Applied Energy 292 (2021) 116928

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Intelligent wind farm control via deep reinforcement learning and high-fidelity simulations

Hongyang Dong, Jincheng Zhang, Xiaowei Zhao [*]

Intelligent Control & Smart Energy (ICSE) Research Group, School of Engineering, University of Warwick, Coventry, UK

ARTICLE INFO

Keywords:
Wind energy
Wind farm control
Power generation optimization
Deep reinforcement learning
CFD simulation

ABSTRACT

Wind farms' power-generation efficiency is constrained by the high system complexity. A novel deep reinforcement learning (RL)-based wind farm control scheme is proposed to handle this challenge and achieve power generation optimization. A reward regularization (RR) module is designed to estimate wind turbines' normalized power outputs under different yaw settings and uncertain wind conditions, which brings strong robustness and adaptability to the proposed control scheme. The RR module is then combined with the deep deterministic policy gradient algorithm to evaluate the optimal yaw settings for all the wind turbines within the farm. The proposed wind farm control scheme is data-driven and model-free, which addresses the limitations of current approaches, including reliance on accurate analytical/parametric models and lack of adaptability to uncertain wind conditions. In addition, a novel composite learning-based controller for each turbine is designed to achieve closed-loop yaw tracking, which can guarantee the exponential convergence of tracking errors in the presence of uncertainties of yaw actuators. The whole control system can be pre-trained offline and fine-tuned online, providing an easy-to-apply solution with enhanced generality and flexibility for wind farms. High-fidelity simulations with SOWFA (simulator for offshore wind farm applications) and Tensorflow show that the proposed scheme can significantly improve the wind farm's power generation by exploiting a sparse data set without requiring any wake model.

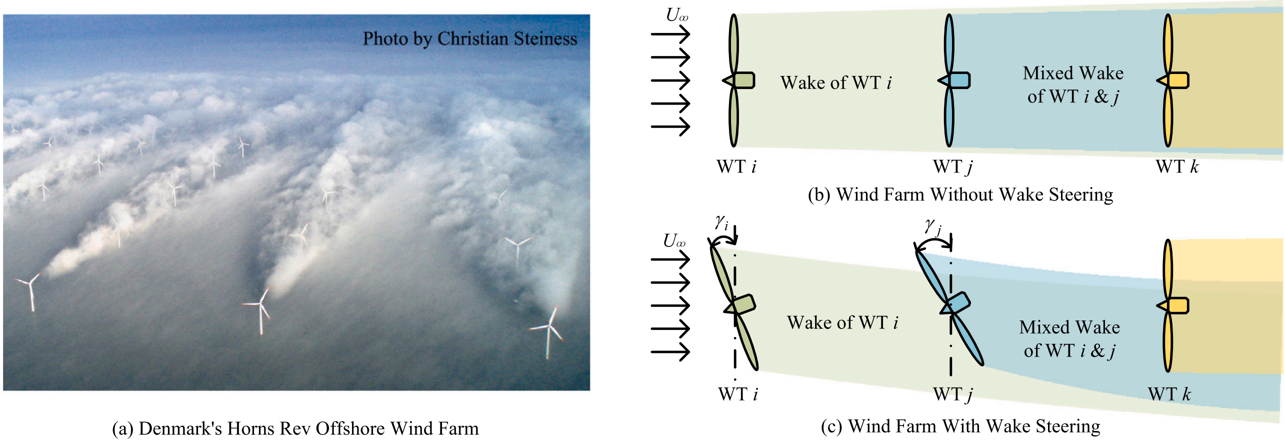
1. Introduction

Wind energy is one of the most important sustainable energy, and it has become an essential source of global power generation. In 2019, it accounted for 4.7% of the electricity usage worldwide, 15% in Europe, and 20% in the UK. Recently, the development of wind farms is growing drastically to harvest more wind power. However, wind farms' power-generation efficiency and economic benefits still severely suffer from the high system complexities and uncertain environments as illustrated in Fig. 1a, which shows a typical wind farm — Denmark's Horns Rev offshore wind farm. From Fig. 1a, one can see that the power exaction process of an upstream wind turbine results in a wake behind it. This wake has a reduced wind speed (compared with the free stream wind), which therefore interferes with the downstream wind turbine's power exaction. This phenomenon is commonly mentioned as wake effect, which has a significant influence on the total power production of wind farms. For example, it results in a 20% loss on annual power production of the Horns Rev offshore wind farm. Many studies have been carried out to investigate wake effects. Ref. [1] studied the detailed wake aerodynamics of horizontal-axis wind turbines. Experimental investigation of wake effects was presented in [2]. Ref. [3] characterized the wake effects considering yaw and pitch angles. A repowering optimization method was designed in [4] under wake effects, and a hybrid wind-farm model was presented in [5] by combining wake effects and stochastic dependability.

Some wind farm optimization & control approaches have been proposed to mitigate wake effects. A layout optimization method was designed in [6] to improve wind farm generation under wake effects. Vali et al. [7] proposed model predictive control (MPC) methods for wind farms, aiming to minimize wake effects by adjusting the induction factors for all turbines. Ref. [8] developed a surrogate deep-learning model to replace the analytical flow field model and achieve induction control via MPC. Another effective strategy to mitigate wake effects is wake steering. To be specific, wake steering aims to manipulate the wakes' directions by judiciously adjusting the yaw angles of wind turbines such that the wake effects on the downstream turbines can be mitigated, as demonstrated in Fig. 1b and c. Though this control strategy may reduce the outputs of upstream wind turbines, it can increase the whole farm's total power generation. However, due to wake effects' inherent stochastic features, accurately analysing the mathematical relationship between the yaw settings and the resulting

[*] Corresponding author.
E-mail addresses: hongyang.dong@warwick.ac.uk (H. Dong), jincheng.zhang@warwick.ac.uk (J. Zhang), xiaowei.zhao@warwick.ac.uk (X. Zhao).

https://doi.org/10.1016/j.apenergy.2021.116928
Received 20 January 2021; Received in revised form 1 April 2021; Accepted 3 April 2021
Available online 13 April 2021



(a) Denmark's Horns Rev Offshore Wind Farm

(b) Wind Farm Without Wake Steering

(c) Wind Farm With Wake Steering

Fig. 1. Illustrations of the wind farm, wake effect, and wake steering. (a) A photo of Denmark's Horns Rev Offshore Wind Farm, which shows a typical wind farm layout and the wake effect. (b) An illustration of a wind farm without wake steering. (c) An illustration of a wind farm with wake steering.

wake effects is challenging and computationally complicated, and it is even more challenging to design effective yaw control strategies upon it. To address this issue, some elegant studies [9–11] developed or employed simplified models for wake effects, and then utilized these surrogate models to develop wind farm control strategies. Particularly, a parametric wake model was developed in [9] to decide yaw offsets. Ref. [10] extended the Jensen wake model [12] to achieve yaw control. Another surrogate wake model was proposed in [11] for wake steering and wind-farm power optimization. Nevertheless, the downside of this type of approaches is that they are sensitive to unmodelled dynamics and system uncertainties, and thus in practice the performance of these model-based controllers can be quite different from the analytical results. Due to these facts, model-free approaches for wind farm power generation optimization become appealing alternatives.

Generally speaking, model-free control aims to achieve control objectives using only input & output data without requiring system models. Therefore, it has strong adapting abilities to the inherent complex system dynamics and can handle challenging tasks that are difficult to address by model-based methods. Though model-free control is currently drawing worldwide attention, it is still in an embryonic stage, in particularly designing model-free wind farm control methods is an open problem. A notable attempt towards this end was presented in [13,14], in which the Bayesian ascent (BA) approach was designed and applied to wind farms. BA is built upon the Bayesian optimization technique, aiming to search the optimal yaw settings through the probability-distribution prediction with Gaussian regression. However BA requires wind conditions maintaining steady during the searching process. In addition, it is actually an open-loop strategy that relies on free explorations of the whole state domain. Therefore, developing new model-free closed-loop control methods with enhanced adaptability and generality is needed to improve wind farms' power generation. In this paper, we achieve this goal through reinforcement learning (RL).

RL [15] is a cutting-edge research area that combines the merits of multiple disciplines including control engineering, machine learning, and statistics. Its fundamental principle is "trial and error", which aims to iteratively improve control policies by judiciously evaluating input & output data. A notable example of RL is the famous deep Q-network (DQN) algorithm [16], which can achieve human-level control in Atari and Go games. As an extension of DQN, the deep deterministic policy gradient (DDPG) algorithm [17] can handle more complicated problems with continuous control domains. Luo et al. employed RL to handle output regulation problems [18] and zero-sum games [19]. Many other RL algorithms have been applied to different complex systems, such as wind farms [20] and autonomous systems [21].

In this paper, a novel RL-based control method is designed to optimize wind farms' total power generation through wake steering. One of the main technical barrier comes from the stochastic feature of wind conditions, which makes the whole problem non-Markovian and thus breaks RL's fundamental assumption [15]. We design a reward regularization (RR) module to address this issue, which can evaluate the normalized increments of power production (compared with the benchmark) under different yaw settings and wind conditions. RR module maintains the essential Markovian property of the whole problem and brings the adapting abilities to our RL-based control method regarding uncertain wind conditions. Then we embed the RR module in DDPG to learn the optimal yaw settings for all the wind turbines in the farm. Two deep neural-network (NN) structures, critic and actor, are employed to approximate the value function and optimal control policy. Finally, based on the result of RR-DDPG, a composite learning (CL)-based controller is proposed to achieve closed-loop yaw tracking.

The novelty and main contributions of this paper are as follows.

- The wind farm's power-generation efficiency can be significantly improved via the proposed deep RL-based control approach. Different from relevant model-based wind farm control methods [9–11], our design is data-driven and model-free, which is insensitive to parameter uncertainties and unmodelled dynamics. Simulation results show that our method can significantly improve the wind farm's total power production by 15% on average compared with the benchmark.
- The proposed control scheme is application-oriented. (1) The training and learning data (power output and yaw angle of each turbine) are easy to collect. The proposed control scheme only needs a sparse training dataset, which can be obtained at limited sets of fixed yaw settings (e.g. 90 sets of data is sufficient in the simulation study of this paper). Note that the data collection process of other RL-based methods usually requires randomly and continuously varying the turbine yaw angles within the whole farm, which is not realistic and might lead to fatigue/damage to wind turbines. (2) The proposed control scheme can be pre-trained offline and fine-tuned online, providing an easy-to-implement solution with enhanced generality and flexibility for wind farms. These two points show that our method has strong applicability to real wind farms.
- High-fidelity simulations with SOWFA (simulator for onshore/offshore wind farm applications), a computational fluid dynamics (CFD) solver developed by the National Renewable Energy Laboratory (NREL) of US for the interaction between wind turbine dynamics and fluid flows [22], are conducted for performance validation. Specifically, large-eddy simulations for a wind farm in a 3 km × 3 km × 1 km flow field are carried out, where the turbine rotors are modelled by the actuator line method. We conduct 90

sets of 1000-second large-eddy simulations with SOWFA to collect offline training data for our RL algorithm built on Tensorflow. Each set of simulation requires around 44 hours' computation. The control scheme is then fine-tuned online with the Tensorflow-SOWFA pipeline. All the tests are carried out using 256 CPU cores on high-performance computing clusters.

- The proposed method can adapt to uncertain wind conditions. This is achieved via the specially designed reward regularization module, which relaxes the requirement of wind speed measurements and therefore brings strong robustness and adaptability to the whole wind farm control system compared with most existing model-free methods [13,14,20] for wake steering and power generation optimization of wind farms. In addition, the proposed method can also ensure the closed-loop yaw tracking and the exponential convergence of tracking errors under uncertainties of yaw actuators.

The remainder of this paper is organized as follows. The wind farm control problem is formalized in Section 2. Then the RL-based control method for wind-farm power optimization is designed in Section 3. Wind farm simulation models are introduced in Section 4. After that, high-fidelity closed-loop simulation results with SOWFA and Tensorflow are provided in Section 5. Finally, we conclude the paper in Section 6.

2. Problem formalization

The wind farm control problem is introduced in this section. Assume there are a total of n wind turbines in a wind farm, denoted by $\mathcal{WT}_1$, $\mathcal{WT}_2$, …, $\mathcal{WT}_n$, respectively. Then the steady-state power production of $\mathcal{WT}_i$ (denoted by E_i) follows [9–11,13,23,24]

$$E_i = \frac{1}{2}\rho A_i C_i(\alpha_i, \gamma_i) U_i^3 \tag{1}$$

where ρ is the air density, A_i is the rotor area of $\mathcal{WT}_i$, C_i is the power coefficient, and U_i is the wind speed in front of $\mathcal{WT}_i$. The power coefficient C_i is decided by the induction factor α_i and the yaw angle offset γ_i (with respect to the wind direction). Under the condition $\gamma_i = 0$, C_i satisfies

$$C_i(\alpha_i, 0) = 4\alpha_i(1-\alpha_i)^2 \tag{2}$$

In model-based methods, corrections on (2) are commonly employed to account for the effect of γ_i on C_i when $\gamma_i \neq 0$. A notable example is given in [9]: $C_i(\alpha_i, \gamma_i) = 4\alpha_i(1-\alpha_i)^2 \eta \cos(\gamma_i)^p$, where η and p are constants that should be decided by experiments.

Conventionally every turbine in a farm aims to maximize its own power outputs, called greedy control strategy. It leads to a game problem and the corresponding Nash equilibrium is $\alpha_i = 1/3$ and $\gamma_i = 0$, $i = 1, 2, \ldots, n$. As illustrated in Fig. 1b and c, the greedy strategy ignores the wake effects among turbines and cannot maximize the total power production of the wind farm, i.e.

$$E = \sum_{i=1}^{n} E_i = \sum_{i=1}^{n} \frac{1}{2}\rho A_i C_i(\alpha_i, \gamma_i) U_i^3 \tag{3}$$

In this paper, we aim to optimize E by wake steering. To be specific, the yaw angles of each turbine are continuously adjusted to mitigate the wake effects on downstream turbines and improve their outputs accordingly, and the induction factors of turbines remain the same as the greedy strategy.

Model-based control methods [9–11] usually build surrogate models for wake effects, aiming to calculate U_i based on the yaw setting $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_n]$ and the free-stream wind speed U_∞. In other words, they aim to analytically establish the mapping from γ and U_∞ to U_i, $i = 1, 2, \ldots, n$, and then use it to evaluate and optimize the total power output of the wind farm. However, as mentioned in the introduction, these methods are sensitive to unmodelled dynamics and system uncertainties, and in practice their performance can be quite different from the analytical results. To address this issue, in the next section, a model-free deep RL-based control strategy is designed to optimize E.

In real-time control, after the reference yaw signal γ_r (for each turbine $\mathcal{WT}_i$ in the farm) is provided by the RL algorithm, an additional controller is required to achieve precise yaw tracking. In this paper, we consider the yaw control system with the following Euler–Lagrange form:

$$M_i \dot{\omega}_i + C_i(\gamma_i, \omega_i)\omega_i + g_i(\gamma_i) = u_i \tag{4}$$

where γ_i and $\omega_i = \dot{\gamma}_i$ are the yaw angle and angular velocity of $\mathcal{WT}_i$, u_i is the control input, M_i is a positive constant, and C_i and g_i are dynamical terms of the system. We mention that the model in (4) can represent most of the yaw actuators such as the one used in the NREL Flow Analysis Software Toolkit (FAST) [25]. Besides, it allows a parameter affine representation for any $x, y, w, z \in \mathbb{R}$:

$$M_i z + C_i(x, y)w + g_i(x) = Y_i(x, y, z, w)\theta \tag{5}$$

where $Y_i(x, y, z, w) \in \mathbb{R}^{1\times h}$ is a regressor matrix, and $\theta \in \mathbb{R}^{h\times 1}$ is a constant parameter vector. In practical engineering, a common scenario is that one knows the lower/upper bounds of θ, i.e. $\theta_{q,\min} < \theta_q < \theta_{q,\max}$, where $\theta_{q,\min}, \theta_{q,\max} \in \mathbb{R}$ and θ_q is the qth entry of θ, while the accurate value of θ is unavailable for controller design. Based on these facts, a novel adaptive controller is also designed in the next section to achieve precise yaw tracking and parameter estimation for each turbine.

3. Deep RL-based wind farm control

A deep RL-based wind farm control method is proposed in this section. It contains three main parts. First, an RR module is designed to evaluate the normalized reward signals based on different yaw settings and wind conditions. Then it is embedded in DDPG to learn the optimal yaw settings. Finally, based on the outputs of RR-DDPG, a composite-learning based controller is designed to achieve precise parameter estimation and closed-loop yaw tracking. The main framework of the whole control system at both farm level and individual turbine level is demonstrated in Fig. 2.

3.1. Reward regularization

Based on the analysis in the previous section, the total power output E is related to the free-stream wind speed U_∞ in front of the wind farm, the yaw setting γ_i of every turbine, and also the time t, formalized by

$$E = f(\gamma, U_\infty, t) \tag{6}$$

where $\gamma = [\gamma_1, \gamma_2, \ldots \gamma_n]$. Due to the stochastic and dynamic features of wake effects, Eq. (6) is complicated and unknown for the controller design. Besides, it also results in severe issues that block directly employing E as the reward signal in deep RL algorithms: (1) U_∞ is time-varying and introduces biases into E. In other words, the instantaneous value of E cannot accurately reflect the effect of the corresponding yaw settings. (2) Wake effect is a dynamic phenomenon. This means E is not steady during the wake propagation process even under constant γ and U_∞. Therefore, directly employing E to build reward signals for deep RL renders the whole problem non-Markovian.

A reward regularization process is designed in this subsection to deal with these issues. As indicated by (1), though U_∞ is unavailable for controller design, the power outputs of front wind turbines (the most upstream turbines in the farm, denoted by F_i with a total number h) under the greedy control strategy can directly reflect the change of U_∞: $E_{F_i}^g = \frac{1}{2}\rho A_{F_i} C_{F_i}(\frac{1}{3}, 0) U_\infty^3$. This fact motivates us to employ $E_F^g = \sum_i^h E_{F_i}^g$ for power normalization purpose. Besides, instead of employing the instantaneous power data, we use the mean power output during a period to build reward signals. This design can effectively alleviate the influence induced by the wake propagation process, making the

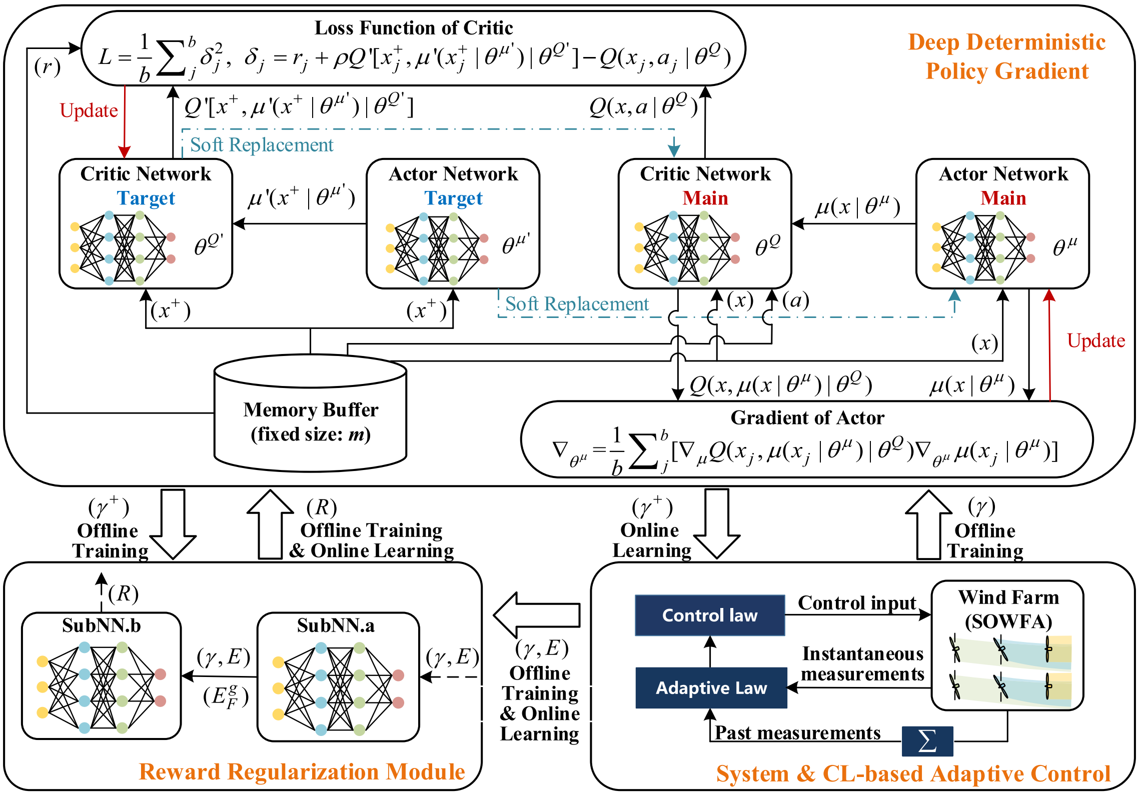

Loss Function of Critic
$L=\frac{1}{b}\sum_j^b\delta_j^2,\ \delta_j=r_j+\rho Q'[x_j^+,\mu'(x_j^+|\theta^{\mu'})|\theta^{Q'}]-Q(x_j,a_j|\theta^Q)$
Deep Deterministic Policy Gradient
(r)
Update
$Q'[x^+,\mu'(x^+|\theta^{\mu'})|\theta^{Q'}]$
$Q(x,a|\theta^Q)$
Soft Replacement
Critic Network Target
$\theta^{Q'}$
$\mu'(x^+|\theta^{\mu'})$
Actor Network Target
$\theta^{\mu'}$
Critic Network Main
θ^Q
$\mu(x|\theta^\mu)$
Actor Network Main
θ^μ
(x^+)
(x^+)
Soft Replacement
(x)
(a)
(x)
Update
$Q(x,\mu(x|\theta^\mu)|\theta^Q)$
$\mu(x|\theta^\mu)$
Memory Buffer (fixed size: m)
Gradient of Actor
$\nabla_{\theta^\mu}=\frac{1}{b}\sum_j^b[\nabla_\mu Q(x_j,\mu(x_j|\theta^\mu)|\theta^Q)\nabla_{\theta^\mu}\mu(x_j|\theta^\mu)]$
(γ^+) Offline Training
(R) Offline Training & Online Learning
(γ^+) Online Learning
(γ) Offline Training
(R)
SubNN.b
(γ,E)
(E_F^g)
SubNN.a
(γ,E)
(γ,E) Offline Training & Online Learning
Control law
Control input
Wind Farm (SOWFA)
Instantaneous measurements
Adaptive Law
Past measurements
Σ
Reward Regularization Module
System & CL-based Adaptive Control

Fig. 2. The main framework and data flow of the deep RL-based wind-farm control system. The main parts include the reward regularization module, the DDPG structure and the CL-based adaptive controller.

whole problem quasi-Markovian. Considering these aspects, we define the following regularized reward:

$$R=\frac{1}{t_a}\int_t^{t+t_a}\left[\frac{E(\tau)}{\kappa_f\sum_i^h E_{F_i}^g(\tau)}-\kappa_r\right]\mathrm{d}\tau \tag{7}$$

where t_a denotes the time period for averaging, and κ_f and κ_r are user-defined gains for weighting and offsetting purposes. It is noteworthy that the total power output $E(\tau)$ satisfies

$$E(\tau)=\sum_{i=1}^n E_i(\tau)$$

and here $E_i(\tau)$ denotes the power output of turbine $\mathcal{WT}_i$ at time τ. Since $E_i(\tau)$ can be directly measured by each turbine $\mathcal{WT}_i$ in the farm, $i=1,2,\dots,n$, $E(\tau)$ can be calculated by summing up all $E_i(\tau)$ together at any time τ. Then the regularized reward R can be calculated accordingly.

Following (7), a NN structure is employed to estimate R with offline data for arbitrary yaw settings. As shown in Fig. 2, it contains two sub NNs, SubNN.a and SubNN.b, and their designs are explained as follows.

(1) The input and output of SubNN.a are γ_{F_i} and $E_{F_i}/E_{F_i}^g$, respectively, where E_{F_i} denotes the power production of $\mathcal{WF}_{F_i}$ under the yaw setting γ_{F_i}. Therefore, the core function of SubNN.a is to evaluate the influence of the yaw setting to the power output. Based on SubNN.a, one can evaluate $E_{F_i}^g$ with γ_{F_i} and E_{F_i} under unknown wind conditions and then calculate R accordingly.

(2) Based on the data set $\{\gamma,R\}$, SubNN.b carries out supervised learning (with γ as the input and R as the output label) to estimate the regularized reward under any yaw setting.

3.2. Deep deterministic policy gradient

Generally, a reinforcement learning agent learns from its interactions with environments rather than being directly guided, and it aims to iteratively improve its actions by judiciously evaluating past experience (exploitation) and making new decisions (exploration). Many deep RL algorithms, e.g. DDPG [17], employ the so-called actor–critic structure [15] as the main RL structure, with deep neural networks as information processors and universal approximators. Particularly, the critic network aims to capture core system information and estimate a long-term reward function. In parallel, the actor aims to optimize the reward function via the policy gradient strategy and improve the control policy iteratively. In this subsection, the RR module is embedded in the DDPG algorithm to optimize the yaw settings and the corresponding power production of the wind farm. The main mechanism of the DDPG algorithm and the training principle of its actor–critic structure are introduced first. After that, the wind farm control problem is moulded into DDPG.

In general, DDPG aims to maximize the following cumulative reward for an unknown system $x(k+1)=F(x(k),u(k))$:

$$V_u(x(k))=\sum_{i=k}^{\infty}\beta^{i-k}r(x(i),u(x(i))) \tag{8}$$

Here $x(k)$ and $x(k+1)$ respectively denote an instant state (at time step k) and its successor state, u is the control input, r is the reward function, and β is the discount factor. The optimal control policy u^* satisfies

$$u^*=\arg\max_u\{V_u(x(k))\} \tag{9}$$

and we denote the corresponding maximal cost function as $V^*(x(k))=V_{u^*}(x(k))$. From (8) and (9), an important property of $V^*(x(k))$ is

$$V^*(x(k))=r(x(k),u^*(x(k)))+\beta V^*(x(k+1)) \tag{10}$$

We note that it is very challenging to solve V^* and u^* analytically, especially when the system model is nonlinear and unknown. Alternatively, this problem can be addressed by iterative learning [15,26]. A notable example is the so-called Q-learning strategy [26], which can iteratively approximate V^* and u^* through policy evaluation and policy improvement.

Policy Evaluation: Solve the unknown Q-function $Q^{(i)}(x(k),a)$ by the following equation,

$$Q^{(i)}(x(k),a) = r(x(k),a) + \beta Q^{(i)}(x(k+1),u^{(i)}) \tag{11}$$

Policy Improvement: Update control policy by

$$u^{(i+1)} = \arg\min_{a} Q^{(i)}(x,a) \tag{12}$$

with $u^{(0)}$ being an admissible control policy.

In (11), $Q^{(i)}(x(k),a)$ is called the action-state value function which represents the long-term reward when action a is taken at state $x(k)$ and a control policy $u^{(i)}$ is pursued thereafter. Under (11) and (12), it has been proved that $Q^{(\infty)}(x(k),a) \to r(x(k),a) + \beta V^*(x(k))$ and $u^{(\infty)} \to u^*$.

However, directly implementing (11) and (12) still leads to unacceptable computational complexities, especially when the system dimension is high. To address this issue, one can employ NNs to approximate Q^* and u^*, which is the main principle of DDPG. Specifically, DDPG has an actor–critic structure, in which the critic and the actor are employed to estimate the action-state value function Q^* and the optimal control policy u^*, respectively. To enhance the learning performance, two sets of actor–critic are employed, i.e. the main actor–critic and the target actor–critic. We denote the parameters of the main actor, main critic, target actor and target critic by θ^{μ}, θ^{Q}, $\theta^{\mu'}$ and $\theta^{Q'}$, respectively. Their outputs are μ, Q, μ' and Q', respectively.

The main structure and training process of DDPG is introduced in Fig. 2, where a fixed-size memory buffer $\mathcal{M}$ (a queue structure with a size of m) is employed to store the previous experiences (in term of transactions) of the system. We denote the transactions stored in $\mathcal{M}$ as $\{(x_i,a_i,x_i^+,r_i)\}_{i=1,2,\dots,m}$, where x_i and x_i^+ respectively denote a stored state and its successor state (i.e. $x(k)$ and $x(k+1)$), and a_i and r_i are respectively the corresponding control action and reward at this time step. Then, in the training process, a small batch of transitions (with a size of b) are selected randomly from $\mathcal{M}$ at each step.

In the training process, we use the loss function $L = \frac{1}{b}\sum_{j=1}^{b}\delta_j^2$ to update the main critic network, where δ_j is the temporal-difference (TD) error of the jth sampled transition:

$$\delta_j = r_j + \beta Q'[x_j^+, \mu'(x_j^+|\theta^{\mu'})|\theta^{Q'}] - Q(x_j,a_j|\theta^{Q}) \tag{13}$$

This design is based on the property in (11). The reference value of $Q(x_j,a_j|\theta^Q)$ (i.e. $r_j + \beta Q'[x_j^+, \mu'(x_j^+|\theta^{\mu'})|\theta^{Q'}]$) is constructed by the outputs of the target networks.

Following (12), the main actor aims to find the optimal control policy based on Q. This is achieved by the policy gradient strategy. Specifically, at each training step, the gradient of Q with respect to θ^{μ} is

$$\begin{aligned}\nabla_{\theta^{\mu}} &= \frac{1}{b}\sum_{j=1}^{b}\frac{\partial Q(x_j,\mu(x_j|\theta^{\mu})|\theta^{Q})}{\partial\theta^{\mu}}\\ &= \frac{1}{b}\sum_{j=1}^{b}[\nabla_{\mu}Q(x_j,\mu(x_j|\theta^{\mu})|\theta^{Q})\cdot\nabla_{\theta^{\mu}}\mu(x_j|\theta^{\mu})]\end{aligned} \tag{14}$$

Moreover, the target critic and actor slowly track their main counterparts via soft replacement:

$$\theta^{\mu'} \leftarrow (1-\tau)\theta^{\mu'} + \tau\theta^{\mu},\ \theta^{Q'} \leftarrow (1-\tau)\theta^{Q'} + \tau\theta^{Q} \tag{15}$$

where $\tau \in (0,1]$ is a user-defined constant. Recalling (13), this design can help break the data correlations and enhance the overall training stability.

Based on these design principles of DDPG, we are ready to mould the wind farm control problem into it. Specifically,

- We employ the regularized power evaluated by the RR module as the reward signal, i.e. $r = R$. As discussed in Section 3.1, this design can handle the non-Markovian issue induced by wake delays.
- For the wind farm control problem considered in this paper, the system state in DDPG is the yaw setting vector $\gamma = [\gamma_1,\gamma_2,\dots\gamma_n]$ of the whole farm, i.e. $x = \gamma$.
- The control action a in DDPG is the change of the yaw angle vector γ at every time step, which is bounded by a positive constant b_a.

Remark 1. As indicated in Fig. 2, a memory buffer $\mathcal{M}$ is employed in the deep RL algorithm to store previous experiences and collect new experiences (in term of transactions) of the system. This memory buffer has a first-in-first-out queue structure and a fixed size (denoted by m). At every learning step, a small batch of transitions (with a size of b) are uniformly randomly sampled from $\mathcal{M}$ to carry out deep NN training. This sampling strategy is called experience replay [16,17], which can break the temporal correlation of sequential transitions, catering to the independent and identical distribution requirement in deep neural network training and therefore enhancing learning stability.

3.3. Adaptive yaw tracking via composite learning

The output of RR-DDPG provides the reference yaw setting γ_r for each wind turbine in the farm. As the final part of our control scheme, a composite learning (CL)-based adaptive controller is designed in this subsection to achieve yaw tracking. Compared with relevant studies [27–30], our method can ensure that the estimates of system parameters are always within pre-determined bounds by employing a specially designed projection law, which enhances the generality of our CL-based controller.

For any wind turbine $\mathcal{WF}_i$ in the farm, we denote its reference yaw angle and angular velocity by γ_r and ω_r, respectively. Then we define the error yaw angle and angular velocity of $\mathcal{WF}_i$ by $\gamma_e = \gamma_i - \gamma_r$ and $\omega_e = \omega_i - \omega_r$, respectively. Recall (4), one has

$$\dot{\gamma}_e = \omega_e,\ \ M_i\dot{\omega}_e = -Y_i(\gamma_i,\omega_i,\dot{\omega}_r,\omega_i) + u_i \tag{16}$$

where $Y_i(\gamma_i,\omega_i,\dot{\omega}_r,\omega_i)$ is the regressor matrix that follows $Y_i(\gamma_i,\omega_i,\dot{\omega}_r,\omega_i)\theta = M_i\dot{\omega}_r + C_i(\gamma_i,\omega_i)\omega_i + g_i(\gamma_i)$. Hereafter, for the sake of brevity, arguments of matrix functions will be ignored. We note that though θ is unknown, Y_i is available for controller design, and its specific expression can be deduced by taking Jacobian of the right-hand side of (6) with respect to θ.

As mentioned in Section 2, we aim to achieve three objectives: (1) Estimating the unknown parameter vector θ; (2) During the estimation process, keeping the estimate $\hat{\theta}$ always within predetermined bounds i.e. $\theta_{q,\min} < \hat{\theta}_q < \theta_{q,\max}$ ($\hat{\theta}_q$ is the qth entry of $\hat{\theta}$, $q = 1,2,\dots,h$); (3) Based on (1) and (2), achieving precise yaw tracking, i.e. $\gamma_e \to 0$.

To achieve these objectives, a novel adaptive control method with composite learning is proposed. First, we consider the following projection law:

$$\hat{\theta}_q = (\theta_{q,\max} - \theta_{q,\min})\mathrm{sig}(\hat{\psi}_q) + \theta_{q,\min} \tag{17}$$

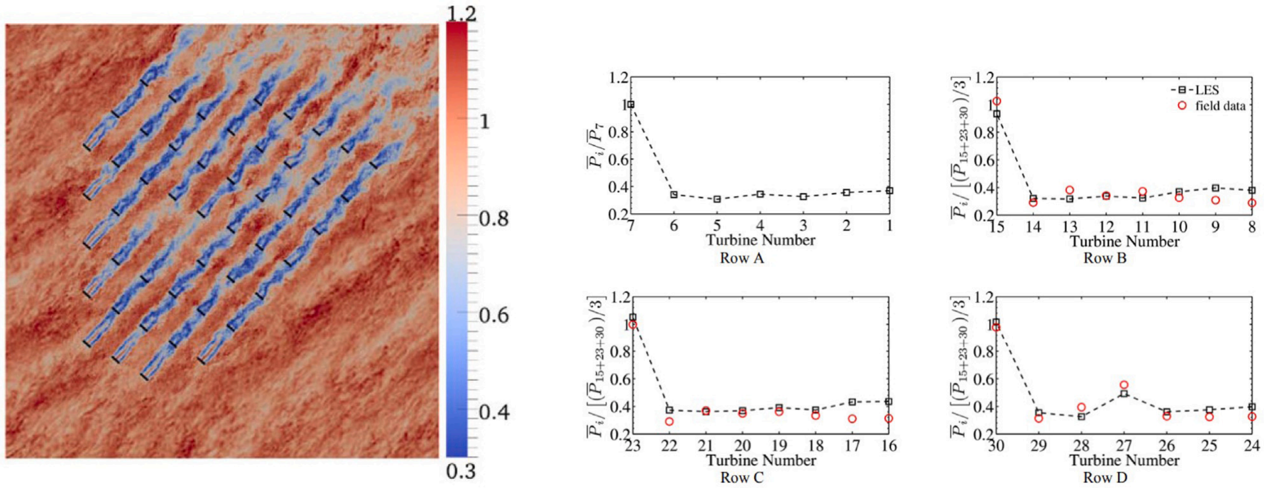
where $\mathrm{sig}(\cdot) : \mathbb{R} \to (0,1)$ is the sigmoid function, defined by $\mathrm{sig}(x) = 1/(1+\mathrm{e}^{-\kappa x}), x \in \mathbb{R}$, and here we set $\kappa = 1$ without loss of generality. With (17), the estimation problem of θ_q in $(\theta_{q,\min},\theta_{q,\max})$ is transferred to the estimation problem of a projected value ψ_q over the entire real domain, and we denote $\hat{\psi} = [\hat{\psi}_1,\hat{\psi}_2,\dots,\hat{\psi}_h]^{\mathrm{T}}$ as the estimate vector of $\psi = [\psi_1,\psi_2,\dots,\psi_h]^{\mathrm{T}}$.

Based on these preliminaries, the CL-based adaptive controller is summarized in the following theorem.

Theorem 1. *Considering the tracking model in (16) and the projection law in (17), design the control and adaptive laws as follows*

$$u_i = -k_p\gamma_e - k_v\omega_e + Y_i\hat{\theta} \tag{18}$$

$$\dot{\hat{\psi}} = -k_yY_i^{\mathrm{T}}\omega_e - k_yk_w\sum_{l=1}^{L}[W_i^{\mathrm{T}}(t_l)W_i(t_l)\hat{\theta} - W_i^{\mathrm{T}}(t_l)u_i(t_l)] \tag{19}$$



(a) Flow field simulation

(b) Normalized power productions of wind turbines

Fig. 3. Validation results of SOWFA from [31].

where k_p, k_v, k_y and k_w are user-defined positive constants, W_i is a regressor matrix that satisfies $W_i\theta = M_i\dot{\omega}_i + C_i\omega_i + g_i$. Besides, t_l, $l = 1, 2, \ldots, L$, denotes some selected past-time points in real-time control, and thus $\sum_{l=1}^{L}[W_i^{\mathrm{T}}(t_l)W_i(t_l)\hat{\theta} - W_i^{\mathrm{T}}(t_l)u_i(t_l)]$ stores the historical information of the corresponding variables. Then, the tracking error γ_e and ω_e converge to zero asymptotically, and the parameter estimate $\hat{\theta}_q$ is always within the predetermined bound, i.e. $\hat{\theta}_q \in (\theta_{q,\min}, \theta_{q,\max})$. Moreover, if proper online data are collected such that $\sum_{l=1}^{L} W_i^{\mathrm{T}}(t_l)W_i(t_l)$ is full-rank, then γ_i, ω_i and $\tilde{\theta}$ exponentially converge to zero. Here $\tilde{\theta} = \hat{\theta} - \theta$ denotes the parameter estimation error.

Proof. See Appendix.

Remark 2. The deep RL-based wind farm control system developed in this paper has the ability to handle the uncertainties in real-time operating conditions of wind farms. Firstly, the ‘pre-trained offline, fine-tuned online’ mechanism ensures a stable learning process and enables our control system to adapt to the mismatch between the offline training environment and the real-time operating environment. Secondly, the specially designed reward regularization module can evaluate the greedy-mode power outputs of the front wind turbines in the farm and employ them to normalize the reward function. This allows the whole algorithm to have adaptability and robustness to the uncertain wind conditions in practical use. Finally, as strictly proved in Theorem 1, the composite-learning adaptive yaw controller proposed in this paper can achieve high-performance real-time yaw tracking for each turbine even in the presence of parameter uncertainties.

4. Wind farm simulation models

In this work, we employ the high-fidelity computational fluid dynamics (CFD) model SOWFA (Simulator for Onshore/Offshore Wind Farm Applications) [22] developed by the National Renewable Energy Laboratory (NREL) of US to carry out simulations and test our deep reinforcement learning-based control algorithm. Before presenting the simulation results in the next section, we discuss the capability of SOWFA and compare it with other wind farm simulation models in this section. It should be emphasized that the deep reinforcement learning-based wind farm control scheme proposed in this paper is data-driven. It is independent of the testing/running environments and does not require any analytical wake model or relevant parameters/conditions to carry out its learning process. These important features are always valid, no matter in simulation tests or in real operating conditions of wind farms, and SOWFA is employed in this paper only for high-fidelity numerical test and performance evaluation purposes instead of algorithm designs.

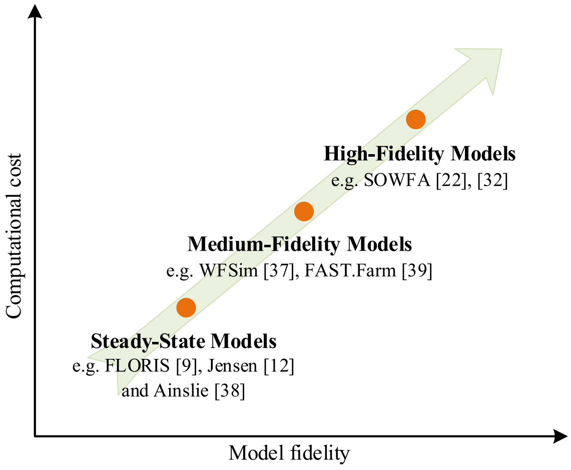


Fig. 4. Illustration of different wind farm models.

SOWFA is able to perform large-eddy simulations (LESs) [32] for wind fields, in which the wind turbines are modelled by the so-called actuator line approach [33] coupled with the FAST (Fatigue, Aerodynamics, Structures and Turbulence) toolkit [25] from NREL. As one of the “best practices” methodologies and most popular models for performing wind-farm LESs, SOWFA has been widely validated and applied in various studies. Some notable examples include wind farm designs and layout optimizations [34], analysis of wind-farm flow fields [35], wind farm control validations [9,36] and model validations [9,37]. Particularly, Ref. [31] validated SOWFA with the actual field data measured at the Lillgrund Wind Farm in Öresund, Sweden. Their main results are given in Fig. 3 (Figs. 5 and 7 from [31]) for easy reference. From Fig. 3b, one can see that SOWFA performs very well in evaluating the power generation of wind turbines in the farm, rendering a root mean square error less than 0.1 (in terms of normalized power productions). It is noteworthy that comparing simulation results with actual field data is very challenging, and public data of wind farm fields are often binned by sparse wind directions and averaged over a very long time [31]. Even under such harsh conditions, the validation results in [31] still successfully show the high-fidelity and high-accuracy features of SOWFA.

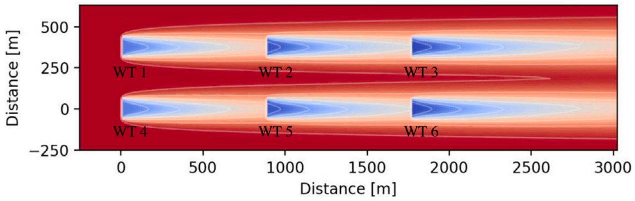


Fig. 5. Simulation result of a six-turbine wind farm with FLORIS.

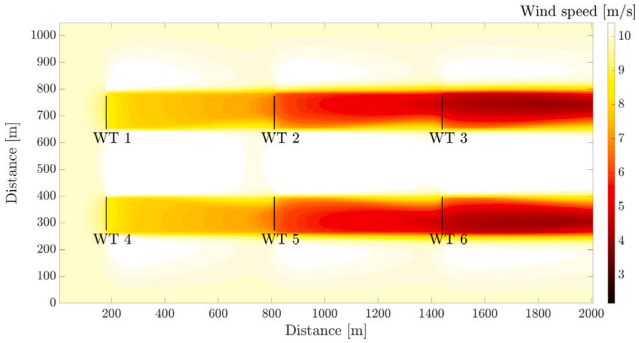


Fig. 6. Simulation result of a six-turbine wind farm with WFSim.

Generally, models for wind farm simulations can be divided into the following categories as illustrated in Fig. 4:

- *Steady-state models:* This type of models usually evaluates the time-averaged features of flow fields and wind farms while ignoring temporal dynamics, e.g. wake meandering. Such a design principle can greatly reduce computational cost. But it also limits the accuracy and fidelity of steady-state models. Examples of popular steady-state models include FLORIS [9], Jensen [12] and Ainslie [38].
- *Medium-fidelity models:* This type of models aims to make a trade-off between computational complexity and fidelity. For example, a recently developed model WFSim [37] employs two-dimension Navier–Stokes equations instead of the full three-dimension Navier–Stokes equations (as employed by SOWFA) to carry out medium-fidelity LESs for wind farms. This enables WFSim to have less computational complexity than SOWFA while achieving better accuracy than steady-state models. But its fidelity and accuracy are much lower than SOWFA. Another recent example of medium-fidelity wind farm models is FAST.Farm [39].
- *High-fidelity models:* As mentioned earlier, high-fidelity simulation models, such as SOWFA [22,32], usually conduct full-dimension LESs with high spatial and temporal resolutions. They have the highest level of accuracy and computational complexity. Such computationally costly simulations allow the users to get detailed results that reflect the actual wind fields well.

We choose three most-recent and popular models, one from each category, for comparisons. They are FLORIS [9] (steady-state), WFSim [37] (medium-fidelity), and SOWFA (high-fidelity) [22,32].

Firstly, the simulation result of a typical six-turbine farm with FLORIS is given in Fig. 5. One can see that FLORIS only evaluates the steady wakes and cannot provide details and temporal dynamics of the flow field.

Secondly, we carry out simulation for a six-turbine wind farm with WFSim. The result is illustrated in Fig. 6. One can see that WFSim can provide more information than FLORIS. It shows more details of the whole flow field with a medium resolution.

Finally, the simulation result with SOWFA is provided in Fig. 7. It is obvious that SOWFA can provide much more flow-field details than FLORIS and WFSim. Even small wind gusts are illustrated in Fig. 7 with a high resolution. It is also noteworthy that SOWFA carries out three-dimension LESs. The two-dimension instantaneous profile in Fig. 7 is employed just for the ease of comparison with FLORIS and WFSim. A three-dimension flow-field simulation domain with SOWFA is illustrated in Fig. 8.

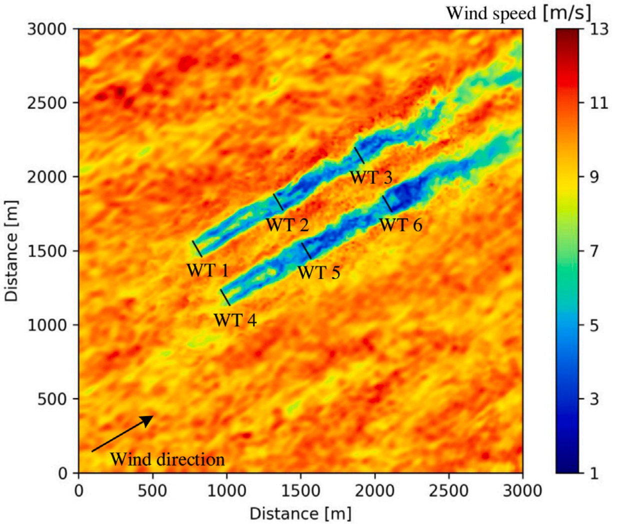


Fig. 7. A 2D Illustration of the instantaneous flow field and wind farm simulated by SOWFA.

Due to the above merits, SOWFA has been a commonly-recognized benchmark for wind farm model validations. For example, FLORIS employed LES data generated by SOWFA to calibrate its parameters. It was shown in [9] that, after calibrations, FLORIS could achieve an around 5% mean absolute error in averaged power outputs when compared with SOWFA. Moreover, Ref. [37] compared the wind speeds of WFSim and SOWFA at the mean flow centrelines. They showed that the mean absolute error between these two models was about 1 m/s at the mean flow centrelines. Both FLORIS and WFSim employed these quantitative validation results with respect to SOWFA as their core evidence to show the accuracy, feasibility and applicability of their modelling methods. One can refer to [9,37] for detailed quantitative validation and comparison results.

All these facts, along with both qualitative and quantitative analysis, show that the SOWFA simulation model to be employed in the case studies of this paper is state-of-the-art and has essential merits compared with many other popular models in the literature. The high-fidelity and high-accuracy features of SOWFA can significantly enhance the applied value of our deep reinforcement learning-based wind farm control method.

5. Numerical simulations

Based on the discussion in Section 4, SOWFA and TensorFlow are employed to achieve high-fidelity wind farm simulations and test our deep RL-based wind farm control method in this section. As illustrated in Fig. 8, a 3 km × 3 km × 1 km flow field is considered. In order to capture the detailed turbine wake dynamics, meshes of 3 m × 3 m × 3 m are used around the turbine rotors, and the time step is set as 0.02 s. A 2-dimension visualization of a instantaneous flow field is given in Fig. 7. We consider a scenario in which six NREL 5MW wind turbines (with a 3 × 2 layout) are embedded in this flow field, which are denoted by black bars in the figure. The rotor diameter of these turbines is $D = 126.4$ m. The distance between the turbines in the same row is $5D$, and the distance between the two rows is $3D$. Fig. 7 also illustrates the simulation result under the greedy mode (i.e. $\gamma_i \equiv 0$, $i = 1, 2, \ldots, 6$).

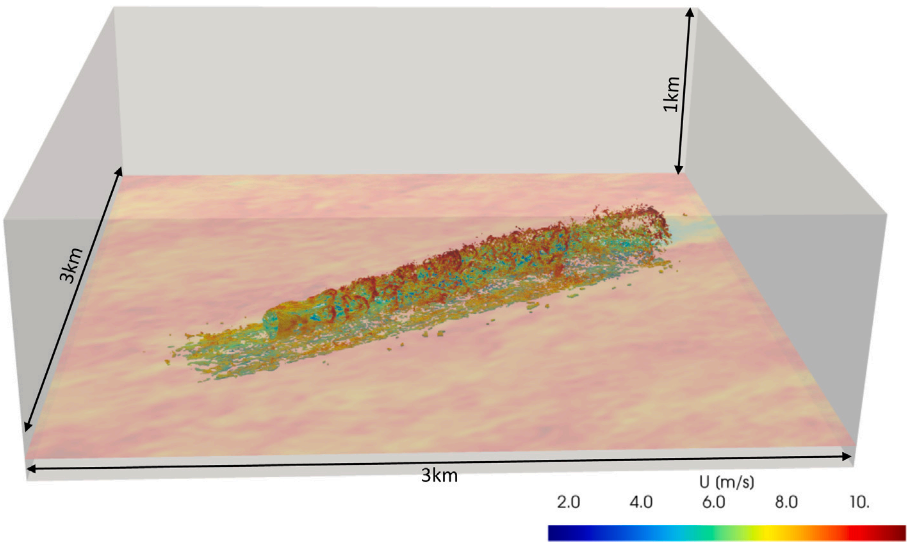


Fig. 8. An Illustration of the 3D flow-field simulation domain with SOWFA. It shows a typical instantaneous vorticity contour coloured by velocity magnitude and also shows the hub-height horizontal plane.

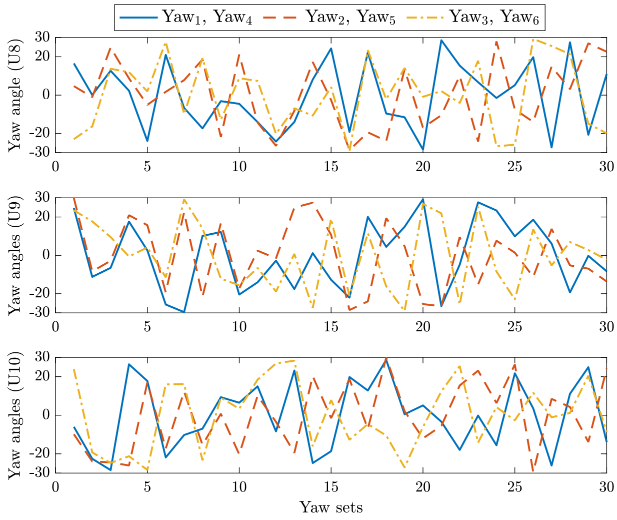
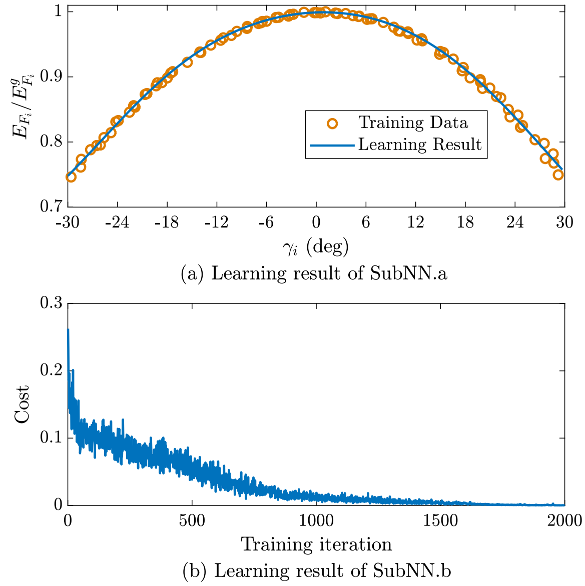


Fig. 9. The specific yaw settings for data collection.



(a) Learning result of SubNN.a

(b) Learning result of SubNN.b

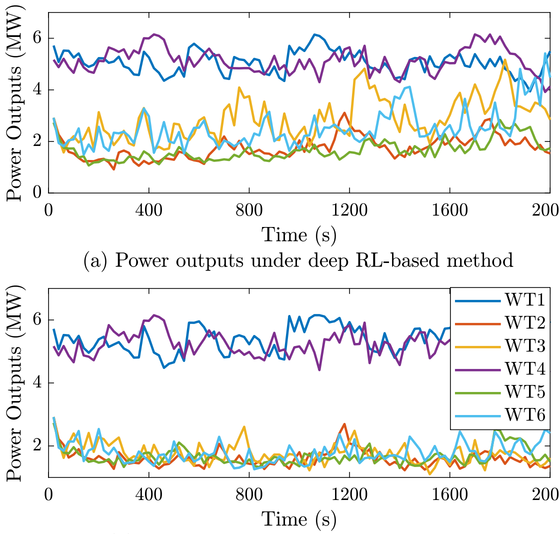
Fig. 10. Learning results of the RR module. (a) Learning performance of SubNN.a. (b) Learning performance of SubNN.b.

It clearly shows that the downstream turbines are severely affected by the wakes induced by the upstream turbines.

We employ the proposed deep RL-based control method to mitigate this problem and improve power generation. Given the wind-farm layout considered here, the wake effect between different rows is negligible. Therefore one can consider the wake steering problem for a single row of turbines and employ identical yaw settings for other rows. This strategy can significantly reduce the whole algorithm's computational complexity, which has been utilized in many relevant studies such as [9].

To generate the training data for our deep RL-based wind farm control method, we conduct 90 sets of 1000-second large-eddy simulations with SOWFA. These simulation sets are evenly divided into three groups (30 sets in each group) according to different free-stream mean wind speeds (8 m/s, 9 m/s and 10 m/s, respectively). Each set of simulation requires around 44 hours' computation using 256 CPU cores. In each simulation, the yaw settings of all turbines are decided within $[-30, 30]$ deg using Latin hypercube sampling method. From a standpoint of practical engineering, collecting data by randomly and continuously varying the turbine yaw angles (like other RL-based methods usually do) is not realistic and might lead to fatigue/damage to wind turbines. Thus in each 1000-second simulation, the turbines' yaw settings are fixed. The specific yaw settings are shown in Fig. 9. The greedy-mode simulations are also conducted for comparison purposes. The following results will show that our deep RL-based wind farm control method can achieve power generation optimization under such a sparse dataset collected by SOWFA. These results indicate that our method can use limited sets of actual wind farm data for algorithm training and learning purposes, and has strong applicability to real wind farms.

The training dataset is fed into the RR module. The SubNN.a and SubNN.b in the RR module are two fully connected NNs, and their neural structures are 1-20-20-1 and 3-32-32-1, respectively. Besides, we set $\kappa_f = 1$ and $\kappa_r = 1.5$. As mentioned in Section 3.1, SubNN.a aims to evaluate how the power output changes with the changes of yaw angles. It employs sigmoid functions as the activation functions in hidden layers, and its learning result is illustrated in Fig. 10a. One



(a) Power outputs under deep RL-based method

(b) Power outputs under greedy strategy

Fig. 11. Power outputs of wind turbines. (a) Results of the proposed deep RL-based method. (c) Results of the greedy strategy.

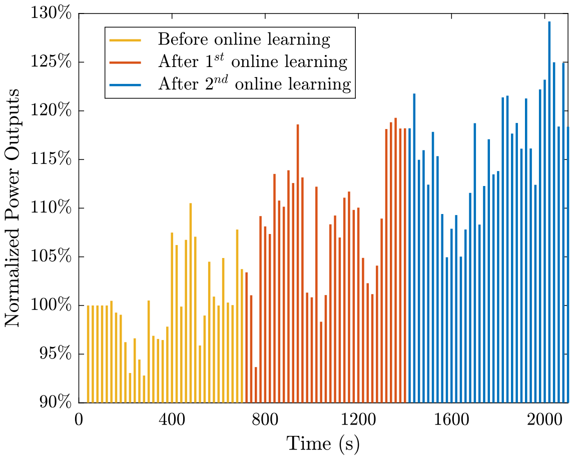


Fig. 12. Normalized power outputs of the wind farm. Different line colours are employed to indicate different online learning procedures.

can see that SubNN.a can successfully approximate the mapping from γ_{F_i} to $E_{F_i}/E^g_{F_i}$ even the training data are collected under different wind conditions. Based on the results of SubNN.a, we can calculate the regularized reward R for all training data and then carry out supervised learning for SubNN.b. We employ the relu functions as the activation functions in hidden layers, and the training error is given in Fig. 10b under the learning rate 0.01. One can see that the training error becomes negligible after 2000 iterations.

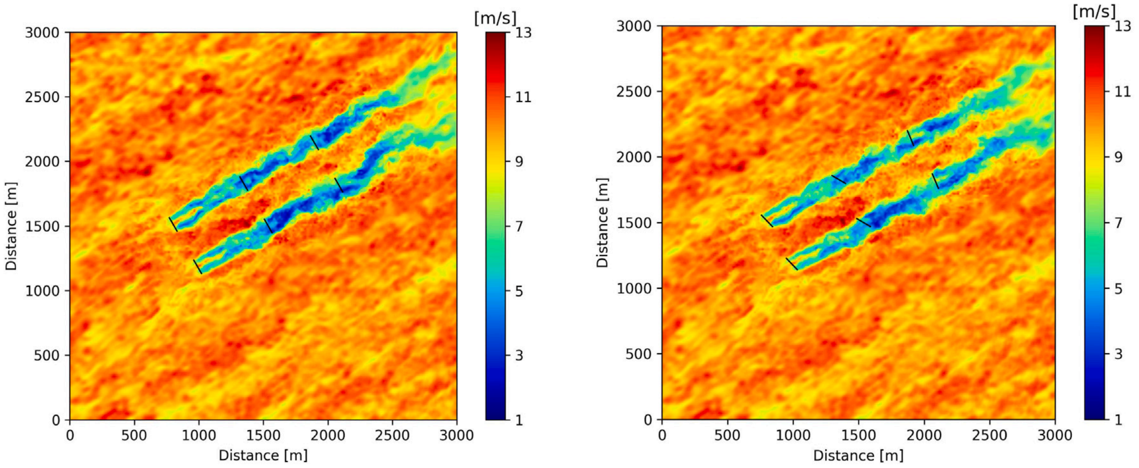
Then we embed the RR module in DDPG to optimize the yaw settings of the whole wind farm. The critic and actor networks are three-layer fully connected NNs with 32 neurons in hidden layers and employ the relu functions as the activation functions. The main critic and actor's learning rates are 0.001 and 0.005, respectively, and the soft replacement rate of target critic and actor is set to be $\tau = 0.01$. The size of the memory buffer $\mathcal{M}$ is set to be $m = 2000$, and the size of the sampling batch is $b = 32$. This indicates, following the experience replay strategy [16,17], 32 transitions are sampled randomly from $\mathcal{M}$ for NN training at each learning step. Other parameters in our algorithm include $\beta = 0.9$ and $b_a = 1$ deg.

Our RR-DDPG is pre-trained offline and fine-tuned online. Specifically, 10000-step training is carried out offline, and then the performance of RR-DDPG is tested and improved online with SOWFA. The online learning is triggered every 700 s (i.e. t_a = 700 s) to mitigate the stochastic features of wind conditions. Once the online learning is triggered, the mean power outputs over the last 700 s and the yaw settings of all the turbines are employed to update the whole algorithm. Then the new yaw settings obtained by the RR-DDPG are applied to the wind farm until the next learning process is triggered. Moreover, to achieve yaw tracking in online testing, we consider the yaw actuator model from the NREL FAST [25]: $Y_I\dot{\omega}_i + Y_D\omega_i + Y_S\gamma_i = u_i$, where Y_I, Y_D, and Y_S are respectively the inertia, torsional damping constant, and torsional spring stiffness. Their true values are chosen to be $Y_I = 38.9$ kg m^2, $Y_D = 102.2$ N m s and $Y_S = 85.9$ N m, which are unknown to the CL-based adaptive controller. Besides, their estimation bounds are set to be $Y_{I,\min} = 20$, $Y_{I,\max} = 50$, $Y_{D,\min} = 60$, $Y_{D,\max} = 150$, $Y_{S,\min} = 50$, and $Y_{S,\max} = 100$. The parameters of the yaw tracking controller are $k_p = 4$, $k_d = 4$, $k_y = 10$, and $k_w = 20$.

Under all these settings, a 2000-second simulation is conducted with SOWFA, in which the online learning process is triggered 2 times. Besides, the greedy-mode simulation under identical wind conditions is also conducted for comparison purposes. The power production of every turbine in the farm and the relative total power production (with respect to the greedy mode) are illustrated in Figs. 11 and 12, respectively. Before the changes of wake effects under new yaw settings are fully propagated (around the first 400 s in the simulation), the power output is decreased due to yaw offsets. After that, we can see that the proposed RL-based control method leads to clear power increases when compared with the greedy strategy. Its performance is furthered improved as the online learning is conducted. Specifically, after the second online learning process is finished, the proposed method can lead to a significant increase (around 15% on average) on the farm's total power generation. The final yaw settings at 2000 s are 22 deg for turbines 1 and 4, 24 deg for turbines 2 and 5, and −1 deg for turbines 3 and 6. Finally, the flow fields at the 700 s, 1400 s, and 2000 s are provided in Figs. 13–15, respectively, in comparison with the greedy-strategy cases. All these results show that our method successfully achieves power generation optimization and wake steering for wind farms.

6. Conclusions

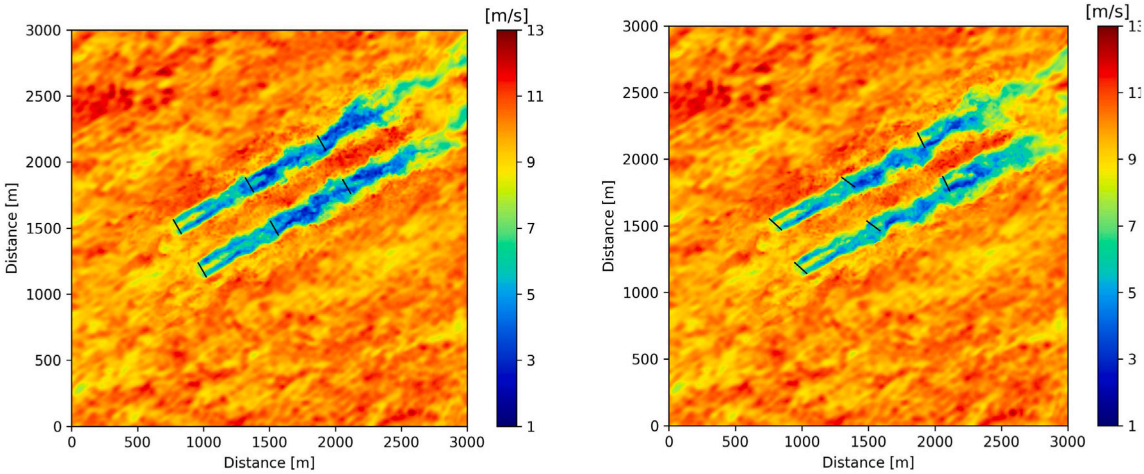
A novel deep reinforcement learning (RL)-based control method has been developed to optimize the total power production of wind farms through wake steering. A special reward regularization module was designed to estimate wind turbines' normalized power outputs under different yaw settings and stochastic wind conditions. It was then embedded in the deep deterministic policy gradient algorithm to optimize the yaw settings of all turbines in the farm. Our deep RL-based wind farm control method is data-driven and model-free, which addresses the limitations of current wind farm control methods, including reliance on accurate analytical/parametric models and lack of adaptability to uncertain wind conditions. Also, a novel composite learning-based controller was designed to achieve accurate closed-loop yaw tracking under uncertainties of yaw actuators. High-fidelity simulations were carried out for performance validation. We conducted 90 sets of 1000-second large-eddy simulations with SOWFA to collect offline training data for our RL algorithm built on Tensorflow. Each set of simulation required around 44 hours' computation using 256 CPU cores. The control scheme was then fine-tuned online with the Tensorflow-SOWFA pipeline. Results showed that our method could significantly improve the wind farm's total power production by 15% on average compared with the benchmark. The method proposed in this paper is application-oriented. It only needs sparse training datasets that are easy-to-collect in actual wind farm applications. Moreover, it can be pre-trained offline and fine-tuned online, rendering strong adaptability to uncertain environments and providing an easy-to-implement wind farm control solution with enhanced generality and flexibility.



(a) Greedy strategy

(b) Deep RL-based control strategy

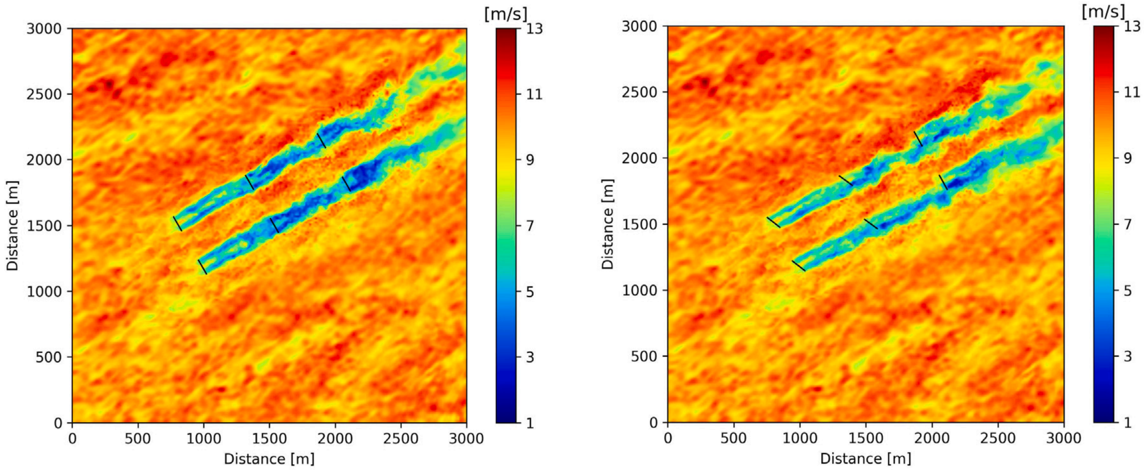
Fig. 13. Simulation results of the flow fields at 700 s. (a) The greedy strategy. (b) The proposed deep RL-based control strategy.



(a) Greedy strategy

(b) Deep RL-based control strategy

Fig. 14. Simulation results of the flow fields at 1400 s. (a) The greedy strategy. (b) The proposed deep RL-based control strategy.



(a) Greedy strategy

(b) Deep RL-based control strategy

Fig. 15. Simulation results of the flow fields at 2000 s. (a) The greedy strategy. (b) The proposed deep RL-based control strategy.

CRediT authorship contribution statement

Hongyang Dong: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Visualization, Writing - original draft. **Jincheng Zhang:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization. **Xiaowei Zhao:** Conceptualization, Funding acquisition, Formal analysis, Investigation, Methodology, Project administration, Resources, Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has received funding through the UK Engineering and Physical Sciences Research Council (grant numbers: EP/R007470/1 and EP/S001905/1). The second author was funded by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 765579. The authors also acknowledge the Scientific Computing Research Technology Platform (SCRTP) at the University of Warwick for providing High-Performance Computing resources.

Appendix. Proof of Theorem 1

We start with the following storage function

$$\begin{aligned} V_a =& \frac{1}{2}(k_p\gamma_e^2 + M_i\omega_e^2) \\ &+ \frac{1}{k_y}\sum_{q=1}^{h}(\theta_{q,\max} - \theta_{q,\min})[\tilde{\psi}_q + \ln(1+\mathrm{e}^{-(\tilde{\psi}_q+\psi_q)}) \\ &- \tilde{\psi}_q \mathrm{sig}(\psi_q) - \ln(1+\mathrm{e}^{-\psi_q})] \end{aligned} \tag{A.1}$$

where $\tilde{\psi}_q$ is the qth entry of $\tilde{\psi}$ with $\tilde{\psi} = \hat{\psi} - \psi$. Here ψ denotes the true projected value of θ. We mention that V_a is a valid Lyapunov function candidate. To show this fact, we define $H(\tilde{\psi}_q) = (\theta_{q,\max} - \theta_{q,\min})[\tilde{\psi}_q + \ln(1+\mathrm{e}^{-(\tilde{\psi}_q+\psi_q)}) - \tilde{\psi}_q\mathrm{sig}(\psi_q) - \ln(1+\mathrm{e}^{-\psi_q})]$. Then the gradient of H with respect to $\tilde{\psi}_q$ follows

$$\nabla_{\tilde{\psi}_q} H = (\theta_{q,\max} - \theta_{q,\min})[\mathrm{sig}(\tilde{\psi}_q + \psi_q) - \mathrm{sig}(\psi_q)] = \tilde{\theta}_q \tag{A.2}$$

where $\tilde{\theta}_q$ is the qth entry of $\tilde{\theta}$ with $\tilde{\theta} = \hat{\theta} - \theta$. Therefore, $\nabla_{\tilde{\psi}_q} H$ is monotonously increasing, which indicates that $\tilde{\psi}_q = 0$ is the global minimizer of H and that the corresponding minimal value is $H(0) = 0$. Thus V_a is a valid Lyapunov function candidate. Substituting (18) and (19) into the time derivative of V_a yields

$$\begin{aligned} \dot{V}_a &= -k_v\omega_e^2 + \omega_e Y_i \tilde{\theta} + \frac{1}{k_y}\tilde{\theta}^{\mathrm{T}}\dot{\tilde{\psi}} \\ &= -k_v\omega_e^2 - k_w\tilde{\theta}^{\mathrm{T}}\sum_{l=1}^{L}[W_i^{\mathrm{T}}(t_l)W_i(t_l)\hat{\theta} - W_i^{\mathrm{T}}(t_l)u_i(t_l)] \end{aligned} \tag{A.3}$$

Recall (4), we have $u_i = W_i\theta$. Substituting this fact back into (A.3), one has

$$\dot{V}_a = -k_v\omega_e^2 - k_w\tilde{\theta}^{\mathrm{T}}\sum_{l=1}^{L}[W_i^{\mathrm{T}}(t_l)W_i(t_l)]\tilde{\theta} \tag{A.4}$$

Since $\sum_{l=1}^{L}[W_i^{\mathrm{T}}(t_l)W_i(t_l)] \geq 0$, one has $\gamma_e, \omega_e, \tilde{\psi} \in \mathcal{L}_\infty$. Then one can ensure that $\gamma_e, \omega_e \to 0$ by Barbalat lemma. Moreover, the boundedness of $\hat{\psi}$ indicates that $\hat{\theta}_q \in (\theta_{q,\min}, \theta_{q,\max})$.

Next we will show that the exponential convergence can be ensured when $\sum_{l=1}^{L}[W_i^{\mathrm{T}}(t_l)W_i(t_l)]$ is full-rank. We denote its minimal eigenvalue as λ_w and consider a storage function:

$$V_b = \alpha V_a + \gamma_e\omega_e \tag{A.5}$$

where $\alpha > 0$ is employed for analysis purposes. One can readily verify that V_b is a valid Lyapunov function candidate under the condition $\alpha > 1/\sqrt{k_pM_i}$. Then we have

$$\begin{aligned} \dot{V}_b =& -k_p\gamma_e^2 - (\alpha k_v - 1)\omega_e^2 - k_v\gamma_e\omega_e + \gamma_e Y_i\tilde{\theta} \\ &- \alpha k_w\tilde{\theta}^{\mathrm{T}}\sum_{l=1}^{L}[W_i^{\mathrm{T}}(t_l)W_i(t_l)]\tilde{\theta} \end{aligned} \tag{A.6}$$

By employing the inequality of arithmetic and geometric means, one has

$$\dot{V}_b \leq -\frac{k_p}{2}\gamma_e^2 - (\alpha k_v - 1 - \frac{k_v^2}{k_p})\omega_e^2 - (\alpha k_w\lambda_w - \frac{\|Y_i\|^2}{k_p})\|\tilde{\theta}\|^2 \tag{A.7}$$

Since the reference signals and tracking errors are bounded, we can ensure that Y_i is also bounded and that there exists a positive constant c_y such that $\|Y_i\|^2 \leq c_y$. Therefore, by setting $\alpha > 2 \cdot \max\{1/k_v + k_v/k_p, c_y/(k_pk_w\lambda_w), 1/\sqrt{k_pM_i}\}$, one has

$$\dot{V}_b \leq -\frac{k_p}{2}\gamma_e^2 - \frac{\alpha k_v}{2}\omega_e^2 - \frac{\alpha k_w\lambda_w}{2}\|\tilde{\theta}\|^2 \tag{A.8}$$

Besides, we state that $\sum_{q=1}^{h} H(\tilde{\psi}_q) \leq c_\psi\tilde{\theta}^2$ when $\tilde{\psi} \in \mathcal{L}_\infty$, where $c_\psi = \max_{q=1,2\ldots,h;t\geq 0}\{\frac{(1+\mathrm{e}^{-\tilde{\psi}_q(t)-\psi_q})^2}{(\theta_{q,\max}-\theta_{q,\min})\mathrm{e}^{-\tilde{\psi}_q(t)-\psi_q}}\}$. This can be verified by analysing the gradient of $\sum_{q=1}^{h} H(\tilde{\psi}_q) - c_\psi\tilde{\theta}^2$ with respect to $\tilde{\psi}$. Based on this fact, one has

$$V_b \geq \frac{\alpha k_v}{4}\gamma_e^2 + \frac{\alpha M_i}{4}\omega_e^2 + \alpha c_\psi\|\tilde{\theta}\|^2 \tag{A.9}$$

Eqs. (A.8) and (A.9) lead to $\dot{V}_b \leq -\beta V_b$, where $\beta = \min\{2k_p/(\alpha k_v), 2k_v/M_i, k_w\lambda_w/(2c_\psi)\}$. This directly guarantees the exponential convergence of γ_e, ω_e and $\tilde{\psi}$, and it completes the whole proof.

References

[1] Vermeer L, Sørensen JN, Crespo A. Wind turbine wake aerodynamics. Prog Aerosp Sci 2003;39(6–7):467–510.

[2] Adaramola M, Krogstad P-Å. Experimental investigation of wake effects on wind turbine performance. Renew Energy 2011;36(8):2078–86.

[3] Marathe N, Swift A, Hirth B, Walker R, Schroeder J. Characterizing power performance and wake of a wind turbine under yaw and blade pitch. Wind Energy 2016;19(5):963–78.

[4] Hou P, Enevoldsen P, Hu W, Chen C, Chen Z. Offshore wind farm repowering optimization. Appl Energy 2017;208:834–44.

[5] Famoso F, Brusca S, D'Urso D, Galvagno A, Chiacchio F. A novel hybrid model for the estimation of energy conversion in a wind farm combining wake effects and stochastic dependability. Appl Energy 2020;280:115967.

[6] Fleming PA, Ning A, Gebraad PM, Dykes K. Wind plant system engineering through optimization of layout and yaw control. Wind Energy 2016;19(2):329–44.

[7] Vali M, Petrović V, Boersma S, van Wingerden J-W, Pao LY, Kühn M. Adjoint-based model predictive control for optimal energy extraction in waked wind farms. Control Eng Pract 2019;84:48–62.

[8] Yin X, Zhao X. Deep neural learning based distributed predictive control for offshore wind farm using high fidelity LES data. IEEE Trans Ind Electron 2020, Early Access.

[9] Gebraad PMO, Teeuwisse F, Van Wingerden J, Fleming PA, Ruben S, Marden J, Pao L. Wind plant power optimization through yaw control using a parametric model for wake effects - a CFD simulation study. Wind Energy 2016;19(1):95–114.

[10] Dar Z, Kar K, Sahni O, Chow JH. Windfarm power optimization using yaw angle control. IEEE Trans Sustain Energy 2016;8(1):104–16.

[11] Hulsman P, Andersen SJ, Göçmen T. Optimizing wind farm control through wake steering using surrogate models based on high-fidelity simulations. Wind Energy Sci 2020;5(1):309–29.

[12] Jensen NO. A note on wind generator interaction. 1983.

[13] Park J, Law KH. Bayesian ascent: A data-driven optimization scheme for real-time control with application to wind farm power maximization. IEEE Trans Control Syst Technol 2016;24(5):1655–68.

[14] Park J, Law KH. A data-driven, cooperative wind farm control to maximize the total power production. Appl Energy 2016;165:151–65.

[15] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press; 2018.

[16] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33.

[17] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Dueling network architectures for deep reinforcement learning. In: International Conference on Machine Learning. 2016.

[18] Luo B, Yang Y, Liu D. Adaptive Q-learning for data-based optimal output regulation with experience replay. IEEE Trans Cybern 2018;48(12):3337–48.

[19] Luo B, Yang Y, Liu D. Policy iteration Q-learning for data-based two-player zero-sum game of linear discrete-time systems. IEEE Trans Cybern 2020, Early Access.

[20] Zhao H, Zhao J, Qiu J, Liang G, Dong ZY. Cooperative wind farm control with deep reinforcement learning and knowledge assisted learning. IEEE Trans Ind Inf 2020;16(11):6912–21.

[21] Dong H, Zhao X, Yang H. Reinforcement learning-based approximate optimal control for attitude reorientation under state constraints. IEEE Trans Control Syst Technol 2020.

[22] Fleming P, Gebraad P, Churchfield M, Lee S, Johnson K, Michalakes J, van Wingerden J-W, Moriarty P. SOWFA + super controller user's manual. Tech. rep., National Renewable Energy Lab (NREL), Golden, CO, US; 2013.

[23] Burton T, Jenkins N, Sharpe D, Bossanyi E. Wind Energy Handbook. John Wiley & Sons; 2011.

[24] Marden JR, Ruben SD, Pao LY. A model-free approach to wind farm control using game theoretic methods. IEEE Trans Control Syst Technol 2013;21(4):1207–14.

[25] Jonkman JM, Buhl Jr ML. FAST user's guide. National Renewable Energy Laboratory (NREL); 2005.

[26] Luo B, Liu D, Huang T, Wang D. Model-free optimal tracking control via critic-only Q-learning. IEEE Trans Neural Netw Learn Syst 2016;27(10):2134–44.

[27] Chowdhary G, Mühlegg M, Johnson E. Exponential parameter and tracking error convergence guarantees for adaptive controllers without persistency of excitation. Internat J Control 2014;87(8):1583–603.

[28] Kamalapurkar R, Reish B, Chowdhary G, Dixon WE. Concurrent learning for parameter estimation using dynamic state-derivative estimators. IEEE Trans Automat Control 2017;62(7):3594–601.

[29] Pan Y, Yu H. Composite learning from adaptive dynamic surface control. IEEE Trans Automat Control 2015;61(9):2603–9.

[30] Pan Y, Yu H. Composite learning robot control with guaranteed parameter convergence. Automatica 2018;89:398–406.

[31] Churchfield M, Lee S, Moriarty P, Martinez L, Leonardi S, Vijayakumar G, Brasseur J. A large-eddy simulation of wind-plant aerodynamics. In: 50th AIAA Aerospace Sciences Meeting. 2012, p. 537.

[32] Churchfield MJ, Lee S, Michalakes J, Moriarty PJ. A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics. J Turbulence 2012;(13):N14.

[33] Sanderse B. Aerodynamics of Wind Turbine Wakes. Petten: ECN; 2009.

[34] Ghaisas NS, Archer CL, Xie S, Wu S, Maguire E. Evaluation of layout and atmospheric stability effects in wind farms using large-eddy simulation. Wind Energy 2017;20(7):1227–40.

[35] Doubrawa P, Churchfield MJ, Godvik M, Sirnivas S. Load response of a floating wind turbine to turbulent atmospheric flow. Appl Energy 2019;242:1588–99.

[36] Fleming P, Gebraad PM, Lee S, van Wingerden J-W, Johnson K, Churchfield M, Michalakes J, Spalart P, Moriarty P. Simulation comparison of wake mitigation control strategies for a two-turbine case. Wind Energy 2015;18(12):2135–43.

[37] Boersma S, Doekemeijer B, Vali M, Meyers J, van Wingerden J-W. A control-oriented dynamic wind farm model: WFSim. Wind Energy Sci 2018;3(1):75–95.

[38] Ainslie JF. Calculating the flowfield in the wake of wind turbines. J Wind Eng Ind Aerodyn 1988;27(1–3):213–24.

[39] Jonkman JM, Annoni J, Hayman G, Jonkman B, Purkayastha A. Development of FAST.Farm: A new multi-physics engineering tool for wind-farm design and analysis. In: 35th Wind Energy Symposium. 2017.