A Multi-Agent Reinforcement Learning Approach for Wind Farm Frequency Control

Yanchang Liang, *Student Member, IEEE*, Xiaowei Zhao, *Member, IEEE*, and Li Sun, *Member, IEEE*

***Abstract*— As wind turbines (WTs) become more prevalent, there is an increasing interest in actively controlling their power output to participate in the frequency regulation for the power grid. Conventional frequency regulation controllers use fixed gains, making it difficult for the WT to adjust its kinetic energy uptake to its operating conditions and to collaborate effectively with other WTs in the wind farm. In addition, the design of conventional frequency controllers does not consider their impacts on mechanical structure. To address these issues, we model the cooperative frequency control problem for all WTs in a wind farm as a decentralised partially observable Markov decision process (Dec-POMDP) and use a multi-agent deep reinforcement learning (MADRL) algorithm to solve it. We also develop a grid-connected wind farm simulation model based on MATLAB/Simulink and OpenFAST, which can reflect the detailed interactions between the electrical and mechanical components of WTs. Simulation results show that the proposed strategy is effective in reducing frequency drops and has less impact on mechanical structure deflections compared with traditional methods.**

***Index Terms*— Wind farm, frequency regulation, multi-agent deep reinforcement learning, wind turbine machinery.**

I. Introduction

In recent years, there has been significant growth in the penetration of offshore wind power into power systems and this trend is expected to continue in the future. Unlike conventional synchronous generators, wind turbines (WTs) do not naturally possess inertial response or participate in frequency disturbance events. The effective system inertia could be severely reduced with high penetration of wind power, resulting in high rates of change of frequency (RoCoF) and large frequency deviation after a sudden loss of generation or the connection of large loads.

Many works have investigated inertia control schemes for variable-speed WTs that temporarily release the kinetic energy stored in their rotating mass to arrest the frequency nadir. These schemes employ additional loops based on the measured frequency, i.e. inertia loop and droop loop [1]–[3]. However, in these schemes, the control gains are set to be fixed, making it difficult to adjust their kinetic energy uptake in real time based on information such as the wind speed and rotor speed of the WT. Due to the wake effects, the releasable kinetic energy of WTs in a wind farm varies significantly in time and space. The effect of wake effects on the inertial response of WTs is analysed in [4]. In order to capture kinetic energy according to the state of WT, adaptive gain schemes [5], [6] are proposed to replace fixed gain. These schemes set the gain of the two loops to be proportional to the kinetic energy in order to speed up the uptake of kinetic energy when it is more available. Wu *et al.* [7] propose an advanced control strategy with time-varying gains for inertia and droop control loops. The proposed strategy determines the gains according to the desired frequency response time. Liu *et al.* [8] propose a coordinated distributed model predictive control method for frequency control of power system with wind farms. An optimal fuzzy-based Proportional-Integral-Differential (PID) droop control method is proposed to improve the performance of wind farm frequency control [9]. Hasanien *et al.* [10] use the symbiotic biological search algorithm to tune PID parameters to improve the frequency response of a multi-area power system including wind farms. In addition, other meta-heuristic algorithms such as particle swarm optimisation [11] and artificial bee colony [12] are used to optimise the PID parameters for wind farm frequency regulation.

Although the above works consider that WTs should have different responses in different states, most of them ignore the synergistic operation between WTs. In [13], the primary frequency response of the WT is significantly improved by continuously adjusting its droop gain in response to wind velocities. However, this approach requires the WTs in the wind farm to be able to communicate with each other, as the droop gain of each WT is somewhat dependent on the performance of other WTs. In order to be free from the limitations of communication, this paper will focus on the use of local information to collaboratively control the WTs in a wind farm.

When WTs are involved in frequency regulation, their output power needs to change frequently in response to changes in frequency, which adds fatigue loads to WTs. However, the inertia and droop control methods proposed in previous works do not take into account the impact on the mechanical structure. To fill the research gap, this paper will design control policies that can reduce the impact on the mechanical structure. When analysing the interaction between the electrical

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 861398. *(Corresponding author: Xiaowei Zhao.)*

Y. Liang and X. Zhao are with the Intelligent Control and Smart Energy (ICSE) Research Group, School of Engineering, University of Warwick, CV4 7AL Coventry, U.K. (e-mail: Yanchang.Liang@warwick.ac.uk; Xiaowei.Zhao@warwick.ac.uk).

L. Sun is with the School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen, 518055 Shenzhen, China. (e-mail: sunli2021@hit.edu.cn).

and mechanical aspects of WTs, the WTs' flexible bodies, such as blades, tower and drive-train [14], are the focus of consideration. Therefore, this paper uses OpenFAST software [15] to model the detailed aerodynamics and structural systems of WTs, which can accurately simulate the dynamics and fatigue loads of WTs.

Due to the detailed consideration of the electrical, aerodynamic and mechanical components, conventional methods are difficult to improve the performance of frequency regulation controllers in such a complex system. In recent years, Deep Reinforcement Learning (DRL) has achieved great success in solving computationally challenging decision-making problems, such as Atari [16], Go [17], and StarCraft [18]. Due to its powerful model-free optimisation capabilities, DRL has recently been used for real-time control problems in wind farms, such as output power maximisation [19], [20] and power tracking [21]. However, these works do not take into account the fast frequency response of the wind farm, and they do not model the power grid or the mechanical structure of WTs.

In this paper, we use DRL algorithm to tune the frequency controller parameters for each WT in real time to improve frequency regulation performance. Specifically, we consider each WT as an agent and model the problem of collaboratively controlling WTs for frequency regulation as a decentralized partially observable Markov decision process (Dec-POMDP) [22]. We use multi-agent Proximal Policy Optimisation (MAPPO) [23], a multi-agent DRL (MADRL) algorithm, to solve this Dec-POMDP. MAPPO algorithm has achieved state-of-the-art performance in recent benchmark multi-agent cooperative learning tasks [24].

Our major contributions are listed below:

1) We model the problem of collaboratively controlling WTs for frequency regulation in a wind farm as Dec-POMDP, with the objective of improving frequency regulation as well as reducing the impact on the mechanical structure of WTs.
2) We solve the Dec-POMDP problem using MAPPO algorithm, which follows the centralised training and decentralised execution paradigm, thus enabling collaborative control of WTs without the need for communication.
3) We develop a detailed model of grid-connected wind farm, including aerodynamic, mechanical and electrical characteristics, to evaluate the impact of frequency control on the mechanical structure of WTs.

The rest of this paper is organised as follows. Section II presents the structure of the inertial and primary frequency controllers. The proposed simulation model of the wind farm is given in Section III. In Section IV, we model the joint frequency control problem for WTs in a wind farm as a Dec-POMDP. We elaborate on the MAPPO algorithm in Section V. In Section VI, we run experiments connecting OpenFAST to Simulink to demonstrate the effectiveness of the proposed method. Finally, we conclude this paper in Section VII.

II. Inertia and Primary Frequency Control

A conventional fixed gain inertia control scheme [1]–[3] is shown in Fig. 1, where the upper and lower loops are the inertia and droop loops respectively. The active power reference of WT, P_{ref}, consists of three terms: P_{MPPT}, for the maximum power point tracking (MPPT) control; ΔP_{ine}, the output of the inertia loop; and ΔP_{dro}, the output of the droop loop.

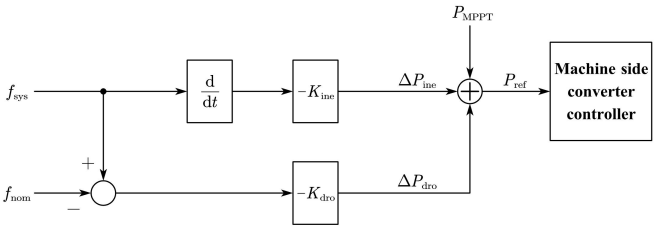

Fig. 1. Frequency regulation controller structure.

ΔP_{ine} can be expressed as

$$\Delta P_{\mathrm{ine}} = -K_{\mathrm{ine}} \frac{\mathrm{d}f_{\mathrm{sys}}}{\mathrm{d}t} \tag{1}$$

where K_{ine} is the inertial gain and f_{sys} denotes the measured system frequency. The function of the differentiator $\frac{\mathrm{d}}{\mathrm{d}t}$ is to calculate the RoCoF, and the inertial gain K_{ine} determines how much the active power output increases when the system frequency drops.

ΔP_{dro} can be expressed as

$$\Delta P_{\mathrm{dro}} = -K_{\mathrm{dro}}(f_{\mathrm{sys}} - f_{\mathrm{nom}}) \tag{2}$$

where K_{dro} is the droop gain and f_{nom} denotes the nominal frequency of power system. A high droop gain allows the droop loop to provide a large output.

III. Wind Farm Modelling

In this paper, the full-scale converter (FSC) WT with permanent magnetic synchronous generator (PMSG) is studied as an example, but the proposed method is not limited to this type of WT. FSC-WT is widely used in offshore wind power due to its simple structure, high power generation efficiency, reliable operation and low maintenance. As shown in Fig. 2, the FSC-WT consists of a turbine, a generator and an FSC. The FSC system can be further divided into a machine side converter (MSC), a DC link and a grid side converter (GSC). The output of FSC-WT is rectified by MSC and then supported by the capacitor, and the energy is fed into the grid via GSC.

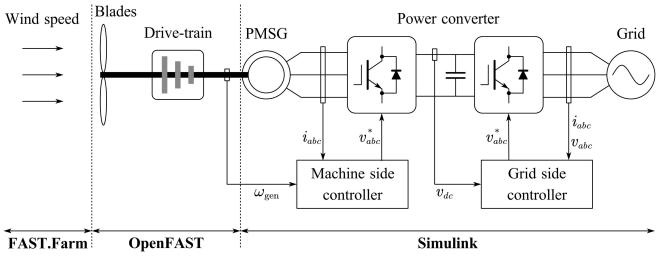

Fig. 2. Developed FSC-WT model using FAST.Farm, OpenFAST and Simulink.

The softwares and specific modules used to simulate the grid-connected wind farm in this paper are shown in Fig. 3.

The wind speed of the wind farm is generated by FAST.Farm [15], the aerodynamic, mechanical and actuator subsystems of each WT are simulated by OpenFAST, and the electrical subsystems of WTs and the power grid are simulated in MATLAB/Simulink. FAST.Farm provides detailed simulations of wake deficits, advection, deflection, meandering and merging, with consideration of the physics for wind-farm-wide ambient wind in the atmospheric boundary layer. OpenFAST models WT as a combination of rigid and flexible bodies. The rigid bodies include the platform, nacelle, hub, gears, tail and part of the yaw system, while the flexible bodies include the blades, tower and drive-train. OpenFAST can describe the dynamics of a 3-bladed WT in detail with up to 24 degrees of freedom (DOFs). Our focus is on the DOFs of flexible bodies, as they are a matter of mechanical lifetime. As shown in Fig. 4, the DOFs of flexible bodies include blade bending (blade flap-wise and edge-wise deflection), tower bending (tower fore-aft and side-to-side displacement), and drive-train torsion.

IV. Dec-POMDP Formulation

For a wind farm consisting of N WTs, we tune the inertia gain and droop gains for each WT in real time at a series of discrete time $t = 1, 2, \cdots$. We consider each WT as an agent, and model the joint frequency regulation problem as a Dec-POMDP, where the major components are as follows:

*1) **State**:* At each time step t, each WT n' observation o_t^n consists of its wind speed, rotor speed, rotor torque, generator power, pitch angle and mechanical structure information. The mechanical structure information includes blade flap-wise tip deflection $d_{n,t}^{\text{b-flap}}$, blade edge-wise tip deflection $d_{n,t}^{\text{b-edge}}$, tower fore-aft displacement $d_{n,t}^{\text{t-fore}}$, tower side-to-side displacement $d_{n,t}^{\text{t-side}}$, and drive-train acceleration $d_{n,t}^{\text{drive}}$. The state of the entire wind farm consists of the observations of all WTs, i.e. $s_t = \{o_t^1, o_t^2, \ldots, o_t^N\}$.

*2) **Action**:* At each time step t, the action of each WT n includes the inertia gain and droop gain in current time step, i.e. $a_t^n = [K_n^{\text{ine}}(t), K_n^{\text{dro}}(t)]$. Both $K_n^{\text{ine}}(t)$ and $K_n^{\text{dro}}(t)$ are non-negative numbers with upper limits $K_{\max}^{\text{ine}}$ and $K_{\max}^{\text{dro}}$ respectively. The joint action of all agents is denoted as $\boldsymbol{a}_t = \{a_t^1, a_t^2, \ldots, a_t^N\}$.

*3) **Reward**:* After all WT agents take actions, they receive a shared reward for reducing frequency deviations, adjusting RoCoF, and reducing vibrations of mechanical structures:

$$r_t = C_{\text{dro}} r_t^{\text{dro}} + C_{\text{ine}} r_t^{\text{ine}} + C_{\text{mech}} r_t^{\text{mech}} \tag{3}$$

where $C_{\text{dro}}, C_{\text{ine}}, C_{\text{mech}}$ are the adjustable weight coefficients for each term and $r_t^{\text{dro}}, r_t^{\text{ine}}, r_t^{\text{mech}}$ are defined as follows.

r_{dro} is used to reduce the deviation of the system frequency from the nominal frequency:

$$r_t^{\text{dro}} = -(f_{\text{nom}} - f_{\text{sys}})^2 \tag{4}$$

The larger the deviation of the system frequency f_{sys} from the nominal frequency f_{nom}, the smaller the r_t^{dro}.

r_t^{ine} is used to adjust the RoCoF:

$$r_t^{\text{ine}} = \begin{cases} \dfrac{\mathrm{d}f_{\text{sys}}}{\mathrm{d}t}, & f_{\text{sys}} < f_{\text{nom}} \\ 0, & f_{\text{sys}} = f_{\text{nom}} \\ -\dfrac{\mathrm{d}f_{\text{sys}}}{\mathrm{d}t}, & f_{\text{sys}} > f_{\text{nom}} \end{cases} \tag{5}$$

When $f_{\text{sys}} < f_{\text{nom}}$, a drop in the system frequency f_{sys} will result in a negative r_t^{ine}, and the faster the drop, the smaller the r_t^{ine}. A rise in f_{sys} will result in a positive r_t^{ine}, and the faster it rises, the greater the r_t^{ine}. On the other hand, when $f_{\text{sys}} > f_{\text{nom}}$, a positive r_t^{ine} is used to excite the frequency decrease, while a negative r_t^{ine} is used to suppress the frequency increase.

r_t^{mech} is used to reduce the vibration of mechanical structures:

$$\begin{aligned} r_t^{\text{mech}} = -\frac{1}{N}\sum_{n=1}^{N}(2|\dot{d}_{n,t}^{\text{b-flap}}| + |\dot{d}_{n,t}^{\text{b-edge}}| + \\ 2|\dot{d}_{n,t}^{\text{t-fore}}| + |\dot{d}_{n,t}^{\text{t-side}}| + |\dot{d}_{n,t}^{\text{drive}}|) \end{aligned} \tag{6}$$

where $\dot{d}_{n,t}^{\text{b-flap}}, \dot{d}_{n,t}^{\text{b-edge}}, \dot{d}_{n,t}^{\text{t-fore}}, \dot{d}_{n,t}^{\text{t-side}}, \dot{d}_{n,t}^{\text{drive}}$ are the standardised mechanical quantities. For example, we record the blades flap-wise deflections over a long period of time and calculate the mean as $\bar{d}^{\text{b,flap}}$ and the standard deviation as $\sigma^{\text{b,flap}}$. If the blades flap-wise deflection of WT n at time step t is $d_{n,t}^{\text{b,flap}}$, then the value after standardisation is

$$\dot{d}_{n,t}^{\text{b-flap}} = \frac{d_{n,t}^{\text{b-flap}} - \bar{d}^{\text{b-flap}}}{\sigma^{\text{b-flap}}} \tag{7}$$

Therefore, the larger the deviation of $\dot{d}_{n,t}^{\text{b-flap}}$ from its mean, the smaller r_t^{mech} will be. Since blade flap-wise deflections typically have more severe extreme external loads than edge-wise deflections [25], and tower fore-and-aft displacements usually have more damage equivalent loads than side-to-side displacements (as shown in Fig. 13), we give them a larger weighting factor in Eq. (6).

It is worth noting that other forms of penalty functions can also be used for Eq. (4)–(6) to reduce frequency deviations and mechanical structure deflections. The frequency and mechanical quantities in a time step are variable, so we can average multiple samples in a time step to calculate the reward. The cumulative discounted reward from time step t to the end of the control horizon is defined as the return R_t:

$$R_t = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau \tag{8}$$

where $\gamma \in [0, 1)$ is the discount factor that is used to trade off the importance between immediate and future rewards.

*4) **Policy and Value Functions**:* A policy is a mapping from states to the actions that should be taken when in those states, denoted as π. The team of agents attempt to learn a joint policy $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_N\}$ that maximises their expected cumulative reward. The value function of a state s_t under policy $\boldsymbol{\pi}$ is defined as the expected return when starting in s_t and following $\boldsymbol{\pi}$ thereafter:

$$V^{\boldsymbol{\pi}}(s_t) = \mathbb{E}_{\boldsymbol{\pi}}[R_t | s_t] \tag{9}$$

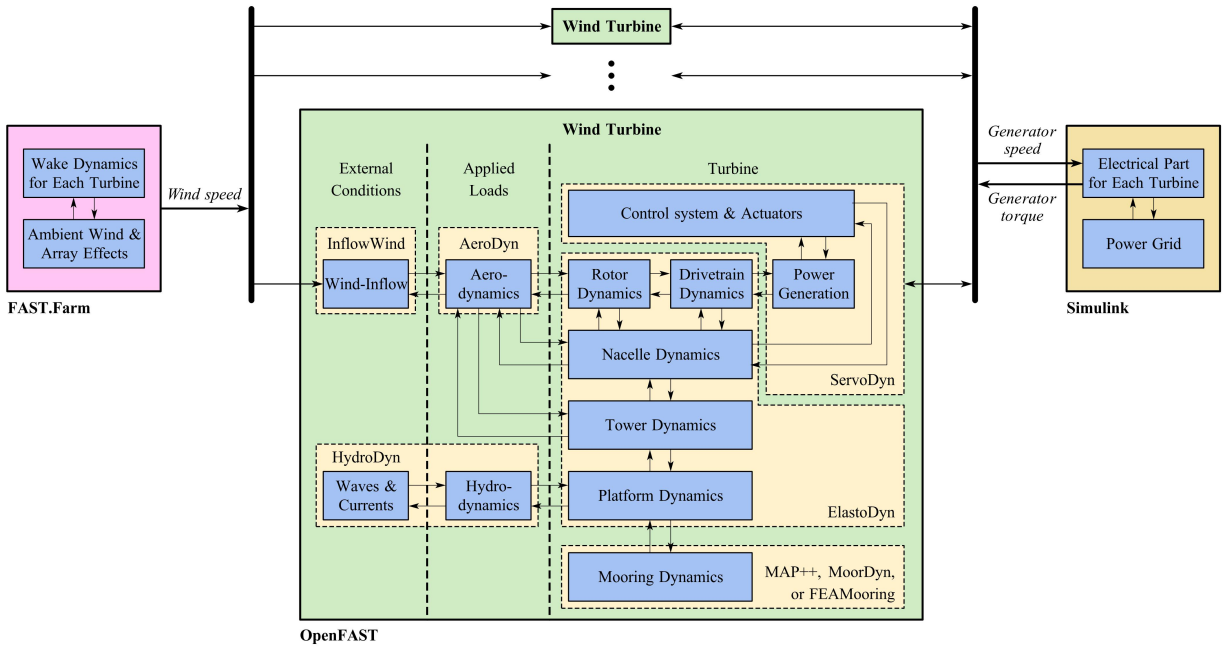

Fig. 3. The softwares and specific modules used to simulate the grid-connected wind farm.

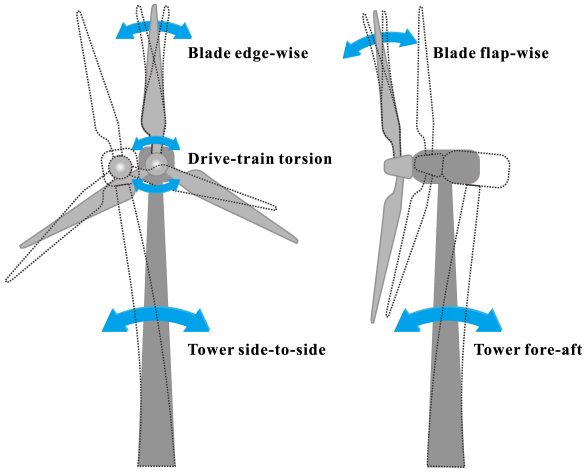

Fig. 4. DOFs of WT flexible bodies in the simulation model.

The value of taking action $\boldsymbol{a}_t$ in state s_t under policy $\boldsymbol{\pi}$ is defined as the expected return starting from s_t, taking the action $\boldsymbol{a}_t$, and thereafter following policy $\boldsymbol{\pi}$:

$$Q^{\boldsymbol{\pi}}(s_t, \boldsymbol{a}_t) = \mathbb{E}_{\boldsymbol{\pi}}[R_t | s_t, \boldsymbol{a}_t] \tag{10}$$

The advantage function $A^{\boldsymbol{\pi}}(s_t, \boldsymbol{a}_t)$ corresponding to a policy $\boldsymbol{\pi}$ describes how much better it is to take a specific action $\boldsymbol{a}_t$ in state s_t than to choose the action at random according to $\boldsymbol{\pi}(\cdot|s_t)$, assuming that the actions will always be taken according to $\boldsymbol{\pi}$ later:

$$A^{\boldsymbol{\pi}}(s_t, \boldsymbol{a}_t) = Q^{\boldsymbol{\pi}}(s_t, \boldsymbol{a}_t) - V^{\boldsymbol{\pi}}(s_t) \tag{11}$$

V. MADRL Algorithm

A. Proximal Policy Optimisation

Proximal Policy Optimisation (PPO) is a practical policy gradient method developed in [26], and is effective for optimising large non-linear policies such as deep neural networks. PPO is an improvement on Trust Region Policy Optimisation (TRPO) [27], where the former can be optimised by a first-order optimiser and is therefore simpler to implement. PPO retains many of the advantages of TRPO, such as monotonic improvement, and has been empirically proven to have better sample complexity. The objective function of TRPO, also known as "surrogate" objective, is

$$\max_{\theta} \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t \right] \tag{12}$$

subject to

$$\hat{\mathbb{E}}_t[\text{KL}[\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t)]] \leq \delta \tag{13}$$

where θ is the parameters of the stochastic policy function π_θ (usually the parameters of a neural network), and $\hat{A}_t$ is an estimator of the advantage function at time step t. The expectation $\hat{\mathbb{E}}_t[\cdot]$ indicates the empirical average over a finite batch of samples and $\text{KL}[\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t)]$ denotes the Kullback-Leibler (KL) divergence between $\pi_{\theta_{\text{old}}}$ and π_θ. The constraint is to limit the size of the policy update so that the new policy does not differ significantly from the old one. TRPO approximately solves this problem using the conjugate gradient algorithm, after making a linear approximation to the objective and a quadratic approximation to the constraint.

We next describe how PPO reduces the computational complexity of TRPO. Let $u_t(\theta)$ denote the probability ratio $u_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$, so $u(\theta_{\text{old}}) = 1$. TRPO maximizes

$$L(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t \right] = \hat{\mathbb{E}}_t[u_t(\theta)\hat{A}_t] \tag{14}$$

subject to the KL divergence constraint. The constraint limits the size of the policy update but makes it significantly more difficult to calculate. To reduce the computational complexity,

PPO modifies the surrogate objective to penalize changes to the policy that move $u_t(\theta)$ away from 1:

$$L^{\text{clip}}(\theta) = \hat{\mathbb{E}}_t[\min(u_t(\theta)\hat{A}_t, \text{clip}(u_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{15}$$

where ϵ is a hyperparameter, usually 0.1 or 0.2. The term $\text{clip}(u_t(\theta), 1-\epsilon, 1+\epsilon)$ modifies the surrogate objective by clipping the probability ratio, thus suppressing r_t beyond the interval $[1-\epsilon, 1+\epsilon]$. The final objective becomes the lower bound of the unclipped objective by taking the minimum value of the clipped and the unclipped objective.

B. Independent Proximal Policy Optimisation

Although PPO has achieved state-of-the-art in many benchmark environments [28], it can only be used to solve single agent problems. A straightforward approach to the current multi-agent control problem is to use the independent PPO (IPPO) algorithm. IPPO performs the PPO algorithm for each agent in a multi-agent system and has been shown to work effectively in a number of multi-agent tasks [29]. However, direct application of the single-agent DRL algorithm to a multi-agent system may result in a non-stationary training environment. Specifically, when considering the training of one agent n in a multi-agent system, the policies of other agents can be considered as part of the environment. The Bellman equation [30] at this point can be derived as

$$V^{\pi_n}(s) = \sum_a \pi_n(a_n|s) \sum_{s',r} p(s', r|s, a_n, \boldsymbol{\pi}^-)(r + V^{\pi_n}(s)) \tag{16}$$

where $p(s', r|s, a_n, \boldsymbol{\pi}^-)$ denotes the state transition probability in the multi-agent system and $\boldsymbol{\pi}^-$ denotes the policies of agents other than n. Since the policy of each agent is updated synchronously, the state transition function p is non-stationary, and thus the convergence of the Bellman equation cannot be guaranteed. However, each agent's policy is updated synchronously and is limited to its own partial observations during training, resulting in a non-stationary state transition p. As a result, the convergence of Bellman equation for IPPO is not guaranteed.

C. MAPPO

MAPPO algorithm [23] improves the decentralised training of the IPPO algorithm into a centralised one to improve stability. MAPPO maintains a policy π_θ (also known as actor) for each agent and a centralised value function $V_\phi(s)$ (also known as critic) for all agents. Both the actor and critic can be approximated by neural networks, forming the neural network structure shown in Fig. 5. The critic network $V_\phi(s)$ maps the global state to a state value. The critic network is only used for the training process to reduce variance, during which additional global information is available to improve performance. During execution process, critic network is discarded so that global information is not required, enabling decentralised execution of policies. The actor network π_θ maps each agent's observation o_t^n to the mean and standard deviation vectors of a Multivariate Gaussian Distribution, from which an action is sampled, in continuous action spaces. The actor network can be shared among all WT agents, as all WTs in a wind farm are usually homogeneous. The objective function for training the actor network is

$$\begin{aligned} L(\theta) = &\frac{1}{B \cdot N} \sum_{b=1}^{B} \sum_{n=1}^{N} \big[\min(r_b^n(\theta) A_b^n, \\ &\text{clip}(r_b^n(\theta), 1-\epsilon, 1+\epsilon) A_b^n) - \eta \mathcal{H}(\pi_\theta(o_b^n)) \big] \end{aligned} \tag{17}$$

where $r_b^n(\theta) = \frac{\pi_\theta(a_b^n|o_b^n)}{\pi_{\theta_{\text{old}}}(a_b^n|o_b^n)}$. A_b^n is computed using the Generalized Advantage Estimation (GAE) method [31], $\mathcal{H}$ is the Shannon Entropy, and η is the entropy coefficient hyperparameter.

The training objective of the critic network is to minimise the loss function

$$\begin{aligned} L(\phi) = &\frac{1}{B \cdot N} \sum_{b=1}^{B} \sum_{n=1}^{N} \max \big[(V_\phi(s_b^n) - \hat{R}_b)^2 \\ &(\text{clip}(V_\phi(s_b^n), V_{\phi_{\text{old}}}(s_b^n) - \varepsilon, V_{\phi_{\text{old}}}(s_b^n) + \varepsilon) - \hat{R}_b)^2 \big] \end{aligned} \tag{18}$$

where B refers to the batch size. Algorithm 1 shows the training process of MAPPO algorithm.

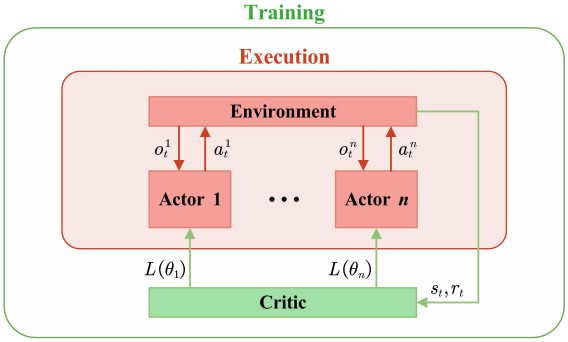

Fig. 5. Neural network structure of MAPPO algorithm.

Algorithm 1: MAPPO algorithm for controlling WTs

input: Initialize parameters θ, ϕ; batch size B; entropy loss weight η; $\mathcal{D} \leftarrow \{\}$; λ for GAE(λ)
for *trajectory* $= 1$ **to** *number of trajectories* **do**
  foreach *time step* t **do**
    $\boldsymbol{a}_t = [\pi_\theta(o_t^n), \forall n = 1, \ldots, N]$
    Execute actions $\boldsymbol{a}_t$, observe r_t, s_{t+1}
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, \boldsymbol{a}_t, r_t, s_{t+1})\}$
  Randomly choose B transitions from $\mathcal{D}$
  Compute advantage $\hat{A}_1, \ldots, \hat{A}_B$ and returns $\hat{R}_1, \ldots, \hat{R}_B$ via GAE(λ)
  for *each training epochs* **do**
    Calculate $L(\phi)$ according to (18)
    Update critic by minimising the loss $L(\phi)$
    Calculate $L(\theta)$ according to (17)
    Update policy by maximising $L(\theta)$

VI. Case Study

As shown in Fig. 6, the simulation experiments are carried out on a two-area test system, which is scaled down from a two-area benchmark power system [32]. The two-area system has four synchronous generators, each rated at 45 MVA, which are divided equally between the two areas. The wind farm is made up of 9 NREL 5 MW Baseline WTs. The NREL 5 MW Baseline WT is a conventional horizontal-axis, three-bladed, variable speed WT with blade-pitch control. The wind speeds are generated by FAST.Farm simulating the operation of WTs without frequency control, where the wind speeds for WT 1-3 are shown in Fig. 7. Although we use different control policies during frequency regulation, the change in wake of one WT does not affect the other WTs during this period (20–30s seconds for primary frequency control [33]), as the wake propagation time between neighbouring WT rows is about 100 seconds for common configurations and wind speeds [34].

In this experiment, the length of each time step is set to 0.2 seconds, i.e., the values of inertia and droop gains are updated every 0.2 seconds. The coefficients $C_{\mathrm{dro}}, C_{\mathrm{ine}}, C_{\mathrm{mech}}$ of the reward function are set to $200, 10^5, 0.1$ respectively. For the training settings of MADRL algorithms, the batch size is 2048, the entropy coefficient is 0.01, the discount factor is 0.99 and the GAE parameter is 0.95. Both the actor network and the critic network have 3 hidden layers with 256, 128, and 64 neurons respectively. The activation function for each hidden layer is the rectified linear unit (ReLU) [35]. All experiments are carried out on a computer with a 8-core 2.90 GHz Intel Core i7-10700 processor and 32 GB of RAM.

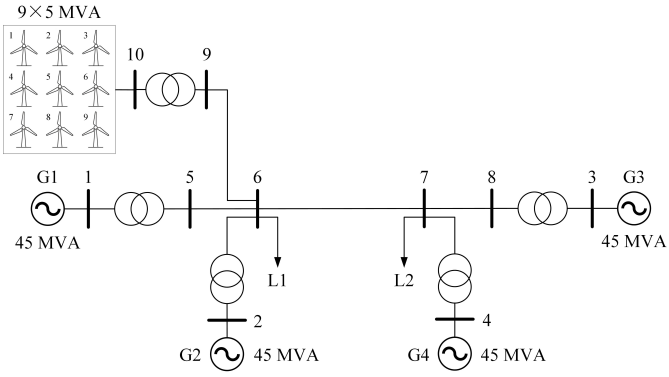

Fig. 6. Four-machine two-area test system with a wind farm.

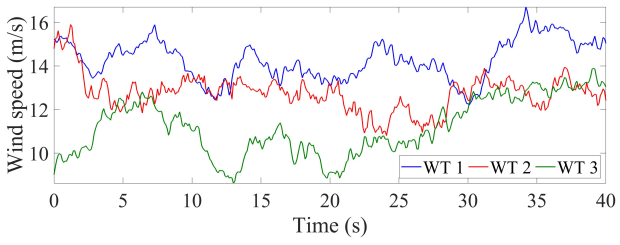

Fig. 7. Wind speeds of WT 1-3.

In this experiment, a 40 MW load is suddenly connected at load 1 to cause a frequency drop and to test the frequency regulation performance of the wind farm under different control methods. The proposed method is compared with three other conventional methods, including no control, constant gain and decreasing gain. No control means that WTs do not participate in frequency regulation. In the constant gain setting, the inertia gain is constant at 25 and the droop gain is constant at 6 [7]. Decreasing gain is a coordinated control method proposed by [7] that combines a parabolic function for the inertia variable and a linear function for the droop variable. In this method, large gains are set to increase the power output from the WTs at the instant of frequency drop. As the time increases, the gains decreases gradually, preventing the WTs from overdecelerating.

The proposed method is also compared with three other MADRL algorithms, i.e. PPO, IPPO and MADDPG [36]. As described in Section V-B, IPPO algorithm uses a decentralised training, decentralised execution approach, which means that each WT agent is trained without considering the condition of the other WTs. MADDPG extends deep deterministic policy gradient (DDPG) into a multi-agent policy gradient algorithm where decentralised agents learn a centralised critic based on the observations and actions of all agents.

We use different MADRL algorithms to train control policy to solve the proposed Dec-POMDP problem. Fig. 8 illustrates the variation in returns during training for the three MADRL algorithms. The training process for the MAPPO algorithm with 20×10^5 training steps takes about 137.67 hours. After 10×10^5 training steps, the returns of all MADRL algorithms converge and outperform conventional methods. The average return of IPPO algorithm has larger confidence intervals during training than MAPPO and converges to a smaller value than MAPPO. This suggests that not considering the cooperation of WTs will cause instability in the training process and also produce less optimal policy. MADDPG is also less stable and has a smaller convergence value than MAPPO, due to the fact that the deterministic policy used in MADDPG has a poor exploration capability and is easy to cause overestimation of the action-value function. PPO algorithm has the similar return with MAPPO, but it requires a centralised controller and relies on communication networks. During execution, PPO gathers global information from the wind farm, calculates the control signals and communicates them to each WT. In contrast, MAPPO can distribute the computational burden across all WTs during execution, resulting in better computational efficiency and real-time performance, and does not require a communication network thus preserving privacy and not being affected by communication link failures. In conventional methods, the decreasing gain has the highest return, while the case without frequency control has the lowest return.

After training, we analyse the frequency regulation performance of MAPPO algorithm and its impact on the mechanical structure. The MAPPO-based gains of WT 1-3 as a function of time are shown in Fig. 9. The gains of the conventional methods are also shown in Fig. 9. Since the MAPPO algorithm tunes the gains based on the state of the WT, the gains are different for different WTs. The conventional methods do not consider the state of the WT, so all WTs have the same gains. The load is suddenly connected at 5s causing

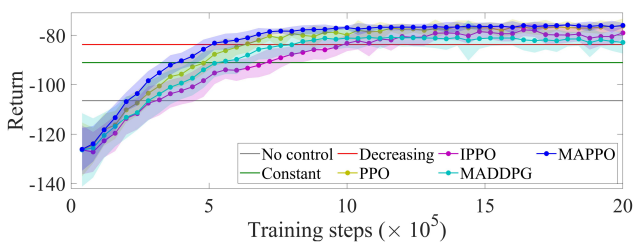

Fig. 8. Average returns for each method during the training process. Error bars are the 95% confidence intervals across 6 experiments with different random seeds.

a frequency drop. It can be seen that MAPPO-based inertia and droop gains increase significantly in the instant after the frequency drop, accelerating the uptake of turbine kinetic energy, which improves the performance of the wind farm frequency regulation during this period. The MAPPO-based gains then decrease between 5s and 10s, thus reducing uptake of kinetic energy from the turbines, which facilitates recovery of the turbines' operation and reduces mechanical vibrations.

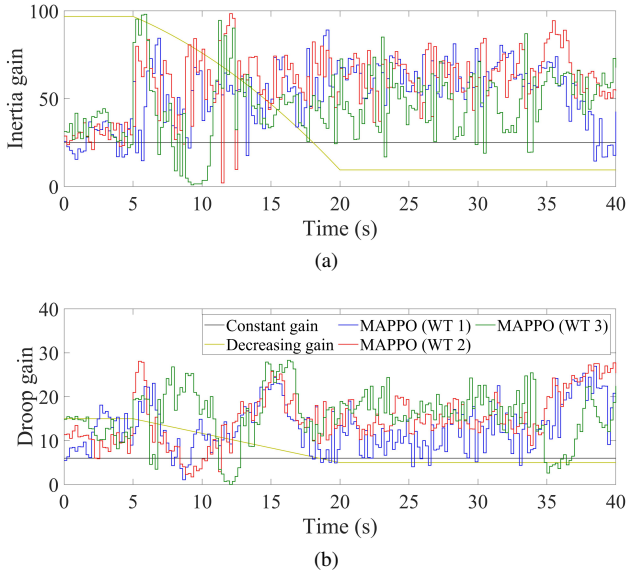

Fig. 9. Inertia and droop gains. (a) Inertia gain and (b) droop gain.

The frequency variation curves for different methods are shown in Fig. 10(a). It can be seen that both MAPPO-based gains and decreasing gains are effective in suppressing the drop in frequency, with frequency nadirs of 49.663 Hz and 49.674 Hz respectively. The constant gain has a frequency nadir of 49.583 Hz, which is less effective than time-varying gains. The frequency nadir without frequency control is the smallest, at 49.475 Hz. The output power of the wind farm for the different methods are shown in Fig. 10(b). It can be seen that the wind farm controlled by MAPPO algorithm outputs more active power in the 3s after the frequency drop than other methods, thus providing more effective support for frequency recovery. In addition, the MAPPO algorithm has smaller active power fluctuations than other methods between 10s and 40s, making the frequency more stable during this time.

The mechanical response of WT 1 for the different methods is shown in Fig. 11. It can be seen that blade flap-wise tip

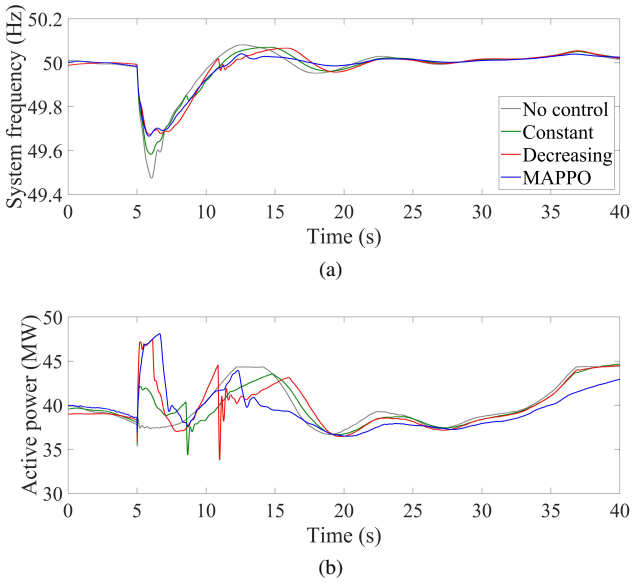

Fig. 10. Response of the power system. (a) Frequency and (b) active power of the wind farm.

deflection and tower fore-aft displacement are greater with time-varying gains than with constant gains. This means that the faster the WT releases kinetic energy to the grid, the better the frequency regulation, but also the greater the mechanical structure vibrates. The MAPPO-based gains and the decreasing gains have similar frequency regulation capabilities, but the mechanical structures have smaller deflections with MAPPO-based gains.

We next investigate the effect of the coefficient C_{mech} on the proposed method. The frequency of the wind farm and the mechanical response of WT 1 for different values of C_{mech} are shown in Fig. 12. It can be seen that as C_{mech} gets smaller, the frequency nadir gets higher, but the vibration of the mechanical structure gets more severe. However, even if $C_{\text{mech}} = 0.01$, the proposed method is more effective in suppressing mechanical deflections than decreasing gain.

The damage equivalent loads are used to equate the fatigue damage represented by the rainflow cycle counting data to that caused by a single load range repeating at a single frequency [37]. The damage equivalent loads are given by the following formula:

$$M_{\text{eq}} = \sqrt[I]{\frac{\sum_i n_i M_i^I}{n_{\text{eq}}}} \tag{19}$$

where M_{eq} is the damage equivalent load and n_{eq} is set to the number of cycles when there is no frequency control. M_i is the amplitude moment in one load cycle. Each kind of M_i and the corresponding number of cycles n_i is given by the rainflow counting method [38]. I represents the slope of the M-N curve (applied moment vs. allowable cycles to failure) [39], for blades $I = 10$ and for tower $I = 5$ [40]. The damage equivalent loads for the blades and towers are shown in Fig. 13. It can be seen that the MAPPO-based gains have more damage fatigue loads than the fixed gains, but less than the decreasing gains. As C_{mech} increases, the damage fatigue loads for MAPPO-based gains decrease significantly.

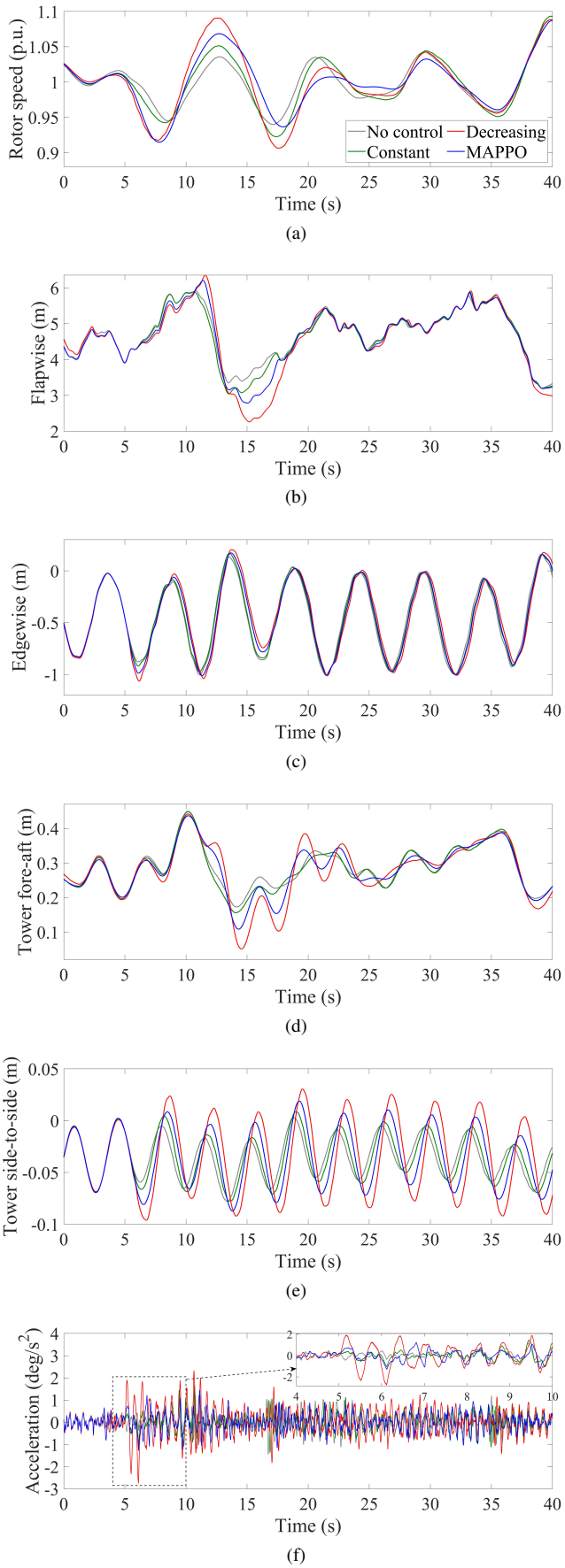

Fig. 11. Mechanical response of WT 1. (a) Rotor speed, (b) blade flap-wise, (c) blade edge-wise tip deflections, (d) tower fore-aft, (e) tower side-to-side displacements, and (f) drive-train acceleration.

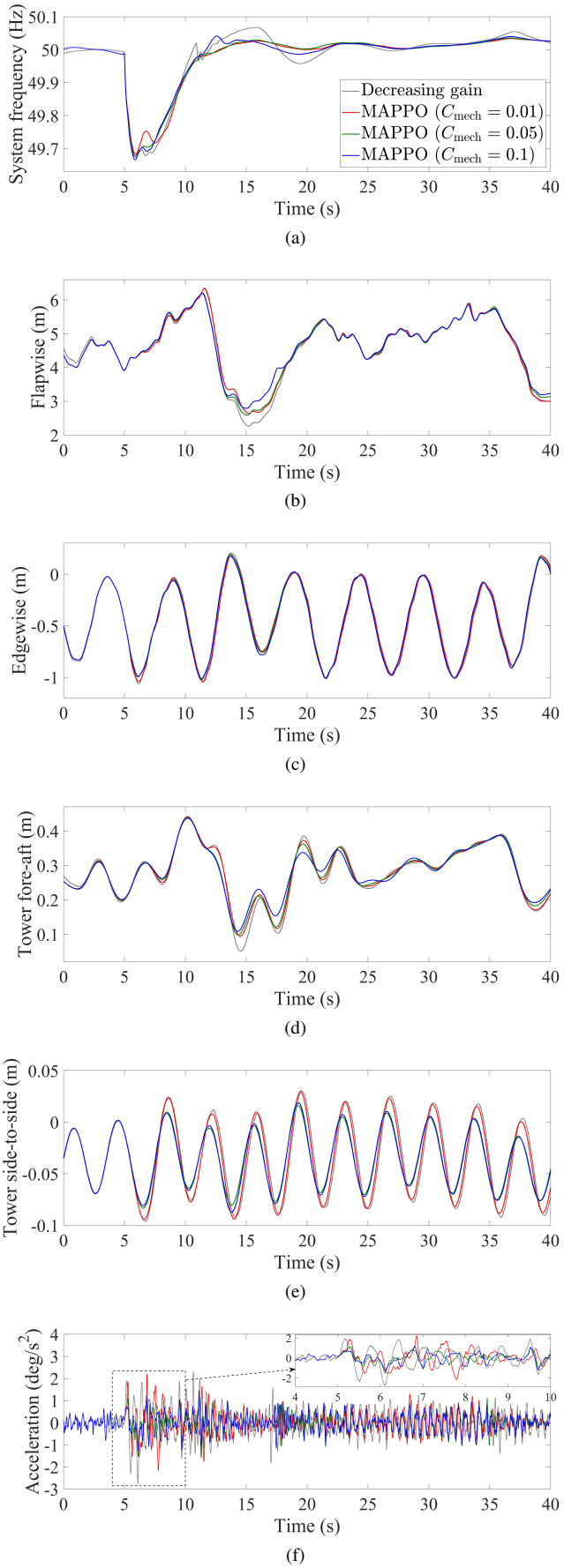

Fig. 12. Mechanical response of WT 1. (a) System frequency, (b) blade flap-wise, (c) blade edge-wise tip deflections, (d) tower fore-aft, (e) tower side-to-side displacements, and (f) drive-train acceleration.

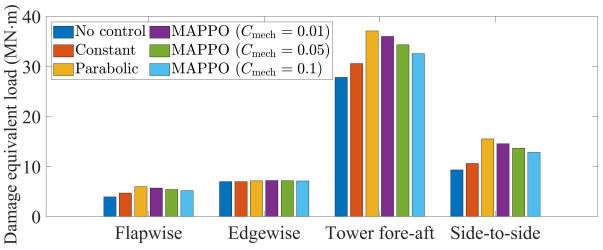

Fig. 13. Damage equivalent loads for each mechanical structure under different methods.

VII. Conclusion

In this paper, we model the cooperative frequency regulation problem of WTs in a wind farm as a Dec-POMDP and solve it using the MAPPO algorithm. Each WT tunes its inertia gain and droop gain in real time based on its own observation. FAST.Farm, OpenFAST and Simulink are used to simulate the aerodynamic, mechanical and electrical parts of a wind farm respectively. The simulation results show that the proposed method is more effective in increasing the frequency nadir, reducing frequency fluctuations and reducing mechanical structure deflections than conventional methods.

References

[1] J. Morren, S. W. De Haan, W. L. Kling, and J. Ferreira, "Wind turbines emulating inertia and supporting primary frequency control," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 433–434, 2006.

[2] J. F. Conroy and R. Watson, "Frequency response capability of full converter wind turbine generators in comparison to conventional generation," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 649–656, 2008.

[3] J. Van de Vyver, J. D. De Kooning, B. Meersman, L. Vandevelde, and T. L. Vandoorn, "Droop control as an alternative inertial response strategy for the synthetic inertia on wind turbines," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1129–1138, 2015.

[4] S. Kuenzel, L. P. Kunjumuhammed, B. C. Pal, and I. Erlich, "Impact of wakes on wind farm inertial response," *IEEE Trans. Sustain. Energy*, vol. 5, no. 1, pp. 237–245, 2013.

[5] J. Lee, E. Muljadi, P. Srensen, and Y. C. Kang, "Releasable kinetic energy-based inertial control of a dfig wind power plant," *IEEE Trans. Sustain. Energy*, vol. 7, no. 1, pp. 279–288, 2015.

[6] J. Lee, G. Jang, E. Muljadi, F. Blaabjerg, Z. Chen, and Y. C. Kang, "Stable short-term frequency support using adaptive gains for a dfig-based wind power plant," *IEEE Trans. Energy Convers.*, vol. 31, no. 3, pp. 1068–1079, 2016.

[7] Y.-K. Wu, W.-H. Yang, Y.-L. Hu, and P. Q. Dzung, "Frequency regulation at a wind farm using time-varying inertia and droop controls," *IEEE Trans. Ind. Appl.*, vol. 55, no. 1, pp. 213–224, 2018.

[8] X. Liu, Y. Zhang, and K. Y. Lee, "Coordinated distributed mpc for load frequency control of power system with wind farms," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5140–5150, 2016.

[9] A. Abazari, H. Monsef, and B. Wu, "Load frequency control by de-loaded wind farm using the optimal fuzzy-based pid droop controller," *IET Renewable Power Generation*, vol. 13, no. 1, pp. 180–190, 2019.

[10] H. M. Hasanien and A. A. El-Fergany, "Symbiotic organisms search algorithm for automatic generation control of interconnected power systems including wind farms," *IET Generation, Transmission & Distribution*, vol. 11, no. 7, pp. 1692–1700, 2017.

[11] V. Gholamrezaie, M. G. Dozein, H. Monsef, and B. Wu, "An optimal frequency control method through a dynamic load frequency control (lfc) model incorporating wind farm," *IEEE Systems Journal*, vol. 12, no. 1, pp. 392–401, 2017.

[12] D. Kumar, A. Mishra, and K. Chatterjee, "Power and frequency control of a wind energy power system using artificial bee colony algorithm," in *2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM)*. IEEE, 2017, pp. 561–565.

[13] K. Vidyanandan and N. Senroy, "Primary frequency regulation by deloaded wind turbines using variable droop," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 837–846, 2012.

[14] G. P. Prajapat, N. Senroy, and I. N. Kar, "Wind turbine structural modeling consideration for dynamic studies of dfig based system," *IEEE Trans. Sustain. Energy*, vol. 8, no. 4, pp. 1463–1472, 2017.

[15] "Openfast documentation." [Online]. Available: https://openfast.readthedocs.io/en/main/#

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

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

[18] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4295–4304.

[19] H. Zhao, J. Zhao, J. Qiu, G. Liang, and Z. Y. Dong, "Cooperative wind farm control with deep reinforcement learning and knowledge-assisted learning," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 6912–6921, 2020.

[20] J. Xie, H. Dong, X. Zhao, and A. Karcanias, "Wind farm power generation control via double-network-based deep reinforcement learning," *IEEE Trans. Ind. Informat.*, 2021.

[21] H. Dong and X. Zhao, "Wind-farm power tracking via preview-based robust reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1706–1715, 2021.

[22] F. A. Oliehoek and C. Amato, *A concise introduction to decentralized POMDPs*. Springer, 2016.

[23] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of mappo in cooperative, multi-agent games," *arXiv preprint arXiv:2103.01955*, 2021.

[24] G. Papoudakis, F. Christianos, L. Schäfer, and S. V. Albrecht, "Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks," *arXiv preprint arXiv:2006.07869*, 2020.

[25] C. Muyan and D. Coker, "Finite element simulations for investigating the strength characteristics of a 5 m composite wind turbine blade," *Wind Energy Science*, vol. 5, no. 4, pp. 1339–1358, 2020.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[27] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[28] K. W. Loon, L. Graesser, and M. Cvitkovic, "Slm lab: A comprehensive benchmark and modular software framework for reproducible deep reinforcement learning," *arXiv preprint arXiv:1912.12482*, 2019.

[29] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in the starcraft multi-agent challenge?" *arXiv preprint arXiv:2011.09533*, 2020.

[30] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, vol. 6, no. 5, pp. 679–684, 1957.

[31] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[32] P. Kundur, "Power system stability," *Power system stability and control*, pp. 7–1, 2007.

[33] J. Aho, A. Buckspan, J. Laks, P. Fleming, Y. Jeong, F. Dunne, M. Churchfield, L. Pao, and K. Johnson, "A tutorial of wind turbine control for supporting grid frequency through active power control," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 3120–3131.

[34] C. Hwang, J.-H. Jeon, G.-H. Kim, E. Kim, M. Park, and I.-K. Yu, "Modelling and simulation of the wake effect in a wind farm," *Journal of International Council on Electrical Engineering*, vol. 5, no. 1, pp. 74–77, 2015.

[35] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[36] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *arXiv preprint arXiv:1706.02275*, 2017.

[37] K. Thomsen, "The statistical variation of wind turbine fatigue loads," Risø National Laboratory, Roskilde, Denmark, Tech. Rep., 1998.

[38] R. Sunder, S. Seetharam, and T. Bhaskaran, "Cycle counting for fatigue crack growth analysis," *International Journal of Fatigue*, vol. 6, no. 3, pp. 147–156, 1984.
[39] G. Freebury and W. Musial, "Determining equivalent damage loading for full-scale wind turbine blade fatigue tests," in *2000 ASME wind energy symposium*, 2000, p. 50.
[40] P. Frohboese and A. Anders, "Effects of icing on wind turbine fatigue loads," in *Journal of Physics: Conference Series*, vol. 75, no. 1. IOP Publishing, 2007, p. 012061.

Yanchang Liang (S'19) received the B.S. and M.S. degrees from the School of Electrical Engineering at North China Electric Power University, China, in 2018 and 2021 respectively. He is currently a Marie Curie Early Stage Researcher and PhD student at the University of Warwick, UK.

His current research interests include control, optimization, reinforcement learning, with applications to power systems and hydrogen fuel cells.

Xiaowei Zhao (M'09) received the Ph.D. degree in control theory from Imperial College London, London, U.K., in 2010. He was a Post-Doctoral 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 currently Professor of control engineering and an EPSRC Fellow with the School of Engineering, University of Warwick.

His main research areas are control theory and machine learning with applications in offshore renewable energy systems, smart grid, and autonomous systems.

Li Sun (S'15–M'21) received the B.Eng. degree and the M.Eng. degrees from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2013, and 2016 respectively, and the Ph.D. degree from The University of Hong Kong, Hong Kong, in 2019, both in electrical engineering. After that she worked as a Research Fellow with the University of Warwick, United Kingdom until June 2021. She is currently an Assistant Professor with the School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen, China. Her current research interests focus on the power system stability control, control design and optimization of islanded microgrids.