Reinforcement Learning-Based Structural Control of Floating Wind Turbines

Jincheng Zhang, Xiaowei Zhao, and Xing Wei

***Abstract*—The structural control of floating wind turbines using active tuned mass damper is investigated in this article. To our knowledge, this is for the first time that reinforcement learning-based control approach is employed to this type of application. Specifically, an adaptive dynamic programming (ADP) algorithm is used to derive the optimal control law based on the nonlinear structural dynamics, and the large-scale machine learning platform Tensorflow is employed for the design and implementation of the neural network (NN) structure. Three fully connected NNs, i.e., a plant network, a critic network, and an action network, are included in the proposed NN structure. Their training requires the gradient information flowing through the whole network, which is tackled by automatic differentiation, a popular technique for deriving the gradients of complex networks automatically. While to our knowledge, the network structures in the existing literature are rather simple and the training of the hidden layer is usually ignored. This allows their gradients to be derived analytically, which is infeasible with complex network structures. Thus, automatic differentiation greatly improves the employed ADP algorithm's ability in solving complex problems. The simulation results of structural control of floating wind turbines show that ADP controller performs very well in both normal and extreme conditions, with the standard deviation of the platform pitch displacement being reduced by around 40%. A clear advantage of ADP controllers over the H_∞ controller is observed, especially in extreme conditions. Moreover, our design considers the tradeoff between the control performance and power consumption.**

***Index Terms*—Active structural control, adaptive dynamic programming (ADP), floating wind turbine, neural networks (NNs), reinforcement learning.**

I. Introduction

As ONE of the most important clean energy resources, wind energy has been investigating extensively all over the world. Due to the installation limitations of the land-based wind turbines and, on the other hand, the high quality of the offshore wind, more and more large wind turbines are being constructed offshore [1]. There are two types of offshore wind turbines according to the foundations, the fix-bottom ones and the floating ones. The fix-bottom turbines are installed in shallow water sites while floating ones are in deep water sites away from the coast, where the fix-bottom structures become economically infeasible [1]. Thus, the modeling and control of the floating wind turbines is becoming a hot topic [1], [2].

The fatigue loads on floating wind turbines are much more severe than the fix-bottom ones, due to the platform motions caused by the significant external disturbances (i.e., strong wind and waves) [1]. Thus, their load mitigation is of great importance for reducing maintenance costs and increasing lifespan. One approach for load mitigation is to make use of the rotor thrust as the restoring force to stabilize the floating platform, which can be achieved through either turbine blade pitch control [3] (including individual pitch control [4] and collective pitch control [5], [6]) or torque control [7]. However, this approach may interfere with the nominal power generation. Another approach is to install additional control devices on the floating platform, such as tuned mass dampers (TMDs) [8] and tuned liquid dampers (TLDs), to dampen the platform/tower vibrations directly without interfering power generation. In [9], passive TMDs were investigated for the structural control of both monopile turbines and floating turbines. Further study considered the use of multiple TMDs [10] and different modeling approaches for monopile turbines [11] and floating turbines [12]. As for TLDs, they were investigated for the structural control of conventional turbines in [13]. Further studies investigated the modeling and optimal design of TLDs [14], the semiactive control approach [15], and the control of floating hydrostatic wind turbines [16].

The performance of a TMD can be further improved by adding active force control to it, which is referred as hybrid mass dampers (HMDs) [17]. The existing works on active structural control of floating wind turbines by using HMDs are rather limited. In [18], structural control of a floating barge-type wind turbine was investigated, where an HMD was positioned in the turbine nacelle to reduce the loads. A limited degree of freedom (DOF) model was constructed through the system identification procedure, then the H_∞ multivariable loop shaping controller was designed. The paper [19] further investigated the effects of both actuator dynamics and control-structure interaction on the active control of floating wind turbines. In [20], load mitigation of floating wind turbines by an HMD installed on the platform was investigated. A linear design model was first identified, then a generalized H_∞ method was employed to optimize control gains, which achieved good performance under normal wind and wave conditions. However, this method was not able to

Manuscript received September 2, 2019; revised December 5, 2019, March 21, 2020, and August 30, 2020; accepted October 13, 2020. Date of publication November 10, 2020; date of current version February 17, 2022. This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant under Agreement 765579. This article was recommended by Associate Editor M. Basin. *(Corresponding author: Xiaowei Zhao.)*

The authors are with the School of Engineering, University of Warwick, Coventry CV4 7AL, U.K. (e-mail: jincheng.zhang@warwick.ac.uk; xiaowei.zhao@warwick.ac.uk; x.wei.4@warwick.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2020.3032622.

Digital Object Identifier 10.1109/TSMC.2020.3032622

work on extreme wind and wave conditions. In [21], a contact nonlinear modeling method for barge-type floating wind turbines was presented, where a stroke-limited HMD was included. The HMD was installed in the turbine's nacelle and a state-feedback linear–quadratic regulator (LQR) controller was proposed for the active structural control.

The control designs in the aforementioned works were all based on linear models, whether they were formulated based on physical principals or identified by synthetic data. However, the motion of floating wind turbines in the offshore environment can be quite complex and the turbine position can be quite far away from equilibrium, due to the extreme wind/waves loads. Thus, a control strategy that can take account of the nonlinear dynamics of floating wind turbines is required. In the present article, for the first time, we apply reinforcement learning-based control approach for the load alleviation of floating wind turbines. Specifically, an adaptive dynamic programming (ADP) algorithm is employed, which can take account of the nonlinear dynamics of the structural system in the control design process, thus the designed controller can achieve optimal performance in both normal and extreme conditions.

ADP is a powerful tool for optimal control problems. Originally proposed by Werbos [22], [23], ADP has caught extensive attention recently on the optimal control of both continuous-time (CT) and discrete-time (DT) systems [24]–[31]. It is specifically designed and developed to tackle the control of complex nonlinear systems. Typical examples include coal gasification [32], energy management systems [33], hypersonic vehicle tracking [34], microgrid system [35], and optimal tracking [36]. However, to the best of our knowledge, its application in structural control of floating turbines has not yet been explored. There are mainly two types of iterative ADP algorithms: 1) value iteration approximate dynamic programming (VI-ADP) and 2) policy iteration approximate dynamic programming (PI-ADP) [37]–[39]. For VI-ADP approaches, the iteration begins with an initial value function and then the policy improvement is carried out according to the iterative value function. This approach does not require an initial admissible control law. However, the initialization of the value function needs to be designed in order to guarantee the stability and convergence of the iteration process. The paper [40] proposed a VI-ADP approach with a value function initialization technique, where the convergence property was also proved. In [41] a generalized VI-ADP algorithm was proposed, which only requires an arbitrary positive semidefinite function to initialize the value function in order to guarantee the convergence property of the algorithm. On the other hand, for the PI-ADP approaches, an admissible initial control law is required for the iteration process. In [42], a DT PI-ADP algorithm was proposed for nonlinear systems with convergence and stability analysis, and an effective method to obtain the initial admissible control law was given. A generalized PI-ADP approach was proposed in [43], which relaxed the requirement of obtaining the initial admissible policy [28].

In our work, the PI-ADP approach for DT systems [28] is employed for the structural control of floating wind turbines. Because the considered open-loop system (the floating wind turbine with passive TMD) is stable, there exists a natural admissible control policy (i.e., active control force set to be 0). It will be used for the initialization of the PI-ADP approach that can ensure the stability of the closed-loop system [28], [42]. The employed algorithm includes three networks, i.e., an action network, a critic network, and a plant network. The training of the plant network and critic network is carried out in supervised manner while the training of the action network aims to minimize the critic network output, which requires the gradient information flowing through all the three networks. The network structures in the existing literature are usually very simple where only weight vectors outside the neural network (NN) activations were included in the NN structure or the weight matrix in the hidden layer were included but not used for training, which undermined the ability of NNs in approximating the plant's complex nonlinear behavior in practical applications. This allows the gradients being derived analytically, but it is infeasible to do so for complex network structures. In this work, the automatic differentiation is employed to calculate the gradients in all the NN training, including the training of the hidden layer and this method is independent of specific applications, which greatly extends the PI-ADP algorithm's ability in solving complex practical problems and simplifies its implementation. The large-scale machine learning platform Tensorflow [44] is used for the implementation of the proposed NN structure. Its highly parallel computing environment makes our NN implementation even more powerful, especially with the use of GPU.

The National Renewable Energy Laboratory (NREL) 5-MW baseline ITI energy barge wind turbine model [1] is used in this study. An HMD is installed on the platform and designed to suppress the vibration in the fore-aft direction. We mention that the side-to-side vibration of the platform can be well tackled by passive TMDs. The NREL Flow Analysis Software Toolkit (FAST) code [2] is employed to simulate the structural system, and the plant network is trained based on the data generated by FAST. After training the plant network, a series of ADP controllers are obtained by varying the penalty term in the action–critic network training, which considers the tradeoff between the control performance and power consumption.

The remainder of this article is organized as follows: the structural control of floating wind turbines is formulated in Section II. The PI-ADP algorithm and its implementation with the proposed NN structure are described in Section III, where the training of the plant network, the critic network, and the action network is presented in detail. The structural control design based on PI-ADP is described in Section IV. The control performance is evaluated in Section V, where a set of wind/wave conditions are considered. Finally, the conclusions are drawn in Section VI.

II. Problem Formulation

The structural control of a floating wind turbine is described in this section. Here, the turbine's structural dynamics with HMD and the control objective are given.

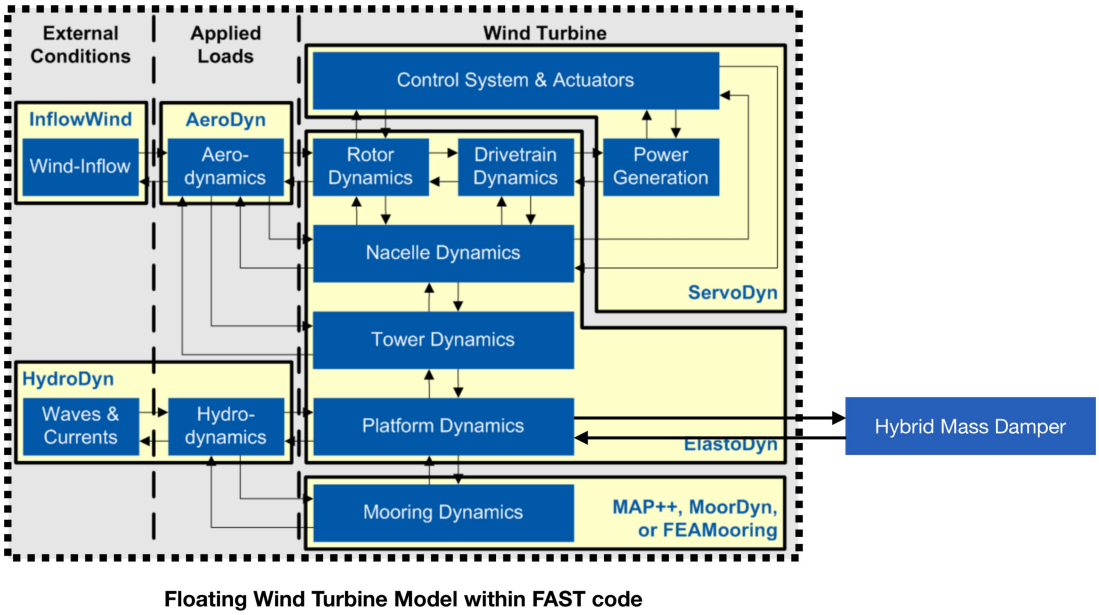


Fig. 1. Schematic illustration of the floating wind turbine model within FAST [45] coupled with an HMD.

A. Floating Turbine System With HMD

The structural control of an NREL 5-MW floating wind turbine model within FAST code [1] is investigated here. An HMD is coupled to the floating platform of this model, which moves in the fore-aft direction to suppress the structural vibration of the floating turbine in this direction (see Fig. 1). A stroke limit of ± 17 m is imposed for the HMD since the length of the platform is 40 m. The damping coefficient, stiffness coefficient, and mass of the HMD are set as 60 393 N/(m/s), 103 019 N/m, and 400 000 kg, respectively, which are the optimal values used in [20]. All the DOFs are enabled except the nacelle yaw DOF as the yaw control is not considered in this article. Among all the DOFs, the main structural dynamics of this turbine-HMD system can be characterized by the platform pitch angle, the tower-top displacement, and the HMD displacement. Therefore, the structural system can be approximated by a discrete system F

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k, u_k), \quad k = 0, 1, 2, \ldots \tag{1}$$

where k is the discrete time step, u is the control variable (i.e., the active HMD force), and $\mathbf{x}$ is the state variable which is defined as

$$\mathbf{x} = \left[x_{hmd}, u_{hmd}, x_{plfm}, u_{plfm}, x_{tt}, u_{tt}\right]. \tag{2}$$

Here, x_{hmd}, u_{hmd}, x_{plfm}, u_{plfm}, x_{tt}, and u_{tt} represent the HMD displacement, HMD velocity, platform pitch angle, platform pitch angular velocity, tower top displacement, and tower-top velocity, respectively.

B. Control Objective

The active structural control aims at reducing the vibrations of the turbine's platform and tower in the fore-aft direction with a minimum amount of power consumption. Denote the sequence of active HMD forces as $\bar{u}_k = \{u_k, u_{k+1}, u_{k+2}, \ldots\}$, then the cost function for the state $\mathbf{x}_0$ under the control $\bar{u}_0$ is defined as

$$J(\mathbf{x}_0, \bar{u}_0) = \sum_{k=0}^{\infty} U(\mathbf{x}_k, u_k) \tag{3}$$

where the utility function $U(\mathbf{x}_k, u_k)$ is defined as

$$U(\mathbf{x}_k, u_k) = (\mathbf{x}_k)^{\mathrm{T}} \cdot A_u \cdot (\mathbf{x}_k) + B_u \cdot (u_k)^2 \tag{4}$$

where $\cdot$ represents the dot product, the superscript T represents the matrix transpose, and the empirical parameters A_u and B_u are used to investigate the tradeoff between the active control force and the control performance. Equation (4) allows the utility function $U(\mathbf{x}_k, u_k)$ to be positive definite [28], [42] and to take account of the costs from both the structural vibrations (which is described by the first term) and the active power consumption (which is described by the second term).

Here, we focus on state-feedback control, thus an arbitrary control law can be expressed as

$$u_k = \mu(\mathbf{x}_k). \tag{5}$$

The cost function for the state $\mathbf{x}_0$ under the control law μ can then be expressed as

$$J^{\mu}(\mathbf{x}_0) = \sum_{k=0}^{\infty} U(\mathbf{x}_k, \mu(\mathbf{x}_k)). \tag{6}$$

The structural control objective is to find an optimal control policy $\mu^*(\mathbf{x}_k)$ such that

$$\mu^*(\mathbf{x}_k) = \arg\min_{u_k} \left\{U(\mathbf{x}_k, u_k) + J^*(F(\mathbf{x}_k, u_k))\right\} \tag{7}$$

where

$$J^*(\mathbf{x}_k) = \min_{u_k} \left\{U(\mathbf{x}_k, u_k) + J^*(F(\mathbf{x}_k, u_k))\right\}. \tag{8}$$

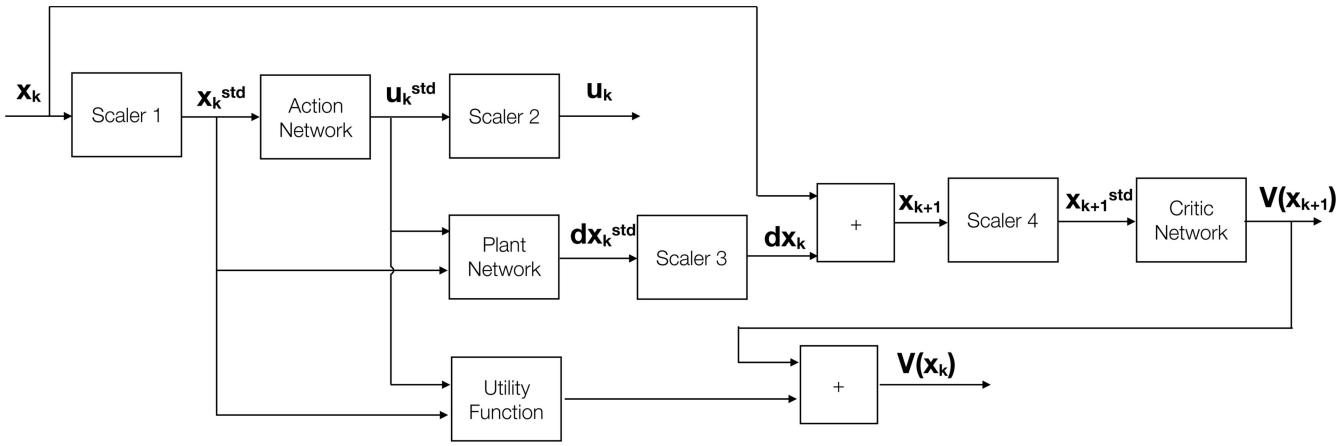


Fig. 2. Whole NN structure, including the plant network, the action network and the critic network.

III. PI-ADP Algorithm and its Implementation

The PI-ADP algorithm begins with an admissible control law μ_0 and then obtains the optimal cost function and the optimal control law iteratively through the policy evaluation and policy improvement procedure. During policy evaluation, the value function V_i is constructed based on the corresponding control law μ_i such that it satisfies the following equation:

$$V_i(\mathbf{x}_k) = U(\mathbf{x}_k, \mu_i(\mathbf{x}_k)) + V_i(F(\mathbf{x}_k, \mu_i(\mathbf{x}_k))). \tag{9}$$

Then, during the policy improvement, the control law μ_{i+1} is updated based on the value function V_i according to

$$\mu_{i+1}(\mathbf{x}_k) = \arg\min_{u_k}\{U(\mathbf{x}_k, u_k) + V_i(F(\mathbf{x}_k, u_k))\}. \tag{10}$$

Through the iteration process ($\mu_0 \rightarrow V_0 \rightarrow \mu_1 \rightarrow V_1 \rightarrow \mu_2 \rightarrow \cdots V_{N-1} \rightarrow \mu_N$), the optimal cost function J^* is approximated by V_N and the optimal control law μ^* is approximated by μ_N. The properties of the PI-ADP algorithm have been proved in [28] and [42], where an admissible initial control law is required to guarantee the convergence and stability of the algorithm. The main focus of this article is a novel NN realization of the employed PI-ADP algorithm and its application on structural control of floating wind turbine (which is also applicable to other complex industrial systems). The interested reader may refer to [28] and [42] for the detailed proof of stability. The proposed NN structure and its training details are described in the following sections.

A. Neural Network Structure

The whole NN structure proposed in this work is illustrated in Fig. 2. The plant network, the action network, and the critic network in Fig. 2 are all simple NNs with one-hidden layer as illustrated in Fig. 3. Here, we have designed the network in order to feed standardized data for all the NN trainings. We employ the standard scaler (denoted as Scaler 1, Scaler 2, Scaler 3, and Scaler 4 in Fig. 2), which normalizes the data by their mean value and standard deviation (SD)

$$\mathbf{d}^{\text{std}} = \frac{\mathbf{d} - m(\mathbf{d})}{s(\mathbf{d})} \tag{11}$$

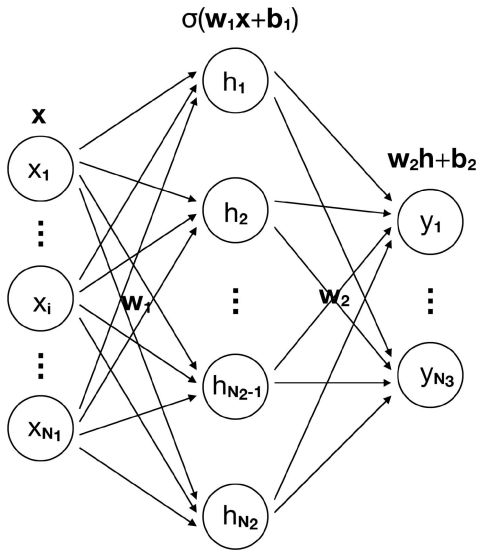


Fig. 3. Illustration of an artificial NN with one hidden layer. $\mathbf{x}$ is the N_1-dimension input variable, $\mathbf{h}$ is the N_2-dimension hidden layer output, and $\mathbf{y}$ is the N_3-dimension output variable. $\mathbf{w}_1$, $\mathbf{b}_1$, $\mathbf{w}_2$, and $\mathbf{b}_2$ are the weight matrix of the NN and σ is the activation function.

where $m(\mathbf{d})$ and $s(\mathbf{d})$ represent the mean and SD of the dataset $\mathbf{d}$. The utility function is redefined in terms of the standardized state and action variables as

$$U\left(\mathbf{x}_k^{\text{std}}, u_k^{\text{std}}\right) = \left(\mathbf{x}_k^{\text{std}}\right)^{\text{T}} \cdot A_u^* \cdot \left(\mathbf{x}_k^{\text{std}}\right) + B_u^* \cdot \left(u_k^{\text{std}}\right)^2 \tag{12}$$

so that the costs arising from structural vibration and the power consumption are comparable. The forward and inverse mappings of Scalers 1–4 are denoted as $\mathscr{S}_1$–$\mathscr{S}_4$ and $\mathscr{S}_1^{-1}$–$\mathscr{S}_4^{-1}$, respectively. The forward mappings of the plant network, the action network, and the critic network are denoted as $\mathscr{P}$, $\mathscr{A}$, and $\mathscr{C}$, respectively. The plant network is designed to approximate the structural system such that

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathscr{S}_3^{-1} \circ \mathscr{P}(\mathscr{S}_1(\mathbf{x}_k), \mathscr{S}_2(u_k)) \tag{13}$$

where $\circ$ represents the composition of functions. The plant network is designed to predict the state change instead of the state variable because this way can greatly increase the

prediction accuracy—the state change is usually subtle compared to the state variable. The action network is designed to approximate the structural controller such that

$$u_k = \mathscr{S}_2^{-1} \circ \mathscr{A} \circ \mathscr{S}_1(\mathbf{x}_k). \tag{14}$$

The critic network is designed to approximate the value function such that

$$V(\mathbf{x}_k) = \mathscr{C} \circ \mathscr{S}_4(\mathbf{x}_k). \tag{15}$$

Thus, with this NN structure, (9) is approximated by

$$\begin{aligned}\mathscr{C}_i \circ \mathscr{S}_4(\mathbf{x}_k) = U(\mathscr{S}_1(\mathbf{x}_k), \mathscr{A}_i \circ \mathscr{S}_1(\mathbf{x}_k)) + \mathscr{C}_i \circ \mathscr{S}_4 \\ \times \left(\mathbf{x}_k + \mathscr{S}_3^{-1} \circ \mathscr{P}(\mathscr{S}_1(\mathbf{x}_k), \mathscr{A}_i \circ \mathscr{S}_1(\mathbf{x}_k))\right)\end{aligned} \tag{16}$$

and (10) is approximated by

$$\begin{aligned}&\mathscr{A}_{i+1} \circ \mathscr{S}_1(\mathbf{x}_k) \\ &= \underset{\mathscr{A} \circ \mathscr{S}_1(\mathbf{x}_k)}{\arg\min} \Big\{ U(\mathscr{S}_1(\mathbf{x}_k), \mathscr{A} \circ \mathscr{S}_1(\mathbf{x}_k)) + \mathscr{C}_i \circ \mathscr{S}_4 \\ &\qquad \times \left(\mathbf{x}_k + \mathscr{S}_3^{-1} \circ \mathscr{P}(\mathscr{S}_1(\mathbf{x}_k), \mathscr{A} \circ \mathscr{S}_1(\mathbf{x}_k))\right)\Big\}.\end{aligned} \tag{17}$$

B. Neural Network Training

The training of the proposed NN structure is detailed here and the overall training process is summarized in Algorithm 1.

Equation (13) can be reformulated as

$$\mathscr{S}_3(\mathbf{x}_{k+1} - \mathbf{x}_k) = \mathscr{P}(\mathscr{S}_1(\mathbf{x}_k), \mathscr{S}_2(u_k)) \tag{18}$$

thus the plant network is trained by minimizing the mean-squared error (MSE) loss

$$l_p = \mathrm{MSE}(\mathscr{S}_3(\mathbf{x}_{k+1} - \mathbf{x}_k), \mathscr{P}(\mathscr{S}_1(\mathbf{x}_k), \mathscr{S}_2(u_k))). \tag{19}$$

The critic network training can be done in two different approaches. In the first approach, the critic network is trained such that (9) is satisfied. In the second approach, the value function is approximated directly by

$$V(\mathbf{x}_k) = \sum_{j=0}^{N} U\left(\mathbf{x}_{k+j}^{\mathrm{std}}, u_{k+j}^{\mathrm{std}}\right) \tag{20}$$

where N is a large number. In this work, we employ the second approach, as it converges faster than the first one. Thus, the critic network is trained by minimizing

$$l_c = \mathrm{MSE}\left(\sum_{j=0}^{N} U\big(\mathscr{S}_1\big(\mathbf{x}_{k+j}\big), \mathscr{A} \circ \mathscr{S}_1\big(\mathbf{x}_{k+j}\big)\big), \mathscr{C} \circ \mathscr{S}_4(\mathbf{x}_k)\right) \tag{21}$$

where $\mathbf{x}_{k+j}$ is obtained by evaluating the plant network iteratively.

According to (17), the action network is trained by minimizing

$$\begin{aligned}l_a = \mathrm{MSE}\Big(0, U(\mathscr{S}_1(\mathbf{x}_k), \mathscr{A} \circ \mathscr{S}_1(\mathbf{x}_k)) + \mathscr{C} \circ \mathscr{S}_4 \\ \times \left(\mathbf{x}_k + \mathscr{S}_3^{-1} \circ \mathscr{P}(\mathscr{S}_1(\mathbf{x}_k), \mathscr{A} \circ \mathscr{S}_1(\mathbf{x}_k))\right)\Big).\end{aligned} \tag{22}$$

Algorithm 1 Training of the Proposed NN Structure

1: Load the training data $\{x_k^*\}$, $\{u_k^*\}$ and $\{x_{k+1}^*\}$, and preprocess them to obtain $\{x_k^{std*}\}$, $\{u_k^{std*}\}$ and $\{dx_k^{std*}\}$.
2: Set the hidden layer neuron number N_2; Set the learning rate lr.
3: Train the plant network by feeding $\{x_k^{std*}\}$, $\{u_k^{std*}\}$ and $\{dx_k^{std*}\}$ into $\mathbf{x}_k^{\mathrm{std}}$, $\mathbf{u}_k^{\mathrm{std}}$, and $\mathbf{dx}_k^{\mathrm{std}}$.
4: Set the maximum iteration number N_{iter}.
5: Initialize $i = 1$; Set the convergence criterion ϵ.
6: Initialize the action network such that the initial control law is admissible.
7: **while** $i < N_{iter} + 1$ **do**
8: Compute the $\{V_i(x_{k+1}^*)\}$ according to (20) by feeding $\{x_{k+1}^*\}$ into $\mathbf{x}_k$.
9: Train the critic network by feeding $\{x_{k+1}^*\}$ and $\{V_i(x_{k+1}^*)\}$ into $\mathbf{x}_{k+1}$ and $V(\mathbf{x}_{k+1})$.
10: Train the action network by feeding $\{x_k^*\}$ into $\mathbf{x}_k$.
11: **if** $i > 1$ **then**:
12: Compute e_{conv}: the MSE between $\{V_{i-1}(\mathbf{x}_k)\}$ and $\{V_i(\mathbf{x}_k)\}$.
13: **if** $e_{conv} < \epsilon$ **then**:
14: The whole process is convergent.
15: Break.
16: **end if**
17: **end if**
18: $i \leftarrow i + 1$
19: **end while**

The critic network and the action network are trained iteratively until the whole process converges which is described in detail in Algorithm 1. All the NN trainings are carried out with the Adam optimization algorithm [46]. Automatic differentiation [47], which calculates the gradients of complex graphs automatically based on the chain rule, is employed here for deriving $\partial l_p/\partial w_p$ ($\partial l_c/\partial w_c$ or $\partial l_a/\partial w_a$) for the plant (critic or action) network training, where w_p (w_c or w_a) represents the weight matrix in the plant (critic or action) network.

The relationship between the three subnetworks is further explained here. The plant network is trained alone in a supervised manner, as only $\mathscr{P}$ is involved in (19). The critic network is also trained alone in a supervised manner as only forward evaluation of $\mathscr{A}$ and $\mathscr{P}$ is needed in deriving $\partial l_c/\partial w_c$ in (21). The training of the action network, however, requires the gradient information flowing through the whole network, as $\mathscr{P}$, $\mathscr{C}$, and $\mathscr{A}$ are all involved in deriving $\partial l_a/\partial w_a$ in (22).

IV. HMD Controller Design of Floating Wind Turbine

In this section, the design of a machine learning-based HMD controller for a floating wind turbine is investigated, using the NN structure and training algorithm proposed in the previous section.

A. Plant Network Training

The plant network is trained by using the data generated by the floating wind turbine simulation model shown

in Fig. 1. The training dataset is a set of training samples with each sample consisting of the state variable at the current time step, the action variable at the current time step and the state variable at the next time step. The time step is set as 0.06 s, which is also the time step of the trained plant network. To generate the training dataset, 100 random initial conditions for $[x_{hmd}, x_{plfm}, x_{tt}]$ in the parameter space $[-15, 15]$ m $\times$ $[-15, 15]$ deg $\times$ $[-3, 3]$ m are at first obtained by the Latin hypercube sampling method, then a 15-s simulation with a time step of 0.01 s is carried out for each initial condition, under the excitation of random HMD force with a time step of 0.1 s and within the interval $[-5000, 5000]$ kN. Next, the first 5-s time series is eliminated and the data sample is extracted with a time interval of 0.1 s. All the data samples are collected together to form the final training dataset.

After generating the training dataset, a plant network with the hidden-layer neuron number $N_2 = 20$ is constructed. The learning rate is set as 0.001 and the training error is set as 10^{-3}. The plant network is then trained to mimic the structural system. In order to assess the accuracy of the trained network, a comparison of the FAST simulation results and the plant network calculations is given in Fig. 4, with the test HMD force time series shown in Fig. 4(a) and the comparison of the structural response under this HMD force excitation shown in Fig. 4(b)–(g). Both calculations are based on the same initial condition at $t = 5$ s, and the test HMD force time series shown in Fig. 4(a) has not been used during training. Fig. 4(b)–(g) shows a perfect match between the FAST simulation and the plant network calculation for the whole simulation period. This demonstrates that the plant network has captured the nonlinear dynamics of the structural system.

B. Action–Critic Network Training

The action–critic network training is conducted iteratively according to Algorithm 1. The hidden-layer neuron number of both networks is set as 20 and the learning rate is set as 0.001. The bias term is not used for the action network, imposing the condition $\mu(0) = 0$. The training error of the critic network is set as 10^{-3} and the training of action network is deemed completed when the training loss l_a drops less than a prescribed threshold with further training, which is set as 10^{-5} here. The action network is initialized by very small random weights as $\mu_0 = 0$ is an admissible control law for the structural system. N in (20) is set as 5000. A_u^* in the utility function is set as $10^{-4} \times \text{diag}\{(1, 1, 25, 25, 1, 1)\}$ and a number of values are chosen for B_u^*, which are reported in Table I. Each chosen B_u^* results in a different action network after the training process. These trained action networks are the state-feedback controllers that will be used for the structural control in the next section (denoted as ADP1–ADP8 hereafter). The final converged training results are reported in the Appendix.

All the NN trainings are carried out with one NVIDIA Tesla K80 GPU card to take advantage of Tensorflow's efficiency with GPU backend. For all the eight (ADP1–ADP8) controller designs, on average, the plant network training requires about 88 s, the critic network training requires about 33 s, and the action network training requires about 245 s. The action network training takes much longer than other NNs as it requires the gradient information flowing through the whole network (all three subnetworks are involved). The same training process is also tested out with 4 INTEL Xeon CPUs (2.40 GHz) and it is more than three times slower than training with GPU. This clearly demonstrates the advantage of the current implementation on Tensorflow with GPU backend.

V. Simulation Study

With the converged plant network capturing the dynamics of the structural system and the converged action network approximating the optimal control law, which are developed above, this section is devoted to simulation tests.

A. Wind and Wave Conditions

The turbulent wind is generated based on the IEC Kaimai spectral model with NTM in TurbSim [48], and the wave condition is generated by the HydroDyn module in FAST based on the JONSWAP spectrum. Two extreme events and two normal events are considered at first to analyze the control performance. Then, a range of wind/wave conditions are included to further demonstrate the effectiveness of the ADP controllers. For the two extreme events (event $E1$ and event $E2$), which were recorded in the report [1], the main hub-height longitudinal wind speeds are, respectively, 22 and 24 m/s, the turbulence intensity is category B, and the peak-spectral periods of the incident waves are 13.4 and 15.5 s with the significant wave heights of 4.7 and 5.5 m, respectively. For the two normal events (event $N1$ and event $N2$), the main hub-height longitudinal wind speeds are, respectively, 9 and 18 m/s, the turbulence intensity is category A, and the peak-spectral periods of the incident waves are 12 and 11 s with the significant wave heights of 2 and 4.5 m, respectively. For the remaining cases (ranging from normal to extreme conditions), the wind speed increases from 9 to 24 m/s with an interval of 3 m/s, and B level turbulence intensity is used for all the cases. The corresponding significant wave heights increase linearly from 2 to 5.5 m and peak-spectral periods increase linearly from 12 to 15.5 s.

B. Performance Evaluation and Discussion

The simulation results are given here, including the calculations with no TMD, passive TMD, HMD using H_∞ controller, and HMD using a series of ADP controllers (ADP1–ADP8).

The SD of the platform pitch angle and the corresponding HMD power consumption for the two normal events ($N1$ and $N2$) and the two extreme events ($E1$ and $E2$) are given in Table I. Fig. 5 shows the corresponding time responses. First, the performances of the ADP8 controller and the H_∞ controller are compared because their active power consumption are similar. In the two normal events, the SDs of the platform pitch displacement are reduced by 12.34% and 11.69% using the controller ADP8, compared with the passive case, while they are reduced by 10.83% and 9.64% using the H_∞ controller. It is concluded that the ADP controller and the H_∞ controller perform similarly in normal events. However, in the two extreme events, as can be seen from Fig. 5(c)

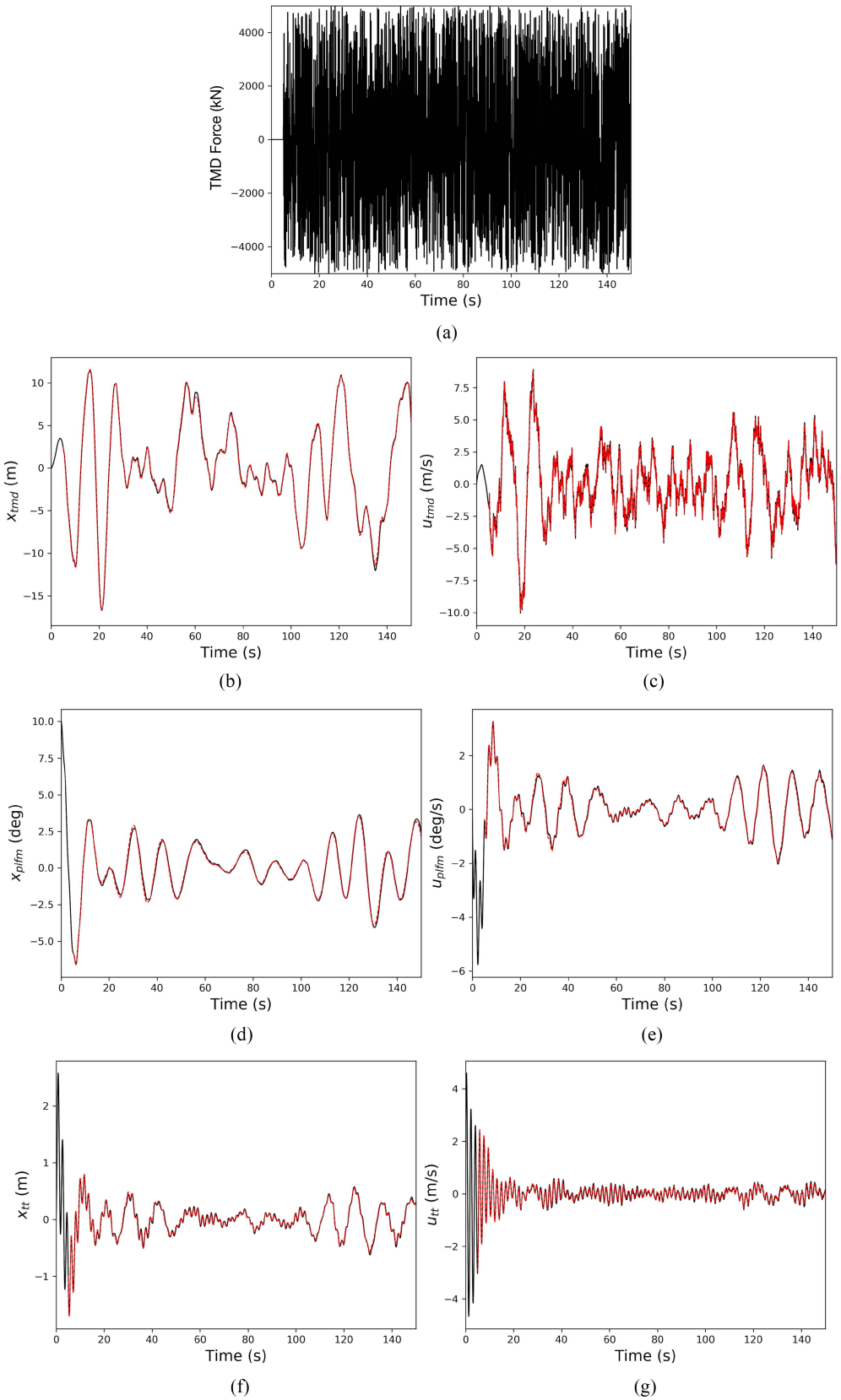


Fig. 4. Comparison of the FAST simulation results (solid line) and the plant network calculations (dashed line). (a) Test HMD force. (b) HMD displacement. (c) HMD velocity. (d) Platform pitch angle. (e) Platform pitch angular velocity. (f) Tower top displacement. (g) Tower top velocity.

and (d), the platform pitch displacement with the controller ADP8 is much smaller than the ones with the H_∞ controller. Compared with the passive case, the SDs of the pitch displacement are reduced by 14.64% and 10.15% using the controller ADP8 while they are reduced by 8.72% and 2.67% using the H_∞ controller. In conclusion, a clear advantage of

TABLE I
RESULTS FOR THE TWO NORMAL EVENTS AND THE TWO EXTREME EVENTS, INCLUDING THE SIMULATION WITH NO TMD, PASSIVE TMD, HMD USING H_∞ CONTROLLER, AND HMD USING A SERIES OF ADP CONTROLLERS. THE SD OF THE PLATFORM PITCH ANGLE (IN DEGREE) AND THE CORRESPONDING HMD POWER(IN KW) ARE REPORTED

Controller	No TMD	Passive	ADP1	ADP2	ADP3	ADP4	ADP5	ADP6	ADP7	ADP8	H_∞
B_u^* $(\times 10^{-4})$	-	-	5	10	25	50	100	250	500	1000	-
N1:HMD Power	-	-	236.73	187.56	172.82	105.82	106.10	52.011	33.842	23.046	20.88
N1:Pitch SD	1.8533	1.4127	0.8684	0.9339	0.9515	1.0283	1.0346	1.1299	1.1837	1.2384	1.2597
N2:HMD Power	-	-	657.62	540.32	434.83	261.96	251.99	111.84	64.230	41.115	40.47
N2:Pitch SD	2.7134	2.0970	1.3513	1.4065	1.4263	1.5297	1.5423	1.6845	1.7693	1.8519	1.8949
E1:HMD Power	-	-	719.25	617.67	499.50	330.88	325.59	188.93	129.03	92.358	114.91
E1:Pitch SD	4.4224	2.9271	1.7267	1.8189	1.8821	2.0319	2.0678	2.2826	2.3984	2.4987	2.6718
E2:HMD Power	-	-	863.59	770.04	620.92	442.48	424.88	258.03	182.69	134.00	177.83
E2:Pitch SD	5.4536	3.5648	2.1236	2.2438	2.3652	2.5654	2.6575	2.9308	3.0783	3.2028	3.4697

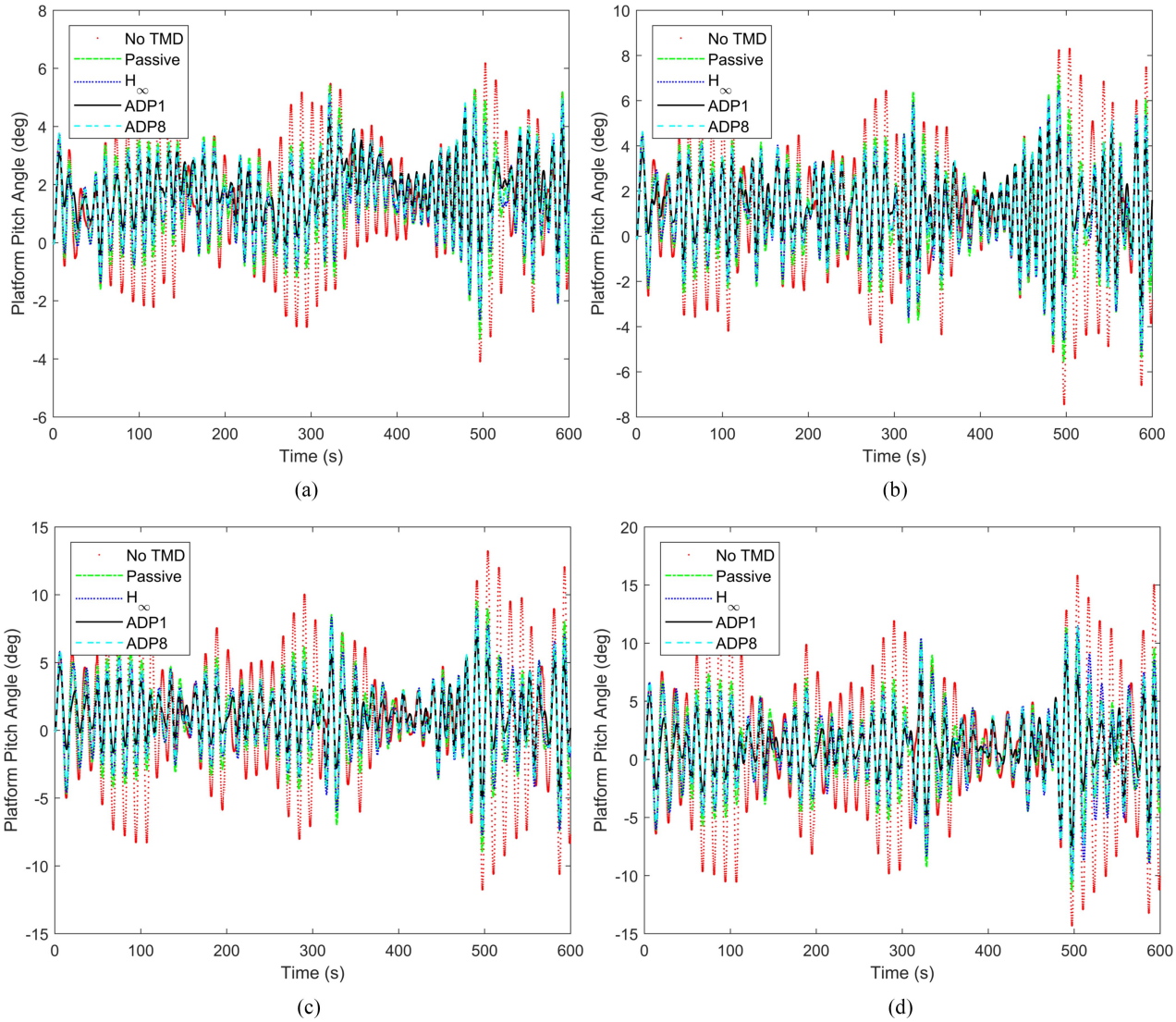


Fig. 5. Simulation results, including the ones with no TMD, passive TMD, HMD using H_∞ controller, HMD using ADP1 controller, and HMD using ADP8 controller, for the normal event (a) $N1$ and (b) $N2$, and the extreme event (c) $E1$ and (d) $E2$. Among ADP controllers, ADP1 controller is the most effective one in terms of vibration suppression and the ADP8 controller uses similar amount of HMD power as the H_∞ controller.

the ADP controller over H_∞ controller is observed in the extreme events but the ADP controller performs only slightly better in the normal events. This observation is reasonable because the H_∞ controller is expected to perform well in the linear range of the dynamic systems and the ADP controller shows its advantages in strongly nonlinear situations.

TABLE II
RESULTS FOR A RANGE OF WIND/WAVE CONDITIONS (WIND SPEED FROM 9 m/s TO 24 m/s), INCLUDING THE SIMULATION WITH NO TMD, PASSIVE TMD, HMD USING H_∞ CONTROLLER, AND HMD USING A SERIES OF ADP CONTROLLERS. THE SD OF THE PLATFORM PITCH ANGLE (IN DEGREE) AND THE CORRESPONDING HMD POWER (IN KW) ARE REPORTED. THE 24 m/s CASE IS EXACTLY THE SAME AS E2 IN TABLE I THUS, OMITTED HERE

Controller	No TMD	Passive	ADP1	ADP2	ADP3	ADP4	ADP5	ADP6	ADP7	ADP8	H_∞
B_u^* $(\times 10^{-4})$	-	-	5	10	25	50	100	250	500	1000	-
9m/s:HMD Power	-	-	237.16	186.46	172.36	106.02	105.54	51.92	33.69	22.84	20.58
9m/s:Pitch SD	1.8369	1.4015	0.8571	0.9207	0.9350	1.0122	1.0172	1.1144	1.1698	1.2253	1.2473
12m/s:HMD Power	-	-	274.50	215.01	205.30	126.28	124.71	59.46	37.83	24.55	22.48
12m/s:Pitch SD	1.9335	1.5401	1.0318	1.1230	1.1905	1.2437	1.2690	1.3259	1.3734	1.3960	1.4147
15m/s:HMD Power	-	-	291.43	236.96	224.31	135.74	140.73	66.73	42.61	28.46	30.97
15m/s:Pitch SD	2.1640	1.7273	1.0793	1.1398	1.1656	1.2584	1.2713	1.3955	1.4636	1.5298	1.5930
18m/s:HMD Power	-	-	424.41	360.42	325.27	211.48	217.04	115.21	78.39	54.36	64.94
18m/s:Pitch SD	3.1292	2.3123	1.3559	1.4225	1.4653	1.5963	1.6209	1.7979	1.8935	2.0019	2.1246
21m/s:HMD Power	-	-	613.38	532.42	454.32	311.65	313.37	183.77	128.66	92.36	115.01
21m/s:Pitch SD	4.2646	2.9511	1.7295	1.8237	1.8980	2.0484	2.0971	2.3410	2.4476	2.5410	2.7425

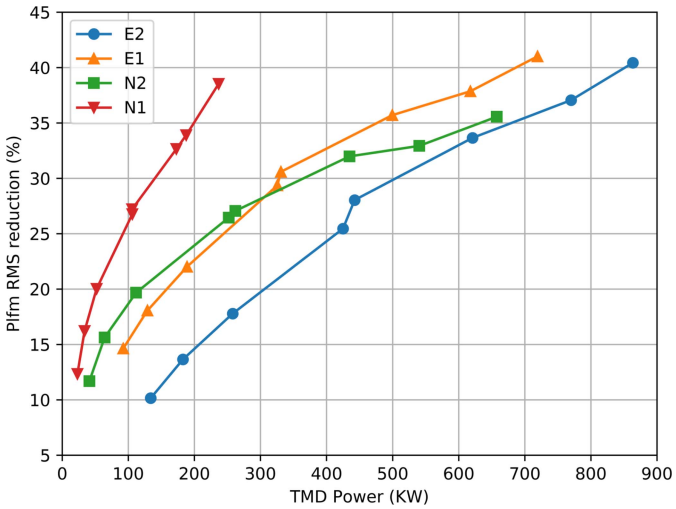


Fig. 6. Reduction of the SDs of the platform pitch displacements against the HMD power, for the events $E1$, $E2$, $N1$, and $N2$. The symbols in each line (from left to right) represent the results obtained by ADP8, ADP7, ..., ADP1.

The simulation results of ADP1, the controller with the most effective vibration suppression performance, are also given in Fig. 5, with the SDs of the platform pitch displacement reduced by 38.53% and 35.56% in the events $N1$ and $N2$, and by 41.01% and 40.43% in the events $E1$ and $E2$. In addition, the HMD power consumptions are 719.25 and 863.59 kW in $E1$ and $E2$ and 236.73 and 657.62 kW in $N1$ and $N2$. In [20], an average reduction of 18.1% for the platform pitch root-mean square by the generalized H_∞ controller was reported with an average power consumption of 684 kW. In addition, they stated that their generalized H_∞ controller was not able to work under the extreme wind and wave conditions. The results here clearly demonstrate the great advantage of the machine learning-based approach over the H_∞ control approach in the structural control of floating wind turbines.

By changing the penalty coefficient B_u^* related to the HMD force magnitude in the utility function, a set of controllers have been obtained which consider the tradeoff between the control performance and power consumption. Fig. 6 shows the reduction of the SDs of the platform pitch displacements against the power consumptions for the events $N1$, $N2$, $E1$, and $E2$. It shows that ADP1 is the suitable choice if the ability of the wind turbines to withstand extreme conditions is the primary concern, while ADP5 may be more suitable if the HMD power consumption becomes more concerned.

To further evaluate the ADP controllers' performance, the simulations ranging from normal conditions to extreme conditions (from 9 to 24 m/s with an interval of 3 m/s) are carried out and the results are given in Table II. As can be seen, the proposed ADP controllers perform very well for all the cases.

VI. CONCLUSION

The machine learning-based structural control of floating wind turbines has been investigated. An HMD was installed on the floating platform in order to reduce the platform vibration, and the ADP approach was employed to obtain the optimal control law. The design of the NN structure and its implementation on the modern large-scale machine learning platform Tensorflow was proposed. Three networks were included in the whole NN structure, including a plant network, a critic network, and an action network. After training, the approximate optimal controller was obtained, based on the plant network that captured the nonlinear dynamics of the structural system. The simulation results showed that the ADP controller performed extremely well in both normal conditions and extreme conditions. A clear advantage of the ADP controllers over the H_∞ controller was observed, especially for extreme conditions—the scenarios that must be considered seriously in offshore wind technology.

In addition, our algorithm allows to consider the tradeoff between the control performance and the power consumption. A series of ADP controllers were obtained by varying the

penalty term in the network training. As expected, the control performance increased with the increase of power consumption. We mention that in practice, the passive TMD is expected to work alone most of the time and the active part only works when the vibration is above a certain limit.

APPENDIX
TRAINING RESULTS

The final ADP controllers used in this article are given here. The ADP state-feedback control can be expressed as

$$u = s_u\left[\mathbf{w}_2\sigma\left(\mathbf{w}_1\frac{\mathbf{x}}{s_{\mathbf{x}}}\right)\right] \tag{23}$$

where s_u, the SD of the action variable in the training dataset, is 2.882487×10^6 and $s_{\mathbf{x}}$, the SD of the state variable in the training dataset, is [7.614702, 4.324959, 3.804557, 1.873835, 0.815144, 1.931296]. σ represents the hyperbolic tangent function. The weight matrix $\mathbf{w}_1$ and $\mathbf{w}_2$ for ADP1–ADP8 can be found in the supporting materials of this article.

REFERENCES

[1] J. M. Jonkman, "Dynamics modeling and loads analysis of an offshore floating wind turbine," Nat. Renewable Energy Lab., U.S. Dept. Energy, Golden, CO, USA, Rep. NREL/TP-500-41958, 2007.

[2] J. M. Jonkman and M. L. Buhl, Jr., "Fast user's guide," Nat. Renewable Energy Lab., U.S. Dept. Energy, Golden, CO, USA, Rep. NREL/TP-500-38230, 2005.

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

[4] H. Namik and K. Stol, "Individual blade pitch control of floating offshore wind turbines," *Wind Energy Int. J. Progress Appl. Wind Power Convers. Technol.*, vol. 13, no. 1, pp. 74–85, 2010.

[5] P. A. Fleming, A. Peiffer, and D. Schlipf, "Wind turbine controller to mitigate structural loads on a floating wind turbine platform," in *Proc. ASME 35th Int. Conf. Ocean Offshore Arctic Eng.*, 2016, Art. no. V006T09A044.

[6] M. A. Lackner, "An investigation of variable power collective pitch control for load mitigation of floating offshore wind turbines," *Wind Energy*, vol. 16, no. 3, pp. 435–444, 2013.

[7] X. Zhao and G. Weiss, "Strong stabilisation of a wind turbine tower model in the plane of the turbine blades," *Int. J. Control*, vol. 87, no. 10, pp. 2027–2034, 2014.

[8] C. C. Chang and H. T. Y. Yang, "Control of buildings using active tuned mass dampers," *J. Eng. Mech.*, vol. 121, no. 3, pp. 355–366, 1995.

[9] G. Stewart and M. Lackner, "Offshore wind turbine load reduction employing optimal passive tuned mass damping systems," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1090–1104, Jul. 2013.

[10] V. N. Dinh and B. Basu, "Passive control of floating offshore wind turbine nacelle and spar vibrations by multiple tuned mass dampers," *Struct. Control Health Monitor.*, vol. 22, no. 1, pp. 152–176, 2015.

[11] X. Tong, X. Zhao, and S. Zhao, "Load reduction of a monopile wind turbine tower using optimal tuned mass dampers," *Int. J. Control*, vol. 90, no. 7, pp. 1283–1298, 2017.

[12] Y. Si, H. R. Karimi, and H. Gao, "Modelling and optimization of a passive structural control design for a spar-type floating wind turbine," *Eng. Struct.*, vol. 69, pp. 168–182, Jun. 2014.

[13] S. Colwell and B. Basu, "Tuned liquid column dampers in offshore wind turbines for structural control," *Eng. Struct.*, vol. 31, no. 2, pp. 358–368, 2009.

[14] C. Roderick, "Vibration reduction of offshore wind turbines using tuned liquid column dampers," M.S. thesis, Dept. Mech. Eng., Univ. Massachusetts Amherst, Amherst, MA, USA, 2012.

[15] C. Coudurier, O. Lepreux, and N. Petit, "Passive and semi-active control of an offshore floating wind turbine using a tuned liquid column damper," *IFAC PapersOnLine*, vol. 48, no. 16, pp. 241–247, 2015.

[16] X. Tong, X. Zhao, and A. Karcanias, "Passive vibration control of an offshore floating hydrostatic wind turbine model," *Wind Energy*, vol. 21, no. 9, pp. 697–714, 2018.

[17] M. A. Lackner and M. A. Rotea, "Passive structural control of offshore wind turbines," *Wind energy*, vol. 14, no. 3, pp. 373–388, 2011.

[18] M. A. Lackner and M. A. Rotea, "Structural control of floating wind turbines," *Mechatronics*, vol. 21, no. 4, pp. 704–719, 2011.

[19] G. M. Stewart and M. A. Lackner, "The effect of actuator dynamics on active structural control of offshore wind turbines," *Eng. Struct.*, vol. 33, no. 5, pp. 1807–1816, 2011.

[20] X. Li and H. Gao, "Load mitigation for a floating wind turbine via generalized H_∞ structural control," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 332–342, Jan. 2016.

[21] Y. Hu and E. He, "Active structural control of a floating wind turbine with a stroke-limited hybrid mass damper," *J. Sound Vib.*, vol. 410, pp. 447–472, Dec. 2017.

[22] P. J. Werbos, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control*. Cambridge, MA, USA: MIT Press, 1990, pp. 67–95.

[23] P. J. Werbos, "Approximate dynamic programming for real time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. New York, NY, USA: Van Nostrand Reinhold, 1992, pp. 493–525.

[24] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, 3rd Quart., 2009.

[25] F. L. Lewis and D. Liu, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, vol. 17. Hoboken, NJ, USA: Wiley, 2013.

[26] D. Wang, D. Liu, Q. Zhang, and D. Zhao, "Data-based adaptive critic designs for nonlinear robust optimal control with uncertain dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 11, pp. 1544–1555, Nov. 2016.

[27] D. Wang, D. Liu, H. Li, B. Luo, and H. Ma, "An approximate optimal control approach for robust stabilization of a class of discrete-time nonlinear systems with uncertainties," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 5, pp. 713–717, May 2016.

[28] D. Liu, Q. Wei, D. Wang, X. Yang, and H. Li, *Adaptive Dynamic Programming With Applications in Optimal Control*. Cham, Switzerland: Springer, 2017.

[29] Y. Jiang and Z. P. Jiang, *Robust Adaptive Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2017.

[30] S. Xue, B. Luo, and D. Liu, "Event-triggered adaptive dynamic programming for zero-sum game of partially unknown continuous-time nonlinear systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 9, pp. 3189–3199, Sep. 2020.

[31] H.-N. Wu and Z.-Y. Liu, "Data-driven guaranteed cost control design via reinforcement learning for linear systems with parameter uncertainties," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4151–4159, Nov. 2020.

[32] Q. Wei and D. Liu, "Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1020–1036, Oct. 2014.

[33] Q. Wei, D. Liu, G. Shi, and Y. Liu, "Multibattery optimal coordination control for home energy management systems via distributed iterative adaptive dynamic programming," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4203–4214, Jul. 2015.

[34] C. Mu, Z. Ni, C. Sun, and H. He, "Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 584–598, Mar. 2017.

[35] D. Wang, H. He, C. Mu, and D. Liu, "Intelligent critic control with disturbance attenuation for affine dynamics including an application to a microgrid system," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4935–4944, Jun. 2017.

[36] H. Dong, X. Zhao, and B. Luo, "Optimal tracking control for uncertain nonlinear systems with prescribed performance via critic-only ADP," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Jul. 7, 2020, doi: 10.1109/TSMC.2020.3003797.

[37] Q. Wei, F. L. Lewis, D. Liu, R. Song, and H. Lin, "Discrete-time local value iteration adaptive dynamic programming: Convergence analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 6, pp. 875–891, Jun. 2018.

[38] B. Luo, D. Liu, T. Huang, and J. Liu, "Output tracking control based on adaptive dynamic programming with multistep policy evaluation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 10, pp. 2155–2165, Oct. 2019.

[39] B. Luo, Y. Yang, H.-N. Wu, and T. Huang, "Balancing value iteration and policy iteration for discrete-time control," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 3948–3958, Nov. 2020.

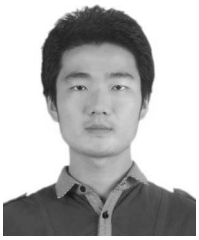
[40] H. Li and D. Liu, "Optimal control for discrete-time affine non-linear systems using general value iteration," *IET Control Theory Appl.*, vol. 6, no. 18, pp. 2725–2736, 2012.
[41] Q. Wei, D. Liu, and Y. Xu, "Neuro-optimal tracking control for a class of discrete-time nonlinear systems via generalized value iteration adaptive dynamic programming approach," *Soft Comput.*, vol. 20, no. 2, pp. 697–706, 2016.
[42] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.
[43] D. Liu, Q. Wei, and P. Yan, "Generalized policy iteration adaptive dynamic programming for discrete-time nonlinear systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 12, pp. 1577–1591, Dec. 2015.
[44] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/
[45] B. Jonkman and J. Jonkman, *FAST v8.16.00a-bjj*, Nat. Renewable Energy Lab., Golden, CO, USA, 2016.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: arXiv:1412.6980.
[47] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1–43, 2017.
[48] B. J. Jonkman and L. Kilcher, "TurbSim user's guide: Version 1.06.00," Nat. Renewable Energy Lab., U.S. Dept. Energy, Golden, CO, USA, 2012. [Online]. Available: https://www.nrel.gov/wind/nwtc/assets/pdfs/turbsim.pdf

Jincheng Zhang received the B.S. degree and the M.S. degree in mechanical engineering from Tsinghua University, Beijing, China, in 2015 and 2018, respectively, and the Diplome d'Ingenieur degree from CentraleSupélec, Gif-sur-Yvette, France, in 2018. He is currently pursuing the Ph.D. degree with the School of Engineering, University of Warwick, Coventry, U.K.

He is a Marie Curie Early Stage Researcher with the School of Engineering, University of Warwick. His research interests include machine learning-based approach for engineering applications, control of wind turbines and wind farms, uncertainty quantification of engineering systems, and computational fluid dynamics.

Xiaowei Zhao received the Ph.D. degree in control theory from Imperial College London, London, U.K., in 2010.

He worked as a Postdoctoral Researcher with the University of Oxford, Oxford, U.K., in 2013. He is a Professor of Control Engineering and an EPSRC Fellow with the School of Engineering, University of Warwick, Coventry, U.K. His main research areas are control theory with applications on offshore renewable energy systems, local smart energy systems, and autonomous systems.



Xing Wei received the master's degree in control systems from Imperial College London, London, U.K. He is currently pursuing the Ph.D. degree with the School of Engineering, University of Warwick, Coventry, U.K.

His research interests include vibration control of offshore wind turbine platform and aeroelastic control of wind turbine blades.