Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Data-driven Structural Control of Monopile Wind Turbine Towers Based on Machine Learning ⋆

Jincheng Zhang * **Xiaowei Zhao** * **Xing Wei** *

* *School of Engineering, University of Warwick, Coventry, UK (e-mail: {jincheng.zhang, xiaowei.zhao, X.Wei.4}@warwick.ac.uk)*

Abstract: This paper studies the data-driven structural control of monopile wind turbine towers based on machine learning approach, by using an active tuned mass damper (TMD) located in the nacelle. The adaptive dynamic programming (ADP) approach is employed to obtain the optimal controller which is derived on the modern large-scale machine learning platform Tensorflow. The proposed network structure includes three simple three-layer neural networks (NNs), i.e. a plant network, a critic network, and an action network. The plant network is used to capture the fully nonlinear dynamics of the structural system while the action network is used to approximate the optimal controller. Their training requires the gradient information flowing through the whole network. The automatic differentiation is used in this paper for all the gradient derivations, which greatly improves the employed ADP algorithm's ability in solving complex practical problems. The simulation results of structural control of monopile turbine towers show that on average the active TMD achieves 15% performance improvement on tower fatigue load reduction over a passive TMD, with small active power consumption (less than 0.24% of the turbine's nominal power production). Besides, the controller design considers the trade-off between control performance and power consumption.

Keywords: Adaptive Dynamic Programming, Reinforcement Learning, Neural Networks, Floating Wind Turbine, Active Structural Control

1. INTRODUCTION

Wind energy has received a lot of research attention in the past decades. As land-based wind turbines have many installation limitations, more and more offshore wind turbines are being constructed, which can also take advantage of the higher wind quality at sea. Offshore wind turbines can be classified as fix-bottom ones and floating ones according to their foundation type. Nowadays, most offshore wind turbines are fix-bottom ones, among which the monopile ones are most popular. Thus this paper focuses on the structural control of monopile wind turbine towers.

The pitch control (Leithead et al. (2004); Soltani et al. (2011)) and torque control (Zhao and Weiss (2014); Zhang et al. (2014)) are widely used for the vibration reduction of wind turbine towers, but they may interfere with the nominal power generation. Therefore, various structural control techniques have been proposed, including Tuned Mass Dampers (TMDs)(Tong et al. (2017); Lackner and Rotea (2011a); Stewart and Lackner (2013)) and Tuned Liquid Dampers (TLDs)(Colwell and Basu (2009)). The performance of a TMD can be further improved by adding an active force control to it, which is referred as Hybrid Mass Dampers (HMDs). This has been studied in the area of floating wind turbines with an HMD installed in the nacelle (Lackner and Rotea (2011b); Stewart and Lackner (2011); Hu and He (2017)), and in the floating platform (Li and Gao (2016)). However, in these works, the controller designs were all based on linear models, thus the resulting controllers worked well only when the structural system was not far from equilibrium. In order to improve control performance, the control strategy should take account of the nonlinear dynamics of the structural system. In this work, the adaptive dynamic programming (ADP) approach is employed, with the fully nonlinear dynamics of the structural system captured by artificial neural networks (NNs).

Approximate Dynamic Programming, also known as Adaptive Critic Design, Reinforcement Learning, Adaptive Dynamic Programming, is a powerful tool for optimal control problems. It was originally proposed by Webos (Werbos (1990, 1992)) and recently has caught a lot of attention for the control of both Continuous-Time (CT) and Discrete-Time (DT) systems (Liu et al. (2017, 2014); Lewis and Liu (2013); Lewis and Vrabie (2009); Jiang and Jiang (2017)). There are mainly two types of iterative ADP algorithms, Value Iteration Approximate Dynamic Programming (VI-ADP) and Policy Iteration Approximate Dynamic Programming (PI-ADP). The VI-ADP approach needs a well-designed initial value function to guarantee the stability of the iteration process while the PI-ADP approach requires an admissible initial control law. In this work, the PI-ADP approach is employed, as the open-loop structural system

⋆ This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 765579.

(including both the monopile wind turbine and the TMD) is stable.

The PI-ADP algorithm is realized by NNs. The employed algorithm includes three networks, i.e. an action network, a critic network, and a plant network, where the plant network is used to capture the fully nonlinear dynamics of the structural system and the action network is used to approximate the optimal controller. The NN training requires the gradient information flowing through all the three networks, which is tackled by automatic differentiation in this work. 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 will become infeasible with complex network structure. Thus the use of automatic differentiation in this work greatly improves the employed ADP algorithm's ability in solving complex practical problems. Besides, the NN structure is implemented on the large-scale machine learning platform Tensorflow (Abadi et al. (2015)), which makes the NN training in this work very efficient, especially with the use of GPU. The NREL (National Renewable Energy Laboratory) 5-MW baseline monopile wind turbine model (Jonkman et al. (2009)) is used in this study and an HMD is installed in the turbine nacelle to suppress the tower vibration in the fore-aft direction. The simulation is done by the modified version of FAST (Fatigue, Aerodynamics, Structures, and Turbulence) code (Jonkman et al. (2005)), which is called FAST-SC (Lackner and Rotea (2011a)). The plant network is trained based on the data generated by FAST-SC. After training the plant network, a series of ADP controllers are obtained by varying the penalty term in the action network training, which considers the trade-off between the control performance and power consumption.

The remaining part of this paper is organized as follows: the ADP approach is described in Section 2, where the network structure is presented. In Section 3, the machine learning based structural control design is described. The simulation results are given in Section 4 and the conclusion is drawn in Section 5.

2. ADAPTIVE DYNAMIC PROGRAMMING APPROACH

2.1 Problem formulation

Consider a general nonlinear system

$$x_{k+1} = F(x_k, u_k),\ k = 0, 1, 2, ... \tag{1}$$

where $x_k \in \mathscr{R}^n$ and $u_k \in \mathscr{R}^m$ are the n-dimensional state variable and the m-dimensional control variable at time step k, respectively. It is assumed hereby that $F(0, 0) = 0$ and $F(x_k, u_k)$ is Lipschiz continuous on a compact set Ω. Denote the sequence of control variables as $\bar{u}_k = \{u_k, u_{k+1}, u_{k+2}, ...\}$, then the cost function for the state x_0 under the control $\bar{u}_0$ is expressed as

$$J(x_0, \bar{u}_0) = \sum_{k=0}^{\infty} U(x_k, u_k) \tag{2}$$

where $U(x_k, u_k)$ is the utility function. It is a positive-definite function of x_k and u_k. Here we focus on state-feedback control thus an arbitrary control law can be expressed as a function of state variables

$$u_k = \mu(x_k). \tag{3}$$

The cost function for the state x_0 under the control law μ is expressed as

$$J^{\mu}(x_0) = \sum_{k=0}^{\infty} U(x_k, \mu(x_k)). \tag{4}$$

The optimal cost function is then given as

$$J^*(x_0) = \inf_{\mu}(J^{\mu}(x_0)). \tag{5}$$

According to Bellman's principle of optimality we get

$$J^*(x_k) = \min_{u_k}\{U(x_k, u_k) + J^*(F(x_k, u_k))\}, \forall x_k \in \Omega \tag{6}$$

and the corresponding optimal control law

$$\mu^*(x_k) = \arg\min_{u_k}\{U(x_k, u_k) + J^*(F(x_k, u_k))\}, \forall x_k \in \Omega. \tag{7}$$

2.2 Policy iteration algorithm

The PI-ADP algorithm begins with an admissible control law μ_0 (i.e. a control law that is continuous on Ω, $\mu_0(0) = 0$, and $J^{\mu_0}(x_0) < \infty, \forall x_0 \in \Omega$), and then obtain 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

$$V_i(x_k) = U(x_k, \mu_i(x_k)) + V_i(F(x_k, \mu_i(x_k))), \forall x_k \in \Omega. \tag{8}$$

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

$$\mu_{i+1}(x_k) = \arg\min_{u_k}\{U(x_k, u_k) + V_i(F(x_k, u_k))\}, \forall x_k \in \Omega. \tag{9}$$

Through the iteration process ($\mu_0 \rightarrow V_0 \rightarrow \mu_1 \rightarrow V_1 \rightarrow \mu_2 \rightarrow ...V_{N-1} \rightarrow \mu_N$), 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 (Liu et al. (2014)), where an admissible initial control law is required to guarantee the convergence and stability of the algorithm.

2.3 Neural network implementation

The system to be controlled, the value function and the control law are all approximated by NNs, which are denoted as plant network, critic network, and action network hereafter. A basic illustration of a NN with only one hidden-layer is given in Fig. 1, and the corresponding input-output relation can be expressed as

$$\mathbf{y} = \mathbf{w}_2\sigma(\mathbf{w}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2. \tag{10}$$

N_1, N_2, and N_3 in Fig. 1 represent the input dimension, the hidden-layer neuron number and the output dimension. $\mathbf{w}_1$ ($\mathbf{b}_1$) and $\mathbf{w}_2$ ($\mathbf{b}_2$) are the weight matrix (bias terms) of the first layer and the output layer, which are all updated during NN training in this work. The hyperbolic tangent function is used as the activation function σ throughout this work.

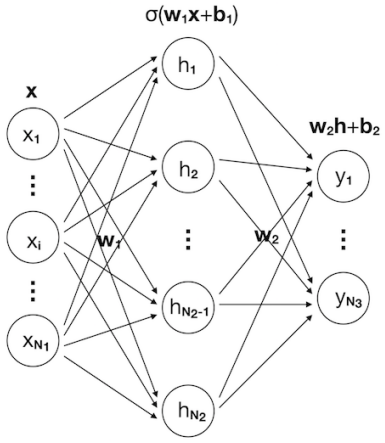

Fig. 1. The illustration of an artificial neural network with one hidden-layer.

The whole network used in this work is illustrated in Fig. 2, where each network (i.e. the plant network, the action network, and the critic network) has the same structure as the one shown in Fig. 1. It is a common practice to standardize the data in NN training, thus here we have designed the networks in order to feed standardized data for plant network training and critic network training. 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

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

where $m(\mathbf{d})$ and $s(\mathbf{d})$ represent the mean and standard deviation of the dataset $\mathbf{d}$.

By feeding the standardized state $\mathbf{x}_k^{std}$, standardized action $\mathbf{u}_k^{std}$ and the target of the standardized state change $\mathbf{dx}_k^{std*}$ to the plant network, it is trained to minimize the mean squared error (MSE) between the target $\mathbf{dx}_k^{std*}$ and the network output $\mathbf{dx}_k^{std}$. After the training process, the prediction of the next state $\mathbf{x}_{k+1}$ is done by first transforming $\mathbf{x}_k$, $\mathbf{u}_k$ into $\mathbf{x}_k^{std}$, $\mathbf{u}_k^{std}$, then obtaining $\mathbf{dx}_k^{std}$ through the plant network, transforming $\mathbf{dx}_k^{std}$ into $\mathbf{dx}_k$, and finally adding $\mathbf{x}_k$ and $\mathbf{dx}_k$. The reason for predicting the state change instead of the state variable by plant network is that it can greatly increase the prediction accuracy, since the state change is usually subtle compared to the state variable. Once the plant network is trained, we can proceed to design the iterative controller by critic training and action training.

For the critic training, the value function is approximated directly by

$$V_i(x_k) = \sum_{j=0}^{N} U(x_{k+j}, \mu_i(x_{k+j})) \tag{12}$$

where N is a large number. Then the training is done by minimising the MSE of the target $V_i(x_k)$ and the critic network output. After the critic training, the action network is trained such that it minimizes the value function

$$V(\mathbf{x}_k) = U(\mathbf{x}_k, \mathbf{u}_k) + V(\mathbf{x}_{k+1}). \tag{13}$$

In practice, the minimization procedure is deemed as completed when the mean square of the value function changes very little (less than a prescribed threshold) with further training. The critic training and the action training are done iteratively until they converge.

3. HMD CONTROLLER DESIGN

3.1 NREL 5MW Monopile Wind Turbine Model within FAST-SC code

The NREL 5MW monopile wind turbine model within FAST-SC code is employed in this paper. The HMD is installed in the turbine nacelle and designed to move in the fore-aft direction, and a stroke limitation of ±7m is imposed. The damping coefficient, stiffness coefficient and HMD mass are set as 7518N/(m/s), 61514N/m and 20000kg, which are the optimal parameters reported in (Tong et al. (2017)). The simulations are carried out in FAST-SC with all the degrees of freedom (DOFs) enabled except the nacelle yaw DOF.

3.2 Plant network training

The structural system can be characterized by the primary state variable $\mathbf{x}$

$$\mathbf{x} = [x_{hmd}, u_{hmd}, x_{tt}, u_{tt}] \tag{14}$$

where x_{hmd}, u_{hmd}, x_{tt} and u_{tt} represent the HMD displacement, HMD velocity, tower top displacement and tower-top velocity, respectively. The training dataset is generated by running the monopile turbine model for a certain amount of time. From these time series data, a set of training samples are extracted, with each sample consisting of state variables at time k, action variables at time k and state variables at time $k+1$.

In this work, a 1200-second simulation with a time interval of $0.0125s$ is carried out under the excitation of random HMD force with a time step of $0.1s$ and within the interval $[-500, 500]$kN. The training dataset is obtained by eliminating the first 5-second time series and extracting one data point with a time interval of $0.1s$. Thus 11950 data points are collected. In each training sample, x_{k+1} is set to have a time difference of $0.05s$ with the current state x_k.

After generating the training dataset, a plant network with the hidden-layer neuron number $N_2 = 15$ is constructed and is trained with Adam optimization algorithm (Kingma and Ba (2014)). The learning rate is set as 0.001 and the training error is set as 1.5×10^{-3}. The trained plant network is used to mimic the structural system. In order to assess the accuracy of the trained plant network, a comparison of the FAST-SC simulation results and the plant network calculations is given in Fig. 3, including the time series of the random HMD force and the comparison of state variables. Both calculations are based only on the same initial condition at $t = 5s$ and the same test perturbation force which has not been used during the plant network training. A perfect match is observed for the whole period. It is concluded that the plant network has captured the fully nonlinear dynamics of the structural system.

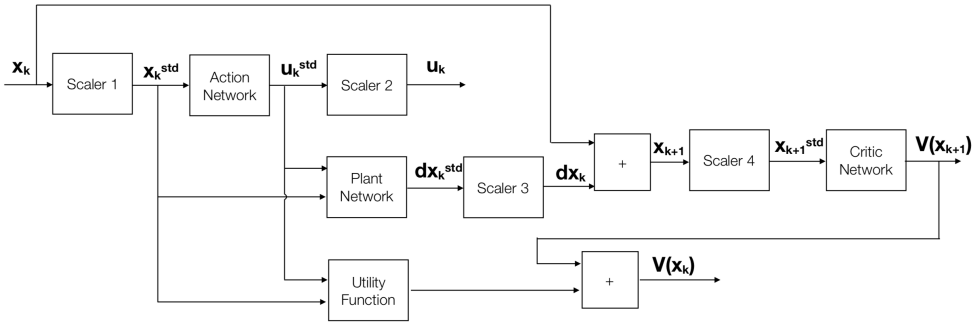

Fig. 2. The whole network structure including the plant network, the action network and the critic network.

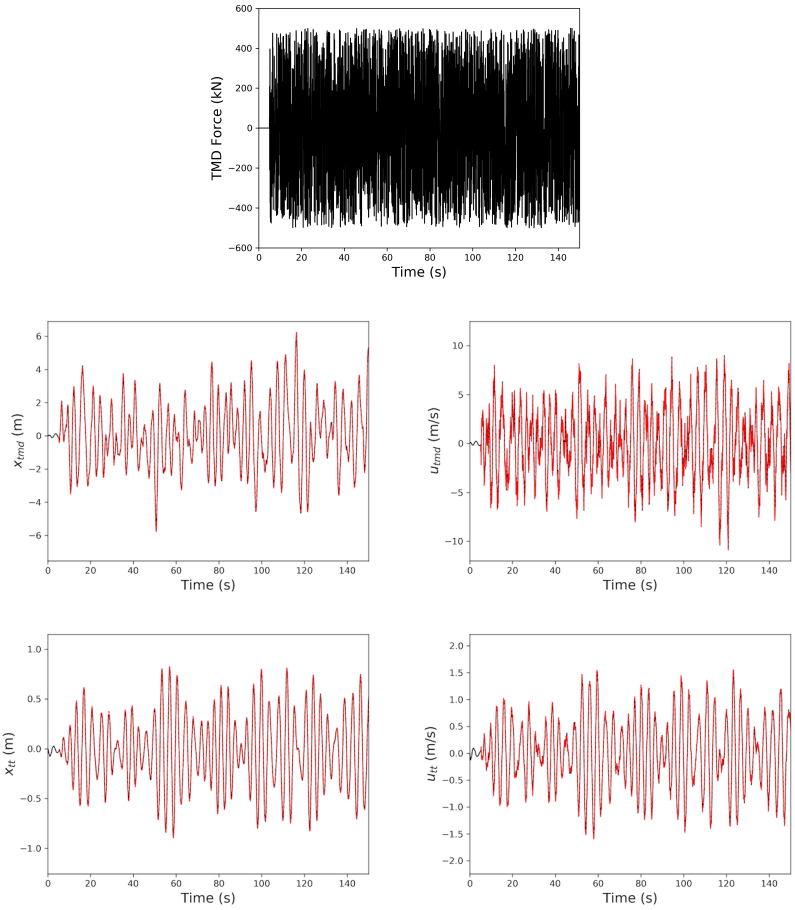

Fig. 3. The comparison of the FAST-SC simulation results (solid line) and the plant network calculations (dashed line). The sub-figures above show the time series of the HMD force for testing the trained plant network, HMD displacement, HMD velocity, Tower top displacement, and Tower top velocity.

3.3 Action-Critic network training

After training the plant network, the training of the action network and the critic network are conducted iteratively. The hidden-layer neuron number of both networks is set as 15, 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 critic network is set as 10^{-3} and the training convergence criterion of action network is set as 10^{-5}. The action network is initialized by very small random weights as $\mu_0 = 0$ is an admissible control law for the structural system. The value function is approximated by equation (12) and N is set as 5000. The utility function still needs to be specified in order to train the networks. In this paper, the utility function is chosen as

$$U(\mathbf{x}_k^{std}, \mathbf{u}_k^{std}) = (\mathbf{x}_k^{std})^T \cdot A_u \cdot (\mathbf{x}_k^{std}) + B_u \cdot (\mathbf{u}_k^{std})^2 \quad (15)$$

where $A_u = 10^{-4} \times diag(1, 1, 25, 25)$. A number of values are chosen for B_u, as reported in Table 1, in order to investigate the compromise between the active control force and the control performance.

4. SIMULATION STUDY

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

The turbulent wind is generated by TurbSim (Jonkman and Kilcher (2012)) using IEC von Karman turbulence model with the power law exponent set as 0.14 and the turbulence intensity set as 15%. The wave condition is generated based on the JONSWAP spectrum with the peak-spectral period of the incident waves set as 12.4s. Four simulations are carried out. For the first two simulations (S1 and S2), the main wind speed is $18m/s$ and the significant wave height is $3.5m$. For the last two simulations (S3 and S4), the main wind speed is $10m/s$ and the significant wave height is $2.0m$. The wind inputs for S1 and S2 (or S3 and S4) are generated by different random seeds.

The MLife code (Hayman and Buhl Jr (2012)) provided by NREL is employed for computing the damage equivalent load (DEQL). The frequency of DEQL is set to be 1 Hz and the Wohler exponent is set as 3. The main simulation results are given here, including the calculations with no TMD, passive TMD, and HMD using a series of ADP controllers (denoted as ADP1-ADP5). The DEQL based on the bending moment at the tower base and the HMD power consumption are listed for all the four simulations (S1, S2, S3, S4) in Table 1. In addition, the fatigue damage reductions over the cases with passive TMD are also reported (the ones in the brackets in Table 1).

The tower top displacements for the simulations S1-S4 are given in Fig. 4. As can be seen, the ADP-based HMD has clearly stabilized the turbine tower. On average the DEQL is reduced by 13.5% from the passive case with an average power consumption of 11.8kW using the controller ADP5 in the cases of S1 and S2. As for the cases of S3 and S4, the DEQL is reduced by 16.3% on average from the passive case with an average power consumption of 2.9kW.

By changing the penalty coefficient B_u related to the HMD force magnitude in the utility function, a set of controllers are obtained. As can been seen from Table 1, the control performance increases monotonically with the active power consumption. Due to the relatively small stroke limitation and relatively small mass of the HMD, the power consumption of these controllers are small. For example the largest power consumption (corresponding to ADP5) is just about 0.24% of the nominal power of the turbine (but with a considerable improvement in fatigue load reduction compared with the passive TMD). We mention that the control performance can be further improved by increasing the stroke limitation and/or HMD mass.

5. CONCLUSION

The data-driven structural control of monopile wind turbine towers has been investigated, by using an HMD installed in the nacelle. The ADP approach was employed to obtain the optimal controller, which was derived on the modern large-scale machine learning platform Tensorflow. The use of automatic differentiation in this work greatly improves the employed ADP algorithms' ability in solving complex industrial problems. In addition, our controller design allows considering the trade-off between control performance and power consumption. The simulation results showed that the HMD achieved an average 15% performance improvement on the fatigue load reduction compared with a passive TMD, with very small active power consumption (less than 0.24% of the turbine's nominal power production).

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Colwell, S. and Basu, B. (2009). Tuned liquid column dampers in offshore wind turbines for structural control. *Engineering Structures*, 31(2), 358–368.

Hayman, G. and Buhl Jr, M. (2012). Mlife users guide for version 1.00. *National Renewable Energy Laboratory, Golden, CO*, 74(75), 112.

Hu, Y. and He, E. (2017). Active structural control of a floating wind turbine with a stroke-limited hybrid mass damper. *Journal of Sound and Vibration*, 410, 447–472.

Jiang, Y. and Jiang, Z.P. (2017). *Robust Adaptive Dynamic Programming*. John Wiley & Sons.

Jonkman, B. and Kilcher, L. (2012). Turbsim user's guide: Version 1.06. 00, tech. rep.

Jonkman, J., Butterfield, S., Musial, W., and Scott, G. (2009). Definition of a 5-mw reference wind turbine for offshore system development. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

Jonkman, J.M., Buhl Jr, M.L., et al. (2005). Fast user's guide. *National Renewable Energy Laboratory, Golden, CO, Technical Report No. NREL/EL-500-38230*.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lackner, M.A. and Rotea, M.A. (2011a). Passive structural control of offshore wind turbines. *Wind energy*, 14(3), 373–388.

Lackner, M.A. and Rotea, M.A. (2011b). Structural control of floating wind turbines. *Mechatronics*, 21(4), 704–719.

Leithead, W., Dominguez, S., and Spruce, C. (2004). Analysis of tower/blade interaction in the cancellation of the tower fore-aft mode via control. In *European Wind Energy Conference 2004*.

Lewis, F.L. and Liu, D. (2013). *Reinforcement learning and approximate dynamic programming for feedback control*, volume 17. John Wiley & Sons.

Lewis, F.L. and Vrabie, D. (2009). Reinforcement learning and adaptive dynamic programming for feedback con-

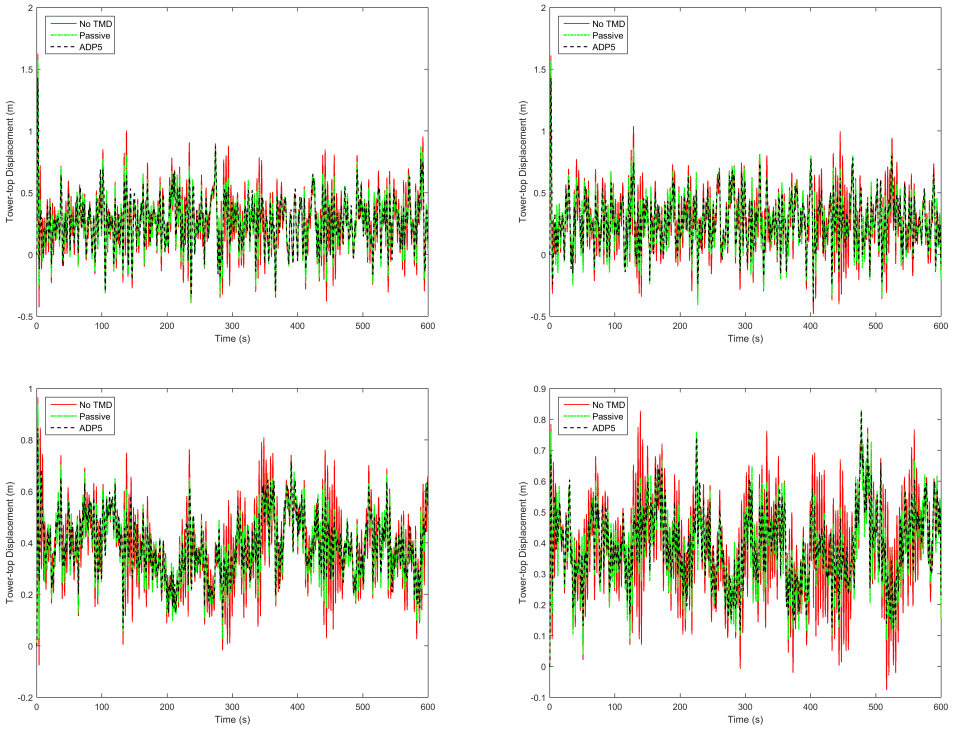

Fig. 4. The simulation results including the ones with no TMD, passive TMD, and HMD using the controller ADP5. The four cases are shown in different sub-figures (S1: upper left; S2: upper right; S3: lower left; S4: lower right).

Table 1. Main simulation results

Controller	No TMD	Passive	ADP1	ADP2	ADP3	ADP4	ADP5
B_u ($\times 10^{-3}$)	-	-	100	50	25	10	5
S1:HMD Power(kW)	-	-	0.133	0.84	2.396	6.248	11.673
S1:DEQL(kN·m)	33822	28487	27459(3.6%)	26915(5.5%)	26258(7.8%)	25254(11.3%)	24567(13.8%)
S2:HMD Power(kW)	-	-	0.160	0.927	2.582	6.504	11.912
S2:DEQL(kN·m)	34408	28995	27753(4.3%)	27201(6.2%)	26560(8.4%)	25641(11.6%)	25157(13.2%)
S3:HMD Power(kW)	-	-	0.0161	0.2007	0.6085	1.6378	3.1335
S3:DEQL(kN·m)	19600	15048	14186(5.7%)	13855(7.9%)	13472(10.5%)	12913(14.2%)	12510(16.9%)
S4:HMD Power(kW)	-	-	0.0219	0.2070	0.5937	1.4685	2.6579
S4:DEQL(kN·m)	20191	14958	14181(5.2%)	13847(7.4%)	13466(10.0%)	12971(13.3%)	12614(15.7%)

trol. *IEEE circuits and systems magazine*, 9(3).

Li, X. and Gao, H. (2016). Load mitigation for a floating wind turbine via generalized h_∞ structural control. *IEEE Transactions on Industrial Electronics*, 63(1), 332–342.

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

Liu, D., Wei, Q., et al. (2014). Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. *IEEE Trans. Neural Netw. Learning Syst.*, 25(3), 621–634.

Soltani, M., Wisniewski, R., Brath, P., and Boyd, S. (2011). Load reduction of wind turbines using receding horizon control. In *2011 ieee international conference on control applications (cca)*, 852–857. IEEE.

Stewart, G. and Lackner, M. (2013). Offshore wind turbine load reduction employing optimal passive tuned mass damping systems. *IEEE Transactions on Control Systems Technology*, 21(4), 1090–1104.

Stewart, G.M. and Lackner, M.A. (2011). The effect of actuator dynamics on active structural control of offshore wind turbines. *Engineering Structures*, 33(5), 1807–1816.

Tong, X., Zhao, X., and Zhao, S. (2017). Load reduction of a monopile wind turbine tower using optimal tuned mass dampers. *International Journal of Control*, 90(7), 1283–1298.

Werbos, P.J. (1990). A menu of designs for reinforcement learning over time. *Neural networks for control*, 67–95.

Werbos, P.J. (1992). Approximate dynamic programming for real time control and neural modeling. *Handbook of intelligent control: neural, fuzzy and adaptive approaches*, 493–525.

Zhang, Z., Nielsen, S., Blaabjerg, F., and Zhou, D. (2014). Dynamics and control of lateral tower vibrations in offshore wind turbines by means of active generator torque. *Energies*, 7(11), 7746–7772.

Zhao, X. and Weiss, G. (2014). Strong stabilisation of a wind turbine tower model in the plane of the turbine blades. *International Journal of Control*, 87, 2027–2034.