

Data-centric predictive control with tuna swarm optimization-backpropagation neural networks for enhanced wind turbine performance

Wei Li, Ravi Kumar Pandit^{*}

Centre for Life-cycle Engineering and Management, Cranfield University, UK

ARTICLE INFO

Keywords:

Deep neural networks
TSO-BPNN
Model predictive control
Reinforcement learning
Wind turbine control

ABSTRACT

Wind energy is a significant renewable resource, but its efficient harnessing requires advanced control systems. This study presents a Data-Centric Predictive Control (DPC) system, enhanced by a Tuna Swarm Optimization-Backpropagation Neural Network (TSO-BPNN) for predictive wind turbine control. It's like a smart tool that uses innovative fusion of deep learning, predictive Control, and reinforcement learning. Unlike traditional control methods, the proposed approach uses real-time data to optimize turbine performance in response to fluctuating wind conditions.

The system is validated using simulations on the FAST platform, which demonstrate its superior performance in two critical operational regions. Specifically, in Region II, where the objective is to maximize power extraction from the wind, the DPC achieves a 1.07 % reduction in overshoot and an improvement of 36.14 units in steady-state error compared to traditional methods. The response time remains comparable to existing Model Predictive Control (MPC) strategies, ensuring real-time applicability without sacrificing efficiency. In Region III, where maintaining constant power output is crucial, the DPC outperforms both the baseline and MPC methods, reducing overshoot by 0.58 % and improving accuracy by 17.27 units compared to the baseline method. These results highlight the effectiveness of the proposed DPC system in optimizing turbine performance under variable wind conditions, offering a significant improvement over traditional methods in both accuracy and control precision.

1. Introduction

The abundance of wind energy, a clean and renewable power source, is increasingly acknowledged across various nations. The upward trend in its actual utilization rate corroborates this increasing prominence [1]. Within the context of new energy development, wind power generation has manifested itself as one of the most promising and forward-looking technologies. Around the world, wind farms have been constructed on a large scale [2]. The UK, in particular, has channelled significant investments into this sector, making it a hotspot of innovation and growth. By 2023, the UK had over 11 thousand wind turbines with a total installed capacity of 28 GW (GW): 14 GW onshore and 14 GW offshore [3], and with the impetus towards a green transition, the trajectory is only expected to rise [4]. These wind turbines function by transmuting the wind's kinetic energy into electrical power.

However, the inherent randomness of wind velocities can induce

considerable fluctuations in wind power, precipitating variations in grid voltage or frequency [5]. This can degrade the quality of the electric power and catalyse a series of stability complications. To mitigate these issues, it is imperative to institute a control strategy facilitating wind turbine operation under low-load conditions [6]. The control strategies employed by wind power systems are paramount in optimizing their operational performance, guaranteeing safety, and maximizing electricity production.

As the core of wind turbine operation, the control system is divided into four main working regions classified by inflow wind speeds: two for shutdown and two for operation [7]. The shutdown regions, including Region I (below cut-in speed) and Region IV (above cut-out speed), don't involve power generation. In contrast, Region II (above cut-in and below rated speed) uses torque control for maximum power extraction, while Region III (above rated and below cut-out speed) employs pitch control to sustain power at rated levels. Therefore, efficient control in Regions II

^{*} Corresponding author.

E-mail address: ravi.pandit@cranfield.ac.uk (R.K. Pandit).

<https://doi.org/10.1016/j.renene.2024.121821>

Received 18 September 2023; Received in revised form 24 October 2024; Accepted 4 November 2024

Available online 6 November 2024

0960-1481/Crown Copyright © 2024 Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

& III plays a crucial role in the maximum power point tracking (MPPT) tasks of wind turbines. Traditional control systems have been somewhat effective in these aspects, but with the advent of sophisticated data-driven techniques, there is room for significant improvement. The integration of data-centric predictive control into wind turbine operation has the potential to revolutionize the way wind energy is harnessed, controlled, and maintained. By using advanced algorithms that can analyse large volumes of data in real time, it is now possible to make accurate predictions about the future status of wind turbines. This, in turn, enables the control system to make proactive adjustments, optimizing power generation, and reducing the need for reactive maintenance.

Traditional control strategies, such as the Proportional-Integral-Derivative (PID) controller, have been widely adopted in wind turbine control due to their simplicity and ease of implementation. However, PID controllers often struggle with handling system nonlinearities and rapid wind speed fluctuations, leading to suboptimal performance. For instance, Sierra-García and Santos [8] noted that while the PID controller's performance is relatively good at low and medium wind speeds, it still lags by 21 % compared to a hybrid control strategy combining fuzzy logic and deep learning. S. Hamoodi [9] and team studied the speed control of wind turbines, comparing traditional PID controllers with adaptive PID controllers based on fuzzy logic. Quang-Vi Ngo and colleagues [10] designed a fuzzy logic-based PID controller for pitch angle control in small-scale wind turbine systems. Yajun Fan and Mu [11] introduced a novel fuzzy PID controller for optimal power extraction from wind, suitable for hydraulic offshore wind turbines. While fuzzy PID controllers offer advantages in handling system nonlinearities and uncertainties, their complexity and tuning challenges remain significant drawbacks. Such inherent limitations necessitate the exploration of more advanced strategies, such as Model Predictive Control (MPC).

MPC is a model-based control strategy that can consider system model information and handle constraints. By optimizing a predicted cost function, MPC determines the control input, achieving optimized control of the wind turbine. Yang et al. [12] underscored the importance of regulating system frequency by leveraging MPC, particularly emphasizing the dynamic nature of wind turbines and the forecasting of their power output. This sentiment was echoed by Bassi et al. [13] who demonstrated the superior efficacy of MPC over traditional PID controllers across varying wind speeds. Wei et al. [14] further extended the application of MPC by introducing a coordinated droop control mechanism, aiming to enhance the high-voltage ride-through capabilities of large-scale wind farms. Babaghorbani et al. [15] provided insights into the potential of MPC in managing low voltage ride-through during system faults, specifically for wind energy conversion systems based on permanent magnet synchronous generators. Bárcena et al. [16] took a structural approach, focusing on the reduction of structural loads in wind turbine components through MPC. On a broader scale, Chen et al. [17] proposed an MPC strategy that optimizes active power set-points at the cluster level, ensuring coordinated power distribution. Validating these theoretical advancements, Dickler et al. [18] conducted a full-scale field test, emphasizing the real-world applicability of a linear time variant MPC system for a 3 MW wind turbine. Fan et al. [19] integrated particle swarm optimization with MPC, addressing the inherent unpredictability of wind turbine generation. Jiang et al. [20] proposed an MPC design that leverages the optimal tip-speed ratio method for maximum power point tracking in direct-driven permanent magnet synchronous generator-based wind energy conversion systems. While the advantages of MPC, such as its ability to handle system constraints and predictive capabilities, are evident, challenges related to computational complexity and the necessity for precise system models persist. However, the computational complexity of MPC is high, and it requires an accurate system model.

Reinforcement learning (RL) has also gained traction in recent years, showing promise in adapting to real-time changes in wind conditions.

However, many RL-based methods face stability and training convergence issues, limiting their effectiveness [21,22]. Autor of [23] propose techniques to optimized vertical-axis wind turbines (VAWT) using Bayesian RL for small-scale turbines. Using the Markov chain Monte Carlo (MCMC) algorithm, their approach effectively enhanced power capture from wind transients. Jia et al. [24] introduced RL-TAD, an RL-based method for optimal twist angle distribution (TAD) of wind turbine blades. Outperforming traditional genetic algorithms, RL-TAD quickly converges to the best TAD offline and provides fast TAD searches for different wind speeds online, enabling real-time blade control. TOMIN et al. [25] introduced RL-based adaptive control techniques for wind turbines, enhancing efficiency while reducing structural stress. Saenz-Aguirre et al. [26] introduced a RL-based yaw control strategy for wind turbines, optimizing power and minimizing mechanical loads. Validated with the NREL 5-MW turbine and real wind data, it outperforms classical control methods. Westerbeck et al. [27] introduced WINDL, an RL-trained neural network policy for wind turbine load control, achieving significant blade load reductions in both steady and turbulent winds. This advancement showcases the potential of RL in optimizing modern turbine controls, surpassing traditional strategies.

A common characteristic of the RL methods in these wind turbine applications is their affiliation with model-free RL [28]. These investigations underscore that RL introduces novel avenues for wind turbine control, adeptly managing the intricacies and uncertainties inherent to wind turbines. However, despite the commendable performance of RL techniques in this domain, they grapple with challenges such as the necessity for training data, algorithmic stability, and convergence issues. To address these concerns, Xie et al. [29] presented an model-based reinforcement learning (MBRL) control method for wind turbines, blending deep neural networks (DNNs) with MPC. Demonstrating reduced computational complexity and heightened robustness, its effectiveness is confirmed through high-fidelity simulations. This also suggests that integrating MPC with RL, a model-based RL approach, presents a nuanced and stable alternative for optimizing wind turbine operations. Table 1 summarises the popularly used control techniques strength and limitation related to wind turbines operations.

This research introduces a data-centric predictive control method for wind turbine, which is rooted in a model-based reinforcement learning wind turbine control method, seeks to explore and develop a predictive control model that is rooted in data analytics, machine learning, and real-time computation. Specifically, the TSO algorithm is harnessed to augment the BPNN, facilitating precise predictions of future system states. This design not only capitalizes on the online system dynamics updating feature of MPC but also enhances the accuracy and stability of state prediction through the TSO-BPNN integration. Consequently, the controller is adept at adapting to real-time system dynamics shifts while retaining the structural and computational benefits of MPC. Then the predicted future state information derived from the TSO-BPNN is

Table 1
Comparative analysis of traditional wind turbine control methods.

Control Method	Advantages	Disadvantages	References
PID Control	Simple, easy to implement, widely adopted	Poor performance in handling non-linearities and rapid changes	[30,31]
Fuzzy PID Control	Better handling of non-linear systems, adaptable to uncertainties	Complex tuning process, increased computational demand	[8,9]
Model Predictive Control (MPC)	Predicts future states, handles constraints efficiently	High computational complexity, requires precise system models	[29,32]
Reinforcement Learning (RL)	Adaptive to real-time changes, good for complex dynamic systems	Requires large amounts of training data, stability issues	[22,33]

channelled into a DNN for reinforcement learning. This configuration empowers the algorithm to execute versatile and efficient control strategies in intricate, nonlinear scenarios. The controller is tailored to optimize a meticulously crafted reward function, enabling it to refine control actions to align with specific performance benchmarks, thus achieving targeted system performance enhancement. Additionally, the integration of Huber loss into the reward function bolsters the robustness of the approach, a pivotal feature for managing outliers and noise in reward signals, further narrowing the gap between the learned dynamics and the actual environment. The performance of this hybrid control scheme is compared with the reference OpenFAST controller for a National Renewable Energy Laboratory (NREL) 5 MW turbine in the final part. The key contributions of this research, when juxtaposed with related studies, can be delineated as follows:

- (1) Data-centric Predictive Approach: Unlike traditional wind turbine control strategies, the presented RL methodology is intrinsically data-centric. This research places significant emphasis on the TSO-BPNN integration for state prediction. This data-centric predictive approach ensures optimal control policies without being tethered to analytical wind turbine models, capturing and updating system information in real-time.
- (2) Model-Based RL: While most existing RL-based wind turbine control strategies are anchored in model-free RL, often rooted in the renowned deep deterministic policy gradient (DDPG) [34] algorithm, they tend to grapple with computational intensity and diminished learning efficiency. This research pioneers a wind turbine control approach grounded in "model-based" RL, leveraging real-time data to craft and refresh a DNN-based surrogate model, making the reward function learning and optimal control policy more stable and efficient.
- (3) Real-time DNN Update: The proposed DNN structure, acting as an information conduit, is refreshed by a real-time dataset comprising current state & action measurements. This design captures potential online shifts in wind turbine models, especially when unforeseen faults arise, bolstering control performance during unexpected uncertainties or faults, thereby enhancing the reliability and applicability of the entire wind turbine control system.

The subsequent sections of this thesis are structured as follows: Section 2 articulates the aim and objectives, laying the groundwork for the methodologies discussed. The methodologies are comprehensively detailed in Section 3, starting with the modelling of the wind turbine, followed by the construction of the future state prediction model, and culminating in the model-based reinforcement learning for wind turbine control. Section 4 presents the results and offers an in-depth discussion, encompassing the outcomes of the future state prediction and the numerical simulations under various conditions. Finally, Section 5 summarizes the key findings and implications of the research.

2. Novelty of work

Wind energy plays a pivotal role in the transition to renewable energy sources, but its efficient utilization is challenged by the variability of wind conditions. Previous research, exemplified by Xie et al. [29], has made progress in applying Model-Based Reinforcement Learning (MBRL) to wind turbine control. However, a critical research gap exists in the precision and robustness of state predictions, particularly vital for real-time control in wind turbines. This study addresses this gap by introducing innovative strategies. It leverages a data-centric predictive paradigm that integrates the Tuna Swarm Optimization-Backpropagation Neural Network (TSO-BPNN) for state predictions, resulting in enhanced precision and robustness. Furthermore, in the reinforcement learning phase, the integration of a reward function combined with Huber loss enhances the algorithm's resilience,

particularly when faced with anomalies and noise in reward signals. These novel approaches not only address latent intricacies in previous research but also underscore improved efficacy and reliability in wind turbine control mechanisms, contributing significantly to the field of wind turbine control and the broader objective of efficient renewable energy generation.

3. Methodologies

3.1. Modelling of wind turbine

3.1.1. Wind turbine fundamentals

The aerodynamic paradigm of a wind turbine is intrinsically associated with the quantity of energy it extracts from the ambient wind. The power coefficient, often denoted as C_p , is established as the proportion of power harnessed by the wind turbine in relation to the inherent power potential of the wind itself. A superior value signifies the optimal functioning of the wind turbine, which corresponds to a scenario where the rotational speed of the wind turbine is at its peak efficiency, and the blades are positioned at the ideal angle to harness the maximum energy from the wind. If P_w is the kinetic power incoming to the turbine disk, then the computation of power retrieved from the wind turbine can be demonstrated through the following equation.

$$P = C_p(\lambda, \beta) P_w = \frac{1}{2} \rho \pi C_p(\lambda, \beta) R^2 v_{wind}^3 \quad (1)$$

where ρ is air density, R is rotor radius, v_{wind} represents the wind speed. C_p is contingent upon both the Tip Speed Ratio (TSR) λ and blade pitch angle β . The function $C_p(\lambda, \beta)$ is predominantly derived empirically, albeit it can be approximated mathematically by Equations (2) and (3).

$$C_p(\lambda, \beta) = c_1 \left(\frac{c_2}{\lambda_i} - c_3 \beta - c_4 \beta^{c_5} - c_6 \right) e^{-\frac{c_7}{\lambda_i}} \quad (2)$$

$$\lambda_i = \left[\left(\frac{1}{\lambda + c_8 \beta} \right) - \left(\frac{c_9}{\beta^3 + 1} \right) \right]^{-1} \quad (3)$$

Here, c_1 to c_9 are a series of constant values characterizing the wind turbine, typically ascertained using identification techniques. This formulation reflects the nonlinear dynamics inherent in the turbine operation, influenced by the wind device's technical specifications, as embodied by parameters c_1 to c_9 that describe a unique type of wind turbine. This equation is routinely employed to generate a spectrum of curves demonstrating the turbine's performance under varying wind conditions, with each curve corresponding to a distinct pitch angle (Fig 1). Furthermore, it serves as a tool for approximating the optimal C_p , λ and β of the wind turbine.

The definition of the TSR is given by Equation (4).

$$\lambda = \frac{\omega_r R}{v_{wind}} \quad (4)$$

where ω_r is rotor speed. For each specified wind speed, there exists an optimal rotor rotational speed that yields the maximum C_p . This particular speed correlates to the ideal tip speed ratio. In the present study, the 5 MW wind turbine model developed by NREL, serves as the basis for analysis. The configurations for this model are detailed in Table 2. These specifications were derived from aerodynamic simulations carried out using the OpenFAST software (by NREL). Post-processing of the data was conducted utilizing both MATLAB and Python tools see Table 7.

3.1.2. Simulation environment

Based on the modelling of the wind turbine generator system in Ref. [35], if θ represents the torsion angle, the unified nonlinear comprehensive model of the wind turbine generator system can be presented as follows:

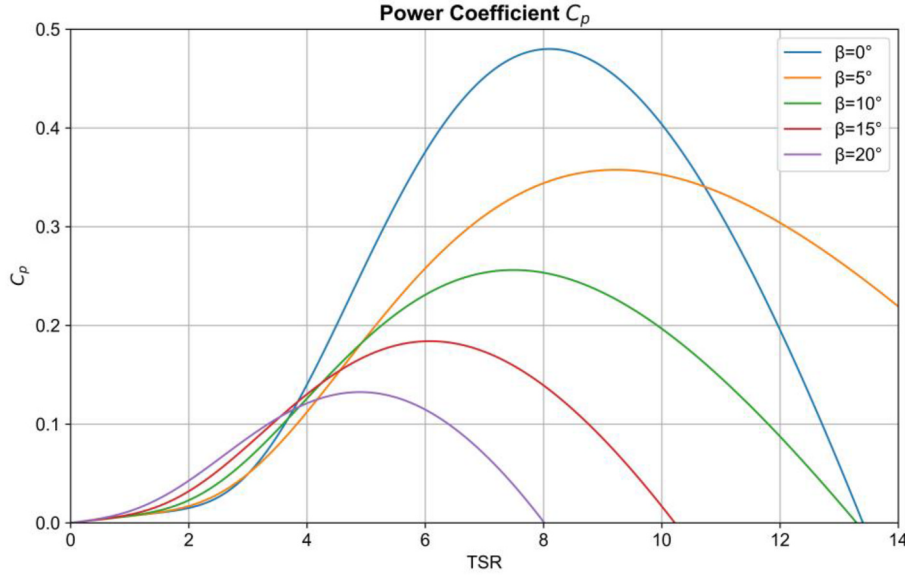
Fig. 1. C_p versus TSR for a typical wind turbine.

Table 2

Parameters of the NREL 5-MW wind turbine.

Parameter	Description	Value
P_{rated}	Rating	5 MW
R	Rotor radius	63 m
N_b	Number of blades	3
ρ	Air density	1.22 kg/m ³
J_r	Rotor moment of inertia	3.54×10^7 kg/m ²
J_g	Generator moment of inertia	5.34×10^2 kg/m ²
K_s	Low speed shaft spring constant coefficient	8.67×10^8 N m/rad
D_s	Low speed shaft viscous friction coefficient	6.22×10^6 N m s/rad
N_g	Gear box ratio	97:1
v_{cut-in}	Cut-in wind speed	3 m/s
v_{rated}	Rated wind speed	11.4 m/s
$v_{cut-out}$	Cut-out wind speed	25 m/s

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega}_r \\ \dot{\omega}_g \end{bmatrix} = \begin{bmatrix} \omega_r - \frac{\omega_g}{N_g} \\ \frac{T_r}{J_r} - \frac{D_s}{J_r} \omega_r + \frac{D_s}{J_r N_g} \omega_g - \frac{K_s}{J_r} \theta \\ \frac{D_s}{J_r N_g} \omega_r - \frac{D_s}{J_r N_g^2} \omega_g + \frac{K_s}{J_g N_g} \theta - \frac{T_{g,ref}}{J_g} \end{bmatrix} \quad (5)$$

The wind speed and generator speed together constitute the state vector of the dynamical system to be controlled.

$$s = [v_{wind} \ \omega_g]^T \quad (6)$$

The actions of the controller are provided by the rate of change of the pitch angle (β) and generator torque (T_g). Ref. [36] provides an update for β and T_g under the action of the controller within the time integral as:

$$\beta(t + \Delta t) = \beta(t) + \dot{\beta} \Delta t \quad (7)$$

$$T_g(t + \Delta t) = T_g(t) + \dot{T}_g \Delta t \quad (8)$$

where Δt is the time step in the integration. The rates are constrained within a specific range, with $\dot{\beta}$ falling between $-\dot{\beta}_{max}$ and $\dot{\beta}_{max}$, and $\dot{T}$ falling between $-\dot{T}_{max}$ and $\dot{T}_{max}$, as outlined in Table 3.

In order to overcome the sensitivity of traditional wind turbine controllers to model errors and their lack of robustness to real-time dynamic changes, the literature [29] has developed a data-driven reinforcement learning strategy. Based on this foundation, this research

Table 3

Input constraints on the NREL 5-MW wind turbine.

Parameter	Description	Value
ω_r^{rated}	Rated rotor speed	12.1 rpm
ω_g^{rated}	Rated generator speed	1173.7 rpm
T_g^{rated}	Rated generator torque	43.1 kN m
T_g^{max}	Maximum generator torque	47.4 kN m
β_{min}/β_{max}	Minimum/Maximum pitch angle	0°/90°
$\dot{T}_{max}$	Maximum generator torque rate	15 kN m/s
$\dot{\beta}_{max}$	Maximum absolute blade-pitch rate	8°/s

adopts a similar approach called data-centric predictive control (DPC). The establishment of the database begins with analytical models in equations (5)–(8). It then incorporates neural networks for additional predictive enhancement. Relying on physical principles furnishes a robust foundation for the accuracy of the model, thus increasing its precision. In this manner, it not only addresses the shortcomings of traditional controllers but also enhances the generalization ability of the data-driven reinforcement learning strategy.

3.2. Construction of future state prediction

3.2.1. Backpropagation neural network

The Backpropagation Neural Network (BPNN), introduced by Rumelhart et al., in 1986 [37], is a supervised learning method that uses gradient descent. It operates in two phases: forward propagation and backward propagation of errors. Robert Hecht-Nielsen demonstrated that a three-layer BPNN approach can model intricate nonlinear tasks [38]. It's capable of handling any transition from N-dimensions to M-dimensions. Given this, the state prediction module in this research employs a three-layer BPNN setup. As depicted in Fig 2, this BPNN configuration consists of input, hidden, and output layers.

BPNN can be comprehensively described through a series of mathematical formulations. This process elucidates how input information traverses from the input layer, through the hidden layers, and ultimately reaches the output layer. When discrepancies arise between the anticipated and actual outputs, the system undergoes a backpropagation process to adjust and optimize. The mathematical representations for the aforementioned processes are delineated below.

For the hidden layer:

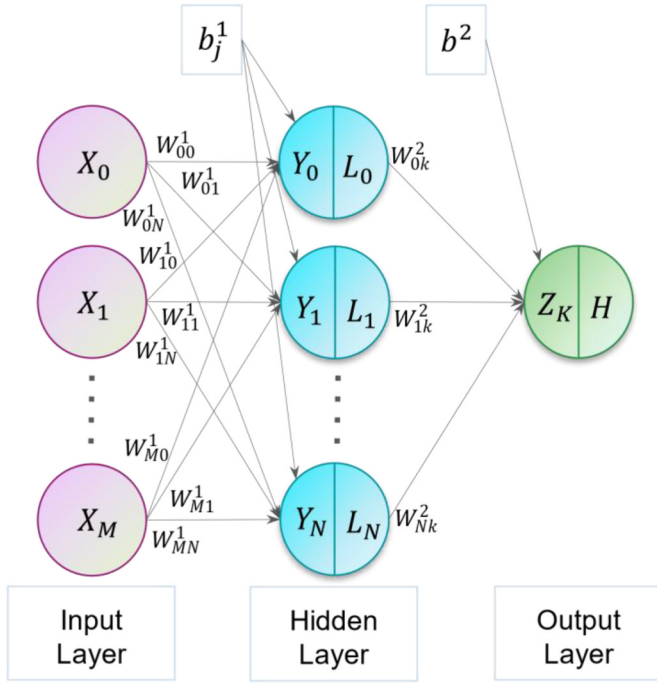


Fig. 2. Topological structure of the BPNN algorithm.

$$Y_j = \sum_{i=0}^M W_{ij}^1 X_i + b_j^1 \quad (9)$$

$$L_j = f(Y_j) = \frac{1}{1 + e^{-Y_j}} \quad (10)$$

For the output layer:

$$Z_K = \sum_{j=0}^N W_{jK}^2 L_j + b^2 \quad (11)$$

$$H = f(Z_K) = \max(0, Z_K) \quad (12)$$

Where X_i represents the input vector. M denotes the total number of nodes in the input layer. W_{ij}^1 signifies the weight connecting the i^{th} neuron in the input layer to the j^{th} neuron. b_j^1 stands for the bias parameter of the hidden layer. Y_j is the input value for the hidden layer's node. L_j represents the output value of the hidden layer node after undergoing a nonlinear transfer function. N is the total number of nodes in the hidden layer. W_{jk}^2 denotes the weight connecting the j^{th} neuron in the hidden layer to the output layer. b^2 symbolizes the bias parameter for the output layer. Z is the input value of the output layer's node. H represents the output value of the output layer's node after application of a nonlinear function.

In instances where the generated output diverges from anticipated values, the system engages in the subsequent backpropagation procedure. This process initiates at the output layer, proceeds through the intermediary hidden layers, and culminates at the input layer, thus effectuating the entire backpropagation phase. Within the framework of the BP algorithm, weight adjustments across layers are orchestrated by minimizing the gradient of the error. Specifically, the weight interlinking the neurons of the input layer and the j^{th} neuron of the hidden layer undergoes a particular rectification as delineated below [39]:

$$W_{ji}(t) = W_{ji}(t-1) + \eta_a \rho_j(t) x_i(t) + \alpha_a \Delta W_{ji}(t) \quad (13)$$

$$b_j(t) = b_j(t) + \eta_b \rho_j(t) x_i(t) + \alpha_b \Delta b_j(t) \quad (14)$$

$$\rho(t) = \frac{1}{2} \sum_{p=1}^G [H - \hat{H}]^2 \quad (15)$$

Within the context of neural network architectures, W and b are respectively characterized as the weight coefficients and threshold parameters. The constants α_a and α_b denote momentum coefficients, instrumental in gauging the influence of prior parameter adjustments on the present propagation direction. The parameters η_a and η_b serve as learning rates, modulating the magnitude of weight updates during training iterations. The error signal for the j^{th} neuron in the hidden layer during the BPNN algorithmic procedure is represented by $\rho_j(t)$. Let G denote the number of instances present within the training dataset. Further, let H represent the anticipated output and $\hat{H}$ signify the actual output produced by the neural network. The process of forward and backward propagation is iteratively executed. This iterative sequence continues until the discrepancy between H and $\hat{H}$ is mitigated to an acceptable threshold or the duration of training reaches a pre-specified limit.

3.2.2. Tuna swarm optimization

This research employs an innovative swarm-based metaheuristic known as Tuna Swarm Optimization (TSO), initially introduced in Ref. [40]. This method is utilized to optimize the BPNN in the future state prediction module of the database within our model. The TSO methodology is fundamentally inspired by two distinct swarm foraging behaviours observed in the species of tuna. The algorithm draws inspiration from the group travel predation strategy utilized by tunas. One such strategy is spiral foraging. As tunas feed, they form a spiral, driving their prey into shallower waters where they become easier to attack. The second strategy is parabolic foraging. Each tuna follows the one before it, forming a parabolic shape to surround and trap their prey.

The optimization process begins by uniformly generating an initial population in the search space at random:

$$X_i^{int} = Rand \cdot (b_u - b_l) + b_l, i = 2, 3, \dots, N_p \quad (16)$$

In the formula, N_p represents the number of individuals in the tuna population, X_i^{int} represents i^{th} initial individual, b_u and b_l are the upper and lower boundaries of the search space, and $Rand$ is a uniformly distributed random vector ranging from 0 to 1.

During the spiral foraging phase, anterior individuals transmit information to posterior individuals, culminating in the establishment of a consistent predation pattern. However, it is misguided to blindly follow the optimal individual for foraging when such an individual cannot locate food. As a remedy, a random coordinate is generated within the search space to serve as a reference point for the spiral search, endowing the TSO with global exploration capabilities. This search method can be analogized from a broad global search to a precise targeted search. As iterations increase, the TSO transitions the reference point for spiral foraging from a random individual to the optimal one. In summary, the final mathematical model for the spiral foraging strategy is presented as:

$$X_i^{t+1} = \begin{cases} \alpha_1 \cdot (X_{rand}^t + \beta \cdot |X_{rand}^t - X_i^t|) + \alpha_2 \cdot X_i^t, i = 1 \\ \alpha_1 \cdot (X_{rand}^t + \beta \cdot |X_{rand}^t - X_i^t|) + \alpha_2 \cdot X_{i-1}^t, i = 2, 3, \dots, N_p \end{cases}, \text{if } rand < \frac{t}{t_{max}} \quad (17)$$

$$X_i^{t+1} = \begin{cases} \alpha_1 \cdot (X_{best}^t + \beta \cdot |X_{best}^t - X_i^t|) + \alpha_2 \cdot X_i^t, i = 1 \\ \alpha_1 \cdot (X_{best}^t + \beta \cdot |X_{best}^t - X_i^t|) + \alpha_2 \cdot X_{i-1}^t, i = 2, 3, \dots, N_p \end{cases}, \text{if } rand \geq \frac{t}{t_{max}} \quad (18)$$

$$\alpha_1 = a + (1 - a) \cdot \frac{t}{t_{max}} \quad (19)$$

$$\alpha_2 = (1 - a) - (1 - a) \cdot \frac{t}{t_{\max}} \quad (20)$$

$$\beta = e^{bl} \cdot \cos(2\pi b) \quad (21)$$

$$l = e^3 \cos(((t_{\max} + 1/t) - 1)\pi) \quad (22)$$

Where X_i^{t+1} represents the i^{th} individual at the $t + 1$ iteration. X_{rand}^t is a reference point randomly generated within the search space. X_{best}^t denotes the current optimal individual. α_1 and α_2 are weighting coefficients governing the propensity of an individual to move towards the optimal and the preceding individual respectively. The constant a determines the extent to which the tuna follows the optimal individual and its predecessor in the initial phase. t stands for the current iteration number, while $t_{\max}$ represents the maximum iteration count. b is a random number uniformly distributed between 0 and 1.

Apart from spiral feeding formations, tuna also utilize a parabolic feeding pattern with the food as a reference. Additionally, tuna scout their immediate surroundings for food. Both strategies are executed simultaneously, each with a 50 % likelihood. The mathematical model is detailed below:

$$X_i^{t+1} = \begin{cases} X_{\text{best}}^t + \text{rand} \cdot (X_{\text{best}}^t - X_i^t) + TF \cdot p^2 \cdot (X_{\text{best}}^t - X_i^t), & \text{if } \text{rand} < 0.5 \\ TF \cdot p^2 \cdot X_i^t, & \text{if } \text{rand} \geq 0.5 \end{cases} \quad (23)$$

TF is a random value of either 1 or -1.

3.2.3. Steps of TSO-BPNN

The workflow of the TSO-BPNN model is illustrated in Fig 3.

The research methodology involves several key steps, including data acquisition, initialization of the BPNN, and TSO algorithm. These steps are followed by fitness computation, parameter optimization, and the implementation of termination criteria. Subsequently, the optimized BPNN undergoes training and testing phases, with error analysis utilizing various metrics to evaluate model performance, as detailed in the following steps:

- 1) Acquisition of sample data, sourced from equations (5)–(8).
- 2) Initialization of the BPNN: The initial architecture of the neural network is determined, encompassing both the initial weight coefficients and bias values connecting the neurons.
- 3) Initialization of the TSO algorithm: The primary weight coefficients as well as bias values are transformed into position vectors

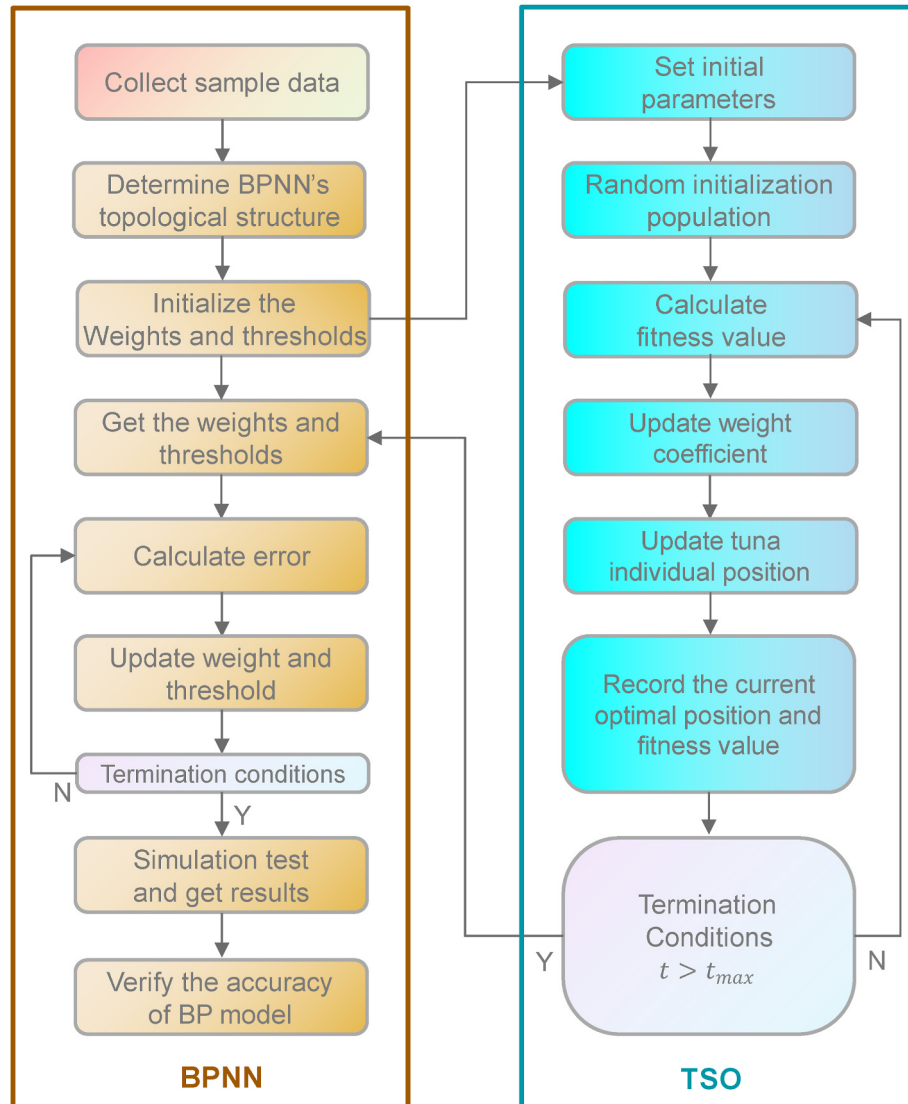


Fig. 3. Flow chart of the TSO-BPNN model.

representing individual tunas. The population size, maximum iteration count, initial weight bounds, target minimum error, optimization parameter bounds are established. If n is the number of data points, y_i is the i^{th} observed or actual value, and $\hat{y}_i$ is the i^{th} predicted value, then the Sum of Absolute Errors (SAE) is selected as the objective function for optimization.

$$SAE = \sum_{i=1}^n |y_i - \hat{y}_i| \quad (24)$$

- 4) Computation of individual fitness values: Identify and document the position corresponding to the optimal fitness value, designating it as the pinnacle individual position.
- 5) Assign the optimized weight coefficients and bias parameters to the BPNN.
- 6) Update singular positions based on the designated position update strategy.
- 7) Implement a termination criterion: Halt operations upon satisfying the maximum iteration count or achieving the desired precision threshold, subsequently assigning the optimal values to the BPNN.
- 8) Subject the optimized BPNN to training and testing phases, followed by an error analysis and accuracy evaluation. Four error metrics, namely Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), are employed as evaluation criteria for the model. Their respective formulas are as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (25)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (26)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (27)$$

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (28)$$

3.3. Model-based reinforcement learning of wind turbine control

3.3.1. Formulation of control objectives

This study addresses the control challenges of wind turbines, encompassing the maximization of power generation (or tracking of the maximum power point) in Region II and power maintenance in Region III. Within the control framework for Region II, we introduce a methodology to adjust the torque, facilitating maximal power extraction from the wind turbine. For the pitch angle control task in Region III, the objective is to sustain both the output power and the turbine rotational speed at their rated specifications. While Ref. [29] has proposed a reward formulation for the control objective, its singular reward structure may not offer robustness against outliers as well as the potential substantial penalties associated with the squaring of large error terms. In this work, we incorporate the Huber loss [41], refining the specific form of the reward function.

Further elaborated, the torque control function for Region II is formulated as follows:

$$L_1(t) = \begin{cases} \frac{1}{2} [T_{g,ref}(t) - T_{g,pre}(t)]^2, & \text{for } |T_{g,ref}(t) - T_{g,pre}(t)| \leq \delta \\ \delta |T_{g,ref}(t) - T_{g,pre}(t)| - \frac{1}{2} \delta^2, & \text{otherwise} \end{cases} \quad (29)$$

$$\Delta T = T_{g,pre}(t+1) - T_{g,pre}(t) \quad (30)$$

$$\min J_{T_g}(t) = \sum_t [L_1(t) + \Delta T^T R \Delta T] \quad (31)$$

Subject to:

$$0 \leq T_{g,ref} \leq T_{g,max} \quad (32)$$

$$-\dot{T}_{max} \leq \Delta T \leq \dot{T}_{max} \quad (33)$$

Where $T_{g,ref}$ represents the torque reference value at time t , $T_{g,pre}$ denotes the torque prediction at time t , δ is a threshold in the Huber loss function, chosen as 0.1 in this context. R is the weight matrix.

For the control objectives in Region III, in addition to fulfilling the pitch angle control targets, it's essential to sustain a stable state proximate to the rated speed. A speed reward-penalty system is incorporated here to realize a more efficient control objective. The specific control function is as follows:

$$L_2(t) = \begin{cases} \frac{1}{2} [\beta_{ref}(t) - \beta_{pre}(t)]^2, & \text{for } |\beta_{ref}(t) - \beta_{pre}(t)| \leq \delta \\ \delta |\beta_{ref}(t) - \beta_{pre}(t)| - \frac{1}{2} \delta^2, & \text{otherwise} \end{cases} \quad (34)$$

$$\Delta \beta = \beta_{pre}(t+1) - \beta_{pre}(t) \quad (35)$$

$$L_3(t) = \frac{1}{N} \sum_{i=1}^N [\omega_{ref}(t) - \omega_{pre}(t)]^2 - \alpha_{reward} |\omega_{ref}(t) - \omega_{pre}(t)| + \alpha_{penalty} |\omega_{ref}(t) - \omega_{pre}(t)| \quad (36)$$

$$\min J_{\beta}(t) = \sum_t [L_2(t) + \Delta \beta^T R \Delta \beta + L_3(t)] \quad (37)$$

Subject to:

$$\beta_{ref}^{min} \leq \beta_{ref} \leq \beta_{ref}^{max} \quad (38)$$

$$-\dot{\beta}_{max} \leq \Delta \beta \leq \dot{\beta}_{max} \quad (39)$$

$$\omega_{ref} \leq 1.1 \omega_g^{rated} \quad (40)$$

Wherein β_{ref} represents the pitch angle reference value at time t , β_{pre} denotes the pitch angle prediction at time t . δ is a threshold in the Huber loss function, chosen here as 0.1, N is the number of data points. ω_{ref} stands for the generator speed reference value at time t , while ω_{pre} represents the predictive generator speed at that moment. α_{reward} and $\alpha_{penalty}$ are the reward and penalty coefficients respectively, set to 0.5 and 1 for this context. R is the weight matrix.

It's worth noting that by tracking the torque, pitch angle, and generator speed to their respective reference values in the two regions, the mechanical efficiency in region II can be maintained at its optimum, while power maintenance can be achieved in region III. These reference values are derived based on wind speed using a lookup table approach as proposed in Ref. [42].

3.3.2. System dynamics approximation

In the realm of Reinforcement Learning (RL) based on Markov Decision Processes (MDP) [43], the primary objective is to optimize the long-term rewards an agent can garner by engaging in interactions within a given environment, thereby facilitating the acquisition of an optimal policy. To elucidate further, at any given temporal instance t , the prevailing state of the system can be denoted as s_t which is a member of the state space S . At this juncture, the agent decides upon an action a_t , which resides in the action space A , leading to the realization of a reward r_t that exists within the reward space R .

Utilizing the transition function defined as $f: S \times A \rightarrow S$, one can anticipate the subsequent state at $t+1$, articulated as $s_{t+1} = f(s_t, a_t)$. The overarching ambition of this RL agent is to master a policy function $\pi(s)$:

$= S \rightarrow A$ that maximizes the cumulative discounted rewards. Mathematically, this policy can be represented by $\sum_{t=t'}^{\infty} \gamma^{t-t'} r_t(s_t, a_t)$, where γ (constrained within the bounds $0 < \gamma < 1$) serves as a discount factor, and $r_t(s_t, a_t)$ quantifies the reward procured post the selection of action a_t from state s_t [18].

Specifically, this work harnesses Deep Neural Networks (DNN) to bridge the prior state prediction TSO-BPNN, forming a cascaded structure, serving as an information processor for approximating system dynamics. This is dynamically updated to bolster the robustness of control. Fundamentally, this approach enhances the conventional reinforcement learning paradigm: the critic architecture for evaluating long-term objective functions. While classical methods employ an actor setup to derive optimal control strategies based on critic outcomes, this study integrates the actor within an MPC framework to discern such optimal strategies. This configuration offers augmented flexibility for online applications and empowers our RL-based wind turbine controller to swiftly adapt to unforeseen model alterations, such as those triggered by abrupt actuator malfunctions [44]. The specifics of this RL methodology are detailed below.

In the designed RL approach, a DNN architecture is utilized to approximate the system dynamics. This approximation of the system dynamics is denoted as $\hat{f}_{\theta}(s_t, a_t)$. For training at a timestep t , the output state $s_t = [\omega_r \ \omega_g \ T_g \ \beta]$ from TSO-BPNN and action a_t are fed as inputs to the DNN, with the subsequent state s_{t+1} being the expected output. The gathered measurement dataset is represented as D , having a size of N_d . As such, D should include a set of states $(s_0, s_1, \dots, s_{N_d-1})$, a set of actions $(a_0, a_1, \dots, a_{N_d-1})$, and a set of successor states $(s_1, s_2, \dots, s_{N_d})$. It's important to note that if the difference between a state a_t and its successor s_{t+1} is almost negligible, then the change between s_t and s_{t+1} (denoted as Δs) can represent the output of $\hat{f}_{\theta}(s_t, a_t)$. In this context, the relationship is defined as $\Delta s_t = s_{t+1} - s_t = \hat{f}_{\theta}(s_t, a_t)$.

3.3.3. Control policy optimization

Using $\hat{f}_{\theta}(s_t, a_t)$, we can recursively predict future states based on candidate actions and select the optimal control strategy $\pi(s) := s_t \rightarrow a_t$ to achieve the highest reward. The Model Predictive Control (MPC) structure is embedded into the RL control method we propose to address the optimal closed-loop control problem described in equations (29)–(40). Compared to other controllers, the MPC method has the fundamental advantage of achieving multi-objective optimal control and handling state and control constraints. For the predicted future states, the objective is to find an optimal control strategy:

$$a^* = (a_t^*, a_{t+1}^*, \dots, a_{t+T_s-1}^*) \quad (41)$$

The specific method for selecting the optimal action is as follows:

$$a^* = \underset{a}{\operatorname{argmax}} \sum_{t=t'}^{t'+T_s-1} \gamma^{t-t'} r_t(\hat{s}_t, a_t) \quad (42)$$

$$\hat{s}_t = s_t, \hat{s}_{t+1} = \hat{f}_{\theta}(\hat{s}_t, a_t) \quad (43)$$

Utilizing the MPC framework, an optimal action sequence is derived, of which the initial action is integrated into the subsequent state evolution of the system. Following the execution at a designated sampling interval T_{int} , the emergent policy-based data, denoted as $(s_t, \dots, s_{t+T_{int}}, a_t^*, \dots, a_{t+T_{int}+1}^*, s_{t+1}, \dots, s_{t+T_{int}+1})$, is incorporated into dataset D . This iterative procedure is recursively undertaken until the entirety of the control operation is concluded. It's imperative to acknowledge that the prediction horizon T_s within each interval must adhere to the condition $T_s \leq T_{int}$.

3.3.4. RL-based wind turbine control

In the context of our reinforcement learning oriented wind turbine control methodology introduced in this study, Equations (31) and (37)

articulate the reward structures for Regions II and III, respectively. This design framework is visually encapsulated in Fig 4.

A salient feature of our approach is the integration of a surrogate model underpinned by DNN techniques, which undergoes continual refinement with incoming real-time data. This architectural choice proves indispensable for intricate wind turbine systems, especially when confronted with inherent uncertainties and malfunction scenarios that impede the derivation of precise models. Our RL-driven wind turbine control procedure is delineated in Algorithm 1, with i_{max} max denoting the overarching simulation iterations. Due to the updatable nature of the data, real-time updated datasets capture potential variations in wind turbine models when faults are present. This design endows RL methods with enhanced robustness, mitigating the adverse effects of potential uncertainties and faults on power production.

ALGORITHM 1. RL-based Wind Turbine Control Algorithm

```

1: Predict Initial States:
   • Predicted States:  $(s_0, s_1, \dots, s_{N_d-1})$ .
   • Actions:  $(a_0, a_1, \dots, a_{N_d-1})$ .
   • Next predicted states:  $(s_1, s_2, \dots, s_{N_d})$ .
2: Pre-process the initial data and add to dataset  $D$ :
   • Normalize and denoise the initial data.
   • Add pre-processed initial states, actions, and corresponding next states to dataset  $D$ .
3: Iterative Training and Optimization:
   For  $i = 0$  to  $i_{max}$  do.
   • Train the wind turbine dynamics model  $\hat{f}_{\theta}(s_t, a_t)$  with the dataset  $D$ .
     For  $t = iT_{int}$  to  $iT_{int} + T_{int}$  do.
     • Learn control policy based on the trained model  $\hat{f}_{\theta}(s_t, a_t)$ .
     • Optimize the objective function as given in Eq. (31) for torque control, or in Eq. (37) for pitch control. Resultantly, obtain a sequence of control actions (either torque or pitch commands), denoted by  $a^* = (a_t^*, a_{t+1}^*, \dots, a_{t+T_s-1}^*)$ .
     • Apply the first control action  $a_t^*$  to the system and observe  $s_{t+1}$ .
     • Store the tuple  $(s_t, a_t^*, s_{t+1})$  in  $D$ .
     end for
   end for

```

In our enhanced approach, we propose a novel integration of Model Predictive Control with data-centric predictive models and deep reinforcement learning. This framework serves as an effective alternative to traditional Actor-Critic Deep Neural Network modules. Initially, we employ the TSO algorithm to enhance BPNN, which is used for precise prediction of future system states. This innovative design not only inherits the advantage of online updating system dynamics from MPC but also further boosts the accuracy and stability of state prediction through the integration of TSO-BPNN. As a result, our controller can adapt more effectively to real-time changes in system dynamics while preserving the structural benefits and computational efficiency of MPC.

In the next stage of our approach, we leverage the predicted future state information, as obtained from the TSO-BPNN, and feed it into a DNN for reinforcement learning. This structural design allows our algorithm to implement flexible and efficient control strategies in complex, nonlinear environments. Specifically, our controller optimizes a carefully designed reward function, making it capable of fine-tuning control actions to meet specific performance criteria, thereby achieving targeted system performance optimization. Furthermore, we incorporate Huber loss into the reward function in order to increase the robustness of our method. This unique addition is crucial for handling outliers and noise in the reward signals, thereby further minimizing the mismatch between the learned dynamics and the real environment.

The hybrid strategy maintains the overall computational complexity within an acceptable range, which is a significant consideration for real-time applications. Moreover, due to its data-driven and adaptive nature, our method exhibits high scalability and can be applied seamlessly to various types of energy systems beyond wind turbines. Overall, this approach enhances learning and data mining efficiency and allows our method to exhibit higher adaptability and robustness under dynamically changing system dynamics, striking a balance between adaptability, accuracy, and computational efficiency.

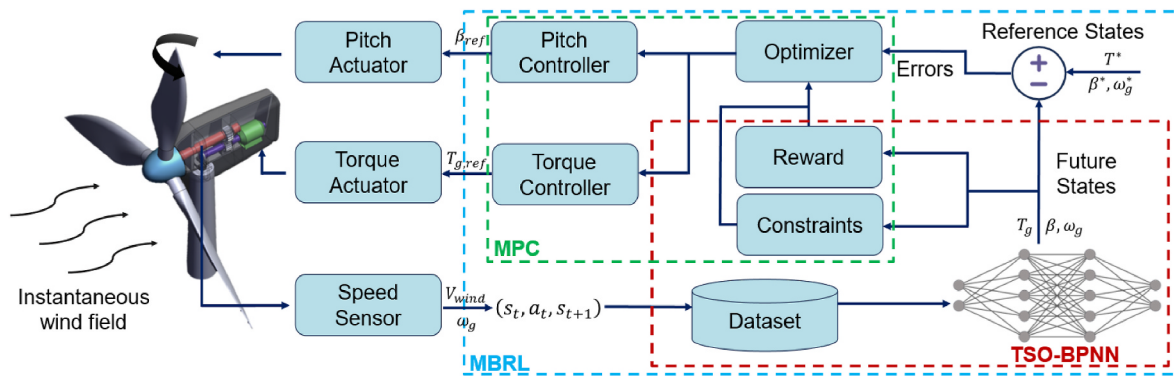


Fig. 4. The framework of the proposed RL-based wind turbine control technique.

4. Result and discussion

4.1. Future state prediction results

4.1.1. Torque prediction

We utilized a total of 16,001 data samples, with 12,800 dedicated to model training, and the remaining 3201 utilized for prediction and evaluation. Fig 5 depicts the comparison between actual torque values from the training set and the predictions made by the TSO-BPNN model. This graphical representation vividly illustrates the remarkable degree of alignment between the predicted values and the ground truth data. This alignment underscores the model's exceptional performance during the critical training phase, affirming its robustness and predictive capabilities.

Fig 6 further contrasts the actual torque values from the test set with predictions from both the TSO-BPNN and traditional BPNN models. This comparison visibly underscores the superiority of the TSO-BPNN in predicting torque values, as its predicted values exhibit a closer resemblance to the actual ones than those of the traditional BPNN.

Fig 7 presents a comparison of errors between the two methodologies. From the error distribution, it is evident that the TSO-BPNN model yields lower errors compared to the traditional BPNN, further affirming its superior performance in torque prediction.

To supplement these visual insights, Table 4 offers a detailed breakdown, comparing error metrics derived from both the TSO-BPNN and traditional BPNN predictions. The evaluation leverages key performance metrics: MAE, MSE, RMSE, and MAPE. This tabulated data

provides a granular perspective on the comparative performance of the two predictive models, shedding light on areas where one might outshine the other.

From the above metrics, it is evident that the TSO-BPNN method outperforms the BPNN method across all the metrics. Whether it's MAE, MSE, RMSE, or MAPE, the TSO-BPNN method exhibits lower error values compared to the BPNN method, indicating a closer approximation to the true torque values. Especially when we look at the MAPE, the error for TSO-BPNN is approximately 7.45 %, while for BPNN it's 14.725 %. This suggests that the TSO-BPNN method's average predictions have a smaller deviation from the true values on average. In conclusion, based on this dataset, the TSO-BPNN method seems superior to the BPNN method.

4.1.2. Pitch angle prediction

In Fig 8, we delve into the alignment of torque predictions made by the TSO-BPNN model with the actual values derived from the training set. This visualization underscores the precision with which the TSO-BPNN model approximates real-world torque measurements during its training phase.

Moving on to Fig 9, a more exhaustive comparative exploration is provided, showcasing the actual pitch angle values from the test set against predictions from both the TSO-BPNN and traditional BPNN models. The juxtaposition clearly underscores the superior predictive prowess of the TSO-BPNN, as evidenced by its predictions adhering more closely to the actual values.

Lastly, Fig 10 offers an analytical deep dive into the error distribution

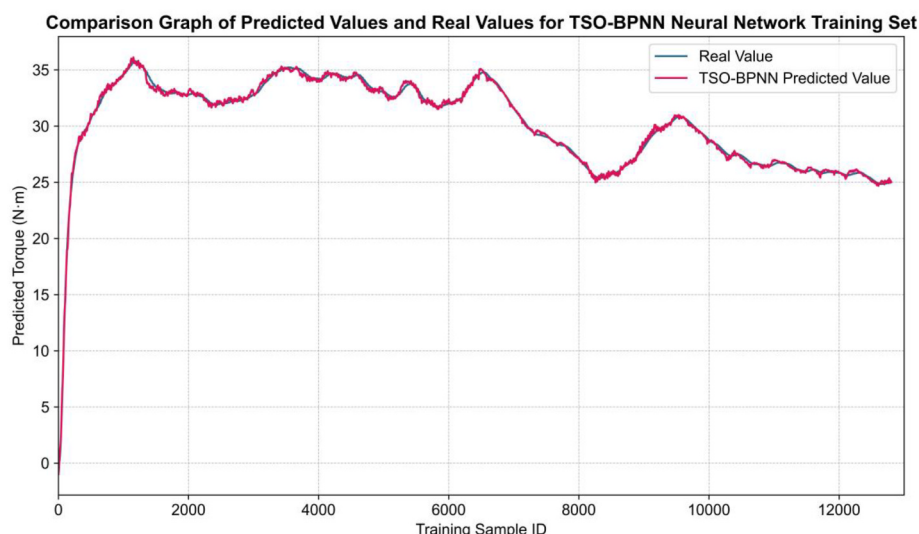


Fig. 5. Comparison of real torque values with TSO-BPNN predictions in the training set.

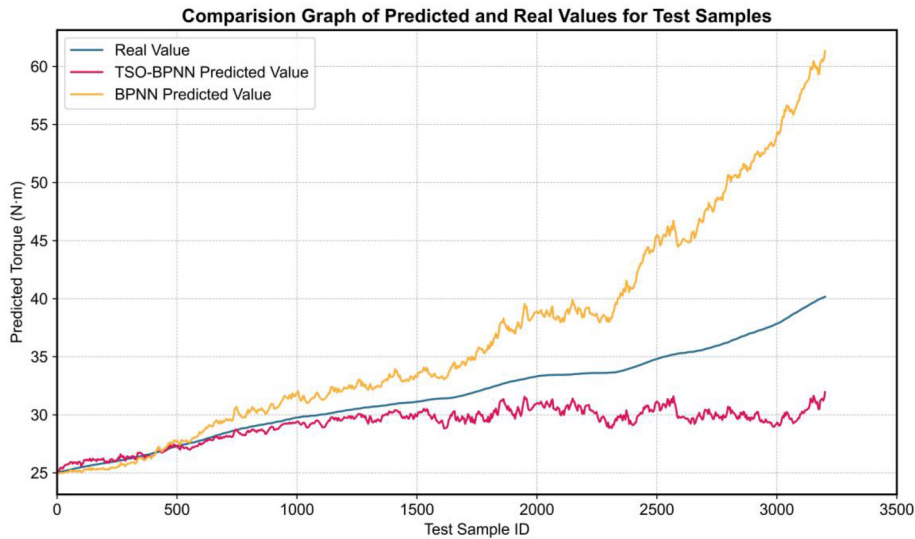


Fig. 6. Comparative analysis of torque predictions: TSO-BPNN vs. BPNN in the test set.

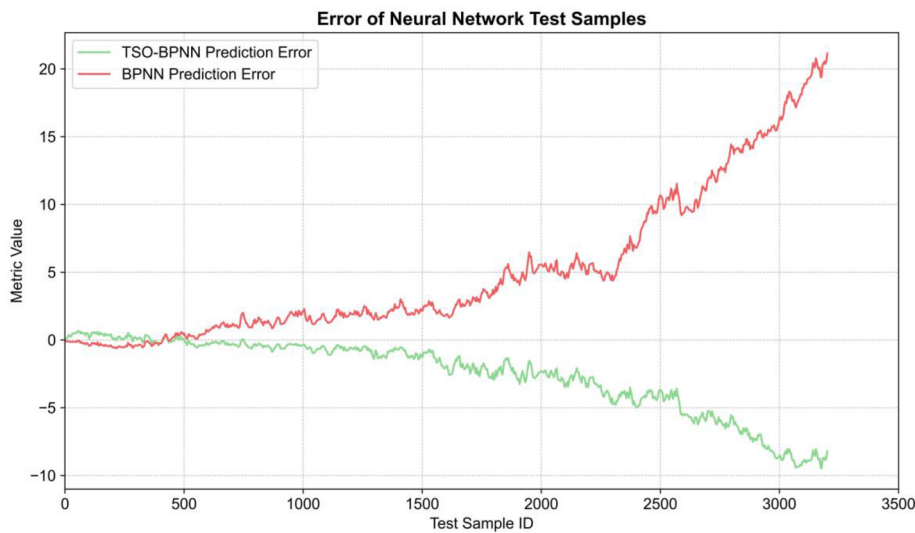


Fig. 7. Error distribution between TSO-BPNN and traditional BPNN predictions.

Table 4

Comparison of error metrics between TSO-BPNN and BPNN predictions.

	MAE	MSE	RMSE	MAPE
TSO-BPNN	2.605	13.867	3.724	7.450 %
BPNN	5.167	56.783	7.535	14.725 %

between the two predictive methodologies. The visual portrayal of errors distinctly indicates that the TSO-BPNN model incurs lower discrepancies in pitch angle predictions compared to its traditional BPNN counterpart, further cementing its superior performance in this domain.

Table 5 presents a detailed assessment of the predictive competencies of both methodologies. By delving into these critical error metrics, the table offers a clear view into the comparative robustness and reliability of the models in their pitch angle forecasts. The presented figures elucidate the precision and reliability of each model, emphasizing the areas where one model might have an edge over the other.

Evaluating the results for the pitch angle predictions, both methods exhibit comparable performances. The TSO-BPNN method marginally surpasses the BP-NN method across all metrics. The differences, while

subtle, are significant enough to indicate the superiority of TSO-BPNN in this scenario. In essence, while both methods offer reliable predictions for the pitch angle, the TSO-BPNN method has a slight edge in terms of accuracy based on the provided dataset.

4.2. Numerical simulations

In this section, the effectiveness of our proposed method is validated through simulations conducted using the FAST (Fatigue, Aerodynamics, Structures, and Turbulence) simulator [45]. FAST is a comprehensive aerodynamic and structural simulator developed by the NREL. Within the context of FAST, we consider the dynamical issues associated with the NREL 5 MW variable-speed, variable-pitch wind turbine. The parameters, states, and input constraints pertaining to this model have been delineated in previous sections. For the purpose of validation, distinct wind profiles with average wind speeds set to 7 m/s and 17 m/s are employed in the simulations, corresponding to the control of the II and III regions of the wind turbine, respectively. These wind profiles are generated using the TurbSim software provided by NREL [46].

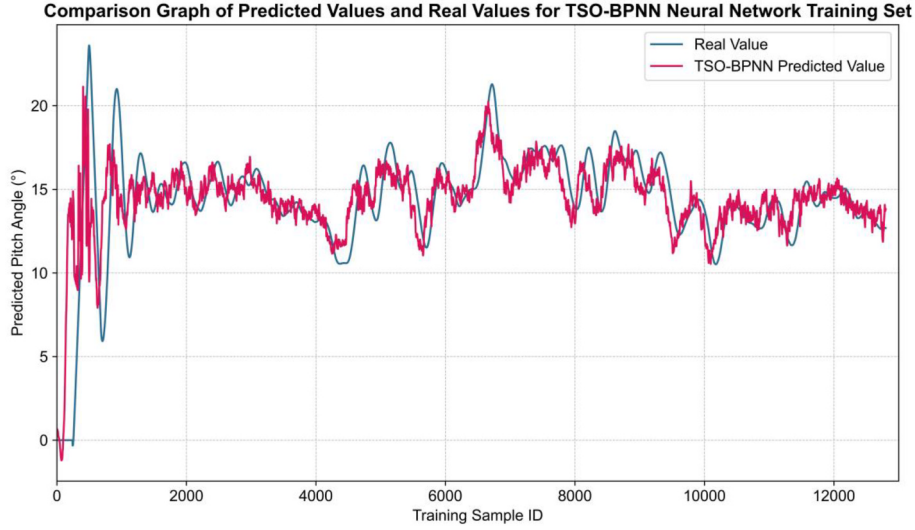


Fig. 8. Comparison of real pitch angle values with TSO-BPNN predictions in the training set.

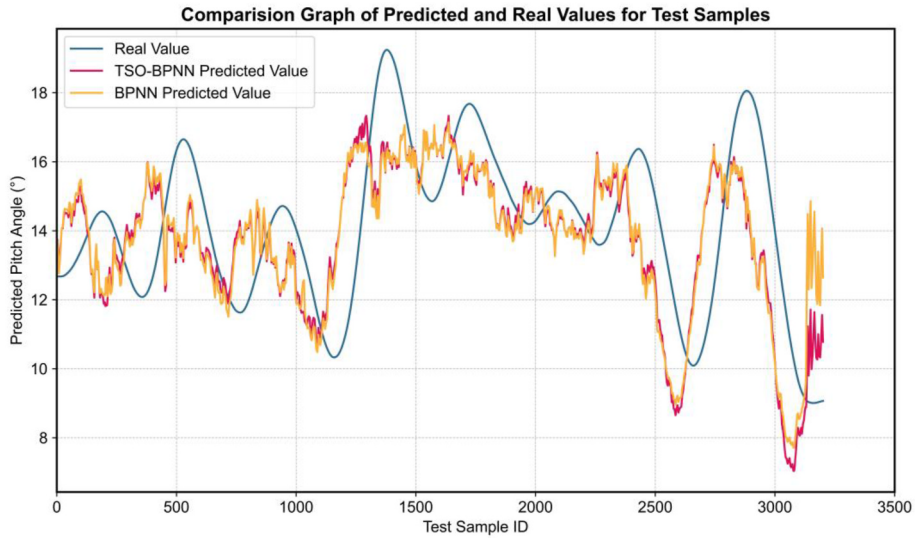


Fig. 9. Comparative analysis of pitch angle predictions: TSO-BPNN vs. BPNN in the test set.

4.2.1. Simulation of various controllers

To conduct a comprehensive performance comparison, this research employs not only the proposed Reinforcement Learning approach, but also incorporates numerical simulations with baseline control and conventional Model Predictive Control strategies.

- (1) Baseline Control: For torque control, the reference generator torque command is derived from the $k\omega^2$ law, where it is proportional to the square of the filtered rotor speed. For pitch control, a gain-scheduled Proportional-Integral (PI) controller is commonly utilized to minimize the discrepancy between the measured rotor or generator speed and their respective setpoint values [47].
- (2) MPC: This strategy leverages the dynamical characteristics of the wind turbine to forecast future states within a finite prediction horizon. Importantly, it should be emphasized that the MPC strategy employed in our case study directly utilizes the FAST model for control. Under normal operating conditions, it is unaffected by any modelling errors or uncertainties, rendering it an excellent and robust benchmark method for comparison with our Reinforcement Learning approach. Based on these anticipated

states, the optimal control sequence is determined by optimizing a specified objective function under various constraints. Each simulation run lasts for 200 s, with a prediction horizon T_s of 0.0125 s and a time interval T_{int} of 2 s.

4.2.2. Control results of wind turbine in region II

For the torque control task in Region II, a wind profile with an average wind speed of 7 m/s is adopted, as depicted in Fig 11. Fig 12 illustrates the torque responses under three distinct control strategies. Within these graphs, the blue solid line represents the desired reference torque trajectory. It is evident from Fig 12 that the torque outputs generated by both our proposed RL-based control approach and the MPC strategy are able to adhere closely and accurately to the desired optimal torque trajectory. Concurrently, Fig 13 presents the power responses under these three different methodologies, where the blue line delineates the prescribed reference power trajectory. Notably, our RL-based approach and MPC method demonstrate significantly superior performance compared to the baseline control scheme, tracking the specified reference power trajectory with remarkable precision.

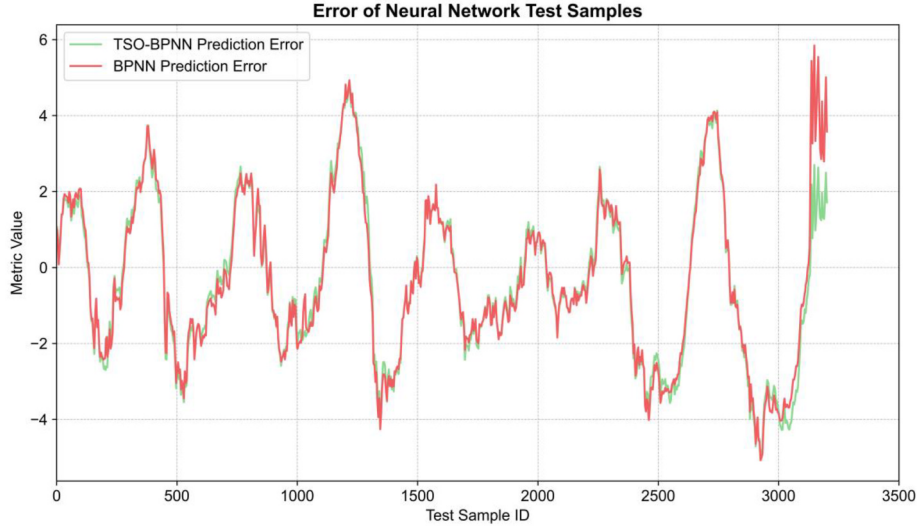


Fig. 10. Error distribution between TSO-BPNN and traditional BPNN predictions.

Table 5

Comparison of error metrics between TSO-BPNN and BPNN predictions.

	MAE	MSE	RMSE	MAPE
TSO-BPNN	1.767	4.435	2.106	12.926 %
BPNN	1.806	4.660	2.159	13.360 %

4.2.3. Control results of wind turbine in region III

For the pitch control in region III, the test scenario is selected under a wind profile with an average speed of 17 m/s, as shown in Fig 14. It is essential to emphasize again that, in this case study, the MPC serves as a benchmark. It directly employs the model embedded in FAST to carry out control, thereby not being influenced by modelling errors and uncertainties under normal working conditions, and thus its performance can be regarded as optimal in this case study.

Fig 15 and Fig 16 illustrate the response characteristics of the generator speed and pitch angle, respectively. From the results, it can be observed that our proposed RL controller exhibits performance closely resembling that of the MPC, indicating that our RL controller is able to successfully approximate this optimal solution. It is noteworthy that, distinct from MPC, our RL method is data-driven and can quickly adapt

to system dynamics changes with the proposed DNN structure.

Lastly, Fig 17 presents the detailed simulation results of power production under different control strategies. Compared to the baseline controller, our RL control algorithm significantly reduces the amplitude of power fluctuations, thereby substantially improving the quality of the power output.

4.3. Performance discussion

In this section, we evaluate and compare the performance of the proposed control strategy for wind turbine power regulation against two other approaches: baseline method and MPC method. To quantitatively assess the performance of each control strategy, several metrics were employed.

Response Time: The time required for the system output to reach 90 % of the final reference value [48].

$$t_{\text{response}} = \min(t | y(t) \geq 0.9y_{\text{ref}}) \quad (44)$$

Overshoot: The percentage by which the system output exceeds the final reference value [49].

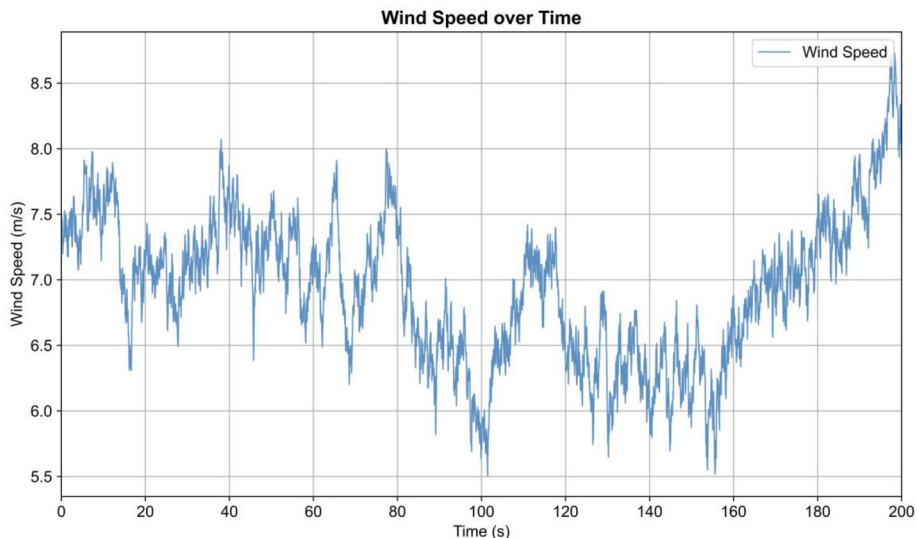


Fig. 11. Wind speed profile with the mean of 7 m/s.

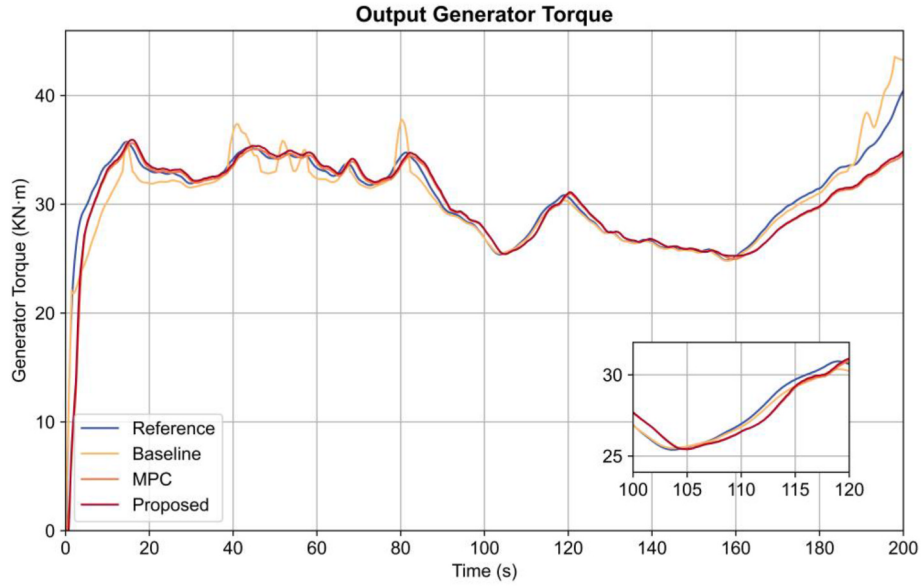


Fig. 12. The variation of generator torque in region II.

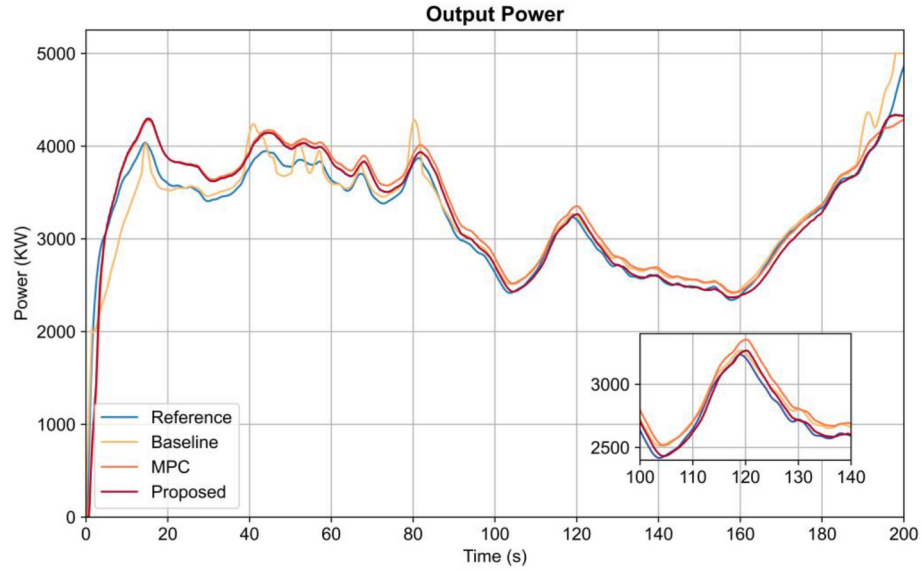


Fig. 13. Simulation results of power production in region II.

$$\text{Overshoot}(\%) = \frac{\max(y(t) - y_{ref})}{y_{ref}} \times 100\% \quad (45)$$

Steady-state Error (SSE): The absolute difference between the final system output and the final reference value [50].

$$e_{ss} = |y(T) - y_{ref}| \quad (46)$$

Variability: The standard deviation of the system output values, used to assess the stability of the system.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2} \quad (47)$$

MSE: The average of the squares of the differences between the system outputs and the corresponding reference values.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - y_{ref,i})^2 \quad (48)$$

Where t is the time, $y(t)$ is the system output at time t , y_{ref} is the reference value. T is the simulation end time. N is the number of data points. y_i is the system output at the i^{th} data point. $\bar{y}$ is the mean of the system output.

In Region II, the control objective is to maximize the power extraction from the wind. From Table 6, it is observed that the proposed method exhibits a similar response time as the MPC method but is slightly slower than the baseline method. The proposed method significantly reduces the overshoot compared to the baseline method. The steady-state error and MSE of the proposed method are lower than those of the MPC method, indicating better tracking performance.

In Region III, the control objective is to maintain the power output constant at its rated value. From the table, it is observed that the proposed method demonstrates a rapid response, closely matching the MPC

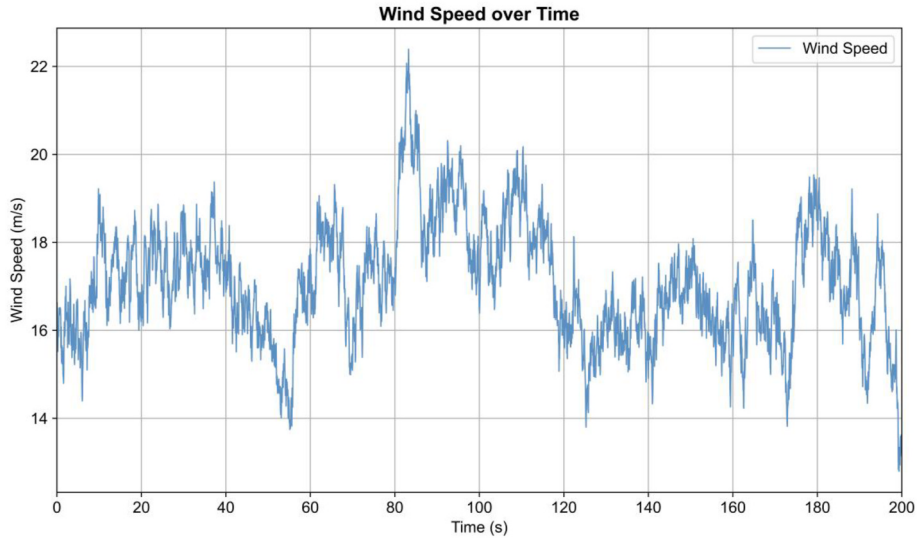


Fig. 14. Wind speed profile with the mean of 17 m/s.

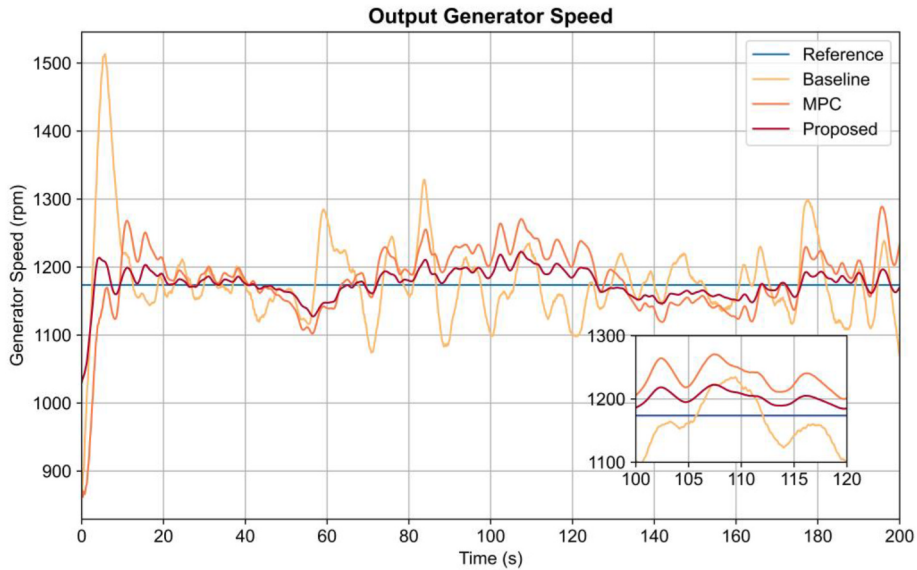


Fig. 15. The variation of generator speed in region III.

method and significantly outperforming the baseline method. The proposed method has the smallest overshoot and steady-state error among the three methods, showing its superior capability in maintaining the power output. Despite a slightly higher variability, the proposed method exhibits a significantly lower MSE compared to the baseline method, indicating its enhanced accuracy in tracking the reference signal.

From the analysis of the data above, it can be concluded that in Region II, although the response time of the proposed model matches that of the MPC method, it reduces the overshoot by 1.07 % and improves the steady-state error by 36.14. For Region III, the proposed approach outperforms both the baseline and MPC across almost all metrics, notably in terms of overshoot and steady-state error. Specifically, it reduces the overshoot by 0.58 % and the steady-state error by 17.27 units relative to the baseline, and by 0.63 % and 9.31 units relative to the MPC, respectively. To obtain a more intuitive comparison, a comprehensive performance assessment is conducted next. Prior to this comparative analysis, we first normalize all the data related to performance indicators. This step is of paramount importance as it ensures that various performance indicators can be compared on a uniform scale.

This, in turn, allows us to evaluate the performance of different control strategies more accurately and fairly. Through normalization by Eq. (49), we eliminate the potential disparities in magnitude and range among different indicators, enabling the colour coding in the heatmap to vividly reflect relative differences between performance metrics.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (49)$$

The heatmap of Fig 18 offers an insightful view, illustrating the performance of various control strategies under different wind speed conditions [51]. In this heatmap, rows represent combinations of different control strategies and wind speed conditions, while columns correspond to distinct performance indicators. The gradient of the colours allows for a quick understanding of which control strategies perform optimally under specific conditions.

The colour intensity illustrates the normalized metric values, spanning from 0 to 1, where a darker hue signifies superior performance. Across nearly all evaluated performance metrics, the Proposed approach consistently manifests a darker tint, suggesting its lower normalized

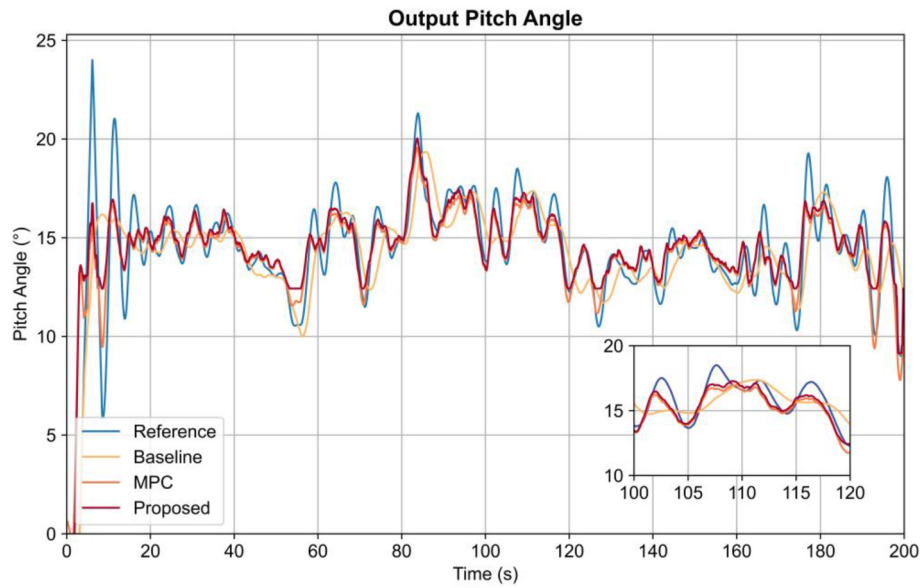


Fig. 16. The variation of pitch angle in region III.

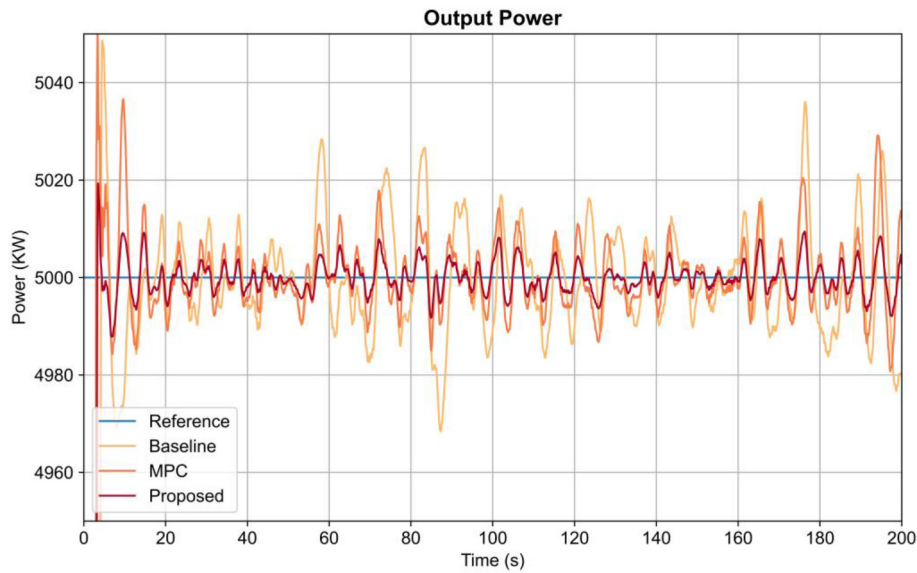


Fig. 17. Simulation results of power production in region III.

Table 6
Performance at region II.

Metric	Baseline	MPC	Proposed
Response Time (s)	194.30	200.00	200.00
Overshoot (%)	2.96	-11.74	-10.67
SSE	141.16	570.82	534.68
Variability	611.11	661.38	677.51
MSE	30601.75	42848.87	31165.24

values and thus superior efficacy. This vividly underscores the effectiveness and superiority of the proposed control strategy for wind turbine power regulation. Especially in Region III, where maintaining a constant power output is crucial, the proposed method shows outstanding performance with minimal steady-state error and overshoot, thereby ensuring stable and efficient operation of the wind turbine under varying wind conditions.

Table 7
Performance at region III.

Metric	Baseline	MPC	Proposed
Response Time (s)	3.60	2.73	2.85
Overshoot (%)	0.97	1.02	0.39
SSE	20.26	12.30	2.99
Variability	479.54	215.47	437.65
MSE	233556.54	46932.31	193892.92

5. Conclusion

This research presents a novel data-driven algorithm that integrates deep neural networks (DNN), Model Predictive Control (MPC), and reinforcement learning (RL) to optimize wind turbine torque and pitch angle control. Enhanced by the Tuna Swarm Optimization-Backpropagation Neural Network (TSO-BPNN), the proposed Data-Centric Predictive Control (DPC) system continuously adapts to real-

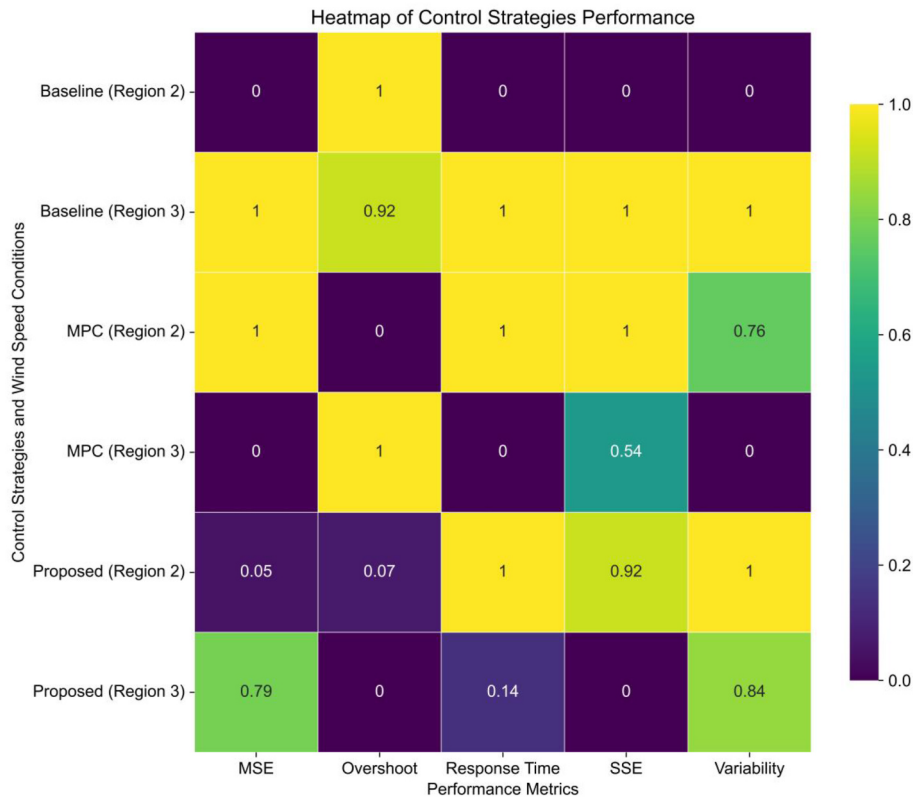


Fig. 18. Performance metrics visualization for different control methods.

time wind fluctuations, providing superior predictive accuracy and improved control performance. Validation using the FAST simulator demonstrated the robustness and adaptability of this approach, with significant performance improvements over traditional control methods.

In Region II, the DPC system matches the response time of the MPC method while reducing overshoot by 1.07 % and improving steady-state accuracy by 36.14 units. In Region III, it surpasses both the baseline and MPC methods, achieving a 0.58 % reduction in error and a 17.27-unit accuracy improvement. These results underscore the potential of the DPC system to enhance wind turbine efficiency under varying conditions, setting a new benchmark in wind energy control.

While the study demonstrates the effectiveness of the proposed system, it focuses primarily on individual turbine control. Future research should investigate collaborative control strategies for optimizing the performance of entire wind farms. Additionally, further studies should assess the long-term economic and environmental impacts of the DPC system to evaluate its broader applicability in the wind energy sector.

CRedit authorship contribution statement

Wei Li: Investigation, Methodology, Software, Validation, Visualization. **Ravi Kumar Pandit:** Funding acquisition, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

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

References

- [1] Y. Guo, H. Wang, J. Lian, Review of integrated installation technologies for offshore wind turbines: current progress and future development trends, *Energy Convers. Manag.* 255 (2022) 115319.
- [2] A. Dalla Riva, J. Hethey, P. Borre Eriksen, S. Luers, F. Mak, S. Kalash, Y. Kikuchi, IEA Wind TCP Task 26: Wind Technology, Cost, and Performance Trends for Denmark, Germany, Ireland, Japan, Norway, Sweden, the European Union, and the United States 2016-2019, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2023 (No. NREL/TP-5000-81246).
- [3] RenewableUK, UK wind energy database (UKWED) [Data set]. RenewableUK, <https://www.renewableuk.com/page/UKWEDhome>, 2023.
- [4] R.K. Pandit, D. Astolfi, I. Durazo Cardenas, A review of predictive techniques used to support decision making for maintenance operations of wind turbines, *Energies* 16 (4) (2023) 1654.
- [5] A.K. Pathak, M.P. Sharma, M. Bunde, A critical review of voltage and reactive power management of wind farms, *Renew. Sustain. Energy Rev.* 51 (2015) 460–471.
- [6] S. Ghosh, S. Kamalasadan, N. Senroy, J. Enslin, Doubly fed induction generator (DFIG)-based wind farm control framework for primary frequency and inertial response application, *IEEE Trans. Power Syst.* 31 (3) (2015) 1861–1871.
- [7] J. Aho, A. Buckspan, J. Laks, P. Fleming, Y. Jeong, F. Dunne, K. Johnson, A tutorial of wind turbine control for supporting grid frequency through active power control, in: 2012 American Control Conference (ACC), IEEE, 2012, June, pp. 3120–3131.
- [8] J.E. Sierra-Garcia, M. Santos, Deep learning and fuzzy logic to implement a hybrid wind turbine pitch control, *Neural Comput. Appl.* (2021) 1–15.
- [9] S.A. Hamoodi, F.I. Hameed, A.N. Hamoodi, Pitch angle control of wind turbine using adaptive fuzzy-PID controller, *EAI Endorsed Trans. Energy Web* 7 (28) (2020) e15 e15.
- [10] Q.V. Ngo, Y. Chai, T.T. Nguyen, The fuzzy-PID based-pitch angle controller for small-scale wind turbine, *Int. J. Power Electron. Drive Syst.* 11 (1) (2020) 135.
- [11] Y. Fan, A. Mu, Adaptive fuzzy Proportion Integration Differentiation control in hydraulic offshore wind turbine for optimal power extraction based on the estimated wind speed, *Energy Sci. Eng.* 8 (5) (2020) 1604–1619.
- [12] J. Yang, X. Sun, K. Liao, Z. He, L. Cai, Model predictive control-based load frequency control for power systems with wind-turbine generators, *IET Renew. Power Gener.* 13 (15) (2019) 2871–2879.
- [13] H. Bassi, Y.A. Mobarak, State-space modeling and performance analysis of variable-speed wind turbine based on a model predictive control approach, *Eng. Technol. Appl. Sci. Res.* 7 (2) (2017) 1436–1443.
- [14] J. Wei, Y. Cao, Q. Wu, C. Li, S. Huang, B. Zhou, D. Xu, Coordinated droop control and adaptive model predictive control for enhancing HVRT and post-event

- recovery of large-scale wind farm, *IEEE Trans. Sustain. Energy* 12 (3) (2021) 1549–1560.
- [15] B. Babaghorbani, M.T. Hamidi Beheshti, H.A. Talebi, An improved model predictive control of low voltage ride through in a permanent magnet synchronous generator in wind turbine systems, *Asian J. Control* 21 (4) (2019) 1991–2003.
 - [16] R. Barcena, T. Acosta, A. Etxebarria, I. Kortabarria, Wind turbine structural load reduction by linear single model predictive control, *IEEE Access* 8 (2020) 98395–98409.
 - [17] Z. Chen, J. Liu, Z. Lin, C. Qu, Variable-constrained model predictive control of coordinated active power distribution for wind-turbine cluster, *Appl. Sci.* 9 (1) (2018) 112.
 - [18] S. Dickler, T. Wintermeyer-Kallen, J. Zierath, R. Bockhahn, D. Machost, T. Konrad, D. Abel, Full-scale field test of a model predictive control system for a 3 MW wind turbine, *Forsch. Im. Ingenieurwes.* 85 (2) (2021) 313–323.
 - [19] W. Fan, Z. Hu, V. Veerasamy, PSO-based model predictive control for load frequency regulation with wind turbines, *Energies* 15 (21) (2022) 8219.
 - [20] P. Jiang, T. Zhang, J. Geng, P. Wang, L. Fu, An MPPT strategy for wind turbines combining feedback linearization and model predictive control, *Energies* 16 (10) (2023) 4244.
 - [21] J.E. Sierra-Garcia, M. Santos, R. Pandit, Wind turbine pitch reinforcement learning control improved by PID regulator and learning observer, *Eng. Appl. Artif. Intell.* 111 (2022) 104769.
 - [22] J. Xie, H. Dong, X. Zhao, Data-driven torque and pitch control of wind turbines via reinforcement learning, *Renew. Energy* 215 (2023) 118893.
 - [23] V.T. Aghaei, A. Agababaoğlu, B. Bawo, P. Naseradinmousavi, S. Yıldırım, S. Yeşilyurt, A. Onat, Energy optimization of wind turbines via a neural control policy based on reinforcement learning Markov chain Monte Carlo algorithm, *Appl. Energy* 341 (2023) 121108.
 - [24] L. Jia, J. Hao, J. Hall, H.K. Nejadkhaki, G. Wang, Y. Yan, M. Sun, A reinforcement learning based blade twist angle distribution searching method for optimizing wind turbine energy power, *Energy* 215 (2021) 119148.
 - [25] N. Tomin, V. Kurbatsky, H. Guliyev, Intelligent control of a wind turbine based on reinforcement learning, in: 2019 16th Conference on Electrical Machines, Drives and Power Systems (ELMA), IEEE, 2019, June, pp. 1–6.
 - [26] A. Saenz-Aguirre, E. Zulueta, U. Fernandez-Gamiz, A. Ulazia, D. Teso-Fz-Betono, Performance enhancement of the artificial neural network-based reinforcement learning for wind turbine yaw control, *Wind Energy* 23 (3) (2020) 676–690.
 - [27] Westerbeck, N., Gonsior, J., Marten, D., & Perez-Becker, S. Reinforcement Learning for Wind Turbine Load Control.
 - [28] Q. Huang, Model-based or model-free, a review of approaches in reinforcement learning, in: 2020 International Conference on Computing and Data Science (CDS), IEEE, 2020, August, pp. 219–221.
 - [29] J. Xie, H. Dong, X. Zhao, Data-driven torque and pitch control of wind turbines via reinforcement learning, *Renew. Energy* 215 (2023), <https://doi.org/10.1016/j.renene.2023.06.014>.
 - [30] D. Song, G. Shen, C. Huang, Q. Huang, J. Yang, M. Dong, Y.H. Joo, N. Duić, Review on the application of artificial intelligence methods in the control and design of offshore wind power systems, *J. Mar. Sci. Eng.* 12 (2024) 424, <https://doi.org/10.3390/jmse12030424>.
 - [31] Abdullah, I. Al-Odienat, Ayman A. Al-Lawama, The advantages of PID fuzzy controllers over the conventional types, *Am. J. Appl. Sci.* 5 (6) (2008) 653–658.
 - [32] C.L. Bottasso, P. Pizzinelli, C.E.D. Riboldi, L. Tasca, LiDAR-enabled model predictive control of wind turbines with real-time capabilities, *Renew. Energy* 71 (2014) 442–452, <https://doi.org/10.1016/j.renene.2014.05.041>.
 - [33] Narayanan V. Lakshmi, P.S. Mayurappriyan, Arun Tom Mathew, M. Saravana Mohan, C. Ganesh, R. Shanmugasundaram, Next-generation cyber-physical microgrid systems, *Next-Gener. Cyber-Phys. Microgrid Syst.* 77 (2024) 89.
 - [34] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Wierstra, Continuous control with deep reinforcement learning, *arXiv preprint arXiv:1509.02971* (2015).
 - [35] Z. Lin, Z. Chen, Q. Wu, S. Yang, H. Meng, Coordinated pitch & torque control of large-scale wind turbine based on Pareto efficiency analysis, *Energy* 147 (2018) 812–825.
 - [36] L. Schena, E. Gillyns, W. Munters, S. Buckingham, M.A. Mendez, Control of a wind-turbine via machine learning techniques, *arXiv preprint arXiv:2207.06206* (2022).
 - [37] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, *Nature* 323 (6088) (1986) 533–536.
 - [38] H.N. Robert, Theory of the backpropagation neural network, *Proc. 1989 IEEE IJCNN* 1 (1989) 593–605.
 - [39] Y. Liu, Q. Zhao, W. Yao, X. Ma, Y. Yao, L. Liu, Short-term rainfall forecast model based on the improved BP-NN algorithm, *Sci. Rep.* 9 (1) (2019) 19751.
 - [40] L. Xie, T. Han, H. Zhou, Z.R. Zhang, B. Han, A. Tang, Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization, *Comput. Intell. Neurosci.* 2021 (2021) 1–22.
 - [41] G.P. Meyer, An alternative probabilistic interpretation of the huber loss, in: Proceedings of the IEEE/cvf Conference on Computer Vision and Pattern Recognition, 2021, pp. 5261–5269.
 - [42] J.E. Sierra-García, M. Santos, Lookup table and neural network hybrid strategy for wind turbine pitch control, *Sustainability* 13 (6) (2021) 3235.
 - [43] C. Wei, Z. Zhang, W. Qiao, L. Qu, Reinforcement-learning-based intelligent maximum power point tracking control for wind energy conversion systems, *IEEE Trans. Ind. Electron.* 62 (10) (2015) 6360–6370.
 - [44] H. Dong, X. Zhao, Composite experience replay-based deep reinforcement learning with application in wind farm control, *IEEE Trans. Control Syst. Technol.* 30 (3) (2021) 1281–1295.
 - [45] J.M. Jonkman, M.L. Buhl, FAST User's Guide, vol. 365, National Renewable Energy Laboratory, Golden, CO, USA, 2005, p. 620.
 - [46] B.J. Jonkman, TurbSim User's Guide: Version 1.50, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2009 (No. NREL/TP-500-46198).
 - [47] J. Jonkman, S. Butterfield, W. Musial, G. Scott, Definition of a 5-MW Reference Wind Turbine for Offshore System Development, National Renewable Energy Lab. (NREL), Golden, CO (United States), 2009 (No. NREL/TP-500-38060).
 - [48] S. Skogestad, I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, John Wiley & sons, 2005.
 - [49] B.C. Kuo, *Automatic Control Systems*, Prentice Hall PTR, 1987.
 - [50] G.F. Franklin, J.D. Powell, A. Emami-Naeini, J.D. Powell, *Feedback Control of Dynamic Systems*, vol. 4, Prentice hall, Upper Saddle River, 2002.
 - [51] T. Metsalu, J. Vilo, ClustVis: a web tool for visualizing clustering of multivariate data using Principal Component Analysis and heatmap, *Nucleic Acids Res.* 43 (W1) (2015) W566–W570.



DSPACE

<https://dspace.org/>

Data-centric predictive control with tuna swarm optimization-backpropagation neural networks for enhanced wind turbine performance

Li, Wei; Pandit, Ravi

2024-12

Elsevier

<https://doi.org/10.1016/j.renene.2024.121821>

Li W, Pandit RK. (2024) Data-centric predictive control with tuna swarm optimization-backpropagation neural networks for enhanced wind turbine performance. Renewable Energy, Volume 237, Part C, December 2024, Article number 121821