this paper, based on the nonlinear equations of wind power generation system and adaptive dynamic programming algorithm, combined with a recurrent neural network, a data-driven robust adaptive control (DRAC) method for maximum power point tracking of wind power generation system is proposed. Simulation experiment results show that the DRAC has a better tracking effect and improves the efficiency of wind energy conversion. At the same time, because it is driven by a data model, this method only requires the input-output data of the wind power generation system, and the requirements for internal dynamics knowledge are low.

However, the DRAC has limitations and is currently only applicable to wind speeds below the rated wind speed. In future research, we will try to combine the latest machine learning technology and propose an MPPT control method that can be applied above the rated wind speed. Further to solve the shortcomings of this method, which requires a sizeable amount of computation and a long time to train the neural network. We will consider more wind power systems and integrate machine learning with wind power systems to improve control performance.

APPENDIX

See Table 5.

REFERENCES

[1] N. Baiomy and R. Kikuuwe, “An amplitude- and rate-saturated collective pitch controller for wind turbine systems,” Renew. Energy, vol. 158, pp. 400–409, Oct. 2020.
[2] G. Mu, T. T. Cai, and G. Yan, “Bidirectional power constraints and influence of doubly fed induction generator participating in continuous frequency regulation,” Trans. China Electrotech Soc., vol. 34, no. 8, pp. 1750–1759, Mar. 2019.
[3] Z. P. Lu and H. Wang, “Research on MPPT strategy of SRG wind power generation based on BP algorithm,” (in Chinese), Mach. Building Autom., vol. 48, no. 6, pp. 217–220, 2019.
[4] Z. Civelek, M. Lüy, E. Çam, and H. Mamur, “A new fuzzy logic proportional controller approach applied to individual pitch angle for wind turbine load mitigation,” Renew. Energy, vol. 111, pp. 708–717, Oct. 2017.
[5] A. Medjber, A. Guessoum, H. Belmili, and A. Mellit, “New neural network and fuzzy logic controllers to monitor maximum power for wind energy conversion system,” Energy, vol. 106, pp. 137–146, Jul. 2016.
[6] J. Yang, W. X. Zheng, S. Li, B. Wu, and M. Cheng, “Design of a prediction-accuracy-enhanced continuous-time MPC for disturbed systems via a disturbance observer,” IEEE Trans. Ind. Electron., vol. 62, no. 9, pp. 5807–5816, Sep. 2015.
[7] X. Wang and H. Su, “Completely model-free RL-based consensus of continuous-time multi-agent systems,” Appl. Math. Comput., vol. 382, Oct. 2020, Art. no. 125312.
[8] A. Gani, K. Mohammadi, S. Shamsirband, A. T. Altameem, D. Petkovic, and C. Sudheer, “A combined method to estimate wind speed distribution based on integrating the support vector machine with firefly algorithm,” Environ. Prog. Sustain Energy, vol. 35, no. 3, pp. 867–875, Oct. 2015.
[9] R. M. Sanner and J.-J. E. Slotine, “Gaussian networks for direct adaptive control,” IEEE Trans. Neural Netw., vol. 3, no. 6, pp. 837–863, Nov. 1992.
[10] K. S. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” IEEE Trans. Neural Netw., vol. 1, no. 1, pp. 4–27, Mar. 1990.
[11] F. Jiang, F. Pourpanah, and Q. Hao, “Design, implementation, and evaluation of a neural-network-based quadcopter UAV system,” IEEE Trans. Ind. Electron., vol. 67, no. 3, pp. 2076–2085, Mar. 2020.
[12] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, “Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers,” IEEE Circuits Syst. Mag., vol. 32, no. 6, pp. 76–105, Dec. 2012.
[13] F. Bianchi, H. Battista, and R. Mantz, Wind Turbine Control Systems Principle. Modelling and Gain Scheduling Desing. Cham, Switzerland: Springer, 2007.
[14] B. Boukhzezzar and H. Siguerdijian, “Nonlinear control of a variable-speed wind turbine using a two-mass model,” IEEE Trans. Energy Convers., vol. 26, no. 1, pp. 149–162, Mar. 2011.
[15] J. Hui and A. Bakhshi, “A new adaptive control algorithm for maximum power point tracking for wind energy conversion systems,” in Proc. IEEE Power Electron. Spec. Conf., Jun. 2008, pp. 4003–4007.
[16] A. R. Youssef, H. H. Moussa, and E. E. M. Mohamed, “Development of self-adaptive P&O MPPT algorithm for wind generation systems with concentrated search area,” Renew. Energy, vol. 154, pp. 875–893, Jul. 2020.
[17] X. Tang, M. Yin, C. Shen, Y. Xu, Z. Y. Dong, and Y. Zou, “Active power control of wind turbine generators via coordinated rotor speed and pitch angle regulation,” IEEE Trans. Sustain. Energy, vol. 10, no. 2, pp. 822–832, Apr. 2019.
[18] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, vol. 1. Cambridge, MA, USA: MIT Press, 2016, pp. 367–415.
[19] X. P. Qiu (2019). Neural Network and Deep Learning. Github Inc. [Online]. Available: https://github.com/mlnl44/ml44.github.io
[20] D. A. White and D. A. Sofge, Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches. New York, NY, USA: Van Nostrand Reinhold, 1992.
[21] P. J. Werbos, “Approximate dynamic programming for real-time control and neural modeling,” in Handbook of Intelligent Control, D. A. White and D. A. Sofge, Eds. New York, NY, USA: Van Nostrand Reinhold, 1992. 2008.
[22] R. E. Bellman, Dynamic Programming, Princeton, NJ, USA: Princeton Univ. Press, 1957.
[23] S. E. Dreyfus and A. M. Law, The Art and Theory of Dynamic Programming, New York, NY, USA: Academic, 1977.
[24] Q.-L. Wei, H.-G. Zhang, D.-R. Liu, and Y. Zhao, “An optimal control scheme for a class of discrete-time nonlinear systems with time delays using adaptive dynamic programming,” Acta Automatica Sinica, vol. 36, no. 1, pp. 121–129, Jan. 2010.
[25] H.-G. Zhang, X. Zhang, Y.-H. Luo, and J. Yang, “An overview of research on adaptive dynamic programming,” Acta Automatica Sinica, vol. 39, no. 4, pp. 303–311, Apr. 2013.
[26] J. D. J. Rubio and W. Yu, “Stability analysis of nonlinear system identification via delayed neural networks,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 2, pp. 161–165, Feb. 2007.
[27] T. Hayakawa, W. M. Haddad, and N. Hovakimyan, “Neural network adaptive control for a class of nonlinear uncertain dynamical systems with asymptotic stability guarantees,” IEEE Trans. Neural Netw., vol. 19, no. 1, pp. 80–89, Jan. 2008.
[28] H. Zhang, L. Cui, X. Zhang, and Y. Luo, “Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method,” IEEE Trans. Neural Netw., vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[29] M. Abu-Khalaf and F. L. Lewis, “Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach,” Automatica, vol. 41, no. 5, pp. 779–791, May 2005.

[30] K. G. Vamvoudakis and F. L. Lewis, “Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem,” Automatica, vol. 46, no. 5, pp. 878–888, May 2010.

[31] H. K. Khalil, Nonlinear System. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

[32] G. Mademlis, Y. Liu, P. Chen, and E. Singhroy, “Design of maximum power point tracking for dynamic power response of tidal undersea kite systems,” IEEE Trans. Ind. Appl., vol. 56, no. 2, pp. 2048–2060, Mar. 2020.

[33] Z. Gao, “The modeling of wind speed based on MATLAB,” Int. J. Sci. Res., vol. 5, no. 3, pp. 2319–7064, Mar. 2016.

[34] C. Wu and M. Lu, “The modeling and simulation for wind model of wind power,” in Proc. Int. Conf. Electr. Control Eng., Yichang, China, Sep. 2011, pp. 2715–2717.

[35] R. Arora, V. K. Tayal, H. P. Singh, and S. Singh, “PSO optimized PID controller design for performance enhancement of hybrid renewable energy system,” in Proc. IEEE 9th Power India Int. Conf. (PIICON), Sonepat, India, Feb. 2020, pp. 1–5.

[36] H. Jafarnejadsani, J. Pieper, and J. Ehlers, “Adaptive control of a variable-speed variable-pitch wind turbine using radial-basis function neural network,” IEEE Trans. Control Syst. Technol., vol. 21, no. 6, pp. 2264–2272, Nov. 2013.

PENG CHEN is currently pursuing the M.S. degree with the School of Information Engineering, Shanghai Maritime University, Shanghai, China. Her current research interests include reinforcement learning and deep learning.

DEZHI HAN (Member, IEEE) received the B.S. degree in applied physics from the Hefei University of Technology, China, in 1990, and the Ph.D. degree in computing science from the Huazhong University of Science and Technology, China, in 2005. He is currently a Professor with the Department of Computer, Shanghai Maritime University, China, in 2010. His research interests include reinforcement learning and deep learning, wireless communication security, and power control systems.

KUAN-CHING LI (Senior Member, IEEE) is currently a Professor with Providence University, Taiwan. He has been actively involved in many major conferences and workshops in program/general/steering conference chairman positions. He has organized numerous conferences related to high-performance computing and computational science and engineering. He is the coauthor/co-editor of several technical professional books published by CRC Press, Springer, McGraw-Hill, and IGI Global. His research interests include GPU/manycore computing, big data, and cloud. He is a member of AAAS and the Program Committee. He is also a Fellow of IET. He was a recipient of awards and funding support from several agencies and industrial companies. He also received distinguished and chair professorships from universities in China and other countries.