Deep Reinforcement Learning-Based Pitch Control for Floating Offshore Wind Turbines

Flavie Didier[1], Salah Laghrouche[2] and Daniel Depernet[3]

***Abstract*— The floating offshore wind turbine (FOWT) technology has great energy potential, however, minimizing the movement of the structure, under the combined effect of wind and waves, while ensuring maximum power extraction on all operating ranges remains a challenge. This paper proposes the design of a deep reinforcement learning (DRL) controller for FOWTs in the operating area III. To our knowledge, this is the first time that DRL-based control approach is used for this application. The proposed DRL controller is based on trust region policy optimization (TRPO) algorithm, composed of two neural networks, the actor and the critic networks, for the learning of the optimal control law. Simulation results and comparison study are provided to validate the proposed DRL controller for the 5-MW baseline ITI Barge wind turbine model on OpenFAST.**

I. INTRODUCTION

In order to reduce greenhouse gas emissions and thus reduce the consumption of fossil fuels, the energy transition to renewable energies is a promising alternative. Particularly, floating offshore wind turbines (FOWTs) have been booming in recent years. This emerging technology shows great potential, allowing the placement of wind farms farther from the coast, making them less visible and implantable in greater numbers. However, this industry faces multiple challenges, with one of the main issues being the development of robust, adaptive, and intelligent control systems that minimize structural movements while ensuring maximum power extraction in all areas of the wind turbine's operation. Indeed, the operating range of the wind turbine is divided into three operating areas depending on the incoming wind speed. In area I, the wind speed is lower than the starting speed of the wind turbine, no electrical power is produced. In area II, the wind speed is between the starting speed and the rated speed, the control objective is then to extract the maximum power from the wind. In area III, the wind speed is higher than the rated speed, the objective is then to regulate the generated power by limiting it to the rated power.

Traditional control methods, such as gain-scheduled proportional integral control (GSPI), from fixed wind power systems suffer from performance limitations when being applied to FOWTs. They are sensitive to disturbances, and are therefore not the most suitable for the control of FOWT which is a strongly nonlinear system and which is subjected to strong disturbances such as wind and waves. A GSPI controller for FOWT in operating area III is implemented in [1], where the movement of the platform pitch is minimized by keeping the blade pitch actuator's frequency lower than the platform's resonance frequency. Another GSPI controller to limit the negative damping phenomenon was used in [2] adjusting this time the rotor speed according to the blade angle activity. This configuration is now considered as the reference controller for FOWTs, on which newly developed controllers are usually tested. Moreover, this reference controller has been analyzed for different types of platforms [3].

Several works propose to use control design approaches based on the multi-input multi-output (MIMO) models that consider the couplings of the outputs to be controlled. In general, the realization of these control methods is based on analytical models of the system dynamics. To date, the most successful and exploitable works for control are those of Betti [4], and Sandner [5]. These nonlinear modeling approaches are both based on restrictive assumptions in order to obtain simplified nonlinear models. These models are then linearized around an operating point for the implementation of linear control systems. However, these controllers suffer from a degradation of their performance when changing the operating point. The main controllers proposed in the recent literature are, linear quadratic regulator (LQR), H_∞ controller, linear parameter-varying control (LPV) and model predictive control (MPC). The LQR controllers can be used to reduce the platform movement. Collective blade pitch controllers (LQR-CBPC) and individual blade pitch controllers (LQR-IBPC) have been designed in [6], with LQR-IBPC providing better performance in terms of rotor speed and power regulation but the platform pitching motion remains significant. A possible improvement is the LQR-IBPC controller with disturbance adaptation control, which ensure better speed and power regulation, thanks to its ability to estimate the wind and thus minimizes its action on the system [7]. Similarly, an H_∞-CPB controller has been developed to manage the motion of the platform, and for power regulation and reduction of associated loads [8]. A controller with gain sequencing (GS) and H_∞ output strategy has led to an improvement in terms of tower load and rotor speed regulation [9]. In [10], a state feedback LPV and an output feedback LQR controller both based on GS are proposed to regulate the power and minimize the structural

*This work was supported by ANR CREATIF Project

[1]Flavie Didier is with Franche-Comté Electronique Mécanique,Thermique et Optique - Sciences et Technologies, Research Institute, Besançon, 15B avenue des Montboucons 25030, France flavie.didier@utbm.fr

[2]Salah Laghrouche is with Franche-Comté Electronique Mécanique,Thermique et Optique - Sciences et Technologies, Research Institute, Besançon, 15B avenue des Montboucons salah.laghrouche@utbm.fr

[2]Daniel Depernet is with Franche-Comté Electronique Mécanique,Thermique et Optique - Sciences et Technologies, Research Institute, Besançon, 15B avenue des Montboucons daniel.depernet@utbm.fr

load. They showed better performance compared to the reference controller. A CBP-LPV controller has also been proposed in [11].

With the increase in computing and big data processing capacities, the implementation of data-based controllers with artificial intelligence techniques is an approach to overcome the difficulties and limitations linked to nonlinear modeling. Specifically, deep reinforcement learning (DRL) seems to be a promising technique for FOWTs control as it does not require a model of the dynamics but proceeds by repeated interaction with the environment through trial and error. DRL methods are thus highly suitable for complex systems with little knowledge of the model or with a model that is difficult to formulate. This approach has the ability to generalize to new situations, which could allow control over all areas of the wind turbine operation.

Reinforcement Learning (RL) methods are based on the dynamic programming principle, introduced by Bellman with the optimal control theory [12]. The aim of RL algorithms [13] is to learn from interaction the optimal policy that would maximize a numerical reward. More specifically, it involves the utilization of an agent capable of learning a policy that can enhance the performance of a system by engaging in direct interactions with its environment, through observations s, actions a, and rewards r. The observations correspond to the system states available to the agent. The goal is then to find a policy giving the optimal action to maximize the expected cumulative reward. This cumulative reward, as determined by an action value function, provides an estimation of the desirability for the agent to be in a given state and execute a given action for this state. In DRL, neural networks are used to represent the policy and/or action value function, thus enable the agent to generalize and apply the policy to new inexperienced states from the previously learned state values. Applied to control problems [14], the policy corresponds to the control law and thus, learning the optimal policy through the interaction between the agent and the environment is equivalent to learning the optimal control law.

DRL algorithms have been applied with success to a variety of control problems, however few works exist for its application on wind turbines power systems. For the fixed wind turbine, an MPPT controller for a variable speed wind power conversion system based on a Q-learning algorithm is proposed in [15]. The learning of an optimal mapping from states to actions is performed online by updating the action values according to future rewards. This approach does not require any knowledge of the wind turbine parameters, since the agent is able to learn by direct interaction with the environment. In [16] MPPT controller for wind power was also studied this time with a DQN algorithm, in order to learn the optimal correlation between electrical power output and rotor speed. For the FOWT system, the optimal control law is derived using an adaptive dynamic programming (ADP) algorithm in [17]. The ADP controller shows great performances especially for extreme conditions.

In this study, we apply a deep reinforcement learning-based approach for pitch control in operating area III of floating wind turbines. Among the operating areas, the third one is the most delicate for control due to the phenomenon of negative damping, which deteriorate the stability of the system [1]. A trust region policy optimization (TRPO) algorithm is employed, which considered the nonlinear dynamics of the system in the control design process, and thus the DRL controller can achieve rated power tracking while stabilizing the structure.

This paper is structured as follows: Section II provides the working environment for the development of a DRL controller. In Section III, TRPO algorithm and the implementation of the DRL controller are introduced. Simulation results are presented in Section IV. Finally, a conclusion and the perspectives on future work are given in Section V.

II. Definition of the Environment

In a DRL learning framework, the environment corresponds to the system to be controlled as well as the disturbances applied to it. This part introduced the FOWT system and the disturbances employed as the learning environment in which the DRL controller interacts.

A. FOWT system and disturbances

The National Renewable Energy Laboratory (NREL) model of the 5-MW baseline wind turbine mounted on ITI Energy's barge [18] is considered for the synthesis of the DRL controller. The FOWT is implemented in the multi-physics tool OpenFAST [19], an open source version of FAST developed by NREL [20], allowing the simulation of coupled dynamic response of wind turbines.

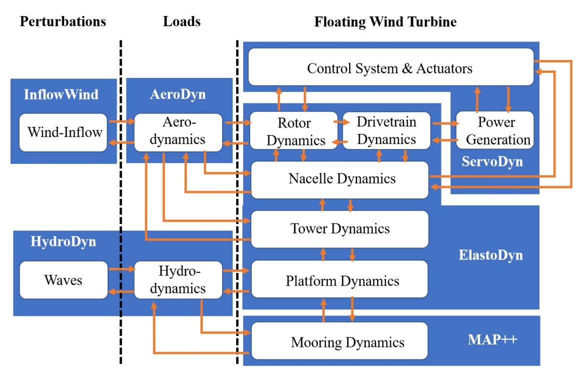

Fig. 1. Architecture OpenFAST

The architecture of the OpenFAST code is illustrated in Fig.1. The floating wind turbine is modeled by a servo-elastic structural model which is coupled to external aerodynamic, hydrodynamic and mooring lines forces. In this paper, all DOFs of the FOWT model are enabled, except the nacelle yaw DOF.

Two disturbances signals are applied to the FOWT, a wind signal and a wave signal. The wind field characteristics have to respect the area III of operation of the FOWT system, and is defined with mean wind speed of 18 m/s and a turbulence intensity of 5%. TurbSim was used to generate the wind file.

Irregular waves are generated with the HydroDyn module from OpenFAST and they are based on JONSWAP spectrum.

B. Area of study

In this article, we focus on FOWT control in region III, with the collective blade pitch angle as the control variable, and thus the input variable of OpenFAST simulation. The generator torque is maintained constant at its rated value 43093.55 $N.m$ and the nacelle yaw angle is fixed against the wind. Output variables of OpenFAST simulation are:

- blade pitch angle β,
- blade pitch angular velocity $\dot{\beta}$,
- generator rotational speed ω_g,
- platform pitch angle θ_y,
- platform pitch velocity $\dot{\theta}_y$,

These output variables are normalized within the interval $[-1;1]$ to feed the neural networks within the DRL Controller.

III. DRL CONTROLLER

The DRL controller is a reinforcement learning agent who is trained from the interaction with the environment in order to reduce the platform movement while ensuring the tracking of the rated electrical power of the 5-MW turbine. Therefore, the reinforcement learning algorithm named trust region policy optimization (TRPO) with neural networks representation [21] for the value function and for the policy is employed. A brief description of TRPO algorithm, its implementation for FOWTs and its specific training details are presented in the following section.

A. TRPO Algorithm

The goal of TRPO algorithm is to learn the value of each state s from the state space S, and to learn the optimal policy, which allow the best action a from the action space A to be taken by the agent given the state.

The state value is defined as the predicted cumulative discounted long-term reward received by the agent for being in that state. The function $V(s_t)$ that return the state value for the state s from the state space S at time step t is given as:

$$V(s_t) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right] \tag{1}$$

Where, $r: S \to \mathbb{R}$ is the reward function, $\gamma \in [0,1]$ is the discount factor that determines the present value of the rewards received by the agent in the future. The closer the factor is to 0, the more immediate rewards prevail over future rewards. On the contrary, the closer the factor is to 1 the more future rewards are considered. In this paper, a discount factor of 0.99 is selected, so the agent is more far-sighted. The stochastic policy $\pi(a|s)$ corresponds to the probability of taking each action a from the space action A when in state s. Here the policy return the mean and standard deviation of the Gaussian probability distribution for each action.

In deep reinforcement learning, neural networks are used to represent the state value function and the policy. The critic network is noted $V(s,\phi)$ and the actor network is noted $\pi(a|s,\theta)$, with neural networks parameters ϕ and θ respectively. During the training of the agent (Fig.2), their parameters are updated in the direction that maximizes the reward signal r. At each time step t, an observation of the environment is given to the agent. The actor network, based on this observation, estimates the probabilities of taking each action possible for that observation and then based on the probability distribution selects an action a_t to perform. The critic network estimates the state value $V(s_t,\phi)$.The application of the action a_t modified the state of the environment. The subsequent observation s_{t+1} is given to the agent along with an immediate reward r_{t+1}. The critic parameters ϕ are then updated in the next step to minimize a critic loss function L_{critic}. The actor network parameters are also updated in the next step to minimize an actor loss function L_{actor} subject to constraint.

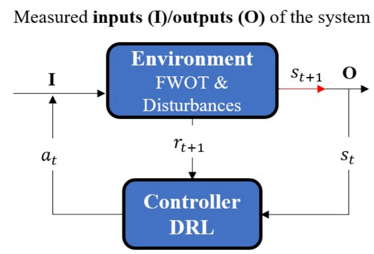

Fig. 2. Interaction Agent-Environment

Algorithm Trust Region Policy Optimization:
1: Initialize the critic $V(s,\phi)$ and the actor $\pi(a|s,\theta)$ with random parameters values ϕ and θ, respectively.
2: Generate N a sequence of experiences by following the current policy :

$$s_1, a_1, r_2, s_2, ..., s_{N-1}, a_{N-1}, r_N, s_N$$

3: For each episode step $t = 1,2,...,N$, compute the advantage function D_t given in equation (2), which is the discounted sum of temporal difference errors. Then compute the return G_t given in equation (3), which is the sum of the reward for that step and the discounted future reward:

$$D_t = \sum_{k=t}^{N} (\gamma\lambda)^{k-t}(r_t + \gamma V(s_t,\phi)) \tag{2}$$

where λ is a smoothing factor equal to 0.95 in this study.

$$G_t = D_t + V(s_t,\phi) \tag{3}$$

4: Randomly select a mini-batch of data of size M from the current set of experiences.
5: Update the critic parameters by minimizing L_{critic} across all sampled mini-batch data.

$$L_{critic}(\phi) = \frac{1}{M}\sum_{i=1}^{M}(G_t - V(s_i,\phi))^2 \tag{4}$$

6: Update the actor parameters by solving the constrained optimization problem. The actor loss function to minimize is defined as:

$$L_{actor}(\theta) = -\frac{1}{M}\sum_{i=1}^{M}\left(\frac{\pi(a_i|s_i,\theta)}{\pi(a_i|s_i,\theta_{old})}D_i + w\mathscr{L}_i(\theta,s_i)\right) \quad (5)$$

Where, $w\mathscr{L}(\theta,s)$ is the entropy loss term, with w the entropy loss weight and $\mathscr{L}(\theta,s)$ the entropy. θ_{old} corresponds to the parameters of the old policy $\pi(a|s,\theta_{old})$. The constraint on the minimization is:

$$\frac{1}{M}\sum_{i=1}^{M} D_{KL}(\theta_{old},\theta,s_i) \leq \delta \quad (6)$$

Here, D_{KL} is the Kullback-Leibler divergence between the old policy $\pi(a|s,\theta_{old})$ and the current policy $\pi(a|s,\theta)$. The limitation by δ of D_{KL} controls how much the new policy can deviate from the old one. To resolve the minimization problem with the constraint (6), a linear approximation of L_{actor} and a quadratic approximation of D_{KL} are realized thanks to the Taylor series expansion around θ. The problem is then written as:

$$\min_{\theta} L_{actor}(\theta) \approx g(\theta-\theta_{old})$$

$$with \quad g(\theta-\theta_{old}) = \nabla_{\theta} L_{actor}(\theta)|_{\theta=\theta_{old}} \cdot (\theta-\theta_{old}) \quad (7)$$

$$subject \quad to \quad \frac{1}{2}(\theta_{old}-\theta)^T H(\theta_{old}-\theta) \leq \delta \quad (8)$$

$$and \quad H = \nabla_{\theta}^2 \sum_{i=1}^{M} D_{KL}(\theta_{old},\theta,s_i)|_{\theta=\theta_{old}} \quad (9)$$

The update of the actor's parameters θ with this approximate optimization problem is:

$$\theta = \theta_{old} + a\sqrt{\frac{2\sigma}{(H^{-1}g)^T H^{-1}(H^{-1}g)}}H^{-1}g \quad (10)$$

with the coefficient a ensuring that the policy simultaneously improves its performance and satisfies the constraint given in equation (8).
By employing conjugate gradient descent as the optimization method with a KL constraint, the updated policy is constrained within a trust region relative to the current policy. This ensures that the performance does not deteriorate significantly during the optimization process.

B. Implementation

The DRL controller for FOWTs, using the TRPO algorithm with the two neural networks presented in the previous part, is implemented on Matlab/Simulink R2022a (Fig.3).

The implementation of DRL algorithms, as success or failure hinges on properly selecting observations, actions, and rewards. Here, the observation and action spaces are defined as follow. The action space is a scalar corresponding to the collective blade pitch angle β within the interval $[0;30]deg$. The observation space is defined as a 5-dimensional vector characterized by:

- the generator rotational speed ω_g,

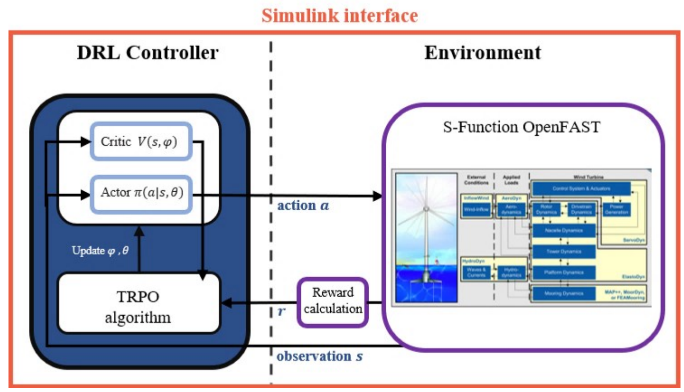

Fig. 3. Implementation of the DRL controller on Matlab/Simulink. The controller interacts with OpenFAST by receiving an observation of the system's state from OpenFAST simulation and then choose a control action from the action space. The chosen action is given as input to OpenFAST, that compute the simulation and give a new observation to the DRL controller, as well as a reward to update the actor and critic networks.

- the error between the generator rotational speed ω_g and the rated rotational speed $\omega_{g,rated}$ equal to $122.9096 rad/s$,
- the platform pitch angle θ_y,
- the platform pitch angular velocity $\dot{\theta}_y$,
- the blade pitch angle β,

$$s_t = [\omega_g, (\omega_g - \omega_{g,rated}), \theta_y, \dot{\theta}_y, \beta] \quad (11)$$

Once an action is taken, the agent receives a reward to evaluate the action. In this study, the blade pitch control aims at reducing the movement of the turbine's platform while ensuring the tracking of the rated power and thus with a minimum of actuation effort. The reward function is thus defined as:

$$r_t = -[Q_1|\omega_g - \omega_{g,rated}| + Q_2|\dot{\theta}_y| + Q_3|u_{-1}|] + G + P \quad (12)$$

Where u_{-1} is the command variable (i.e. β) at the previous time step. The Q_i factors correspond to the positive weights assigned to each of the reward terms. G and P, are both scalar, a positive gain if the error $(\omega_g - \omega_{g,rated})$ is below a defined threshold, and a penalization if ω_g is outside its feasible working area, respectively.

The first term penalizes the deviation from the rated generator speed in order to stabilize the FOWT at its rated electrical power value. The second and third terms ensure that the system stays stable. The command variable is penalized in the fourth term to limit the actuator effort and energy cost.

The weights involved are independent of the system states and inputs thus designing a reward signal for achieving the control requirements is a delicate step and required empirical tuning of each of these weights.

Moreover, constraints assure the feasible operation range for the FOWT:

$$\beta_{min} \leq \beta \leq \beta_{max} \quad (13)$$

$$-\dot{\beta}_{max} \leq \dot{\beta} \leq \dot{\beta}_{max} \tag{14}$$

$$\omega_{g,min} \leq \omega_g \leq \omega_{g,max} \tag{15}$$

Where the first line limits the blade pitch angle to its feasible working area, the second line restricts the blade pitch rate, and the third one limits the range in which the generator rotational speed is feasible in area III.

The action network and the critic network are simple neural networks with fully connected hidden layers. The proposed structures are illustrated in Fig.4.

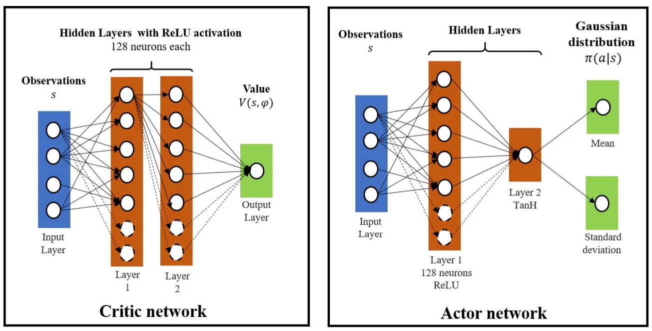

Fig. 4. Architecture neural networks

C. Training process

The training of the agent is done by episode. Each training episode starts with the system in initial conditions such as:

- blade pitch angle β equal to $15deg$ with random noise,
- random platform pitch angle θ_y within the interval $[-5;5]deg$,

A training episode lasts at most 100s with a time step of 0.0125s for the agent in the DRL controller. If the generator rotational speed is outside the range $[80;180]rad/s$ when learning, then the episode is terminated before reaching the maximum number of steps.

During the training process, the agent updates the parameters of the critic and actor neural networks. After training parameters remain at their learned value and the trained actor networks is then stored in $\pi(a|s)$ and can be used in the simulation.

IV. TRAINING AND TEST RESULTS

This section presents the first results obtained with the actor network approximating the optimal control law. The DRL controller is also compared to the baseline controller, a PI collective blade pith controller, who is implemented on OpenFAST as an external dynamic link library (DLL) [18].

A. Training results

The training results of the DRL controller can be seen in Fig.5. The agent was trained on 580 episodes, i.e 6.96 million steps for a duration of approximately 25 hours with a CPU.

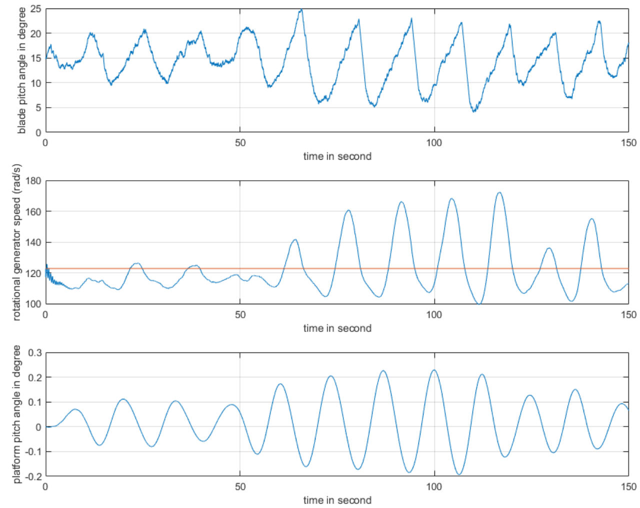

Fig. 5. Training result for the control law, the generator speed and the platform pitch

B. Test results

The test results for a 600s OpenFAST simulation with the trained actor network representing the control law are given in Fig.6, Fig.7 and Fig.8.

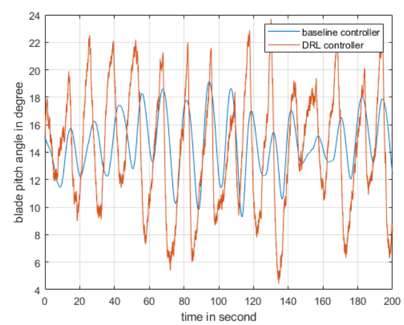

Fig. 6. Test results for the blade pitch angle with comparison to the baseline controller

C. Discussion

For this training duration, the DRL Controller is able to learn a control law for FOWTs systems only from the interaction data.

When comparing the test results, the DRL Controller doesn't perform as well as the baseline controller but it is still able to show signs of learning the desired behavior to reach the control objective. The training should be resumed in order to improve the performances of the agent, and the control law obtained needs to be test for more extreme external conditions. Moreover, the validity of this method should be further analyzed in the future.

The significant computational resources, due to the large numbers of episodes needed to properly account for the interaction of the agent and the environment, is the main

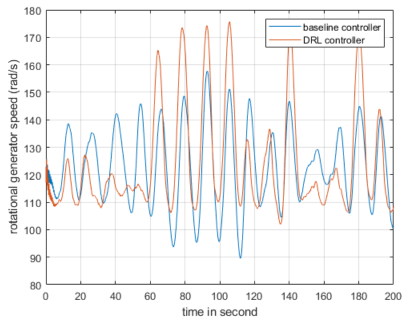

Fig. 7. Test results for the generator speed with comparison to the baseline controller

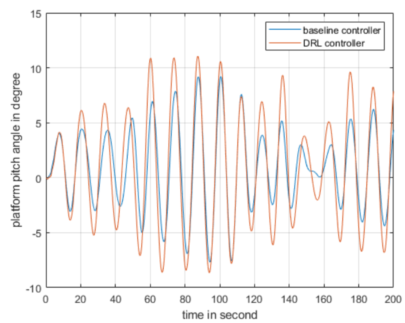

Fig. 8. Test results for the platform pitch with comparison to the baseline controller

limitation encountered. However, the implementation could be more time efficient and powerful with the use of GPU with parallel computing environments.

V. CONCLUSIONS

This paper has proposed a deep reinforcement learning controller for FOWTs. The controller is based on TRPO algorithm with neural networks representation to learn the optimal control law for the control of the FOWT system in operating area III. The effectiveness of the proposed DRL controller has been demonstrated by simulation results for the 5-MW baseline wind turbine with OpenFAST. In future work we consider resuming the the training of the agent, especially with more extreme external conditions to improve its ability to deal with strong perturbations. Moreover, we plan to apply other variations of DRL methods to this framework, and extend the methods to the operating area II.

ACKNOWLEDGMENT

This work has been supported by the ANR CREATIF project, by the EIPHI Graduate School (contract ANR-17-EURE-0002) and the Region Bourgogne Franche-Comté.

FEMTO-ST Institute, Univ. Bourgogne Franche-Comté, UTBM, CNRS, Belfort, France

REFERENCES

[1] T. Larsen and T. Hanson, "A method to avoid negative damped low frequent tower vibrations for a floating, pitch controlled wind turbine," *Journal of Physics: Conference Series*, vol. 75, p. 012073, Jul. 2007.

[2] J. Jonkman, "Dynamics modeling and loads analysis of an offshore floating wind turbine," National Renewable Energy Laboratory, Golden, Colorado, USA, Tech. Rep. NREL/TP-500-41958, Dec. 2007.

[3] D. Matha, "Model development and load analysis of an offshore wind turbine on a tension leg platform," National Renewable Energy Laboratory, Golden, Colorado, Tech. Rep. SR-500-45891, Feb. 2010.

[4] G. Betti, M. Farina, G. Guargliardi, A. Marzorati, and R. Scattolini, "Development of a control-oriented model of floating wind turbines," *IEEE Transactions on Control Systems Technology*, vol. 22, pp. 69–82, Jan. 2014.

[5] F. Sandner, D. Schlipf, D. Matha, R. Seifried, and P. W. Cheng, "Reduced nonlinear model of a spar-mounted floating wind turbine," in *German Wind Energy Conference (DEWEK)*, Bremen, Germany, Nov. 2012.

[6] H. Namik and K. Stol, "Individual blade pitch control of floating offshore wind turbines," *Wind Energy*, vol. 13, pp. 74–85, Apr. 2009.

[7] ——, "Performance analysis of individual blade pitch control of offshore wind turbines on two floating platforms," *Elsevier, Mechatronics*, vol. 21, pp. 691–703, Jun. 2011.

[8] C. C. Sanchez, "Wind and wave disturbance rejection control of floating offshore wind turbines," Master's thesis, University of British Columbia, Vancouver, BC, Canada, 2018.

[9] H. K. T. Bakka and N. Duffie, "Gain scheduling for output h control of offshore wind turbine," in *Twenty-second international offshore and polar engineering conference*, Rhodes, Greece, Jun. 2012.

[10] O. Bagherieh and R. Nagamune, "Gain-scheduling control of a floating offshore wind turbine above rated wind speed," *Control Theory Technol.*, vol. 13, pp. 160–172, Jun. 2015.

[11] P. Zhao and R. Nagamune, "Switching lpv control of a floating offshore wind turbine on a semi-submersible platform," in *IEEE 28th International Symposium on Industrial Electronics (ISIE)*, Vancouver, BC, Canada, Jun. 2019.

[12] R. Bellman, *Dynamic Programming*. Princeton, New Jersey: Princeton University Press, 1957.

[13] R. Sutton and A. Barto, *Reinforcement learning: An introduction*, 2nd ed. The MIT Press, 2018.

[14] D. Cao, W. Hu, J. Zhao, G. Zhang, B. Zhang, Z. Liu, Z. Chen, and F. Blaabjerg, "Reinforcement learning and its applications in modern power and energy systems: A review," *Journal of Modern Power Systems and Clean Energy*, vol. 8, pp. 1029–1042, Nov. 2020.

[15] W. Q. C. Wei, Z. Zhang and L. Qu, "Reinforcement-learning-based intelligent maximum power point tracking control for wind energy conversion systems," *IEEE Transactions on Industrial Electronics*, vol. 62, pp. 6360–6370, Oct. 2015.

[16] ——, "An adaptive network-based reinforcement learning method for mppt control of pmsg wind energy conversion systems," *IEEE Transactions on Power Electronics*, vol. 31, pp. 7837–7848, Nov. 2016.

[17] X. Z. J. Zhang and X. Wei, "Reinforcement learning-based structural control of floating wind turbines," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, pp. 1603–1613, Mar. 2022.

[18] W. M. J.M. Jonkman, S. Butterfield and G. Scott, "Definition of a 5-mw reference wind turbine for offshore system development," National Renewable Energy Laboratory, Golden, Colorado, USA, Tech. Rep. NREL/TP-500-38060, Feb. 2009.

[19] N. R. E. Laboratory. (2023) Openfast documentation. [Online]. Available: https://openfast.readthedocs.io

[20] J. M. Jonkman and M. L. Buhl, "Fast user's guide," in *Nat. Renewable Energy Lab.*, U.S. Dept. Energy, Golden, CO, USA, 2005.

[21] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," *ArXiv*, vol. abs/1502.0547, Feb. 2015.