Data-Driven Wind Farm Control via Multi-Player Deep Reinforcement Learning

Hongyang Dong and Xiaowei Zhao

***Abstract*—This brief paper proposes a novel data-driven control scheme to maximize the total power output of wind farms subject to strong aerodynamic interactions among wind turbines. The proposed method is model-free and has strong robustness, adaptability and applicability. Particularly, distinct from state-of-the-art data-driven wind farm control methods that commonly employ the steady-state or time-averaged data (such as turbines' power outputs under steady wind conditions or from steady-state models) to carry out learning, the proposed method directly mines in-depth the time-series data measured at turbine rotors under time-varying wind conditions to achieve farm-level power maximization. The control scheme is built on a novel multi-player deep reinforcement learning method (MPDRL), in which a special critic-actor-distractor structure, along with deep neural networks (DNNs), is designed to handle the stochastic feature of wind speeds and learn optimal control policies subject to a user-defined performance metric. The effectiveness, robustness and scalability of the proposed MPDRL-based wind farm control method are tested by prototypical case studies with a dynamic wind farm simulator. Compared with the commonly employed greedy strategy, the proposed method leads to clear increases in farm-level power generation in case studies.**

***Index Terms*—Wind energy; wind farm control; wind turbine control; reinforcement learning; machine learning.**

I. Introduction

As one of the most efficient forms of green energy, wind power plays a key role in the global effort towards net-zero emissions. Particularly, 15 large offshore wind farms were put into operation in 2020, with an average capacity of 347MW [1]. Though single turbine's control strategies have been widely studied, directly employing these methods to control every turbine in a wind farm via a non-cooperative way can lead to significantly degraded operating efficiency. A lot of studies [2], [3], [4] have demonstrated that the greedy strategy (i.e. every turbine in the farm aims to maximize its own power outputs) is not the optimal control strategy for farm-level power generation maximization. This phenomenon is caused by the aerodynamic couplings among turbines – the wakes induced by upstream turbines can severely influence the power generation of downstream turbines, which is commonly mentioned as the wake effect in the literature. Therefore, farm-level control strategies should be considered to operate all turbines cooperatively in order to mitigate wake effects and increase the whole farm's power generation.

This work was funded by the UK Engineering and Physical Sciences Research Council (grant numbers: EP/R007470/1 and EP/S000747/1).

H. Dong and X. Zhao (corresponding author) are with the Intelligent Control and 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.

Compared with single turbine cases, the main challenges of farm-level control tasks come from the complex aerodynamics and stochastic properties of wakes. On the one hand, wakes are hard to be accurately modelled, bringing barriers to controller design. On the other hand, control-induced wake change has time-delayed features – it typically requires tens/hundreds of seconds to propagate from upstream turbines to downstream turbines, resulting in difficulties in evaluating control performance, especially from a relatively long-term standpoint. Many studies employed simplified wake models [2], [5], [6] to evaluate steady-state wake locations and estimate the effective flow velocities and power outputs at each turbine. Then these estimations were utilized to decide optimal control actions. For example, a famous steady-state parametric model, referred to as FLORIS, was proposed in [2] to achieve yaw setting optimization. Though such designs are easy to implement, their performance can be severely influenced by the model accuracy, and they have difficulties in achieving closed-loop wind farm control under uncertain environments [3]. Some studies employed dynamic models that typically have higher fidelity than steady-state models to develop wind farm control methods and achieve better control performance. For example, a model predictive control (MPC) method was proposed in [7] based on a control-oriented dynamic wind farm model developed in [8]. This meaningful method was shown to be effective in increasing the whole farm's power generation. A small limitation is that it employed a relatively large number of states (tens of thousands or even hundreds of thousands) in receding-horizon control. As an extension study, Ref. [9] reduced the number of system states via deep auto-encoder and carried out MPC based on the reduced-order model. However, those elegant MPC methods still rely on the accuracy of underlying wind flow & wind farm models, and their performance could be degraded due to modelling errors and unmodelled dynamics.

In summary, model-based wind farm control methods have shown good effectiveness and are usually easy to implement, but they may suffer from high system complexity, inevitable modelling inaccuracy and stochastic environment uncertainty. In recent years, data-driven and model-free wind farm methods have drawn extensive research interest and have been treated as promising candidates to overcome the limitations of their model-based counterparts. For wind farm power maximization tasks, a game-theoretic method was proposed in [10] to decide the optimal induction factors for every turbine in the farm. Ref. [11] proposed a Bayesian-based searching method for yaw settings optimization. A deep reinforcement learning (DRL)-based wind farm control method was proposed in [12], in

which an internal wake model was employed to guide the learning process. However, all these important results are based on steady wind conditions, and they can only provide fixed yaw and/or induction factor settings subject to unchanged wind speeds. A DRL-based yaw control method that aimed to handle stepwise-varying inflow conditions was designed in [13], but it was built upon a steady-state wind farm model. Another two wind farm control approaches via DRL were proposed in [14], [15]. They are model-free and can adapt to time-varying wind speeds. But a limitation is that they employed averaged power outputs over particular time spans to construct rewards. In addition, they only considered yaw control tasks. To sum up, though data-driven wind farm control methods have aroused extensive attention, there are still many essential issues that limit the applicability and feasibility of current results, including the reliance on steady-state or time-averaged data (such as turbines' power outputs under steady wind conditions or from steady-state models), lack of robustness to time-varying wind conditions, and insufficiency in data mining.

Aiming to address the aforementioned challenges in current approaches, this paper proposes a new intelligent wind farm control method via a novel DRL algorithm. DRL [16], [17], [18] is a booming artificial intelligence technique that is currently triggering the technological revolution of many industrial sectors. It is capable of fulfilling high-performance data-driven control for multiple tasks of a large class of complex systems. In this paper, an application-oriented DRL method is designed to maximize wind farm power generation by yaw and induction control. Particularly, a multi-player DRL (MPDRL) algorithm is designed to learn the optimal control policy subject to a user-defined performance metric using the available measurements at turbine rotors, and deep neural networks (DNNs) are employed in it as information processors and universal approximators. Its effectiveness is verified by a dynamic wind farm simulator [8] for wind farms with different specifications. Test results show that our method can achieve clear farm-level power generation increases (over 30% in case studies) compared with the benchmark. Further explanations regarding the novelty of the proposed method in comparison with relevant studies are discussed as follows.

- Unlike mainstream wind farm control approaches [2], [3], [5], [6], [19], [7], our method relaxes the dependence on wind farm models. By employing the merits of DNN and DRL, it has the ability to capture the core system information and maximize the farm-level power production only by data. Our method has strong robustness against unmodelled dynamics and stochastic environments.
- Compared with important data-driven wind farm control methods (including some results based on DRL) [10], [11], [20], [12], [13], no steady-state or time-averaged data are required to carry out the learning process in our design. Instead, by embedding DNNs that can handle time-series data (such as long-short-term-memory (LSTM) networks, gated recurrent units (GRU), transfomer networks, and so on) into the main DRL structure, our MPDRL can mine in-depth the data measured at turbine rotors, handle wake propagation delay, and achieve closed-loop wind farm control under time-varying wind conditions. These essential features render our method to have enhanced performance.
- Distinct from the most recent DRL-based wind farm control approaches [12], [13], [14], [15], a special distractor network is employed in our method. This design not only mitigates the non-Markovian feature induced by the stochastic nature of wind conditions but also enhances our method's robustness.

It is noteworthy that the method proposed in this paper is partially built upon [14], [15], [4], and substantial new contributions have been made. Specifically, though Refs. [14], [15] also designed DRL algorithms to optimize farm-level power generation, they employed averaged power outputs over particular time spans to construct rewards, leading to potentially limited adaptability and robustness. In addition, they only considered yaw control tasks. These limitations are fully addressed in this work. The method proposed in this paper does not rely on any averaged data. Instead, time-series data that are easy to measure are employed to help our RL agent mine system information and adapt to real-time changes in environmental conditions. Moreover, not only yaw control but also axial-induction-based control strategies are considered in this paper. There are also several essential differences between the work in this paper and Ref. [4]. Firstly, these two studies consider different tasks. Instead of achieving power generation maximization, Ref. [4] focused on providing ancillary services by adjusting induction factors. The yaw control strategy was not designed in it, which is vital for farm-level power generation maximization tasks. In contrast, the present paper develops a DRL structure for both yaw & induction control to maximize the whole farm's power generation. These two kinds of tasks have quite different features and inherent control system design logic. It should be emphasized that simultaneously achieving yaw and induction control is challenging due to the distinctive and incompatible features of yaw angles and induction-related states. We employ the multi-player (MP) concept to address this issue. Our MP-DRL algorithm has two actors to learn the optimal yaw control and induction control policies simultaneously yet separately, one distractor to evaluate the worst-case external disturbances (i.e. wind speed changes) to the performance metric, and one critic to learn the optimal long-term performance functional. Such a special DRL structure allows us to not only handle the incompatibility of yaw & induction control but also bring robustness against external disturbances. Based on these facts, the proposed MPDRL-based wind farm control method is more general and flexible, and it renders enhanced adaptability and robustness compared with the results in [14], [15], [4].

In the remainder of this paper, we introduce the farm-level power maximization task in Sec. II. Following that, the MPDRL wind farm control method is explained in Sec. III. To verify its effectiveness, simulations with a dynamic wind farm simulator for different wind farms are presented in Sec. IV. After that, we conclude the whole work in Sec. V.

II. Problem Formalization

We consider a wind farm with n turbines that are denoted by $\mathcal{WT}_1$, $\mathcal{WT}_2$, ..., $\mathcal{WT}_n$, respectively. It is well-understood that the power captured by a turbine $\mathcal{WT}_i$ is directly related to its inflow wind speed U_i, the induction-related state α_i (such as the induction factor or some other states that are related to the induction factor, e.g. the modified thrust coefficient), and yaw offset β_i [2], [10], [11], formalized by

$$E_i = h(U_i, \alpha_i, \beta_i) \tag{1}$$

For turbine-level control, the induction-related state α_i decides the blade pitch angle and rotor torque. It should be emphasized that, in this paper, the specific expression of h is not required in controller design. Based on (1), the whole farm's power is

$$E = \sum_{i=1}^{n} E_i = \sum_{i=1}^{n} h(U_i, \alpha_i, \beta_i) \tag{2}$$

Before introducing the specific wind farm control objectives considered in this paper, we first define control inputs, states and exogenous inputs (i.e. disturbances) of the system. The control inputs are the changes of induction-related states and yaw angles of all turbines, denoted by $\delta\alpha = [\delta\alpha_1, \delta\alpha_2, ..., \delta\alpha_n]$ and $\delta\beta = [\delta\beta_1, \delta\beta_2, ..., \delta\beta_n]$, respectively. Then we define a regularized state vector to be

$$\bar{x} = [\bar{\alpha}_1, \bar{\beta}_1, \bar{E}_1, \bar{\alpha}_2, \bar{\beta}_2, \bar{E}_2, ..., \bar{\alpha}_n, \bar{\beta}_n, \bar{E}_n] \tag{3}$$

with $\bar{\alpha}_i = \frac{(\alpha_i - \alpha_{\max}) + (\alpha_i - \alpha_{\min})}{\alpha_{\max} - \alpha_{\min}}$, $\bar{\beta}_i = \frac{(\beta_i - \beta_{\max}) + (\beta_i - \beta_{\min})}{\beta_{\max} - \beta_{\min}}$, and $\bar{E}_i = \frac{E_i}{E_r}$. Here $\alpha_{\max}$ and $\alpha_{\min}$ are the upper and lower bounds of induction-related states, respectively; $\beta_{\max}$ and $\beta_{\min}$ are the upper and lower bounds of yaw angles, respectively; and E_r is the turbine's rated power. Therefore, one has $\bar{\beta}_i \in [-1, 1]$ and $\bar{\alpha}_i \in [-1, 1]$. We also define the following the regularized exogenous input

$$w = [\bar{U}_1, \bar{U}_2, ..., \bar{U}_n]^{\mathrm{T}} \tag{4}$$

where $\bar{U}_i = 2U_i/U_N - 1$ with U_N is a user-defined constant for normalization purposes.

The wind farm controller considered in this paper should make a trade-off for the following two objectives subject to wake effects: (a) maximizing the farm-level power generation – this is the main objective; (b) avoiding large loads caused by induction & yaw control.

Based on these objectives, we define the following reward function for time step k:

$$\begin{aligned} r(k) = &- c_1 \sum_{i=1}^{n} \bar{E}_i(k) + \frac{c_2}{n} \sum_{i=1}^{n} |F_i(k) - F_i(k-1)| \\ &+ \frac{c_3}{n} \sum_{i=1}^{n} |\frac{\beta_i(k)}{b_\beta}| \end{aligned} \tag{5}$$

and here c_1, c_2, and c_3 are user-defined weighting constants. We explain the terms in Eq. (5) in detail as follows.

(1) The first term in Eq. (5) is for Objective (a). The smaller the value of it, the larger the farm's power generation.

(2) The second term in Eq. (5) is for Objective (b). Particularly, F_i is the axial force that the wind flow exerts on WT_i, and it is defined as [21], [8]

$$F_i = \frac{1}{2} \rho A_i U_i^2 \cos^2(\beta_i) C'_{T_i} \tag{6}$$

Here ρ denotes the air density, A_i denotes the swept area of the rotor plane, and C'_{T_i} is called the modified thrust coefficient. Following [21], the whole term is related to dynamical turbine loading. Therefore, MPDRL can balance power generation maximization with dynamical load minimization by introducing this term into (5).

(3) The third term is employed to avoid unacceptable yaw offsets and large yaw-induced structural loads, and here b_β denotes the maximum acceptable yaw angle.

Remark 1: To test the performance of the MPDRL algorithm proposed in this paper, we utilize the dynamic wind farm simulator (WFSim) proposed in [8] to carry out case studies (see Sec. IV). As explained in [8] and adopted in many studies (e.g., [21], [9], [7]), WFSim employs the modified thrust coefficient C'_{T_i} and the yaw angle β_i or their changes as the control variables. It is noteworthy that C'_{T_i} is directly related to the turbine set point in terms of blade pitch angle and rotational speed (see Appendix in [8] for details). Moreover, it should be emphasized that the proposed MPDRL method is data-driven, and its learning process does not rely on underlying models of any specific wind farm simulators. WFSim is employed in this paper because it is a control-oriented simulator that can keep the key features of dynamic flow fields and wind farms while balancing simulation fidelity and computational complexity. Therefore, to indicate that other proper induction-related states can also be employed as control variables in our MPDRL, we still denote the induction-related state as α_i instead of directly denoting it as C'_{T_i} in MPDRL design to keep generality.

The main control goal is finding the best control policies for $\delta\alpha$ and $\delta\beta$ to minimize the following long-term performance metric, where $\gamma \in (0, 1]$ is a constant discount factor.

$$V = \sum_{k=0}^{\infty} \gamma^k r(k) \tag{7}$$

We note that solving this task is challenging, and it is intractable for conventional control methods. The main difficulties are from the following aspects:

(1) As mentioned in the introduction, wind-farm control tasks are commonly subject to complicated aerodynamic couplings among turbines. Particularly, due to the existence of wake effects, the power generation of downstream turbines are not only decided by the control actions of themselves but also influenced by that of other turbines. Particularly, w in (4) is related to $\bar{x}$ in (3) and the time-varying inflow wind speed (denoted by U_∞). Such a complicated relationship is challenging to be accurately modelled given the stochastic nature of wake effects and U_∞. It also renders no analytical solution for V in (7) and hinders the feasibility of conventional optimal control methods. This paper addresses this complex task via DRL, which allows us to approximate V and learn the optimal control policies by deep neural networks, achieving

data-driven farm-level power maximization under time-varying wind speeds without requiring any analytical models.

(2) Wake effects have time-delayed features. The wake changes induced by control actions typically require tens/hundreds of seconds to propagate from upstream turbines to downstream turbines, leading to difficulties in evaluating control performance from a long-term perspective. Moreover, this issue also renders the whole task to be a partially observable Markov decision process, blocking the direct application of mainstream DRL algorithms, such as DDPG [18]. To handle this issue in this paper, we employ a multi-player DRL structure to encode look-back data and evaluate the long-term reward V while considering the stochastic nature of wind speeds.

III. Design of a Multi-Player DRL Algorithm for Wind Farm Control

In order to handle the time-delayed feature of wake effects and alleviate the non-Markovian feature of the whole control task, we feed not only the instantaneous measurements of x but also its past data into the wind farm control method. Particularly, we define

$$x(k) = [\bar{x}(k-l), \bar{x}(k-l+1), ..., \bar{x}(k)]^{\mathrm{T}} \tag{8}$$

where $l >= 0$ denotes a user-defined look-back step to mitigate the non-Markovian feature. Then we can describe the whole wind farm control system as follow.

$$x(k+1) = f(x(k), \delta\alpha, \delta\beta, w) \tag{9}$$

However, due to the wake effect's inherent complexity and stochastic nature, f is unknown for controller design. Based on the principle of H_∞ control technique, the control objective is to learn the optimal control policies $\delta\alpha^*$ and $\delta\beta^*$ under the influence of the potentially worst-case w (denoted by w^*), formalized by

$$\delta\alpha^*(k), \delta\beta^*(k) = \arg\min_{\delta\alpha,\delta\beta} V^*(k) \tag{10}$$

Here $V^*(k)$ is based on the following definition for any $k \in \mathbb{N}$

$$V^*(k) \quad = \min_{\delta\alpha,\delta\beta} \max_{w} \{V(x(k), \delta\alpha(k), \delta\beta(k), w(k)\} \tag{11}$$

Given all these preliminaries, we can summarize the wind farm control task as follows.

Solving $\delta\alpha^*$, $\delta\beta^*$, and w^* for

$$V^*(k) = \min_{\delta\alpha,\delta\beta} \max_{w} \{V(x(k), \delta\alpha(k), \delta\beta(k), w(k)\} \tag{12}$$

Subject to

$$x(k+1) = f(x(k), \delta\alpha, \delta\beta, w) \tag{13}$$

$$\delta\alpha_{\min} \leq \delta\alpha_i \leq \delta\alpha_{\max},\ i = 1, 2, ..., n \tag{14}$$

$$\delta\beta_{\min} \leq \delta\beta_i \leq \delta\beta_{\max},\ i = 1, 2, ..., n \tag{15}$$

$$-1 \leq \bar{\alpha}_i \leq 1,\ i = 1, 2, ..., n \tag{16}$$

$$-1 \leq \bar{\beta}_i \leq 1,\ i = 1, 2, ..., n \tag{17}$$

$$\bar{U}_i \in \mathcal{L}_\infty,\ i = 1, 2, ..., n \tag{18}$$

Here $\delta\alpha_{\min}$, $\delta\alpha_{\max}$, $\delta\beta_{\min}$, and $\delta\beta_{\max}$ are the bounds for one-step control actions.

With Eqs. (12) - (18), the wind farm control task is transformed to a game problem with two minimizing players (i.e. $\delta\alpha$ and $\delta\beta$) and one maximizing player (i.e. w). However, since f is an unknown function with high nonlinearity and complexity, it is impossible to analytically solve the Nash equilibrium $\{\delta\alpha^*, \delta\beta^*, w^*\}$ for this game. Instead, we propose a multi-player DRL algorithm to approximate V^* and the corresponding optimal control policies $\delta\alpha^*$ and $\delta\beta^*$.

An important property of the game problem considered here is that V^* satisfies the so-called discrete-time Hamilton-Jacobi-Isaacs (HJI) equation [22], which is in line with the Principle of Optimality:

$$V^*(k) = \min_{\delta\alpha,\delta\beta} \max_{w} \{r(k) + \rho V^*(k+1)\},\ \forall k \in \mathbb{N} \tag{19}$$

After that, we define the Q-function [23], [24], [16], [17]:

$$\begin{aligned} &Q_{\delta\alpha,\delta\beta,w}(x(k), a_\alpha, a_\beta, d) \\ &= \sum_{i=k+1}^{\infty} \gamma^{i-k} r(x(k+1), \delta\alpha(k+1), \delta\beta(k+1), w(k+1)) \\ &\quad + r(x(k), a_\alpha, a_\beta, d) \\ &= \gamma Q_{\delta\alpha,\delta\beta,w}(x(k+1), \delta\alpha(k+1), \delta\beta(k+1), w(k+1)) \\ &\quad + r(x(k), a_\alpha, a_\beta, d) \end{aligned} \tag{20}$$

Here $Q_{\delta\alpha,\delta\beta,w}$ is commonly referred to as an action-state value function [23], [24], [16], [17]. It represents the value of the performance metric obtained when the inputs a_α a_β, d are applied at time step k, and the policies $\delta\alpha$, $\delta\beta$ and w are pursued thereafter [24]. It should be emphasized that Eq. (20) also applies to the Nash equilibrium $\{\delta\alpha^*, \delta\beta^*, w^*\}$, i.e.,

$$\begin{aligned} &Q_{\delta\alpha^*,\delta\beta^*,w^*}(x(k), a_\alpha, a_\beta, d) \\ &= \gamma Q_{\delta\alpha^*,\delta\beta^*,w^*}(x(k+1), \delta\alpha(k+1), \delta\beta(k+1), w(k+1)) \\ &\quad + r(x(k), a_\alpha, a_\beta, d) \end{aligned} \tag{21}$$

The Q-function defined above plays a key role in the learning of control policies and in the update of DNN parameters.

A specially designed critic-actor-distractor structure is employed in our MPDRL algorithm. An illustration that shows the main modules and frameworks of MPDRL is given in Fig.1. Particularly, the critic aims to approximate V^*; the two actors are employed to learn the optimal control policies for $\delta\alpha$ and $\delta\beta$ (i.e. $\delta\alpha^*$ and $\delta\beta^*$), respectively; and the function of the distractor is to evaluate the worst-case disturbance, i.e. w^*. All these modules are built by DNNs. We denote the parameters of the critic, the two actors, and the distractor DNNs as θ^Q, θ^α, θ^β and θ^w, respectively, and their outputs are Q, $\delta\alpha$, $\delta\beta$, and w, respectively.

Furthermore, an additional set of DNNs is also employed, which is the "soft copy" of the main critic-actor-distractor structure. Particularly, these DNNs have exactly the same neuron types & numbers and network layers as their counterparts in the main critic-actor-distractor structure. These additional DNNs form a target critic-actor-distractor structure. We denote their parameters as $\theta^{Q'}$, $\theta^{\alpha'}$, $\theta^{\beta'}$, and $\theta^{w'}$, respectively, and

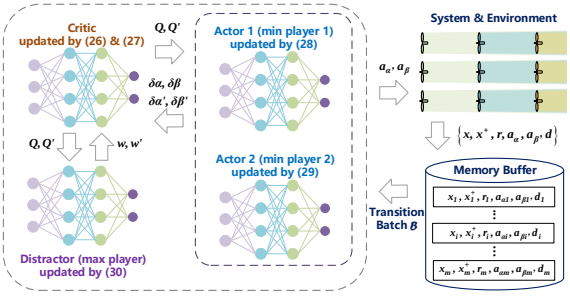

Figure 1: Main structure of MPDRL.

their outputs are denoted by Q', $\delta\alpha'$, $\delta\beta'$, and w', respectively. These parameters slowly track their counterparts with a small positive rate τ. It should be emphasized that the idea of employing such an additional set of DNNs is originally from [17]. This design can significantly improve the learning stability, which has been proven by many relevant studies [4], [14], [17], [18], [25].

Moreover, we also employ the experience replay strategy [17], [18] in our MPDRL to mitigate the correlation issue of sequential learning data. Specifically, as shown in Fig. 1, a memory buffer $\mathcal{M}$ (with a size of m) is employed to store the system data $\{x(k), x(k+1), r(k), \delta\alpha(k), \delta\beta(k), d(k)\}$ (which is commonly referred to as a transition) at every time step k. It is noteworthy that here $d(k)$ is the actual disturbance observed at time step k instead of the policy $w(k)$ generated by the distractor. For each learning iteration, a small batch (with a size of b) of transitions (denoted by $\{x_i, x_i^+, r_i, \delta\alpha_i, \delta\beta_i, d_i\}$, $i = 1, 2, ..., b$) are randomly selected from $\mathcal{M}$ to update DNNs.

Then we are ready to introduce the updating laws for the main critic-actor-distractor structure. Based on the Q-function in Eq. (20), we define the following temporal difference error (TD-error) for any transition $\{x_i, x_i^+, r_i, \delta\alpha_i, \delta\beta_i, d_i\}$:

$$\begin{aligned} e_i =& r_i + \gamma Q'(x_i^+, \delta\alpha'(x_i^+|\theta^{\alpha'}), \delta\beta'(x_i^+|\theta^{\beta'}), w'(x_i^+|\theta^{w'})|\theta^{Q'}) \\ & - Q(x_i, \delta\alpha_i, \delta\beta_i, d_i|\theta^Q) \end{aligned} \tag{22}$$

The critic aims to minimize the amplitude of e_i for all the transition i in a sampled batch at every learning iteration. To this end, we define the following loss function for the transition batch, which directly drives the updating of the main critic's parameters (i.e. θ^Q):

$$L = \sum_{i=1}^{b} \|e_i\|^2 \tag{23}$$

As discussed before, the actors $\delta\alpha$ and $\delta\beta$ are minimizing players that aim to minimize the value of the long-term performance metric (which is estimted by Q), while the purpose of the distractor w is the opposite. Based on that, their parameters' updating process can be driven by the corresponding gradients of Q, which are defined as follows.

$$\nabla_{\theta^\alpha} = \frac{1}{b} \sum_{i=1}^{b} [\nabla_{\delta\alpha} Q(x_i, \delta\alpha, \delta\beta, w) \cdot \nabla_{\theta^\alpha} \delta\alpha(x_i|\theta^\alpha)] \tag{24}$$

$$\nabla_{\theta^\beta} = \frac{1}{b} \sum_{i=1}^{b} [\nabla_{\delta\beta} Q(x_i, \delta\alpha, \delta\beta, w) \cdot \nabla_{\theta^\beta} \delta\beta(x_i|\theta^\beta)] \tag{25}$$

$$\nabla_{\theta^w} = \frac{1}{b} \sum_{i=1}^{b} [\nabla_{w} Q(x_i, \delta\alpha, \delta\beta, w) \cdot \nabla_{\theta^w} w(x_i|\theta^w)] \tag{26}$$

In every learning iteration, the updating process of θ^α and θ^β are driven by $-\nabla_{\theta^\alpha}$ and $-\nabla_{\theta^\beta}$, respectively, while θ^w is updated by ∇_{θ^w}.

Based on all these designs, we summarize our MPDRL algorithm for wind farm control in Algorithm 1.

Algorithm 1 Multi-Player Deep Reinforcement Learning (MPDRL) Algorithm for Wind Farm Control.

- Aggregate and normalize the system states, control inputs and disturbances to transform the wind farm control problem into a multi-player game as described in Eqs. (12)-(18).
- Decide hyperparameters for the critic-actor-distractor DRL structure, including networks' layer numbers and neuron numbers & types for each layer.
- Initialize θ^Q, θ^α, θ^β, θ^w, $\theta^{Q'}$, $\theta^{\alpha'}$, $\theta^{\beta'}$, $\theta^{w'}$ and all the other user-defined parameters.

for each learning episode **do**
  for $k = 0$ to the maximum steps per episode **do**
    Based on the current system state $x(k)$, generate control actions a_α and a_β via $a_\alpha = \delta\alpha(x(k)|\theta^\alpha) + \epsilon_\alpha(k)$ and $a_\beta = \delta\beta(x(k)|\theta^\beta) + \epsilon_\beta(k)$, respectively, where $\epsilon_\alpha(k)$ and $\epsilon_\beta(k)$ are noises for exploration purposes.
    Apply the control actions a_α and a_β to the wind farm control system, and observe $x(k+1)$, d, and r.
    Organize the transition $\{x(k), x(k+1), r, a_\alpha, a_\beta, d\}$ and store it in the memory buffer $\mathcal{M}$.
    Sample a batch of transitions from $\mathcal{M}$, denoted by $\{x_i, x_i^+, r_i, a_{\alpha i}, a_{\beta i}, d_i\}$, $i = 1, 2, ..., n$.
    Update the critic's parameter θ^Q via the loss function L defined by Eqs. (22) and (23).
    Update θ^α, θ^β and θ^w via Eqs. (24) - (26).
    Update $\theta^{Q'}$, $\theta^{\alpha'}$, $\theta^{\beta'}$ and $\theta^{w'}$ via 'soft replacement'.
  end for
end for

Remark 2: The proposed MPDRL algorithm is built upon the Q-learning theory [16], [23], [24]. One can refer to [23] for the key idea behind a standard Q-learning algorithm and its general convergence analysis. Distinct from mainstream DRL algorithms that also employ Q-learning or its variants such as Deep Q-Network [17] and Deep Deterministic Policy Gradient [18], we propose a special multi-player structure consisting of two actors (i.e. $\delta\alpha$ and $\delta\beta$) that aim to minimize the reward function and a distractor (i.e. w) that has the opposite objective. On the one hand, this design can evaluate the potential worst-case disturbances, guiding control policies and enhancing the whole system's robustness. On the other hand, employing two separated actors (instead of aggregating the yaw and induction control signals together) can adapt to the distinctive features (e.g. different changing rates) of different control inputs and enhance the learning effectiveness and the algorithm's overall performance.

IV. Case Studies

In this section, we utilize WFSim [8] to test the performance of MPDRL. As mentioned in Remark 1, WFSim utilizes the modified thrust coefficient (i.e. C'_{T_i}) and yaw angles or their changes as control variables. Therefore, we replace all terms related to α_i in MPDRL with the corresponding terms in C'_{T_i} to adapt to the requirement in WFSim.

In addition to the MPDRL method proposed in this paper, we also employ three other wind farm control methods in simulations to compare their performance with our MPDRL:

(1) The greedy control strategy. This strategy is the benchmark for wind farm control tasks, and it is currently the most commonly-employed wind farm control strategy in the industry. In the greedy strategy, every turbine in the farm aims to maximize its own power generation without considering the influence of other turbines. Following that and [8], one can set $C'_{Ti} \equiv 2, \beta_i \equiv 0°, i = 1, 2, ..., n$, for all the turbines in the farm to conduct the greedy strategy.

(2) A model-based wind farm control method (MB-WFC) in [3]. This important method carries out receding-horizon optimization based on the FLORIS model. It can achieve closed-loop wind farm control under time-varying wind conditions - a very suitable candidate to compare with MPDRL.

(3) A DRL-based method. This method has the same main structure and settings as MPDRL, but it does not have the distractor. We mention it as DRL-W/D in simulations. It can help us test the robustness of MPDRL further and show the advantage of the multi-player structure.

Two simulation scenarios are considered in case studies. Specifically, we employ a prototypical wind farm with nine NREL 5MW wind turbines in the first scenario. Then a large-scale wind farm from the European CL-Windcon[1] project that contains 80 DTU 10MW wind turbines is employed to test the scalability of MPDRL.

Table I: Simulation Settings.

Parameters	Values
$C'_{T\min}, C'_{T\max}, \beta_{\min}, \beta_{\max}, b_\beta$	$0.1, 2, -30°, 30°, 30°$
$\delta C'_{T\min}, \delta C'_{T\max}, \delta\beta_{\min}, \delta\beta_{\max}$	$-0.1,\ 0.1,\ -1°,\ 1°$
c_1, c_2, c_3	$1, 0.05, 0.1$
l, γ, τ, b, m	$100, 0.99, 0.05, 128, 10000$

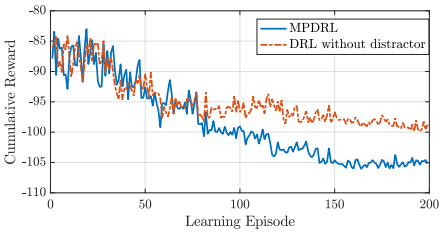

Figure 2: Cumulative reward in DRL learning.

[1]http://www.clwindcon.eu/

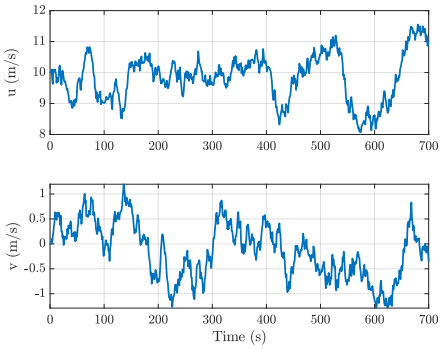

Figure 3: Wind speed (u: longitudinal; v: lateral).

A. Case Study with a Prototypical Nine-Turbine Wind Farm.

In this case study, we simulate a 2518.8m×1558.4m flow field with a wind farm that consists of nine NREL 5MW turbines in WFSim. A bird's-eye view of the flow field and the wind farm is in Fig. 4. Our method needs to employ DNNs that can handle time-series data, such as long-short-term-memory (LSTM) networks, gated recurrent units (GRU), transfomer networks, etc. Without loss of generality, here we employ LSTM DNNs in our case studies. We carry out DNN training based on the settings in Table I. It is noteworthy that time-varying wind conditions are employed in the training and testing of MPDRL. A 700-second example for a time-varying wind profile is given in Fig. 3, with u being the longitudinal wind speed and v being the lateral wind speed. It should be emphasized that the wind profile differs in each learning episode, and our MPDRL can only employ current & past measured wind conditions (no preview information for the future). In addition, measurement errors and process noises are also taken into account. Specifically, zero-mean Gaussian noises with standard deviations 0.5 and 0.05 are set to be the measurement errors of β_i and C'_{T_i}, respectively; the control signals $\delta\beta_i$ and $\delta C'_{T_i}$ are also polluted by such noises but the standard deviations are 0.1 and 0.01, respectively.

Given all these settings, the learning curves (i.e. the curve of the cumulative reward r per learning episode with 200 steps) of the two DRL-based wind farm control methods are shown in Fig. 2. Though both methods have the essential learning ability to improve control policy (we aim to minimize rewards in this paper), MPDRL leads to clearly superior learning performance compared to the case without the distractor module. It has an averagely lower episode-reward level under the influence of the same exogenous inputs, measurement errors and process noises (though these disturbances differ in each episode).

Based on the training results, we test MPDRL by a 700-second run with the wind profile in Fig. 3 and initial conditions as $C'_{Ti}(0) = 2, \beta_i(0) = 0°, i = 1, 2, ..., n$. Simulations with the other three strategies are also conducted for performance validation and comparison purposes. The flow field at $t = 700$s under MPDRL is shown in Fig. 4. One can see that MPDRL successfully achieves wake steering. The normalized power outputs (w.r.t the greedy power output at $t = 0$s) under different controllers are illustrated in Fig. 5. It can be observed that MPDRL, DRL-W/D and MB-WFC all lead to power

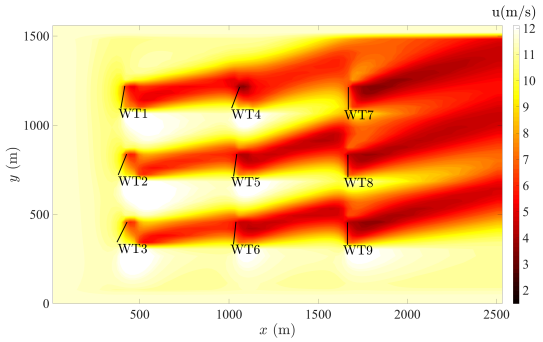

Figure 4: Flow field under MPDRL (at $t = 700$s).

increases w.r.t the greedy strategy, and MPDRL has the best performance among all these methods - MPDRL achieves a 34.23% power increase on average compared to the greedy strategy during the 700-second period while that of DRL-W/D and MB-WFC are 26.27% and 21.78%, respectively.

These results demonstrate the robustness of MPDRL from two aspects: (1) Compared to the important model-based method in [3], MPDRL is data-driven and model-free, showing robustness to potential uncertainties and modelling errors and leading to higher power generation in simulations. (2) MPDRL also leads to better learning & testing performance than DRL-W/D in the case study. As we mentioned before, the key difference between them is the distractor module in our multi-player structure. As verified by simulations, that brings MPDRL the robustness against the exogenous disturbance w.

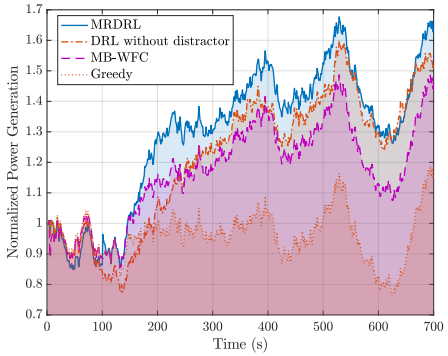

Figure 5: Normalized farm-level power generation.

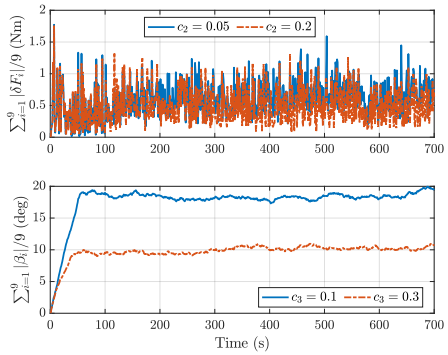

Figure 6: Responses of load-related terms.

As discussed in Sec. II, MPDRL should balance two competing objectives (a) maximizing the farm-level power generation (core objective); and (b) avoiding large loads induced by induction & yaw control (secondary objective). The trade-off between these two objectives can be achieved by adjusting the parameters c_1, c_2 and c_3 in the definition of r in Eq. (5). To illustrate that, we conduct two additional simulations for MPDRL under different parameters: (1) Adjust c_2 from 0.05 to 0.2 and keep all the other parameters unchanged. That enlarges the weight of the term $\frac{c_2}{n}\sum_{i=1}^{n}|F_i(k) - F_i(k-1)|$ in (5). As mentioned in Sec. II, this term is related to the dynamical turbine loading. (2) Adjust c_3 from 0.1 to 0.3 and keep all the other parameters unchanged. That enlarges the weight of the term $\frac{c_3}{n}\sum_{i=1}^{n}|\frac{\beta_i(k)}{b_\beta}|$, aiming to avoid unacceptable yaw offsets and large yaw-induced structural loads. Simulation results of MPDRL under these changes are given in Fig. 6. Compared to the results with the original settings, the averaged dynamical turbine load and the averaged yaw offset are reduced by 19.49% and 45.34%, respectively.

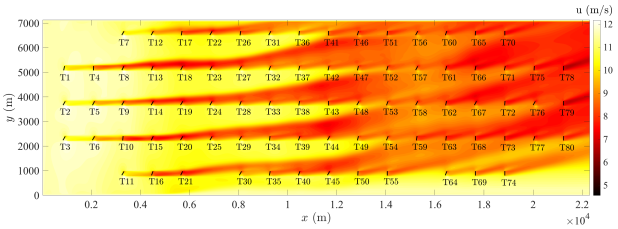

Figure 7: Flow field of the 80-turbine farm under MPDRL.

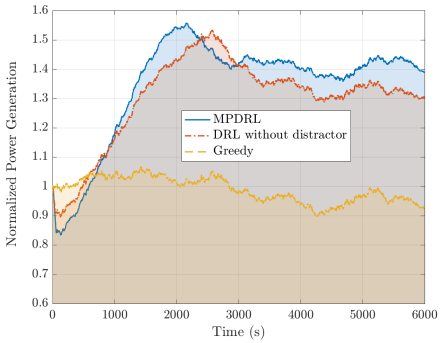

Figure 8: Normalized power of the 80-turbine wind farm.

B. Case Study with a Large-Scale Wind Farm.

In order to test the scalability of MPDRL, we consider a large wind farm with 80 DTU wind turbines (layout illustrated in Fig. 7) in this case study. Notably, the flow field in this case study is 40 times larger than the one in Sec. IV.A. Therefore, a 6000-second run under stochastic wind speeds is employed to comprehensively test the performance of MPDRL from a long-term standpoint. The flow field at $t = 6000$s under MPDRL is given in Fig. 7. One can see that the wake effect in this case study is much more complicated than in Sec. IV.A. The normalized power outputs (with respect to the greedy power generation at $t = 0$s) under different control strategies are illustrated in Fig. 8. It indicates that MPDRL can still lead to

a significant farm-level power generation increase compared with the benchmark – an over 34.3% increase in average is achieved. Moreover, its performance is better than DRL-W/D, showing its strong robustness.

V. Conclusions

A deep reinforcement learning (DRL)-based wind farm control method was proposed in this paper to maximize the farm-level power generation under strong wake effects and stochastic wind speeds. Distinct from conventional wind farm control methods, the proposed method is data-driven and model-free – it does not require any analytical models (e.g. wake model) to carry out wind farm control. Benefiting from the multi-player concept, the robust control technique, and a specially designed critic-actor-distractor structure, our DRL-based method has good robustness, adaptability and applicability. Simulation results verified the proposed method's effectiveness. It increased farm-level power generation by over 30% compared with the greedy strategy and showed better performance than relevant approaches. It should be emphasized that the rate of farm-level power generation improvement can be influenced by many factors, including the operating conditions (e.g. the main wind direction), the wind farm layouts (e.g. the longitudinal and lateral distances between turbines), and also wind turbine types. The simulation results demonstrated were for the wind farms with the operating conditions and specifications in case studies. Particularly, the dominant wind direction was along the rows of wind turbines. Though this is a commonly-used and most adopted setting in relevant studies, it could not represent the entire lifetime operating conditions of wind farms. Therefore, the magnitudes of benefit in case studies could not be achieved when considering the full-lifetime operations of real wind farms. It is noteworthy that, though different operation conditions and specifications vary the magnitude of benefit, our DRL algorithm always aims to learn the optimal wind farm control strategies, and it can adapt to different scenarios. As future work, hybrid reinforcement learning and grouping strategies will be explored in the future to enhance the learning efficiency and reduce computational complexity of the proposed method.

References

[1] GWEC, "Global wind report," 2021.

[2] P. M. O. Gebraad, F. Teeuwisse, J. Van Wingerden, P. A. Fleming, S. Ruben, J. Marden, and L. Pao, "Wind plant power optimization through yaw control using a parametric model for wake effects - a CFD simulation study," *Wind Energy*, vol. 19, no. 1, pp. 95–114, 2016.

[3] B. M. Doekemeijer, D. van der Hoek, and J.-W. van Wingerden, "Closed-loop model-based wind farm control using floris under time-varying inflow conditions," *Renewable Energy*, vol. 156, pp. 719–730, 2020.

[4] H. Dong and X. Zhao, "Wind-farm power tracking via preview-based robust reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1706–1715, 2021.

[5] J. Annoni, P. Seiler, K. Johnson, P. Fleming, and P. Gebraad, "Evaluating wake models for wind farm control," in *2014 American Control Conference*. IEEE, 2014, pp. 2517–2523.

[6] T. Knudsen, T. Bak, and M. Svenstrup, "Survey of wind farm control—power and fatigue optimization," *Wind Energy*, vol. 18, no. 8, pp. 1333–1351, 2015.

[7] M. Vali, V. Petrović, S. Boersma, J.-W. van Wingerden, L. Y. Pao, and M. Kühn, "Adjoint-based model predictive control for optimal energy extraction in waked wind farms," *Control Engineering Practice*, vol. 84, pp. 48–62, 2019.

[8] 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.

[9] K. Chen, J. Lin, Y. Qiu, F. Liu, and Y. Song, "Model predictive control for wind farm power tracking with deep learning-based reduced order modeling," *IEEE Transactions on Industrial Informatics*, 2022.

[10] J. R. Marden, S. D. Ruben, and L. Y. Pao, "A model-free approach to wind farm control using game theoretic methods," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1207–1214, 2013.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] H. Dong and X. Zhao, "Composite experience replay-based deep reinforcement learning with application in wind farm control," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 3, pp. 1281–1295, 2021.

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[17] 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.

[18] 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," *arXiv preprint arXiv:1509.02971*, 2015.

[19] M. Vali, V. Petrović, S. Boersma, J.-W. van Wingerden, and M. Kühn, "Adjoint-based model predictive control of wind farms: Beyond the quasi steady-state power maximization," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4510–4515, 2017.

[20] J. Park and K. H. Law, "A data-driven, cooperative wind farm control to maximize the total power production," *Applied Energy*, vol. 165, pp. 151–165, 2016.

[21] 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.

[22] H. Zhang, C. Qin, B. Jiang, and Y. Luo, "Online adaptive policy learning algorithm for H_∞ state feedback control of unknown affine nonlinear discrete-time systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2706–2718, 2014.

[23] 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.

[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] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.