Multi-agent reinforcement learning control of a hydrostatic wind turbine-based farm

Yubo Huang, Shuyue Lin, and Xiaowei Zhao

Abstract—This paper leverages multi-agent reinforcement learning (MARL) to develop an efficient control system for a wind farm comprising a new type of wind turbines with hydrostatic transmission. The primary motivation for hydrostatic wind turbines (HWT) is increased reliability, and reduced manufacturing, operating, and maintaining costs by removing troublesome components and reducing nacelle weight. Nevertheless, the high system complexity of HWT and the wake effect pose significant challenges for the control of HWT-based wind farms. We therefore propose a MARL algorithm named multi-agent policy optimization (MAPO), which allows agents (turbines) to gradually improve their control policies by repeatedly interacting with the environment to learn an optimal operation curve for wind farms. Simulation results based on a wind farm simulator, FAST.Farm, show that MAPO outperforms the greedy policy and a popular learning-based method, multi-agent deep deterministic policy gradient (MADDPG), in terms of power generation.

Index Terms—Wind farm control, hydrostatic wind turbines, multi-agent reinforcement learning, power generation.

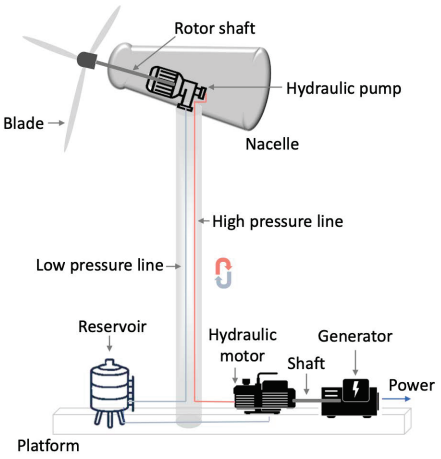

Fig. 1. The substructures of a hydrostatic wind turbine.

I. Introduction

Developing renewable energy to substitute traditional fossil energy is one of the most promising ways to reduce environmental pollution. In Europe, wind energy accounts for the highest share of clean energy generation and is also the fastest-growing electricity source in the market [1]. Nonetheless, there is an intractable drawback for offshore wind farms comprising of gearbox-based wind turbines—their maintenance is costly. Hydrostatic wind turbines (HWT) can help tackle this problem [2] because the hydrostatic transmission system is more robust than the gearbox-based transmission and can offer a longer life cycle. In addition, HWT allows to shift the heavy motor and generator to the platform (Fig. 1), and therefore the mass of the nacelle can be significantly reduced, which vastly facilitates ease of installation and maintenance of wind turbines. Furthermore, the frequency/inertial response exhibited by HWTs is of clear value to large-scale power systems because they are installed with synchronous generators. These economical advantages motivate us to study the HWT-based wind farm. We focus on its control in this paper.

Like the case for the traditional wind turbines/farms, the control method for a single HWT is not suitable for a HWT-based wind farm due to the wake effect. Specifically, the optimal control policy for an isolated HWT is maximum power point tracking (MPPT [3], see Fig. 2): when the wind speed is below rated, the objective is to control the generator torque to maximize its power output. When the wind speed is sufficient to drive the full-power operation of HWTs, the goal becomes to maintain the output at the rated level to alleviate the structural load via the joint control of blade pitch and generator torque. In wind farms, turbines are normally installed in arrays, and thus the actions of upstream turbines affect the environmental state of their downstream counterparts through the wake effect. Although MPPT can achieve optimal solutions for upstream turbines, the power outputs of HWTs within the wake planes of upstream turbines are reduced greatly, causing a decline in power generation of the entire wind farm. Therefore, how to design a control policy for wind farms which can overcome the wake effect is an ongoing issue.

For the farm-level control, the model-free methods may offer more benefits than the model-based methods due to the high system complexity and environmental uncertainty of wind farms. Firstly, model-based control methods (e.g. Model Predictive Control) require an accurate wind farm model, but the high environmental uncertainty of wind farms will inevitably introduce considerable modelling errors. Control policies designed based on the model with modelling errors are likely to be

Y. Huang and X. Zhao (corresponding author) are with the Intelligent Control & Smart Energy (ICSE) Research Group, School of Engineering, University of Warwick, Coventry, CV4 7AL, UK. Emails: Yubo.Huang@warwick.ac.uk, Xiaowei.Zhao@warwick.ac.uk.

S. Lin is with the Department of Engineering, University of Hull, Cottingham Road, Hull, HU6 7RX, UK. Email: S.Lin@hull.ac.uk.

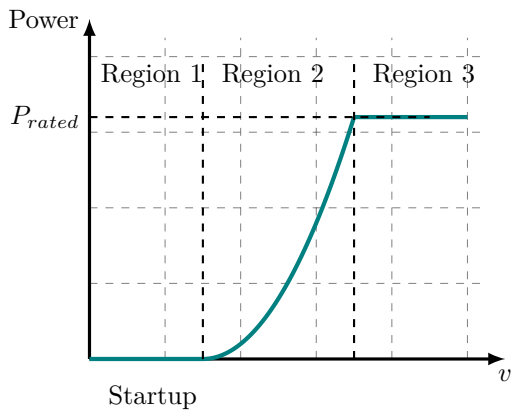

Fig. 2. The optimal operation curve of an individual wind turbine—MPPT [4].

sub-optimal. Additionally, the algorithm complexity of model-based methods is usually higher than the model-free methods, which can cause greater computational cost. For example, when the task has a long horizon like the wind farm control case, it might be difficult for model predictive control to achieve real-time control because of the expensive computation cost. Thus many studies have recently attempted to leverage model-free data-driven methods to approach a better wind farm control policy, including dynamic programming [5], genetic algorithm [6], and swarm optimization [7].

Among multitudinous model-free methods, model-free reinforcement learning (RL) has its exclusive advantages in solving the wind farm control task. For example, dynamic programming is impractical for large-scale wind farm control since it has high memory expenditure when the state space is large. As for the genetic algorithm and swarm optimization, they cannot guarantee the convergence or stability of the control policy during the optimization process. Model-free RL [8] can effectively tackle these challenges with the assistance of deep neural networks and has achieved excellent results in wind farm control. Dong et al. integrated deep deterministic policy gradient (DDPG) and the high-fidelity wind farm model to learn the control policy [9]. Zhao et al. used the knowledge-assisted DDPG to optimize the control policy as well as ensuring safety during training [10]. Bay et al. introduced a distributed RL-based method to wind farm power capture maximization using yaw control [11]. These works demonstrated that model-free RL can be applied smoothly to wind farm control and achieve better results than many selected data-driven methods.

Almost all existing model-free RL control methods for wind farms (which consist of multiple turbines) regard the wind farm as a single agent, but using multi-agent RL (MARL) to train wind farm control policy is obviously more rational than using single-agent RL (SARL). There are some limitations encountered in applications of SARL:

- SARL is not scalable since the dimensions of the joint action space will grow exponentially with the increase in the number of HWTs in a wind farm.
- In execution, each HWT demands to acquire the states of their teammates to generate its action based on the control policy. This high degree of communication can not be satisfied in the real-world scenarios.

Both limitations can be addressed by introducing the centralized training with decentralized execution (CTDE) principle [12] in MARL. This implies that the concatenation of the states of all HWTs is inputted to the value network to estimate the future return (power) of each HWT during training, but each HWT only uses their private state to sample its action (low dimension) rather than the joint action based on the individual policy in execution (communication-free).

On the other hand, there are also several challenges in designing the control system of a HWT-based wind farm within the MARL framework. Firstly, to bridge the simulation to reality gap, in the construction of the wind farm simulator, we should not only consider the aerodynamics of the wind farm but also the dynamics of multifarious substructures of HWTs, which are typically ignored in the existing wind farm control research. Moreover, there are significant differences in the RL-based control designs between wind farms consisting of gearbox-based wind turbines and the ones consisting of HTWs. For example, to standardize the control task as a complete MDP (Markov decision process, a compulsory condition for RL design), the former only includes the rotor speed in the state space because gearbox-based wind turbines have constant gearbox ratios between the rotors and generators. However, the latter must consider the dynamics of the hydrostatic transmission of each HWT besides the rotor speed. Last but not least, the developed MARL algorithm need effectively enhance the coordination between HWTs to overcome the wake effect. This paper makes the following contributions to address the aforementioned issues:

- Developed a HWT-based wind farm model based on FAST.FARM [13], where the gearbox transmission of the wind turbine is replaced by the hydrostatic transmission. This model includes both the aerodynamics of large-scale wind farms and the mechanical dynamics of substructures of a HWT. Then, the FAST.Farm driven by the proposed model is integrated with Python to build a high-fidelity HWT-based wind farm simulator used for training MARL algorithms.
- Proposed a novel CTDE-based MARL algorithm named multi-agent policy optimization (MAPO) to learn the wind farm control policy. MAPO balances the collective return and the individual return by a dynamical weight, which induces agents to explore new policies in the initial training and exploit the explored information to subsequently maximize the group return. By encouraging agents to maximize the collective return, MAPO can efficaciously promote the coordination between HWTs and further minimize

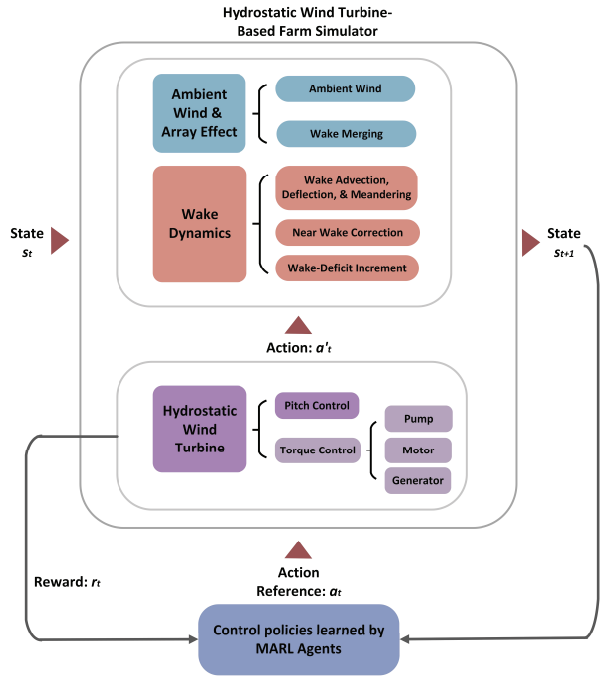

Fig. 3. Sub-model hierarchy of the HWT-based farm simulator for MARL. Note that we only illustrate one turbine in this figure for convenience. In fact, this simulator can include multiple turbines during operation.

the negative effect of wakes on the power generation.

- Simulation results show that the control policy trained by MAPO achieves high performance in different wind farm power layout and fluctuating environments. The structural dynamic analysis shows that MAPO does not cause unusual vibrations of the main sub-structures.

II. Constructing a HWT-based wind farm simulator for MARL

Before we train control policies for HWTs by using the MARL algorithms, a high-fidelity simulator should be developed. This simulator includes the models of the aerodynamics of the wind farm and the elastic-servo dynamics of HWT. Different from the traditional control methods that use the HWT-based wind farm model to design the control policy, MARL aims to teach each agent (turbine) to learn the control policy through interacting with the simulator. Please refer subsection III-A for details. Below, we will introduce the used hydraulic wind turbine models and its control modules.

A. Modeling the dynamics of hydrostatic wind turbines

At the farm level, the aerodynamic torque T_r^i[1] of the rotor and the thrust force F_{thrust}^i[2] exerted by the turbine i can be described through a quasi-static model [14]:

$$\begin{aligned} T_r^i &= \frac{1}{2}\rho\pi R^3 {v^i}^2 C_p(\lambda^i,\beta^i) \\ F_{thrust}^i &= \frac{1}{2}\rho\pi R^2 {v^i}^2 C_T(\lambda^i,\beta^i) \end{aligned} \tag{1}$$

where $i = 1, 2, \cdots, n$ and n is the number of HWTs in a farm; $\rho, R, \omega_r^i, \beta^i$ are the air density, blade length, rotor speed, and pitch angle of turbine i, respectively; v^i is the wind speed at the i-th turbine. $\lambda^i = \omega_r^i R/v^i$ is the tip speed ratio; C_p and C_T are the the power coefficient and the disk-based thrust coefficient [15], respectively.

FAST.Farm uses a gearbox-based turbine model to simulate the operation of a wind farm. The main task in this subsection is to embed the HWT model into the farm-level aerodynamics model introduced in subsection II-A to construct a complete HWT-based wind farm simulator.

For the i-th HWT, the dynamics of its rotor speed is proportional to the difference between T_r^i obtained from Eq. 1 and T_p^i (the torque of pump):

$$\dot{\omega}_r^i = \frac{1}{J_r^i + J_p^i}(T_r^i - T_p^i) \tag{2}$$

where J_r^i and J_p^i are the rotational mass moments of inertia of the rotor and pump, respectively.

A hydrostatic drivetrain transmits the mechanical power on the low-speed rotor side to the high-speed generator side for electricity generation. As shown in Fig. 1, this hydrostatic drivetrain comprises a hydraulic pump, high-pressure and low-pressure lines, and a hydraulic motor. First, the rotation of the low-speed shaft with the rotor-pump assembly can pump the hydraulic oil from the low-pressure transmission line to the high-pressure line and the pump torque is [16]:

$$T_p^i = D_p P_p^i + B_p \omega_r^i + C_{fp} D_p P_p^i \tag{3}$$

where D_p is the pump displacement, meaning the volume of fluid pumped per revolution, P_p^i represents the pressure difference across the pump, B_p is the viscous damping, and C_{fp} is the Coulomb friction coefficient of the pump. The net volumetric flow of the pump Q_p^i is computed by:

$$Q_p^i = D_p \omega_r^i - C_{sp} P_p^i \tag{4}$$

where C_{sp} is the laminar leakage coefficient of the pump.

Then, we use a dissipative model to interpret the dynamics of transmission lines [17]. Specifically, this model describes how changes in the net volumetric flows of the pump Q_p^i and motor Q_m^i cause the state transform of hydraulic lines (Eq. 5), and further result in the variation of pressure difference in pump and motor (Eq. 6), where P_m^i denotes the pressure difference across the motor.

$$\dot{\boldsymbol{x}}^i = \boldsymbol{A}\boldsymbol{x}^i + [\boldsymbol{B_1}, \boldsymbol{B_2}] \begin{bmatrix} Q_p^i \\ Q_m^i \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} P_p^i \\ P_m^i \end{bmatrix} = \begin{bmatrix} \boldsymbol{C_1} \\ \boldsymbol{C_2} \end{bmatrix} \boldsymbol{x^i} \tag{6}$$

[1]All variables in this paper are in the International System of Units.

[2]In this paper, the superscript i denotes the i-th turbine (HWT).

The presented model uses the form of state space to represent the dynamics of fluid in a hydrostatic drivetrain. Here, $\boldsymbol{A}$, $\boldsymbol{B} = [\boldsymbol{B_1}, \boldsymbol{B_2}]$, and $\boldsymbol{C} = [\boldsymbol{C_1}; \boldsymbol{C_2}]$ are the state matrix, input matrix, and output matrix, respectively, and their values are determined by the length L and inner diameter r of transmission lines, and the density ρ, kinematic viscosity ν, and effective bulk modulus E of the hydraulic oil (please see [18] for specific calculations). $\boldsymbol{x^i}$ is the state vector, $\boldsymbol{Q^i} = [Q_p^i, Q_m^i]^T$ is the input vector and $\boldsymbol{P^i} = [P_p^i, P_m^i]^T$ is the output vector.

Similar to the pump, the motor can also be characterized by its volumetric displacement D_m^i, but the function of the motor is to convert hydraulic power into mechanical power. Thus, for the hydraulic motor, we only reverse the sign of the leakage flow and friction torques in the pump model [16]. The net volumetric flow Q_m^i and torque T_m^i of the pump are:

$$\begin{aligned} Q_m^i &= D_m^i \omega_m^i + C_{sm} P_m^i \\ T_m^i &= D_m^i P_m^i - B_m \omega_m + C_{fm} D_m P_m^i \end{aligned} \tag{7}$$

where ω_m^i is the motor speed, C_{sm} is the laminar leakage coefficient of the motor, B_m is the viscous damping, and C_{fm} is the Coulomb friction coefficient of the motor.

In a hydrostatic transmission system, we can control the motor torque by changing its displacement D_m (Eq. 7). The response of motor displacement is characterized via a time constant $t_m = 0.5$ and a displacement reference $\hat{D}_m^i$:

$$\dot{D}_m^i = \frac{1}{t_m}(\hat{D}_m^i - D_m^i) \tag{8}$$

And the power produced by the generator is:

$$P_g^i = \eta T_m^i \omega_m^i \tag{9}$$

where η is the generator efficiency.

At this point, we have integrated the aerodynamic model of the wind farm and the hydrostatic transmission model of the turbine. Next we will implement them in FAST.Farm. We replace the gearbox-based drivetrain with the hydrostatic drivetrain by modifying the ServoDyn module in FAST.Farm[3]. Firstly, the drivetrain rotational-flexibility DOF is closed in the ElastoDyn input file (.dat) and the GBRatio is set to 1. Then, we regard the generator in gearbox-based wind turbines as the hydraulic pump in HWTs and modify its inertial in the FAST input file (.fst). The transmission dynamics (Eqs. 5-6) of the hydraulic system in HWTs is modeled as a function in the ServoDyn module and it will be called before the state update of the servo system. Finally, in the UserVSCont_KP.f90 file, we provide an interface to write the trained MARL control policy and the MARL training samples can be collected in the .out files. Now the HWT-based wind farm simulator is constructed and can perform its core function shown in Fig. 3.

[3]the source code of FAST.Farm: https://github.com/OpenFAST/openfast

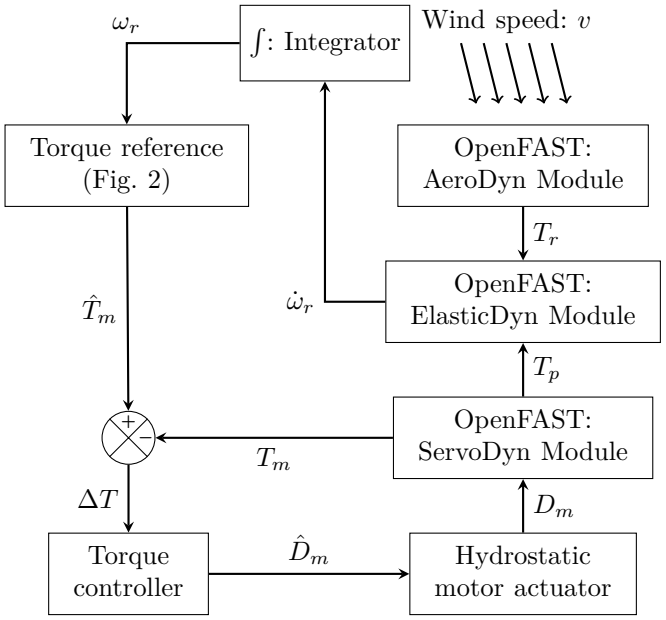

Fig. 4. The torque control system of HWTs. In the simulator, the AeroDyn module can compute the load of HWT according to the inflow wind. The ElasticDyn module determines the kinematics of each substructure of a wind turbine. The ServoDyn module describes the dynamics of the servo system, and the control system is also embedded in this module. $\hat{D}_m$ is the displacement command of the hydraulic motor.

B. The control framework of an individual HWT

Above we have constructed a simulator of the HWT-based wind farm. Then we will introduce the torque control and blade pitch control regimes of HWTs in the simulator.

1) Torque control: For a single variable-speed HWT, its operation curve (MPPT: maximal power point tracking, also called the greedy policy) can be divided into three regions shown in Fig. 2. In region 2, below the rated wind speed, the wind is not sufficient to drive the turbine to operate at its full-power point. The blade pitch angle will keep at its minimum to capture wind energy as much as possible. The primary task in region 2 is to control the motor torque to make the HWT run on its optimal torque curve (Fig. 2), maximizing the output power. Considering the motor displacement actuator, the closed-loop torque control system is shown in Fig. 4. It is worth mentioning that we find the respond of motor displacement control is obviously swift than that of the pump in pre-experiments since the pump affects the generator torque by changing the pressure and flow rate of the hydraulic oil but the motor can directly determine the input mechanical torque of generator.

2) Blade pitch control: According to MPPT, in region 3 (see Fig. 2), the output power of a HWT should be kept at its nominal value via the blade pitch control [2]. The dynamics of pitch actuator can be represented by a first-order differential equation:

$$\dot{\beta} = \frac{1}{t_\beta}(\hat{\beta} - \beta) \tag{10}$$

where β and $\hat{\beta}$ are the real-time pitch angle and its reference determined by MPPT and the pitch controller, respectively, and $t_\beta = 0.1$ is the time constant of the blade pitch actuator.

From the above introduction, for a single wind turbine, the torque and pitch references are calculated by MPPT during its operation. This coordination-free control policy is optimal for an isolated turbine but is unsuitable for a wind farm due to the wake effect. For instance, if all upstream wind turbines adopted this greedy control strategy[4], although they could maximize their power output, within their wake plane, the downstream wind speed would experience a rapid drop and the power generation of turbines situated at this area will plummet. As a result, the power production of the entire wind farm would keep at a relatively low level. To tackle this problem, in the next section, a novel MARL method will be proposed to train a collaborative control policy for all the HWTs in a wind farm to overcome the wake effect. Then, the real-time references of torque and pitch angle of HWTS will be generated by the trained policy.

III. Multi-agent reinforcement learning control of a HWT-based wind farm

In Section II-B, we have introduced the greedy control policy (MPPT) that uses the optimal operation curve to calculate the control references of a single HWT. For wind farm control, however, there is no one-size-fits-all optimal operation curve, but the policy network in RL can approximate it through interacting with the simulator. In this section, we propose the Multi-Agent Policy Optimization (MAPO) algorithm to control the wind farm. And we also illustrate how MAPO trains a collaborative control policy for a HWT-based wind farm by using the simulator introduced in Section II, and how the control policy guides the actions of HWTs to alleviate the wake effect and further boost the power generation of the whole wind farm.

A. Modeling the HWT-based wind farm control task as a Markov decision process

In MAPO, we regard each HWT in the wind farm as an agent which has an independent policy network/function π^i and agent value network/function $V^i, \forall i \in [1, 2, \cdots, n]$. Overall, there is a group value network/function V^{gru} used for estimating the future return of the wind farm based on its state s_t. The policy network π^i outputs the action a_t^i (control reference signals) for turbine i given its observation o_t^i and the agent value network V^i estimates the future return of turbine i (Eq. 12). The concrete simulator state, agent action, and reward are defined as follows:

- State: the observation o^i of turbine i includes not only its external information (e.g. the wind speed on the rotor, the turbine location) but also its internal status— the rotor speed ω_r^i, and the pump and motor pressure differences (P_p^i and P_m^i). The group (farm) state s is the concatenation of observations of all agents (Eq. 11).
- Action: the action a^i is the control reference signals (torque reference and pitch reference) that the corresponding substructure of wind turbine i should track to maximize the output power.
- Reward: the reward r^i should be proportional to the power generated by turbine i. Hence the reward function is designed as Fig 5. We expect all turbines can work in their rated state, so the reward of turbine i is maximal at its rated point. When the power exceeds its rated value, the reward is set to 0 to punish the agent. The group reward r is the sum of all agent rewards (Eq. 11).

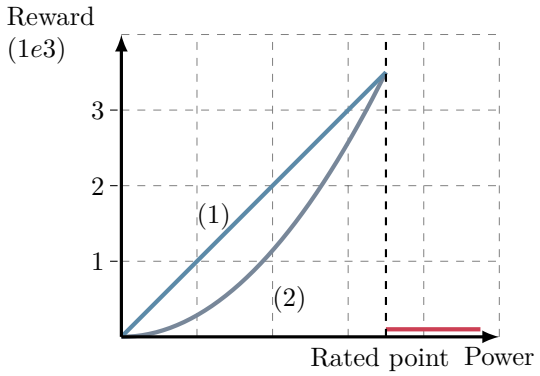

Fig. 5. The reward functions in the wind farm control task.

And they satisfy that:

$$\begin{aligned} s_t &= o_t^1 \oplus o_t^2 \oplus \cdots \oplus o_t^n \\ a_t &= a_t^1 \oplus a_t^2 \oplus \cdots \oplus a_t^n \\ r_t &= r_t^1 + r_t^2 + \cdots + r_t^n \\ s_{t+1} &= o_{t+1}^1 \oplus o_{t+1}^2 \oplus \cdots \oplus o_{t+1}^n \end{aligned} \tag{11}$$

where $\oplus$ is the operator of concat and n is the number of turbines in the simulator.

Based on these concepts, the agent state value function $V_{\pi^i}(o_t^i)$ under policy π^i and the group state value function $V_\pi(s_t)$ under policy π can be defined as (Hereinafter, $V_{\pi^i}(o_t^i)$ and $V_\pi(s_t)$ are abbreviated as V_t^i and V_t, respectively):

$$\begin{aligned} V_{\pi^i}(o_t) &= \mathbb{E}\left\{\sum_{l=0}^{\infty} \gamma^l r^i(o_{t+l}^i)\right\} = r^i(o_t^i) + \gamma V_{\pi^i}(o_{t+1}) \\ V_\pi(s_t) &= \mathbb{E}\left\{\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})\right\} = r(s_t) + \gamma V_\pi(s_{t+1}) \end{aligned} \tag{12}$$

where γ is the discount coefficient.

The interaction between the RL agents and the HWT-based wind farm simulator can be standardized as a Partially Observable MDP. Initially, the weights of all policy networks are randomly initialized and thus the corresponding farm control policy is of low quality. At each

[4]In this paper, MPPT is also known as the greedy control strategy.

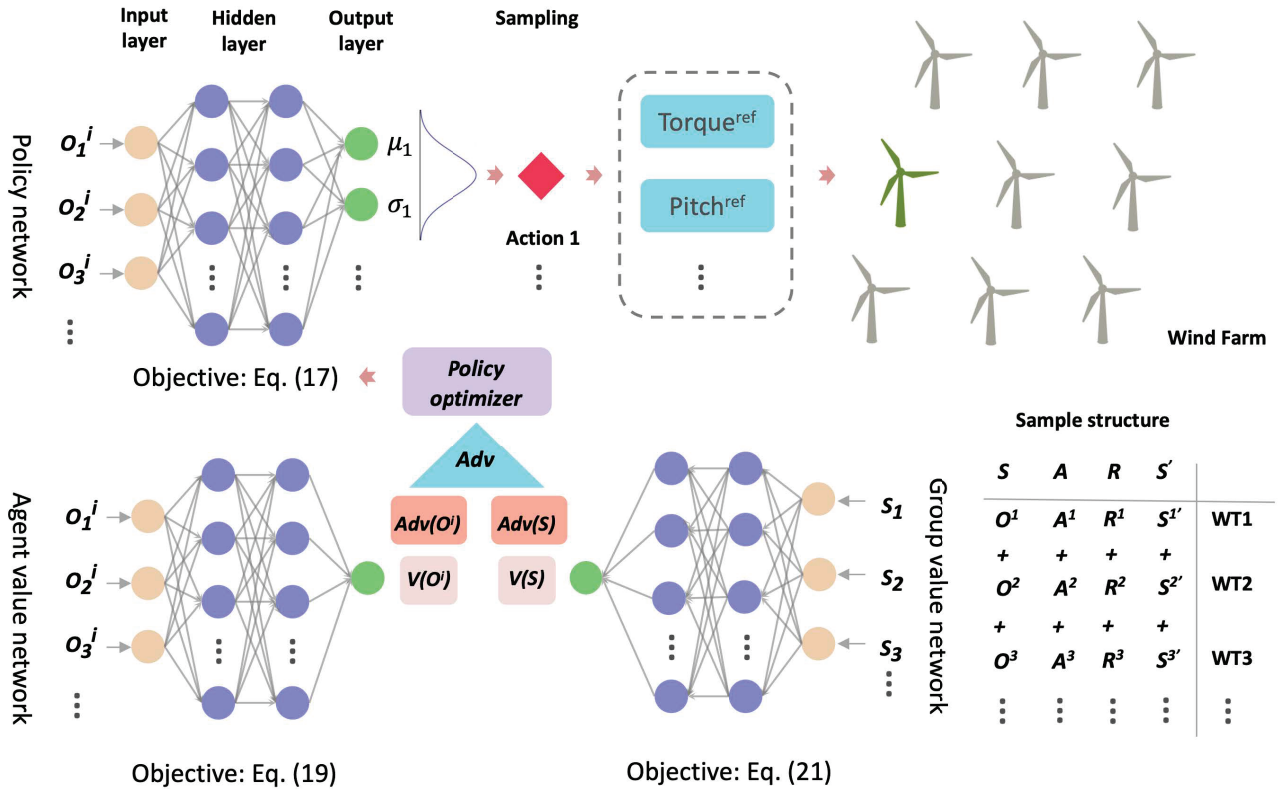

Fig. 6. The MAPO-based control system for wind farms

discrete time t, as shown in Fig. 6, the agent i (turbine) observes its private status $o_t^i \in \mathcal{O}^i$ from the simulator. The concatenation of observations of all agents is the group state $s_t \in \mathcal{S}$ (Eq. 11). Based on the observation o_t^i, the policy network π^i of agent i will sample an action a_t^i (control reference signal) for different turbine substructures ($\mathcal{O}^i \rightarrow \mathcal{A}^i$). Then all turbines will take their actions (e.g. torque reference), and the simulator will feed back a reward $r^i \in \mathcal{R}$ to each agent while jumping to the next state s_{t+1} (refer to Fig 3). The sample (s_t, a_t, r_t, s_{t+1}) will be collected to train the policy and value network (see the next subsection for details) to improve the performance of the control policy, and then this interaction will continue. At each iteration, the quality of the policy $\pi^i, \forall i \in [1, 2, \cdots n]$ can be evaluated by the expected return (power generated by the wind farm):

$$L(\pi^i) = \mathbb{E}_{(o_t^i, a_t^i)\sim\rho(s_0),\pi^i,\mathcal{P}}\left\{\sum_{t=0}^{\infty}\gamma^t r^i(o_t^i, a_t^i)\right\} \tag{13}$$

where s_0 is the start state of the simulator and ρ is its probability distribution.

After this process is iterated enough times, the original random control policy will converge to a superior solution that can be deployed to real-world machines. Additionally, as illustrated in Fig. 6, we input the private observation o^i and the group state s to the value network $V^i(o^i)$ and $V(s)$ to estimate the future return of agent i and the group future return, respectively. However, in the policy network π_i, only the private observation o^i is leveraged to sample the action references. This setting is to satisfy the principle of CTDE, which can avoid the communication and environment non-stationary issues in MARL.

In the HWT-based wind farm control task, if all turbines aim to maximize their own return, the ultimate control policy will probably fall into a locally optimal solution. Otherwise, if the objective of all agents is always to maximize the group return throughout the training, in the initial stage, agents tend to exploit the explored information to increase the collective return rather than discovering new states. It will limit the exploration of each agent and thus the learning speed is extremely slow at this stage. We expect the agent to focus on increasing their own return at the beginning of the training but dedicate to accumulating the group return in the latter stage to find the best collaborative control policy. We can leverage a dynamical parameter η, whose value gradually grows from 0 at the beginning to 1 after the training, to Eq. 13 to achieve this purpose. Now the objective of the policy network π^i changes from Eq. 13 to:

$$\begin{aligned} L(\pi^i) = (1-\eta)\mathbb{E}_{(o_t^i, a_t^i)\sim\rho(s_0),\pi^i,p}\left\{\sum_{t=0}^{\infty}\gamma^t r^i(o_t^i, a_t^i)\right\} \\ + \eta\mathbb{E}_{(s_t, a_t)\sim\rho(s_0),\pi^i,p}\left\{\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right\} \end{aligned} \tag{14}$$

Whereupon, for the i-th agent, under the policy π^i, the advantage of action a^i over other actions is:

$$\begin{aligned} Adv_{\pi^i}(o_t^i, a_t^i) = (1-\eta)[r^i(o_t^i, a_t^i) + \gamma V(o_{t+1}^i) - V(o_t^i)] \\ + \eta[(r(s_t, a_t) + \gamma V(s_{t+1})) - V(s_t)] \end{aligned} \tag{15}$$

To enhance the stability and facilitate the performance of RL algorithms, in this paper, we use the general advantage estimator (GAE) [19], [20] to calculate the advantage:

$$AG^i(a_t^i, o_t^i) = \sum_{k=0}^{\infty} (\gamma\lambda)^k Adv_{\pi^i}(o_{t+k}^i, a_{t+k}^i) \tag{16}$$

where λ is a constant less than 1.

B. Training the multi-agent RL functional networks

In this subsection, we present the training method of functional networks in MAPO. During the interaction between agents and the simulator, the operation trajectory $\mathcal{D}$ of the wind farm (include the trajectory D^i of turbine i, $\forall\ i = 1, 2, \cdots, n$) can be collected for training. The sample structures of these trajectories are $(s_t, a_t, r_t, s_{t+1}) \in \mathcal{D}$ used to train the group value network and $(o_t^i, a_t^i, r_t^i, o_{t+1}^i) \in \mathcal{D}^i$ used to train the policy and value network of agent i.

At the k-th iteration, the weight matrix of agent i's policy network π_k^i is θ_k^i. The objective of π_k^i is to maximize Eq. 14. However, in practical, it is impossible that using Eq. 14 to optimize π_k^i directly. Instead, [21] proposed a surrogate objective to update it based on the collected samples D_k^i:

$$\begin{aligned} \theta_{k+1}^i = \arg\max_{\theta^i} \frac{1}{|\mathcal{D}_k^i T|} \sum_{\tau^i \in \mathcal{D}_k^i} \sum_{t=0}^{T} \\ \min\left(\frac{\pi^i(a_t^i, o_t^i)}{\pi_k^i(a_t^i, o_t^i)} AG^i(o_t, a_t), g(\epsilon, AG^i(o_t, a_t))\right) \end{aligned} \tag{17}$$

where T is the total time steps of an episode τ, AG^i is the advantage function calculated by Eq. 16 and $g(\epsilon, AG^i)$ is the clip function:

$$g(\epsilon, AG^i) = \begin{cases} (1+\epsilon)AG^i, & AG^i \geq 0 \\ (1-\epsilon)AG^i, & AG^i < 0 \end{cases} \tag{18}$$

The update rule of agent i's value network V^i (ϕ_k^i denotes the weight matrix of network V^i at the k-th iteration) is:

$$\phi_{k+1}^i = \arg\min_{\phi^i} \frac{1}{|\mathcal{D}_k^i T|} \sum_{\tau^i \in \mathcal{D}_k^i} \sum_{t=0}^{T} (V^i(o_t^i) - R_t^i)^2 \tag{19}$$

where R_t^i is the discounted return of agent i at time t:

$$R_t^i = r_t^i + \gamma r_{t+1}^i + \gamma^2 r_{t+2}^i + \cdots \tag{20}$$

After all agents' value and policy networks are updated, we can train the group value network V^{gru} (ϕ_k^{gru} denotes the weight matrix of network V^{gru} at the k-th iteration) by:

$$\phi_{k+1}^{gru} = \arg\min_{\phi^{gru}} \frac{1}{|\mathcal{D}_k T|} \sum_{\tau^i \in \mathcal{D}_k} \sum_{t=0}^{T} (V^{gru}(s_t) - R_t^{gru})^2 \tag{21}$$

where R_t^{gru} is the discounted return of the wind farm at time t.

$$R_t^{gru} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \tag{22}$$

The complete training process of MAPO is showed in Algorithm 1.

IV. Results

In our simulations, the observation o^i of turbine i includes its rotor speed ω_r^i, pump and motor pressure differences (P_p^i and P_m^i). The group state s is formed by concatenating all agents' observations. In the training curves, the solid line represents the average episode return of 5 trials started from random time seeds, and the standard deviation of the episode return of the 5 trials bounds the shaded region of a curve. There are two criteria for evaluating the performance of RL algorithms in wind farm control tasks: cumulative return (the solid line) and stability (the shaded region). High returns show that the tested control policy is effective in wind farm power generation, and the small shaded region signifies the corresponding agents can achieve similar performance under fluctuating initial conditions and vice versa. To reproduce the results, we provide the parameters used in the HWT-based farm simulator and the hyper-parameters of MAPO in Table I - Table III, respectively. The pseudo-code of MAPO is shown in Algorithm 1. In addition, we employ two useful techniques, namely policy smoothing regularization and dual value network, to reduce the variance of results during training.

During training and testing, the time step in Fast.Farm is set to 0.00625s. The total simulation time of one episode (the period from turbines launch to stop) in final testing is 3600s, while this number is 250s in training. The inflow surface (left) of the wind field follows a normal distribution of: $V_x = \mathcal{N}(10, 4), V_y = \mathcal{N}(0, 5), V_z = \mathcal{N}(0, 1)(m/s)$, where $\mathcal{N}$ denotes the normal distribution. Prior to calculating the wake dynamics, the ambient wind is generated by the inflow module in FAST.Farm at the beginning of each episode. The parameters of the NREL 5-MW reference wind turbine used in our simulations are listed in Table I.

A. Comparative evaluations

Fig. 8 compares the training curves of MAPO traced by the cumulative returns in 200 episodes, with the benchmark results of MADDPG and the greedy control policy (MPPT). We conclude that MAPO can forcefully raise the wind farm power generation, which suggests the agents have learned how to cope with the wake effect in turbine arrays. As shown in Fig. 9, the RL agents' strategy involves slightly reducing the power output of the upstream turbine (WT1) to weaken its wake effect

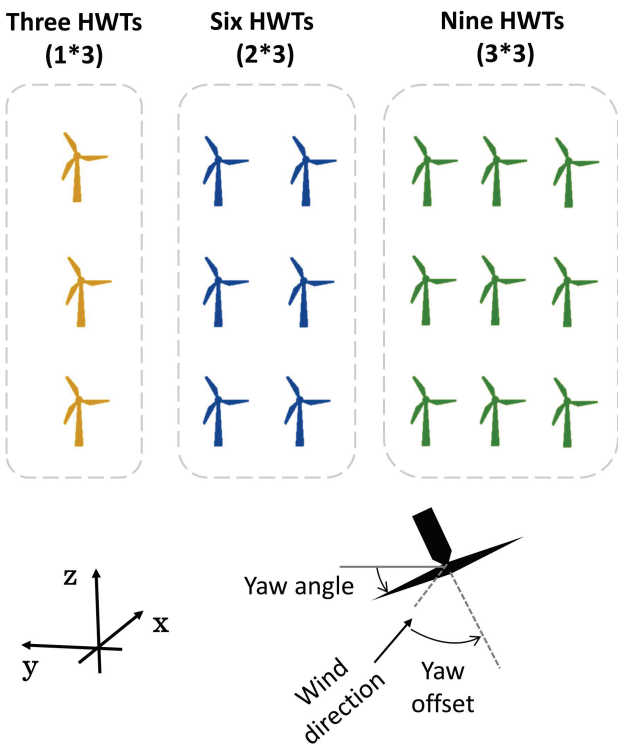

Fig. 7. Layouts of the tested three wind farms.

TABLE I
Parameters of the wind farm and wind turbine

Wind farm parameters	Unit	Value
Size	m^3	$1000 * 3000 * 1000$
Timestep	s	0.00625
Radial increment	m	5.0
Number of Radii	/	40
Number of wake Planes	/	136
Air density	Kg/m^3	1.29
Speed of sound	m/s	331
Atmospheric pressure	Pa	101,325
Wind turbine parameters	**Unit**	**Value**
Rating power	W	5e6
Rotor, Hub Diameter	m	126, 3
Hub Height	m	90
Rotor Mass	kg	110,000
Nacelle Mass	kg	240,000
Tower Mass	kg	347,460
Number of blade stations	/	49
Airfoil	/	NACA64_A17

on downstream turbines. During the training process, upstream turbines aim to seek an equilibrium that can maximize the power output of their downstream turbines while minimizing their losses.

In both Fig. 8 and Fig. 9, the variance (shaded region) of MAPO is relatively large in the initial training stage because we encourage agents to explore new states and policies at this stage. Afterward, all agents focus on maximizing the group return, implying that the objectives of agents are consistent now (coordination). As a result, the variance gradually diminishes to a low level. In

TABLE II
Parameters of the hydraulic transmission system

Name	Sign	Unit	Value
High pressure oil line length	L	m	100
Oil pipe line internal diameter	r	m	0.25
Density of mineral oil	ρ	$kg \cdot m^3$	917
Kinematic viscosity of oil	ν	m^2/s	4^{-5}
Effective bulk modulus of oil	E	Pa	1.039
Pump displacement	D_p	L/rev	626
Motor displacement	D_m	L/rev	4.9
Viscous damping of pump	B_p	$N \cdot m \cdot s$	5e4
Viscous damping of motor	B_m	$N \cdot m \cdot s$	2.5
Pump Coulomb friction coefficient	C_{fp}	-	0.02
Motor Coulomb friction coefficient	C_{fm}	-	0.02
Pump laminar leakage coefficients	C_{sp}	$m^3/s/Pa$	$7.1e{-11}$
Motor laminar leakage coefficients	C_{sm}	$m^3/s/Pa$	$7.0e{-11}$

TABLE III
Hyper-parameters of MAPO

Name	Value	Name	Value
Learning rate	1e-4	Clip range ϵ	0.2
Discounter coefficient	0.99	λ return	0.95
Activation function	tanh	Layer units	[64, 64]
Episodes	200	Batch size	1024

contrast, the variance of MADDPG remains high even at the end of training. Thus the policy learned by MAPO is more stable than MADDPG for deployment in real-world HWT-based wind farms. The curves of MAPO and MADDPG have both converged after being sufficiently trained by samples collected from FAST.Farm. Notably, the convergence value of MAPO is significantly greater than that of MADDPG, indicating that MAPO can increase the power generation of HWT-based wind farms more than MADDPG.

To illustrate how MAPO captures wind changes and

Algorithm 1 Multi-Agent policy optimization for a wind farm with n HWTs

For all $i = 1, 2, \cdots, n$, initialize the weight vectors ϕ_0^{gru}, ϕ_0^i and θ_0^i of V_0^{gru}, V_0^i and π_0^i, respectively.
for $k = 0, 1, 2, \cdots$ do
  Collect set of trajectories $\mathcal{D}_k$ which includes $\mathcal{D}_k^i = \{\tau_j^i | j = 1, 2, \cdots, J\}, \forall i = 1, 2, \cdots, n$ by running policy π_k in the simulator;
  Compute rewards-to-go R_t^{tol} and $[R_t^1, R_t^2, \cdots, R_t^n]$;
  for each agent $i = 1, 2, \cdots, n$ do
    Compute advantage estimates AG_t^i based on Eq. (16);
    θ_{k+1}^i: Update the policy π_{k+1}^i by maximizing the clip objective - Eq. (17);
    ϕ_{k+1}^i: Fit the value function V_{k+1}^i by regression on mean-squared error - Eq. (19);
  end for
  ϕ_{k+1}^{tol}: Fit the group value function V_{k+1}^{gru} by regression on mean-squared error: Eq. (21).
end for

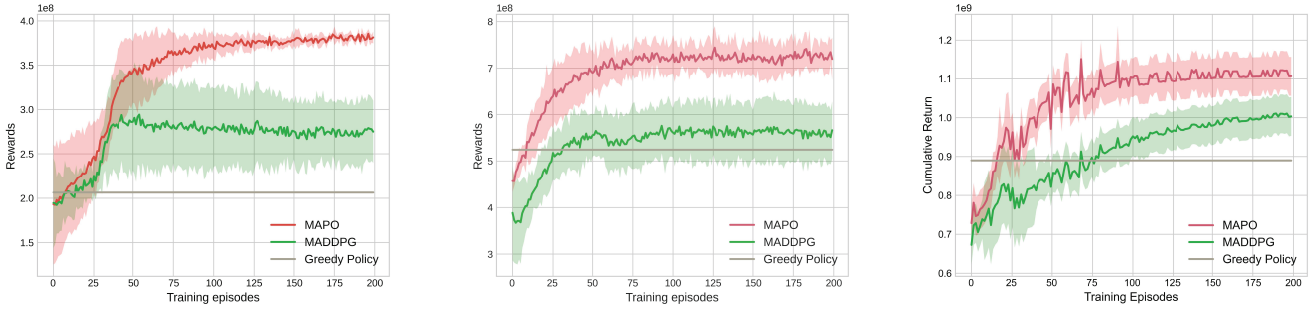

Fig. 8. Comparison of MAPO with MADDPG and the greedy control policy. Left: results of the wind farm composed of three hydrostatic wind turbines; Middle: results of the wind farm composed of six hydrostatic wind turbines. Right: results of the wind farm composed of nine hydrostatic wind turbines. Please see Fig. 7 for the layouts of the three wind farms.

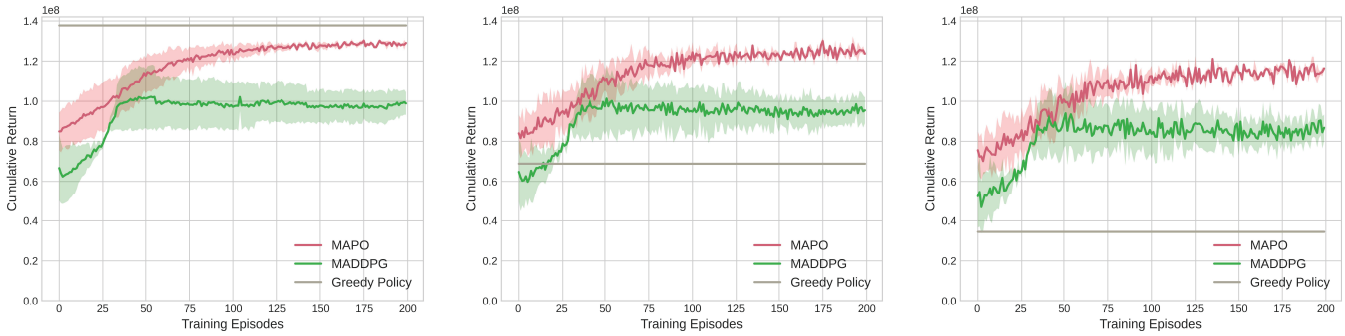

Fig. 9. Training curves of each HWT in a wind farm consisting of three HWTs. The sequence of them is: WT1, WT2 and WT3.

maximizes the power output of the wind farm, we generated heat maps during the training process. Fig. 10a shows the wake effect of upstream wind turbines on downstream turbines, indicating that without additional control, the turbines located in the wake planes would experience a significant decrease in the wind energy captured. In contrast, Fig. 10b shows the learned strategy that controls turbines to avoid wakes during the training process under the similar state, where the wind direction is mainly along the x-axis. In this strategy, each turbine selects a suitable yaw angle to minimize the impact of its wake on the surrounding turbines. Figs. 10c-d demonstrate the control strategies learned by the turbines to adapt to changes in the wind direction along the y-axis. As observed, all turbines have adjusted their yaw angles to align with the direction of the inflow wind, thereby maximizing wind speed on their rotational planes. Moreover, they have also been rotated to an optimal angle, directing their wakes towards a direction that has minimal effect on surrounding turbines.

We also test the final trained MAPO control policy via embedding it into wind farms and Table IV lists the test results. In this table, the *mean* column shows the average power output of the wind farm over five episodes, each lasting 3600 seconds. This data directly reflects the amount of power generated by wind farms. The *std* column indicates the standard deviation of the mean power output across the five episodes, which helps to evaluate the effect of different initial conditions on the performance of the controllers. The *max* and *min* columns respectively represent the highest and lowest power output values during the five episodes, and the difference between them, $|max-min|$, measures the power fluctuations. Based on the results presented in this table, it can be concluded that the MAPO controller is the most effective at driving wind turbines to generate power, and it demonstrates greater stability across the different episodes compared to the other controllers. Additionally, the wind turbine controlled by the MAPO controller exhibits less power output fluctuation, indicating higher power quality. Fig. 11 shows the variations in power output of the nine-turbine wind farm. Compared with the greedy control policy and wake steering-a fine industrial method derived from a relatively low-fidelity wind farm model named FLORIS [22], the wind farm manipulated by MAPO generated more power, which is consistent with the training curves. What's more, the power output by the MAPO-driven wind farm is more stable thanks to a fourth-order filter being used to smooth the control actions.

Since our HWT-based wind farm model, adapted from FAST.Farm, includes the sub-structural dynamics of HWTs, which is an advantage over other wind farm

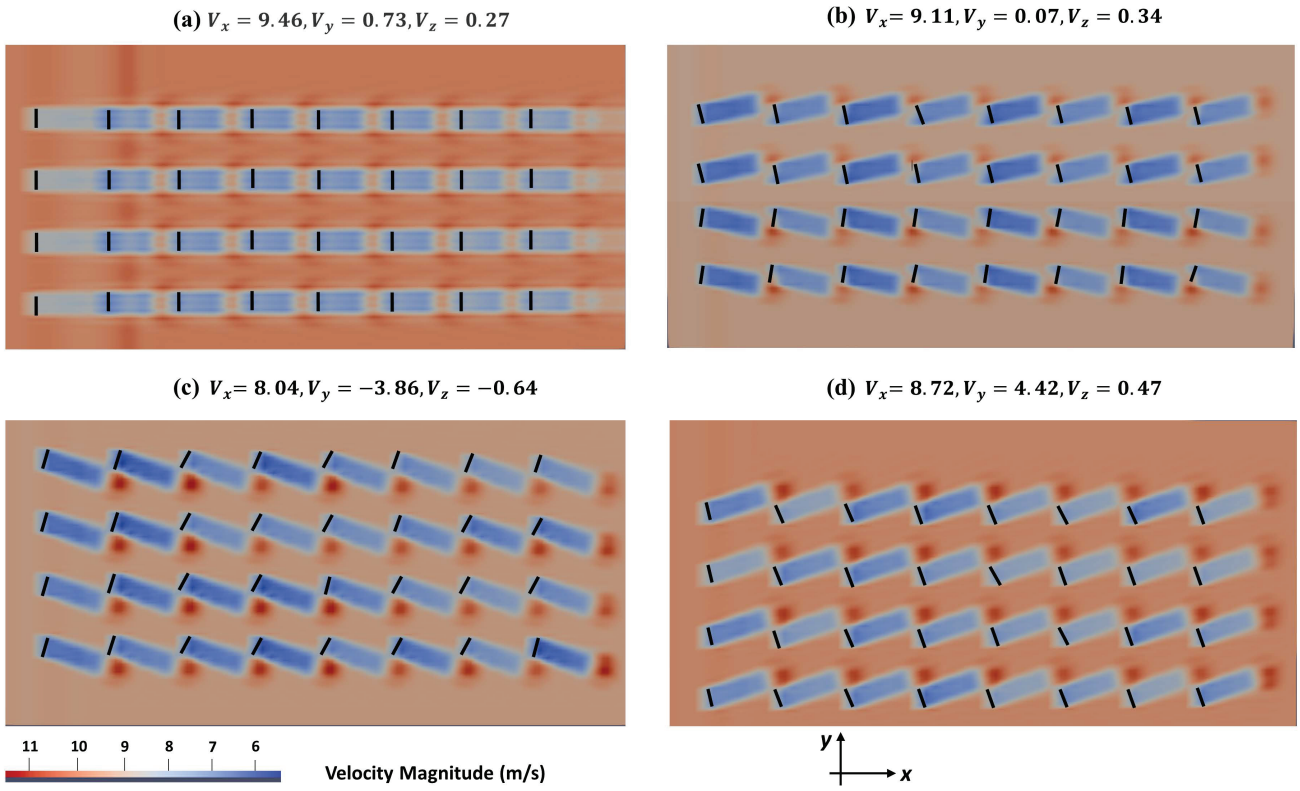

Fig. 10. The yaw control policy of MAPO for overcoming the wake effect.

TABLE IV
Test results of three controllers in four wind farms, unit (W)

Farm	Method	mean	std	max	min
1*3	MPPT	7.9767e6	2.1635e6	1.0413e7	5.8736e6
	Wake Steering	8.5399e6	1.8913e6	1.0868e7	6.6319e6
	MAPO	1.0097e7	1.2692e6	1.1823e7	9.7644e6
2*3	MPPT	1.5940e7	3.1723e6	1.9271e7	1.1834e7
	Wake Steering	1.6414e7	2.0236e6	1.9021e7	1.4130e7
	MAPO	1.9593e7	1.9280e6	2.1302e7	1.8206e7
3*3	MPPT	2.5057e7	2.5458e6	2.2675e7	2.9152e7
	Wake Steering	2.6155e7	3.3494e6	3.0546e7	1.9254e7
	MAPO	2.8620e7	1.3961e6	3.0235e7	2.5837e7
4*8	MPPT	7.7148e7	8.1264e6	6.3488e7	9.7482e7
	Wake Steering	9.0032e7	1.1331e7	7.5314e7	1.2136e8
	MAPO	9.9584e7	3.7853e6	9.1420e7	1.1527e8

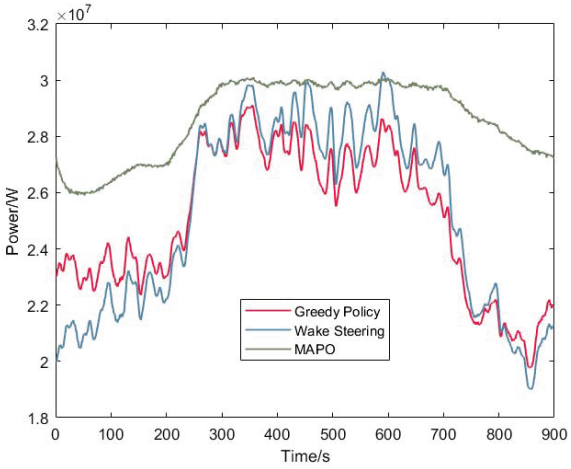

Fig. 11. Power output of the nine-turbine wind farm.

models, we analyzed the flapwise tip deflection of one blade and the fore-aft displacement of the tower of the front-left HWT in a six-turbine farm layout under MAPO, MADDPG, and the greedy control policy (Fig. 12). The results show that none of these three control strategies cause unusual vibrations of the blade and tower, and other HWTs have similar results. This implies that HWTs operate within safe structural limits under these three controllers.

Furthermore, MAPO has an advantage: when additional turbines are installed in the wind farm, we can transfer the weights of the value and policy networks to new turbines as the pre-trained model. It can greatly facilitate the sample efficiency of the algorithm.

B. Parameter analysis

In MAPO, we use a group value network with the input of the group state s to estimate the future wind farm return, and an individual value network for each agent with the input of its observation o to estimate its future return. Without violating the principle of CTDE, the

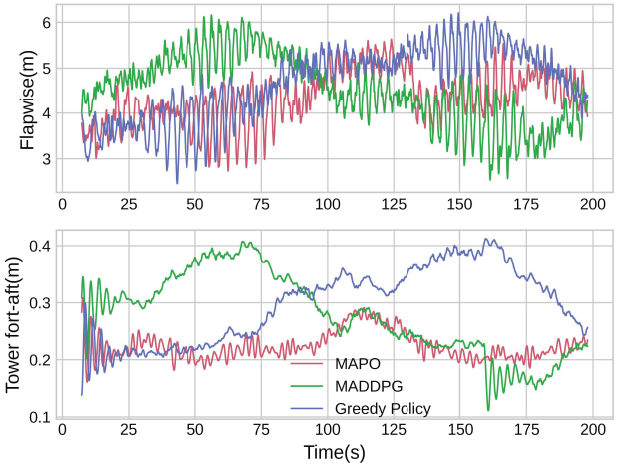

Fig. 12. Displacements of the Blade 1 and tower of the front-left HWT in the 6-turbine wind farm, under different control policies. Top: Blade 1 flapwise tip deflections. Bottom: Tower fore-aft displacements.

input of individual value networks can also be the group state s, which is referred to as MAPO-v2. Intuitively, MAPO-v2 can predict the agent return more precisely and faster as the network acquires more state information about the wind farm. However, Fig. 13 shows that, in terms of variance or cumulative return evaluation criteria, the performance of MAPO-v2 is distinctly worse than that of MAPO. Based on this result, we think that the observations of other HWTs are not conducive to the estimation of the target agent and even become noisy. Therefore, using the local information to estimate the individual return is more appropriate in the RL agent training.

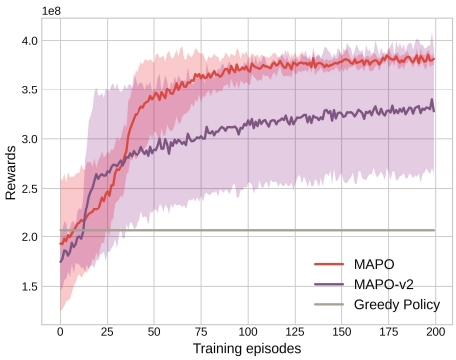

Fig. 13. Results of using local state or global state to estimate the agent return

The core idea of MAPO is to utilize a dynamical parameter η to balance the agent return and the group return. There are two additional options: 1) Fixed weight - η in Eq. 14 is set to a fixed value. 2) Agent weight - η in Eq. 14 is set to 0. The fixed weight method assigns equivalent weights to agents exploring their own policies and boosting the group return. This results in a large variance being maintained throughout the training process (Fig. 14). The objective of the agent weight method remains unchanged, causing low variances of results. However, the learned control policy eventually falls into a locally optimal solution (Fig. 14). In conclusion, the dynamical weight method exhibits its superiority thanks to a proper balance of the exploration-exploitation dilemma.

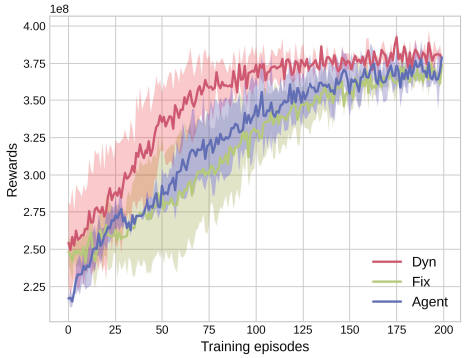

Fig. 14. Results of using different methods to balance the individual agent return and group return. Dyn: $\eta = num\_epoch/200$; Fix: $\eta = 0.5$; Agent: $\eta = 0$.

V. Conclusion

In this paper, we developed a HWT-based wind farm model by adapting FAST.Farm. HWTs have the potential to reduce the the maintenance cost of wind farms. We also proposed MAPO (multi-agent policy optimization) to optimize the wind farm control policy to boost the power generation of HWT-based farms. Our simulation results show that MAPO is of high performance in different wind farm layout cases and fluctuating environments. In addition, the control policy trained by MAPO has not caused any unusual vibrations in the substructures of HWTs, indicating it does not affect the safe operation of turbines. Moreover, the CTDE paradigm utilized in MAPO is beneficial for real-world deployment as it avoids the real-time communication issue between turbines within a wind farm.

Acknowledgment

This work has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 861398.

References

[1] Eurostat, "Renewable energy statistics," European Commission, Tech. Rep., 2021.

[2] X. Tong and X. Zhao, "Power generation control of a monopile hydrostatic wind turbine using an h_∞ loop-shaping torque controller and an lpv pitch controller," IEEE Transactions on control systems technology, vol. 26, no. 6, pp. 2165–2172, 2017.

[3] A. Wright and L. Fingersh, "Advanced control design for wind turbines," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2008.

[4] N. Diepeveen and A. Jarquin-Laguna, "Wind tunnel experiments to prove a hydraulic passive torque control concept for variable speed wind turbines," in Journal of Physics: Conference Series, vol. 555, no. 1. IOP Publishing, 2014, p. 012028.

[5] C. Santoni, U. Ciri et al., "Development of a high fidelity cfd code for wind farm control," in 2015 American Control Conference (ACC). IEEE, 2015, pp. 1715–1720.

[6] L. Wang, A. Tan, and Y. Gu, "A novel control strategy approach to optimally design a wind farm layout," Renewable energy, vol. 95, pp. 10–21, 2016.

[7] M. Abbes and M. Allagui, "Centralized control strategy for energy maximization of large array wind turbines," Sustainable Cities and Society, vol. 25, pp. 82–89, 2016.

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

[9] H. Dong and X. Zhao, "Wind-farm power tracking via preview-based robust reinforcement learning," IEEE Transactions on Industrial Informatics, 2021.

[10] H. Zhao, J. Zhao et al., "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.

[11] C. Bay, J. King, K. Johnson, and P. Stanfel, "A distributed reinforcement learning yaw control approach for wind farm energy capture maximization," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2020.

[12] S. Omidshafiei, J. Pazis et al., "Deep decentralized multi-task multi-agent rl under partial observability," 2017.

[13] J. M. Jonkman, J. Annoni et al., "Development of fast.farm: A new multi-physics engineering tool for wind-farm design and analysis," in 35th Wind Energy Symposium, 2017, p. 0454.

[14] F. D. Bianchi, H. De Battista, and R. J. Mantz, Wind turbine control systems: principles, modelling and gain scheduling design. Springer Science & Business Media, 2006.

[15] J. Meyers and C. Meneveau, "Large eddy simulations of large wind-turbine arrays in the atmospheric boundary layer," in 48th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, 2010, p. 827.

[16] N. D. Manring and R. C. Fales, Hydraulic control systems. John Wiley & Sons, 2019.

[17] A. J. Laguna, "Modeling and analysis of an offshore wind turbine with fluid power transmission for centralized electricity generation," Journal of Computational and Nonlinear dynamics, vol. 10, no. 4, p. 041002, 2015.

[18] J. Makinen, P. Pertola, and H. Marjamaki, "Modeling coupled hydraulic-driven multibody systems using finite element method," in The 1st Joint International Conference on Multibody System Dynamics, Lappeenranta, Finland, 2010, pp. 25–27.

[19] J. Schulman, P. Moritz et al., "High-dimensional continuous control using generalized advantage estimation," arXiv preprint arXiv:1506.02438, 2015.

[20] Y. Huang, X. Wang et al., "Soft policy optimization using dual-track advantage estimator," in 2020 IEEE International Conference on Data Mining (ICDM). IEEE, 2020, pp. 1064–1069.

[21] J. Schulman, F. Wolski et al., "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[22] E. Simley, P. Fleming, J. King, and M. Sinner, "Wake steering wind farm control with preview wind direction information," in 2021 American Control Conference (ACC). IEEE, 2021, pp. 1783–1789.

Yubo Huang received his B.S. degree in automation from Northwestern Polytechnical University, Xi'an, China in 2018, and his M.S. degree in control engineering from Shanghai Jiao Tong University, Shanghai, China in 2021. Currently he is a Marie-Curie Early Stage Researcher and is pursuing a Ph.D. degree in engineering from the University of Warwick, Coventry, U.K. His main research interests include reinforcement learning and control theory (with applications in wind farms), computational fluid dynamics (CFD), and nonlinear dynamics.

Shuyue Lin is a lecturer in electrical and electronic engineering at the University of Hull, UK. Before that, she was an assistant professor at Fuzhou University, China. She obtained her PhD in engineering (University of Warwick), MSc in control systems (Imperial College London), and BEng in electrical power engineering (the University of Bath and North China Electric Power University) in 2019, 2013, and 2012, respectively. Her current research interests include power generation and grid integration of renewable energies, fault detection and diagnosis in power systems.

Xiaowei Zhao received the Ph.D. degree in control theory from Imperial College London, London, U.K., in 2010. He was a Postdoctoral Researcher with the University of Oxford, Oxford, U.K., for three years before joining the University of Warwick, Coventry, U.K., in 2013. He is Professor of control engineering and an EPSRC Fellow with the School of Engineering, University of Warwick. His main research interests include control theory and machine learning with applications in offshore renewable energy systems, smart grids, and autonomous systems.