IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 25, 2019, accepted January 19, 2020, date of publication January 23, 2020, date of current version January 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968853

Reinforcement-Based Robust Variable Pitch Control of Wind Turbines

PENG CHEN[1], DEZHI HAN[1], (Member, IEEE), FUXIAO TAN[1], (Member, IEEE), AND JUN WANG[2], (Senior Member, IEEE)

[1]Department of Computer Science and Technology, Shanghai Maritime University, Shanghai 200120, China
[2]Department of ECE, University of Central Florida, Orlando, FL 32816-2362, USA

Corresponding author: Dezhi Han (dzhan@shmtu.edu.cn)

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

ABSTRACT Due to the influence of wind speed disturbance, there are some uncertain phenomena in the parameters of the nonlinear wind turbine model with time in an actual working environment. In order to mitigate the side effects of uncertainties in speed models of wind turbines, researchers have designed a variety of controllers in recent years. However, traditional control methods require more knowledge of dynamics. Therefore, based on reinforcement learning and system state data, a robust wind turbine controller that adopts adaptive dynamic programming (ADP) is proposed. The ADP algorithm is a combination of Temporal-Difference (TD) algorithm and actor-critic structure, which can guarantee the rotor speed is stable around the rated value to indirectly adjust the wind energy utilization coefficient by changing the pitch angle in the area of high wind speed and achieve online learning in real-time. In addition, the variation of the pitch angle command of the proposed controller is relatively gradual, which can reduce the energy consumption of the variable pitch actuator, and extend the service life of the equipment. Finally, the wind speed model is simulated by combined wind speed based on Weibull distribution, the comprehensive simulation results show that the proposed controller has better control effect than some existing ones.

INDEX TERMS Neural dynamic programming, reinforcement learning, robust control, wind turbine system.

I. INTRODUCTION

With the growth of the energy demand in the world, environmental problems are becoming more and more serious; thus, attracting a lot of attention from renewable energies. As a kind of renewable clean energy, wind energy can be applied as an essential energy source into different fields. Wind power is one of the most effective methods to utilize wind energy. However, because of the uncertainties of the environment about the wind farm, and the stochastic change of high fluctuation wind speed and other factors, the control of wind turbines has brought great difficulties [1].

The traditional regulation methods mainly include fixed pitch stall control and variable pitch control. The variable pitch control method adjusts the blade pitch angle according to the change of wind speed to control the stability of the rotor speed of the wind turbines. Traditional variable pitch control inevitably increases pitch servo fatigue and blade stress due to frequent adjustment of pitch angle, thus reducing the service life of wind turbines [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Canbing Li.

At present, most wind turbines adopt Proportion-Integration- Differentiation (PID) or Proportional-Integral (PI) control. The PI control method is simple and easy to implement, but there may be large overshoot. The PID controller with fixed parameters is difficult to ensure the stability of output power. Among them, there is a fuzzy adaptive PID control, which is used to adjust the hydraulically driven variable pitch system [3]. However, the algorithm parameters need to be reset according to the actual situation in the application process, which does not have a perfect generalization. A Proportional-integral-resonance (PI-R) pitch control method based on MBC coordinate transformation is proposed in literature [4]. It can suppress the low frequency and high-frequency components of the unbalanced load, which is easily disturbed by other random frequency components.

Based on the above problems and combined with the analysis of the wind turbine models in [5], [6], this paper designs a pitch angle controller based on reinforcement learning with stable pitch angle change, which stabilizes the rotor speed at the rated value in an environment with higher wind speed than the rated speed. Thereby the power generation stability of the wind turbine is indirectly controlled.

In a high wind speed environment, the rotor speed will increase with the rise of the wind speed, so that the centrifugal force generated on the blade will explode, which will damage the blade. The controller proposed in this paper controls the speed of the rotor to be stable at the rated value, which can avoid rotor over-speed to reduce the damage to the equipment and prolong the service life of the equipment.

Reinforcement learning is a type of machine learning, which is an approach for solving optimization problems. Meanwhile, it is based on real-time evaluation information about the environment [7], [8], so it can be called action-based learning, that is, it converts the action to a predetermined goal by reward or punishment.

One type of reinforcement learning algorithms makes use of the actor-critic structure shown in Fig.1 [9]. Therein, the machine learning mechanism employs the actor-critic structure includes two steps, namely policy assessment, which is run by the critic; and policy improvement, which is performed by the actor. To be specific, the structure shows the policy evaluation step by observing the results of applying current actions from the environment. Besides, using performance indicators or value functions assess the results in [8], and [10], which quantifies the extent to which the current action is close to optimal.

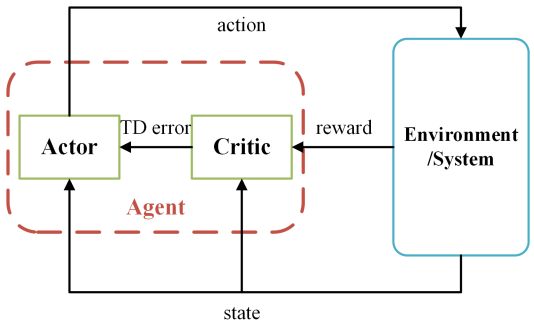

FIGURE 1. Schematic diagram of the actor-critic structure.

Werbos [11] developed actor-critic techniques for feedback control of discrete-time dynamical systems [9]. These methods are known as approximate dynamic programming (ADP) or adaptive dynamic programming. ADP has been widely used in feedback control applications such as aircraft landing control [12], [13], missile control [14], power system control [15], and automotive control [16]. Therefore, it is a significant attempt to apply the ADP method based on reinforcement learning to the design of the feedback controller of the wind turbines.

We can study reinforcement learning to use a framework based on the Markov decision process (MDPs). The basics of MDP and Bellman Equation will be covered in part C of section II.

A. RESEARCH CONTRUBUTIONS

1) A robust wind turbine controller that adopts adaptive dynamic programming (ADP) is proposed. The ADP algorithm is a combination of Temporal-Difference (TD) algorithm and actor-critic structure, it can guarantee the rotor speed is stable around the rated value to indirectly improve the efficiency of use of wind energy by changing the pitch angle in the area of high wind speed and achieve online learning in real-time. In conclusion, the controller can adjust the system to the preset objective so that the speed of wind turbines is stable at the rated speed and the range of pitch angle is smaller.

2) The variation of the pitch angle command of the proposed controller is relatively gradual, which can reduce the energy consumption of the variable pitch actuator, and extend the service life of the equipment.

3) Compared with the previous work, the method proposed in this paper only needs to set the control objective and does not need to know how to reach the objective, for example, it does not need to adjust the control parameters.

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, the design ideas are presented as well as the details of the variable pitch robust controller based on reinforcement learning. In section IV, the simulation experiment is given. Finally, the conclusion and prospects are provided in section V.

II. PROBLEM FORMULATION AND PRELIMINARIES

This section respectively gives the energy transmission model of the wind turbine and the mathematical simulation of wind speed prediction in parts A and B, including the simple derivation of these models. Markov's decision-making process and the Bellman equation are essential knowledge and foundation in reinforcement learning. Due to the better understanding of controller's design principles, the decision-making process and formula will be briefly introduced in part C.

A. WIND TURBINE ENERGY TRANSMISSION MODEL

A simple structure diagram of the wind turbine is showed in Fig. 2. In the energy transmission model of the wind turbine, there is a wind energy utilization coefficient C_p. Assuming that the wind speed is the same at the surface of the wind wheel, C_p can be expressed approximately by the following equation (1) [17].

$$\left.\begin{aligned} C_p &= 0.52\, sin\left(\frac{116}{\Lambda} - 0.4\beta - 5\right) e^{-21/\Lambda} \\ \frac{1}{\Lambda} &= \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} \end{aligned}\right\} \tag{1}$$

where $\boldsymbol{\beta}$ is the pitch angle, $\boldsymbol{\Lambda}$ is the intermediate variable, and $\boldsymbol{\lambda}$ is the tip speed ratio; $\boldsymbol{\lambda}$ can be defined by equation (2).

$$\lambda = \frac{\omega R}{v} \tag{2}$$

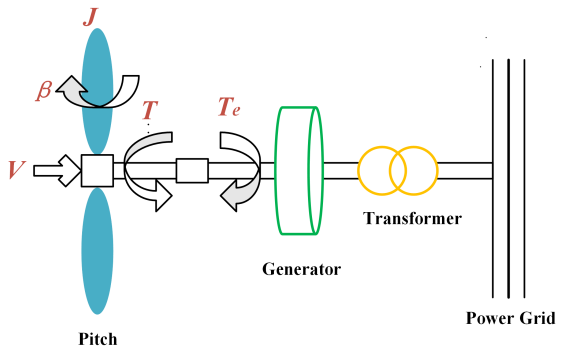

FIGURE 2. The simple structure diagram of the wind turbine.

In equation (2), $\boldsymbol{\omega}$ is the angular velocity of wind rotor rotates, R is the radius of the rotor, and $\boldsymbol{v}$ is the wind speed.

With reference to literature [18], ignoring the transmission damping of the wind turbine and the generator, the simplified motion equation of the wind turbine's transmission system is defined by equation (3).

$$\left(J_r + N^2 J_g\right)\frac{d\omega}{dt} = \frac{1}{2}\rho A R C_T v^2 - N T_e \tag{3}$$

where J_r, J_g, N, ρ, A and T_e denote the inertia of the rotor, inertia of the generator, transmission ratio, air density, sweeping area of the wind turbine and counter-torque of the engine, respectively; C_T is represented by equation (4).

$$C_T = \frac{1}{\lambda}C_p \tag{4}$$

B. WIND SPEED PREDICTION SIMULATION MODEL

To accurately describe the randomness and intermittency of wind speed, four wind speed component combination wind speed mathematical model are adopted in this paper. The four components are basic wind, gust, gradual wind, and random wind, respectively [19].

1) BASIC WIND

The Weibull distribution parameters can approximate the basic wind v_b [20]:

$$v_b = C \cdot \Gamma(1 + \frac{1}{k}) \tag{5}$$

where C, k is the scale parameter and form parameter of Weibull distribution respectively. $\Gamma(\cdot)$ is the gamma function. In practice and simulation, we can approximate that v_b is a component that does not change with time, that is, v_b is taken as a constant.

2) GUST OF WIND

Gust v_g describes a sharp shift of wind speed what causes the wind power ramp. The wind speeds in two time periods is different:

$$v_g = \begin{cases} v_{gcos} & T_{1g} < t < T_{2g} \\ 0 & \text{else} \end{cases} \tag{6}$$

where v_{gcos} denotes:

$$v_{gcos} = \frac{G_{max}}{2}\left[1 - cos2\pi\left(\frac{t - T_{1g}}{T_{1g} - T_{2g}}\right)\right] \tag{7}$$

G_{max} is the maximum gust, T_{1g} is the start time of gust, T_{2g} is the dead time, t is the time in a cycle.

3) GRADIENT WIND

Gradient wind v_r can simulate the gradual change of wind speed:

$$v_r = \begin{cases} 0 & \text{else} \\ R_{max} \cdot \dfrac{t - T_{1r}}{T_{2r} - T_{1r}} & T_{1r} \le t \le T_{2r} \\ R_{max} & T_{2r} < t \le T_{2r} + T_r \end{cases} \tag{8}$$

where R_{max} the maximum gradient wind, T_{1r} is the start time of gradient wind, T_{2r} is the dead time, T_r is the gradual time hold time.

4) RANDOM WIND

Random wind v_n reflects the randomness of wind speed variation, and its model is:

$$v_n = v_{nmax} \cdot R_n(-1, 1) \cdot cos(\omega_v + \varphi_v) \tag{9}$$

where v_{nmax} is the maximum random wind, $R_n(-1, 1)$ is a random number uniformly distributed in $(-1, 1)$, ω_v is the average distance of wind speed fluctuations, φ_v is a uniformly distributed random quantity in $(0, 2\pi)$.

In conclusion, the combined wind speed can be expressed by the following equation:

$$v = v_b + v_g + v_r + v_n \tag{10}$$

which the Specific parameter value is will be given in Section IV.

Considering the limitation of wind speed detection, the time interval of wind speed change is set as 1s to make up for the operational impact caused by the insufficient wind speed detection equipment in simulation.

C. MARKOV DECISION PROCESSES (MDP) AND BELLMAN EQUATION

According to [9], it is understandable that a fundamental MDPs problem can be expressed as a five-tuple (S, A, P, R, γ), where S is a set of states, and A is a set of actions or controls. The transition probability P describes, for each state $s \in S$ and action $a \in A$; and the conditional probability $P^a_{ss'}$ of transition to state $s' \in S$ since the MDP is in state s and takes action a. The reward function R is the expected immediate cost paid after the transition to state $s' \in S$ and takes action $a' \in A$. It represents the short-term return. γ is a discount factor that is mainly used to balance current and future rewards.

The objective of MDPs is to find a policy $\pi(s, a)$ that allows the agent to get the maximum return G_t when taking the corresponding action a under state s. The return G_t is

the total discounted reward from time-step t. G_t is defined as equation (11).

$$G_t = R_{t+1} + \gamma^1 R_{t+2} + \gamma^2 R_{t+3} + \dots \tag{11}$$

where $0 < \gamma < 1$ is a discount factor that reduces the weight of rewards incurred further in the future. R_{t+1} represents the reward of the state s_t to s_{t+1}.

Since G_t is not a determined value (it involves a probabilistic selection action in the process), the cumulative return function is calculated using the expectation. The state value function $v^\pi(s)$ is the expected value of being in the state s given that the policy π. It is determined by action a, so add the actions to the equation. The action-value function can be defined as $q^\pi(s, a)$.

The key to finding optimal values and optimal policies that can be executed is the Bellman equation [7], [21]. The Bellman equation of the action-value function is as follows:

$$q^\pi(s, a) = R_s^a + \gamma \sum\nolimits_{s' \in S} P_{ss'}^a \sum\nolimits_{a' \in A} \pi\left(s', a'\right) q^\pi\left(s', a'\right) \tag{12}$$

where R_s^a is the expected return in state s and takes action a.

The Bellman equation is utilized to solve the MDP problem, which is to find the optimal policy and its corresponding action-value function. The optimal action-value function is defined as follows:

$$q^*(s, a) = max_\pi q^\pi(s, a). \tag{13}$$

by solving (9), the optimal action taken under the optimal policy can be obtained; that is, the learning objective of reinforcement learning is achieved.

III. DESIGN OF PITCH ROBUST CONTROLLER BASED ON REINFORCEMENT LEARNING

This section introduces the main controller design ideas and the control process of the controller. Then the design of the neural network is presented in detail. The controller proposed in this paper is based on the MDP framework, and its goal is to find the optimal strategy to satisfy the Bellman equation.

A. MAIN DESIGN IDEAS

The agent learns of reinforcement learning in a ''trial and error'' way, and through the interaction with the environment to obtain reward and punishment guidance action; the learning objective is to make the agent get the maximum reward. Therefore, the reinforcement learning algorithms are constructed on the idea that effective control decisions should be remembered, by a feat of a reinforcement signal, such that they become more likely to be used a second time. In the interaction, the reinforcement signal $r(t)$ is an evaluation index to judge the performance of an action.

From equation (1)–(4) of the energy transmission model of the wind turbine, we can find that the rotor speed can be modified by controlling β. However, how to obtain the control value through the current state, which involves the knowledge of dynamics and has high requirements for the solution of the nonlinear equation; and the processes are complicated. Thus, the excellent learning ability of neural networks can be utilized to combine neural networks with reinforcement learning, take the change value of the data of system state and wind speed as input, and through the training of the neural network, get an output. Then this output is passed through the mapping function to get the corresponding control value β.

Based on the principle of reinforcement learning and the control objective of the controller, a robust variable pitch controller based on reinforcement learning is proposed, and intended to be applied to the variable pitch control of the wind turbines in this paper. The control objective is tantamount to control the output rotor speed of the wind turbine to stabilize at the rated speed in the high wind speed zone. The controller fully considers the disturbance factors such as the nonlinearity of the system transmission model and the error of the wind speed detection signal.

The proposed controller is a type of ADP controller, which is composed of an action network to generate action and a critic network to evaluate this action. In the initial state of the system, the weights/parameters of the action network and the critic network are random. Firstly, the upper limit of rotation speed error of the wind wheel is preset. If the initial state is within the preset range, the corresponding reinforcement signal r is ''0'', indicating success. In the subsequent process, if the error exceeds the upper limit, r is ''-1'', indicating failure. The controller is only reinforced when control fails.

When the system state and the corresponding wind speed disturbance are observed, combining the wind speed disturbance at the previous moment, it will generate a corresponding action through the action network under the weight parameters based on the current state. The critic network ''critiques'' the generated action value to optimize a future ''reward-to-go'' by propagating a temporal difference between two consecutive estimates from the critic network. Then the critic network updates the weight with the reinforcement signal to get the optimal approximation value. This formulation is utterly consistent with the Bellman equation. It uses the obtained approximation to affect the weight update of the action network, to reduce the defined performance function value, and to get the current optimal output value of the action network. Finally, the output value yields a better control value through the mapping function

The connection and memory between the input of the action network and control output will strengthen the control output every time, thus making the control value of the output value mapping better control effect. In the case of a specific system state and corresponding wind speed, a better action output value will make the optimization equation more balanced.

B. THE CONTROL FLOW OF THE CONTROLLER

Each control process of the controller is completed by two neural networks adaptive learning. Fig.3 displays a schematic diagram of the data flow of the controller designed in

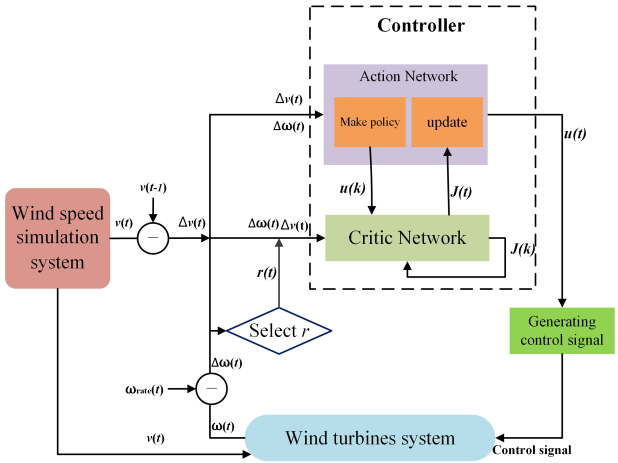

FIGURE 3. The schematic diagram of controller data flow.

this paper. The specific process of its control at time t is as follows:

1) To determine whether the angular velocity $\omega(t)$ of the wind turbine observed by the sensor exceeded the preset range or not, select the reinforcement signal $r(t)$.

2) Obtain wind speed disturbance data $v(t)$ through wind speed simulation system; and then, compared it with $v(t-1)$ to get the wind speed variation $\Delta v(t)$. Similarly, compared angular velocity $\omega(t)$ with the rated value $\omega_{rate}(t)$ to get the difference value $\Delta\omega(t)$. $\Delta v(t)$ and $\delta\omega(t)$ are taken as the input of the action network. The output $u(k)$ is obtained through the action network, k is the number of internal iterations.

3) $\Delta v(t)$, $\Delta\omega(t)$ and $u(t)$ is used as input to the critic network. The cumulative return approximation through the critic network is $J(k)$.

4) Train the critic network and update the network weight with the reinforcement signal $r(t)$. In conclusion, the critic network obtains the final $J(t)$

5) The neural network indirectly back propagates $J(t)$ to update the weights of the action network. Then when the iteration of the internal cycle for the action network complete, the final $u(t)$ is obtained.

6) Firstly, judge whether the previous control is successful or not. If it get failure, break out of the loop and start learning again from the initial state, and vice versa, move on to the next step.

7) Input $u(t)$ into the control signal generation system, obtain the corresponding control value and then update the wind turbine system states, enter the next cycle.

In order to better quantify the ''performance'' of the output, as shown in Fig.3, the critic network takes the input and output of the action network as the input of the network at the same time, and the output value J can approximate the discounted total reward-to-go referring to the equation (11) of accumulative reward. The approximate $R(t)$ at time t can be defined, as shown in equation (14).

$$R(t) = r(t+1) + \alpha^1 r(t+2) + \alpha^2 r(t+3) \ldots \quad (14)$$

where $R(t)$ is the value of the cumulative future reward at time t, and α is the discount factor for the infinite-horizon problem ($0 < \alpha < 1$). $\alpha = 0.9$ will be used in the implementations. r $(t+1)$ is the value of the external reinforcement signal when the time is t+1.

The controller designed in this paper is based on [22] and combined with the actual situation of the wind turbine system. The detailed design process of the neural network will be given in parts C and D.

C. CRITIC NETWORK

The output value $J(t)$ of the critic network is used as the approximate value of $R(t)$ in (10) to predict the accumulative ''reward-to-go'' of the control output of the action network.

The TD algorithm updates the value function online, and it will get the state value of the current state that only needs to wait until the jump to the next state. However, there is an error between the real value and the evaluated value. As it is an update process, the purpose is to minimize the error between the final predicted value and the actual value. Therefore, the learning goal of the critic network in the controller is to minimize the error between the amount of $J(t)$ and the real value function, while optimizing the future accumulative ''reward-to-go.'' It shows the parameter tuning diagram in Fig.4.Thus, based on the above ideas, the prediction error for the element of the critic is defined as equation (15).

$$e_c(t) = \alpha J(t) - [J(t-1) - r(t)] \quad (15)$$

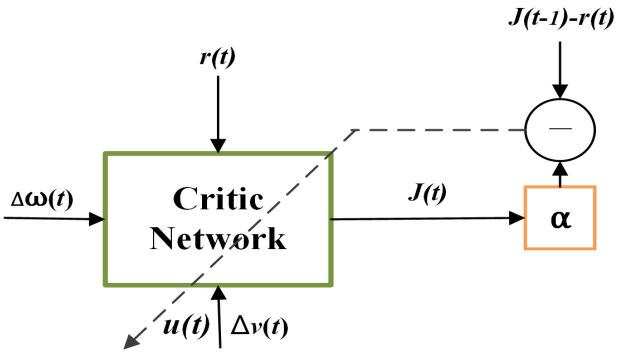

FIGURE 4. The schematic diagram for parameter tuning of the critic network. (solid black line is the signal flow, the gray dashed line is the parameter tuning path.)

The objective function to be minimized in the critic network is shown in equation (16).

$$E_c(t) = \frac{1}{2} e_c^2(t). \quad (16)$$

Fig.5 is the construction of the critic network. It is a BP neural network with a hidden layer. Where, $\{x_1, x_2, \ldots, x_n\}$ and u are the input and output of the action network, respectively. J is the output of the critic network. $J(t)$ will be obtained from equation (17)-(19).

$$q_i(t) = \sum_{j=1}^{n+1} w_{cij}^{(1)}(t)\, x_j(t), \quad i = 1, \ldots, N_h \quad (17)$$

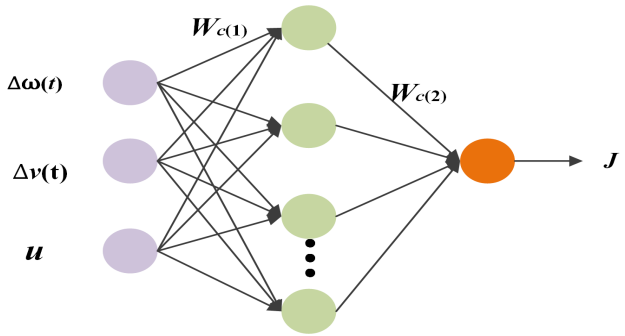

FIGURE 5. The schematic diagram of the critic network using a feedforward network with one hidden layer.

$$p_i(t) = \frac{1 - exp^{-q_i(t)}}{1 + exp^{-q_i(t)}} \tag{18}$$

$$J(t) = \sum_{i=1}^{N_h} w_{c_i}^{(2)}(t) p_i(t) \tag{19}$$

where

- q_i is the ith hidden node input of the critic network,
- p_i is the corresponding output of the ith hidden node,
- N_h is the total number of the hidden nodes in the critic network,
- $n+1$ is the total number of inputs into the critic network, including the action value $u(t)$ from the action network,
- w_c is the weight vector in the critic network.

According to the error propagation equation of the back-propagation algorithm, and the chain rule, the adaptation of the critic network is summarized as follow:

1)$\Delta w_c^{(2)}$(hidden to output layer)

$$\begin{aligned}\Delta w_{c_i}^{(2)}(t) &= l_c(t)\left[-\frac{\partial E_c(t)}{\partial w_{c_i}^{(2)}(t)}\right] \\ &= l_c(t)\left[-\alpha e_c(t) p_i(t)\right]\end{aligned} \tag{20}$$

2)$\Delta w_c^{(1)}$(input to hidden layer)

$$\begin{aligned}\Delta w_{c_{ij}}^{(1)}(t) &= l_c(t)\left[-\frac{\partial E_c(t)}{\partial w_{c_{ij}}^{(1)}(t)}\right] \\ &= -\alpha l_c(t) e_c(t) w_{c_i}^{(2)}(t) \\ &\quad \cdot[\frac{1}{2}(1-p_i^2(t))]x_j(t)\end{aligned} \tag{21}$$

where $l_c(t) > 0$ is the learning rate of the critic network at time t, which usually decreases with time to a small value.

D. ACTION NETWORK

The action network expects that the mapping control value of each output can make the control successful. The parameter tuning diagram of the action network is shown in Fig.6. In the preceding part of the text, "0" was defined as the reinforcement signal for "success," in order to satisfy the Bellman equation and maximize the state value function, the ultimate learning target denoted by U_c, is set to "0" in the paradigm. Through observation, it is found that the principle of the adjustment of the action network is to indirectly

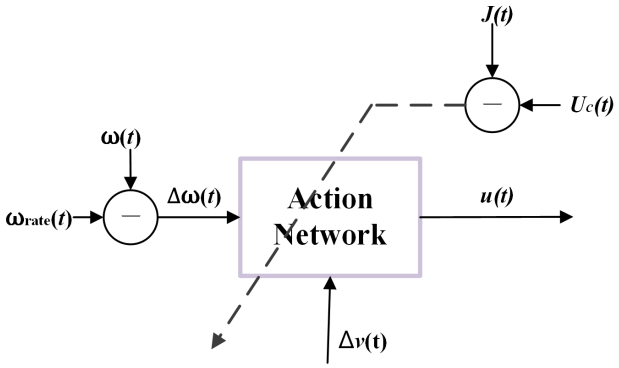

FIGURE 6. The schematic diagram for parameter tuning of the action network. (solid black line is the signal flow; the gray dashed line is the parameter tuning path.)

back-propagate the error between the approximate J function from the critic network and U_c. Let:

$$e_a(t) = J(t) - U_c(t) \tag{22}$$

The objective of updating the weights in the action network is to minimize the following performance error measure:

$$E_a(t) = \frac{1}{2} e_a^2(t) \tag{23}$$

The action network implemented by a feedforward network is similar to the critic network that is shown in Fig.5. Except that the inputs are system states and wind speed disturbances, and the output is $u(t)$. The relevant equations of the action network are defined as follows:

$$m_i(t) = \sum_{j=1}^{n} w_{a_{ij}}^{(1)}(t) x_j(t), \quad i = 1, \ldots, N_h \tag{24}$$

$$n_i(t) = \frac{1 - exp^{-m_i(t)}}{1 + exp^{-m_i(t)}} \tag{25}$$

$$v(t) = \sum_{i=1}^{N_h} w_{a_i}^{(2)}(t) n_i(t) \tag{26}$$

$$u(t) = \frac{1 - exp^{-v(t)}}{1 + exp^{-v(t)}} \tag{27}$$

where

- m_i is the ith hidden node input of the action network,
- n_i is the corresponding output of the ith hidden node of the action network,
- v is the ith output node input of the action network,
- u is the corresponding output of the ith output node of the action network, control value,
- w_a is the weight vector in the action network.

The number of inputs to the action network and the critic network is different, and the input of the action network is the difference value of the measured states and wind speed disturbances of two adjacent steps. The action network adds a transfer function to the output layer that the critic network does not. Referring to the parameter update rules of the critic network, and the parameter update rules of the action network are summarized, as shown in equation (28)-(31).

1)$\Delta w_a^{(2)}$(hidden to output layer)

$$\Delta w_{a_i}^{(2)}(t) = l_a(t)\left[-\frac{\partial E_a(t)}{\partial w_{a_i}^{(2)}(t)}\right] \tag{28}$$

$$\frac{\partial E_a(t)}{\partial w_{a_i}^{(2)}(t)} = e_a(t)\left[\frac{1}{2}\left(1-u^2(t)\right)\right] n_i(t) \cdot \sum_{i=1}^{N_h}\left[\frac{1}{2}w_{c_i}^{(2)}(t)(1-p_i^2(t))w_{c_{i,n+1}}^{(1)}(t)\right] \tag{29}$$

2)$\Delta w_a^{(1)}$(input to hidden layer)

$$\Delta w_{a_{ij}}^{(1)}(t) = l_a(t)\left[-\frac{\partial E_a(t)}{\partial w_{a_{ij}}^{(1)}(t)}\right] \tag{30}$$

$$\frac{\partial E_a(t)}{\partial w_{a_{ij}}^{(1)}(t)} = e_a(t)\left[\frac{1}{2}\left(1-u^2(t)\right)\right] \cdot w_{a_i}^{(2)}(t)\left[\frac{1}{2}\left(1-n_i^2(t)\right)\right] x_j(t) \cdot \sum_{i=1}^{N_h}\left[\frac{1}{2}w_{ci}^{(2)}(t)(1-p_i^2(t))w_{c_{i,n+1}}^{(1)}(t)\right] \tag{31}$$

where $l_a(t) > 0$ is the learning rate of the action network at time t.

In conclusion, normalization is executed in both networks to confine the values of the weights into appropriate scope by equation (32)-(33).

$$\mathrm{w}_a(t+1) = \frac{\mathrm{w}_a(t) + \Delta \mathrm{w}_a(t)}{\| \mathrm{w}_a(t) + \Delta \mathrm{w}_a(t)\|_1} \tag{32}$$

$$\mathrm{w}_c(t+1) = \frac{\mathrm{w}_c(t) + \Delta \mathrm{w}_c(t)}{\| \mathrm{w}_c(t) + \Delta \mathrm{w}_c(t)\|_1} \tag{33}$$

IV. SIMULATION

The simulation experimental environment 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

In the simulation experiment, the dynamic model of the wind turbine is shown in equation (3). By combining equation (1) and (2), the nonlinear differential equation of the system can be solved, and obtain the current state of the angular velocity of the wind rotor. The data of a 1.5MW large variable-rotor wind turbine is used to test the effectiveness of the proposed controller. The main parameters are shown in the Table 2 as follows [23]:

The accurate mathematical model of wind speed is one of the important conditions for evaluating the performance of the variable pitch control system. Then, the wind speed model adopted in this simulation can be obtained from equation (5)-(10). Table 3 shows the values of the parameters of the model, and T is the wind speed period. After simulation in MATLAB, we can obtain a typical analog signal diagram of wind speed in Fig.7.

TABLE 2. Parameters for wind turbine.

Symbol	Quantity	Value
R	blade radius	40.25 m
ρ	air density of wind farm	1.25 kg/m^3
V_N	rated wind speed	12.5 m/s
J_r	inertia of the rotor	$4.9 \times 10^6\ kg \cdot m^2$
J_g	inertia of the generator	$107.87\ kg \cdot m^2$
N	transmission ratio	104
T_e	rated electromagnetic torque of generator	9000 N · m
n	rated speed of the wind rotor	17.5 r/min
ω_{rate}	rated angular velocity of the wind rotor	1.83 r/s

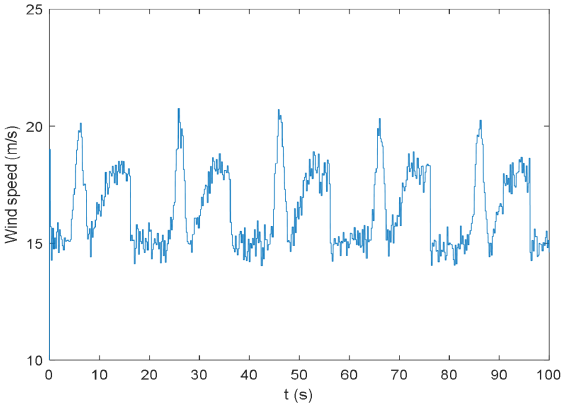

FIGURE 7. A typical wind speed analog signal diagram based on the ARMA model. (Average of 19m/s wind speed curve.)

The values and symbolic meanings of the learning parameters of the neural network in the controller are as the same as the one in [22], which is the usual setup for neural networks and depends on some prior experience. The specific settings are as follows:

-$\boldsymbol{l_c}$ (**0**) 0.3, initial learning rate of the critic network,

-$\boldsymbol{l_a}$(**0**) 0.3, initial learning rate of the action network,

- $\boldsymbol{l_c}(t)$ learning rate of the critic network at time t that is decreased by 0.05 every five-time steps until it reaches 0.005 and it stays at 0.005 thereafter,

- $\boldsymbol{l_a}(t)$ learning rate of the action network at time t that is decreased by 0.05 every five-time steps until it reaches 0.005 and it stays at 0.005 thereafter,

- $\boldsymbol{N_c}$ 50, internal cycle of the critic network,

- $\boldsymbol{N_a}$ 100, internal cycle of the action network,

- $\boldsymbol{T_c}$ 0.05, internal training error threshold for the critic network,

TABLE 3. Parameters for wind speed model.

Symbol	Value	Symbol	Value
v_b	15 m/s	T_{1g}	4s
G_{max}	5 m/s	T_{2g}	8s
R_{max}	3 m/s	T_{1r}	8s
v_{nmax}	1 m/s	T_{2r}	12s
T	20s	T_r	4s

- $\boldsymbol{T_a}$ 0.005, internal training error threshold for the action network,

- $\boldsymbol{N_h}$ 6, number of the hidden nodes.

In the simulation, the target set is that the error between the speed and the rated speed is within 0.05. If the goal is met, the control is successful, that is, the reinforcement signal $r(t)$ is 0; otherwise, $r(t)$ is 1.

The weight update of the action network and critic network is realized through its internal cycle. In each step, the maximum number of parameter updates of each neural network is the number of the set internal loop value (N_a and N_c). Alternatively, the training error of the neural network is less than the set error threshold (T_c and T_a). This processing limits the number of training for the neural network and to stop training when the value reaches the predetermined threshold.

To observe the influence of different factors on the controller, three implementation scenarios were studied in various settings. In Setting 1, the run consists of a maximum of 1000 consecutive trials. It is considered successful if the last trial (the number less than 1000) of the run has lasted 5000-time steps (with a step size of 0.02s). In Setting 2, the run consists of a maximum of 2000 consecutive trials, and the other parameters are the same as Setting 1. In Setting 3, the step size is 0.01s, and the other parameters are the same as Setting 1; if the last trial (the number less than 1000) of the run has lasted 10000-time steps, this trial is considered auspicious.

The proposed variable pitch robust controller, which is based on reinforcement learning, has been evaluated, and it summarizes the results in Table 4. The simulation results of the experiments in Table 4 are the data averages of 100 simulation experiments, and the initial state of each experimental run is random. If a run is unbeaten, it then records the number of trials. The number of trials in Table 4 is the average of successful trials. The percentage of successful experiments (out of 100) is also necessary to record.

TABLE 4. Performance evaluation of proposed variable pitch robust controller.

Implementation	Success Rate	# of trials
Setting 1	95%	57
Setting 2	99%	92
Setting 3	92%	165

According to the Table 4, compare the data onto Setting 1 and Setting 2, it can be concluded that if the learning chance of the controller increase appropriately, the success rate of the controller will increase obviously. By comparing Settings 1 and 3, it can also be concluded that the success rate of the controller can be improved if the step size is appropriately reduced while other Settings remain unchanged.

In the wind turbines system, the disturbance of the randomness of wind speed is significant, and the uncertain factors more, so that the relationship between different states is difficult to find the regularity. However, the proposed controller is required to seek out the association rules between different states and their corresponding control actions to control the stability of the system. Therefore, it is difficult for the controller to reinforce the process of positive reinforcement. Due to the randomness of wind speed and the existence of uncertain factors, the time required for the controller to learn successfully will also be different. Since the increase in the maximum number of experiments will make the controller have more chances to learn, the success rate of the controller will increase. By narrowing the step size, the controller can find the changing trend of the system state more quickly in order to adjust the control action in time and to fluctuate the output of the system within the specified range.

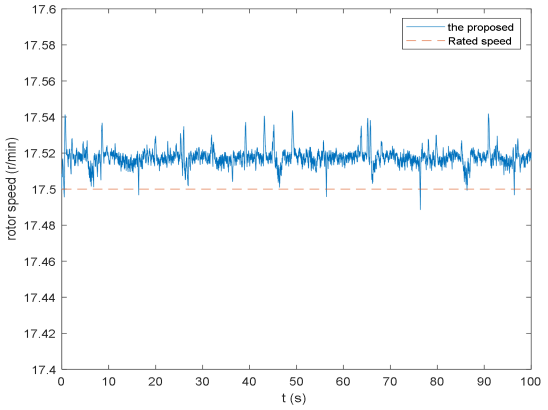

FIGURE 8. A typical schematic diagram of the speed trajectory during a successful learning trial for the wind turbine. The solid line represents the trajectory of the proposed controller; the dashed line is the rated speed of the wind turbine.

Fig.8 is a typical schematic diagram of the speed trajectory during a successful learning trial for the wind turbine. It carries the simulation out under the premise that the wind speed is higher than the rated wind speed. Combined with Fig. 7, we can observe that the speed of the wind turbine fluctuates within a tiny range of rated speed when the wind speed is random with a large disturbance. Continue to observe that in the gust wind speed section where the wind speed changes sharply, the wind rotor speed can still be stable near its rated value, and its error is within the objective error. Therefore, the variable pitch robust controller based on reinforcement learning proposed in this paper can effectively achieve reliable control and has good robustness.

The evolution of the use of wind energy C_p during the control process of Fig. 9 can be observed. Under the circumstance of random variation of wind speed and fluctuation of rotor speed around the rating value, we can find a pattern that C_p value will decrease when the wind speed increases; and C_p value will increase when the wind speed decrease. Therefore, C_p value can be adjusted to some extent by controlling the stability of wind rotor speed, so that the power of wind turbine can be adjusted accordingly.

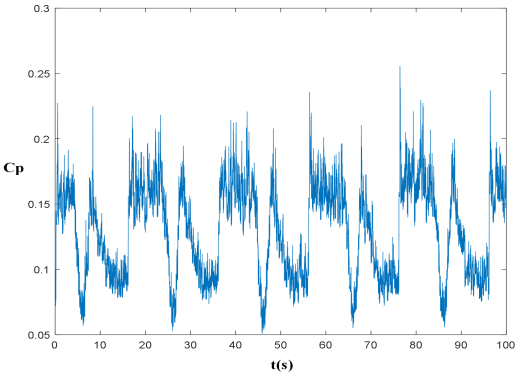

FIGURE 9. The change curve of wind energy utilization c_p during the control process.

β, as the input of wind turbine, controls the stability of the rotor speed. The curves of the input as shown in Fig.10. Then, we can observe that the pitch angle has a smaller interval, so its response time will be relatively short. This improves the control accuracy to some extent.

In the past, some variable pitch controllers of wind turbine proposed to control the power and rotate speed. The comparison between the data and simulation results shows the advantages of the new controller in the context of controlling errors. The relevant comparison of the new controller based on reinforcement learning and the controllers in other literatures is shown in Table 5. The error floating ratio is defined as follows:

$$g_a = \frac{e_a}{\omega^*} \tag{34}$$

$$g_b = \frac{e_b}{\omega^*}. \tag{35}$$

where g_a is the upper floating error ratio, g_b is the lower floating error ratio. e_a represents the maximum value of the floating error above the rated speed, that is, the value of the floating number above the maximum rated speed minus the rated speed. Similarly, e_b represents the maximum value of the floating error below the rated speed. ω^* is the rated speed of wind turbines. Where g_a is the upper floating error ratio, g_b is the lower floating error ratio.

As can be seen from Table 5, compared with the previous work, the controller proposed in this paper has a smaller error between the wind rotor speed and the rated value, and its stability is stronger. Wind speed changes at random and the disturbance is large, but the control effect of the proposed controller is proper, so the controller has strong robustness.

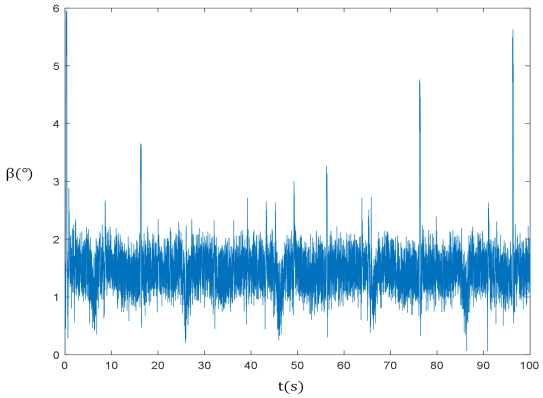

FIGURE 10. The change curve of pitch angle β during the control process.

TABLE 5. The floating error ratio between the proposed controller and controllers from other literatures.

Controller	Upper floating error ratio g_a(%)	Lower floating error ratio g_b(%)
The Proposed	0.2286	0.1714
Fuzzy Neural Network [24]	7.3333	6.3333
Fuzzy control [25]	12.6667	10.0667
Nonlinear PID control [26]	17.3333	13.6667

V. CONCLUSION

This paper combines adaptive control with optimal control using computer intelligence technology; and introduces the method from reinforcement learning to the adaptive controller. By measuring the objective data of the system, it converges to the optimal control solution in real-time.

When the operating state of the wind turbines deviates from the stable point, it will significantly reduce the control effect of the ordinary PID, and most of the pitch angle controllers based on modern control theory are sometimes difficult to implement. In this paper, a robust variable pitch controller based on reinforcement learning is proposed. Of which the purpose is to stabilize the output speed of the control system at the rated speed when the wind turbine is in an environment with higher wind speed than the rated speed. On the basis of the simulation results, it is concluded that the control performance of the proposed controller is better than others, and system speed can be maintained at the rated rotor speed basically; and it also controls the fluctuation in a small range.

However, the proposed controller still has some problems, such as failure rate, the number of successful tests is large, and it needs to be improved in terms of duration. The next step should be to solve these issues, analyze more connection between the dynamics of the system, and improve the training of the neural networks to achieve the effect of increasing the success rate and speeding up learning. In the following research, I will further try to study the wind turbine model under the action of wake interaction and tower shadow effect, as well as the power level variation of the system at the load side. On this premise, the design of the controller can

minimize the consumption of the equipment, find the optimal rotational speed of the wind turbine in different wind speed segments, and achieve the optimal utilization rate of wind energy of the wind turbine.

At the same time, the development of machine learning is rapid, and new learning algorithms emerge endlessly. In the process of research, to further solve problems that the convergence success rate of machine learning in the control system, different algorithms need to be tried to improve the design of the controller. According to the advantages of the algorithms, the algorithm is combined with practical application to design better controllers to efficiently solve problems. While the theory is developing, the practical application should keep up with the theory. That is what must be carried out in the future.

REFERENCES

[1] F. Blaabjerg and K. Ma, "Future on power electronics for wind turbine systems," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 1, no. 3, pp. 139–152, Sep. 2013.
[2] L.-R. Chang-Chien, C.-C. Sun, and Y.-J. Yeh, "Modeling of wind farm participation in AGC," *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1204–1211, May 2014.
[3] H. Wen-Sheng and D. Yan-Jun, "Research on new pitch control algorithm of wind power generators," *Power Electron.*, vol. 47, no. 2, pp. 53–54, Feb. 2013.
[4] W. Yang, H. Geng, and S. Xiao, "PI-R individual pitch control for large-scale wind turbine," *Electr. Power Autom. Equipments*, vol. 37, no. 1, pp. 87–92, Jan. 2017.
[5] P. Li, W. Hu, R. Hu, Q. Huang, J. Yao, and Z. Chen, "Strategy for wind power plant contribution to frequency control under variable wind speed," *Renew. Energy*, vol. 130, pp. 1226–1236, Jan. 2019.
[6] S. Jalbi and S. Bhattacharya, "Minimum foundation size and spacing for jacket supported offshore wind turbines considering dynamic design criteria," *Soil Dyn. Earthquake Eng.*, vol. 123, pp. 193–204, Aug. 2019.
[7] R. S. Stuton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA, MIT Press, 2018.
[8] D. Lee, H. Seo, and M. W. Jung, "Neural basis of reinforcement learning and decision making," *Annu. Rev. Neurosci.*, vol. 35, no. 1, pp. 287–308, Jul. 2012.
[9] 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.
[10] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, FL, USA: CRC Press, 2009.
[11] P. J. Werbos, "Approximate dynamic programming for realtime control and neural modelling," in *Handbook of Intelligent Control*, D. A. White and D. A. Sofge, Eds. New York, NY, USA: Van Nostrand Reinhold, 1992.
[12] C.-L. Lee and J.-G. Juang, "Aircraft landing control in wind shear condition," in *Proc. Int. Conf. Mach. Learn.*, Guilin, China, Jul. 2011, pp. 1180–1185.
[13] M. Lungu and R. Lungu, "Autonomous adaptive control system for airplane landing," *Asian J. Control*, vol. 21, no. 3, pp. 1328–1341, May 2019.
[14] B. Zhao, S. Xu, J. Guo, R. Jiang, and J. Zhou, "Integrated strapdown missile guidance and control based on neural network disturbance observer," *Aerosp. Sci. Technol.*, vol. 84, pp. 170–181, Jan. 2019.
[15] C. Lu, J. Si, and X. Xie, "Direct heuristic dynamic programming method for power system stability enhancement," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 1008–1013, Aug. 2008.
[16] D. Prokhorov, *Computational Intelligence in Automotive Applications*. New York, NY, USA: Springer-Verlag, 2008.
[17] J. Liu and Y. L. Zhou, "Research on power smoothing control strategy for permanent magnet synchronous wind turbine units," *Elect. Autom.*, vol. 41, no. 2, pp. 25–28, Feb. 2019.
[18] 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.
[19] Z. Gao, "The modeling of wind speed based on MATLAB," *Int. J. Sci. Res.*, vol. 5, no. 3, pp. 2319–7064, Mar. 2016.
[20] C. Wu and M. Lu, "The modeling and simulation for wind model of wind power," in *Proc. Int. Conf. Elect. Control Eng.*, Yichang, China, Sep. 2011, pp. 2715–2717.
[21] W. B. Powell, *Approximate Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2007.
[22] J. Si and Y.-T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.
[23] X. X. Lu, S. Fu, and W. Y. Tong, "Robust control for variable Pitch of wind turbines," *J. Shenyang Univ. Technol.*, vol. 40, no. 6, pp. 627–631, Jun. 2018.
[24] Q. S. Liu and S. Q. Qian, "Sliding mode variable pitch control of wind turbine via fuzzy neural network," in *Proc. CCC*, Hefei, China, 2012, pp. 3187–3191.
[25] V. Galdi, A. Piccolo, and P. Siano, "Designing an adaptive fuzzy controller for maximum wind energy extraction," *IEEE Trans. Energy Convers.*, vol. 23, no. 2, pp. 559–569, Jun. 2008.
[26] A. Gambier and Y. Yunazwin Nazaruddin, "Nonlinear PID control for pitch systems of large wind energy converters," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Copenhagen, Denmark, Aug. 2018, pp. 996–1001.

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.

FUXIAO TAN (Member, IEEE) received the Ph.D. degree in control theory and control engineering from Yanshan University, Qinhuangdao, China, in 2009. He is currently a Professor with the College of Information Engineering, Shanghai Maritime University, Shanghai, China. His current research interests include deep reinforcement learning, adaptive filtering, online sparse kernel learning, adaptive dynamic programming, cooperative control of multiagent systems, and robust control and dynamic optimization of nonlinear systems.

JUN WANG (Senior Member, IEEE) is currently a Full Professor of computer science and engineering with the University of Central Florida, Orlando, FL, USA. He has authored over 80 publications in premier journals, such as the IEEE TRANSACTIONS ON COMPUTERS and the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and leading HPC and systems conferences, such as VLDB, HPDC, EuroSys, ICS, Middleware, FAST, and IPDPS. His current research interests include computer systems, high performance computing, and deep reinforcement learning.