Wind-Farm Power Tracking via Preview-Based Robust Reinforcement Learning

Hongyang Dong and Xiaowei Zhao

***Abstract*—This paper aims to address the wind-farm power tracking problem, which requires the farm's total power generation to track time-varying power references and therefore allows the wind farm to participate in ancillary services such as frequency regulation. A novel preview-based robust deep reinforcement learning (PR-DRL) method is proposed to handle such tasks which are subject to uncertain environmental conditions and strong aerodynamic interactions among wind turbines. To our knowledge, this is for the first time that a data-driven model-free solution is developed for wind-farm power tracking. Particularly, reference signals are treated as preview information and embedded in the system as specially designed augmented states. The control problem is then transformed into a zero-sum game to quantify the influence of unknown wind conditions and future reference signals. Built upon the H_∞ control theory, the proposed PR-DRL method can successfully approximate the resulting zero-sum game's solution and achieve wind-farm power tracking. Time-series measurements and long short-term memory (LSTM) networks are employed in our DRL structure to handle the non-Markovian property induced by the time-delayed feature of aerodynamic interactions. Tests based on a dynamic wind farm simulator demonstrate the effectiveness of the proposed PR-DRL wind farm control strategy.**

***Index Terms*—Reinforcement learning, wind farm control, model-free control, wind power, renewable energy.**

I. Introduction

As one of the most important renewables, wind energy plays a key role in the essential move towards net-zero emissions. Over 200GW wind power capacity has been installed in Europe by 2020, accounted for 14% of the total electricity demand. With the rapid development of wind energy, how to operate wind farms efficiently has become a bottleneck problem in the wind industry. The main challenges of wind farm control problems come from the uncertain wind conditions and the time-varying aerodynamic couplings among wind turbines in the farm. A commonly-used wind farm control strategy is to establish analytical or parametric wind-farm models firstly and then design controllers based on them. Following this pattern, many model-based methods have been proposed to optimize the power generation of wind farms [1], [2]. However, due to high system complexities, model-based wind farm control methods suffer from uncertainties and unmodelled dynamics, and thus in practice they could have quite different performance compared with theoretical results. Model-free methods are promising alternatives to avoid these drawbacks. Some attempts to this end were presented in [3], [4], [5], [6], [7], [8]. Notably, several reinforcement learning (RL)-based methods were proposed in [5], [6], [7], [8] to maximize wind-farm power generation. However, all these elegant results either employed mean power outputs or still relied on the power outputs estimated by underlying wake models to carry out the learning process. They cannot react to real-time/instantaneous measurements or handle time-series data. Also, they lack the extending capacity to undertake complex tasks that require dynamic control trajectories, such as farm-level power tracking used in grid ancillary services.

Wind farms are gradually replacing traditional power plants, leading to low-inertia power systems and a decrease in the available supply of ancillary services. Therefore, employing wind farms to provide ancillary services becomes necessary to ensure the safety and stable operation of the power grid. This has aroused extensive research interest recently [9], [10], [11], [12], [13], [14], [15], [16]. For example, wind farms can achieve secondary frequency control (SFC, or referred to as automatic generation control (AGC) in the literature) - a main ancillary service that can be employed to regulate grid frequency [9], balance power supply with load [10], and maintain scheduled power exchanges between areas [11]. To achieve this, the farm's total power generation is required to track a reference power signal set by the system operators over several minutes or tens of minutes [9], and the power generation of each turbine in the farm needs to be controlled cooperatively in order to achieve a good tracking performance. We note that wind-farm power tracking is a much more complicated task than wind-farm power generation maximization. Particularly, the instantaneous control inputs can lead to a long-term influence on turbines' aerodynamic interactions and the whole farm's power generation outputs, rendering power tracking a difficult task for wind farms. Some studies [12], [13], [14], [15], [16] achieved this goal by carrying out induction control for all the turbines in the farms. However, these important results either depended on underlying simplified wake models or relied on accurate estimations of future power generations (or other states that are directly related to power generations, such as rotor-averaged wind velocities). They are sensitive to modelling errors and uncertainties, degrading their feasibility and applicability in practical applications.

The deep reinforcement learning (DRL)-based control strategy has the potential to address the aforementioned limitations of the existing wind farm control approaches for farm-level power tracking. DRL [17] is a cutting-edge artificial intelligence area, which has been applied in many systems such as

This work was funded by the UK Engineering and Physical Sciences Research Council (grant number: EP/S001905/1). H. Dong and X. Zhao (Corresponding Author) are with the Intelligent Control & Smart Energy (ICSE) Research Group, School of Engineering, University of Warwick, Coventry CV4 7AL, UK. Emails: hongyang.dong@warwick.ac.uk, xiaowei.zhao@warwick.ac.uk.

board games, robots, satellites, and power systems [18], [19], [20], [21]. A DRL agent focuses on improving its control policy to optimize a long-term reward via its interactions with the environment. Distinct from conventional DRL design, a novel preview-based robust DRL (PR-DRL) structure is proposed in this paper to handle time-varying references and uncertain environmental conditions. Specifically, reference signals are treated as preview information and embedded in the system as augmented system states. This allows us to transform the tracking control problem into a zero-sum game and therefore quantify the influence of the uncertain environmental conditions and future reference signals. Also, time-series states are employed to address the non-Markovian property of wind-farm power tracking problems, guaranteeing the fundamental requirement of DRL. Built upon the H_∞ control theory and the deep deterministic policy gradient (DDPG) algorithm [22], our PR-DRL method can approximate the solution of the resulting zero-sum game. In addition to the actor-critic mechanism, an additional deep neural network (DNN) structure, termed a distractor, is employed in our design to evaluate the worst-case exogenous inputs (i.e. unknown wind conditions and future reference signals) with respect to a user-defined performance index, bringing strong robustness to the whole system. Long short-term memory (LSTM) networks are employed in our PR-DRL to handle time-series data and address the non-Markovian property induced by the time-delayed feature of aerodynamic interactions. The main contributions of this paper are summarized below.

- To our knowledge, this is for the first time that a data-driven model-free solution is developed for wind-farm power tracking. The proposed PR-DRL method addresses this challenging problem by employing only the system's input & output data without requiring any analytical model. It overcomes the drawbacks of model-based wind farm control methods [1], [2], [9], [12], [13], [14], [15], [16] that are sensitive to modelling errors and uncertainties.
- The proposed PR-DRL method addresses the limitations of existing DRL-based wind farm control methods [5], [6], [7], [8]. It can handle complex tasks (i.e. power tracking) under uncertain environmental conditions, bringing strong adaptability and robustness to the whole system. Compared with the recent robust DRL algorithm in [23], our PR-DRL can deal with preview information and avoid the additional internal loop for actor updating as required in [23], rendering an easy-to-implement framework with enhanced generality.
- The proposed PR-DRL method does not need the assumption of full-flow state measurements, which was used in most recent wind-farm power tracking methods [15], [16]. It only needs time-series measurements at turbine rotors (instead of at all spatial cells of the staggered grid over the whole flow field as in [15], [16]) and uses LSTM networks to capture the key information regarding power tracking tasks. This feature offers strong applicability to real wind farms.

The remainder of this paper is organized as follows. The PR-DRL method is designed in Sec. II. Then it is adapted in Sec. III to handle wind-farm power tracking tasks. After that, case studies with a dynamic wind farm simulator are demonstrated in Sec. IV. Finally, we conclude the paper in Sec. V.

II. Development of Preview-Based Robust DRL

A. Preview-Based Robust Control

A preview-based robust DRL method for tracking control problems is proposed in this section. We start with a general discrete-time system:

$$x(k+1) = f(x(k), u(k), w(k)) \tag{1}$$

In Eq. (1), $x(k) \in \mathbb{R}^n$ and $x(k+1) \in \mathbb{R}^n$ are the system state and its successor, respectively. In addition, $u \in \mathbb{R}^m$ denotes the system's control input, $w \in \mathbb{R}^l$ denotes the external disturbance, and $f(x, u, w)$ is an unknown mapping from $x(k)$, $u(k)$ and $w(k)$ to $x(k+1)$ for any time instant k.

We use $x_d \in \mathbb{R}^n$ to denote the reference control signal, and $x_e(k) = x(k) - x_d(k)$ is the tracking error. By substituting x_d into Eq. (1), one has

$$x_e(k+1) = f(x_e(k) + x_d(k), u(k), w(k)) - x_d(k+1) \tag{2}$$

If the preview information of the reference signal x_d is available, it can enable the control policy to act in advance and therefore enhance performance. We consider a N time-step preview, which means that, at time k, $x_d(k)$, $x_d(k+1)$, ..., and $x_d(k+N-1)$ are available for controller design. Here $N \in \mathbb{N}^+$, and $N = 1$ means there is no preview.

To handle preview information, we define $\bar{x}_d(k) = [x_d(k)^{\mathrm{T}}, x_d(k+1)^{\mathrm{T}}, ..., x_d(k+N-1)^{\mathrm{T}}]^{\mathrm{T}}$. Then one has

$$\bar{x}_d(k+1) = \Phi_1 \bar{x}_d(k) + \Phi_2 x_d(k+N) \tag{3}$$

$$x_d(k) = \Phi_3 \bar{x}_d(k) \tag{4}$$

where

$$\Phi_1 = \begin{bmatrix} 0 & I & 0 & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & & \dots & I \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}, \ \Phi_2 = \begin{bmatrix} 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & \vdots \\ \vdots & \dots & \vdots & 0 \\ 0 & \dots & 0 & I \end{bmatrix}$$

and

$$\Phi_3 = \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix}$$

Therefore, Eq. (2) can be transformed to be

$$x_e(k+1) = f(x_e(k) + \Phi_3 \bar{x}_d(k), u(k), w(k)) - \Phi_3 \bar{x}_d(k+1) \tag{5}$$

$$\bar{x}_d(k+1) = \Phi_1 \bar{x}_d(k) + \Phi_2 x_d(k+N) \tag{6}$$

Defining $\bar{x}(k) = [x_e(k)^{\mathrm{T}}, \bar{x}_d(k)^{\mathrm{T}}]^{\mathrm{T}}$ and $\bar{w}(k) = [w(k)^{\mathrm{T}}, (x_d(k+N) - x_d(k+N-1))^{\mathrm{T}}]^{\mathrm{T}}$, then Eqs. (3) and (4) can be re-organized to the following compact form.

$$\bar{x}(k+1) = F(\bar{x}(k), u(k), \bar{w}(k)) \tag{7}$$

Based on Eq. (7), we aim to design a controller to satisfy the following inequality for any $\bar{w} \in \mathcal{L}_\infty$:

$$\sum_{i=k}^{\infty} \rho^{i-k} R(\bar{x}(i), u(i)) \leq \gamma^2 \sum_{i=k}^{\infty} \rho^{i-k} \|\bar{w}(i)\|_P^2 \tag{8}$$

Here $R(\bar{x}, u)$ is a user-defined reward function to reflect control objectives, and $\|\bar{w}\|_P^2 \triangleq \bar{w}^{\mathrm{T}} P \bar{w}$ with $P \geq 0$. Also, $\rho \in (0, 1]$ denotes a user-defined discount factor, and $\gamma > 0$ denote a prescribed level of disturbance attenuation.

The system described in Eq. (7), along with Eq. (8), forms a typical H_∞ control task. With Eq. (8), we can quantitatively describe the influence induced by a general disturbance vector $\bar{w}$ (including the unknown external disturbance and the uncertain future reference signal) to the tracking control system. According to Eq. (8), we consider the following performance metric that is commonly mentioned as a state value function:

$$V(k) \triangleq \sum_{i=k}^{\infty} \rho^{i-k} l(\bar{x}(i), u(i), \bar{w}(i)) \tag{9}$$

where $l(\bar{x}, u, \bar{w}) = R(\bar{x}, u) - \gamma^2 \|\bar{w}\|_P^2$ is the step reward. Then the whole control problem can be regarded as a zero-sum game [24] between the control policy u and the unknown general disturbance vector $\bar{w}$. Specially, u aims to minimize the long-term reward function in Eq. (9) while $\bar{w}$ aims to maximize it. We need to find an optimal control policy u^* under the potential worst-case $\bar{w}$ (denoted by $\bar{w}^*$), i.e.

$$u^*(k) = \arg\min_u V^*(k) \tag{10}$$

and here

$$\begin{aligned} V^*(k) &= V^*(\bar{x}(k), u^*(k), \bar{w}^*(k)) \\ &= \min_u \max_{\bar{w}} \{V(\bar{x}(k), u(k), \bar{w}(k))\}, \forall k \in \mathbb{N}^+ \end{aligned} \tag{11}$$

However, the Nash equilibrium $(u^*(x), \bar{w}^*(x))$ of such a zero-sum game is almost impossible to be analytically solved subject to an unknown, nonlinear system as in Eq. (7). In this paper, we use DRL to solve this problem. Based on the Bellman's optimality principle, a vital feature of V^* is given in the following discrete-time Hamilton-Jacobi-Isaacs equation.

$$V^*(k) = \min_u \max_{\bar{w}} \{l(\bar{x}(k), u(k), \bar{w}(k)) + \rho V^*(k+1)\} \tag{12}$$

Built upon (12), we define the so-called Q-function as follows.

$$\begin{aligned} &Q_{u,\bar{w}}(\bar{x}(k), a, d) \\ &= l(\bar{x}(k), a, d) + \sum_{i=k+1}^{\infty} \rho^{i-k} l(\bar{x}(k+1), u(k+1), \bar{w}(k+1)) \\ &= l(\bar{x}(k), a, d) + \rho V(k+1) \\ &= l(\bar{x}(k), a, d) + \rho Q_{u,\bar{w}}(\bar{x}(k+1), u(k+1), \bar{w}(k+1)) \end{aligned} \tag{13}$$

The function $Q_{u,\bar{w}}(x(k), a, d)$ in Eq. (13) is called an action-state value function [25]. It represents the value of the performance metric obtained when the control action a and disturbance d are applied at state $x(k)$, and the control policy u and disturbance policy $\bar{w}$ are pursued thereafter. Based on Eq. (13), we have

$$\begin{aligned} &Q_{u^*,\bar{w}^*}(\bar{x}(k), a, d) = l(\bar{x}(k), a, d) + \rho V^*(k+1) \\ &= l(\bar{x}(k), a, d) + \rho Q_{u^*,\bar{w}^*}(\bar{x}(k+1), u^*(k+1), \bar{w}^*(k+1)) \end{aligned} \tag{14}$$

where the optimal control policy u^* follows Eq. (10) and the worst-case disturbance policy $\bar{w}^*$ is defined by

$$\bar{w}^*(k) = \arg\max_{\bar{w}} V^*(k) \tag{15}$$

Remark 1: By employing preview information as augmented states and organizing the unknown external disturbance and the uncertain future reference as a maximizing player, we successfully reformulate the tracking control problem of the original system (1) to a stationary zero-sum game of the system (7). This reformulation not only addresses the challenge associated with the nonautonomous nature of optimal tracking control problems but also allows us to quantitatively evaluate the influence of the general disturbance vector (i.e. $\bar{w}$) to the system, laying a backbone for the application of our PR-DRL algorithm as introduced in the following subsection.

Remark 2: The fundamental difference between the state value function V in (9) and the action-state value function $Q_{u,\bar{w}}$ in (13) is that the latter allows us to employ the measurements under arbitrary control policy a and disturbance policy d to carry out algorithm training. This feature is essential for the design of PR-DRL because reference signals and disturbances cannot be manipulated in practice. In other words, the employment of Q-function will enable our PR-DRL algorithm to be off-policy and utilize both offline and online data to carry out learning process, rendering enhanced feasibility and flexibility than the on-policy adaptive dynamic programming (ADP) methods (e.g. [26], [27]) for robust control, which require the exact target control & disturbance policies to be applied for data collecting and learning purposes.

B. Preview-Based Robust Deep Reinforcement Learning

We show how to evaluate $Q_{u^*,\bar{w}^*}$, u^* and $\bar{w}^*$ via DRL in this subsection. Our design origins from the deep deterministic policy gradient (DDPG) [22] algorithm, which employs the actor-critic mechanism as the main DRL framework. The critic aims to approximate the optimal Q-function in (14) whilst the actor aims to evaluate the optimal control policy u^*. Furthermore, two sets of actor-critic deep neural networks (DNNs) are employed, named a main actor-critic and a target actor-critic, respectively. We use θ^u, $\theta^{u'}$, θ^Q and $\theta^{Q'}$ to denote the parameters of main actor, target actor, main critic and target critic, respectively. Following [22], we set

$$\theta^{u'} \leftarrow (1-\tau)\theta^{u'} + \tau\theta^u, \quad \theta^{Q'} \leftarrow (1-\tau)\theta^{Q'} + \tau\theta^Q \tag{16}$$

where τ is a small constant. As discussed in [28] and [22], the employment of target DNNs along with such a "soft replacement" strategy in (16) can enhance the learning stability.

Distinct from DDPG and other standard DRL algorithms, a novel DNN strucutre, termed a distractor, is employed in our design to estimate the worst-case disturbance policy, i.e. $\bar{w}^*$. We also employ a main-target DNN pair for our distractor and use $\theta^{\bar{w}}$ and $\theta^{\bar{w}'}$ to denote the parameters of main distractor and target distractor, respectively. Following (16), we set

$$\theta^{\bar{w}'} \leftarrow (1-\tau)\theta^{\bar{w}'} + \tau\theta^{\bar{w}} \tag{17}$$

Before providing the update laws for θ^u, θ^Q and $\theta^{\bar{w}}$, we simply introduce the experience replay strategy as employed by deep Q-network (DQN) [28] and DDPG [22]. In general, experience replay refers to the strategy of sampling a small batch (with size n) of past experience (in terms of transitions) from a memory buffer $\mathcal{M}$ (with size m) at every learning

Algorithm 1 Preview-Based Robust Deep Reinforcement Learning (PR-DRL) Algorithm.

Employ preview information and tracking errors as augmented system states and transform the whole tracking control problem to a stationary zero-sum game as described by (7) and (9).
Initialize DNN parameters θ^u, $\theta^{u'}$, θ^Q, $\theta^{Q'}$, $\theta^{\bar{w}}$, $\theta^{\bar{w}'}$ and other user-defined parameters.
Decide termination conditions or learning steps (i.e. K and H at here).

1: **for** each episode **do**
2: **for** $k = 0$ to K **do**
3: Given current state $\bar{x}(k)$, choose the control action $a = u(\bar{x}(k)|\theta^u) + \epsilon(k)$, where $\epsilon(k)$ denotes an exploration noise.
4: Apply a to the system, and observe $\bar{x}(k+1)$, d, and $l(\bar{x}(k), a, d)$.
5: Store the transition $(\bar{x}(k), \bar{x}(k+1), l(\bar{x}(k), a, d), a, d)$ in the memory buffer $\mathcal{M}$.
6: **for** $h = 0$ to H **do**
7: Sample a mini batch $\mathcal{B}$ from $\mathcal{M}$, which contains n transitions, i.e., $\{(\bar{x}_i, \bar{x}_i^+, l_i, a_i, d_i)\}$, $i = 1, 2, ..., n$.
8: Update θ^Q for the main critic by minimizing the loss function L constructed by the TD-error in (18).
9: Update $\theta^{\bar{w}}$ for the main distrator via the policy gradient strategy as described in (20).
10: **end for**
11: Update θ^u for the main actor via the policy gradient strategy as described in (19).
12: Update the target networks' parameters, i.e. $\theta^{u'}$, $\theta^{Q'}$ and $\theta^{\bar{w}'}$, by the soft replacement strategy as in (16) and (17).
13: **end for**
14: **end for**

step to carry out DNN training. This design caters to the independent and identical distribution requirement in DNN training and therefore enhance the learning stability. We denote the transitions in a sampled batch by $\{(\bar{x}_i, \bar{x}_i^+, l_i, u_i, d_i)\}$, $i = 1, 2, ...n$. Here $\bar{x}_i$ and $\bar{x}_i^+$ denote a state and its successor (i.e. $\bar{x}(k)$ and $\bar{x}(k+1)$), respectively, and l_i, a_i, d_i denote the step reward, control input, and disturbance, respectively.

Then we are ready to propose the training laws of main DNNs. First, the main critic is trained by the so-called temporal difference error (TD-error). The TD-error is constructed by the essential feature of Q-function as given in (13) and (14). For a transition $(\bar{x}_i, \bar{x}_i^+, l_i, a_i, d_i)$, its TD-error is defined by

$$\begin{aligned}\delta_i =& l_i + \rho Q'(\bar{x}_i^+, u'(\bar{x}_i^+|\theta^{u'}), \bar{w}'(\bar{x}_i^+|\theta^{\bar{w}'})|\theta^{Q'}) \\ & - Q(\bar{x}_i, a_i, d_i|\theta^Q)\end{aligned} \tag{18}$$

Here Q, Q', $u'(\bar{x}_i^+|\theta^{u'})$, and $\bar{w}'(\bar{x}_i^+|\theta^{\bar{w}'})$ are the outputs of main critic, target critic, target actor, and target distractor, respectively. Based on (18), at every training step, we aim to update θ^Q such that the following loss function of the sampled batch can be minimized: $L = (1/n)\sum_{i=1}^n \delta_i^2$.

Based on Eqs. (10) and (15), the main actor aims to minimize the Q-function whilst the main distractor aims to maximize the Q-function. Their training can be carried out by the policy gradient strategy. Particularly, for a sampled training batch, the gradients of θ^u and $\theta^{\bar{w}}$ with respect to $Q(\bar{x}_i, u(\bar{x}_i|\theta^u), \bar{w}(\bar{x}_i|\theta^{\bar{w}}))$ are given as follows.

$$\begin{aligned}\nabla_{\theta^u} =& \frac{1}{n}\sum_{i=1}^n \frac{\partial Q(\bar{x}_i, u(\bar{x}_i|\theta^u), \bar{w}(\bar{x}_i|\theta^{\bar{w}})|\theta^Q)}{\partial \theta^u} \\ =& \frac{1}{n}\sum_{i=1}^n [\nabla_u Q(\bar{x}_i, u(\bar{x}_i|\theta^u), \bar{w}(\bar{x}_i|\theta^{\bar{w}})|\theta^Q)\nabla_{\theta^u} u(\bar{x}_i|\theta^u)]\end{aligned} \tag{19}$$

$$\begin{aligned}\nabla_{\theta^{\bar{w}}} =& \frac{1}{n}\sum_{i=1}^n \frac{\partial Q(\bar{x}_i, u(\bar{x}_i|\theta^u), \bar{w}(\bar{x}_i|\theta^{\bar{w}})|\theta^Q)}{\partial \theta^{\bar{w}}} \\ =& \frac{1}{n}\sum_{i=1}^n [\nabla_{\bar{w}} Q(\bar{x}_i, u(\bar{x}_i|\theta^u), \bar{w}(\bar{x}_i|\theta^{\bar{w}})|\theta^Q)\nabla_{\theta^{\bar{w}}} w(\bar{x}_i|\theta^{\bar{w}})]\end{aligned} \tag{20}$$

The update of θ^u and $\theta^{\bar{w}}$ should be driven by $-\nabla_{\theta^u}$ and $\nabla_{\theta^{\bar{w}}}$, respectively. Based on these preliminaries, the proposed PR-DRL method is organized in Algorithm 1.

Remark 3: As shown in Algorithm 1, our PR-DRL method has two major learning loops. The inner loop (lines 6-10) updates the main critic and main distractor under an unchanged main actor. This allows the distractor to evaluate the worst-case $\bar{w}(\bar{x}|\theta^{\bar{w}})$ under a fixed control policy $u(\bar{x}|\theta^u)$. After that, the outer loop (lines 11-12) updates the main actor under the resulting critic and distractor, aiming to search an optimal $u(\bar{x}|\theta^u)$ with respect to $\bar{w}(\bar{x}|\theta^{\bar{w}})$ and $Q(\bar{x}_i, u(\bar{x}_i|\theta^u), \bar{w}(\bar{x}_i|\theta^{\bar{w}})|\theta^Q)$. It also updates the parameters of target networks. Integrating these two learning loops together allows us to iteratively approximate Q^*, u^* and $\bar{w}^*$.

III. Wind-Farm Power Tracking via PR-DRL

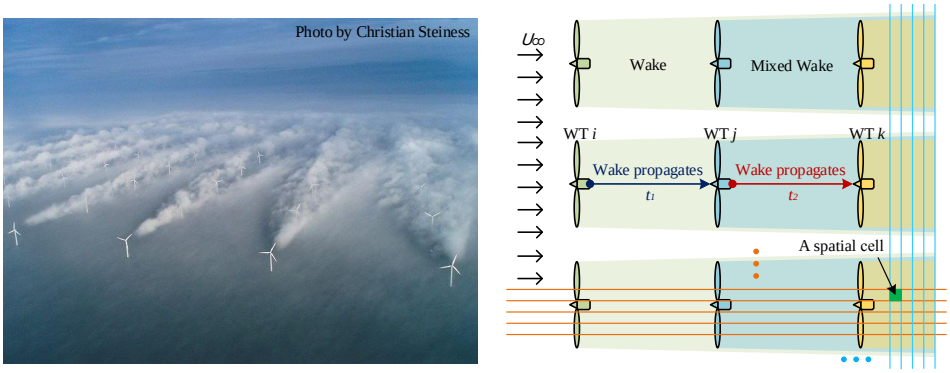
$\mathcal{WT}_h$ denotes a single wind turbine in a wind farm, with $h = 1, 2, 3..., q$ and here q is the total turbine number. The power output (denoted by E_h) of $\mathcal{WT}_h$ is a function of its thrust coefficient (denoted by C'_{T_h}), its yaw offset (denoted by α_h, with respect to inflow wind direction) and the wind speed at its rotor (denoted by U_h), formulized by

$$E_h(k) = \phi(C'_{\mathrm{T}_h}(k), \alpha_h(k), U_h(k)) \tag{21}$$

where $\phi(\cdot)$ is an unknown function from $C'_{\mathrm{T}_h}(k)$, $\alpha_h(k)$, and $U_h(k)$ to $E_h(k)$, and here k is the time index.

Therefore, the whole farm's total power generation satisfies

$$E_T(k) = \sum_{h=1}^q E_h(k) = \sum_{h=1}^q \phi(C'_{\mathrm{T}_h}(k), \alpha_h(k), U_h(k)) \tag{22}$$



(a) Denmark Horns Rev Offshore Wind Farm

(b) An Illustration of Offshore Wind Farm in a Flow Field

Figure 1: Illustrations of wind farm, flow field, and wake effect

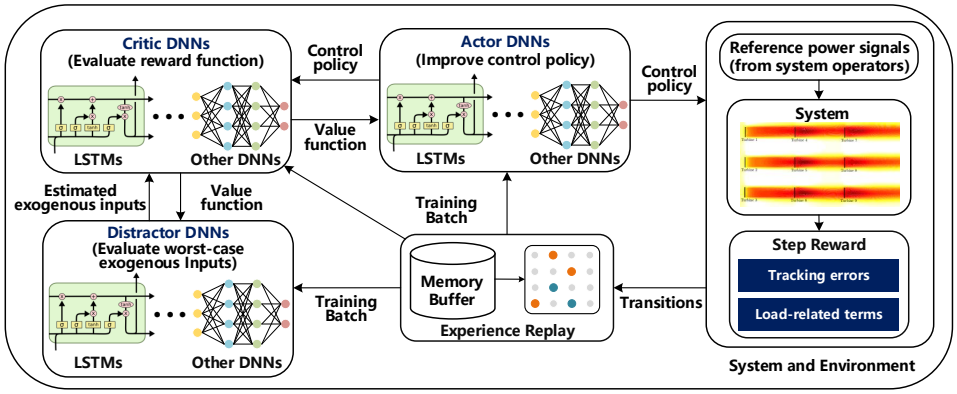


Figure 2: The main framework and data flow of the preview-based robust DRL method.

We consider a time-varying reference power signal in a near future, denoted by $\{E_R(k), E_R(k+1), ..., E_R(k+N-1)\}$. In ancillary services such as frequency regulation, SO requires the farm's total power generation to track $E_R(k)$ at every time k, i.e. minimizing the following tracking error:

$$E(k) = |E_T(k) - E_R(k)| \tag{23}$$

The controllable variables in such a power tracking task are the thrust coefficient (i.e. C'_{T_h}, which is directly related to the rotor torque and blade bitch angle [29]) and the yaw offset (i.e. α_h) of every turbine in the farm. However, since yaw actuators usually have large time constants (i.e. long response time) and yaw offsets can lead to large structural loads, we only employ turbines' thrust coefficients as control inputs and keep their yaw angles to be zero in power tracking. But in a case study, we will demonstrate how to combine yaw control strategies with PR-DRL for capacity enhancement.

We illustrate a typical wind farm in Fig. 1.a (Denmark Horns Rev Offshore Wind Farm, photo by Christian Steiness). This figure demonstrates the aerodynamic interactions among turbines. It shows that the wakes induced by the upstream turbines can lead to a complicated time-delayed influence (as wakes propagate) on downstream turbines and render the whole farm's power outputs difficult to control. The wind speed U_h at the rotor of $\mathcal{WT}_h$ is not only influenced by the free stream wind speed (denoted by U_∞) but also affected by the control sequences of all its upstream turbines in a past period of time. Such a complicated aerodynamic interactions, along with time-delay and stochastic features, is very difficult to be accurately and analytically modelled. Some recent studies employed MPC method [15], [16] to achieve wind-farm power tracking. These elegant results directly employ the discretized model in large-eddy simulation (LES) and also the full-flow states to carry out controller design. We illustrate that in Fig. 1.b (a vertical view of a nine-turbine wind farm). Particularly, a staggered grid is employed to discretize the whole flow field in LES. At every time step, the wind conditions in every cell of the staggered grid are employed by the MPC method proposed in [15], [16], allowing the controller capture the whole-flow-field information and predict future power generations. However, the use of such tens of thousands of full-state-style measurements is infeasible for practical wind farms, in addition to the drawbacks inherent in the underlying model-based structures of MPC.

In this paper, we aim to carry out a pioneering study to achieve model-free and data-driven power tracking for wind farms by employing only the available measurements at turbine rotors, including U_h and E_h. However, simply employing real-time, instantaneous measurements will render the whole problem to be partially-observable and non-Markovian. This fact is illustrated by Fig. 1.b. The control action of an upstream turbine $\mathcal{WT}_i$ will change the wake behind it. Such a change needs to propagate for a period of time (e.g. t_1 in Fig. 1.b) before it leads to direct influence on the power generation of

a downstream turbine $\mathcal{WT}_j$. Therefore, not only the instantaneous measurements but also the measurements during a past period of time (denoted by t_b) should be employed to allow the PR-DRL capture the key task-relevant information and achieve real-time power tracking. This looking-back time t_b should be longer enough to cover potential wake-propagation time of all upstream-downstream turbine pairs, i.e. $t_b = \max\{t_1, t_2, ...\}$.

We are ready to summarize how to mould the wind-farm power tracking control problem into the proposed PR-DRL:

(a) Following the design in the section II, the augmented state vector $\bar{x}(k)$ contains 1) the power outputs of every turbine in the farm during $[k-t_b, k]$, i.e. $[E_h(k-t_b), E_h(k-t_b+1), ..., E_h(k)]$, $h = 1, 2, ..., q$; 2) the wind speeds measured at turbine rotors during $[k-t_b-1, k-1]$, i.e. $[U_h(k-t_b-1), U_h(k-t_b), ..., U_h(k-1)]$, $h = 1, 2, ..., q$; 3) the farm-level power tracking errors during $[k-t_b, k]$, i.e. $[E(k-t_b), E(k-t_b+1), ..., E(k)]$; 4) the changes of free-stream wind speeds U_∞ (w.r.t the expected nominal wind speed) during $[k-t_b-1, k-1]$, denoted by $[\triangle U_\infty(k-t_b-1), \triangle U_\infty(k-t_b), ..., \triangle U_\infty(k-1)]$; 4) the reference signals during $[k-t_b, k]$, i.e. $[E_R(k-t_b), E_R(k-t_b+1), ..., E_R(k)]$; 5) the preview information of reference signals, i.e. $\bar{x}_d(k) = [E_R(k), E_R(k+1), ..., E_R(k+N-1)]$; 6) the thrust coefficient of all the turbines during $[k-t_b, k]$, i.e. $[C'_{T_h}(k-t_b), C'_{T_h}(k-t_b+1), ..., C'_{T_h}(k)]$, $h = 1, 2, ..., q$.

(b) The general disturbance vector (i.e. $\bar{w}(k)$) to be evaluated by the distractor is constructed by: 1) the unknown change of future reference power signal, i.e. $E_R(k+N) - E_R(k+N-1)$; 2) the unknown change of free-stream wind speed, i.e. $\triangle U_\infty(k)$.

(c) The control input $u(k)$ contains the changes of C'_{T_h} of all the turbines at time k, denoted by $\triangle C'_{T_h}(k)$.

(d) Following (8) and (9), the step reward $l(k)$ satisfies

$$l(k) = a_1 E(k)^2 - a_2[E_R(k+N) - E_R(k+N-1)]^2 - a_3 \triangle U_\infty(k)^2 + a_4 \sum_{h=1}^{q} [C'_{T_h}(k) - C'_{T_h}(k-1)]^2 \quad (24)$$

The first term in (24) represents the tracking objective (i.e. $E(k) \to 0$). The second and third terms quantify the disturbance attenuation requirement. The last term in (24) is a load-related term adapted from [16]. This term evaluates the change of thrust applied to the turbine rotor, helping make a balance between power tracking and load mitigation. And a_1, a_2, a_3 and a_4 are weighting constants.

(e) To effectively handle time-series information, LSTM networks are employed in the PR-DRL. Fig. 2 shows the main structures of our PR-DRL for wind-farm power tracking.

IV. Case Studies

We employ a dynamic wind farm simulator (WFSim) developed in [29] to carry out case studies, which simulates flow fields and wind farms via the 2D Navier-Stokes equations. We consider a flow field with a size of 2518.8m×1558.4m . The change of U_∞ follows an Ornstein-Uhlenbeck process with a mean wind speed of 10 m/s. A wind profile under such a condition is illustrated in Fig. 3. Following the design in the sections II and III, the specific DNN structures (including neuron types and numbers) of the actor, critic and distractor of the proposed PR-DRL in case studies are illustrated in Fig. 4. Both the actor and the distrator have five layers, and the critic employs a six-layer DNN structure. LSTM networks are utilized to handle time-series data, and dropout layers are employed to avoid the vanishing gradient problem. We set $\gamma = 1$, $\rho = 0.99$ and $\tau = 0.05$. At each training step, 64 transitions (i.e. $n = 64$) are randomly sampled from the memory buffer $\mathcal{M}$ (with a total size of 50000). In addition, 10000 offline transitions generated by WFSim are fed into $\mathcal{M}$ at the beginning of the training. This offline dataset is also employed to normalize states, disturbances, control inputs, and step rewards by using the z-score method, making the learning process more effective and stable. After normalization, we set $a_1 = 1$, $a_2 = 0.2$, $a_3 = 0.2$ and $a_4 = 0.1$ to make a balance among power tracking, disturbance attenuation and load mitigation requirements. The minimum and maximum values of C'_{T_h} are set to be 0.1 and 2, respectively. In addition, the maximum step change of C'_{T_h} is 0.1.

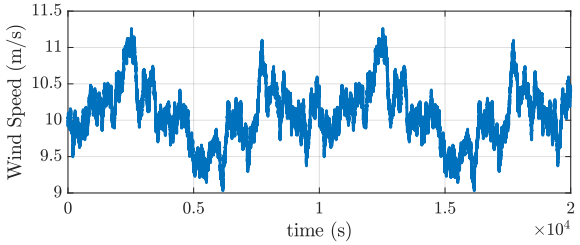


Figure 3: Illustration of an Ornstein-Uhlenbeck wind profile.

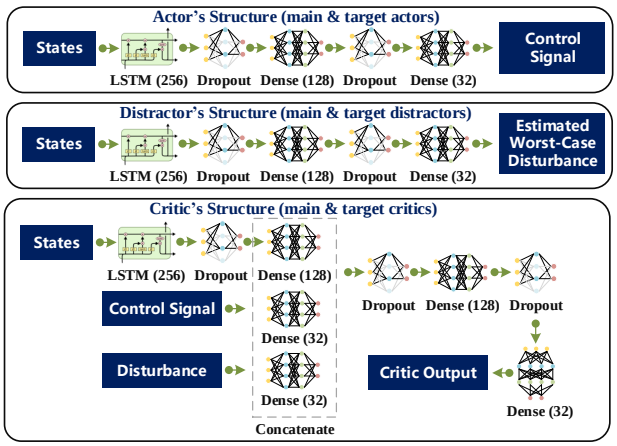


Figure 4: DNN structures and neuron numbers for case studies.

For comparison purposes, another two controllers for wind farm power tracking problems are also employed in case studies. They are:

(a) The MPC controller proposed in [15], [16] (referred to as MPC in simulation results). This advanced wind farm control method has strong optimizing abilities. Moreover, it is highly compatible with the dynamic wind farm simulator employed in this section (i.e. WFSim) – it can directly employ the inherent mathematical model of WFSim to carry out controller design, and therefore it is not influenced by modelling errors

and uncertainties in case studies. These important features enable the MPC method proposed in [15], [16] to be an excellent baseline to carry out performance comparison with the PR-DRL method proposed in this paper.

(b) The distributed wind farm control method proposed in [13] (referred to as DWFC in simulation results). DWFC is another popular method for wind-farm power tracking. Its core idea is to distribute the power tracking task of the entire wind farm to a series of power tracking tasks of each turbine in the farm. To pursue better performance, the reference signal for every turbine in the farm is continuously changed based on the instantaneous & immediate-future power estimations for the corresponding turbines.

The state & control constraints of MPC and DWFC are set to be the same with PR-DRL for a fair comparison. Moreover, the power outputs in which all turbines work in the maximum power point tracking (MPPT) mode are also illustrated in case studies. Such a working condition is commonly referred to as the greedy mode for wind farms [1]-[7], in which every turbine aims to capture the possible maximum wind power based on its local wind condition.

A. Case Study with a Prototypical Wind Farm

In this subsection, we consider a 3×3 wind farm consisting of NREL 5MW wind turbines. The distances between two wind turbines in the x-direction and the y-direction are 5D and 3D, respectively, and here $D = 126.4$m is the diameter of turbines. Following [13], [15], [16], the farm is required to track a RegD-style signal defined by: $E_R = E_{greedy}(0.8+0.3s_{regD})$. Here E_{greedy} denotes the farm's total power output under the greedy mode with the expected mean free-stream wind speed (10m/s); s_{regD} is a normalized RegD signal (within ± 1), which is one the most irregular reference signals in anciliary services [13], [15], [16]. The preview step (looking-ahead step of reference) is $N = 100$.

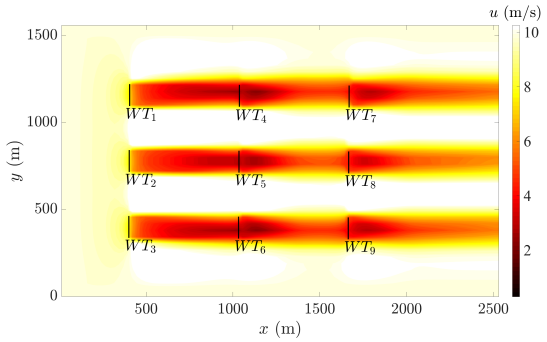


Figure 5: Illustration of wind farm and flow field (Case A).

The time step of WFsim is set to be 1s. Based on all these setting, we choose $t_b = 100$s for our PR-DRL. This means, at every time step, the measurements during the last 100s are fed into PR-DRL, allowing the algorithm to capture the key system information and achieve power tracking. After the training, we test the performance of our PR-DRL with the RegD-style signal E_R mentioned above under time-varying wind speeds. An instantaneous flow field (vertical view) is shown in Fig. 5 to

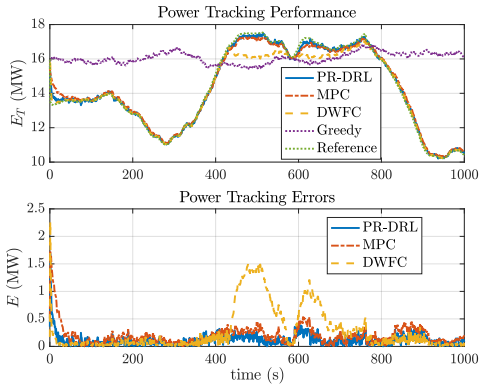


Figure 6: Wind-farm power tracking performance (Case A).

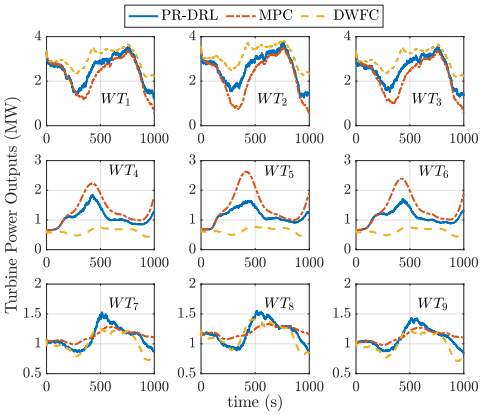


Figure 7: Turbine power outputs (Case A).

illustrate the wind farm considered here and the aerodynamic interactions among turbines (wind speed differences among the whole flow field are indicated by color differences). The power tracking results under PR-DRL, MPC and DWFC are shown in Fig. 6. The wind farm power outputs of the greedy mode under the same wind speed profile are also illustrated in this figure. Moreover, the power outputs of every turbine in the farm under different controllers are shown in Fig. 7. These figures show that though DWFC has a good performance when the reference output is lower than the greedy mode, it lacks the ability to track a reference signal that is higher than the greedy mode (420s-760s in this case study) and lead to significant tracking errors. This limitation comes from the inherent design principle of DWFC. To be specific, it only evaluates the turbine power generations at the instantaneous time step and the immediate future, and then uses such information to carry out power distribution and set reference signals for every turbine in the farm. Therefore, DWFC is "short-sighted" and lacks the ability to carry out long-term planning and track "non-trivial" references (e.g., a reference signal that is higher than the greedy mode output). In contrast, PR-DRL and MPC have better capacities and achieve high-performance power tracking under uncertain wind conditions in this case study.

B. Case Study with a Different Wind Farm Layout

In this subsection, we test the PR-DRL's performance with a wind farm that has a different layout from case study A.

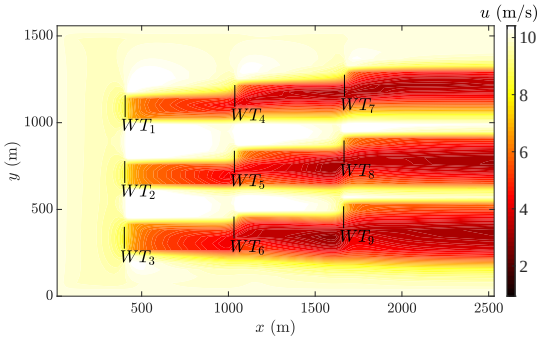


Figure 8: Illustration of wind farm and flow field (Case B).

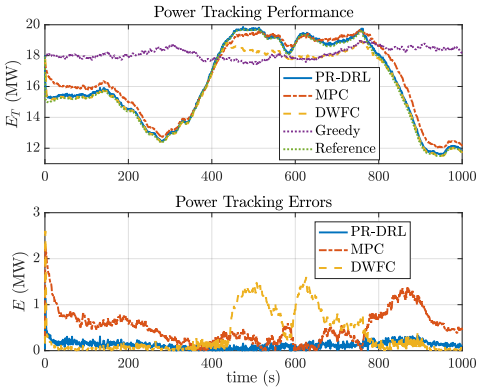


Figure 9: Wind-farm power tracking performance (Case B).

Particularly, we translate each row of turbines 0.5D in the y-direction. Simulation results are given in Figs. 8, 9 and 10. One can see that, under the situation that downstream turbines are influenced by partial wakes, the PR-DRL method proposed in this paper still has the ability to understand the system's main mechanism and accomplish power tracking. It also leads to a minimum averaged tracking error among the three controllers. These results demonstrate the adaptability of our method.

C. Case Study Considering Yaw Control Strategy

As discussed in the section III, though we employ thrust coefficients as the main control inputs (i.e. carrying out induction control), our method has the ability to cooperate with yaw control strategies to enhance the capacity. In this case study, we employ the Bayesian ascent (BA) method proposed in [3] for yaw optimization. BA is an advanced sequential searching method that is built upon Bayesian optimization. It allows us to search the optimal yaw offsets and therefore steer wakes and enlarge the whole farm's potential maximum power outputs. To avoid big structural loads, we set the yaw offset searching range to be $\alpha_h \in [-30°, 30°]$, $h = 1, 2, ..., 9$. The resulting optimal yaw angles by employing BA is $\alpha = [23.58, 17.77, 19.93, 19.40, 12.77, 15.41, 0, 0, 0]$ deg, and here $\alpha = [\alpha_1, \alpha_2, ..., \alpha_9]$. These yaw offsets are applied to both PR-DRL and MPC. An illustration of the flow field and the wind farm with these yaw offsets are given in Fig. 11. DWFC keeps the turbine yaw angles to be zero based on its design principle. In this case study, we consider a new power reference: $E_R = E_{greedy}(0.8 + 0.6s_{regD})$. The simulation results with this new power reference are shown in Figs. 11-14. Particularly, the power outputs and tracking errors under different controllers are given in Fig. 12. One can see that DWFC fails to track this new power reference (which can be occasionally much higher than the power generation under the greedy mode) and leads to large tracking errors during the period of 420s-850s. In comparison, PR-DRL and MPC cooperate well with the yaw control strategy and successfully achieve power tracking in this case study. The power outputs and the changes of C'_T of every turbine in the farm are given in Figs. 13 and 14, respectively.

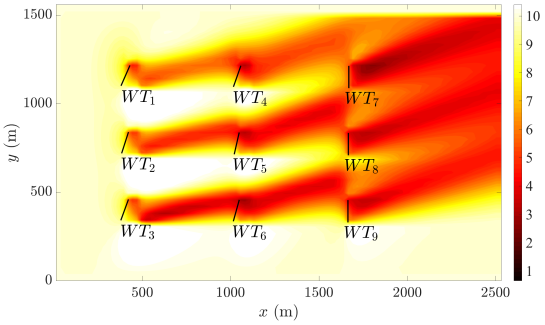


Figure 11: Illustration of wind farm and flow field (Case C).

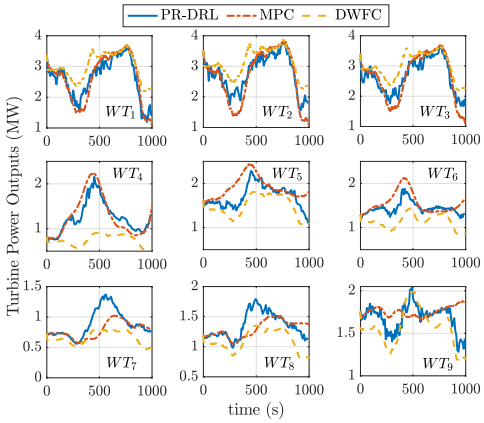


Figure 10: Turbine power outputs (Case B).

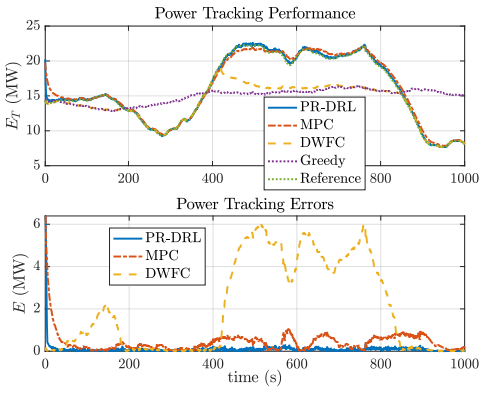


Figure 12: Wind-farm power tracking performance (Case C).

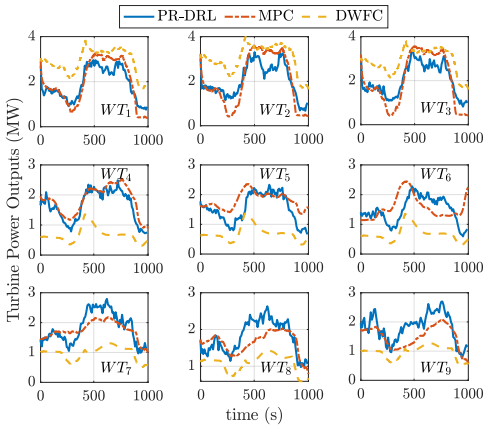


Figure 13: Turbine power outputs (Case C).

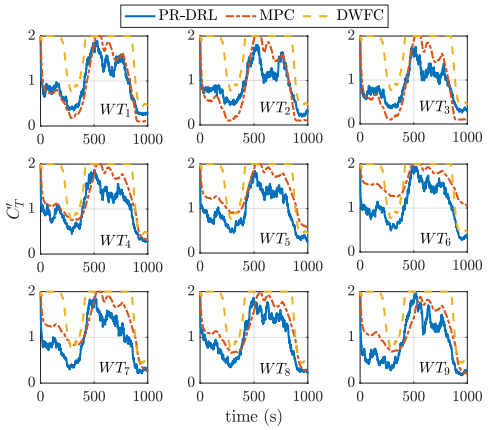


Figure 14: Changes of thrust coefficients (C_T') in Case C.

In addition, we summarize the root mean square (RMS) errors of all case studies in Table I. It is clear that PR-DRL has the minimum RMS errors in all the three case studies, showing its effectiveness, adaptability and robustness for wind-farm power tracking tasks.

Table I: RMS errors under different controllers (MW).

Case	PR-DRL	MPC [15], [16]	DWFC [13]
A	0.214	0.315	0.485
B	0.248	0.559	0.598
C	0.185	0.623	2.980

Remark 4: Simulation results in this section show the key features of PR-DRL, MPC [15], [16] and DWFC [13]. Particularly, though DWFC is easy to implement, it has troubles in handling the reference signals that are higher than the greedy mode power outputs, leading to large tracking errors under such circumstances. In contrast, PR-DRL and MPC have the ability to carry out long-term planning, leading to enlarged ranges of trackable power. They show stronger capacities and better tracking performance than DWFC. It is noteworthy that our PR-DRL and the MPC method in [15], [16] have quite different features from each other. As discussed in Sec. III and illustrated in Fig. 1.b, MPC [15], [16] utilizes the wind conditions in every staggered-grid cell of the whole flow field to calculate control signals. However, the use of such tens of thousands of full-state-style measurements is infeasible in practice (e.g., a 100×42 staggered grid with over 10000 states are employed by MPC in case studies). Moreover, as a model-based method, MPC suffers from inevitable modelling errors and uncertainties in practical uses. In contrast, our PR-DRL is model-free and data-driven. It only employs the measurements at turbine rotors (which are easy to obtain) rather than the whole flow field and does not rely on any underlying analytical models.

Remark 5: Using a stand-alone turbine to achieve power tracking commonly needs to de-rate the turbine from the MPPT mode [9], [10], [11]. This leads to revenue loss due to the power generation reduction but brings ancillary-service income from the market, and an economic trade-off between these two aspects should be considered. A recent study [30] (which employed commercially available wind turbines to fulfill power tracking tasks and providing ancillary services) showed that the income from the regulation market by power tracking fully has the potential to be greater than the induced energy loss. Such a potential can be further enhanced when a wind farm is employed rather than a stand-alone turbine. Due to the strong aerodynamic couplings, de-rating upstream turbines can mitigate wake effects and lead to the power increase of the downstream turbines or even the whole wind farm. Many studies [1]-[7] have shown that the greedy mode (i.e. all turbines in the farm working in the MPPT mode) cannot maximize the whole farm's power generation. These facts are also verified by the case studies in this section. As shown in Figs. 6, 9 and 12, our method can track reference signals that are occasionally higher than the greedy mode power generations in all case studies, particularly in Case C (in which even the averaged power output during the whole simulation time span under our PR-DRL is greater than the greedy mode). These facts enhance the feasibility and economic profitability of using wind farms to provide ancillary services such as SFC/AGC. But it should be emphasized that farm-level power tracking still inevitably results in energy loss with respect to the potentially maximum power generation that a farm can obtain (even though the trackable power range can be enlarged with respect to the greedy mode), and a future in-depth economic analysis should be considered to maximize the profit of wind-farm power tracking methods in practical applications.

V. Conclusions

A new wind-farm power tracking control method was proposed in this paper via deep reinforcement learning (DRL). To the best of the authors' knowledge, this is for the first time that a data-driven model-free solution was designed for such problems. Built upon the H_∞ control theory, we developed a novel preview-based robust DRL (PR-DRL) method to handle preview information and achieve tracking control goals. LSTM networks and other deep neural network structures were embedded in PR-DRL to handle complex tasks with time-series measurements and partial-observable properties. Case studies

with a dynamic wind farm simulator showed that our PR-DRL could excellently accomplish wind-farm power tracking tasks under uncertain environments and strong aerodynamic interactions among turbines. It could adapt to different wind farm layouts and be integrated with yaw optimization strategies for capacity enhancement.

REFERENCES

[1] P. Hulsman, S. J. Andersen, and T. Göçmen, "Optimizing wind farm control through wake steering using surrogate models based on high-fidelity simulations," *Wind Energy Science*, vol. 5, no. 1, pp. 309–329, 2020.

[2] B. Dou, T. Qu, L. Lei, and P. Zeng, "Optimization of wind turbine yaw angles in a wind farm using a three-dimensional yawed wake model," *Energy*, vol. 209, Paper ID 118415, 2020.

[3] J. Park and K. H. Law, "Bayesian ascent: A data-driven optimization scheme for real-time control with application to wind farm power maximization," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1655–1668, 2016.

[4] U. Ciri, M. A. Rotea, and S. Leonardi, "Model-free control of wind farms: A comparative study between individual and coordinated extremum seeking," *Renewable Energy*, vol. 113, pp. 1033–1045, 2017.

[5] P. Stanfel, K. Johnson, C. J. Bay, and J. King, "A distributed reinforcement learning yaw control approach for wind farm energy capture maximization," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 4065–4070.

[6] A. Saenz-Aguirre, E. Zulueta, U. Fernandez-Gamiz, J. Lozano, and J. M. Lopez-Guede, "Artificial neural network based reinforcement learning for wind turbine yaw control," *Energies*, vol. 12, no. 3, p. 436, 2019.

[7] H. Zhao, J. Zhao, J. Qiu, G. Liang, and Z. Y. Dong, "Cooperative wind farm control with deep reinforcement learning and knowledge assisted learning," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 6912–6921, 2020.

[8] H. Dong, J. Zhang, and X. Zhao, "Intelligent wind farm control via deep reinforcement learning and high-fidelity simulations," *Applied Energy*, vol. 292, p. 116928, 2021.

[9] C. R. Shapiro, P. Bauweraerts, J. Meyers, C. Meneveau, and D. F. Gayme, "Model-based receding horizon control of wind farms for secondary frequency regulation," *Wind Energy*, vol. 20, no. 7, pp. 1261–1275, 2017.

[10] P. Fleming, J. Aho, P. Gebraad, L. Pao, and Y. Zhang, "Computational fluid dynamics simulation study of active power control in wind plants," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 1413–1420.

[11] J. Aho, L. Y. Pao, P. Fleming, and E. Ela, "Controlling wind turbines for secondary frequency regulation: an analysis of AGC capabilities under new performance based compensation policy," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2015.

[12] H. Zhao, Q. Wu, Q. Guo, H. Sun, and Y. Xue, "Distributed model predictive control of a wind farm for optimal active power control part I: Clustering-based wind turbine model linearization," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 3, pp. 831–839, 2015.

[13] J.-W. van Wingerden, L. Pao, J. Aho, and P. Fleming, "Active power control of waked wind farms," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4484–4491, 2017.

[14] C. J. Bay, J. Annoni, T. Taylor, L. Pao, and K. Johnson, "Active power control for wind farms using distributed model predictive control and nearest neighbor communication," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 682–687.

[15] S. Boersma, B. M. Doekemeijer, T. Keviczky, and J. van Wingerdenl, "Stochastic model predictive control: uncertainty impact on wind farm power tracking," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4167–4172.

[16] S. Boersma, B. Doekemeijer, S. Siniscalchi-Minna, and J. van Wingerden, "A constrained wind farm controller providing secondary frequency regulation: An LES study," *Renewable Energy*, vol. 134, pp. 639–652, 2019.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[18] W. Wu, P. Yang, W. Zhang, C. Zhou, and S. Shen, "Accuracy-guaranteed collaborative DNN inference in industrial IoT via deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, 2020, Early Access.

[19] H. Dong, X. Zhao, and H. Yang, "Reinforcement learning-based approximate optimal control for attitude reorientation under state constraints," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1664–1673, 2021.

[20] C. Chen, M. Cui, F. F. Li, S. Yin, and X. Wang, "Model-free emergency frequency control based on reinforcement learning," *IEEE Transactions on Industrial Informatics*, 2020, Early Access.

[21] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[22] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *in International Conference on Machine Learning*, 2016.

[23] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," *In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018.

[24] B. Luo, Y. Yang, and D. Liu, "Policy iteration Q-learning for data-based two-player zero-sum game of linear discrete-time systems," *IEEE Transactions on Cybernetics*, 2020, Early Access.

[25] B. Luo, D. Liu, T. Huang, and D. Wang, "Model-free optimal tracking control via critic-only Q-learning," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 10, pp. 2134–2144, 2016.

[26] Y. Zhu, D. Zhao, X. Yang, and Q. Zhang, "Policy iteration for H_∞ optimal control of polynomial nonlinear systems via sum of squares programming," *IEEE Transactions on Cybernetics*, vol. 48, no. 2, pp. 500–509, 2017.

[27] J. Hou, D. Wang, D. Liu, and Y. Zhang, "Model-free H_∞ optimal tracking control of constrained nonlinear systems via an iterative adaptive learning algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[29] S. Boersma, B. Doekemeijer, M. Vali, J. Meyers, and J.-W. van Wingerden, "A control-oriented dynamic wind farm model: WFSim," *Wind Energy Science*, vol. 3, no. 1, pp. 75–95, 2018.

[30] E. Rebello, D. Watson, and M. Rodgers, "Ancillary services from wind turbines: automatic generation control (AGC) from a single type 4 turbine," *Wind Energy Science*, vol. 5, no. 1, pp. 225–236, 2020.

Hongyang Dong is currently a Research Fellow in Machine Learning and Intelligent Control at the School of Engineering, University of Warwick, Coventry, UK. He obtained his Ph.D. degree in Control Science and Engineering from Harbin Institute of Technology, Harbin, China, in 2018. His current research interests include reinforcement learning, deep learning, intelligent control, and adaptive control.

Xiaowei Zhao is Professor of Control Engineering and an EPSRC Fellow at the School of Engineering, University of Warwick, Coventry, UK. He obtained the PhD degree in Control Theory from Imperial College London in 2010. After that he worked as a postdoctoral researcher at the University of Oxford for three years before joining Warwick in 2013. His main research areas are control theory with applications on offshore renewable energy systems, local smart energy systems, and autonomous systems.