Check for updates

Wind Energy

WILEY

RESEARCH ARTICLE OPEN ACCESS

Enhancing Energy Generation While Mitigating Noise Emissions in Wind Turbines Through Multi-Objective Optimization: A Deep Reinforcement Learning Approach

Martín Frutos[1] | Oscar A. Marino[1] | David Huergo[1] | Esteban Ferrer[1,2]

[1]ETSIAE-UPM-School of Aeronautics, Universidad Politécnica de Madrid, Madrid, Spain | [2]Center for Computational Simulation, Universidad Politécnica de Madrid, Campus de Montegancedo, Boadilla del Monte, Madrid, Spain

Correspondence: Martín de Frutos (m.defrutos@upm.es)

Received: 20 September 2024 | **Revised:** 31 March 2025 | **Accepted:** 5 June 2025

Funding: Esteban Ferrer and Oscar A. Marino would like to thank the support of Agencia Estatal de Investigación for the grant "Europa Excelencia" for the project EUR2022-134041 funded by MCIN/AEI/10.13039/501100011033) and the European Union NextGenerationEU/PRTR and also the funding received by the Grant DeepCFD (Project no. PID2022-137899OB-I00) funded by MICIU/AEI/10.13039/501100011033 and by ERDF, EU. This research has been cofunded by the European Union (ERC, Off-coustics, project number 101086075). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. All authors gratefully acknowledge the Universidad Politécnica de Madrid (www.upm.es) for providing computing resources on Magerit Supercomputer and the computer resources at MareNostrum and the technical support provided by Barcelona Supercomputing Center (projects RES-IM-2024-1-0003 and RES-IM-2025-1-0011). Finally, the authors gratefully acknowledge the EuroHPC JU for the project EHPC-REG-2023R03-068 for providing computing resources of the HPC system Vega at the Institute of Information Science.

Keywords: aeroacoustic | blade element momentum theory | Brooks Pope and Marcolini | deep reinforcement learning | multi-objective optimization | Q-learning | torque-pitch control | wind turbine

ABSTRACT

We develop a torque-pitch control framework using deep reinforcement learning for wind turbines to optimize the generation of wind turbine energy while minimizing operational noise. We employ a double deep Q-learning, coupled to a blade element momentum solver, to enable precise control over wind turbine parameters. In addition to the blade element momentum, we use the wind turbine acoustic model of Brooks Pope and Marcolini. Through training with simple winds, the agent learns optimal control policies that allow efficient control for complex turbulent winds. Our experiments demonstrate that reinforcement learning can find optimals at the Pareto front when maximizing energy while minimizing noise. In addition, the adaptability of the reinforcement learning agent to changing turbulent wind conditions underscores its efficacy for real-world applications. We validate the methodology using a SWT2.3-93 wind turbine with a rated power of 2.3 MW. We compare the reinforcement learning control with classic controls to show that they are comparable when noise emissions are not taken into account. When including a maximum limit of 45 dBA in the noise produced (100-m downwind of the turbine), the extracted yearly energy decreases by 22%. The methodology is flexible and allows for easy tuning of the objectives and constraints through the reward definitions, resulting in a flexible multi-objective optimization framework for wind turbine control. In general, our findings highlight the potential of RL-based control strategies to improve wind turbine efficiency while mitigating noise pollution, thus advancing sustainable energy generation technologies.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2025 The Author(s). *Wind Energy* published by John Wiley & Sons Ltd.

1 | Introduction

In recent years, the aerodynamic design of wind turbines has undergone significant advances, reaching near-optimal efficiency through substantial investments in aerodynamic optimization, as well as advancements in manufacturing techniques and materials. Consequently, the focus has shifted towards addressing the issue of noise generated by wind turbines, which is emerging as a competitive factor within the wind energy industry. Numerous studies have explored the correlation between wind turbine sound power levels and public-reported perceptions of annoyance [1, 2]. Accurate prediction of wind turbine noise under real operational and atmospheric conditions is crucial to design quieter turbines and complying with imposed noise regulations [3]. This necessity underscores the importance of fast turn-around methods for incorporating noise calculations into design and optimization processes, as well as assessing noise in real time during operation. Such efforts not only optimize wind resource utilization but also minimize the impact on the quality of life of nearby communities and wildlife.

Aerodynamic noise poses a significant limitation to further exploiting wind energy resources. This type of noise results from the turbulent flow interacting with the airframe, necessitating a detailed resolution of the flow for accurate far-field noise prediction. However, computational fluid dynamics (CFD) solvers, while capable of simulating the flow field, incur a high computational cost which escalates further when resolving the acoustic field. Consequently, numerical approaches to wind turbine noise prediction remain challenging. Therefore, most noise prediction models for wind turbines are based on aeroacoustic semi-empirical models rather than numerical simulations [4]. Despite these obstacles, wind turbines remain an essential component in the generation of clean and renewable energy. However, effective control strategies are imperative to optimize their performance under variable wind conditions.

Wind turbine control systems are designed to maximize energy generation while ensuring structural integrity and safe operation [5–7]. Given the dynamic nature of wind, adaptive control strategies are essential, with classic mechanisms including adjustments to yaw angle and rotational speed, as well as blade pitch angle modulation. Using real-time wind measurements, turbine dynamics, and advanced control algorithms enables simultaneous adjustments to rotor speed and pitch, enhancing energy generation, reducing fatigue loads, and extending turbine lifespan.

Machine learning (ML) techniques have become integral to optimizing renewable energy systems, offering advanced solutions for prediction, control, and decision-making processes. In the realm of supervised learning, support vector machines (SVMs) have been widely utilized to forecast critical parameters. For example, [8] or [9] used ensemble techniques for solar power forecasting. [10] used also SVM power prediction of turbines. Artificial neural networks (ANN) have also been pivotal as function approximators in renewable energy applications. [11] used a graph neural network (NN) to create a layout-independent wind farm load estimator. Optimization of wind farm layouts has also benefited from ML-oriented techniques. [12] applied evolutionary algorithms to optimize turbine placement, focusing on maximizing power output, while [13] extended layout optimization by incorporating noise generation constraints into the optimization process, balancing energy production with environmental noise considerations.

However, traditional ML approaches often struggle with dynamically changing wind conditions and complex interactions between turbine parameters. Thus, reinforcement learning (RL) has emerged as a powerful alternative for wind turbine control, offering adaptive decision-making capabilities and the ability to learn optimal control policies in real time [14, 15]. RL involves an agent learning to make decisions in an environment to maximize cumulative rewards over time [16]. Applied to wind turbines, RL offers autonomous learning of control inputs to maximize power generation by capturing complex nonlinear relationships between wind conditions, turbine states, and actions. RL-based control methods adapt in real time to changing wind conditions, offering significant advantages in wind turbine operation.

While significant research has been conducted on noise reduction techniques in wind turbine blade design and layout optimization for noise generation, there is a notable gap in studies integrating RL with wind turbine control that explicitly considers noise. The existing literature extensively explores RL-based wind turbine control strategies, but none addresses noise constraints as a key part of the objective. A detailed discussion of the state of the art is provided in Section 2. Therefore, the development of a silent control strategy through RL presents a novel research opportunity. This study addresses the challenge of optimizing wind turbine control to maximize energy generation while simultaneously minimizing aeroacoustic noise. To bridge this gap, we propose a multi-objective RL framework that dynamically adjusts turbine operations to achieve a balance between efficiency and noise reduction. The main research contributions of this work are as follows:

- A multi-objective RL control of wind turbines is proposed. Attempting to maximize energy generation while maintaining acceptable decibel levels.
- The methodology is validated on a SWT2.3-93 wind turbine using OpenFAST, a BEMT solver, coupled with aeroacoustic models to compute wind turbine noise.
- An effective power curve (EPC) is computed for silent control of wind turbines, enabling the estimation of annual energy generation for different locations.

The paper is structured as follows. First, we discussed the state of the art on Section 2. Then, we summarize the methodology in Section 3. There, we include the wind turbine model, validating the aeroacoustic model with three different wind conditions. Additionally, the multi-objective RL strategy is explained, and we provide details on the reward, the NN architecture, and the training procedure. In Section 4, we validate the controller with simple steady winds to later challenge the method with turbulent wind conditions obtained from experimental measurements. Finally, we characterize the new silent control strategy estimating its effective power curve and end up with conclusions and outlooks.

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2 | Related Works

Wind turbine optimization has been extensively studied from two distinct perspectives: noise reduction techniques, which focus on minimizing aeroacoustic emissions through design modifications, and RL approaches for turbine control, primarily aimed at maximizing energy efficiency and structural stability. However, these two fields have remained largely separate, and RL-based control strategies rarely incorporate noise constraints. This section reviews the existing literature on noise mitigation strategies in wind turbines and RL approaches to turbine control, highlighting the gap that motivates our work.

A widely studied technique is the use of serrated trailing edges, which modify turbulent flow structures to reduce noise generation [17]. Experimental and numerical studies confirm their effectiveness in minimizing trailing-edge noise, a dominant aerodynamic source, making them a practical solution for quieter turbine operation [18]. For a broader overview of wind turbine noise mechanisms and conventional mitigation strategies, see [19] and [20]. Recent advances in ML have further improved blade design and noise management, optimizing blade geometry to lower noise emissions while preserving aerodynamic performance. [21] utilized a genetic algorithm, whereas [22] employed autoencoders to refine blade geometry, achieving noise mitigation without compromising power output. Beyond design optimization, ML has also been applied to wind turbine noise identification, with SVM used to analyze noise measurements and classify aeroacoustic noise sources [23]. Noise-aware optimization techniques extend to broader operational and layout strategies. While traditional wind farm layouts prioritize energy production, recent studies incorporate noise constraints to minimize overall emissions [13]. Advanced models using the Jensen wake model and particle swarm optimization maximize power output while reducing noise by up to 15%, accounting for wake effects and community proximity [24].

With regard to wind turbine control, this paper reviews recent advances in RL-based strategies, focusing on pitch angle and rotor speed modulation. Previous studies have proposed RL algorithms with comprehensive reward definitions, showcasing their efficacy in optimizing wind turbine performance under varying wind conditions. For example, [25] proposed an RL pitch controller to maintain nominal power using an adaptive dynamic programming algorithm, reducing the energy consumption of the pitch actuator. [26] developed a data-driven model to perform a torque-pitch controller, modeling the dynamics and using RL to control the wind turbine. [27] discussed different reward definitions for wind turbine controllers, while [28] and [29] developed RL methods for yaw control avoiding control parameter tuning. [30] and [31] employed Q-learning RL methods for maximum power point tracking (MPPT) control of the generator speed. Furthermore, multiple RL strategies have been proposed in wind farm-level control, where multi-agent methods mitigate wake effects, [32]. In general, these studies demonstrate the adaptability of RL systems to realistic wind conditions, thus enhancing the overall energy generation and efficiency of wind turbines.

Our proposed RL control strategy introduces several key advancements over existing methods. In particular, it is the first to explicitly incorporate noise generation as an optimization objective, addressing a critical yet often overlooked aspect of wind turbine control. Unlike most of previous RL-based approaches, which rely on simplified analytical models of wind turbine dynamics, our method integrates a blade element momentum theory (BEMT) solver. This enhances the physical accuracy of the model and avoids dependency on specific analytical fits, making it applicable to a broader range of turbine designs and operating conditions. Another significant improvement of our research is to estimate a noise-constrained EPC, allowing for better generalization compared to existing studies that typically evaluate performance under a single wind condition.

3 | Methodology

In this section, we detail the methodology to integrate deep reinforcement learning (DRL) with the dynamic control of a wind turbine. We begin by describing the wind turbine model, focusing on how both the power output and the noise levels are computed, and validate the methodology using field measurements for a SWT2.3-93 wind turbine with a rated power of 2.3 MW. Subsequently, we detail the setup of the DRL algorithm, which is designed to maximize power generation within specified noise constraints, demonstrating the application of advanced ML techniques to real-world energy optimization challenges. The flowchart illustrated at Figure 1 outlines the process of training the RL agent for wind turbine control, from initial setup to performance evaluation. The process begins with configuring

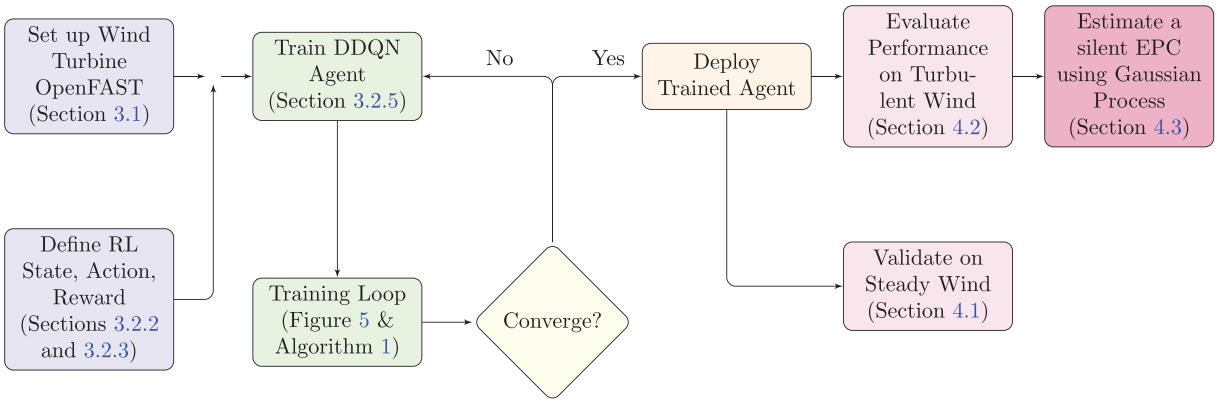


FIGURE 1 | Flowchart of the DDQN-based wind turbine control method, from training to evaluation of the RL agent.

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

the wind turbine in OpenFAST and defining the RL framework. The RL agent is then trained using a DDQN algorithm. Once training is complete, the agent is deployed and evaluated under various wind conditions to assess its performance.

3.1 | Wind Turbine Modeling Using OpenFAST

One critical requirement for incorporating a wind turbine solver into the DRL control framework is the ability to perform rapid evaluations, as the DRL training process requires a large number of simulations. To meet this need, we have chosen to employ an efficient BEMT solver. Specifically, we use OpenFAST [33], a well-known open-source software for simulating wind turbine dynamics and acoustic predictions.

BEMT is known for its efficiency and offers a simple yet accurate method for estimating the aerodynamic forces and energy generation of wind turbines. Its ability to perform rapid function evaluations is crucial for training and validating the agent within a reasonable time frame. In the realm of BEMT, the wind turbine blade is segmented into smaller sections along its span. The aerodynamic forces exerted on each section are computed based on the local wind conditions and the airfoil's geometry. These local flow conditions, defined for every section and time step, encompass the wind's speed and direction, along with the turbulence intensity. Polar curves for each airfoil section are used to compute the aerodynamic forces (lift, drag, and moment coefficients). By integrating the forces along the span of the blades, we can derive the overall power and thrust generated by the wind turbine.

In addition, OpenFAST includes an aeroacoustic module that enables the computation of the noise levels generated by the wind turbine at a specific location for the observers. To determine the aerodynamic noise sources from wind turbine blades, various semi-empirical noise models are included [34] and we select Brooks Pope and Marcolini model. The sound pressure level (SPL) for each blade segment is calculated based on individual noise mechanisms. The cumulative effect of these mechanisms yields the noise source for each airfoil. Finally, the noise sources from all blade segments are combined as noncorrelated noise sources, which contributes to the overall computation of the wind turbine sound power level. The essential aspect of this process is the precise identification and modeling of the various noise mechanisms associated with each blade section. These mechanisms can be categorized into two groups: turbulent inflow noise and airfoil self-noise. OpenFAST implements the turbulent inflow model presented by [35], and among the airfoil self-noise models described by [36], we have specifically selected the following: turbulent boundary layer trailing edge noise and tip vortex formation noise.

3.1.1 | Validation of OpenFAST With a SWT2.3-93 Wind Turbine

The onshore wind turbine selected for the study is based on the SWT2.3-93, which has a rated power of 2.3 MW. This turbine has undergone extensive experimental field testing, and complete details of its geometry and benchmark data are available in the open access repository Zenodo (through the work of the European project [37]). Table 1 shows general geometric and operational specifications of the SWT2.3-93 wind turbine. Figure 2 illustrates the chord and twist distributions across the blade. Polar airfoil curves are available in the airfoil catalog compiled by [38]. More details can be found in [39].

All open source information enables us to create an OpenFAST SWT2.3-93 model. Furthermore, the benchmark results from the Zenodo data set can be used to validate the model. Figure 3 presents the validation of both the power output and the SPL of the wind turbine. Figure 3a compares the experimental power curve (from the Zenodo database) with that generated by the OpenFAST solver, showing good agreement. Meanwhile, Figure 3b shows the one-third octave SPL for frequencies ranging from 10 Hz to 10 kHz, comparing the Zenodo dataset with the OpenFAST results. These comparisons cover three different operational conditions, as detailed in Table 2. The Zenodo acoustic results are computed for an observer positioned 100 meters downstream of the wind turbine, at ground level. We observe a good agreement with the experimental data for the three operating conditions. We conclude that OpenFAST, with the acoustic model of Brooks Pope and Marcolini, provides accurate predictions of power

TABLE 1 | SWT2.3-93 geometric and operational specifications.

Rated power	2.3 MW
Rotor diameter	93.0 m
Tower height	80.0 m
Rotor speed range	6–18 rpm
Cut-in wind speed	4.0 m/s
Rated wind speed	12.5 m/s
Cut-out wind speed	25.0 m/s
Survival wind speed	55.0 m/s
Tip-speed	78.0 m/s

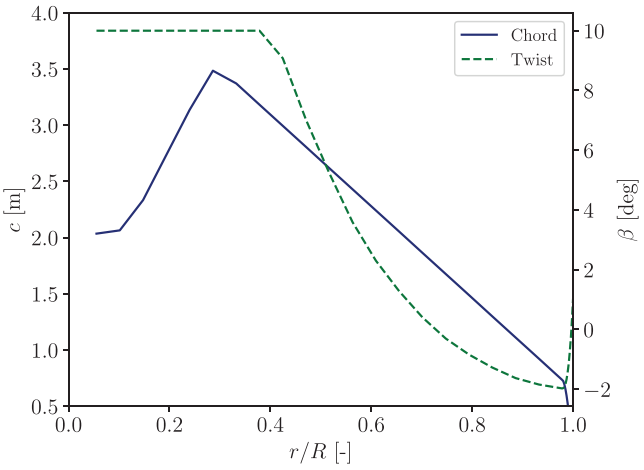
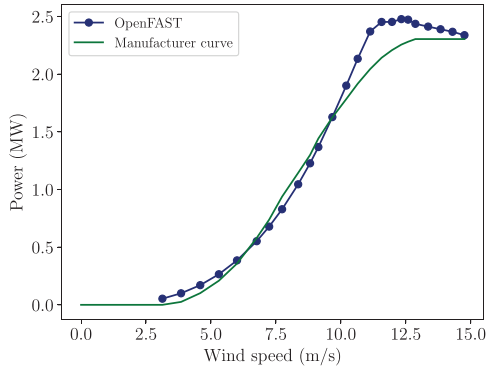
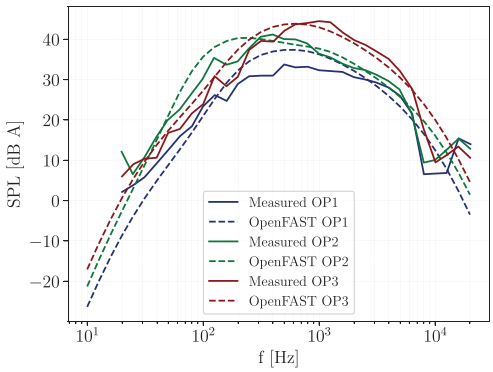


FIGURE 2 | Chord c and twist β span-wise distributions of the SWT2.3-93 wind turbine.

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a) Power curve for the SWT2.3-93 wind turbine. The manufacturer curve is obtain from the Zenodo open access repository.



(b) One-third octave SPL diagram for three different operational points. The measurements of the SPL spectrum are obtained from the Zenodo open access repository.

FIGURE 3 | Comparison of Zenodo dataset benchmarks and OpenFAST modeling of the SWT2.3-93 wind turbine.

TABLE 2 | Operational conditions studied on the Zenodo aeroacoustic dataset.

Operation point	U_∞ (m/s)	Ω (rpm)	θ (°)
OP1	6.0	13	3
OP2	8.0	14	−2
OP3	9.5	17	5

Note: The operation point is defined by the wind speed U_∞, the rotational speed Ω and the blade pitch angle θ.

generation and acoustics and is therefore a valid tool to perform multi-objective optimization.

3.1.2 | Sensitivity to Control Parameters

The parameters selected to control the wind turbine power and noise include the rotation speed Ω and the angle of rotation of the blade θ. Before discussing the RL setup, it is crucial to illustrate the sensitivity of these parameters for the two performance metrics: the power coefficient and the overall SPL.

In Figure 4, the sensitivity analyses for both the rotational speed and the blade pitch angle are displayed for a single incoming wind speed (U_∞). The values for one-third octave SPL, power coefficient, and overall SPL are shown for different operational conditions. Figure 4a depicts the influence of Ω; increasing the rotational speeds increases both the SPL and the power coefficient, highlighting the trade-off between maximizing power and minimizing noise. The entire SPL spectrum increases uniformly with increasing Ω, due to a higher relative velocity in the blades and leading to an increase in SPL regardless of the noise mechanism. A similar analysis is presented in Figure 4b for the blade pitch angle. Although the general conclusion about the opposition between power and noise remains valid, the SPL spectrum behaves differently across frequencies. The pitch angle mainly affects trailing edge noise, leading to changes at relatively low frequencies ranging from 10 Hz to 1 kHz.

3.2 | Design of a RL Control

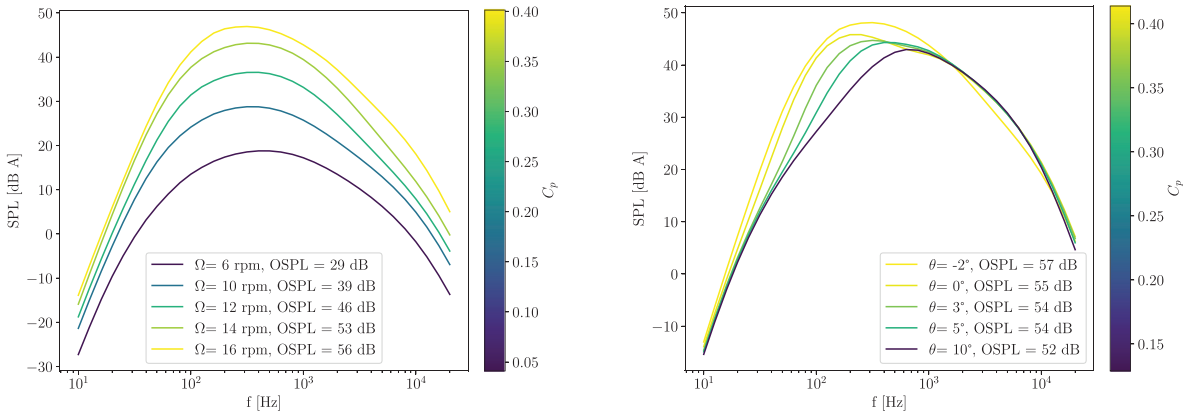
RL is a branch of ML that focuses on how agents should take actions in an environment to maximize the cumulative reward. Unlike supervised learning, where the model learns from a labeled dataset, RL is driven by agent-environment interactions. The agent takes actions based on the current state of the environment and receives feedback in the form of rewards. The state represents the situation of the environment at a given time, while the actions are the possible moves the agent can make. The reward is the feedback indicating the immediate benefit or cost of an action, guiding the agent toward better actions over time. In particular, in this work, we use Q-learning RL, which is detailed in the next section.

3.2.1 | RL for Multi Objective Control

Q-learning is a widely recognized RL algorithm [40]. It is categorized as a model-free RL algorithm, implying that it operates without the need for prior knowledge or explicit models that represent the dynamics of the system. The fundamental component of Q-learning is the Q-value, which quantifies the anticipated cumulative reward for executing a specific action in a given state. The Q-value is updated iteratively via the Bellman equation, which formulates the optimal action-value function in terms of the maximum expected future reward. During the learning process, the wind turbine interacts with the environment, transitions between states, and takes actions according to its current policy. The Q-learning algorithm employs an ϵ-greedy exploration-exploitation trade-off to strike a balance between exploring new actions (ϵ times) and exploiting current knowledge ($1-\epsilon$ times) to maximize cumulative rewards. In RL, the cumulative reward is calculated taking into account that a reward received immediately is worth more than a reward received in the future, specifically, each time step the reward is discounted by γ, the discount rate.

Initially, the Q-values are arbitrarily initialized. As the wind turbine explores the environment and receives feedback in the form of rewards, the Q-values are updated using the temporal

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a) Sensitivity analysis for rotational speed. Operational point: $U_\infty = 12$ m/s and $\theta = -1°$.

(b) Sensitivity analysis for pitch. Operational point: $U_\infty = 12$ m/s and $\Omega = 16.5$ rpm.

FIGURE 4 | Sensitivity analyses for control parameters in relation to OSPL, power coefficient (C_p), and one-third octave SPL (dBA) spectra.

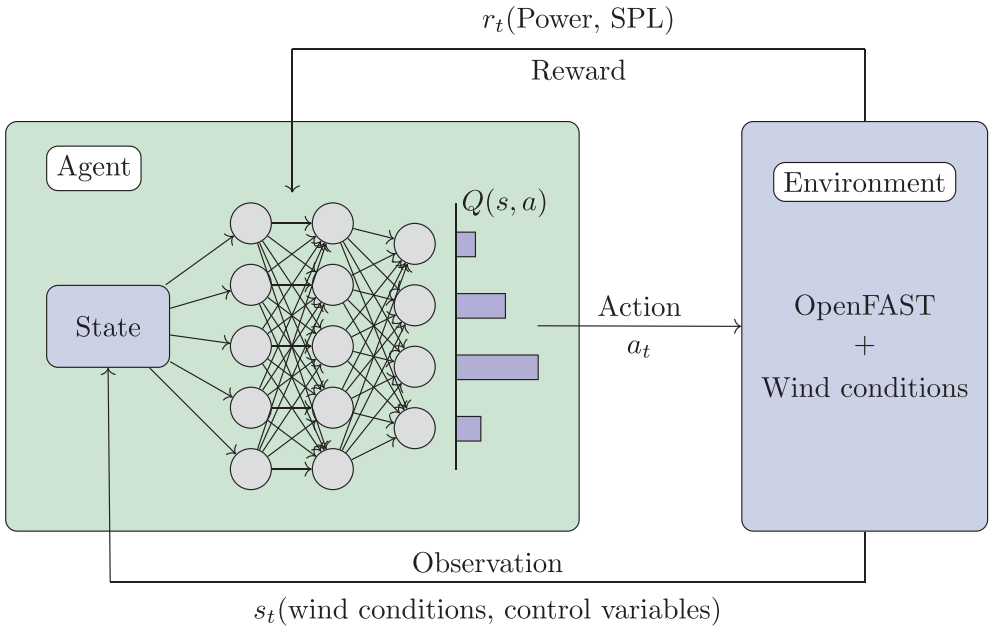


FIGURE 5 | Flow diagram of the reinforcement learning control methodology.

difference error. The temporal difference error represents the discrepancy between the observed reward and the predicted reward based on the Q-values. Through repeated iterations, the Q-learning algorithm gradually converges to an optimal policy. In this state, the wind turbine learns the best actions to take in different states, thus maximizing power generation while minimizing noise. In our case, these agent-environment interactions for wind turbine control are illustrated in Figure 5.

Deep Q-learning (DQN) is a variant of Q-learning that employs a deep NN to estimate Q-values [41]. DQN replaces the traditional lookup table with a NN, enabling generalizations across states to handle large state spaces efficiently. In this study, a double deep Q-learning (DDQN) is employed. DDQN is an extension of DQN that uses two NNs: the primary network and the target network. The primary network selects the action, and the target network evaluates its Q-value. This way of decoupling the selection and evaluation of actions addresses the overestimation of Q values, often observed in DQN algorithms due to the maximization bias [42]. The weights of the primary network are obtained by minimizing the following loss function:

$$\mathcal{L}(\phi) = \mathbb{E}_{(s,a,r,s')}\left[\left(r + \gamma Q_{\phi'}\left(s', \arg\max_a Q_\phi(s',a)\right) - Q_\phi(s,a)\right)^2\right], \tag{1}$$

where r is the reward, s is the state of the environment, a denotes a possible action that the agent can take, $Q(s,a)$ is the Q-function, and ϕ and ϕ' are the set of weights of the primary and target network, respectively. The loss function $\mathcal{L}(\phi)$ quantifies the residual of the Bellman equation, which formally defines the

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

optimal values of the Q-values [16]. The set of weights from the target network, ϕ', is updated using a soft update rule to enhance the stability of the learning process [43].

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi'. \tag{2}$$

To train the DDQN, an experience replay buffer is utilized. During the training phase, the agent interacts with the environment and stores the experiences (state, action, reward, next state) in the replay buffer. Subsequently, random batches of experiences are sampled from the replay buffer to train the network and update its weights. This process helps to break the correlation between consecutive samples and improves stability during the learning process.

An additional consideration in solving this RL problem is the need to balance maximizing power output with minimizing noise impact. These objectives are inherently conflicting, placing this problem within the multi-objective reinforcement learning (MORL) framework. MORL extends traditional reinforcement learning to handle problems that involve multiple, often conflicting, objectives.

Various strategies exist to address MORL problems. One of the simplest methods is to define the reward using a scalarized function that combines the rewards for each objective into a single global reward, thus transforming the problem into a single-objective RL task [44]. Another approach involves Pareto optimization, which aims to find a set of optimal policies that lie on the Pareto front, where no other policy is superior in all objectives [45]. There are already methods that apply these MORL approaches using deep learning implementations [46]. In this work, a scalarized method is adopted to define a reward that balances the two objectives of maximizing power and minimizing noise.

3.2.2 | State-Action Structure

The agent state must include all the necessary information about the environment to allow the agent to take the best possible action. If the state lacks relevant information, the agent may not be able to achieve optimal performance. The agent state is defined by the incoming wind conditions, specifically the wind speed U_∞, along with the control variables of the wind turbine, which are the rotational speed Ω, and the pitch angle of the blade θ. To fit within the DDQN framework, the state space $\mathcal{S}$ must be bounded. Some variables (rotational speed and pitch) are bounded by mechanical/structural limitations, whereas the wind speed is bounded by physical range. Note that these can be tuned for specific wind turbines and geographic sites. We include an additional constraint on the tip speed ratio $\lambda = \frac{\Omega R}{U_\infty}$, with the blade radius $R = 46.5$ m, to ensure the correct behavior of the BEMT solver. The specific values of all the constraints are outlined below:

- $U_\infty \in [4,16]$ m/s,
- $\Omega \in [6,18]$ rpm,
- $\theta \in [-5,10]^\circ$,
- $\lambda \in [3,12]$.

Notice that we could include the power and noise generated by the wind turbine in the agent state definition. However, we consider that those magnitudes are not directly observable by the agent and are only included via the reward function.

The actions available to the agent involve either increasing or decreasing the control variables. Since Q-learning is defined for a discrete action space $\mathcal{A}$, the control variables can only be adjusted by fixed increments. Five distinct actions are defined: two for each control variable (one for increasing and one for decreasing) and one for maintaining the current state (doing nothing). The specific fixed increment for each possible action is determined based on the sensitivity analysis detailed in Section 3.1.2. Since the rotational speed is more sensitive to both power generation and SPL compared to the pitch angle, incremental adjustments for each variable have been designed so that their corresponding actions have effects of the same magnitude. The actions that the agent can take are specified as follows:

- a_1: increase Ω by 0.5 rpm,
- a_2: decrease Ω by 0.5 rpm,
- a_3: increase θ by 1°,
- a_4: decrease θ by 1°,
- a_5: do nothing.

It is important to note that the transition between states is not deterministic a priori. Although we can freely adjust the control variables, the wind conditions depend on the environment and are beyond our control. This motivates the use of a model-free RL method, as model-based approaches only guarantee convergence if the transition function between states is known.

Regarding the state-action transition and its integration with the wind turbine solver, OpenFAST is used to compute power and SPL values for each encountered state, without accounting for the torque and pitch dynamics during state transitions. This assumption is reasonable as long as the time between actions is much longer than the transient characteristic time between states. Since the RL agent transitions between steady-state solutions, the action interval does not influence the training process. Therefore, the control frequency can be selected after training based on operational requirements and desired control update intervals.

3.2.3 | Reward Definition

The reward function is a key to defining the RL algorithm, as it is the only feedback that quantify the success of the actions taken by the agent. Therefore, the reward must be carefully crafted for each specific problem to learn an appropriate policy. As mentioned in Section 3.2.1, this is a multi-objective optimization problem (or MORL), requiring a specific strategy to address the two conflicting objectives: maximizing power extraction while minimizing noise generation. In this work, we choose to blend the two objectives through a linear function to define the overall reward. The reward can be expressed as follows:

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

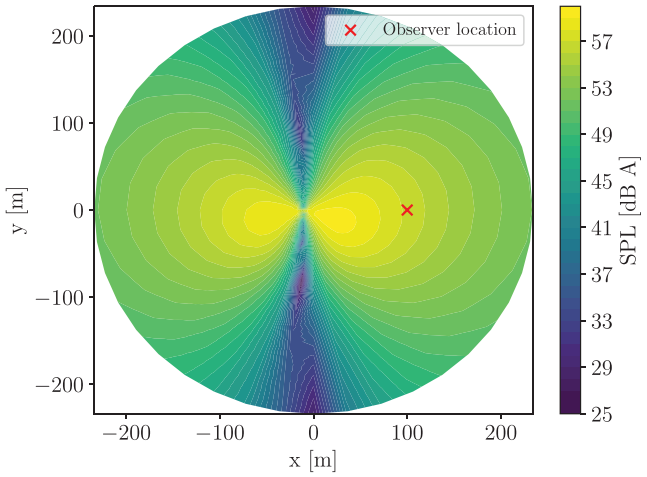


FIGURE 6 | Directivity map of sound pressure level (SPL) generated by OpenFAST. The wind direction is oriented along the positive x-axis (left to right), perpendicular to the wind turbine rotor. The operational conditions are $U_\infty = 12$ m/s, $\Omega = 16.5$ rpm and $\theta = -1°$. The observer location "×" is situated 100-m downwind.

$$r = r_{\text{PW}} + r_{\text{SPL}}, \tag{3}$$

where r_{PW} denotes the reward associated to the power objective and r_{SPL} the one related to the SPL one. The reward function is structured such that r_{PW} incentivizes power generation, making it a positive component, while r_{SPL} penalizes excessive noise, resulting in a negative contribution. This positive–negative reward structure is known to improve the convergence of learning and the performance of the wind turbine controller [47].

As we discussed previously in our previous work [48], the reward power component should encourage the agent to obtain the highest energy generation possible, regardless of the wind conditions. To achieve this, we use the power coefficient C_p of the wind turbine. We set this reward to increase linearly from 0 to 1, with 1 corresponding to the maximum possible value of the power coefficient within the state space, $C_{p,\text{nom}}$. Therefore, the reward power component reads as follows:

$$r_{\text{PW}} = \frac{C_p}{C_{p,\text{nom}}}. \tag{4}$$

The reward term related to sound generation, r_{SPL}, is highly dependent on the specific problem being modeled. First, we need to select the observer locations where the SPL is computed, typically in critical areas where noise mitigation is a priority. In this work, we decide to set an observer 100-m downstream of the wind turbine; see Figure 6. Next, we decide how to penalize sound generation (SPL) in the reward function. We opt to use a ReLU activation function that begins to penalize once the SPL exceeds a certain threshold SPL_{thr}. Below this threshold, the agent focuses solely on maximizing power. Additionally, we define a Δ dB value that specifies how much the SPL threshold can be exceeded before the reward becomes −1. Beyond this point, no matter how much power the agent generates, the total reward will be negative. Therefore, $\text{SPL}_{\text{thr}} + \Delta$ dB serves as an effective noise limit. For this specific

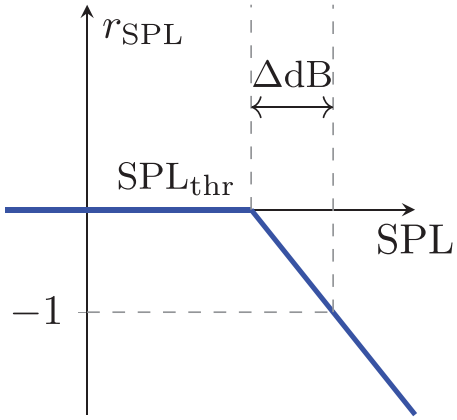


FIGURE 7 | Reward noise component.

application, we set $\text{SPL}_{\text{thr}} = 45$ dBA and Δ dB $= 5$ dBA, but note that these values can be adapted to specific sites or regulations. The reward noise component can be seen in Figure 7 and reads as follows:

$$r_{\text{SPL}} = -\text{ReLU}\left(\frac{\text{SPL} - \text{SPL}_{\text{thr}}}{\Delta\,\text{dB}}\right), \tag{5}$$

where ReLU states for a Rectified Linear Unit activation function.

In addition, we need to include the bounds of $\mathcal{S}$ in the reward. To make the agent learn the limits, it receives punishments whenever it performs a forbidden action, that is, an action that leads to a state $s_{t+1} \notin \mathcal{S}$. In such cases, the agent receives a negative reward with a value of $r = -3$, and the action is revoked so that the control variables remain the same. The punishment is set to −3 to differentiate it from the possible negative reward of r_{SPL}. This distinction is made because exceeding the limits of $\mathcal{S}$ is considered worse than generating noise above the threshold. Finally, the reward function for the agent is the following:

$$r(s_t, a_t, s_{t+1}) = \begin{cases} \dfrac{C_p(s_{t+1})}{C_{p,nom}} - \text{ReLU}\left(\dfrac{\text{SPL}(s_{t+1}) - \text{SPL}_{\text{thr}}}{\Delta\,\text{dB}}\right), \\ -3 \qquad \text{if} s_{t+1} \notin \mathcal{S}. \end{cases} \tag{6}$$

3.2.4 | NN Architecture

When using DQN, NNs are used to approximate the Q-function. Typically, the NN is designed to approximate the Q-vectors, $\boldsymbol{q}(s)$, which represent the Q-values in the state s for all possible actions. That is, $\boldsymbol{q}(s)_i = Q(s, a_i)$. This approach is used because $\mathcal{A}$ is a discrete space, and encoding these discrete actions as an input can be problematic; it is more convenient to create a mapping between real subspaces. The NN map is defined as $\boldsymbol{q}_\phi(\boldsymbol{s})$: $\mathcal{S} \subset \mathbb{R}^3 \to \mathbb{R}^5$, where ϕ denotes all NN weights, the output space dimension is $|\mathcal{A}| = 5$ and $\boldsymbol{s}$ denotes the state vector, which is $\boldsymbol{s} = [U_\infty, \Omega, \theta]^T$.

The NN architecture follows a multilayer perceptron structure with two dense hidden layers, each using ReLU activation functions. The final layer employs a linear activation function

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

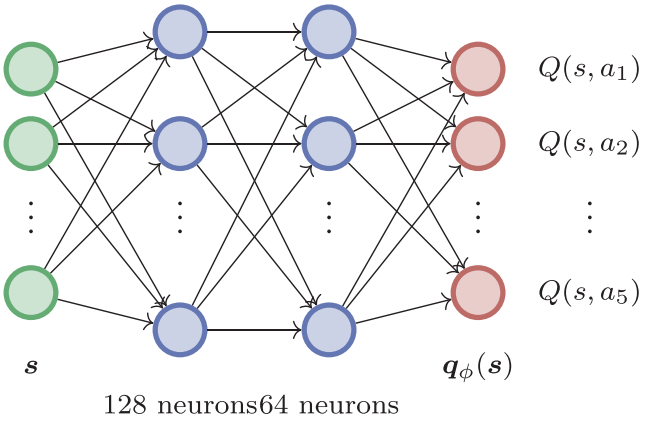


FIGURE 8 | Q-network architecture.

instead of ReLU, allowing the Q-values to take any real value rather than being restricted to positive numbers.

In RL, NNs are typically smaller compared to those used in supervised classification tasks. Most studies employ architectures with two or three relatively small hidden layers [41, 49, 50]. Following this common practice, we found that a network with two hidden layers of 128 and 64 neurons was sufficient to accurately approximate the Q-function. The final architecture of the Q-network used to train the DDQN agent is illustrated in Figure 8.

3.2.5 | Training of the Wind Turbine DDQN Agent

The main ideas of Q-learning have already been explained in Section 3.2.1. However, here the specific details of the DDQN training to design the wind turbine controller are included. During the training phase, the agent faces random steady wind conditions during short episodes of 20 time steps. This allows the agent to adapt to virtually any wind, even if the wind speed changes faster than 20 time steps. This adaptability is achieved because experiences are stored in the replay buffer and batches are selected randomly. Consequently, the specific temporal evolution of states during the agent's experience is not critical, provided that the stored transitions comprehensively represent all actual transitions in the system.

TABLE 3 | Training parameters used for the DDQN agent.

Parameter	Value
Environment interactions	200 k
Steps of the environment per training iteration	5
Maximum capacity of the replay buffer	50 k
Batch size	64
Learning rate	5e-4
Discount factor	0.95
Epsilon value for the epsilon greedy policy	0.50
Tau soft update parameter	0.1
Period of update of the target network	20

The double deep Q-learning (DDQN) agent is trained using the hyperparameters listed in Table 3. To ensure sufficient exploration of the state-action space within a reasonable time,

Algorithm 1 Double Deep Q-Learning Algorithm

Initialize primary network Q_ϕ with random weights ϕ
Initialize target network $Q_{\phi'}$ with weights $\phi' = \phi$
Initialize replay buffer $\mathcal{R}$
Set hyperparameters: α (learning rate), γ (discount factor), ϵ (exploration probability), τ (soft update parameter), N (number of training iterations), n (number of steps taken every training step, m (period of update of the target network).
Initialize random state s
for iter=1 **to** N **do**
  for n steps **do**
    Select action a using ϵ-greedy policy from Q_ϕ
    Take action a, compute reward r and new state s' using the wind turbine solver
    Store transition (s, a, r, s') in replay buffer $\mathcal{R}$
    if last step of the episode **then**
      Initialize random state s
    else
      Update current state: $s \leftarrow s'$
    end if
  end for
  Sample random mini-batch of transitions (s_j, a_j, r_j, s'_j) from $\mathcal{R}$
  Estimate loss function from eq. (1) with the mini-batch transitions
  Update the primary set of weights ϕ using a gradient optimizer.
  $\phi \leftarrow \phi - \alpha \nabla_\phi \mathcal{L}$
  each m iterations **do** Soft update of the target network weights ϕ' using eq. (2) rule.
end for

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

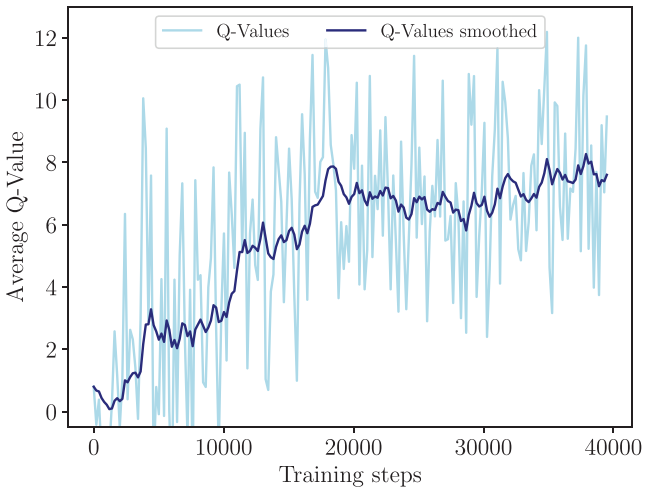


FIGURE 9 | Average Q-values taken during the training process of the DDQN agent.

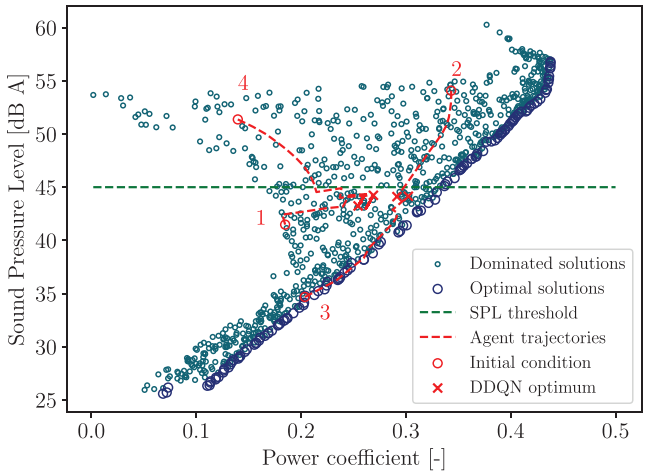


FIGURE 10 | Pareto diagram at a wind speed of 10 m/s, showing the power coefficient C_p and sound pressure level SPL (dBA) computed for 1000 random states within the state space. Different agent trajectories are displayed on the Pareto diagram. The numbering of the initial conditions correspond to Table 4.

the epsilon value in the epsilon-greedy policy is set to 0.5. The training process is outlined in the pseudocode presented in Algorithm 1. Learning progress is assessed by monitoring the evolution of the Q-values for the state action pairs encountered, as shown in Figure 9. As expected, the Q-values increase as the agent learns optimal actions. Notably, these Q-values are obtained from short evaluation phases conducted during training using a pure greedy policy. By the end of training, the smoothed Q-values stabilize around 8, indicating convergence of the NN optimization.

The computational cost of training the agent is primarily dominated by the wind turbine solver, OpenFAST. This is a typical situation in flow control problems where the environment is the most computationally expensive component. In this case, the BEMT solver is sufficiently efficient, allowing training to be completed in 60 computational hours with the hyperparameters outlined in Table 3. However, for more complex problems with higher computational demands, techniques such as parallel environment algorithms can be employed to accelerate training [49, 50].

For implementation, we used OpenAI Gym [51] to create the environment, serving as a bridge between the RL formulation and the OpenFAST wind turbine solver. TF-Agent [52] was employed to develop the DDQN agent and manage the entire training process. All NNs were constructed using Keras API [53], and Adam optimizer was used for training [54].

4 | Results

The agent's performance is assessed under various wind conditions. First, we validate the operational point that the agent reaches under constant wind conditions, assessing its optimality through a Pareto front. Next, we evaluate the agent's ability to adapt to turbulent wind conditions by comparing its control strategy against a classic controller. Finally, we estimate the agent's annual energy production and compare it against a classic control strategy designed to maximize energy extraction.

TABLE 4 | Four different cases evaluated under steady wind speed of 10 m/s.

Case	1	2	3	4
Ω_0 (rpm)	9.61	17.98	8.43	16.54
Ω_f **(rpm)**	11.11	10.98	10.93	11.04
θ_0 (°)	9.94	3.87	−1.25	8.37
θ_f (°)	5.93	3.87	2.75	5.37
$\mathbf{r(s_0)}$	0.41	−1.05	0.45	−0.56
$\mathbf{r(s_f)}$	0.56	0.65	0.67	0.60

Note: The initial condition is defined by the initial control variables Ω_0 and θ_0 and the agent get to the optimum defined by the control variables Ω_f and θ_f. The initial and final rewards of each trajectory, $r(s_0)$ and $r(s_f)$, are also presented.

4.1 | Steady Wind Validation

The simplest test for evaluating the agent is to assess its performance under steady wind conditions. In this scenario, with unchanging wind conditions, the agent must identify, reach, and maintain the state that maximizes the cumulative reward. This optimal state does not depend on the initial conditions of pitch and rotor speed, as the wind conditions are steady. However, it is important to note that the agent's actions are discrete, limiting its ability to reach every possible state.

To validate the performance and robustness of the agent, a Pareto diagram is used. The agent is tested for different initial conditions (with the same steady wind speed) to determine whether it can consistently reach optimal states (at the pareto front), regardless of the initial state. To illustrate the agent's trajectory (sequence of state-action-reward) for each initial condition, these trajectories are displayed on a power coefficient—SPL diagram, along with values for 1000 random states from $\mathcal{S}$. Figure 10 presents the Pareto diagram, illustrating the

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

agent's trajectory from four different initial conditions. The Pareto front was obtained following a Monte Carlo approach, where 1000 random states were sampled within the state space. Each state was evaluated in terms of the power coefficient C_p and the SPL. The nondominated solutions, those for which no other sampled state simultaneously improves both objectives, were selected to form the Pareto front. It is clear that, regardless of the initial condition, the agent successfully achieves high power outputs up to the maximum permissible decibel level. Furthermore, the agent demonstrates robust performance by consistently avoiding the maximum limit of SPL (dBA) while remaining close to the limit to maximize power. It can be seen that the RL does not always reach the same final state, but that the optima are relatively close to each other. This suggests the existence of a local optimum. In addition, the discrete nature of the Q-learning actions may not allow the agent to reach certain optima, since not all states are reachable from an initial state. Despite these issues, the agent consistently avoids acoustic penalties and achieves high power outputs, with power coefficients ranging from 0.26 to 0.30.

For completeness, Table 4 shows the initial conditions of the control variables for the trajectories displayed on Figure 10, as well as the final-state control variables.

Overall, the agent robustness has been tested in a simple steady wind scenario. The agent is able to reduce wind turbine noise to admissible levels while maximizing power. Furthermore, the agent finds optimum operational conditions regardless of the initial condition, which shows the robustness of the algorithm. In other words, the NN that approximates $Q(s, a)$ has covered all his input space $\mathcal{S} \times \mathcal{A}$.

4.2 | Control Strategy for Experimental Winds

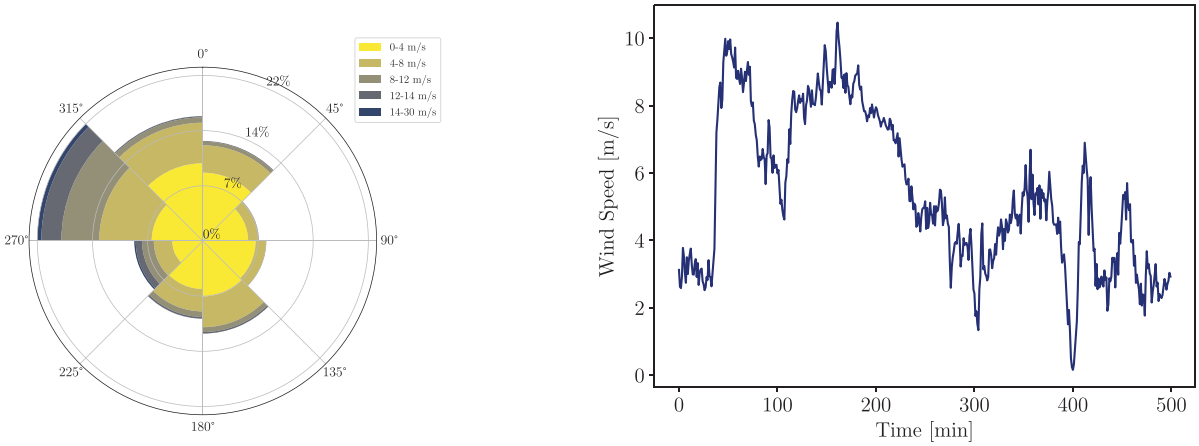
The agent capabilities are now tested in experimental wind conditions. We compare the energy extraction between our agent and two controllers that are designed solely to maximize power. In doing so, we can demonstrate how much power we need to sacrifice to keep the wind turbine at acceptable decibel levels.

The performance of the three controllers is compared. These controllers include:

- **Classic wind turbine controller**: Standard wind turbine controller designed to reach the power curve of the wind turbine, using torque or pitch control depending on whether the wind speed is above or below the rated wind speed. Details can be seen in Appendix C.
- ***Power*** *DDQN*: Agent designed solely to maximize power. It is trained without noise penalization; that is, only the power reward is included; see equation (4).
- ***Quiet*** *DDQN*: Agent designed to maximize power without producing more than 45 dBA decibel levels at 100 m downwind of the rotor. It is trained with the complete reward definitions including power and noise; see equation (6).

Wind data used to validate control performance under real wind conditions were sourced from the Flatirons M2 site, part of the Measurement and Instrumentation Data Center (MIDC) operated by the National Renewable Energy Laboratory (NREL); see [55]. These daily wind measurements are openly accessible as open source data. For this study, wind speed and wind direction measurements were selected at an 80-m height, spanning from June 1, 2023 to June 1, 2024. The wind speed data consist of mean velocities sampled at a 1-min frequency. Figure 11a displays a wind rose that illustrates the wind speed and wind direction of this data set. Since this study focuses on torque-pitch control, it is assumed that the incoming wind is consistently aligned with the wind turbine, a condition typically managed by yaw control. Therefore, we assume perfect alignment, and only the wind speed distribution is used in the subsequent results.

Note that the *Power* DDQN agent considers power optimization uniquely. Therefore, it is only applicable to the below rated wind speed region defined in Appendix C. When the wind speed exceeds the rated value, the control strategy maintains the nominal power rather than maximizing it. To achieve this behavior with a RL agent, the reward function would need to be modified. Consequently, the wind speed distribution used to validate the agent under turbulent wind conditions is restricted to

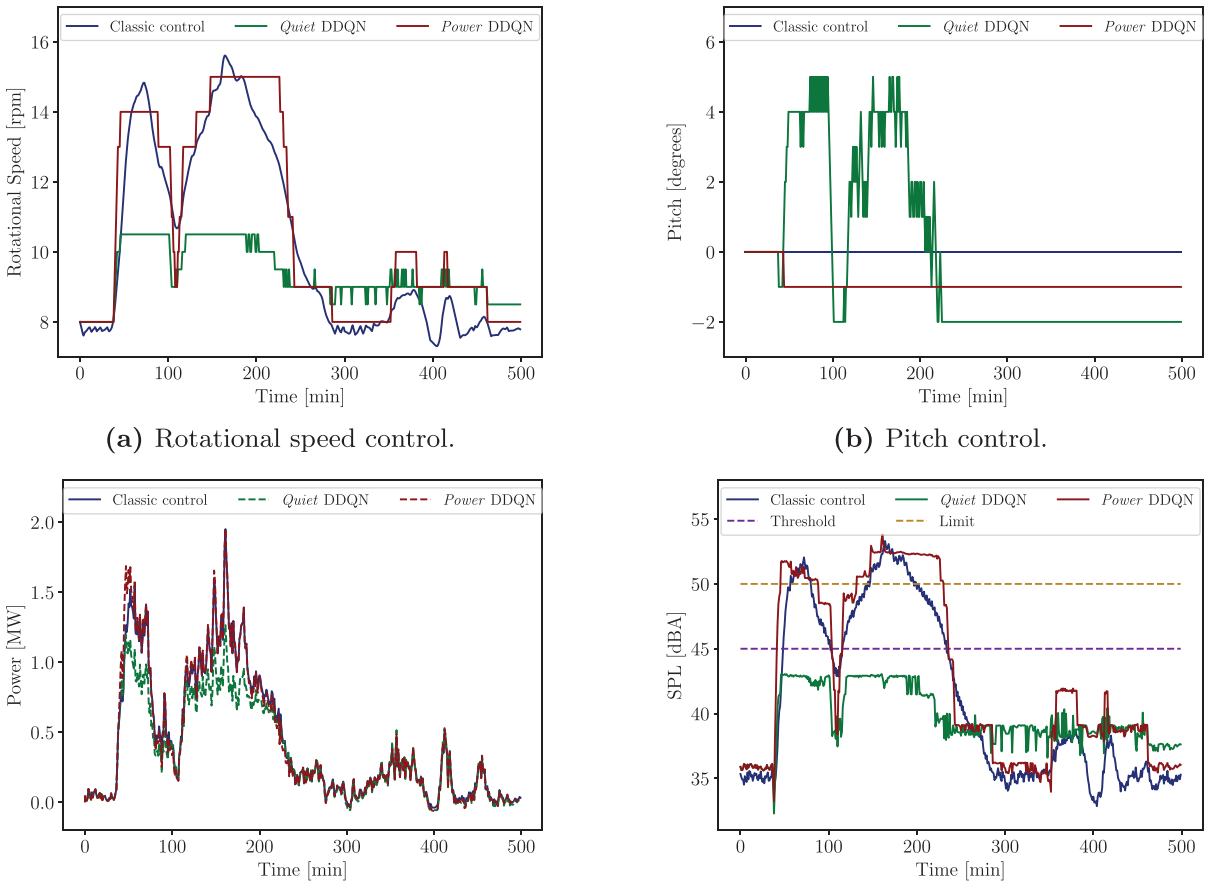


(a) Wind rose for one year of data.

(b) Wind speed distribution for 8 hours.

FIGURE 11 | Wind conditions on the *experimental wind environment*.

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a) Rotational speed control.

(b) Pitch control.

(c) Power generation.

(d) Sound Pressure Level of the wind turbine.

FIGURE 12 | Control results for the three different agents on the 8-h *experimental wind environment*.

below-rated wind speeds, enabling a meaningful comparison between the three controllers.

The control performance of the three agents is analyzed in detail over an 8-h time span, using the wind speed distribution shown in Figure 11b. RL controllers are allowed to control each minute which is the experimental data frequency sample. In the next section, we estimate the annual energy production for all controllers.

Figure 12 shows the results of the different controllers during the first 8 h of the yearly data set. Figure 12a,b displays the control parameters evolution, while Figure 12c,d illustrates the power and the noise generated 100-m downwind. It is noted that the *Power* agent matches the power extraction achieved by the classic control strategy, essentially implementing the same control approach but with the discrete actions defined for the RL agent. Since neither of these controllers is designed to consider the acoustics of the wind turbine, both generate high levels of noise when the wind speed is sufficiently high. In contrast, the *Quiet* agent can match the power generation of the power-oriented controllers when the wind speed is moderate. When wind speeds are higher, it extracts as much power as possible while keeping noise levels below the threshold value. In addition, all three controllers maintain a constant pitch value to maximize power extraction. However, the *Quiet* agent adjusts the pitch angle to reduce the noise levels when the wind speeds increase.

This test shows the flexibility of the RL strategy for control and highlights the possibility of including multiple objectives. In addition, we see that there is no need to have an a priori knowledge of the turbine performance (e.g., the power curve or rated maximum power) since the RL will learn these characteristics when trained.

4.3 | Annual Wind Energy Estimation

There are different methodologies for obtaining an estimate of the annual generation of wind energy [56]. The standard procedure is based on decoupling the wind turbine from the wind distribution of the particular site. Consider the observed wind speed frequency histogram to fit a theoretical probability density function (PDF) for the wind speed. It also requires a transfer function that models the relation between power output and wind speed. Typically, the Weibull distribution is used to fit the wind speed frequency histogram. [57] showed that although the Weibull distribution may not be substantiated for most sites, it does not include important errors in the energy estimations. The probability density function of the Weibull distribution is given by the following:

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$f_U(u) = \left(\frac{k}{c}\right) \cdot \left(\frac{u}{c}\right)^{k-1} \exp\left(-\left(\frac{u}{c}\right)^{k}\right), \tag{7}$$

where U denotes the random variable that models the wind speed. The fit of the shape and scale parameters k and c is established from the mean and variance of the wind speed, $\mu_U = \mathbb{E}[U]$ and $\sigma_U^2 = \mathbb{V}[U]$. The specific relations can be seen in [58] work and are the following:

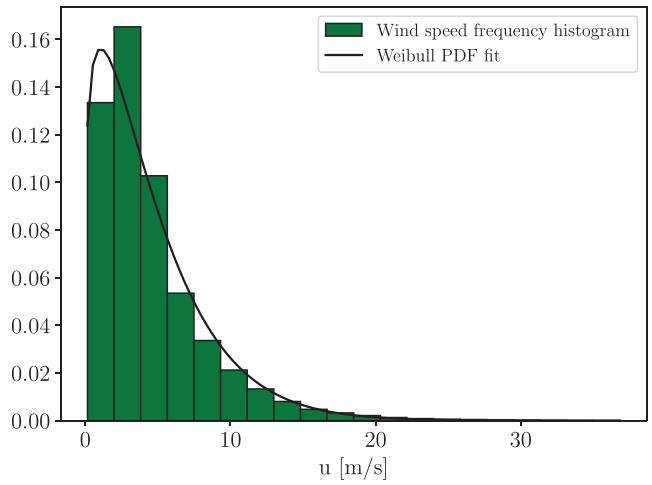


FIGURE 13 | Wind speed frequency histograms of one year experimental data and PDF of the adjusted Weibull distribution. The shape and scale parameters are $k = 1.195$ and $c = 4.837$, respectively.

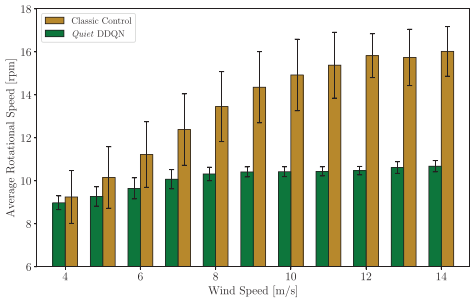
$$k = \left(\frac{\sigma_U}{\mu_U}\right)^{-1.086} \tag{8}$$

and

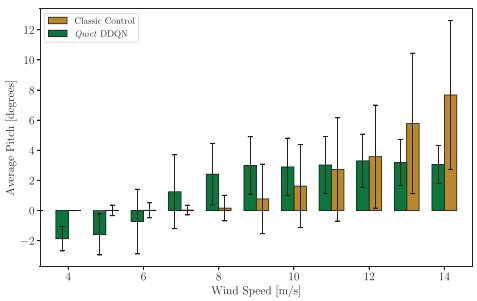
$$c = \frac{\mu_U}{\Gamma\left(1 + \frac{1}{k}\right)}, \tag{9}$$

where Γ denotes the special gamma function. This formulation can be used to obtain the Weibull probability density function that represents the 1-year experimental data reported by [55]. This is illustrated in Figure 13.

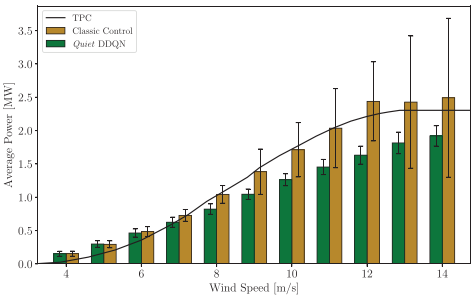
Regarding the transfer function between power and wind speed, there are various strategies [59]. The theoretical power curve (TPC) does not account for control mechanisms. Furthermore, since our wind turbine control strategy considers acoustic generation, the wind turbine will exhibit a significantly different EPC. It is necessary to compute an EPC that accurately represents the transfer function between power and wind speed for our specific control scenario. The EPC can be computed using simulations of the wind turbine control. The agent is faced against a turbulent wind that covers all the range of interest of wind speed, mainly between cut-in and cut-off wind speeds. This turbulent wind must be representative of the turbulent nature of the wind that the wind turbine is going to face during operation. Once the simulation is completed, all pairs of data points (U_∞, C_p) can be used to obtain a transfer function for the power coefficient $C_p(U_\infty)$.



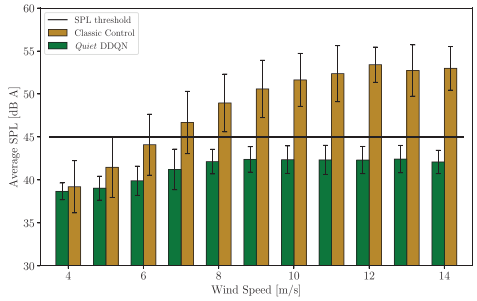
(a) Rotational speed empiric operational curve.



(b) Pitch empiric operational curve.



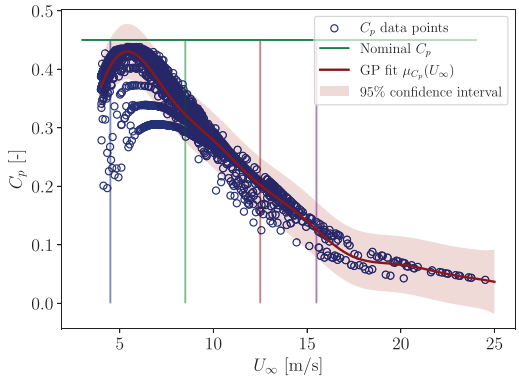
(c) Effective Power Curve (EPC) of both control strategies against the Theoretical Power Curve (TPC) from the manufacturer.



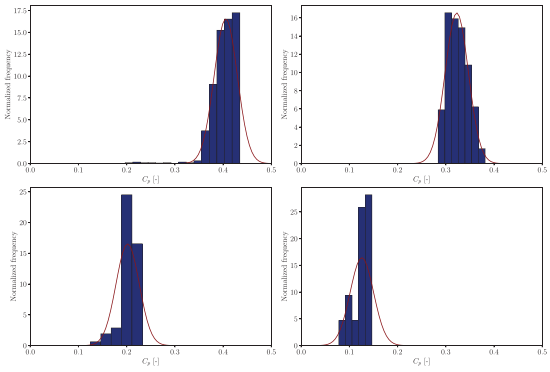
(d) Sound Pressure Level of the wind turbine observed at 100 m downwind for both control strategies against the SPL threshold.

FIGURE 14 | Average values of the control variables for wind speed bins, obtained from simulating two different control agents over a 100-h experimental wind speed period. The error bars denote the standard deviation.

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a) Power coefficient EPC. The data points represent all the observed (U_∞, C_p) pairs observed during the simulation performed to obtain the EPC. The Gaussian Process Regression fit, mean and confidence interval is included.



(b) Comparison of the C_p frequency histograms and the Gaussian probability distribution (—) obtained with the GPR for different wind speed values. The wind speed values for each histogram are specified in Figure 15a with vertical colored lines. The order is the following:
Top Left: , Top Right: , Bottom Left: , Bottom Right: .

FIGURE 15 | Power coefficient obtained from simulating the *Quiet* DDQN controller over a 100-h experimental wind speed period.

A subset of 100 h of MIDC experimental wind measurements [55] has been used to obtain the EPC of the SWT2.3-93 wind turbine using the classic control and the *Quiet* DDQN agent already introduced on Section 4.2, a detailed discussion on the influence of this subset in the EPC estimation is done in Appendix B. Figure 14 illustrates the results of these simulations, showing the operational laws of control for each agent on Figure 14a,b and the SPL and power associated on Figure 14c,d, respectively. For these figures, data from the 100-h simulation is aggregated and binned according to wind speed. Specifically, the width of the wind speed bin is 1 m/s. The average and standard deviation of the outputs within each bin were calculated. These are visualized as bars and error bars, respectively. This binning process enables the analysis of output variability and performance as a function of wind speed under real-world conditions. The behavior is as expected is the *Quiet*. The agent does not increase the rotational speed above 10.5 rpm to avoid surpassing the SPL threshold and uses the pitch to reduce noise if needed, which explains the high variance bars on the pitch (see Figure 14b) and the low ones on the rotational speed (see Figure 14a). Meanwhile, the classic control can increase the rotational speed freely, and the pitch is only used in above-rated wind speed scenarios; see C. The large standard deviations on the classic control pitch are due to the PID control, which dynamically adjusts to the turbulent wind. In Figure 14c, it is illustrated how the classic control resembles on average the TPC. However, it is not able to adjust perfectly to the turbulent wind, as shown by its high variance on the above-rated wind speed region. The *Quiet* agent achieves less power than the classic one but is able to maintain the SPL below the specified threshold of 45 dBA; see Figure 14d.

Traditional approaches use only the average value or polynomial fits of the historical/simulated data to construct the EPC. All of these methods do not capture the variance of the data in the model. To account for this variability on the EPC model, we introduce a statistical method. For simplicity, we model the power coefficient $C_p(U_\infty)$, which is obtained by nondimensionalizing the EPC data. A Gaussian process regression (GPR) [60] can be used to model the power coefficient at each wind speed as a Gaussian probability distribution, $C_p(U_\infty) \sim \mathcal{N}[\mu_{C_p}(U_\infty), \sigma_{C_p}(U_\infty)]$. Figure 15a is generated from the 100-h wind turbine control simulation, producing approximately 5000 (U_∞, C_p) pairs plotted as individual data points. These data are used to fit the GPR model and obtain the mean and standard deviation of the power coefficient as functions of the wind speed. We employed a composite kernel for the GPR model, combining a radial basis function kernel and a white noise kernel. The hyperparameters were optimized via maximum likelihood estimation, ensuring a robust fit. The mean fit, $\mu_{C_p}(U_\infty)$, is shown as a red line. To show the standard deviation $\sigma_{C_p}(U_\infty)$, the 95% confidence interval is depicted as a shaded red area, as this interval is mathematically defined as $[\mu_{C_p}(U_\infty) - 1.96 \cdot \sigma_{C_p}(U_\infty), \mu_{C_p}(U_\infty) + 1.96 \cdot \sigma_{C_p}(U_\infty)]$, for each wind speed value, and 1.96 corresponds to the z-score for a 95% confidence level under a Gaussian assumption. The C_p distribution for specific values of the wind speed is illustrated in Figure 15b where it is compared with the histogram of the power coefficient of the data.

Assuming the Weibull probability distribution for wind speed U_∞ combined with the GPR model for the distribution C_p, the estimation of annual wind energy can be performed by computing the expectation of wind energy E_w. Mathematical details on the statistical distributions are provided in Appendix A. Table 5 presents the annual energy estimation for each control strategy. It is important to note that the *Power* DDQN controller is applicable only in the below-rated wind speed region. Therefore, when computing annual wind energy generation with this control, we impose nominal power for wind speeds above the rated value. The *Power* DDQN controller is included in the comparison to ensure that it remains competitive with the Classical Control within the below-rated wind speed range. The *Quiet* DDQN controller is capable of controlling the wind turbine without

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 5 | Annual wind energy generation for the three different controllers.

Controller	E_w (MWh)	$\widehat{\sigma}_{C_p}$
Quiet DDQN	2722	0.024
Power DDQN	3541	0.042
Classic controller	3458	0.057

Note: Additionally, the average standard deviation across wind speed for each controller is presented, $\widehat{\sigma}_{C_p} = \int_{U_{\text{in}}}^{U_{\text{off}}} \sigma_{C_p}(u) f_U(u) du$, details on Appendix A at Equation (A5).

surpassing the SPL threshold and obtains a 78% of the annual energy production compared to classical control. Additionally, in Table 5, the average standard deviation is shown for the GPR power coefficient fit. There, it is shown how the *Quiet* agents exhibit a considerable less variance than the classic control. This is primarily due to the classic controller performing intensive pitch control at high wind speeds to optimize power generation, whereas the *Quiet* agent reduces control activity to comply with noise constraints. Reduced pitch actuation in the *Quiet* agent (see Figure 14b) may lead to lower mechanical wear on the pitch actuator system, potentially extending its operational lifespan. Furthermore, the *Power* agent also demonstrates lower variance than the classic controller, highlighting that RL-based control strategies enable more efficient regulation, even when solely optimizing for power maximization.

5 | Conclusions

In conclusion, integration of RL with wind turbine control holds promise for optimizing energy generation and efficiency while minimizing acoustic environmental impact. A DDQN RL agent can replicate the control strategy of a standard wind turbine controller without prior explicit knowledge of the wind turbine, relying solely on a wind turbine solver for experiential learning. In addition, advanced control strategies can be easily implemented by modifying the reward function. In this work, an RL controller is defined to maximize power output while maintaining acceptable decibel levels, thus incorporating acoustic effects into the control strategy. This shows that MORL is capable of dynamically balancing two different objectives effectively.

An EPC for a silent RL-based wind turbine control strategy is derived from turbulent wind control simulations. This characterization provides a direct means to estimate annual energy production for any site by simply adjusting the yearly wind speed distribution. The methodology is validated using a SWT2.3-93 wind turbine with a rated power of 2.3 MW, and yearly energy production is evaluated for a realistic site using two different RL-based control strategies. The power-based DDQN RL control achieves an energy yield similar to that of a traditional control approach. In contrast, noise-constrained control results in a 22% reduction in annual energy extraction when imposing a maximum noise level of 45-dBA 100-m downwind. Additionally, the silent control requires significantly less pitch actuation, which may reduce mechanical wear on the pitch actuator system and extend its operational lifespan.

While the proposed RL control strategy offers several advantages, it also has certain limitations. One drawback is the need to retrain the model for each specific case, such as different wind turbine designs or observer locations for noise assessment. However, we note that the computational cost of training remains low. The proposed RL control strategy does not account for dynamic effects in the control strategy and does not explicitly model the torque dynamics of the generator, which could limit the real world implementation. Future work will extend the proposed RL framework by incorporating turbine drivetrain dynamics.

Finally, further research directions include investigating Multi-agent reinforcement learning algorithms for cooperative control of wind turbines within farms, which could enhance overall system performance while controlling noise at the farm level.

Acknowledgments

Esteban Ferrer and Oscar A. Marino would like to thank the support of Agencia Estatal de Investigación for the grant "Europa Excelencia" for the project EUR2022-134041 funded by MCIN/AEI/10.13039/501100011033) and the European Union NextGenerationEU/PRTR and also the funding received by the Grant DeepCFD (Project no. PID2022-137899OB-I00) funded by MICIU/AEI/10.13039/501100011033 and by ERDF, EU. This research has been cofunded by the European Union (ERC, Off-coustics, project number 101086075). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. All authors gratefully acknowledge the Universidad Politécnica de Madrid (www.upm.es) for providing computing resources on Magerit Supercomputer and the computer resources at MareNostrum and the technical support provided by Barcelona Supercomputing Center (projects RES-IM-2024-1-0003 and RES-IM-2025-1-0011). Finally, the authors gratefully acknowledge the EuroHPC JU for the project EHPC-REG-2023R03-068 for providing computing resources of the HPC system Vega at the Institute of Information Science.

Peer Review

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/we.70041.

Data Availability Statement

The data used in this research are available from the following open-source repositories:

- Wind turbine specifications: Available at the Zenodo repository.
- Wind conditions dataset: MIDC measurements from June 1, 2023, to June 1, 2024, at 80 m height. Available at MIDC NWTC.

Additional data generated during this study, including OpenFAST input files for the SWT2.3-93 model and NN weights for RL controllers, will be made available upon request to the corresponding author.

References

1. M. Wolsink, M. Sprengers, A. Keuper, T. H. Pedersen, and C. A. Westra, "Annoyance From Wind Turbine Noise on Sixteen Sites in Three Countries," in *European Community Wind Energy Conference*, Vol. 271, (1993): 276.

2. E. Pedersen, F. Van Den Berg, R. Bakker, and J. Bouma, "Response to Noise From Modern Wind Farms in the Netherlands," *Journal of the Acoustical Society of America* 126, no. 2 (2009): 634–643.

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

3. J. L. Davy, K. Burgemeister, and D. Hillman, "Wind Turbine Sound Limits: Current Status and Recommendations Based on Mitigating Noise Annoyance," *Applied Acoustics* 140 (2018): 288–295, https://www.sciencedirect.com/science/article/pii/S0003682X17311428.

4. S. Wagner, R. Bareiß, and G. Guidati, "Noise Mechanisms of Wind Turbines," *Wind Turbine Noise* 1996 (1996): 67–92.

5. J. G. Njiri and D. Söffker, "State-of-the-Art in Wind Turbine Control: Trends and Challenges," *Renewable and Sustainable Energy Reviews* 60 (2016): 377–393, https://www.sciencedirect.com/science/article/pii/S1364032116001404.

6. E. J. Novaes Menezes, A. M. Araújo, and N. S. Bouchonneau da Silva, "A Review on Wind Turbine Control and Its Associated Methods," *Journal of Cleaner Production* 174 (2018): 945–953, https://www.sciencedirect.com/science/article/pii/S0959652617326021.

7. L. E. Andersson, O. Anaya-Lara, J. O. Tande, K. O. Merz, and L. Imsland, "Wind Farm Control - Part I: A Review on Control System Concepts and Structures," *IET Renewable Power Generation* 15, no. 10 (2021): 2085–2108, https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/rpg2.12160.

8. E. Aylı and E. Koçak, "Supervised Learning Method for Prediction of Heat Transfer Characteristics of Nanofluids," *Journal of Mechanical Science and Technology* 37, no. 5 (2023): 2687–2697.

9. M. Abuella and B. Chowdhury, "Random Forest Ensemble of Support Vector Regression Models for Solar Power Forecasting," in *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, (2017): 1–5.

10. X. Huang, Q. Lu, H. Zhou, W. Huang, and S. Wang, "Study on Hydroturbine Power Trend Prediction Based on Machine Learning," *Energy Reports* 10 (2023): 1996–2005.

11. G. Duthé, F. de N Santos, I. Abdallah, W. Weijtjens, C. Devriendt, and E. Chatzi, "Flexible Multi-Fidelity Framework for Load Estimation of Wind Farms Through Graph Neural Networks and Transfer Learning," *Data-Centric Engineering* 5 (2024): e29.

12. V. S. Bokharaie, P. Bauweraerts, and J. Meyers, "Wind-Farm Layout Optimisation Using a Hybrid Jensen–LES Approach," *Wind Energy Science* 1, no. 2 (2016): 311–325.

13. J. Cao, W. Zhu, W. Shen, and Z. Sun, "Wind Farm Layout Optimization With Special Attention on Noise Radiation," in *Journal of Physics: Conference Series*, Vol. 1618. IOP Publishing, (2020): 42022.

14. S. Le Clainche, E. Ferrer, S. Gibson, E. Cross, A. Parente, and R. Vinuesa, "Improving Aircraft Performance Using Machine Learning: A Review," *Aerospace Science and Technology* 138 (2023): 108354, https://www.sciencedirect.com/science/article/pii/S1270963823002511.

15. P. Garnier, J. Viquerat, J. Rabault, A. Larcher, A. Kuhnle, and E. Hachem, "A Review on Deep Reinforcement Learning for Fluid Mechanics," *Computers & Fluids* 225 (2021): 104973, https://www.sciencedirect.com/science/article/pii/S0045793021001407.

16. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, (1998), http://www.cs.ualberta.ca/~sutton/book/the-book.html.

17. S. Oerlemans, "Reduction of Wind Turbine Noise Using Blade Trailing Edge Devices," in *22nd AIAA/CEAS Aeroacoustics Conference*, (2016): 3018.

18. M. Sadeghimalekabadi, A. Davari, and M. Fadaei, "Noise Reduction in Small Wind Turbines With Optimized Serrated Blades," *Physics of Fluids* 36, no. 5 (2024): 57135.

19. S. Deshmukh, S. Bhattacharya, A. Jain, and A. R. Paul, "Wind Turbine Noise and Its Mitigation Techniques: A Review," *Energy Procedia* 160 (2019): 633–640.

20. M. F. Barone, "Survey of Techniques for Reduction of Wind Turbine Blade Trailing Edge Noise," (2011), Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA.

21. S. Lee, S. Lee, J. Ryi, and J.-S. Choi, "Design Optimization of Wind Turbine Blades for Reduction of Airfoil Self-Noise," *Journal of Mechanical Science and Technology* 27 (2013): 413–420.

22. J. Kou, L. Botero-Bolívar, R. Ballano, et al., "Aeroacoustic Airfoil Shape Optimization Enhanced by Autoencoders," *Expert Systems with Applications* 217 (2023): 119513.

23. G. Ciaburro, G. Iannace, V. Puyana-Romero, and A. Trematerra, "Machine Learning-Based Tools for Wind Turbine Acoustic Monitoring," *Applied Sciences* 11, no. 14 (2021): 6488.

24. X. Wu, W. Hu, Q. Huang, C. Chen, M. Z. Jacobson, and Z. Chen, "Optimizing the Layout of Onshore Wind Farms to Minimize Noise," *Applied Energy* 267 (2020): 114896.

25. P. Chen, D. Han, F. Tan, and J. Wang, "Reinforcement-Based Robust Variable Pitch Control of Wind Turbines," *IEEE Access* 8 (2020): 20493–20502.

26. J. Xie, H. Dong, and X. Zhao, "Data-Driven Torque and Pitch Control of Wind Turbines via Reinforcement Learning," *Renewable Energy* 215 (2023): 118893, https://www.sciencedirect.com/science/article/pii/S0960148123007905.

27. J. E. Sierra-Garcia, M. Santos, and R. Pandit, "Wind Turbine Pitch Reinforcement Learning Control Improved by PID Regulator and Learning Observer," (2020), https://www.sciencedirect.com/science/article/pii/S0952197622000598.

28. E. Kadoche, S. Gourvénec, M. Pallud, and T. Levent, "MARLYC: Multi-Agent Reinforcement Learning Yaw Control," *Renewable Energy* 217 (2023): 119129, https://www.sciencedirect.com/science/article/pii/S0960148123010431.

29. A. Saenz-Aguirre, E. Zulueta, U. Fernandez-Gamiz, J. Lozano, and J. M. Lopez-Guede, "Artificial Neural Network Based Reinforcement Learning for Wind Turbine Yaw Control," *Energies* 12, no. 3 (2019): 436, https://www.mdpi.com/1996-1073/12/3/436.

30. A. Kushwaha, M. Gopal, and B. Singh, "Q-Learning Based Maximum Power Extraction for Wind Energy Conversion System With Variable Wind Speed," *IEEE Transactions on Energy Conversion* 35, no. 3 (2020): 1160–1170.

31. C. Wei, Z. Zhang, W. Qiao, and L. Qu, "An Adaptive Network-Based Reinforcement Learning Method for MPPT Control of PMSG Wind Energy Conversion Systems," *IEEE Transactions on Power Electronics* 31, no. 11 (2016): 7837–7848.

32. M. Abkar, N. Zehtabiyan-Rezaie, and A. Iosifidis, "Reinforcement Learning for Wind-Farm Flow Control: Current State and Future Actions," *Theoretical and Applied Mechanics Letters* 13, no. 6 (2023): 100475.

33. National Renewable Energy Laboratory, "OpenFAST," U.S. Department of Energy, (2024), Accessed: 2024-04-07.

34. W. J. Zhu, N. Heilskov, W. Z. Shen, and J. N. Sørensen, "Modeling of Aerodynamically Generated Noise From Wind Turbines," (2005).

35. P. Moriarty, G. Guidati, and P. Migliore, "Recent Improvement of a Semi-Empirical Aeroacoustic Prediction Code for Wind Turbines," in *10th AIAA/CEAS Aeroacoustics Conference*, (2004): 3041.

36. T. F. Brooks, D. S. Pope, and M. A. Marcolini, "Airfoil Self-Noise and Prediction," (1989).

37. J. Christophe, S. Buckingham, C. Schram, and S. Oerlemans, "Zephyr-Large on Shore Wind Turbine Benchmark," (2022), Accessed : 2024-04-07.

38. F. Bertagnolio, N. Sørensen, J. Johansen, and P. Fuglsang, "Wind Turbine Airfoil Catalogue," (2001).

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

39. J. C. Matthew, "A Method for Designing Generic Wind Turbine Models Representative of Real Turbines and Generic Siemens SWT-2.3-93 and VESTAS v80 specifications," (2012), NREL, the United states of America.

40. C. J. C. H. Watkins and P. Dayan, "Q-Learning," *Machine Learning* 8 (1992): 279–292.

41. V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Playing ATARI With Deep Reinforcement Learning," (2013), arXiv preprint arXiv:1312.5602.

42. H. Van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning With Double Q-Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30, (2016).

43. T. P. Lillicrap, J. J. Hunt, A. Pritzel, et al., "Continuous Control With Deep Reinforcement Learning," (2015), arXiv preprint arXiv:1509.02971.

44. K. Van Moffaert, M. M. Drugan, and A. Nowé, "Scalarized Multi-Objective Reinforcement Learning: Novel Design Techniques," in *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE, (2013): 191–199.

45. K. Van Moffaert and A. Nowé, "Multi-Objective Reinforcement Learning Using Sets of Pareto Dominating Policies," *Journal of Machine Learning Research* 15, no. 1 (2014): 3483–3512.

46. H. Mossalam, Y. M. Assael, D. M. Roijers, and S. Whiteson, "Multi-Objective Deep Reinforcement Learning," (2016), arXiv preprint arXiv:1610.02707.

47. J. E. Sierra-García and M. Santos, "Exploring Reward Strategies for Wind Turbine Pitch Control by Reinforcement Learning," *Applied Sciences* 10, no. 21 (2020): 7462.

48. D. Soler, O. Mariño, D. Huergo, M. de Frutos, and E. Ferrer, "Reinforcement Learning to Maximize Wind Turbine Energy Generation," *Expert Systems with Applications* 249 (2024): 123502, https://www.sciencedirect.com/science/article/pii/S0957417424003671.

49. J. Viquerat and E. Hachem, "Parallel Bootstrap-Based on-Policy Deep Reinforcement Learning for Continuous Fluid Flow Control Applications," *Fluids* 8, no. 7 (2023): 208.

50. B. Font, F. Alcántara-Ávila, J. Rabault, R. Vinuesa, and O. Lehmkuhl, "Active Flow Control of a Turbulent Separation Bubble Through Deep Reinforcement Learning," in *Journal of Physics: Conference Series*, Vol. 2753. IOP Publishing, (2024): 12022.

51. M. Towers, J. K. Terry, A. Kwiatkowski, et al., "Gymnasium," (2023), Zenodo, https://zenodo.org/record/8127025.

52. S. Guadarrama, A. Korattikara, O. Ramirez, et al., "TF-Agents: A Library for Reinforcement Learning in Tensorflow," (2018), [Online; accessed 06-April-2024].

53. F. Chollet, "Keras," (2015), https://keras.io.

54. D. P. Kingma and J. Ba, "ADAM: A Method for Stochastic Optimization," (2014), arXiv preprint arXiv:1412.6980.

55. D. Jager and A. Andreas, "Nrel National Wind Technology Center (NWTC): M2 Tower; Boulder, Colorado (Data)," (1996), https://doi.org/10.5439/1052222, NREL Report No. DA-5500-56489.

56. E. Garcia-Bustamante, J. F. González-Rouco, P. A. Jiménez, J. Navarro, and J. P. Montávez, "A Comparison of Methodologies for Monthly Wind Energy Estimation," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 12, no. 7 (2009): 640–659.

57. E. García-Bustamante, J. F. González-Rouco, P. A. Jiménez, J. Navarro, and J. P. Montávez, "The Influence of the Weibull Assumption in Monthly Wind Energy Estimation," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 11, no. 5 (2008): 483–502.

58. P. Spiru and P. L. Simona, "Wind Energy Resource Assessment and Wind Turbine Selection Analysis for Sustainable Energy Production," *Scientific Reports* 14, no. 1 (2024): 10708.

59. A. T. Abolude and W. Zhou, "Assessment and Performance Evaluation of a Wind Turbine Power Output," *Energies* 11, no. 8 (2018): 1992.

60. D. J. C. MacKay, "Introduction to Gaussian Processes," *NATO ASI Series F Computer and Systems Sciences* 168 (1998): 133–166.

61. A. Gambier and Y. Yunazwin Nazaruddin, "Nonlinear PID Control for Pitch Systems of Large Wind Energy Converters," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, (2018): 996–1001.

62. T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Wind Energy Handbook*, (Wiley, 2011), https://books.google.es/books?id%3Ddip2LwCRCscC.

63. K. J. Åström and T. Hägglund, *Advanced PID Control*, (ISA - The Instrumentation, Systems and Automation Society, 2006).

64. S. P. Mulders and J. W. Van Wingerden, "Delft Research Controller: An Open-Source and Community-Driven Wind Turbine Baseline Controller," in *Journal of Physics: Conference Series*, Vol. 1037. IOP Publishing, (2018): 32009.

Appendix A

Statistical Details for the EPC Model

Let us consider a two-dimensional random variable $\mathbb{X} = (C_p, U)$. This random variable models the probability of obtaining a certain wind speed with a certain power coefficient. The marginal distribution of wind speed accounts for the global distribution of wind speed of the localization of the wind turbine. Meanwhile, the distribution of power coefficients measures the performance of the wind turbine at different wind speeds.

There exists an a priori unknown joint probability density $f(c_p, u)$. The power generated by the wind turbine, P, is a function of this random variable, so it is itself a random variable,

$$P = \frac{1}{2}\rho A C_p U^3.$$

The wind energy, E_w, that the wind turbine extracts from the wind for a given period can be written as $E_w = \int_0^T P dt$. However, this statistical model does not include information about the temporal evolution of C_p and U. We can compute the expectation of the wind energy using the expectation of the power over a period of time.

$$\mathbb{E}[E_w] = \Delta T \mathbb{E}[P] = \Delta T \int \frac{1}{2}\rho A c_p u^3 f(c_p, u) dc_p du, \tag{A1}$$

where ΔT denotes the period of time. Notice that this makes sense only if the unknown wind speed evolution $U_\infty(t)$ fits into the annual distribution modeled by the random variable U.

Although we do not know the joint PDF, we know that the wind speed random variable U follows a Weibull distribution. Therefore, the marginal probability density function of U, $f_U(u)$ is a weibull PDF that follows Equation (7).

On the other hand, we can obtain the distribution of power coefficient for each wind speed value. This would be the conditioned power coefficient PDF, that is, $f_{C_p}(c_p|U = u)$. From these two PDFs, we can obtain the joint PDF using the following relation:

$$f_{C_p}(c_p|U = u) = \frac{f(c_p, u)}{f_U(u)}. \tag{A2}$$

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

In this work, the conditional power coefficient PDF is obtained using a GPR algorithm. Therefore, its density function is the following:

$$f_{C_p}(c_p|U=u) = \frac{1}{\sqrt{2\pi\sigma_{C_p}(u)^2}} \exp\left(\frac{c_p - \mu_{C_p}(u)}{\sigma_{C_p}(u)}\right)^2, \tag{A3}$$

where the mean $\mu_{C_p}(u)$ and standard deviation $\sigma_{C_p}(u)$ are obtained from the wind turbine control simulations.

Finally, the expectation of the wind energy can be computed as follows:

$$\begin{aligned}\mathbb{E}[E_w] \quad &= \Delta T \mathbb{E}[P] = \left(\frac{1}{2}\Delta T \rho A\right) \\ &\int_{U_{\text{in}}}^{U_{\text{off}}} \left(\int_0^{C_{p,\text{nom}}} c_p f_{C_p}(c_p|U=u) dc_p \right) u^3 f_U(u) du.\end{aligned} \tag{A4}$$

Notice that the inner integral is the expectation of the conditional distribution, $\mathbb{E}[C_p|U=u] = \mu_{C_p}(u)$. Hence, the expectation of the wind energy only requires the mean of the distribution fitted by the GP. The variance of the control, $\sigma_{C_p}(u)^2$, has no influence on the estimation of the wind energy. However, it gives us information about the control and can be useful to measure; this can be done by computing the expectation of the variance.

$$\mathbb{E}[\sigma_{C_p}(U)] = \int_{U_{\text{in}}}^{U_{\text{off}}} \sigma_{C_p}(u) f_U(u) du. \tag{A5}$$

Appendix B

Sensitivity of the EPC Estimation

The EPC should accurately capture the relationship between wind speed and power performance, reflecting the influence of the control strategy on this dependency. To ensure that the EPC remains representative for annual energy estimation, the wind speed data used in its derivation must be generalizable throughout the year. Here, we demonstrate that a 100-h subset of wind speed data is sufficient to accurately approximate the EPC.

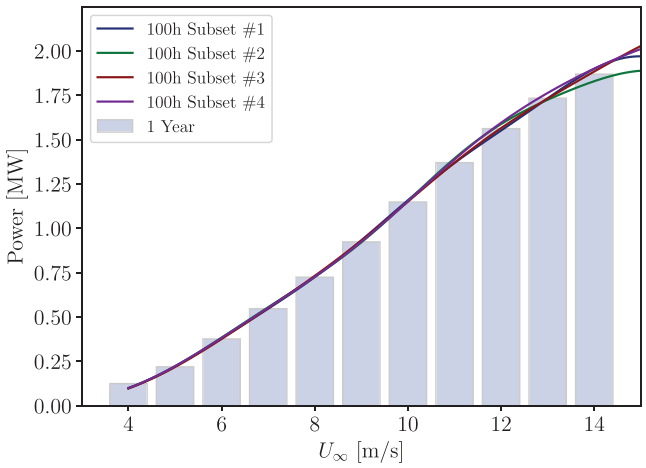


FIGURE B.1 | Comparison of the EPC obtained for the *Quiet* agent using different wind speed data for the simulations. The EPC obtained with the wind speed data of the complete year ( ) is compared against the mean power GPR fit for different 100-hour subsets of wind speed.

Seasonal variations in wind speed throughout the year or how wind speed is distributed annually are not taken into account directly in the EPC but in the Weibull distribution $f_U(u)$ for the annual energy estimate at equation (A4). However, this EPC should capture how the control deals with turbulent wind speed conditions. Assessing whether a 100-h wind speed subset of data is representative for a complete year in terms of turbulence intensity can be a challenge. To address this, we directly compared the EPC obtained from the complete year of wind speed data with those derived from multiple 100-h subsets. This analysis, conducted for the *Quiet* agent, is illustrated in Figure B.1.

To facilitate comparison, the EPCs for different subsets are represented using the GPR fit for the power coefficient described in Section 4.3, yielding the power curve $\frac{1}{2}\rho A U_\infty^3 \mu_{C_p}(U_\infty)$. The EPC for the entire year is represented using the binning strategy from Figure 14a, as applying the GPR fit to the entire data set is computationally prohibitive due to the scaling of the kernel matrix.

As shown in Figure B.1, the EPCs obtained from different 100-h subsets closely match the EPC calculated using the entire year of data, confirming the consistency of the approach. Moreover, the strong agreement among the subsets highlights the robustness of the control strategy to different wind speed profiles. The only selection criterion for the subsets is that they must contain sufficient data across the full wind speed range to ensure that the control encounters a representative distribution of operating conditions.

Appendix C

Standard Wind Turbine Control Strategy

The control strategy for variable-speed horizontal-axis wind turbines can be divided into four regions based on wind speed. Although each region definition may vary depending on the specific control design, the fundamental objectives within each region are as follows:

- **Region I**: When the wind speed is below the cut-in value, the turbine cannot operate.
- **Region II**: At wind speeds above the cut-in threshold but below the rated speed, the primary objective is to optimize power generation. This is achieved by adjusting the rotor speed to align with the power curve of the wind turbine, utilizing a predetermined lookup table.
- **Region III**: When wind speeds exceed the rated value, the focus shifts to maintaining a consistent rotor speed across a wide range of wind speeds. This is typically achieved through adjustment of the blade pitch, commonly implemented using a proportional–integral–derivative (PID) control strategy, although there are more sophisticated approaches [61].
- **Region IV**: When the wind speed exceeds the cutoff value, the turbine must be shut down for safety.

The transition between Regions II and III, sometimes referred to as Region II $\frac{1}{2}$, is characterized by maintaining a constant rotor speed. However, there are different options depending on the specific control design. Figure C.1 illustrates these regions on the wind turbine power curve. Further details on classical wind turbine control strategies can be found in the works of [62] or [63].

The controller module in OpenFAST facilitates the customization of controllers. In this study, the wind turbine controller is derived from the OpenFAST implementation of [64], tailored to the characteristics of the SWT wind turbine.

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

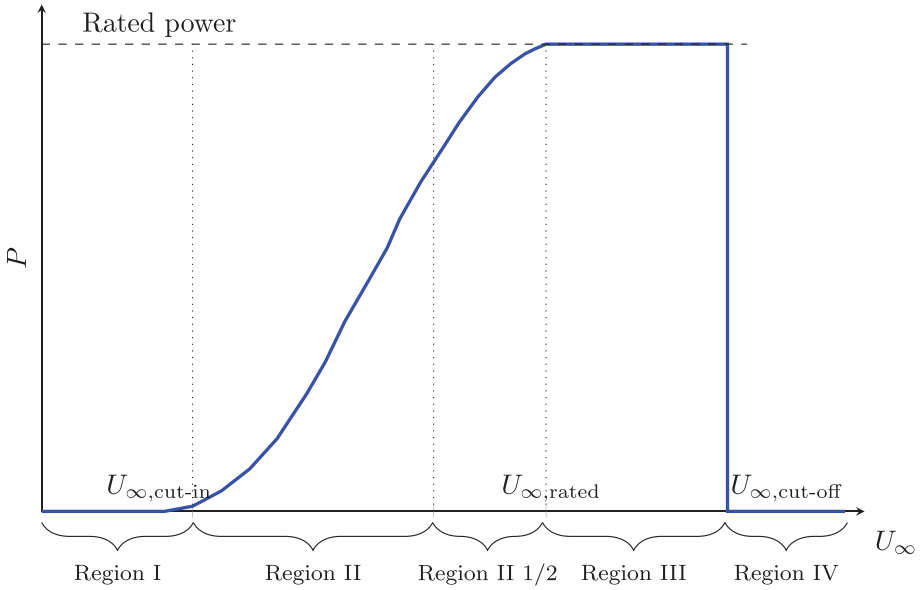


FIGURE C.1 | Control strategy regions for a variable-speed, horizontal-axis wind turbine on the power curve.

10991824, 2025, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/we.70041 by Battelle Memorial Institute Pacific Northwest Division, Wiley Online Library on [04/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License