ISA Transactions 148 (2024) 307–325

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Research article

Fuzzy-based collective pitch control for wind turbine via deep reinforcement learning

Abdelhamid Nabeel *, Ahmed Lasheen, Abdel Latif Elshafei, Essam Aboul Zahab

Electric Power Department – Faculty of Engineering – Cairo University, Giza 12613, Egypt

ARTICLE INFO

Keywords:
Reinforcement learning
Imitation learning
DDPG algorithm
Fuzzy logic
Collective pitch control
Wind turbines

ABSTRACT

Wind turbines (WTs) have highly nonlinear and uncertain dynamics due to aerodynamic complexity, mechanical factors, and fluctuations in wind conditions. Turbulence and wind shear add complexity to modelling, especially in constant power region (region 3). Thus, an effective control design demands a deep understanding of the nonlinearities and uncertainties. This paper suggests a novel model-free reinforcement learning (RL) collective pitch angle controller to operate efficiently in region 3. The proposed controller stabilizes generator speed, maximizes power output, and minimizes fluctuations while accommodating system uncertainties, nonlinearity, and pitch limits. The disparity between WT dynamics due to wind speed perturbations and uncertainties is measured using a gap-metric criterion. The controller design adopts a deep deterministic policy gradient (DDPG) algorithm to train six agents in a medium-fidelity WT environment at different mean wind speeds to ensure the controller's robustness. Initially, imitation learning is used for efficient sample collection to fasten training convergence. Afterwards, the agent learns by interacting with the environment. After the training, the pitch control outputs from multi-trained agents are processed by a fuzzy system to have smooth transitions under different operating conditions. The resulting fuzzy DDPG (F-DDPG) controller is deployed to obtain the optimal pitch control action. The performance of the proposed F-DDPG controller is compared to the gain-scheduled PI (GSPI), Linear-Quadratic-Regulator (LQR), and single-DDPG-agent controllers. The controllers are simulated in high-fidelity onshore and offshore 5-MW WT environments using the OpenFAST/MATLAB simulation tools. The results reveal the superiority of the proposed controller in generalizing its optimal performance in different operating conditions.

1. Introduction

Wind turbines (WTs) have recently become more potent electrical-generating plants as the globe attempts to migrate to sustainable and renewable energy sources. The benefits of WTs are witnessed by their positive effects on the environment, reliability as a source, and status as a powerful energy source. According to the International Renewable Energy Agency (IRENA), in 2023, the total installed power capacity globally of WTs is 898.856 GW by the end of 2022 [1]. The Global Wind Energy Council (GWEC), in 2023, expects 550 GW of additional installations in 2023–2027, with an average annual installation of 110 GW [2]. The main objective of the operation of WTs is to be economically feasible, which can be achieved through maximizing the annual GWh energy production. The vital role of control systems for WTs is to stabilize the operation and improve annual energy harvesting.

WTs have three regions of operation, depending on wind speeds. Region 1 specifies the range of low wind speeds up to the cut-in value, where the WT starts to operate by speeding up the rotor with no power generation. In region 2, the wind speeds are entrapped between the cut-in speed and the rated wind speed where the rotor reaches its rated speed. Region 3 refers to the operational state where the turbine maintains its rated power and speed. This study focuses on region 3, called the constant power region. The main goal of control design in this region is to limit the power captured by the WT to its rated power while maintaining the operation at the rated rotor speed at high wind speeds [3]. This goal can be achieved by an optimal pitch control mechanism that effectively regulates the rotor loads and aerodynamic power without exceeding the safety constraints and limits of the design [4]. The blade inertia determines the velocity of blade rotation and how robust the pitch mechanism is, so the maximum pitch rate of the WT model

* Corresponding author.
E-mail addresses: Abdelhamid.Nabeel@eng.cu.edu.eg (A. Nabeel), Ahmed_lasheen59@cu.edu.eg, Ahmed_lasheen@eng.cu.edu.eg (A. Lasheen), elshafei@eng.cu.edu.eg (A.L. Elshafei), zahab0@eng.cu.edu.eg (E. Aboul Zahab).

https://doi.org/10.1016/j.isatra.2024.03.023
Received 7 September 2023; Received in revised form 22 March 2024; Accepted 22 March 2024
Available online 26 March 2024

used is (8°/sec) [3], [5]. The pitch control consists of three types: individual pitch control (IPC) [6], cyclic pitch control, and collective pitch control (CPC). This study focuses on CPC, which is responsible for adjusting all blades simultaneously to the same pitch angle while achieving the rated power extraction and rated rotor speed from the WT. The primary core of this paper focuses on a CPC that preserves the blades' pitch angle constraints and pitch rate.

In the region-three operation mode of WTs, pitch control encounters several challenges. These include aerodynamic instabilities due to high-speed winds, model uncertainties, and limitations in pitch actuator speed. Additional challenges involve time delays in control actions, variable and unpredictable wind gusts, and the need for rapid response to fluctuating wind conditions [6], [7]. These complexities require highly advanced control strategies to ensure system efficiency and stability. Designing an efficient pitch controller demands an accurate WT system model that comprehensively encapsulates the highly nonlinear dynamics, potential instabilities, and unpredictable nature of wind patterns in this region. Achieving such a detailed and precise model is a daunting task, crucial for overcoming the inherent difficulties in controlling WTs under these demanding operational conditions.

Nowadays, researchers focus on developing cutting-edge artificial intelligence and RL control algorithms for WT operation, as summarized in Table 1. In [8], the authors develope an adaptive model-based controller that uses radial-basis function neural networks and Lyapunov stability for maximum power point tracking and pitch angle control with no tracking error. In [9], the authors focus on pitch angle control using DDPG in tuning the parameters of a nonlinear backstepping controller. In [10], the authors use adaptive dynamic programming with a temporal difference technique for CPC. In [11], the authors develope a hybrid RL and neural networks pitch controller to maximize power output. In [12], the authors propose a data-based RL pitch angle controller by constructing an augmented time-delay system based on the control input, time delay, and state as an augmented vector and augmented Bellman equation. In [13], the authors develope a model-based RL method that relies on a deep neural network for system dynamics approximation and MPC for pitch angle control. In [14], the authors develope a hybrid RL-based control algorithm with a conventional PID controller for pitch angle control. In [15], the authors develope a trust region policy optimization (TRPO) RL pitch angle controller accounting for system constraints. In [16], the authors develope a hybrid controller using deep learning for wind speed forecasting which is fed to a fuzzy logic controller. In [17], the authors develope an incremental model for nonlinear system approximation and dynamic heuristic programming for pitch angle control. In [18], the authors develope an actor-critic RL-based pitch angle controller for effective power extraction.

Some challenges are noticed in the previous work. First, there is reliance on on-policy RL learning strategies, as seen in [15], [18], and [10]. These strategies depend only on current state information while neglecting past experiences during the learning process. That requires a lot of exploration and many samples for convergence. Second, hybrid techniques between conventional controllers and RL-based controllers are implemented due to the reservations about applying only RL-based controllers, as seen in [9], [13], [14], and [16]. These reservations come from the lack of sample efficiency, performance instability, and slow RL-controller training convergence in highly stochastic environments. Third, the complexity of the controller design increases due to the desire to compensate for the unmodelled dynamics and uncertainties through data-driven system identification, as observed in [8], [13], [17], and [12]. Fourth, reduced system efficiency, lower power extraction, and increased fluctuations have been observed when implementing only an RL-based controller, as inferred from the results in [15]. Fifth, RL algorithms are tested on a simplified model version of WT, which cannot emulate the actual high stochastic WT environment as in the cases of [10] and [11].

Most RL algorithms face multiple challenges when applied to

Table 1
Literature review summary about the latest AI and RL-based controllers.

Ref.	Applied Method	Objective	Problems	Treatments
[8]	Adaptive Neural-Network.	Pitch angle control. Generator speed regulation.	High tracking error. High disturbance and uncertain dynamics.	Zero-converging tracking error method. Auxiliary adaptive control.
[9]	Nonlinear integral backstepping + Digital twin DDPG.	Pitch angle control. Generator speed and power regulations.	Challenging system dynamics modelling and uncertainties. Hard to find the optimal parameters for the nonlinear integral backstepping.	Twin Delayed-DDPG for parameters tunning.
[10]	Adaptive Dynamic Programming + Temporal difference + Actor-critic RL structure	Pitch angle control. Generator speed regulation.	Wind speed disturbances. Uncertainties in wind turbine model parameters.	Model-free pitch angle controller.
[11]	RL + Neural networks.	Pitch angle control. Generator speed and power regulations.	Optimality of pitch angle choice.	Crafting reward function accounts for power regulation.
[12]	Databased RL + Bellman equation.	Pitch angle control. Maximize extracted power.	Time-delay system. Inaccurate system modelling.	Construction of augmented system. Model-free RL controller.
[13]	Deep RL + MPC.	Pitch and torque control. Generator speed and power regulations.	Dynamics uncertainties. Unpredictable actuator faults.	Data-driven model-free control + real time system dynamics adaptability.
[14]	RL + PID controller + Learning observer.	Pitch angle control. Maximize extracted power.	Slowness of training convergence. Uncertainties and disturbances. Oscillations of training.	PID training accelerator. Model-free RL controller. Action selector + exploration window.
[15]	RL Trust Region Policy Optimization.	Pitch and torque control. Maximize extracted power.	Constraints. Nonlinearity in system dynamics. Uncertainties in wind speed.	Designing of trust region policy optimization model-free RL controller.
[16]	Fuzzy logic + Deep learning.	Pitch angle control. Maximize extracted power.	Complex dynamics. Wind speed uncertainties.	Fuzzy logic control + Deep learning for estimation of the current and future wind speed.
[17]	Incremental model-based dual heuristic programming.	Pitch angle control. Extracted power regulation.	Absence of accurate modelling. Challenging approximation of all model dynamics.	Approximation of nonlinear dynamics. Online-learning of partial model dynamics.
[18]	Actor-critic structure + RL.	Pitch and Torque control. Reduction in power fluctuations.	Interactions between subsystems in various operating areas are ignored.	Designing a model-free continuous actor-critic RL.

complex systems like WTs with continuous dynamics. For discrete RL, issues include quantization errors due to approximating continuous actions with discrete values, suboptimal solutions, and computational inefficiency in high-dimensional spaces. Also, model-free RL algorithms struggle with the exploration-exploitation dilemma and low sample efficiency, requiring extensive environmental interactions to learn an optimal policy. Additionally, selecting a practical reward function is complex. Poor design of reward functions can lead to unintended behaviour [19], [20]. The DDPG algorithm is recommended to achieve performance generalization and stabilization in a continuous action space without the need for mathematical modelling equations [21]. It is a simple model-free off-policy RL controller based on deep Q-Networks and actor-critic structure [22]. The term "deep" in DDPG refers to using deep neural networks as function approximators for actor and critic networks. The term "deterministic policy gradient" refers to deterministic action taken by an actor-network that is updated continuously based on policies that conduct gradient ascent on a scalar objective function called the quality (value) function (Q-function) [19]. Consecutively, this paper aims to design a modified version of a DDPG deep-RL-based collective pitch controller where the training is initially guided by a fuzzy model predictive controller (FMPC) [23]. The design goals include fastening the learning time, robustness against stochastic winds, maximizing power output, maintaining rated generator speed, reducing fluctuations, handling high nonlinearity and model uncertainty, and improving controller response to actuator limitations.

The DDPG algorithm offers five critical advantages for continuous control: it extends deep Q-learning to continuous action spaces [24], efficiently explores through adding noise to control action, ensures learning stability via actor-critic networks [25], employs off-policy learning that can learn from past experiences stored in a replay buffer to improve stability, and is compatible with complex function approximators like neural networks. However, it also faces challenges such as high sample requirements for convergence because it is a model-free algorithm [19], potential learning instability, policy gradient bias [24], and limited robustness in stochastic environments like wind turbines (WT). Model predictive control (MPC) is a model-based potential candidate for pitch control with four primary advantages [23]. It is robust against disturbances and model errors due to its predictive nature. It can handle various constraints. It utilizes optimization techniques for the best control actions. It offers transparent decision-making [25]. Compared to MPC, DDPG offers unique strengths: it handles the WT's nonlinear dynamics and high-dimensional spaces using deep neural networks, offers online learning and adaptability, and is both cost-efficient and scalable [19]. Unlike MPC, which requires a linear approximation of the model and real-time optimization, a trained DDPG agent is computationally efficient at execution [25].

This study offers a modified version of a DDPG algorithm where six distinct DDPG agents are trained to optimize the agents' collective pitch control action and to construct a robust RL pitch angle controller through the fuzzy logic principle. Each agent is trained under a stochastic wind speed profile with a predetermined average speed in order to mitigate the issues of learning instability, unmodelled dynamics, and uncertainties caused by high-turbulent wind. The FMPC assists the early learning episodes of training by providing efficient samples to fasten the training convergence. After training (at the deployment phase), the outputs from all trained agents are manipulated by a fuzzy system to construct the optimal CPC action and to ensure controller robustness at all operating points. The proposed novel control strategy is called F-DDPG. This innovative control strategy inherits the MPC's strengths with the advanced capabilities of the modified DDPG to manage WT control effectively in region 3.

The contribution of this paper is summarized through the following four aspects.

- It introduces imitation learning integrated with reinforcement learning [26] in WT control applications to increase the collection of high-efficiency samples to shorten training convergence time [27]. Initially, the FMPC is deployed to enrich the DDPG agent's replay buffer with high-quality samples. Following this, the DDPG agent self-learns using these samples and environmental interactions to improve its performance.
- The enhancement of the DDPG algorithm's learning process in this study involves two essential modifications. Firstly, it incorporates K-step bootstrapping returns, which allow the algorithm to learn the consequences of its actions over longer horizons more efficiently. This reduces the overestimation of the Q-value challenge that faces the DDPG algorithm [25]. Secondly, changes to the topology of the actor-critic networks have been proposed to reduce the computational costs of learning compared to the standard DDPG algorithm in [19] and [28].
- A simple fuzzy system is implemented for the CPC actions of the multi-trained DDPG agents to improve the controller's robustness, stability, and performance. This approach has overcome the WTs' high wind uncertainties and stochastic behavior in region three, where wind speeds are intense.
- Being trained on a medium-fidelity onshore WT model, this study demonstrates the proposed F-DDPG controller's ability to generalize its performance to high-fidelity 5-MW onshore and offshore WT models, closely approximating real-world conditions in region 3. The study emphasizes the proposed controller's compensation of unmodelled dynamics and adaptation to new unconsidered dynamics.

The paper is structured as follows: Section 2 outlines the RL framework and the WT model used in simulations. Section 3 details the DDPG agents' training phase methodology, discusses the F-DDPG deployment phase, presents the actor-critic network topology, shows the rationale behind the chosen reward function, and explains the training algorithm's mathematics. This section also describes integrating fuzzy logic with the collective pitch control actions of the trained agents. Section 4 presents the training and simulation results of the proposed controller, including a comparative analysis with other controller types. Section 5 engages in a discussion of these results, addressing limitations and outlining future work. Finally, Section 6 concludes the study.

2. RL framework and wind turbine system analysis

This section is organized into four distinct subsections. Subsection 2.1 focuses on defining the RL framework's key components. Subsection 2.2 provides a detailed description of the WT system and its parameters. Subsection 2.3 explains the complexity of the WT model by highlighting the activated degrees of freedom essential for the training and deployment of our proposed controller. Finally, subsection 2.4 introduces the gap-metric criterion, a tool for measuring disparities between different WT models arising from wind uncertainties.

2.1. RL Framework

In the context of our study on wind turbine control, the definition of the RL problem is integral to developing an effective control strategy. This definition comprises four fundamental terminologies: state, action, reward function, and environment. These elements play a vital role in shaping the learning process and the resulting controller's performance. By defining these components, we aim to construct a robust and efficient framework capable of addressing the unique challenges presented by the dynamic and complex nature of WT operations.

- **Environment**: The environment is the framework which comprehensively contains external conditions, applied loads, and the dynamic behaviour of the wind turbine, providing a realistic and challenging setting for the RL problem. In the context of this study, the term "medium-fidelity environment" represents the WT

framework operating at a certain average wind speed in region 3. Meanwhile, the "high-fidelity environment" means WT framework operation at all wind speed ranges in region 3.

- **Observations (states):** The WT system is characterized by low-dimensional states, which include the torsional displacement of the drive-train θ_{DT} , error in rotor speed $\Delta\omega_{rotor}$, and the torsional speed of the drive-train ω_{DT}. These states offer partial but informative insights into the WT's operational status.
- **Action**: The action is the control command applied to the system. In our RL framework, the collective pitch control signal is the action. This action is critical as it adjusts the turbine blades' pitch angles, directly influencing the system's performance in region three operation mode.
- **Reward Function**: The reward function is the objective function that needs to be maximized throughout the training process. It is designed to ensure the generator's speed and power output remain close to their optimal levels. The term "cumulative reward" is the sum of all the rewards that an agent receives during an episode. The agent's goal in RL is to learn a policy that maximizes the expected cumulative reward over many episodes. The cumulative reward is employed during the training and deployment phases as a performance evaluation metric.

2.2. Wind turbine system description and parameters

The WT model relies on Open-FAST (Fatigue, Aerodynamics, Structure, and Turbulence), an open-source software package [5]. Open-FAST is an aero-hydro-servo-elastic platform created by the National Renewable Energy Laboratory (NREL) to model the dynamic reactions of both onshore and offshore wind turbines. Open-FAST allows users to enable or disable specific degrees of freedom (DOFs) to simplify or complicate the simulation. It offers three main models for testing and simulation purposes: the onshore, fixed-bottom offshore, and floating offshore. The performance of a 5-MW horizontal-axis onshore model and a floating offshore wind turbine model termed "OC3-Hywind spar buoy" [29] are tested in this study. The model framework, illustrated in Fig. 1, shows the pitch angle controller, system dynamics, applied loads, external conditions, and their interactions for an offshore model. The onshore model has the same framework, excluding the hydrodynamic, platform, and mooring system dynamics. The model's main specifications are listed in Table 2. Additional information regarding the wind turbine modularization framework exists in [5], [29], and [30].

2.3. Activated DOFs in control design and implementation

The DOFs activated during the training and deployment phases represent the complexity of the WT model in these phases. DDPG is a model-free RL algorithm that generalizes the performance without the need for mathematical modeling equations describing the complex dynamics of WT [31]. It is trained in a medium-fidelity WT environment. The activated dynamics during training are the blades, drive-train, generator, nacelle, and tower DOFs. The FMPC initially aids in the training phase by collecting highly efficient samples. FMPC is a model-based controller that relies on fuzzy linearized models of the WT. Its design depends on a reduced-order model that activates only the drive-train and generator DOFs. The states used in the model design are

Table 2
Wind turbine model mean specifications.

Parameter description	Value
Hub height	90 m
Rated generator speed	1173.7 rpm
Rated generator output power	5000 kW
(Cut in – rated - cut out) wind speeds	(4 – 11.4 – 25) m/s
Number of blades	3
Rotor diameter – Blade length	126 m – 63 m
Floating platform depth	120 m
Pitch angle permissible range	[0 – 90] degrees – [0-$\frac{\pi}{2}$]$rads$
Maximum pitch rate	±8degs/s (0.139 rad/s)

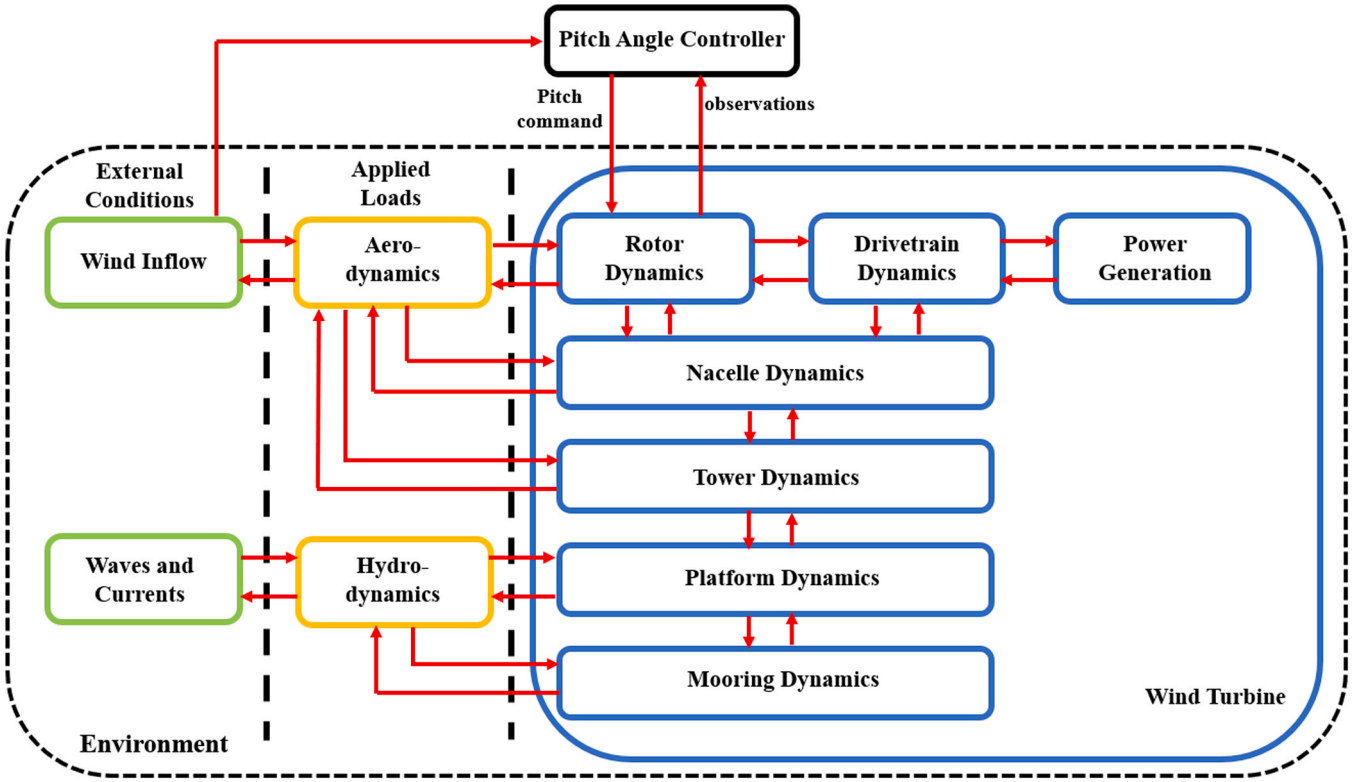

Fig. 1. Open-FAST modularization framework dynamics for offshore model.

$(\theta_{DT}, \Delta\omega_{rotor}, \omega_{DT})$. The design criteria of FMPC are precisely mentioned in [23].

In the deployment phase, the proposed controller is evaluated in high-fidelity environments using both onshore and offshore wind turbine models. For the onshore model, all DOFs are enabled except for the platform-related ones. In the offshore model, all DOFs are activated, including additional dynamics like platform DOFs, hydrodynamics, and mooring system dynamics, as detailed in Table 3 and shown in Fig. 1. This is done to test the controller's ability to generalize its performance across different settings.

2.4. Disparity between wind turbine models

A WT model is time-varying due to uncertainties in wind inflow. To ensure the controller's robustness, a measure of the disparity between the WT systems dynamics due to random perturbations of wind speeds is critical. Consecutively, a simplified analysis using the gap-metric criterion is introduced [23]. A reduced-order nonlinear WT model, whose generator and drive-train DOFs are only activated, is linearized at seven different operating points using the Open-FAST package linearization tool. The operating point is determined by three variables: the rotor speed, generator azimuth angle, and wind speed. The operational points are represented by mean wind speeds of (12, 14, 16, 18, 20, 22, and 24) m/s. Since the linearization is done in region three operation mode, the rotor speed is fixed at the rated value. At each mean wind speed, the Open-FAST linearization tool produces 36 linearized models at different generator azimuth angles with a 10-degree angle gap between each two consecutive models. The multi-blade coordinate transformation tool is used to calculate the average model of the 36 linearized models and express the integrated dynamics of turbine blades to a fixed (non-rotating) frame [32]. As a result, the only variable that distinguishes the average linearized models is wind speed.

The operating points are selected based on the gap metric criterion, which evaluates the disparity of WT reduced-order linear model dynamics at these points. The dynamics of two linear models are considered different if the gap distance between the two models is more significant than a predetermined value. Eq. 1 describes the maximum gap distance $D(T_n, T_m)$ between two single-input-single-output transfer functions (T_n, T_m) at average wind speeds (n,m). The gap distance values lie between [0,1] where zero means no difference between models and one means quite different models. $\overline{T_n}$ and $\overline{T_m}$ are the conjugates of T_n and T_m. The correlation matrix representing the disparity and distances between the linearized models are shown in Table 4. Thus, the dynamics of the linearized models at 2 m/s wind speed gap are considered sufficient to describe the WT system at all possible wind speeds in region 3.

$$D(T_n, T_m) = \left\| \frac{T_n - T_m}{\sqrt{1 + T_n\overline{T_n}}\sqrt{1 + T_n\overline{T_m}}} \right\|_\infty \tag{1}$$

Table 3
Three-blade wind turbine model internal DOFs.

Element	Number of DOFs	Description
Blades	2	Flap-wise modes per blade (3-blade) (1st and 2nd)
	1	1st Edge mode per blade (3-blade)
Drive-train	1	rotational flexibility
Generator (Rotor)	1	Variable speed
Nacelle	1	Yaw bearing
Tower	2	side-to-side bending modes
	2	fore-aft bending mode
Platform (for offshore model only)	3	Translation modes (horizontal surge, horizontal sway, vertical heave)
	3	Rotational modes (Roll tilt, pitch tilt, yaw)

Table 4
Gap metric correlation matrix of average linearized models.

Transfer functions	T_{12}	T_{14}	T_{16}	T_{18}	T_{20}	T_{22}	T_{24}
T_{12}	0.000	0.149	0.244	0.324	0.381	0.430	0.512
T_{14}	0.149	0.000	0.099	0.185	0.246	0.300	0.381
T_{16}	0.244	0.099	0.000	0.088	0.152	0.208	0.263
T_{18}	0.324	0.185	0.088	0.000	0.065	0.123	0.194
T_{20}	0.381	0.246	0.152	0.065	0.000	0.059	0.103
T_{22}	0.430	0.300	0.208	0.123	0.059	0.000	0.072
T_{24}	0.512	0.381	0.263	0.194	0.103	0.072	0.000

3. Proposed controller design

The proposed F-DDPG controller, innovatively designed to regulate the wind turbine system in region 3, is introduced in this section. Subsection 3.1 presents the training and deployment phases of the controller through insightful block diagrams. Details regarding the actor and critic neural network topologies are provided in Subsections 3.2 and 3.3, respectively. The rationale behind the choice of the objective function (reward function) is explored in Subsection 3.4. The mathematics of the training algorithm are described in Subsection 3.5. Additionally, the implementation of the fuzzy system is discussed in detail in Subsection 3.6.

3.1. Proposed controller structure

The application of the proposed controller has two phases: the training and deployment phases. During the training phase, attempts to train a single DDPG agent with a highly turbulent wind inflow covering the entire wind speed range encountered issues with training instabilities, performance, and convergence. These issues led to the decision to train six separate DDPG agents. Each agent is individually trained at a specific average wind speed, covering all the key operating points in region 3 (14, 16, 18, 20, 22, and 24 m/s), thereby ensuring comprehensive training across the entire spectrum of wind conditions typically encountered. As shown in Fig. 2, each agent comprises six main blocks: FMPC, experience replay buffer, online actor network, online critic network, target actor network, and target critic network.

Imitation learning is initially used to provide efficient samples to fasten training convergence. The FMPC acts as the demonstrator to guide the DDPG agents. During the first five training episodes, the switch selects the FMPC block to provide the control action to the environment. The selection of five initial episodes for efficient sample collection is made empirically through trial and error. Testing with one to four initial episodes yielded unsatisfactory results.

The experience replay buffer R_B block serves as a memory storage for past experiences. The buffer contains tuples, each of which comprises the current state (s_t), the current action output (A_t), the observed reward (r_t), and the subsequent state (s_{t+1}) following the execution of A_t at time step t. Initially, FMPC supplies efficient samples to the experience replay buffer block.

The online actor network block is parameterized by ϕ^μ weights. It is responsible for generating specific (deterministic) pitch control action $\mu(s|\phi^\mu)$ given the states (s) of the environment as inputs. Ornstein-Uhlenbeck random noise, represented in equations (4) to (6), is added to the actor network output. It allows the agent to explore the action space more coherently due to temporal correlation, leading to better performance and faster learning [19].

The online critic network block is parameterized by ϕ^Q weights. It is used to evaluate the online actor network performance. This is done by estimating the value function based on the state-action pair as input. It outputs a single scalar value $Q(s, A|\phi^Q)$ representing the estimated return (Q-value) of that state-action pair.

The target actor network block mitigates the issues of data correlation and non-stationarity, improving the algorithm's ability to converge

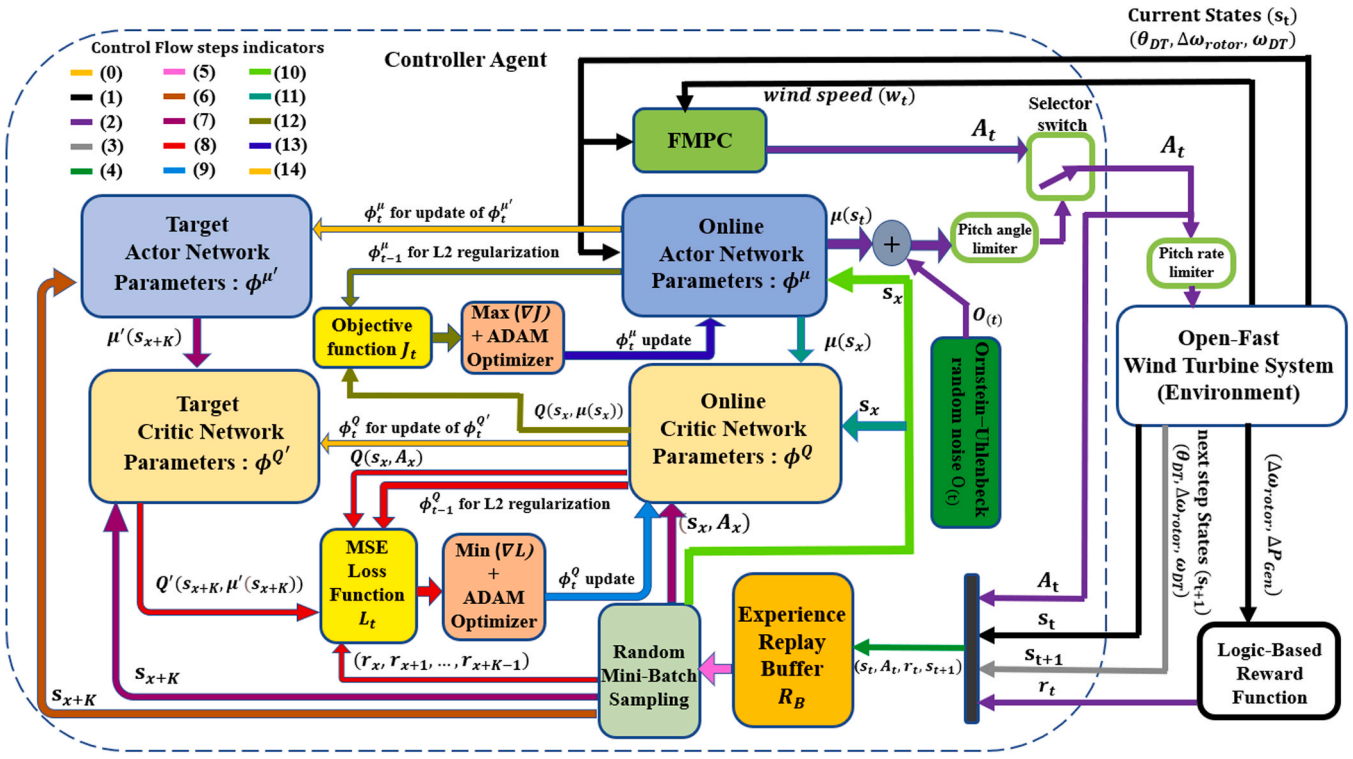

Fig. 2. The training stage of F-DDPG controller.

to a good policy [19]. It is parameterized by $\phi^{\mu'}$ weights. It serves as a more stable approximation of the policy represented by the online actor network. It is used to generate the target action $\mu'(s_{x+K}|\phi^{\mu'})$ based on the following K-step states s_{x+K}.

The target critic network block mitigates the risks of policy oscillation and divergence, making the learning process more robust and stable. It is parameterized by $\phi^{Q'}$. It is used to output the target estimated return $Q'(s_{x+K}, \mu'(s_{x+K}|\phi^{\mu'})|\phi^{Q'})$ based on the target action and next K-step states. Then, the K-step target Q-value y_x is computed using the discounted sum of the target estimated return and the actual rewards received $(r_x, r_{x+1}, \ldots, r_{x+k-1})$. K refers to the number of time steps into the future that the algorithm considers when calculating y_x. K-step target return often balances the bias-variance tradeoff. A 1-step return is unbiased but may have high variance, while a return that considers the entire future (i.e., until the end of the episode) has low variance but may be biased [25]. K-step returns represent a middle choice that yields faster and more stable learning.

The learning process of each agent is done iteratively based on predefined numbers of episodes (E). Each episode consists of predefined numbers of timesteps (T). The learning process continuously updates the parameters of the online actor-critic networks and target actor-critic networks at each timestep (t). The online actor network parameters are updated to produce actions that maximize the objective function (J_t) representing the cumulative expected return, as approximated by the online critic network. The online critic network parameters are updated to minimize the mean-square-error (MSE) loss function (L_t), which represents the difference between estimated and target Q-values.

In developing the proposed F-DDPG controller, conventional gradient descent optimization techniques are superseded by adopting both imitation learning and adaptive moment estimation (ADAM) optimization approaches. This mitigates the challenges of slow convergence rates, much like recent advancements of the PID accelerator in [14]. The online updates of the actor-critic networks parameters are carried out using the ADAM optimization technique. It is chosen because of the randomness in the optimization process due to the random mini-batched data used. Also, it is computationally efficient, needs fewer memory requirements, and is a suitable choice for stochastic optimization [33]. The target actor-critic networks parameters are softly updated to track the online actor-critic networks, making the learning process more stable.

The control signal flow shown in the top left of Fig. 2 is demonstrated as steps. The initial step (zero step) is to initialize the experience replay buffer memory and online actor-critic networks weights. The weights are then passed to initialize the target actor-critic networks. Steps 1 through 5 are dedicated to the sequence followed to store tuples in R_B. In the first step, the agent receives the current states s_t (θ_{DT} , $\Delta\omega_{rotor}$ and ω_{DT}) from the environment and passes them to the FMPC and the online actor network. The selector switch is pointed to the FMPC in the initial five learning episodes. After the five episodes, the selector switch is directed to the online actor network to take control. In the second step, the control action A_t, whether exerted from the FMPC or the actor network, is passed to the environment and executed. Also, the reward r_t is calculated based on the normalized absolute error of rotor speed $||\Delta\omega_{rotor}||$ and the normalized absolute error of generator output power $||\Delta P_{Gen}||$, as mentioned in (3). In the third step, the next states s_{t+1} are observed based on the A_t executed in step 2. In the fourth step, the transition tuple (s_t, A_t, r_t, s_{t+1}) is stored in the R_B. In the fifth step, random M-numbered tuples are selected to form mini-batch samples to train online actor-critic networks.

Steps 6 through 9 are dedicated to updating online critic network parameters. In step six, s_{x+k} is passed to the target actor network. The objective of step seven is to prepare the inputs for the online critic network and target critic network. $\mu'(s_{x+K}|\phi^{\mu'})$ and s_{x+k} are passed to the target critic network. The state-action pair (s_x, A_x) is passed to the online critic network. The objective of step eight is to construct the MSE loss function L_t. The output $Q(s_x, A_x|\phi^{Q})$ from the online critic network, the output from the target critic network $Q'(s_{x+K}, \mu'(s_{x+K}|\phi^{\mu'})|\phi^{Q'})$, the actual reward sequence $(r_x, r_{x+1} \ldots r_{x+k-1})$, and the previous tunned parameters of the critic network ϕ_{t-1}^{Q} are used to construct L_t as shown in equations (8) and (9). The objective in step nine is to minimize L_t through calculating gradient descent ∇L_t with respect to the online critic network parameters ϕ^{Q} as shown in equation (10). To calculate the

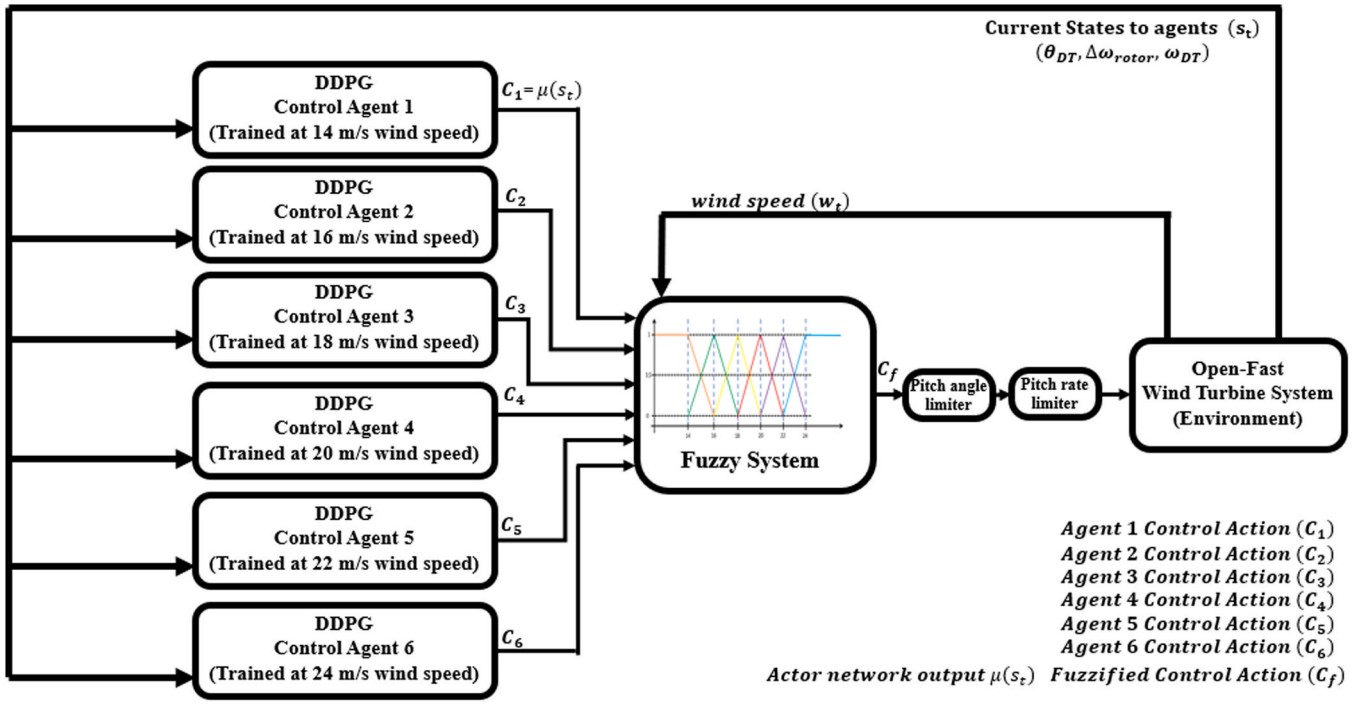

Fig. 3. The deployment stage of F-DDPG controller.

newly updated parameters ϕ_t^Q based on ∇L_t, ADAM optimization technique is used.

Steps 10 through 14 are dedicated to updating online actor network parameters. In step 10, the state s_x is passed to the online actor network. In step 11, the output control action $\mu(s_x)$ and s_x are passed to the online critic network to output the estimated Q-values $Q(s_x, \mu(s_x))$. In step 12, the objective function J_t is constructed using $Q(s_x, \mu(s_x))$ and previously tunned parameters of actor network ϕ_{t-1}^{μ}, as mentioned in equation (14). The objective of step 13 is to maximize J_t through calculating its gradient with respect to the online actor network parameters ϕ^{μ}, then to calculate the newly updated parameters ϕ_t^{μ} using the ADAM optimization technique. In the final step (14th step), the target actor-critic networks parameters $(\phi_t^{Q'}, \phi_t^{\mu'})$ are smoothly updated based on the recent values of online actor-critic networks parameters $(\phi_t^{Q}, \phi_t^{\mu})$ as mentioned in equations (16) and (17). The training sequence is repeated in the next iteration starting from step 1. The previously mentioned steps are explained mathematically in detail in Subsection 3.5.

After training (at the deployment stage), the agents' online actor networks are the only activated networks with already-trained parameters that are no longer updated. The online actor network makes decisions without the need to learn from new reward values. Its primary focus in this stage is to exploit the output, not to explore the environment anymore. As a result, Ornstein-Uhlenbeck noise is not added to the actor network output. Each agent outputs the optimal pitch control action based on the current states, as shown in Fig. 3. Then, the agents' outputs are processed by a fuzzy system, as explained in detail in subsection 3.6. This results in the generation of the fuzzy optimal control action C_f which is applied to the environment.

3.2. Actor networks topology

The choice of actor network layers and neurons has been initially based on two hidden series of fully connected layers with 400 and 300 neurons, respectively as suggested in [19] and [28]. The training results have not been satisfactory due to the high complexity of the relation between the observed states and the output control action. The number of layers and neurons in each has been a crucial hyperparameter to tune. So, further modifications have been made. The empirical training findings have been significant in selecting four series layers.

The proposed architecture is applicable to both the online and target actor networks. Each actor network starts with a feature input layer for the states, succeeded by four hidden layers with neuron counts of 125, 100, 75, and 50, in that order. A tanh activation function follows every hidden layer. The architecture ends with a single-neuron output layer representing the control action.

The computational training time depends on several factors: the complexity of the model, the number of layers and neurons of a neural network, optimization technique, and batch size. Doubling the number of parameters in a neural network often doubles its computation training time [34]. According to the method of calculating the number of parameters in the fully connected network in [35], the estimated adjustable parameters for each actor network is 24,525 with 4-hidden layers. Using the topology stated in [19] and [28], each actor network would have 130,000 parameters with 2-hidden layers. According to our available processing capabilities, the computational training time of the suggested topology of the actor and critic neural networks has scored 375.78 s as an average total elapsed time per episode during the training process. In comparison, the topology stated in [19] and [28] has scored 602.79 s. Thus, our suggested topology has decreased the computational training time by a factor of 1.6.

The selection of activation functions in neural networks, specifically for hidden units, remains a key research area with few definitive guidelines [34]. While the rectified linear unit (ReLU) is often preferred [34], [36], it was not effective in our tests due to the "dying ReLU" issue, where negative inputs lead to zero outputs and vanishing gradients in deeper networks [37]. The tanh function act similar to the barrier function in the adaptive dynamic programming for safety learning-based controller. Given that the state inputs are bounded, we choose the tanh function, which scales inputs to be between − 1 and 1. This maintains bounded neuron outputs without zero outputs and softens gradient vanishing issues [36]. The weights of the networks are initialized using glorot xavier's initialization method [41]. Furthermore, the DDPG algorithm's performance is greatly influenced by the choice of hyperparameters [38], including the activation function. Thus, the tanh function has proven to stabilize the learning process and enable the network to capture more complex non-linear patterns in the

Table 5
Parameters and hyperparameters for F-DDPG agents training.

Parameters	Symbol	Value
Number of episodes	E	300
Time steps per episode	T	8000
Sampling interval	T_s	0.0125
Time step	t	0.0125
Minimum permissible pitch angle	$A_{\min}$	0 rad
Maximum permissible pitch angle	$A_{\max}$	$\pi/2$ rad
Learning rate of actor network	α_μ	$1 * 10^{-4}$
Learning rate of critic network	α_Q	$1 * 10^{-3}$
L2 Regularization factor of actor network	λ_μ	0.1
L2 Regularization factor of critic network	λ_Q	0.01
Discount factor	γ	0.95
exponential decay rate for the first moment estimates	β_1	0.9
exponential decay rate for the second moment estimates	β_2	0.999
Minibatch size	M	128
Target network smooth factor	τ	0.005
Experience buffer samples size	R_B	10^6
Initial noise value = Initial action value	$O_{(t_0=1)}$	0.3
Noise process mean	μ	0
Noise model mean attraction constant	θ	0.15
Initial noise standard deviation	$\sigma_{(t_0)}$	$\frac{0.01 * (A_{\max} - A_{\min})}{\sqrt{T_s}} = 0.14$
Half-life-time Samples	HLT	4000
Noise standard deviation decay rate	σ_δ	$1.7327 * 10^{-4}$
Minimum standard deviation	$\sigma_{\min}$	0
K-step lookahead	K	10
Random number (with mean = 0, standard deviation =1)	$R(t)$	Auto − generated
Wiener process	$W_{(t)}$	$R(t) * \sqrt{T_s}$ = Auto − generated

data, underlining its empirical advantage for maintaining learning stability in our study.

3.3. Critic networks topology

The proposed architecture is applicable to both online and target critic networks. The architecture consists of three pathways (states path, action path, and main path), each of which handles distinct aspects of the network's function. In "States Path," the input states are processed through a series of five fully connected layers with 125, 100, 75, 50, and 5 neurons, respectively. The "Action Path" is responsible for processing the action input to learn the essential patterns needed for Q-value function estimation. It consists of four successive fully connected layers with 100, 75, 50, and 5 neurons, respectively. The tanh activation function follows each hidden layer in both paths. Both pathways are combined at the main pathway to be processed with an additional hidden layer with five neurons. The architecture ends with a single-neuron output layer representing the estimated / target Q-value. According to [35], the total number of tunable parameters for each critic network is 36,571, resulting in a shorter computational training time as compared to the usage of the topology suggested in [19] and [28].

3.4. Reward function formulation

Choosing a reward function and its state representation is challenging, as it depends on understanding the environment, states, and actions. The MSE, mean error (ME), only-positive (O-P), and positive-negative (P-N) dense reward functions in [39] have been tested on our continuous-action WT problem to see their behavior. Also, a suggested logic-based reward mechanism is tested to keep the generator speed and output power at rated levels, as mentioned in (3). The logic-based reward function has provided the finest results during training.

Dense and sparse functions are two main types of shaping reward functions. The dense reward functions give frequent, incremental feedback for actions, facilitating rapid learning. The sparse reward functions offer feedback only at significant milestones, making learning slower [20]. The logic-based reward function is considered a dense reward function in our problem, as feedback is provided to the agent at every time step based on the current state.

The inputs to the reward function must be informative and relevant to the environment. The rotor speed and generator output power are chosen as inputs. The chosen inputs differ from the states provided to the critic network but overlap in rotor speed as a state for both. In [39], the generator output power is used to estimate the states. In [40], the generator power of each turbine in the wind farm is a function of control action and states. As a result, the generator output power, in our case, depends on the pitch control action A_t and the states s_t. This implies that the reward function r_t is a function in both A_t and s_t too.

In our case, the goal is to operate around the rated rotor speed and rated generator output power. This goal is achieved by minimizing error bands $||\Delta\omega_{rotor}||$ and $||\Delta P_{Gen}||$. Equation (3) describes the behavior of the actual reward function received. The error bands are required neither to be large, thus increasing fluctuations around rated values, nor very small, thus leading to a sparse-reward function. The target error bands for rotor speed and generator output power are chosen by trial and error as ±0.3% of the rated values. The training process based on the logic-based reward function helped the agent focus on moving the environment rotor speed and generator output power states to operate at the closest range to the rated values with minimum possible fluctuations.

3.5. F-DDPG training algorithm

Algorithm 1 explains the training algorithm of each DDPG control agent using imitation learning in the initial learning episodes. Imitation learning is achieved by collecting efficient samples from a fuzzy model predictive controller to guide and fasten the training process. The parameters and hyperparameter values of the training scenarios are selected based on various trials, as shown in Table 5. The training process for each agent is completed once all episodes have been iterated. Subsequently, at the deployment phase, the control actions of the six-trained DDPG agents are processed by a fuzzy system using the method described in Subsection 3.6.

Algorithm 1. Deep deterministic policy gradient training enhanced by imitation learning and k-step bootstrapping.

- Online critic and actor networks weights (ϕ^Q, ϕ^μ) are initialized randomly using the Xavier-Initialization [41] :

$$\phi_j^Q \text{ or } \phi_j^\mu \sim U\left[-\frac{\sqrt{6}}{\sqrt{N_j+N_{j+1}}}, \frac{\sqrt{6}}{\sqrt{N_j+N_{j+1}}}\right] \quad (2)$$

where ϕ_j^Q , ϕ_j^μ are the critic and actor networks parameters in certain layer j, respectively. N_j is the number of neurons in layer j, N_{j+1} is the number of neurons in the next layer, U is a uniform distribution.

- The critic and actor target networks weights $(\phi^{Q'}, \phi^{\mu'})$ are initialized with the same online critic and actor networks weights.
- Empty experience replay buffer R_B is initialized with length 10^6 samples as suggested in [19].
- Initialization of first $\left(f_{m_{(0)}}\right)$ and second $\left(s_{m_{(0)}}\right)$ moments values of ADAM optimization with zeros [33].
- Set the steps to look-ahead K = 10.
- for episode number = 1: E do
  - Receiving initial observation states tuple s_1
  - If (episode number < 5) → for t (step) = 1: T d
    - Exert output pitch control action A_t based on the FMPC.
    - Execute pitch control action A_t, receive next step states s_{t+1} and receive the reward r_t from the environment. Whereas the logic-based reward function is defined as:

$$r_t = \begin{cases} 20, \ if \ \left(\left||\Delta\omega_{rotor}|\right| < 0.3\% \ and \ \left||\Delta P_{Gen}|\right| < 0.3\%\right) \\ 10, \ else \ if \ \left(\left||\Delta\omega_{rotor}|\right| < 0.3\% \ or \ \left||\Delta P_{Gen}|\right| < 0.3\%\right) \\ 0, \ otherwise \\ where \ \left||\Delta P_{Gen}|\right| = \left|\frac{P_t - P_{rated}}{P_{rated}}\right| \ and \ \left||\Delta\omega_{rotor}|\right| = \left|\frac{\omega_t - \omega_{rated}}{\omega_{rated}}\right| \end{cases} \quad (3)$$

      The choice of 20, 10, and 0 values is based on fine-tuning.
    - Store (s_t, A_t, r_t, s_{t+1}) transition in R_B for purpose of collecting guided samples.
  - Else → for t = 1: T do
    - For action exploration, Ornstein-Uhlenbeck random noise process formula is used:

$$O_{(t)} = O_{(t-1)} + \theta * \left(\mu - O_{(t-1)}\right) * T_s + \sigma_{(t)} * W_{(t)} \quad (4)$$

$$\sigma_\delta = 1 - 10^{\frac{log_{10}(0.5)}{HLT}} \rightarrow \sigma_{decayed} = \sigma_{(t)} * (1 - \sigma_\delta) \quad (5)$$

$$\sigma_{(t+1)} = max\left(\sigma_{decayed}, \sigma_{min}\right) \quad (6)$$

    Where $O_{(t-1)}$ is the noise value at the previous step, μ is the noise mean value, θ is the mean attraction constant, T_s is the sampling interval, $\sigma_{(t)}$ noise standard deviation, $W_{(t)}$ is wiener process, σ_δ is the constant value representing the standard deviation decay rate, HLT is the half-life time for noise standard deviation (as number of samples) , $\sigma_{decayed}$ is decayed noise standard deviation, $\sigma_{(t+1)}$ next-step noise standard deviation, σ_{min} is the minimum value of noise standard deviation. The suggested values for the parameters are shown in Table 5.

    - Initialization of Ornstein-Uhlenbeck random noise process parameters is created only at the first time step $(t_0 = 1)$ and it is mentioned in Table 5.
    - Exert output pitch control action, based on random noise and actor policy at the instant timestep:

$$A_t = clip(\ \mu(s_t|\phi^\mu) + \ O_{(t)}, \ A_{min}, \ A_{max}) \quad (7)$$

(continued on next page)

3.6. Fuzzy logic of F-DDPG

One of the inherent challenges with the model-free control strategy, as utilized by the DDPG algorithm for wind turbine pitch control, is the complexity of conducting systematic analysis. This limitation arises because such analysis traditionally relies on detailed mathematical modelling of the system being controlled [31]. To overcome such a challenge, a fuzzy system is implemented to offer a smooth transition between different trained agents to handle the perturbations of the wind inflow and uncertainties of WT dynamics in order to ensure the controller's robustness at all operating conditions. Due to a lack of general-purpose design methodologies, the design of fuzzy control

(*continued*)

Where $\mu(s_t|\phi^{\mu})$ is the action taken at the current state s_t as per the online actor network with parameters ϕ^{μ}. A_{min} and A_{max} are the minimum and maximum permissible pitch angles.

- Execute pitch control action A_t, receive next step states s_{t+1} and receive the reward r_t from the environment.
- Store (s_t, A_t, r_t, s_{t+1}) transition in R_B.
- Transitions tuples are sampled randomly from R_B based on M number of minibatch with at least K-consecutive-steps sequence of transitions i.e., zero transition (s_x, A_x, r_x, s_{x+1}), first transition $(s_{x+1}, A_{x+1}, r_{x+1}, s_{x+2})$, … K transition $(s_{x+K}, A_{x+K}, r_{x+K}, s_{x+K+1})$
- Calculate the K-step Q-target return (y_x) (target expected future reward) which takes into account a sum of discounted future rewards plus the value of the future state-action pair estimated by the target critic network. For a given sample x^{th} in the mini-batch, y_x could be formulated as:

$$y_x = r_x + \gamma r_{x+1} + \gamma^2 r_{x+2} + \cdots + \gamma^{K-1} r_{x+K-1} + \gamma^K Q'\left(s_{x+K}, \mu'\left(s_{x+K}|\phi^{\mu'}\right)|\phi^{Q'}\right) \quad (8)$$

where $r_x, r_{x+1}, \ldots, r_{x+K-1}$ are the actual rewards at each step from x to $x+K-1$, γ is the discount factor, $\mu'\left(s_{x+K}|\phi^{\mu'}\right)$ is the action taken at the state s_{x+K} as per the target actor network with parameters $\phi^{\mu'}$. $Q'\left(s_{x+K}, \mu'\left(s_{x+K}|\phi^{\mu'}\right)|\phi^{Q'}\right)$ is the Q-value of taking that action at s_{x+K} according to the target critic network with parameters $\phi^{Q'}$.

- The loss function for the critic network is formulated by employing the MSE as the metric for evaluating the discrepancy between predicted and target values, and incorporating L2 regularization to prevent overfitting and enhance model generalization:

$$L_t\left(\phi_{ij}^{Q}\right) = \frac{1}{M}\left(\sum_{x=1}^{M}\left(Q\left(s_x, A_x|\phi_{ij}^{Q}\right) - y_x\right)^2\right) + \lambda_Q \sum_{i=action\ path}^{main\ path} \sum_{j=1}^{P} ||\phi_{ij}^{Q}||^2 \quad (9)$$

Where M is the number of minibatch samples, P is the number of layers in each path, λ_Q is the L2 regularization factor, $Q\left(s_x, A_x|\phi_{ij}^{Q}\right)$ is the estimated Q-value of taking action A_x at state s_x as per online critic network.

- Calculate the gradient descent of the loss function $L\left(\phi_{ij}^{Q}\right)$ with respect to the parameters $\left(\phi_{ij}^{Q}\right)$ for minimization purpose:

$$\nabla_{\phi_{ij}^{Q}} L_t = \left(\frac{2}{M}\right)\left(\sum_{x=1}^{M}\left(Q\left(s_x, A_x|\phi_{ij}^{Q}\right) - y_x\right) * \nabla_{\phi_{ij}^{Q}} Q\left(s_x, A_x|\phi_{ij}^{Q}\right)\right) + 2 * \lambda_Q * \left|\left|\phi_{ij}^{Q}\right|\right| \quad (10)$$

- critic network parameters (ϕ^{Q}) are updated using ADAM optimization technique:
  - update the first $f_{m_{\phi^Q}}$ and second $s_{m_{\phi^Q}}$ moments estimate then correct the bias:

$$f_{m(t)_{\phi^Q}} = \beta_1 f_{m(t-1)_{\phi^Q}} + (1-\beta_1) * (\nabla_{\phi_{ij}^{Q}} L_t) \rightarrow \hat{f}_{m(t)_{\phi^Q}} = \frac{f_{m(t)_{\phi^Q}}}{(1-\beta_1^t)} \quad (11)$$

$$s_{m(t)_{\phi^Q}} = \beta_2 s_{m(t-1)_{\phi^Q}} + (1-\beta_2) * \left(\nabla_{\phi_{ij}^{Q}} L_t\right)^2 \rightarrow \hat{s}_{m(t)_{\phi^Q}} = \frac{s_{m(t)_{\phi^Q}}}{(1-\beta_2^t)} \quad (12)$$

Where β_1, β_2 are the exponential decay rates for first and second moment estimates, respectively. $f_{m(t)_{\phi^Q}}$, $s_{m(t)_{\phi^Q}}$ are the first and second moment estimates of the critic network parameters.

(*continued on next page*)

systems is typically done using heuristic procedures. The design is based on the measure of disparity between the wind turbine dynamics due to perturbations in the external wind inflow. Based on the gap-metric correlation matrix between the models stated in Table 4, six operating points at mean wind speeds of (14, 16, 18, 20, 22, and 24) m/s are chosen for training the DDPG agents. The optimal control action of F-DDPG is obtained based on a fuzzy system which processes the six-trained DDPG agents' control actions. The triangular membership function with a 0.5 cut is used [42]. Fig. 4 illustrates the shape of the membership function where the centers of the membership functions are the average wind speeds. The interference of control action regions between each of two consecutive wind speed breakpoints represents

(*continued*)

$\hat{f}_{m(t)_{\phi^Q}}, \hat{s}_{m(t)_{\phi^Q}}$ are the corrected first and second moments respectively.

- update the parameters ϕ^Q for each layer as follows:

$$\phi_t^Q = \phi_{t-1}^Q - \alpha_Q \left(\frac{\hat{f}_{m(t)_{\phi^Q}}}{\sqrt{\hat{s}_{m(t)_{\phi^Q}}} + \epsilon} \right) \tag{13}$$

where α_Q is the learning rate of critic network, ϵ is a numerical factor to avoid division by zero.

- The objective function J_t which represents the cumulative reward for the actor network parameterized by ϕ^μ is defined as:

$$J_t\left(\mu(s_x|\phi_j^\mu)\right) = \left(\frac{1}{M}\right)\sum_{x=1}^{M} Q\left(s_x, \mu(s_x|\phi_j^\mu)|\phi^Q\right) + \lambda_\mu \sum_{j=1}^{U} ||\phi_j^\mu||^2 \tag{14}$$

Where U is the total number of layers in actor policy network, $\mu(s_x|\phi^\mu)$ is the action taken at the state s_x as per the online actor network with parameters ϕ_j^μ, $Q\left(s_x, \mu(s_x|\phi_j^\mu)|\phi^Q\right)$ is the estimated Q-value of taking that action at state s_x according to the online critic network with parameters ϕ^Q. λ_μ is the L2 regularization factor of actor network to avoid overfitting.

- Calculate the gradient ascent of the objective function $J_t\left(\mu(s_x|\phi_j^\mu)\right)$ with respect to the parameters (ϕ_j^μ) based on chain rule for maximization purposes:

$$\nabla_{\phi_j^\mu} J_t = \left(\frac{1}{M}\right)\sum_{x=1}^{M} \nabla_{\mu(s_x)} Q(s_x, \mu(s_x|\phi^\mu)|\phi^Q) \nabla_{\phi_j^\mu} \mu(s_x|\phi^\mu) + 2 * \lambda_\mu * \left|\left|\phi_j^\mu\right|\right| \tag{15}$$

- Actor network parameters ϕ^μ for each layer are updated using ADAM optimization technique.
- Smooth update for both actor and critic target networks parameters $(\phi_t^{\mu'}, \phi_t^{Q'})$ respectively using smoothing factor τ:

$$\phi_t^{Q'} = \tau\phi_t^Q + (1-\tau)\phi_{t-1}^{Q'} \tag{16}$$

$$\phi_t^{\mu'} = \tau\phi_t^\mu + (1-\tau)\phi_{t-1}^{\mu'} \tag{17}$$

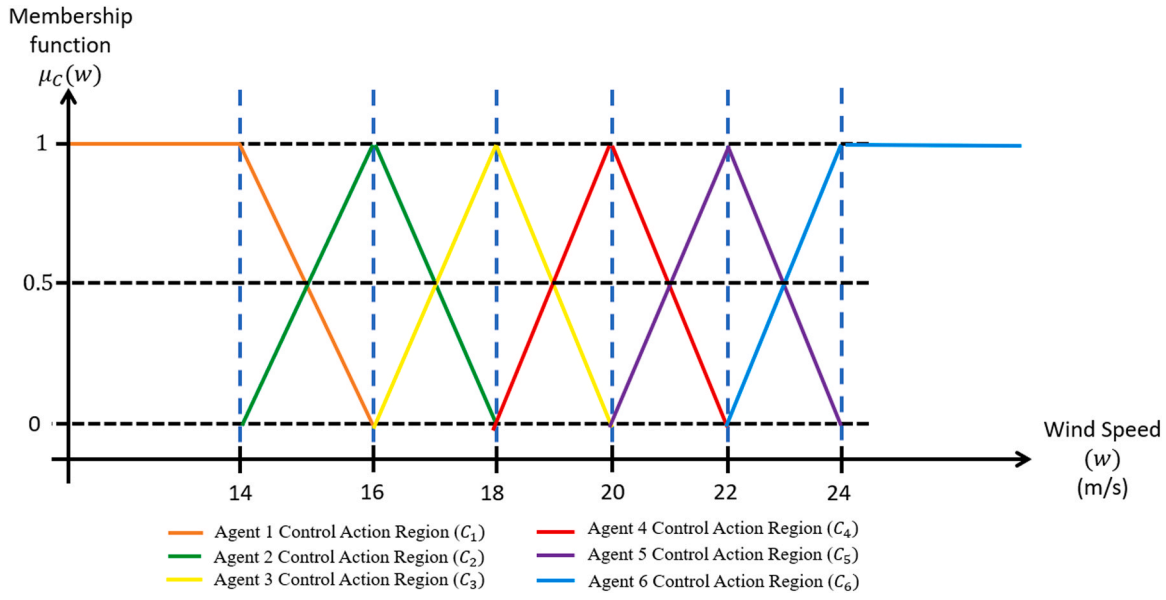

Fig. 4. Triangular-membership fuzzy logic function of the optimal control action.

Table 6
Comparison between the cumulative reward of using FMPC and trained F-DDPG agents over 300 episodes.

Average Wind Speed (m/s)	Cumulative reward of using FMPC	Cumulative Reward of DDPG Agents during training	Percentage of performance Enhancement (%)	Max episode reward of DDPG Agents
14	62472	82148	31.49	97720
16	58319	69479	19.14	77680
18	53464	63995	19.70	74850
20	45683	56362	23.38	64760
22	37931	52571	38.60	62590
24	32612	43915	34.66	52340

how the ratio of each agent control action is selected. Based on the instantaneous wind speed, the optimal control action is defuzzied according to the mean. The fuzzy control law C_f represents the input collective pitch control action to the wind turbine model, as shown in Fig. 3.

4. Simulation results

The following results are based on high-fidelity 5-MW wind turbine models from the National Renewable Energy Lab's Open-FAST package, integrated with MATLAB/Simulink, specifically in region three operation mode [5]. Open-FAST, a tool certified by the National Renewable Energy Lab, accurately reflects real-world conditions. The "OC3-Hywind spar buoy" offshore and the 5 MW onshore turbine models are utilized, representing advanced wind turbine simulation scenarios. The OC3-Hywind model, representing floating offshore turbines, is particularly noted for its complex dynamics behavior. In addition to complex hydrodynamics, the model combines high-turbulent wind and external wave conditions. The simulations encompass two scenarios: the first with an offshore turbine using the IEC Kaimal wind profile, designed for high-turbulent oceanic conditions, and the second with an onshore turbine employing the Great Plains Low-Level Jet (GP_LLJ) wind profile for moderate turbulence. Both scenarios' stochastic wind profiles are 3-dimensional models which are exported from running the Turbsim tool [43]. The wind profiles for both cases encompass stochastic wind speeds from 12 m/s to 25 m/s to demonstrate a generalized and realistic version of results for all potential wind speeds in region 3. The durations for both scenarios are 3000 s

This section summarizes the proposed F-DDPG Controller training

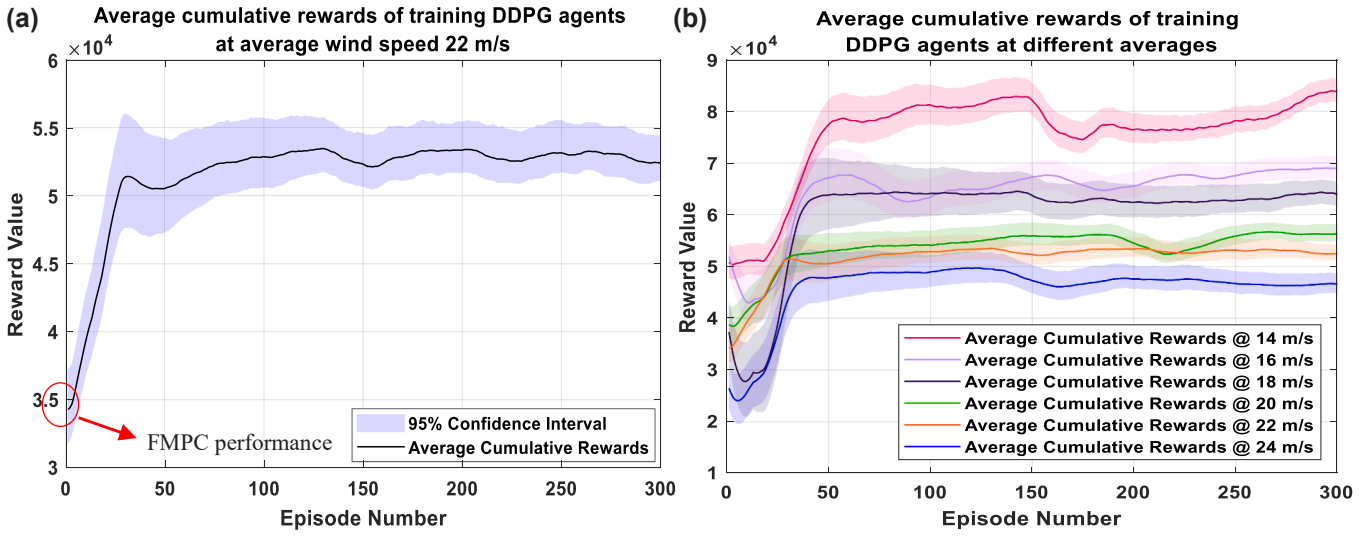

Fig. 5. Average cumulative rewards of the DDPG agent training with 95% confidence intervals. (a) Average cumulative rewards at an average wind speed 22 m/s. (b) Average cumulative rewards at different wind speed operating points.

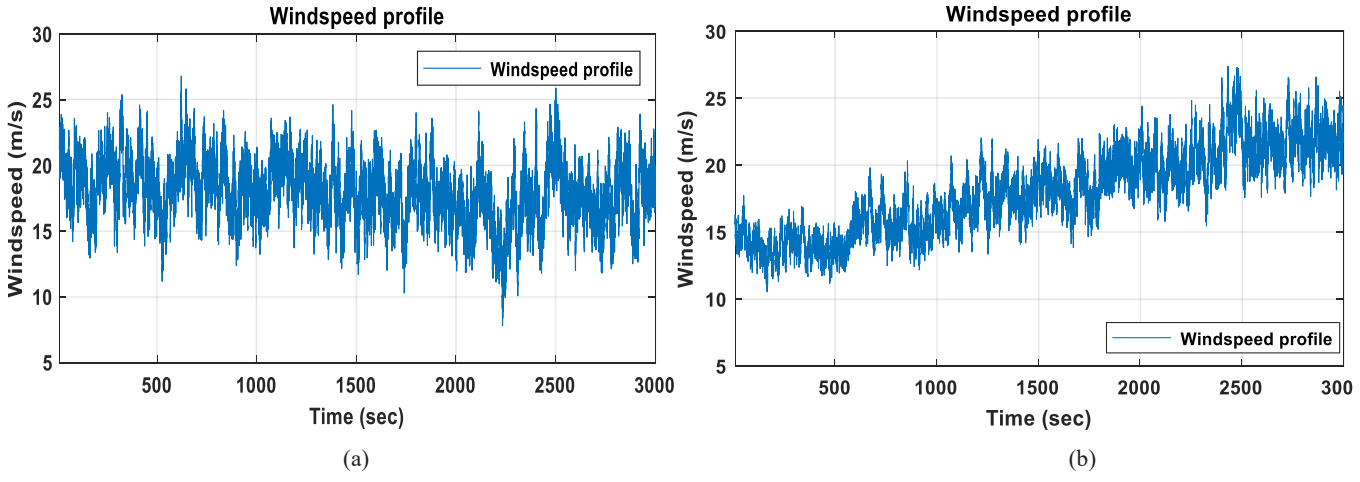

Fig. 6. Wind profile turbulence models. (a) IEC Kaimal high-turbulent (Scenario I offshore) (b) GP_LLJ moderate-turbulent (Scenario II onshore).

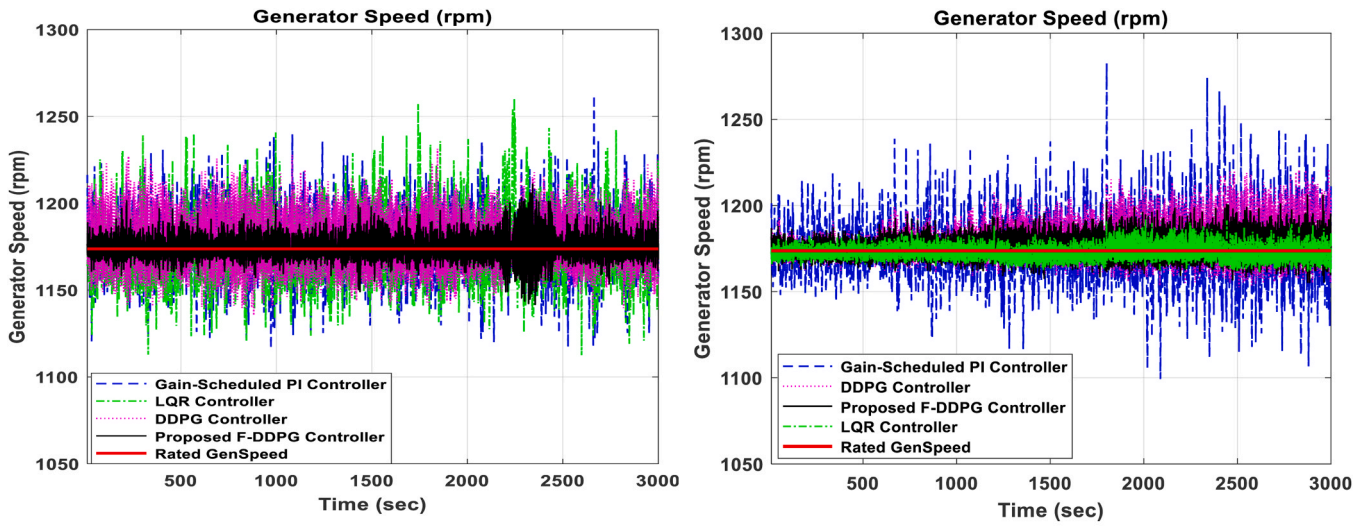

Fig. 7. Comparison of different controllers concerning generator speed. Rated generator speed = 1173.7 rpm. (a) Scenario I (offshore model) (b) Scenario II (onshore model).

Table 7
Statistical analysis of the acquired results for both scenarios.

Controllers Types		Scenario I (offshore model test)				Scenario II (onshore model test)			
		GSPI	LQR	DDPG	**F-DDPG**	GSPI	LQR	DDPG	**F-DDPG**
Generator speed (rpm)	Mean	1173.855	1172.65	1178.446	**1174.649**	1174.428	1172.786	1178.879	**1175.103**
	|Mean value deviation| %	0.013	0.089	0.404	**0.081**	0.062	0.078	0.441	**0.119**
	STD	17.760	17.92	12.902	**4.920**	18.920	3.166	8.011	**4.605**
	Peak-to-Peak	145.496	147.838	96.176	**69.288**	183.086	31.029	68.558	**50.625**
	Min	1116.471	1112.543	1135.313	**1141.723**	1099.287	1160.828	1152.562	**1155.361**
	Max	1261.967	1260.381	1231.489	**1211.011**	1282.373	1191.858	1221.120	**1205.986**
Electric power (kW)	Mean	4709.243	4669.984	4876.033	**4945.377**	4709.963	4913.639	4981.652	**4955.265**
	|Mean value deviation| %	5.815	6.600	2.479	**1.092**	5.801	1.727	0.367	**0.895**
	STD	433.242	432.279	295.920	**131.103**	428.580	103.013	117.228	**102.693**
	Peak-to-Peak	1857.317	1887.57	1546.307	**1396.048**	2104.231	728.718	1264.756	**1061.629**
	Min	3518.702	3481.697	3699.878	**3762.898**	3358.719	4348.637	3937.255	**4075.909**
	Max	5376.019	5369.267	5246.186	**5158.946**	5462.950	5077.355	5202.011	**5137.539**
Pitch angle command (rad)	Mean	0.254	0.258	0.252	**0.253**	0.267	0.257	0.259	**0.257**
	STD	0.042	0.045	0.057	**0.047**	0.073	0.076	0.078	**0.077**
	Peak-to-Peak	0.276	0.379	0.421	**0.399**	0.367	0.439	0.462	**0.466**
	Min	0.108	0.008	0.000	**0.000**	0.062	0.001	0.000	**0.000**
	Max	0.384	0.387	0.421	**0.399**	0.428	0.449	0.462	**0.466**
Optimality indexcumulative reward (x10^5)		5.704	5.507	7.078	**24.515**	6.945	21.883	15.128	**22.771**
Expected annual energy production (GWh)		16.916	16.775	17.514	**17.764**	9.905	10.330	10.476	**10.421**
Expected Energy sales in US dollar		1860760	1845250	1926540	**1954040**	1089550	1136300	1152360	**1146310**

results in subsection 4.1. Subsections 4.2 and 4.3 are dedicated to performance analysis for the first and second testing case scenarios. They compare the proposed F-DDPG controller's performance alongside other controllers, including GSPI [44], LQR [45], and a single-DDPG-agent (DDPG controller) trained at an average wind speed of 18 m/s [19].

4.1. Proposed F-DDPG controller training results

The training phase involves the development of six distinct DDPG agents within a medium-fidelity onshore environment. Each agent focuses on a specific average wind speed. The number of episodes in the training of each agent is 300. The episode duration is 100 s. The agent with the highest reward, as indicated in Table 6, is selected among the different episodes. A stochastic wind profile of 30000 s (8.33 h) is generated for training. The wind medium-turbulence model used in the training phase is the National Wind Technology Center model [43].

The cumulative rewards gathered by DDPG agents during training at different wind speed profiles are evaluated against the total rewards earned when FMPC runs throughout the same profiles. The performance enhancement indicator (PEI) in percentage, which refers to performance development, is calculated based on the following formula:

$$PEI = \frac{J_{F-DDPG} - J_{FMPC}}{J_{FMPC}} * 100 \quad (2)$$

J_{F-DDPG} and J_{FMPC} are the cumulative rewards of F-DDPG and FMPC, respectively. F-DDPG has outperformed the FMPC in terms of cumulative rewards. Fig. (5-a) displays a sample learning curve during the training of a DDPG agent across all episodes, specifically at an average wind speed of 22 m/s. The training is repeated 10 times and the 95% confidence interval is calculated based on the z-score and marginal error. Initially, the first five episodes utilized FMPC for control guidance, achieving an estimated cumulative reward of 35,000. Subsequently, the DDPG agent took over control actions completely and reached an estimated cumulative reward of 52,000 by the end of the training episodes. The training of the agent quickly converged, with the cumulative reward reaching a consistent optimum level after roughly 25 episodes. Table 6

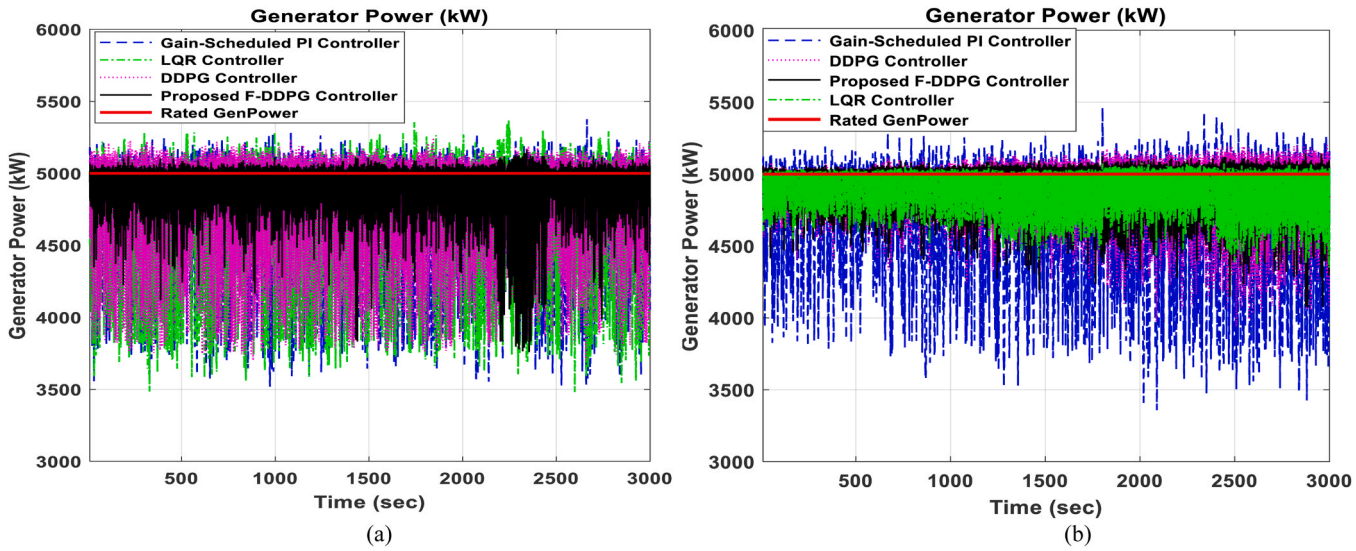

Fig. 8. Comparison of different controllers concerning generator power. Rated generator power = 5000 kW. (a) Scenario I (offshore model) (b) Scenario II (onshore model).

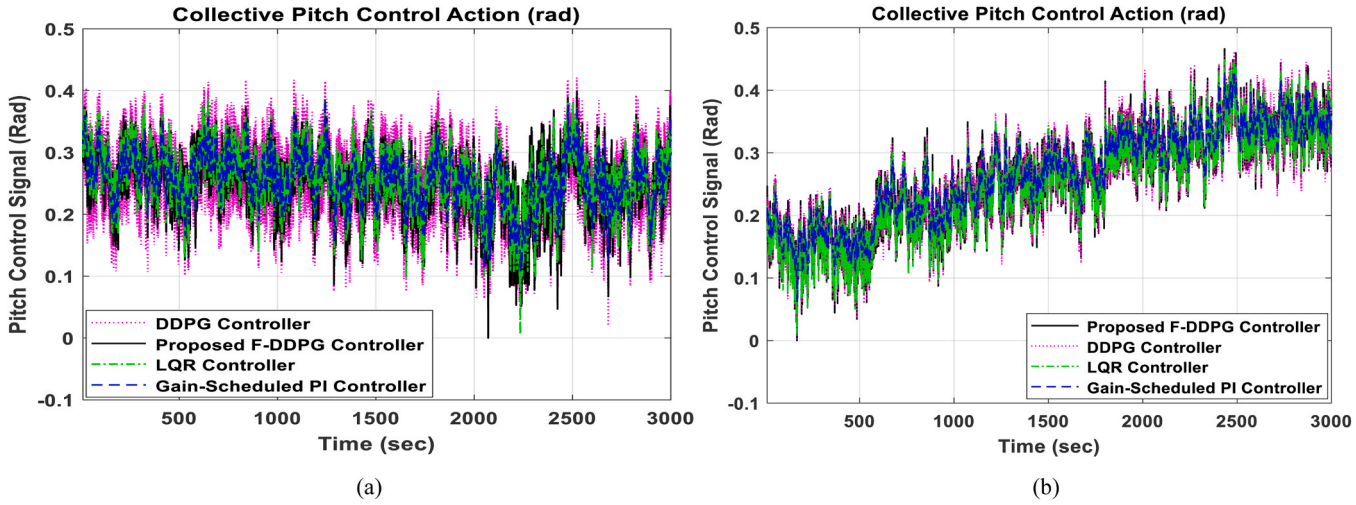

Fig. 9. Comparison of different controllers concerning the pitch angle control signals applied on 3-blades at all durations. (a) Scenario I (offshore model). (b) Scenario II (onshore model).

and Fig. (5-b) summarize the training process of the six DDPG agents at various average wind speeds.

The training of the DDPG agents achieved different levels of cumulative reward functions at different wind speeds, as depicted in Fig. (5-b). This variation, along with the gap-metric analysis, suggests we are essentially managing six distinct WT systems. Consequently, the integration of a fuzzy system becomes crucial. It blends the control actions from all trained agents, ensuring the optimal control action is effectively applied across all operating conditions.

4.2. Scenario I performance analysis (offshore test)

In this scenario, the simulations have been conducted on the high-fidelity Open-FAST floating offshore spar buoy WT model. The IEC Kaimal wind profile, chosen for its highly turbulent nature over ocean surfaces, spans all wind speed ranges in region three and is simulated for a duration of 3000 s, as depicted in Fig. (6-a). To assess the performance of the proposed F-DDPG controller and other controllers, we focus on six key aspects: generator speed, generator output power, collective pitch control signals, validation tests, optimality index, and economic feasibility.

Regarding the generator speed, the proposed F-DDPG controller demonstrates notable proficiency in stabilizing the generator speed at its rated value, exhibiting fewer fluctuations compared to the GSPI, LQR, and DDPG controllers. This is evident in Fig. (7-a) and is summarized in Table 7, where the achieved standard deviation (STD), peak-to-peak error, and maximum values are consistently lower for the proposed controller. Despite all controllers operating within similar rpm ranges, the F-DDPG controller maintains a mean generator speed closer to the rated value, with only a minor deviation of 0.081% less than the LQR and DDPG controllers but slightly higher than the GSPI. Notably, the DDPG controller displays the highest mean value deviation among the controllers, as indicated in Table 7, while the LQR achieves the highest STD compared to the other controllers.

Concerning generator output power, the proposed F-DDPG controller exhibits superior performance in sustaining the generator power at its rated value, exhibiting fewer fluctuations than alternative controllers. As illustrated in Fig. (8-a), the F-DDPG controller optimally refines power generation, maintaining fluctuations within the range of [4500–5000] kW along almost all wind profile periods. This stands in contrast to other controllers, which experience fluctuations in the range

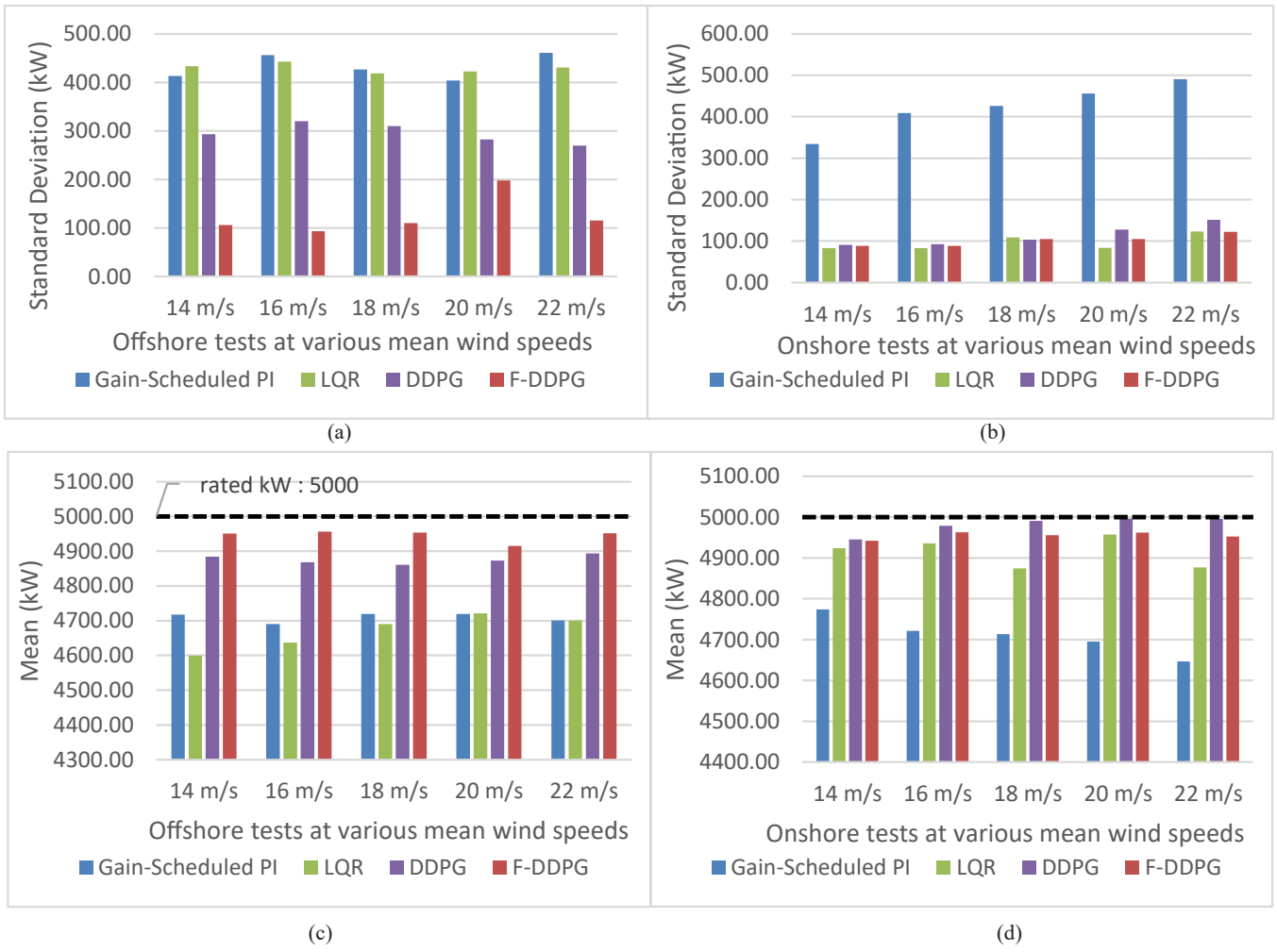

Fig. 10. Comparison of different controllers concerning generator power at various average wind speeds. (a, b) STD value at offshore and onshore tests respectively. (c, d) Mean value at offshore and onshore tests respectively.

of [3500–5100] kW. Notably, the proposed controller achieves the highest mean extracted power value, accompanied by the lowest STD, peak-to-peak error, and maximum values when compared to alternative controllers.

Concerning the collective pitch control signals, as depicted in Fig. (9-a) and further stated in Table 7, it is evident that the collective pitch control signals from both the proposed F-DDPG controller and other controllers follow a remarkably similar pattern. Notably, the STD is highest for the DDPG controller. However, it should be highlighted that the F-DDPG controller exhibits marginally more fluctuations than the GSPI controller and LQR.

Validation tests have been conducted using five distinct wind profiles, each lasting 600 s, at varying average wind speeds. The proposed and alternative controllers are evaluated across these wind profiles to affirm the proposed controller's effectiveness under diverse wind conditions. Regarding generator power, as illustrated in Figs. (10-a) and (10-c), the proposed controller consistently demonstrates the lowest standard deviation (STD) and highest average extracted power across all tested conditions. Regarding generator speed, Fig. (11-a) shows the proposed controller achieving the lowest STD, while Fig. (11-c) indicates its mean speed closely aligns with the rated value. Conversely, the DDPG controller exhibits the most significant deviation in mean value from the rated speed, which is undesirable for WT system operation.

The metric utilized to assess the performance of various controllers in terms of optimality is known as the optimality index. This index is derived from the reward function, which is designed to be maximized. Essentially, it represents the total reward accumulated by each controller over the entire simulation duration of 3000 s. As indicated in Table 7, the proposed controller has attained the highest value on this index, signifying its superior performance and greater optimality compared to other controllers.

Regarding the economic aspects of WT operations, we refer to the European Wind Energy Association's data, which indicates an average capacity factor of 41% for offshore turbines [46]. Under the assumption of the consistent wind profile shown in Fig. (6-a) throughout the year, our analysis reveals that the F-DDPG controller yields the highest estimated average annual energy production of 17.764 GWh compared to other controllers. With the levelized energy cost set at 11 US cents per kWh [47], the expected annual energy sales are calculated as shown in Table 7. This analysis ensures the economic viability and superiority of the proposed F-DDPG controller in offshore wind turbine applications.

4.3. Scenario II performance analysis (onshore test)

In this particular scenario, simulations have been conducted using the high-fidelity Open-FAST model of an onshore WT. This subsection presents the outcomes of applying various controllers to this onshore WT model. The employed wind profile, known as GP_LLJ, is characterized by moderate turbulence. It is implemented at different stochastic averages of wind speed operating points to cover all possible wind speed ranges in region 3. The duration of the total wind profile is 3000 s, as depicted in Fig. (6-b). To assess the performance of the proposed F-DDPG controller

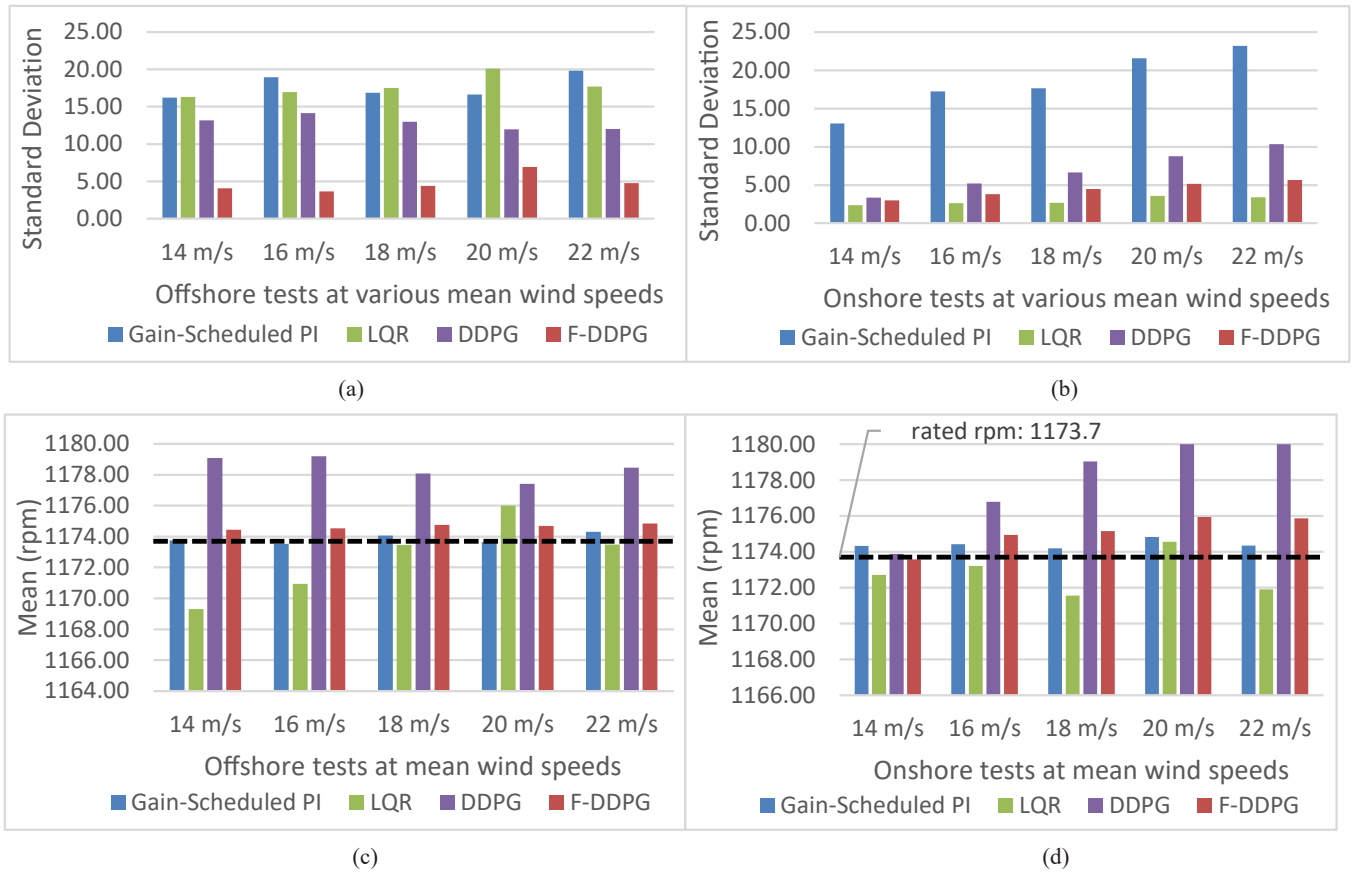

Fig. 11. Comparison of different controllers concerning generator speed at various average wind speeds. (a, b) STD value at offshore and onshore tests respectively. (c, d) Mean value at offshore and onshore tests respectively.

Table 8
Performance enhancement percentages of the proposed controller over other controllers for both scenarios.

Enhancement Ratio (%) $= \frac{\lvert Controller_{value} - F-DDPG_{value} \rvert}{Controller_{value}} * 100$		Scenario I (offshore model test)			Scenario II (onshore model test)		
		GSPI Controller	LQR Controller	DDPG Controller	GSPI Controller	LQR Controller	DDPG Controller
Generator speed (rpm)	Mean value deviation	-	8.989	79.951	-	-	73.016
	STD	72.297	72.545	61.866	75.661	-	42.517
	Peak-to-Peak	52.378	53.132	27.957	72.349	-	26.157
	Min	2.262	2.623	0.565	5.101	-	0.243
	Max	4.038	3.917	1.663	5.957	-	1.239
Electric power (kW)	Mean extracted power	5.014	5.897	1.422	5.208	0.847	-
	Mean value deviation	81.221	83.455	55.949	84.572	48.176	-
	STD	69.739	69.672	55.696	76.038	0.311	12.399
	Peak-to-Peak	24.835	26.040	9.717	49.548	-	16.061
	Min	6.940	8.077	1.703	21.353	-	3.522
	Max	4.038	3.917	1.663	5.957	-	1.240
Optimality index enhancement %		329.786	345.161	246.355	227.876	4.058	50.522
Annual energy production improvement %		5.013	5.896	1.427	5.209	0.881	-
Energy sales improvement in US dollar		93280	108790	27500	56760	10010	-

and other controllers, we focus on the same aspects used in Scenario I.

Regarding the generator speed, the F-DDPG and LQR controllers demonstrate a notable reduction in fluctuations, effectively sustaining the generator speed at its rated values, as illustrated in Fig. (7-b). Conversely, the GSPI controller exhibits the highest standard deviation (STD) and the most significant fluctuations among the controllers, with the DDPG controller following closely behind, as detailed in Table 7. Furthermore, the DDPG controller shows the most significant mean value deviation from the rated value compared to its counterparts, signalling a less optimal performance in this specific aspect.

Regarding generator power, both the DDPG and F-DDPG controllers excel, achieving the highest average power extraction, as indicated in Table 7. Meanwhile, the F-DDPG and LQR controllers exhibit the lowest fluctuations in generator power, as demonstrated in Fig. (8-b). On the other hand, the GSPI controller shows the most significant fluctuations, as evidenced by its high STD value, and it also records the lowest power extraction. This combination of factors suggests that the GSPI controller's performance is less effective compared to the other controllers in this aspect. Regarding the collective pitch control signals, as illustrated in Fig. (9-b) and further detailed in Table 7, it becomes clear that the signals from the proposed F-DDPG controller and those from alternative controllers display a similar pattern.

Table 9
Ranking of controllers' performance in different operational conditions.

Controllers Types		Scenario I (offshore model test)				Scenario II (onshore model test)			
		GSPI	LQR	DDPG	**F-DDPG**	GSPI	LQR	DDPG	**F-DDPG**
Operational aspects	Maintaining generator speed close to rated	4	2	1	**3**	4	2	1	**3**
	Generator speed fluctuations handling	2	1	3	**4**	1	4	2	**3**
	Maximizing generator output power	2	1	3	**4**	1	2	4	**3**
	Generator power fluctuations handling	1	2	3	**4**	1	3	2	**4**
	Dealing with hydrodynamics and waves external perturbations in case of offshore	2	1	3	**4**	-	-	-	**-**
	Optimality	2	1	3	**4**	1	3	2	**4**
Total scored points		13	8	16	**23**	8	14	11	**17**

Table 10
Prospective applications subjected to the proposed approach.

Environment	Prospective observations	Prospective reward function	Prospective target
Autonomous Vehicle Systems	Position, velocity, acceleration, proximity to obstacles	Safe navigation, minimal traffic rule violations, and efficient path planning.	Proposed controller could handle complex driving scenarios, nonlinear vehicle dynamics, and unpredictable road conditions.
Robotic Manipulators in Manufacturing	Joint angles, velocities, torque, end-effector position, and object properties	Precision in object manipulation, energy efficiency, speed of operation, and safety	The proposed controller is useful in controlling robotic arms with complex, nonlinear dynamics and requiring precision in dynamic environments.
Smart Grid Energy Management	Energy demand, supply levels, grid stability metrics, and stochastic weather conditions	Balance between supply and demand, grid stability, cost-effectiveness, and renewable energy utilization.	The proposed controller could manage the variable nature of renewable energy sources and fluctuating demand.

Validation tests employing five distinct wind profiles, each with a duration of 600 s and varying average wind speeds, were executed. These tests were designed to evaluate the proposed F-DDPG controller and its alternatives under a range of wind conditions, thereby verifying the effectiveness of the proposed controller. In terms of generator power, as illustrated in Fig. (10-b), the F-DDPG, along with the LQR and DDPG controllers, exhibit notably low standard deviations, indicating minimal fluctuations. In contrast, the GSPI controller shows the highest level of variability. Furthermore, as shown in Fig. (10-d), both the F-DDPG and DDPG controllers achieve the highest mean power generation, with the DDPG controller slightly outperforming. Concerning generator speed, depicted in Fig. (11-b), the F-DDPG and LQR controllers are observed to have the lowest standard deviation values, indicating superior stability. Additionally, as presented in Fig. (11-d), the F-DDPG and GSPI controllers are noteworthy for having the slightest mean value deviations from the rated speed, setting them apart from the other controllers.

In terms of optimality, as shown in Table 7, the proposed controller has attained the highest value on this index, signifying its superior performance and greater optimality compared to other controllers. Regarding the economic aspects of WT operations, we refer to the European Wind Energy Association's data, which indicates an average capacity factor of 24% for onshore turbines [46]. Under the assumption of the consistent wind profile shown in Fig. (6-b) throughout the year, our analysis reveals that the proposed F-DDPG and DDPG controllers yield the highest estimated average annual energy production of 10.421 and 10.476 GWh compared to other controllers. The expected annual energy sales are calculated with a levelized cost of 7 US cents per kWh [47], as shown in Table 7.

5. Discussion

The proposed F-DDPG controller shows promising performance in effective collective pitch control for WT systems. Its efficacy is benchmarked against three distinct controllers: GSPI, LQR, and DDPG, each exhibiting unique strengths and weaknesses. GSPI excels in keeping the generator speed nearly at its rated value but falls short in reducing fluctuations and enhancing output power. The LQR, an optimal controller designed at a reduced-order WT system, excels in reducing fluctuations and ensuring stability, particularly in onshore conditions. However, its effectiveness decreases in offshore models with added hydrodynamic challenges and external wave perturbations. DDPG, an RL controller, is trained at a median wind speed of 18 m/s, within the wind speed range (12 m/s to 24 m/s). While it maximizes output power in moderately turbulent onshore winds, it struggles to maintain rated generator speed and faces increased fluctuations and diminished power generation in highly turbulent offshore conditions. The proposed F-DDPG controller marks a significant improvement in these areas, effectively maintaining generator speed near its rated value in both moderate and highly turbulent profiles, thanks to the fuzzy system which smooths the DDPG agents' transitions, thereby enhancing the controller's robustness. It also reduces generator speed and power fluctuations, adeptly manages new and complex hydrodynamics, including external wave and current disturbances, and achieves greater optimality. These strengths and weaknesses are highlighted by the enhancement ratios of the proposed F-DDPG controller compared to others, as detailed in Table 8. Controllers are also ranked from one to four in various aspects, with one being the lowest and four the highest, as presented in Table 9. These tables show that the F-DDPG controller is generally superior in different operational conditions, as reflected in its total scored points.

The proposed method has its strengths in terms of design and operation. First, imitation learning is used to provide the RL agent with highly efficient samples during the initial training episodes, which helps in faster convergence, reaching a consistent optimal level of training. Second, the actor-critic networks topology modification results in a reduction of the training computational cost by a factor of 1.6 from the standard DDPG networks topology. Third, the robustness of the controller is ensured by measuring the effect of wind inflow perturbations and uncertainties on varying the system dynamics using the gap-metric criterion. Fourth, the fuzzy system aids in providing smooth transitions and blending the control actions of different DDPG agents, achieving optimal, stable, and robust operation. Fifth, the proposed controller is trained in a medium-fidelity onshore environment. Still, it has the capability to generalize its performance in newly unconsidered complex hydrodynamics and external conditions (waves and currents), as shown in the case of the high-fidelity offshore model test.

The proposed F-DDPG controller, while effective, presents certain limitations. First, the training relies on imitation learning by collecting samples from a model-based FMPC controller. The design of the FMPC necessitates a reduced-order WT system model, which could pose technical challenges in systems lacking explicit models. Second, formulating the reward function requires a high understanding of the desirable outcome. In this study, the reward function is formulated to be logic-based, which achieves the requirements for maintaining generator speed and power close to rated values. However, this type of reward function could be challenging in other applications. Third, the DDPG training relies on the bellman equation, and the truncation of the control action during the training could limit the exploration. Due to the insights gained from the optimal control action exerted by the FMPC, we know the optimal region of control action exploration. Fourth, the reinforcement learning controllers need severe testing and validation on real-world applications to compensate for the stability proof, which is a challenging research topic. To address this issue, simulations on high-fidelity onshore and offshore cases validate the proposed controller performance to ensure its optimality and stability. The intended future work will treat all these issues and provide stability analysis proof for the RL controller.

An enormous number of control processes are subjected to the proposed approach. Examples illustrating the prospective systems in which this approach could be used are shown in Table 10. In each of these systems, the key characteristics that align with our approach include the need for real-time adaptive control, dealing with stochastic (random and unpredictable) environmental or system conditions. The combination of fuzzy logic systems and RL agents provides a powerful toolset for such scenarios, offering the flexibility to handle model uncertainty (through fuzzy logic) and the ability to learn optimal control strategies in complex, multidimensional spaces (via DDPG). In addition, any legacy controller applicable to such environments could act as a demonstrator for imitation learning in the initial learning episodes, which could speed up the convergence of the training.

6. Conclusions

This paper has suggested a novel collective pitch angle controller design depending on hybrid imitation learning and model-free RL techniques. Six DDPG agents have been trained at different wind speed profiles with averages covering region 3. The agents have been trained in a medium-fidelity environment by enabling the onshore WT degrees of freedom. In the initial training phase, the FMPC has served as a guide for imitation learning, collecting highly effective samples for storage in the experience replay buffer. These samples have allowed the agents to adopt the characteristics of the FMPC initially. Subsequently, DDPG agents have taken over to learn through direct interaction with the environment to achieve better cumulative rewards than FMPC. This hybrid technique has fastened the training, which has converged after approximately 25 episodes. The actor-critic networks topology has reduced the training computational time by 1.6 while deepening the complex patterns learned. The actor network has been trained to maximize the cumulative estimated return. The critic network has been trained to minimize the difference between the estimated and the target returns to track the actual reward's behaviour. The logic-based reward function has achieved the objective of maintaining the operation of WT at rated generator speed and power. At the deployment phase, the best six trained agents have been processed by a fuzzy system to form the optimal CPC action and to ensure controller robustness under different operating conditions. The simulations have been conducted on the Open-FAST package integrated with MATLAB/Simulink. The proposed F-DDPG controller's performance is compared to that of different controllers. The results have been validated and generalized on two case scenarios: high-fidelity 5-MW onshore and offshore models. The F-DDPG controller's performance results have indicated the generalization and robustness nature of performance, particularly in the offshore test where the challenging hydrodynamics and external conditions of waves have been activated. A significant improvement has been noticed in generator output power, energy extraction, reduction in fluctuations, and optimality.

CRediT authorship contribution statement

Abdel Latif Elshafei: Writing – review & editing, Supervision, Methodology, Conceptualization. **Essam Aboul Zahab:** Writing – review & editing, Supervision. **Abdelhamid Nabeel:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Formal analysis, Conceptualization. **Ahmed Lasheen:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] I. Renewable Energy Agency, "RENEWABLE ENERGY STATISTICS 2023 STATISTIQUES D'ÉNERGIE RENOUVELABLE 2023 ESTADÍSTICAS DE ENERGÍA RENOVABLE 2023 About IRENA," 2023. Accessed: Jan. 20, 2024. [Online]. Available: https://www.irena.org/Publications/2023/Jul/Renewable-energy-statistics-2023

[2] Global Wind Energy Council, "GWEC-2023_interactive," 2023. Accessed: Jan. 20, 2024. [Online]. Available: https://gwec.net/globalwindreport2023/

[3] Manwell JF, McGowan JG, Rogers AL. Wind Energy Explained: Theory, Design and Application. 2. Aufl. New York: Wiley; 2010. https://doi.org/10.1002/9781119994367.

[4] Burton T, Sharpe D, Jenkins N, Bossanyi E. The Wind Energy Handbook. 2001. https://doi.org/10.1002/0470846062.

[5] J. Jonkman, "OpenFAST Documentation." Accessed: Aug. 24, 2023. [Online]. Available: https://openfast.readthedocs.io/en/main/.

[6] Johnson K, Pao L, Balas M, Fingersh L. Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture. Cont. Syst. IEEE Aug. 2006;vol. 26:70–81. https://doi.org/10.1109/MCS.2006.1636311.

[7] Bossanyi E ~A. The Design of closed loop controllers for wind turbines. Wind Energy Jul. 2000;vol. 3(3):149–63. https://doi.org/10.1002/we.34.

[8] Bagheri P, Behjat L, Sun Q. Nonlinear control of a class of non-affine variable-speed variable-pitch wind turbines with radial-basis function neural networks. ISA Trans 2022;vol. 131:197–209. https://doi.org/10.1016/j.isatra.2022.05.004.

[9] Parvaresh A, Abrazeh S, Mohseni S-R, Zeitouni MJ, Gheisarnejad M, Khooban M-H. A novel deep learning backstepping controller-based digital twins technology for pitch angle control of variable speed wind turbine. Design 2020;vol. 4(2). https://doi.org/10.3390/designs4020015.

[10] Chen P, Han D, Tan F, Wang J. Reinforcement-based robust variable pitch control of wind turbines. IEEE Access 2020;vol. 8:20493–502. https://doi.org/10.1109/ACCESS.2020.2968853.

[11] Sierra-Garcia JE, Santos M. Combination of neural networks and reinforcement learning for wind turbine pitch control. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Science and Business Media Deutschland GmbH; 2022. p. 385–92. https://doi.org/10.1007/978-3-031-15471-3_33.

[12] S. Qin, Y. Liu, Z. Liu, and M. Sun, "Data-based Reinforcement Learning with Application to Wind Turbine Pitch Control," in *2021 6th International Conference on Power and Renewable Energy (ICPRE)*, 2021, pp. 538–542. doi: 10.1109/ICPRE52634.2021.9635193.

[13] Xie J, Dong H, Zhao X. Data-driven torque and pitch control of wind turbines via reinforcement learning. Renew Energy Oct. 2023;vol. 215. https://doi.org/10.1016/j.renene.2023.06.014.

[14] Sierra-Garcia JE, Santos M, Pandit R. Wind turbine pitch reinforcement learning control improved by PID regulator and learning observer. Eng Appl Artif Intell May 2022;vol. 111. https://doi.org/10.1016/j.engappai.2022.104769.

[15] Tomin N. Robust reinforcement learning-based multiple inputs and multiple outputs controller for wind turbines. Mathematics Jul. 2023;vol. 11(14). https://doi.org/10.3390/math11143242.

[16] Sierra-Garcia JE, Santos M. Deep learning and fuzzy logic to implement a hybrid wind turbine pitch control. Neural Comput Appl Jul. 2022;vol. 34(13):10503–17. https://doi.org/10.1007/s00521-021-06323-w.

[17] Xie J, Dong H, Zhao X. Power regulation and load mitigation of floating wind turbines via reinforcement learning. IEEE Trans Autom Sci Eng 2023. https://doi.org/10.1109/TASE.2023.3295576.

[18] Fernandez-Gauna B, Graña M, Osa-Amilibia JL, Larrucea X. Actor-critic continuous state reinforcement learning for wind-turbine control robust optimization. Inf Sci (N Y) Apr. 2022;vol. 591:365–80. https://doi.org/10.1016/j.ins.2022.01.047.

[19] T.P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," Sep. 2015, [Online]. Available: http://arxiv.org/abs/1509.02971.
[20] Sutton RS, Barto AG. Reinforcement learning: An introduction, 2nd ed. Adaptive computation and machine learning. Cambridge, MA, US: The MIT Press; 2018.
[21] Yan H-S, Wang G-B. Adaptive tracking control for stochastic nonlinear systems with time-varying delays using multi-dimensional Taylor network. ISA Trans 2023; vol. 132:246–57. https://doi.org/10.1016/j.isatra.2022.06.004.
[22] Jin X, Ma H, Tang J, Kang Y. A self-adaptive vibration reduction method based on deep deterministic policy gradient (dDPG) reinforcement learning algorithm. Appl Sci (Switz) Oct. 2022;vol. 12(19). https://doi.org/10.3390/app12199703.
[23] Lasheen A, Elshafei AL. Wind-turbine collective-pitch control via a fuzzy predictive algorithm. Renew Energy Mar. 2016;vol. 87:298–306. https://doi.org/10.1016/j.renene.2015.10.030.
[24] Mnih V, et al. Human-level control through deep reinforcement learning. Nature Feb. 2015;vol. 518(7540):529–33. https://doi.org/10.1038/nature14236.
[25] Buşoniu L, De Bruin T, Tolić D, Kober J, Palunko I. Reinforcement learning for control: Performance, stability, and deep approximators. Annu Rev Control 2018; vol. 46:8–28. https://doi.org/10.1016/j.arcontrol.2018.09.005.
[26] Q. Zou, K. Xiong, and Y. Hou, "An end-to-end learning of driving strategies based on DDPG and imitation learning," in *2020 Chinese Control And Decision Conference (CCDC)*, 2020, pp. 3190–3195. doi: 10.1109/CCDC49329.2020.9164410.
[27] H. Xie, X. Xu, Y. Li, W. Hong, and J. Shi, "Model Predictive Control Guided Reinforcement Learning Control Scheme," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8. doi: 10.1109/IJCNN48605.2020.9207398.
[28] Fang J, et al. Wind turbine rotor speed design optimization considering rain erosion based on deep reinforcement learning. Renew Sustain Energy Rev 2022; vol. 168:112788. https://doi.org/10.1016/j.rser.2022.112788.
[29] J. Jonkman, "Definition of the Floating System for Phase IV of OC3," 2010. [Online]. Available: http://www.osti.gov/bridge.
[30] J. Jonkman, "The New Modularization Framework for the FAST Wind Turbine CAE Tool Preprint," 2013. [Online]. Available: http://www.osti.gov/bridge.
[31] R.E. Precup, R.C. Roman, and A. Safaei, *Data-driven Model-free Controllers*. CRC Press, 2021. [Online]. Available: https://books.google.com.eg/books?id=yFi2zgEACAAJ.
[32] G.S. Bir, "User's Guide to MBC3 (Multi-blade Coordinate Transformation Utility for 3-Bladed Wind Turbines)," 2008.
[33] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015. doi: https://doi.org/10.48550/arXiv.1412.6980.
[34] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016.
[35] C.C. Aggarwal, "Neural Networks and Deep Learning," *Neural Networks and Deep Learning*, 2018, doi: 10.1007/978–3-319–94463-0.
[36] S. Sharma, S. Sharma, and A. Athaiya, "ACTIVATION FUNCTIONS IN NEURAL NETWORKS," *International Journal of Engineering Applied Sciences and Technology*, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:225922639.
[37] Lu L. Dying ReLU and initialization: theory and numerical examples. Commun Comput Phys Jun. 2020;vol. 28(5):1671–706. https://doi.org/10.4208/cicp.oa-2020-0165.
[38] Ashraf N, Mostafa R, Sakr R, Rashad M. Optimizing hyperparameters of deep reinforcement learning for autonomous driving based on whale optimization algorithm. PLoS One 2021;16:e0252754. https://doi.org/10.1371/journal.pone.0252754.
[39] Sierra-García JE, Santos M. Exploring reward strategies for wind turbine pitch control by reinforcement learning. Appl Sci (Switz) Nov. 2020;vol. 10(21):1–23. https://doi.org/10.3390/app10217462.
[40] He B, Zhao H, Liang G, Zhao J, Qiu J, Dong ZY. Ensemble-based Deep Reinforcement Learning for robust cooperative wind farm control. Int J Electr Power Energy Syst 2022;vol. 143:108406. https://doi.org/10.1016/j.ijepes.2022.108406.
[41] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. J Mach Learn Res - Proc Track Aug. 2010;vol. 9:249–56.
[42] Sadollah A. Fuzzy Logic Based in Optimization Methods and Control Systems and Its Applications. Rijeka: IntechOpen; 2018. https://doi.org/10.5772/intechopen.73112.
[43] B.J. Jonkman, "Turbsim User's Guide V2.0," 2014. [Online]. Available: www.nrel.gov/publications.
[44] Hawari Q, Kim T, Ward C, Fleming J. A robust gain scheduling method for a PI collective pitch controller of multi-MW onshore wind turbines. Renew Energy 2022;vol. 192:443–55. https://doi.org/10.1016/j.renene.2022.04.117.
[45] R.M. Imran, D.M.A. Hussain, and M. Soltani, "DAC with LQR control design for pitch regulated variable speed wind turbine," in *2014 IEEE 36th International Telecommunications Energy Conference (INTELEC)*, 2014, pp. 1–6. doi: 10.1109/INTLEC.2014.6972153.
[46] European Wind Energy Association, "Wind Energy Fact Sheet." Accessed: Aug. 24, 2023. [Online]. Available: https://www.ewea.org/wind-energy-basics/facts/
[47] Morthorst PE, Kitzing L. Economics of building and operating offshore wind farms. Offshore Wind Farms: Technologies, Design and Operation. Elsevier Inc.; 2016. p. 9–27. https://doi.org/10.1016/B978-0-08-100779-2.00002-7.