Power Regulation and Load Mitigation of Floating Wind Turbines via Reinforcement Learning

Jingjie Xie, Hongyang Dong, and Xiaowei Zhao

***Abstract*—Floating offshore wind turbines (FOWTs) are often subjected to heavy structural loads due to challenging operating conditions, which can negatively impact power generation and lead to structural fatigue. This paper proposes a novel reinforcement learning (RL)-based control scheme to address this issue. It combines individual pitch control (IPC) and collective pitch control (CPC) to balance two key objectives: load reduction and power regulation. Specifically, a novel incremental model-based dual heuristic programming (IDHP) strategy is developed as the IPC solution to reduce structural loads. It integrates the online-learned FOWT dynamics into the dual heuristic programming process, making the entire control scheme data-driven and free from dependence on analytical models. Furthermore, the proposed method differs from existing IDHP methods in that only partial system dynamics need to be learned, resulting in a simplified design structure and improved training efficiency. Tests using a high-fidelity FOWT simulator demonstrate the effectiveness of the proposed method.**

***Note to Practitioners*—This work achieves power regulation and load reduction simultaneously for FOWTs to guarantee the reliability of wind turbine operations. Such a task is still an open problem because existing FOWT controllers commonly rely on accurate turbine models and lack adaptability to potential uncertainties and errors in practical situations. A new data-driven, model-free control strategy based on the RL technique is developed to address these issues. Our method has the ability to capture potential changes in system dynamics by updating a so-called incremental model via online measurements. Unlike current advances in this direction that need to approximate the whole system dynamics, the proposed control algorithm only needs to update partial system information for the incremental model. This naturally simplifies the design structure and enhances learning effectiveness while providing adaptability and robustness against uncertainties and errors. The proposed control strategy can also be extended and implemented in other systems, such as autonomous systems and other renewable energy systems.**

***Index Terms*—Intelligent control, wind turbine control, reinforcement learning, wind energy.**

I. Introduction

THE development of floating offshore wind turbines (FOWTs) has been growing dramatically in recent years. Compared with conventional wind turbines, FOWTs have many merits, e.g. extended installation area (to the sea with water depth up to 900m) and the ability to harvest high-quality wind power resources. However, FOWTs commonly suffer from heavy structural loads since they are affected by the stochastic wind-wave environment [1], especially far offshore. That can lead to severe platform & tower motions and negatively influence the power capture efficiency and system stability [2]. Therefore, mitigating structural loads while regulating power generation is crucial for FOWTs to improve power quality, avoid structure damage, and prolong service life.

Pitch control is a commonly-employed and easy-to-implement method to maintain rated power and alleviate fatigue loads in the above-rated region [3]. Specifically, it regulates the blade pitch angles to change rotor thrust, serving as a restoring moment to stabilize the turbine tower and platform [4]. In many studies, pitch control can be subdivided into collective pitch control (CPC) and individual pitch control (IPC). In CPC, all blades are adjusted collectively by the same pitch angle command to maintain the output power at a desired rated value [5]. In contrast, they are controlled independently further by individual pitch signals in IPC to mitigate structural loads potentially [6].

A lot of pitch controllers have been designed for FOWTs. For example, classical proportional-integral-derivative (PID) controllers have been widely adopted [7], [8], [9]. PID-based pitch controllers have simple structures and are easy to build, but their load mitigation ability and robustness are limited. Advanced FOWT control approaches have been proposed to overcome the limitations of PID-based methods. For example, a low-authority linear-quadratic (LQ)-based IPC controller combined with an integral action-based CPC was designed in Ref. [1]. Ref. [10] proposed a radial basis function-based sliding mode controller to suppress the floating foundation motion and reduce power oscillation, and a further study in Ref. [11] developed a variable-gain high-order sliding mode pitch control strategy. Ref. [3] designed another pitch control method based on the sliding mode technique, in which an adaptive second-order sliding surface was considered. In Ref. [12], the blade pitch control of FOWTs was solved by an H_∞ method to achieve two competing objectives, i.e., power regulation and load mitigation. Notably, these elegant results are model-based and rely on accurate FOWT models. In addition, they have difficulties dealing with strict state and input constraints.

Model predictive control (MPC) has received extensive attention in recent years due to its ability to handle multi-objective optimal control problems under various constraints [13], [14], [15]. Several MPC-based methods have been successfully applied to wind turbine pitch control problems [13], [16], [17], [18]. Specifically, in Ref. [13], a nonlinear MPC-based IPC was investigated for FOWT to alleviate blade

This work has received funding from the EPSRC Supergen ORE Hub ECR Research Fund and the UK Engineering and Physical Sciences Research Council under grant number EP/S000747/1. J. Xie, H. Dong and X. Zhao (Corresponding Author) are with the Intelligent Control & Smart Energy (ICSE) Research Group, School of Engineering, University of Warwick, Coventry CV4 7AL, UK. Emails: jingjie.xie@warwick.ac.uk, hongyang.dong@warwick.ac.uk, xiaowei.zhao@warwick.ac.uk.

loads. Ref. [16] also employed MPC to stabilize power output and reduce dynamic loads & platform motion for FOWT. A constrained subspace predictive repetitive control (SPRC) approach was introduced in Ref. [17] for similar tasks. In Ref. [18], MPC was studied based on a Takagi-Sugeno fuzzy model to maintain the rated power. Although these MPC-based FOWT controllers have shown promising performance, they still highly rely on accurate system models for future state prediction. Those methods are sensitive to modelling errors and unmodeled dynamics. Due to the strong non-linearity and high complexity of FOWT, the accuracy of analytical models is hard to guarantee during the entire lifespan of FOWT, especially considering the harsh wind-wave environment [19].

These facts motivate us to develop a data-driven, model-free control strategy for FOWTs, rendering strong robustness and adaptability against unmodelled dynamics and uncertainties. This method also should balance the requirements of power generation regulation and structural load mitigation simultaneously. To this end, a novel reinforcement learning (RL)-based wind turbine pitch controller is proposed in this paper. Reinforcement learning is a disruptive technology for the operations of offshore renewable energy systems, including offshore wind farms and turbines [20]. It aims to find the control policy to optimize long-term rewards/costs via mining the data obtained through interactions with the environment [21]. It is noteworthy that several studies [22], [23] have applied RL methods to wind turbine pitch control problems, showing effectiveness in power regulation. However, these essential results ignore the load mitigation requirement - as we mentioned before, load mitigation is crucial for FOWTs. It is still a blank area to develop RL-based control algorithms for FOWTs that can not only reduce power fluctuation but also alleviate loads, and we fill this research gap in this paper. Specifically, an incremental wind turbine model-based dual heuristic programming (IDHP) algorithm is designed as the IPC to cooperate with a PI-based CPC. Such a framework allows the whole control system to achieve reliable power regulation performance and mitigate loads at the same time by regulating blade pitch angles. The proposed method does not rely on any pre-determined analytical models. It is able to capture critical system information via real-time measurements (by building a so-called incremental model online). In addition, a critic-actor structure is employed to approximate the long-term cost function and find the optimal control policy. The effectiveness of our method is validated by the FAST (Fatigue, Aerodynamics, Structures, and Turbulence) simulator developed by the US National Renewable Energy Laboratory (NREL) [24]. We further summarize the contributions of our paper from the following two aspects.

1. To our best knowledge, this is the first attempt to address the power regulation and load mitigation simultaneously for FOWTs via the RL technique. The proposed strategy is data-driven and model-free, which can solve optimal FOWT control problems without relying on accurate analytical models. Unlike existing RL-based wind turbine control task that considers only the power regulation issue, this paper aims to achieve two competing objectives, i.e. (1) reducing power fluctuations and (2) mitigating loads & motions on the tower and platform.

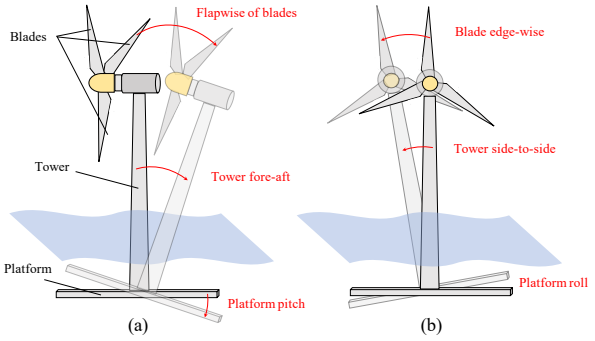

Fig. 1. The structure of FOWT with potential load-induced motions. (a) Out-of-plane direction. (b) In-plane direction.

2. The proposed method differs from current IDHP methods in that only partial system dynamics need to be learnt online. This novelty avoids the complex approximation and learning process for the whole system dynamics, leading to enhanced training efficiency and a simplified design structure.

The remainder of this paper is laid out as follows. The FOWT pitch control task is introduced in Section II. Our RL-based pitch controller considering power regulation and load mitigation is designed in Section III. Then, simulation results under different scenarios based on the FAST simulator are shown in Section IV. Some conclusions are given in Section V to finish this paper.

II. Problem Formulation

In this section, the FOWT pitch control problem is formulated, and the structure of FOWT, followed by the control objectives, is introduced in detail.

A. Floating Turbine System

The structure of a typical FOWT is depicted in Fig. 1. It also illustrates the potential motions in the out-of-plane (along-wind) and in-plane (cross-wind) directions. The motion in the out-of-plane direction is subjected to the major aerodynamic loads [1]. Therefore, this paper focuses on suppressing turbine motions/loads in the out-of-plane direction via pitch control.

Without loss of generation, we consider a 5-MW FOWT developed by NREL to design and evaluate our RL-based FOWT control method. As shown in Ref. [25], such a FOWT has 22 degrees of freedom (DoFs). In this work, the DoFs related to the out-of-plane motion are chosen for designing the pitch control strategy, including the first flap-wise bending mode for three blades, the first fore-aft tower bending mode, and the platform pitch. Accordingly, the structural dynamics of a FOWT with these DoFs for the IPC can be described as

$$\dot{\boldsymbol{s}} = f(\boldsymbol{s}) + G(\boldsymbol{s})\boldsymbol{a} \tag{1}$$

where $\boldsymbol{s} = [s_P \quad s_{TFA} \quad s_{FB1} \quad s_{FB2} \quad s_{FB3} \quad e_{w_g}]^T$ denotes the state vector, s_P, s_{TFA}, s_{FB1}, s_{FB2}, and s_{FB3} represent the platform pitch, tower fore-aft, and flap-wise of blade 1, blade 2, and blade 3, respectively, e_{w_g} is the error between the generator speed and the reference. In addition, $\boldsymbol{a} = [\beta_1 \quad \beta_2 \quad \beta_3]^T$ is the control input for IPC, and here β_1,

β_2, β_3 represent the individual pitch angles for three blades. Moreover, $f(\boldsymbol{s})$ and $G(\boldsymbol{s})$ denote the state and input function of the FOWT pitch control system. Different with mainstream wind turbine control methods, our design does not require the specific definitions of $f(\boldsymbol{s})$ and $G(\boldsymbol{s})$. In other words, the RL-based FOWT controller proposed in the following section is model-free.

B. Control Objectives

This paper focuses on wind turbine pitch control in the above-rated region to achieve power regulation and load mitigation. It has been reported by many researchers that the reduction of structural loads can cause the deterioration of the power quality [1], [3]. Moreover, improving the power quality usually has a negative impact on the platform motions. Such a contradictory situation brings the challenge of developing control algorithms that are not only capable of improving power production but also able to reduce loads & motions simultaneously. This challenge is addressed in our study by designing an RL-based pitch controller that includes both CPC and IPC. On the one hand, CPC regulates the power output by maintaining the generator speed at its rated levels. On the other hand, IPC solves an optimal control problem to reduce loads & motions exerted on the blade, tower, and platform in the out-of-plane direction. We define a continuous-time cost function for IPC:

$$\mathcal{Q}(s(t), a(t)) = \sum_{l=t}^{\infty} \gamma^{l-t} r(t) \tag{2}$$

where $l = t, t+1, ..., \infty$, $\gamma \in (0, 1)$ is a discount factor, r is called the one-step reward function, which is defined in a quadratic form of tracking errors. The one-step reward function $r(t)$ at time t is defined as

$$r(t) = (\boldsymbol{s}(t) - \boldsymbol{s}^*(t))^T Q_w (\boldsymbol{s}(t) - \boldsymbol{s}^*(t)) + \boldsymbol{a}(t)^T R_w \boldsymbol{a}(t) \tag{3}$$

where $\boldsymbol{s}^*(t)$ is desired state, Q_w and R_w are weight matrices.

As mentioned in the introduction, we develop an IDHP-based method in the next section to solve the above optimal control problem. Furthermore, integrating it with a collective pitch control strategy allows the whole control system to balance load mitigation and power regulation requirements.

III. RL-Based Controller for FOWTs

This section investigates an RL-based pitch control design framework for the FOWT power regulation and loads mitigation. A novel incremental FOWT pitch control dynamic model-based dual heuristic programming (IDHP) method is designed as IPC to alleviate the loads. Meanwhile, a proportional-integration (PI)-based controller is integrated with IPC as CPC to maintain the power output. The structure of our control system is shown in Fig. 2. The pitch angle control signal $\boldsymbol{\beta} \in \mathcal{R}^{3\times 1}$ follow

$$\boldsymbol{\beta} = \beta_c + \boldsymbol{a} \tag{4}$$

where $\beta_c \in \mathcal{R}^{1\times 1}$ is the collective pitch control signal obtained from the CPC scheme, and $\boldsymbol{a} = [\beta_1, \beta_2, \beta_3]^T \in \mathcal{R}^{3\times 1}$ denotes the individual pitch control signal derived from the IDHP-based IPC strategy.

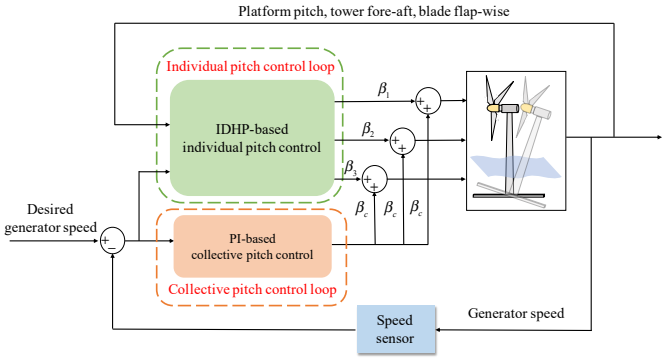

Fig. 2. The structure of the proposed RL-based control method for FOWT.

A. PI-Based CPC

For CPC, the PI method is applied to achieve the power regulation. It aims to maintain the power output to the rated value by minimizing the generator speed error e_{w_g}, which is defined as

$$e_{w_g}(t) = w_g(t) - w_g^{rated}(t) \tag{5}$$

where $w_g(t)$ and $w_g^{rated}(t)$ are the generator speed and its rated value at time t, respectively. The PI-based collective pitch control law can be described as

$$\beta_c(t) = K_P \cdot e_{w_g}(t) + K_I \cdot \int e_{w_g}(\tau) d\tau \tag{6}$$

where K_P and K_I denote proportional and integral gains, respectively.

B. IDHP-Based IPC

In addition to the PI-based CPC scheme, the IDHP-based IPC method is designed to mitigate loads and load-induced motions for FOWTs. Our design is built upon the approximate dynamic programming (ADP) theory.

The ADP technique, introduced by Werbos [26], [27], [28], has proven to be an effective reinforcement learning (RL) tool for addressing optimal control problems [29]. Distinct from other RL approaches such as deep Q-network (DQN) algorithms, which often exhibit slow convergence and excessive randomness during training [30], ADP methods can ensure rapid convergence under specific conditions [31] and maintain system stability throughout the iterative process [32]. Furthermore, when compared to the DDPG algorithm—a widely utilized RL method with a complex structure and substantial computational demands—ADP offers a comparatively simpler structure [33]. This reduced complexity effectively alleviates computational burdens, making ADP well-suited for wind turbine control challenges. Consequently, we develop a novel ADP-based solution for wind turbine control, aimed at addressing power regulation and load reduction requirements.

One of the most widely used ADP schemes is heuristic dynamic programming (HDP), which utilizes a critic network and an actor network to directly approximate the long-term reward function and control policy, respectively [34]. Another type of ADP is dual heuristic programming (DHP), which employs the critic network to approximate the derivatives of

the long-term reward function with respect to critic inputs [35]. It has been reported that estimating derivatives as in DHP is beneficial in reducing potential errors induced by the back-propagation process as in HDP [29]. Therefore, we chose DHP as the underlying design benchmark to develop our RL-based individual pitch controller.

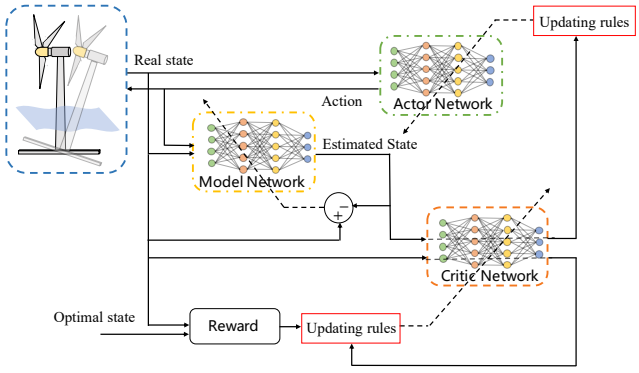

Fig. 3. The framework of a conventional DHP method for FOWT control.

The framework of a conventional DHP control scheme for a FOWT is illustrated in Fig. 3. Therein, the whole system dynamics is approximated either online or offline by an artificial neural network. On the one hand, an offline trained model can be inaccurate given uncertainties and modelling errors induced by long-time working conditions and complex environments. On the other hand, highly computational burden is inevitably required for online approximation of the entire system dynamics.

To address these problems, we combine the incremental model concept [35], [36], [37] with DHP. Such a design allows us to learn the key system information online but only needs to update the task-related part of system dynamics, bringing strong adaptability and robustness against uncertainties and modelling errors while keeping a moderate level of computational complexity. We elaborate our design as follows.

1) Incremental FOWT Model Learning:

The incremental model approximates nonlinear dynamics by taking the first-order Taylor series expansion around the latest system operation condition [35], [36]. It transforms the unknown control system into a new version that includes the incremental state and input with only partial model dynamics. The FOWT dynamics given in Eq. (1) under the stochastic wind, wave, and disturbances can be described as

$$\begin{aligned}\dot{\boldsymbol{s}} &= f(\boldsymbol{s}(t)) + \Delta f(\boldsymbol{s}(t))\delta_{\mathcal{W}} \\ &\quad + \left(G(\boldsymbol{s}(t)) + \Delta G(\boldsymbol{s}(t))\delta_{\mathcal{W}}\right)\boldsymbol{a}(t) + d(t)\end{aligned} \tag{7}$$

where $\boldsymbol{s}(t) \in \mathcal{R}^n$ is the system state at time t, $\boldsymbol{a}(t) \in \mathcal{R}^m$ is the control input, $f(\boldsymbol{s}(t))$ and $G(\boldsymbol{s}(t))$ are unknown functions. $\Delta f(\boldsymbol{s})$ and $\Delta G(\boldsymbol{s})$ represent model variations induced by stochastic wind and/or wave, and $\delta_{\mathcal{W}}$ is a binary constant (to be either 0 or 1) that is employed to represent the effects of wind and/or wave. If $\delta_{\mathcal{W}} = 0$, it means the nominal case. In contrast, when $\delta_{\mathcal{W}} = 1$, it denotes that the strong wind and/or wave occur, and accordingly the FOWT model deviates from its nominal case.

Then, according to the Taylor series expansion, the system can be approximated around the operating point $[\boldsymbol{s}(t_0), \boldsymbol{a}(t_0)]$ as follows [36]

$$\begin{aligned}\dot{\boldsymbol{s}}(t) &= \dot{\boldsymbol{s}}\left(t_0\right) + \bar{\mathbf{G}}\left[\boldsymbol{s}\left(t_0\right), \delta_{\mathcal{W}}\left(t_0\right)\right]\Delta\boldsymbol{a}(t) \\ &\quad + \left.\frac{\partial[\mathbf{H}(\boldsymbol{s}(t), \delta_{\mathcal{W}}(t))]}{\partial \boldsymbol{s}(t)}\right|_{\boldsymbol{s}(t_0),\boldsymbol{a}(t_0)} \cdot \Delta\boldsymbol{s}(t) \\ &\quad + \left.\frac{\partial[\mathbf{H}(\boldsymbol{s}(t), \delta_{\mathcal{W}}(t))]}{\partial \delta_{\mathcal{W}}(t)}\right|_{\boldsymbol{s}(t_0),\boldsymbol{a}(t_0)} \cdot \Delta\delta_{\mathcal{W}}(t) \\ &\quad + \Delta d(t) + \mathcal{O}\left(\Delta\boldsymbol{s}^2(t)\right)\end{aligned} \tag{8}$$

where t_0 is the latest sampling time, $\mathbf{H}(\boldsymbol{s}(t), \delta_{\mathcal{W}}(t)) = \bar{f}(\boldsymbol{s}(t), \delta_{\mathcal{W}}(t)) + \bar{G}(\boldsymbol{s}(t), \delta_{\mathcal{W}}(t))\boldsymbol{a}(t)$. $\bar{f}(\boldsymbol{s}(t), \delta_{\mathcal{W}}(t)) = f(\boldsymbol{s}) + \Delta f(\boldsymbol{s})\delta_{\mathcal{W}}$ and $\bar{G}(\boldsymbol{s}(t), \delta_{\mathcal{W}}(t)) = G(\boldsymbol{s}) + \Delta G(\boldsymbol{s})\delta_{\mathcal{W}}$ denote real models under uncertainties. $\Delta\boldsymbol{a}(t) = \boldsymbol{a}(t) - \boldsymbol{a}(t_0)$, $\Delta\boldsymbol{s}(t) = \boldsymbol{s}(t) - \boldsymbol{s}(t_0)$, $\Delta\delta_{\mathcal{W}}(t) = \delta_{\mathcal{W}}(t) - \delta_{\mathcal{W}}(t_0)$, and $\Delta d(t) = d(t) - d(t_0)$ are increments of control inputs, system states, uncertain dynamics trigger, and disturbances, respectively. Moreover, $\mathcal{O}\left(\Delta\boldsymbol{s}^2(t)\right)$ denotes higher-order residual terms. Note that if the sampling time is sufficiently short, the higher-order term and $\Delta\boldsymbol{s}(t)$ could be omitted [36], and the above system can be further simplified as

$$\begin{aligned}\Delta\dot{\boldsymbol{s}}(t) &\approx \bar{\mathrm{G}}\left[\boldsymbol{s}\left(t_0\right), \delta_{\mathcal{W}}\left(t_0\right)\right]\Delta\boldsymbol{a}(t) \\ &\quad + \left.\frac{\partial[\mathbf{H}(\boldsymbol{s}(t), \delta_{\mathcal{W}}(t))]}{\partial \delta_{\mathcal{W}}(t)}\right|_{\boldsymbol{s}(t_0),\boldsymbol{a}(t_0)} \cdot \Delta\delta_{\mathcal{W}}(t) + \Delta d(t)\end{aligned} \tag{9}$$

Eq. (9) is called an incremental model, and its discrete version satisfies

$$\boldsymbol{s}_{k+1} \approx \boldsymbol{s}_k + \bar{\mathbf{G}}_k \cdot \Delta\boldsymbol{a}_k + \frac{\partial \mathbf{H}_k}{\partial \delta_{\mathcal{W}_k}} \cdot \Delta\delta_{\mathcal{W}_k} + \Delta d_k \tag{10}$$

where k denotes the k-th discrete step, $\mathbf{G}_k = G(\boldsymbol{s}_k, \delta_k)$ is the input matrix at the k-th step, $\mathbf{H}_k = \mathbf{H}(\boldsymbol{s}_k, \delta_{\mathcal{W}_k})$, $\Delta\boldsymbol{a}_k = \boldsymbol{a}_k - \boldsymbol{a}_{k-1}$ and $\Delta\delta_{\mathcal{W}_k} = \delta_{\mathcal{W}_k} - \delta_{\mathcal{W}_{k-1}}$ represent the increments of control input and uncertain dynamics trigger from the step $k-1$ to the step k, respectively.

Note that the time-varying input matrix $\mathbf{G}_k$ should be approximated by online estimation strategies to reflect the real dynamics. We denote its approximation as $\widehat{\mathbf{G}}_k$, then the future states can be propagated as

$$\hat{\boldsymbol{s}}_{k+1} = \hat{\boldsymbol{s}}_k + \hat{\mathbf{G}}_k \cdot \Delta\boldsymbol{a}_k + h_k \cdot \Delta\delta_{\mathcal{W}_k} + \Delta d_k \tag{11}$$

where $\hat{\boldsymbol{s}}$ denotes the estimated state, $\hat{\mathbf{G}}_k = \hat{G}(\boldsymbol{s}_k) + \Delta\hat{G}(\boldsymbol{s}_k)\delta_{\mathcal{W}}$, and $h_k = \frac{\partial \mathbf{H}_k}{\partial \delta_{\mathcal{W}_k}}$. We call Eq. (11) an incremental model.

Remark 1: The incremental model results in a smaller perturbation being suppressed before and after an undesirable condition compared to the conventional model, which would be beneficial for improving the system's adaptability under uncertain conditions.

The theoretical analysis supporting this remark is presented below. Specifically, the approximation of $\hat{\boldsymbol{s}}_{k+1}$ in DHP is based on the equation below (we refer to it as the conventional learning model for comparison with the incremental model).

$$\begin{aligned}\hat{\boldsymbol{s}}_{k+1} &= \hat{f}(\boldsymbol{s}_k) + \Delta\hat{f}(\boldsymbol{s}_k)\delta_{\mathcal{W}} \\ &\quad + \left(\hat{G}(\boldsymbol{s}(t)) + \Delta\hat{G}(\boldsymbol{s}(t))\delta_{\mathcal{W}}\right)\boldsymbol{a}_k + d_k\end{aligned} \tag{12}$$

Then we consider the following cases: (1) $\delta_{\mathcal{W}} = 0$; (2) the transient condition with $\delta_{\mathcal{W}}$ changes from 0 to 1 (denoted as $\delta_{\mathcal{W}} = 0 \rightarrow 1$); (3) $\delta_{\mathcal{W}} = 1$. Based on these cases, Eq. (12) can be expressed as

$$\hat{\boldsymbol{s}}_{k+1} = \begin{cases} \hat{f}(\boldsymbol{s}_k) + \hat{G}(\boldsymbol{s}_k)\,\boldsymbol{a}_k + d_k, & \text{if } \delta_{\mathcal{W}} = 0 \\ \hat{f}(\boldsymbol{s}_k) + \Delta\hat{f}(\boldsymbol{s}_k)\psi + \hat{G}(\boldsymbol{s}_k)\,\boldsymbol{a}_k & \\ \quad + \Delta\hat{G}(\boldsymbol{s}_k)\,\psi\boldsymbol{a}_k + d_k, & \text{if } \delta_{\mathcal{W}} = 0 \rightarrow 1 \\ \hat{f}(\boldsymbol{s}_k) + \Delta\hat{f}(\boldsymbol{s}_k) + \hat{G}(\boldsymbol{s}_k)\,\boldsymbol{a}_k & \\ \quad + \Delta\hat{G}(\boldsymbol{s}_k)\,\boldsymbol{a}_k + d_k, & \text{if } \delta_{\mathcal{W}} = 1 \end{cases} \tag{13}$$

where ψ is a positive unknown parameter within (0,1).

Different from the conventional model approximation in (13), the learning in the incremental framework has the following representation:

$$\hat{\boldsymbol{s}}_{k+1} = \begin{cases} \hat{\boldsymbol{s}}_k + \hat{G}(\boldsymbol{s}_k)\,\Delta\boldsymbol{a}_k + \Delta d_k, & \text{if } \delta_{\mathcal{W}} = 0 \\ \hat{\boldsymbol{s}}_k + \hat{G}(\boldsymbol{s}_k)\,\Delta\boldsymbol{a}_k + \Delta\hat{G}(\boldsymbol{s}_k)\,\Delta\boldsymbol{a}_k & \\ \quad + h_k \cdot \Delta\delta_{\mathcal{W}_k} + \Delta d_k, & \text{if } \delta_{\mathcal{W}} = 0 \rightarrow 1 \\ \hat{s}_k + \hat{G}(\boldsymbol{s}_k)\,\Delta\boldsymbol{a}_k + \Delta\hat{G}(\boldsymbol{s}_k)\,\Delta\boldsymbol{a}_k & \\ \quad + \Delta d_k, & \text{if } \delta_{\mathcal{W}} = 1 \end{cases} \tag{14}$$

By comparing (13) with (14), it can be seen that: (1) For $\delta_{\mathcal{W}} = 0$, the differences of perturbation between the two methods are d_k and Δd_k. There are usually periods of time where $d_k > \Delta d_k$ and $\boldsymbol{a}_k > \Delta\boldsymbol{a}_k$, meaning that the perturbation in the incremental model is reduced and its magnitude is smaller than that in the conventional model. (2) At the instant when $\delta_{\mathcal{W}} = 0 \rightarrow 1$, the transient perturbation in both models are similar. (3) For $\delta_{\mathcal{W}} = 1$, the term $\Delta\hat{G}(\boldsymbol{s}_k)\boldsymbol{a}_k + d_k$ in Eq. (13) and $\Delta\hat{G}(\boldsymbol{s}_k)\Delta\boldsymbol{a}_k + \Delta d_k$ in Eq. (14) can be regarded as the perturbation in each case. In a small step, there usually exist periods of time where $d_k > \Delta d_k$ and $\boldsymbol{a}_k > \Delta\boldsymbol{a}_k$. Again, this means that the perturbation in the incremental model is smaller than that in the conventional model. To conclude, except for the instant $\delta_{\mathcal{W}} = 0 \rightarrow 1$, the incremental model leads to smaller perturbation than the conventional model.

It is noted that another key distinction between the incremental and conventional models is whether the whole system (both $\bar{f}$ and $\bar{G}$) is used for dynamics learning in real time. As can be seen, the incremental mode does not require the information of $\bar{f}$ and only needs to approximate partial system information ($\hat{\mathbf{G}}_k$). That renders a simple structure and reduces learning complexity.

2) Learning Strategy:

Based on the incremental model learnt from (11) and (14), we are ready to develop our IDHP. We employ a critic-actor structure in the design. Particularly, the critic network is to approximate the derivatives of the state-value function with respect to the system states, and the actor network is to generate the optimal pitch control policy for the FOWT. The critic-actor network weights are updated through gradient descent schemes in a back-propagation manner. The specific learning strategy is composed of the critic network update law, the actor network update law, the input matrix learning process for the incremental model, and the target critic network integration, which are introduced in detail in the following.

(a) Critic Network

The critic network approximates the derivative of a so-called state-value function $\mathcal{Q}(\boldsymbol{s}_k)$ with respect to the system state vector, which can be expressed as

$$\lambda(\boldsymbol{s}_k) = \frac{\partial\mathcal{Q}(\boldsymbol{s}_k)}{\partial\boldsymbol{s}_k} \tag{14}$$

In this equation, $\lambda(\boldsymbol{s}_k)$ is the output of the critic network, and $\mathcal{Q}(\boldsymbol{s}_k)$ is a cumulative summation of future costs accounting from the current state $\boldsymbol{s}_k$:

$$\mathcal{Q}(\boldsymbol{s}_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r_i \tag{15}$$

and here $i = k, k+1, ..., \infty$, r_i is the one-step cost function at the i-th step in a discrete form, which is defined as

$$r_k = (\boldsymbol{s}_k - \boldsymbol{s}_k^*)^T Q_w (\boldsymbol{s}_k - \boldsymbol{s}_k^*) + \boldsymbol{a}_k^T R_w \boldsymbol{a}_k \tag{16}$$

where $\boldsymbol{s}_k^*$ is the desired state at the k-th step.

The weights of the critic network are updated by minimizing the following mean squared error loss function

$$\mathbf{E}_k^c = \frac{1}{2}\mathbf{e}_k^{c\,T}\mathbf{e}_k^c \tag{17}$$

Here the error $\mathbf{e}_k^c$ is a partial derivative of Temporal Difference (TD) error with respect to the state:

$$\begin{aligned} \mathbf{e}_k^c &= \frac{\partial[\widehat{\mathcal{Q}}(\boldsymbol{s}_{k-1}) - r_{k-1} - \gamma\widehat{\mathcal{Q}}(\boldsymbol{s}_k)]}{\partial(\boldsymbol{s}_{k-1})} \\ &= \widehat{\lambda}(\boldsymbol{s}_{k-1}) - \frac{\partial r_{k-1}}{\partial\boldsymbol{s}_{k-1}} - \gamma\cdot\widehat{\lambda}(\boldsymbol{s}_k)\cdot\frac{\partial\boldsymbol{s}_k}{\partial\boldsymbol{s}_{k-1}} \end{aligned} \tag{18}$$

where $\widehat{\mathcal{Q}}(\boldsymbol{s}_{k-1}) - r_{k-1} - \gamma\widehat{\mathcal{Q}}(\boldsymbol{s}_k)$ is called the TD error, and $\widehat{\mathcal{Q}}(\cdot)$ and $\widehat{\lambda}(\cdot)$ represent the estimations of the state-value function and critic output, respectively.

According to the incremental model shown in Eq. (11), the term $\frac{\partial\boldsymbol{s}_k}{\partial\boldsymbol{s}_{k-1}}$ can be approximated as

$$\frac{\partial\boldsymbol{s}_k}{\partial\boldsymbol{s}_{k-1}} = \hat{\mathbf{G}}_{k-1}\Delta\boldsymbol{a}_{k-1} \tag{19}$$

Substituting this result back into Eq. (18), one has

$$\mathbf{e}_k^c = \widehat{\lambda}(\boldsymbol{s}_{k-1}) - \frac{\partial r_{k-1}}{\partial\boldsymbol{s}_{k-1}} - \gamma\cdot\widehat{\lambda}(\boldsymbol{s}_k)\cdot\hat{\mathbf{G}}_{k-1}\Delta\boldsymbol{a}_{k-1} \tag{20}$$

Then, the weights of critic network $\mathbf{w^c}$ can be updated by the gradient-descent method

$$\mathbf{w}_{k+1}^c = \mathbf{w}_k^c - \eta_c\cdot\frac{\partial\mathbf{E}_k^c}{\partial\mathbf{w}_k^c} \tag{21}$$

where $\eta_c > 0$ is the learning rate for the critic network, and

$$\frac{\partial\mathbf{E}_k^c}{\partial\mathbf{w}_k^c} = \frac{\partial\mathbf{E}_k^c}{\partial\widehat{\lambda}(\boldsymbol{s}_{k-1})}\cdot\frac{\partial\widehat{\lambda}(\boldsymbol{s}_{k-1})}{\partial\mathbf{w}_k^c} = \mathbf{e}_k^c\cdot\frac{\partial\widehat{\lambda}(\boldsymbol{s}_{k-1})}{\partial\mathbf{w}_k^c} \tag{22}$$

(b) Actor Network

The actor network aims to approximate the optimal control policy $\boldsymbol{a}_k^*$ by minimizing the state-value function $\mathcal{Q}(\boldsymbol{s}_k)$, following that

$$\boldsymbol{a}_k^* = \arg\min_{a_k}\mathcal{Q}(\boldsymbol{s}_k) = \arg\min_{a_k}[r_k + \gamma\mathcal{Q}(\boldsymbol{s}_{k+1})] \tag{23}$$

Accordingly, the loss function for updating the actor network is given as

$$\mathbf{E}_k^a = \mathcal{Q}(\boldsymbol{s}_k) = r_k + \gamma \mathcal{Q}(\boldsymbol{s}_{k+1}) \tag{24}$$

Then, the weights of the actor network are updated by the gradient-descent method, which can be described as

$$\mathbf{w}_{k+1}^a = \mathbf{w}_k^a - \eta_a \cdot \frac{\partial \mathbf{E}_k^a}{\partial \mathbf{w}_k^a} \tag{25}$$

where $\eta_a > 0$ is the learning rate for the actor network. Based on Eqs. (24) and (14), we have

$$\begin{aligned} \frac{\partial \mathbf{E}_k^a}{\partial \mathbf{w}_k^a} &= \frac{\partial \mathbf{E}_k^a}{\partial \boldsymbol{a}_k} \cdot \frac{\partial \boldsymbol{a}_k}{\partial \mathbf{w}_k^a} \\ &= [\frac{\partial r_k}{\partial \boldsymbol{a}_k} + \gamma \cdot \lambda(\boldsymbol{s}_{k+1}) \cdot \frac{\partial \boldsymbol{s}_{k+1}}{\partial \boldsymbol{a}_k}] \cdot \frac{\partial \boldsymbol{a}_k}{\partial \mathbf{w}_k^a} \end{aligned} \tag{26}$$

where the term $\lambda(\boldsymbol{s}_{k+1})$ can be approximated by the estimation of the critic output $\widehat{\lambda}(\widehat{\boldsymbol{s}}_{k+1})$ under the estimated state $\widehat{\boldsymbol{s}}_{k+1}$ from the incremental model in Eq. (11). Therefore, Eq. (26) can be rewritten as

$$\frac{\partial \mathbf{E}_k^a}{\partial \mathbf{w}_k^a} = [\frac{\partial r_k}{\partial \boldsymbol{a}_k} + \gamma \cdot \widehat{\lambda}(\widehat{\boldsymbol{s}}_{k+1}) \cdot \frac{\partial \widehat{\boldsymbol{s}}_{k+1}}{\partial \boldsymbol{a}_k}] \cdot \frac{\partial \boldsymbol{a}_k}{\partial \mathbf{w}_k^a} \tag{27}$$

Notably, similar to the critic network, the incremental model is utilized to approximate the term $\frac{\partial \widehat{\boldsymbol{s}}_{k+1}}{\partial \boldsymbol{a}_k}$ in the above updating law for actor-network weights. Specifically,

$$\frac{\partial \widehat{\boldsymbol{s}}_{k+1}}{\partial \boldsymbol{a}_k} \approx \widehat{\mathbf{G}}_k \tag{28}$$

Thus, one has

$$\frac{\partial \mathbf{E}_k^a}{\partial \mathbf{w}_k^a} = [\frac{\partial r_k}{\partial \boldsymbol{a}_k} + \gamma \cdot \widehat{\lambda}(\widehat{\boldsymbol{s}}_{k+1}) \cdot \widehat{\mathbf{G}}_k] \cdot \frac{\partial \boldsymbol{a}_k}{\partial \mathbf{w}_k^a} \tag{29}$$

(c) Input Matrix Estimation

As mentioned previously, the input matrix should be learnt and updated online for the incremental model such that the real wind turbine dynamics under uncertain conditions can be reflected to the control scheme on time. In this work, the recursive least square (RLS) estimation process is employed to update $\widehat{\mathbf{G}}_k$. The updating law satisfies

$$\widehat{\mathbf{G}}_k^T = \widehat{\mathbf{G}}_{k-1}^T + \frac{\mathrm{P}_{k-1}\,\Delta \boldsymbol{a}_k}{\tau + \Delta \boldsymbol{a}_k^T\,\mathrm{P}_{k-1}\,\Delta \boldsymbol{a}_k}(\Delta \boldsymbol{s}_{k+1}^T - \Delta \widehat{\boldsymbol{s}}_{k+1}^T) \tag{30}$$

$$\mathrm{P}_k = \frac{1}{\tau}\left(\mathrm{P}_{k-1} - \frac{\mathrm{P}_{k-1}\,\Delta \boldsymbol{a}_k \Delta \boldsymbol{a}_k^T\,\mathrm{P}_{k-1}}{\tau + \Delta \boldsymbol{a}_k^T\,\mathrm{P}_{k-1}\,\Delta \boldsymbol{a}_k}\right) \tag{31}$$

where $\Delta \boldsymbol{s}_{k+1} = \boldsymbol{s}_{k+1} - \boldsymbol{s}_k$, $\Delta \widehat{\boldsymbol{s}}_{k+1} = \widehat{\boldsymbol{s}}_{k+1} - \widehat{\boldsymbol{s}}_k$, $\tau \in [0, 1]$ is the forgetting factor, and $\mathrm{P}_k \in \mathcal{R}^{(m \times m}$ is the estimation covariance matrix.

(d) Target Critic Network

It has been reported that the TD error-based control policy evaluation might diverge and the learning instability is inevitable due to the nonlinearity of $\widehat{\mathcal{Q}}(\boldsymbol{s}_k)$ [38]. To alleviate this potential problem, a target critic network $\lambda^{'}(\boldsymbol{s}_k)$ is integrated into the learning process. It is updated slower than the critic network $\lambda(\boldsymbol{s}_k)$. The weights of the target critic network are updated by a soft replacement strategy [39], defined as

$$\mathbf{w}_{k+1}^T = \sigma \mathbf{w}_{k+1}^c + (1 - \sigma)\mathbf{w}_k^T \tag{32}$$

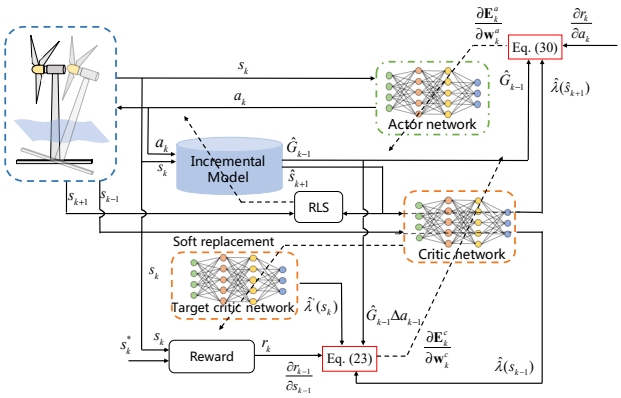

Fig. 4. The architecture of IDHP-based FOWT control strategy.

where $\mathbf{w}^T$ denote the weights of the target critic network, $\sigma \in (0, 1]$ is a user-defined scalar factor. This updating strategy illustrates that the weights of the target network are slowly tracking that of the critic network, guaranteeing the training stability and reliability [38], [40]. Accordingly, with the target critic network, the error e_k^c in Eq. (18) for updating the weights of the critic network can be rewritten as

$$\mathbf{e}_k^c = \widehat{\lambda}(\boldsymbol{s}_{k-1}) - \frac{\partial r_{k-1}}{\partial \boldsymbol{s}_{k-1}} - \gamma \cdot \widehat{\lambda}^{'}(\boldsymbol{s}_k) \cdot \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{s}_{k-1}} \tag{33}$$

where $\widehat{\lambda}^{'}(\boldsymbol{s}_k)$ is the estimation of the target critic output.

Based on all these designs, the whole architecture of our IDHP-based pitch controller for FOWT is depicted in Fig. 4. Compared with the conventional DHP method, as shown in Fig. 3, our method only needs to learn partial system information (the input matrix in the incremental model) instead of updating the whole system dynamics. In addition, we employ the target critic to enhance our control scheme's learning stability and reliability. Moreover, our control method collects real-time data to reflect the influence of real dynamics, disturbances, and stochastic wind-wave environments, providing strong robustness and adaptability to the load reduction tasks of FOWTs.

IV. Numerical Simulations

In this section, numerical simulations are conducted to validate the performance and advantages of the proposed method with the FAST simulator. The FOWT simulator we employed is the NERL-5MW-Barge type, with the rated power being 5MW and the floating platform being the barge category. This turbine is a three-bladed upwind turbine, with the hub height being 63 m, the tower height being 87.6m and the rotor diameter being 126 m. The discount factor γ, forgetting factor τ, and scalar factor σ are set as $\gamma = 0.9$, $\tau = 0.9$, $\sigma = 0.1$, respectively. The weight matrices Q_w and R_w are set as $Q_w = diag[10^{-2}, 10^4, 10^5, 10^2, 10^2, 10^2]$ and $R_w = diag[10^{-2}, 10^{-2}, 10^{-2}]$. The rated generator speed is selected as $w_g^{rated}(t) = 122.9$rad/s [41]. The wind profile generated by the NREL software TurbSim [42] is employed for verification. The PI control gains are chosen as $K_P = 0.015$ and $K_I = 0.005$. The total simulation time is set as $600s$.

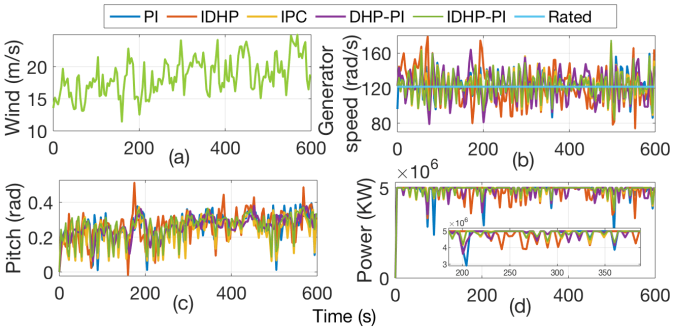

Fig. 5. Simulation results under different controllers. (a) Wind speed profile. (b) Generator speeds. (c) Blade pitch angles. (d) Power outputs.

A. Simulation Results Under Different Controllers

In this subsection, five controllers are carried out for comparison. They are

1) PI: The PI-based CPC, which serves as the baseline controller for the FOWT simulator employed in simulations.
2) IDHP: The IDHP-based CPC, in which (1) only CPC is employed to achieve the control task without combining with IPC and (2) IDHP employs the incremental model.
3) IPC: A MPC-based IPC, in which only the IPC is employed to achieve the control task.
4) DHP-PI: This control algorithm integrates the DHP-based IPC with the PI-based CPC, in which DHP employs the conventional model.
5) IDHP-PI: The proposed method integrates the IDHP-based IPC with the PI-based CPC, in which IDHP employs the incremental model.

As described in Introduction, the pitch control of wind turbines can be achieved by CPC, IPC, or by combining CPC with IPC. Note that the "IDHP" method means only the IDHP-based CPC is employed to achieve the control task, while the "IDHP-PI" method denotes the algorithm combined the IDHP-based CPC and the PI-based IPC.

The generator speed, pitch angle, and power output under these controllers are presented in Fig. 5. The wind profile in simulations is shown in Fig. 5(a), with a mean speed of 18m/s. It can be observed from Fig. 5(b) that all the controllers can ensure the generator speed track the rated value consistently. The smallest deviations are achieved by the proposed IDHP-PI controller, while the IDHP-based CPC creates larger errors. Fig. 5(c) indicates that the IDHP-PI strategy performs better since it leads to the smoothest pitch angles than other controllers. In contrast, remarkable vibrations exist in the blade pitch angles under the IDHP-based CPC controller, which is undesirable in practice. As shown in Fig. 5(d), better power outputs are produced by IDHP-PI, IPC, and DHP-PI compared to the PI controller.

In addition, a quantitative analysis of these simulation results is presented. Specifically, the mean square error (MSE) [43] is employed for the characterization of the power regulation performance, which is defined as the averaged squared errors:

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(y_i - y_i^*)^2 \tag{34}$$

where y_i denotes the measured data, y_i^* is the reference data, and M is the total number of data. In this work, y_i^* is the rated power value – 5MW, y_i is the power generated by different controllers. Note that the smaller the MSE value, the smaller the power fluctuation, and the stronger the power regulation ability of the corresponding controller.

The comparison of MSE values of power production under different controllers is provided in Table I. It can be seen that the IDHP-PI method proposed in this paper leads to the smallest MSE among all the controllers, followed by IPC, DHP-PI, and IDHP, while the PI controller results in the largest MSE.

TABLE I
MSE VALUES OF THE POWER PRODUCTION UNDER DIFFERENT CONTROLLERS.

Methods	**PI**	**IDHP**	**IPC**	**DHP-PI**	**IDHP-PI**
MSE $(*10^{11})$	2.18	1.95	1.04	1.17	0.94

Fig. 6 displays the structural responses of the FOWT. It can be observed that the magnitudes of structural responses in the out-of-plane direction (including the blade out-of-plane deflection, tower fore-aft displacement, and platform pitch displacement) under the proposed IDHP-PI method are significantly reduced compared to the PI and IDHP methods. These results illustrate that the proposed method can successfully achieve the objective formulated in Section. II-B to mitigate loads. As for the structural responses in the in-plane direction, the IDHP-PI, DHP-PI, and IPC controllers are comparable and perform better than the PI and IDHP controller in several periods. In addition, the bending moments for the FOWT are depicted in Fig. 7. The blade out-of-plane bending moment, tower fore-aft bending moment, and platform fore-aft bending moment are reduced with IDHP-PI, showing superiority over IDHP and PI controllers in the out-of-plane direction.

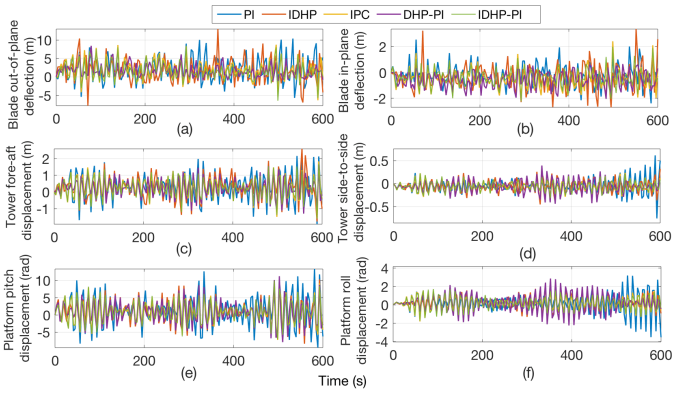

Fig. 6. Structural responses of the FOWT under different controllers. (a) Blade out-of-plane deflection. (b) Blade in-plane deflection. (c) Tower fore-aft displacement. (d) Tower side-to-side displacement. (e) Platform pitch displacement. (f) Platform roll displacements.

To make the simulation results shown in Figs. 6 and 7 more noticeable, we carry out quantitative analysis further to clearly explain the difference between these different controllers. Specifically, the fatigue damage equivalent load (DEL) [44] is employed for the characterization of the bending moments.

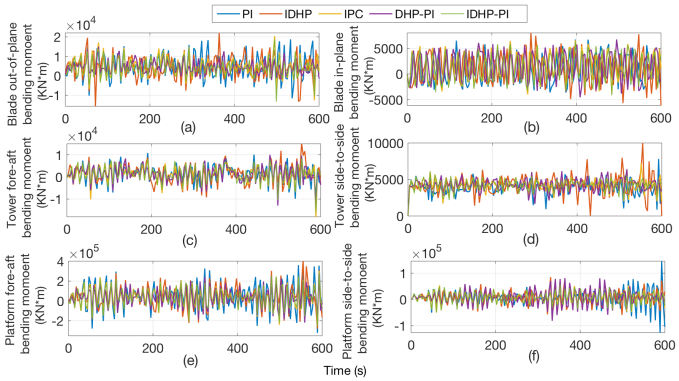

Fig. 7. Bending moments of the FOWT under different controllers. (a) Blade out-of-plane. (b) Blade in-plane. (c) Tower fore-aft. (d) Tower side-to-side. (e) Platform fore-aft. (f) Platform side-to-side.

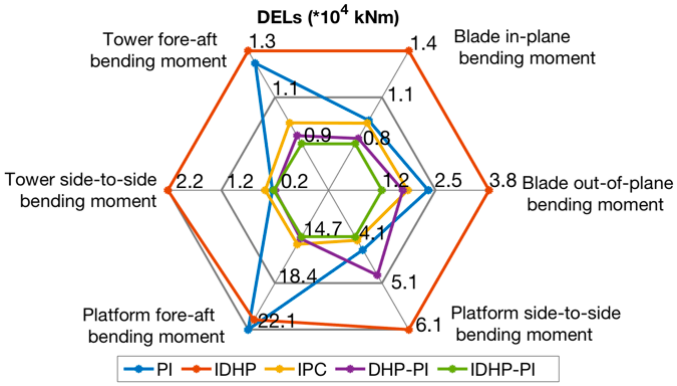

Fig. 8. Spider plot of the fatigue DELs of the FOWT bending moments.

The DEL D_{eq} is defined as the fatigue load variation corresponding to a number of equivalent load cycles n_{eq} that create the same damage level at the amplitude moment in one load cycle D_i with respect to the number of cycles n_i. The DEL is solved by the rain-flow counting technique, which has the following form [45].

$$D_{eq} = \left(\frac{\sum n_i D_i^{m_{eq}}}{n_{eq}}\right)^{\frac{1}{m_{eq}}} \tag{35}$$

where m_{eq} is the Wöhler exponent. In this paper, we select $m_{eq} = 10$ for blades, $m_{eq} = 5$ for tower, and $m_{eq} = 3$ for platform [44], [45].

As a result, the fatigue DELs corresponding to the bending moments in Fig. 7 are shown in Fig. 8, which demonstrate the differences between these methods. It can be seen that the DELs values of the proposed IDHP-PI method are the lowest for all the bending moments in both the out-of-plane direction and in-plane direction, followed by the DHP-PI and IPC methods. This demonstrates that the proposed IDHP-PI method based on the incremental model can reduce fatigue loads compared to the DHP-PI method based on the conventional learning model. Although the DHP-PI has slightly higher DELs values than the IPC method in the platform side-to-side bending moment, it has lower DELs values in all the other bending moments. The PI controller can reduce the fatigue loads in the blade in-plane direction, tower side-to-side bending moment, and platform side-to-side bending moment, but it does not perform well in the tower fore-aft bending moment and platform fore-aft bending moment. The PI controller has a higher DELs value in the platform fore-aft bending moment than the IDHP-based CPC method, but the IDHP-based CPC method has the highest DELs values in other bending moments.

The variations of the updated weights in the critic-actor network are presented in Fig. 9. In which $\mathbf{k^c}$ and $\mathbf{w^c}$ represent the weights of the critic network from the input layer to the hidden layer and from the hidden layer to the output layer, respectively, and $\mathbf{k^a}$ and $\mathbf{w^a}$ denote that of the actor network, respectively. It is noted that these time-varying network weights are generated to adapt to the real dynamics and meet the optimal control goals for both power regulation and load mitigation.

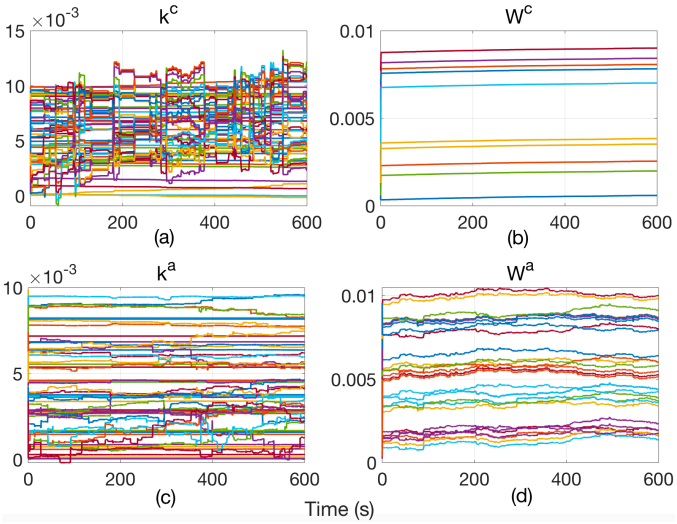

Fig. 9. Simulation results of the weights in the critic-actor network. (a) $\mathbf{k^c}$. (b) $\mathbf{w^c}$. (c) $\mathbf{k^a}$. (d) $\mathbf{w^a}$.

In summary, the proposed RL-based FOWT control method has the capability to improve power quality and alleviate structural loads simultaneously, showing superior performance than other controllers.

B. Simulation Results Under Different Wind-Wave Conditions

To verify the robustness and adaptability of the proposed FOWT controller further, we employ different wave directions, wave heights, and extreme wind speeds in simulations. The settings are listed in Table II. Case 1 is regarded as the base case for comparison. Cases 2-4 refer to various wave directions. The effect of different wave heights on the fatigue loads is conducted in Cases 5-8. Furthermore, two extreme wind speeds are considered in Cases 9-10 to validate the performance of our method under extreme conditions.

The fatigue DELs under different controllers considering different cases are provided in Fig. 10. It can be concluded that:

(1) In comparison with the DHP-PI method, significant reductions of fatigue DELs in the out-of-plane direction can be

TABLE II
DIFFERENT WIND-WAVE CASES

Cases	Wind speed	Wave direction	Wave height
1	18 m/s	60^o	5 m
2	18 m/s	0^o	5 m
3	18 m/s	30^o	5 m
4	18 m/s	90^o	5 m
5	18 m/s	60^o	3 m
6	18 m/s	60^o	4 m
7	18 m/s	60^o	6 m
8	18 m/s	60^o	7 m
9	22 m/s	60^o	5 m
10	24 m/s	60^o	5 m

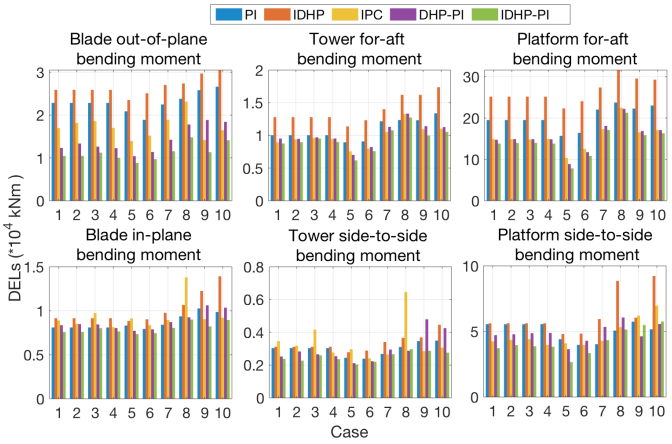

Fig. 10. The fatigue DELs of the FOWT bending moments in different cases under different controllers.

achieved by the proposed IDHP-PI controller in all the cases. In addition, the DELs for the in-plane direction's bending moments also exhibit noticeable reductions in all cases under the IDHP-PI method.

(2) In most cases, the proposed IDHP-PI performs best for reducing the fatigue loads under different wind-wave cases. Then, DHP-PI and IPC methods are comparable, followed by the PI controller, while pure IDHP performs the worst.

(3) From Cases 1-4, it can be observed that when the significant wave height is fixed, the fatigue DELs change slightly under different wave directions.

(4) As for Cases 5-8 with fixed wave directions, there exist notable growing trends of the DEL values under both controllers as the significant wave height increases.

(5) The fatigue DELs under both controllers become larger with the increase of wind speeds, as can be seen in Cases 9-10. It is noteworthy that our RL-based method can still mitigate DELs under extreme wind conditions compared with other controllers.

Table III shows the MSE values for the power outputs of various controllers under different wind-wave scenarios, allowing for a quantitative comparison of power regulation capabilities. It is evident that the smallest MSE values in each case correspond to our proposed method, demonstrating its ability to reduce loads and regulate power simultaneously, even under varying wind-wave conditions. The DHP-PI strategy and the IPC algorithm both show smaller MSE values for power output compared to the IDHP-based CPC method and the PI controller. The PI controller has the largest MSE values among all the methods, revealing significant power fluctuations in certain cases.

TABLE III
MSE VALUES OF THE POWER PRODUCTION UNDER DIFFERENT CONTROLLERS CONSIDERING DIFFERENT WIND-WAVE CONDITIONS. $(*10^{11})$

Cases	PI	IDHP	IPC	DHP-PI	IDHP-PI
1	2.86	1.92	1.12	1.32	0.86
2	2.17	1.87	1.25	1.19	0.92
3	1.86	1.79	1.23	1.24	0.95
4	2.64	1.92	1.21	1.19	0.87
5	2.69	2.01	1.15	1.32	.1.03
6	2.64	1.86	1.04	1.21	0.84
7	2.73	1.93	1.02	1.12	0.79
8	2.18	1.76	1.15	1.31	0.85
9	2.05	2.74	1.24	1.29	1.16
10	2.13	2.83	1.17	1.28	0.96

C. Simulation Results Under Loads In Out-of-Plane And In-Plane Direction

As described in Section II-A, the motion in the out-of-plane direction of the FOWT is subjected to the major loads [1], thus, the above simulation results focus on suppressing turbine motions/loads in the out-of-plane direction via pitch control. In order to make a fair comparison, other loads in the in-plane direction including the platform roll, tower side-to-side bending moment, and edge-wise bending moments of blade 1, blade 2, and blade 3 are also considered.

Accordingly, the DELs corresponding to the bending moments of fatigue loads in both out-of-plane and in-plane directions are depicted in Fig. 11. Compared with the Fig. 8, we can conclude that all DELs values shown in Fig. 11 are smaller than the DELs values in Fig. 8, which only considering the fatigue loads in the out-of-plane direction. Another observation from Fig. 11 is that the proposed IDHP-PI still has the smallest DELs values of fatigue loads among these different controllers, followed by the DHP-PI, IPC algorithms, and PI controller. While the IDHP-based CPC performs worst in reducing most of the fatigue loads.

To further illustrate the effectiveness and robustness of these controllers, the DELs of the bending moments considering fatigue loads in both out-of-plane and in-plane directions in different wind-wave cases, introduced in Table II, are shown in Fig. 12. It clearly indicates that the proposed IDHP-PI method has the smallest DELs values in all cases, which demonstrates that the proposed method can significantly reduce fatigue loads. In addition, compared with the results under Fig. 10, it can be observed that the DELs values in Fig. 12 are smaller in all bending moments. It illustrates that when considering more fatigue loads including the loads in both out-of-plane and in-plane directions, our controllers still have the best performance among all controllers.

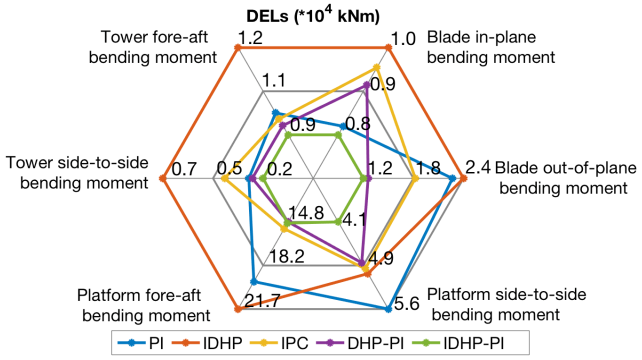

Fig. 11. Spider plot of the DELs of the FOWT bending moments considering fatigue loads in both out-of-plane and in-plane directions.

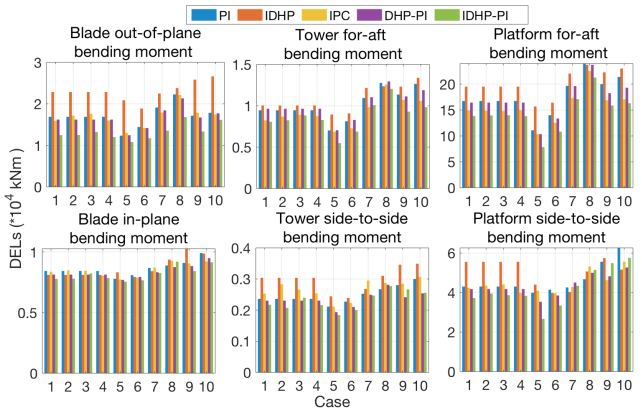

Fig. 12. The DELs of the FOWT bending moments in different cases under different methods considering fatigue loads in both out-of-plane and in-plane directions.

D. Simulation Results Under All Essential Operational Regions.

It is important to note that in our method, the training set employed actually covers all the essential operational ranges, including Regions II and III, and also the transient Region between them [46], for running simulations on the high-fidelity FAST simulator. The rated wind speed for the utilized NREL-5MW FAST simulator is $12.1m/s$ [47]. As a result, we chose a wind profile with a mean wind speed of $12m/s$ for the simulation. The wind profile is displayed in Fig. 13, which includes inflow wind speeds both below and above the rated wind speeds for simulating under Region II, transient Region, and Region III. Simulation results of the rotor speed, generator power, and some structural responses using the proposed IDHP-PI method are presented in Fig. 14. These findings demonstrate that the proposed IDHP-PI-based FOWT controller is suitable not only for Region II but also for the transient Region and Region III.

E. Simulation Results Under Full Loads Set to Reduce Fatigue and Extreme Loads

In this subsection, the full loads set is considered to investigate the impact of different controllers on both fatigue loads and extreme loads.

(1) Fatigue loads analysis

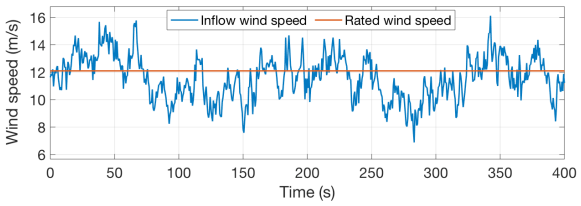

Fig. 13. Wind profile for the simulation with all operation regions.

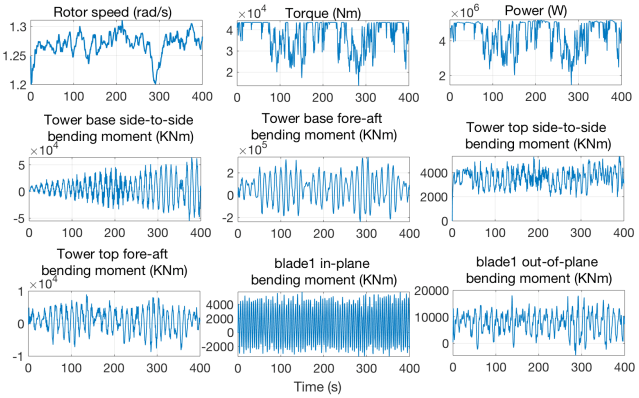

Fig. 14. Simulation results under the proposed IDHP-PI algorithm with the mean wind speed of $12m/s$.

For the fatigue loads analysis, a full set of loads is examined under different controllers. Specifically, a ten-minute simulation with varying wind speeds and turbulence is conducted to test fatigue loads. The rain-flow counting technique is employed to calculate the Damage Equivalent Loads (DEL) [44] for characterizing fatigue loads. The Wöhler exponents are chosen as $m_{eq} = 10$ for blades, $m_{eq} = 5$ for the tower, and $m_{eq} = 3$ for the platform [44].

The ratio between the DELs of fatigue loads under different controllers and the DELs of uncontrolled fatigue loads is depicted in Fig. 15. Firstly, it can be observed that all the fatigue loads are either reduced or maintained unchanged under these methods. The proposed IDHP-PI controller can decrease the tower fore-aft displacements and platform rotational roll displacements by up to 11% and up to 7% for tower side-to-side displacements and tower base side-to-side shear force. A reduction of 5% to 6% for the tower side-to-side displacement is observed under other controllers. The fatigue loads in the platform pitch and yaw displacements, as well as the tower base fore-aft, have experienced a noticeable decrease under all these controllers.

(2) Extreme loads analysis

As the design of large wind turbine blades is increasingly affected by extreme loads rather than fatigue loads [48], it is crucial to investigate the impact of these controllers on extreme loads. A 10-minute simulation with extreme 50-year wind speed and turbulence has been conducted on the FOWT during power production for the extreme-load analysis. The extreme wind speed, defined by a power law with a shear exponent of 0.2, is set at $25m/s$ for all the simulations. The extreme turbulence, defined as the standard deviation of the simulated wind time series, is also included. This extreme

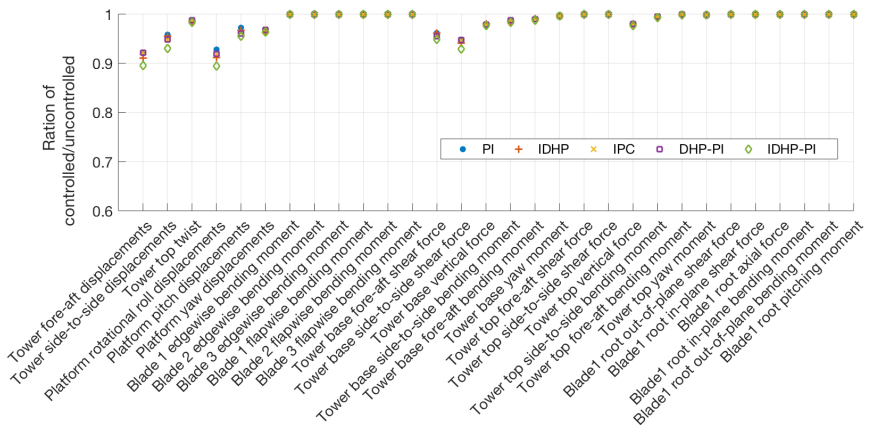

Fig. 15. The ratio between the DELs of fatigue loads under different controllers to the DELs of uncontrolled fatigue loads.

condition is generated by TurbSim [42].

The ratio between the maximum of the extreme loads to the uncontrolled extreme loads is illustrated in Fig. 16. The large differences can be observed in the tower side-to-side displacements and platform yaw displacements. All these methods can decrease the extreme loads in the tower side-to-side displacement, which is reduced by 21 % under the proposed IDHP-PI controller. While the DHP-PI, IPC, IDHP-based CPC, and PI achieve a reduction of 19%, 16%, 15%, and 13%, respectively. A prominent behavior is displayed in the platform yaw displacements, which get reduced by 80% by the IDHP-based CPC method. A reduction in platform pitch displacements of 8 % – 11% is observed for all controllers. Another bigger reduction is achieved in the bending moment at the interface between the tower and the nacelle, namely the tower top fore-aft bending moment, which is decreased by approximately 7 % - 10%.

V. Conclusion

A data-driven, model-free control strategy was proposed for floating offshore wind turbines (FOWTs) to achieve power regulation and load mitigation via reinforcement learning. On the one hand, to maintain power regulation, a PI controller was employed as the collective pitch control. On the other hand, to alleviate the structural loads, a novel IDHP control algorithm was designed to perform as the individual pitch control. Our design differs from existing IDHP methods in that only the input matrix was required to be dynamically updated to learn the optimal control policy. Theoretical analysis indicates that the incremental technique leads to smaller residuals being suppressed in a higher sampling frequency than approximating the entire model information. The proposed method has solid adaptability against uncertainties, modeling errors, and stochastic environmental conditions and can lead to better performance than existing solutions in both power regulation and load reduction. In the future, we will investigate faster incremental model identification techniques to improve computational efficiency further. We will also investigate novel physics-informed reinforcement learning methods to achieve better overall control performance for wind turbine control problems by leveraging AI and physics information simultaneously.

References

[1] S. Sarkar, B. Fitzgerald, and B. Basu, "Individual blade pitch control of floating offshore wind turbines for load mitigation and power regulation," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 1, pp. 305–315, 2020.

[2] H. Namik and K. Stol, "Individual blade pitch control of a spar-buoy floating wind turbine," *IEEE transactions on control systems technology*, vol. 22, no. 1, pp. 214–223, 2013.

[3] C. Zhang and F. Plestan, "Individual/collective blade pitch control of floating wind turbine based on adaptive second order sliding mode," *Ocean Engineering*, vol. 228, p. 108897, 2021.

[4] J. Zhang, X. Zhao, and X. Wei, "Reinforcement learning-based structural control of floating wind turbines," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[5] M. N. Sinner and L. Y. Pao, "A comparison of individual and collective pitch model predictive controllers for wind turbines," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 1509–1514.

[6] H. Badihi, Y. Zhang, P. Pillay, and S. Rakheja, "Fault-tolerant individual pitch control for load mitigation in wind turbines with actuator faults," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 1, pp. 532–543, 2020.

[7] M. A. Lackner, "Controlling platform motions and reducing blade loads for floating wind turbines," *Wind Engineering*, vol. 33, no. 6, pp. 541–553, 2009.

[8] H. Guo, X. Lu, and T. Qiu, "Research on pitch control of floating offshore wind turbines," in *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE, 2012, pp. 2966–2970.

[9] S. Li and F. Yang, "Load optimization control of large deep-water floating wind turbines," *Journal of Mechatronics*, vol. 2, no. 3, pp. 169–175, 2014.

[10] F. Li, L. Zhou, L. Li, H. Wang, H. Guo, and Y. Liang, "Individual blade pitch control for floating wind turbine based on rbf-smc," in *2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, 2019, pp. 949–953.

[11] S. Li, Y. Han, W. Pan, S. Liu, and M. Hou, "Variable-gain higher-order sliding mode pitch control of floating offshore wind turbine," *Journal of Marine Science and Engineering*, vol. 9, no. 11, p. 1172, 2021.

[12] N. Hara, S. Tsujimoto, Y. Nihei, K. Iijima, and K. Konishi, "Experimental validation of model-based blade pitch controller design for floating wind turbines: system identification approach," *Wind energy*, vol. 20, no. 7, pp. 1187–1206, 2017.

[13] S. Raach, D. Schlipf, F. Sandner, D. Matha, and P. W. Cheng, "Nonlinear model predictive control of floating wind turbines with individual pitch control," in *2014 American control conference*. IEEE, 2014, pp. 4434–4439.

[14] P. Kou, D. Liang, J. Li, L. Gao, and Q. Ze, "Finite-control-set model predictive control for dfig wind turbines," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1004–1013, 2017.

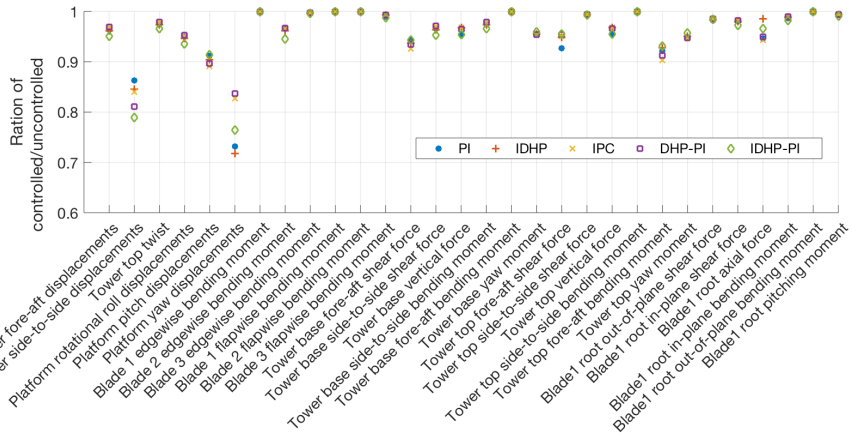

Fig. 16. The ratio of the maximum loads to the uncontrolled extreme loads.

[15] X. Liu and X. Kong, "Nonlinear model predictive control for dfig-based wind power generation," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1046–1055, 2013.

[16] T. Wakui, A. Nagamura, and R. Yokoyama, "Stabilization of power output and platform motion of a floating offshore wind turbine-generator system using model predictive control based on previewed disturbances," *Renewable Energy*, vol. 173, pp. 105–127, 2021.

[17] Y. Liu, R. M. Ferrari, and J.-W. van Wingerden, "Periodic load rejection for floating offshore wind turbines via constrained subspace predictive repetitive control," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 539–544.

[18] A. Lasheen and A. L. Elshafei, "Fuzzy predictive control of the collective pitch in large wind turbines," in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 1528–1533.

[19] S. Tang, D. Tian, X. Wu, M. Huang, and Y. Deng, "Wind turbine load reduction based on 2dof robust individual pitch control," *Renewable Energy*, vol. 183, pp. 28–40, 2022.

[20] H. Dong, J. Xie, and X. Zhao, "Wind farm control technologies: From classical control to reinforcement learning," *Progress in Energy*, 2022.

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

[22] E. Hosseini, E. Aghadavoodi, and L. M. F. Ramírez, "Improving response of wind turbines by pitch angle controller based on gain-scheduled recurrent anfis type 2 with passive reinforcement learning," *Renewable Energy*, vol. 157, pp. 897–910, 2020.

[23] M. Coquelet, L. Bricteux, M. Moens, and P. Chatelain, "A reinforcement-learning approach for individual pitch control," *Wind Energy*, 2022.

[24] J. M. Jonkman, M. L. Buhl *et al.*, *FAST user's guide*. National Renewable Energy Laboratory Golden, CO, USA, 2005, vol. 365.

[25] M. Karimi, B. Buckham, and C. Crawford, "A fully coupled frequency domain model for floating offshore wind turbines," *Journal of Ocean Engineering and Marine Energy*, vol. 5, no. 2, pp. 135–158, 2019.

[26] P. J. Werbos, "Neural networks for control and system identification," in *Proceedings of the 28th IEEE Conference on Decision and Control,*. IEEE, 1989, pp. 260–265.

[27] W. T. Miller, R. S. Sutton, and P. J. Werbos, "A menu of designs for reinforcement learning over time," 1995.

[28] P. Werbos, "Approximate dynamic programming for real-time control and neural modeling," *Handbook of intelligent control*, 1992.

[29] Y. Zhou, "Efficient online globalized dual heuristic programming with an associated dual network," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[30] Y. Yang, L. Juntao, and P. Lingling, "Multi-robot path planning based on a deep reinforcement learning dqn algorithm," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 3, pp. 177–183, 2020.

[31] Q. Wei, F. L. Lewis, D. Liu, R. Song, and H. Lin, "Discrete-time local value iteration adaptive dynamic programming: Convergence analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 6, pp. 875–891, 2016.

[32] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE computational intelligence magazine*, vol. 4, no. 2, pp. 39–47, 2009.

[33] D. Liu, Q. Wei, D. Wang, X. Yang, and H. Li, *Adaptive dynamic programming with applications in optimal control*. Springer, 2017.

[34] B. Sun and E.-J. van Kampen, "Incremental model-based heuristic dynamic programming with output feedback applied to aerospace system identification and control," in *2020 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2020, pp. 366–371.

[35] Y. Zhou, E.-J. van Kampen, and Q. P. Chu, "Incremental model based online dual heuristic programming for nonlinear adaptive control," *Control Engineering Practice*, vol. 73, pp. 13–25, 2018.

[36] X. Wang and S. Sun, "Incremental fault-tolerant control for a hybrid quad-plane uav subjected to a complete rotor loss," *Aerospace Science and Technology*, vol. 125, p. 107105, 2022.

[37] T. Han, Q. Hu, H.-S. Shin, A. Tsourdos, and M. Xin, "Incremental twisting fault tolerant control for hypersonic vehicles with partial model knowledge," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1050–1060, 2021.

[38] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio, "An actor-critic algorithm for sequence prediction," *arXiv preprint arXiv:1607.07086*, 2016.

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

[40] S. Heyer, D. Kroezen, and E.-J. Van Kampen, "Online adaptive incremental reinforcement learning flight control for a cs-25 class aircraft," in *AIAA Scitech 2020 Forum*, 2020, p. 1844.

[41] I. Poultangari, R. Shahnazi, and M. Sheikhan, "Rbf neural network based pi pitch controller for a class of 5-mw wind turbines using particle swarm optimization algorithm," *ISA transactions*, vol. 51, no. 5, pp. 641–648, 2012.

[42] B. J. Jonkman and M. L. Buhl Jr, "Turbsim user's guide," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2006.

[43] H. Marmolin, "Subjective mse measures," *IEEE transactions on systems, man, and cybernetics*, vol. 16, no. 3, pp. 486–489, 1986.

[44] Y. Liang, X. Zhao, and L. Sun, "A multi-agent reinforcement learning approach for wind farm frequency control," *IEEE Transactions on Industrial Informatics*, 2022.

[45] Z. Cheng, H. A. Madsen, W. Chai, Z. Gao, and T. Moan, "A comparison of extreme structural responses and fatigue damage of semi-submersible type floating horizontal and vertical axis wind turbines," *Renewable Energy*, vol. 108, pp. 207–219, 2017.

[46] D. Palejiya and D. Chen, "Performance improvements of switching control for wind turbines," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 526–534, 2015.

[47] M. Ghane, A. R. Nejad, M. Blanke, Z. Gao, and T. Moan, "Statistical fault diagnosis of wind turbine drivetrain applied to a 5mw floating wind turbine," in *Journal of Physics: Conference Series*, vol. 753, no. 5. IOP Publishing, 2016, p. 052017.

[48] L. O. Bernhammer, G. A. van Kuik, and R. De Breuker, "Fatigue and extreme load reduction of wind turbine components using smart rotors," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 154, pp. 84–95, 2016.