energies

MDPI

Article

Anticipatory Pitch Control for Small Wind Turbines Using Short-Term Rotor-Speed Prediction with Machine Learning

Ernesto Chavero-Navarrete *, Juan Carlos Jáuregui-Correa *, Mario Trejo-Perea, José Gabriel Ríos-Moreno and Roberto Valentín Carrillo-Serrano

Facultad de Ingeniería, Universidad Autónoma de Querétaro, Santiago de Querétaro 76010, Mexico; mtp@uaq.mx (M.T.-P.); riosg@uaq.mx (J.G.R.-M.); roberto.carrillo@uaq.mx (R.V.C.-S.)
* Correspondence: echn_7@hotmail.com (E.C.-N.); jc.jauregui@uaq.mx (J.C.J.-C.); Tel.: +55-442-436-01-07 (E.C.-N.)

Abstract

Small wind turbines operating at low heights frequently experience rapidly fluctuating and highly turbulent wind conditions that challenge conventional reactive pitch-control strategies. Under these non-stationary regimes, sudden gusts produce overspeed events that increase mechanical stress, reduce energy capture, and compromise operational safety. Addressing this limitation requires a control scheme capable of anticipating aerodynamic disturbances rather than responding after they occur. This work proposes a hybrid anticipatory pitch-control approach that integrates a conventional PI regulator with a data-driven rotor-speed prediction model. The main novelty is that short-term rotor-speed forecasting is embedded into a standard PI loop to provide anticipatory action without requiring additional sensing infrastructure or changing the baseline control structure. Using six years of real wind and turbine-operation data, an optimized Random Forest model is trained to forecast rotor speed 20 s ahead based on a 60 s historical window, achieving a prediction accuracy of RMSE = 0.34 rpm and R^2 = 0.73 on unseen test data. The predicted uses a sliding-window representation of recent wind–rotor dynamics to estimate the rotor speed at a fixed horizon (t + Δt), and the predicted signal is used as the feedback variable in the PI loop. The method is validated through a high-fidelity MATLAB/Simulink model of 14 kW small horizontal-axis wind turbine, evaluated under four wind scenarios, including two previously unseen conditions characterized by steep gust gradients and quasi-stationary high winds. The simulation results show a reduction in overspeed peaks by up to 35–45%, a decrease in the integral absolute error (IAE) of rotor speed by approximately 30%, and a reduction in pitch-actuator RMS activity of about 25% compared with the conventional PI controller. These findings demonstrate that short-term AI-based rotor-speed prediction can significantly enhance safety, dynamic stability, and control performance in small wind turbines exposed to highly variable atmospheric conditions.

Keywords: anticipatory pitch control; short-term wind prediction; rotor-speed forecasting; ML regression models; turbulent wind dynamics

Check for updates

Academic Editor: Tonio Sant

Received: 28 November 2025
Revised: 27 December 2025
Accepted: 3 January 2026
Published: 4 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Despite the remarkable growth and technological maturity achieved by large-scale wind energy, the development of solutions specifically tailored for small wind turbines and sites with limited wind resources has remained relatively lagging. This gap is largely driven by the industry's predominant focus on large-scale systems, where increased rotor

diameters maximize annual energy production and improve financial returns. Nevertheless, small-scale wind turbines remain essential for distributed generation and remote ap-plications, particularly in regions were expanding or reinforcing long-distance transmission infrastructure is economically prohibitive or logistically unfeasible [1,2].

In this context, a small variable-speed wind turbine is considered, consisting of a rotor with n blades and an adjustable pitch angle, a mechanical drivetrain, an electric generator, and the associated power electronics and control systems. This architecture enables adaptation of the rotor speed to changing wind conditions, improving energy efficiency while reducing mechanical loads under non-stationary operation.

Pitch control plays a central role in such systems by dynamically regulating the blade pitch angle to control rotor speed, generated power, and overall turbine safety. However, at sites typically targeted by small wind turbines, the atmospheric flow is strongly influenced by complex orography, heterogeneous surface roughness, and nearby obstacles. As a result, the wind field exhibits intermittent gusts, high turbulence intensity (TI), and variable shear, conditions characteristic of low-height, small- and medium-scale turbines [2,3]. Under these circumstances, the quasi-stationary wind assumption no longer holds, and the incoming flow becomes inherently nonlinear, non-Gaussian, and highly transient [4].

These characteristics limit the performance of linear and reactive control schemes, such as conventional Proportional–Integral–Derivative (PID) controllers, which are typically tuned for steady or slowly varying wind conditions. In particular, PID-based pitch control acts only after gusts impact the rotor, which may lead to actuator saturation, delayed corrective actions, and overspeed events, thereby increasing mechanical stress, reducing energy capture, and compromising system safety [5–7]. Consequently, anticipatory and nonlinear control approaches capable of predicting the arrival and magnitude of gusts before they affect the rotor are required. This need has motivated the development of advanced control and optimization strategies [7,8].

In the domain of mathematical and robust control methods, several approaches combine pitch control with Maximum Power Point Tracking (MPPT) algorithms, notably using Fractional Proportional–Integral (FPI) controllers, which improve settling times and angle selection accuracy compared with conventional PI controllers [9]. In parallel, Dynamic Sliding Mode Control (D-SMC) has been applied, integrating Sliding Mode Observers (SMO) to estimate additional states and generate smooth and robust control signals under uncertainty [10]. Other metaheuristic approaches, such as Fractional Particle Swarm Optimization (FPSO) combined with Taylor series–based linearization, have demonstrated improvements in rotor-speed stability and pitch signal smoothness compared with standard Particle Swarm Optimization (PSO) and Takagi–Sugeno–Kang (TSK) fuzzy control [11]. More recently, Uncertainty and Disturbance Estimator (UDE) techniques have been introduced, incorporating elliptical mathematical constraints to avoid integrator windup and guarantee safe operational domains; this strategy employs filtered error tracking to transform the non-affine model into an affine one [12]. Although these methods represent significant advances over classical PID control, their performance relies heavily on model accuracy, proper parameter tuning, and the adequate representation of wind conditions. In parallel, purely aerodynamic approaches based on blade-geometry optimization and flow manipulation have also been investigated to improve turbine performance through CFD-based analysis, addressing the problem from a fundamentally different perspective [13].

Regarding fuzzy logic–based strategies, which do not require exact mathematical models, several studies have proposed hybrid schemes that combine fuzzy controllers with PID loops optimized through PSO and Genetic Algorithms (GA). These approaches have demonstrated substantial reductions in accumulated error and improved output-power stability compared with conventional fuzzy controllers [14]. Other works have proposed

multi-model structures that linearize turbine dynamics at multiple operating points and design $H\infty$ controllers for each subsystem, switching between them via triangular membership functions [15]. Additional physics-informed fuzzy learning approaches use GA to optimize rules and membership functions, training the system on simulated wind profiles to reduce overshoot and oscillations in power generation [16]. Although these techniques enhance regulation under uncertainty, most validations remain simulation-based and do not incorporate actuator dynamics or real turbulent wind conditions.

Within the field of Artificial Intelligence (AI)-based techniques, Recurrent Neural Networks (RNNs) have been employed to capture nonlinear temporal dynamics, using wind speed, Tip Speed Ratio (TSR), and power coefficient as inputs to predict the required pitch angle to maintain nominal power under variable wind conditions [17]. Likewise, Artificial Neural Networks (ANNs) with architectures such as Levenberg–Marquardt, resilient backpropagation, and Adaptive Network-based Fuzzy Inference Systems (ANFIS, Takagi–Sugeno type) have shown improved damping and reduced overshoot compared with PID controllers [18]. Another approach trains an ANN offline using simulated wind profiles and log-sigmoid, tan-sigmoid, and linear functions to learn the relationship between generator speed and the optimal pitch angle; the controller adjusts the angle based on both the magnitude and the rate of speed deviation, resulting in more anticipatory and accurate responses [19]. A hybrid scheme integrating deep Long Short-Term Memory (LSTM) networks for wind-speed estimation and prediction with TSK fuzzy control has been proposed [20]. Although these techniques show high potential for handling nonlinearities and complex dynamics, all reviewed studies rely on offline training and simulation-based validation, with no experimental testing or evaluation under real turbulent wind conditions.

Finally, anticipatory strategies based on upstream Light Detection and Ranging (LiDAR) sensors and predictive control have gained relevance for acting before disturbances impact the turbine. LiDAR-based measurements have been integrated into Model Predictive Control (MPC) with feedforward (FF) loops, improving torque and pitch regulation [21]. Fu et al. [22] employ a spinner-mounted single-beam LiDAR to calculate an optimal preview time and design a feedforward controller that significantly reduces rotor-speed errors and structural loads. In [23], this concept is extended to Individual Pitch Control (IPC), where optimal filters and gains are designed in the frequency domain to reduce blade-root loads and tower oscillations. However, these methods rely on accurate, low-latency LiDAR measurements and still lack experimental validation.

Short-term wind speed prediction offers a complementary approach to anticipate flow variations before they reach the turbine. By forecasting rather than measuring in advance, these models enable proactive control actions that enhance stability under turbulent, rapidly changing winds. Recent research has focused on hybrid architectures combining deep learning, signal decomposition, and metaheuristic algorithms capable of capturing the nonlinear and highly variable nature of wind over horizons from seconds to one hour.

Early approaches, such as the hybrid WT–NNAR–LSTM–GBM model [24], integrated the Wavelet Transform (WT) for multiscale decomposition, Sample Entropy (SampEn) to evaluate the complexity of subseries, and two complementary prediction schemes: Neural Network Autoregression (NNAR) for deterministic components and Long Short-Term Memory (LSTM) for highly variable components. The results were fused through a Gradient Boosting Machine (GBM), achieving high accuracy for horizons ranging from minutes to one hour and strong responsiveness to abrupt fluctuations.

Subsequently, predictive control–oriented models, such as the LSTM–MPC scheme [25], combined data-driven forecasting with real-time regulation, achieving notable reductions in rotational fluctuation within 30 s horizons. In [26], uncertainty management was incorporated through the TCN–MCM–EKF model, which combines Temporal Convolutional

Networks (TCN) to capture temporal dependencies, Monte Carlo Methods (MCM) to introduce stochastic uncertainty, and the Extended Kalman Filter (EKF) for adaptive correction, achieving high real-time probabilistic accuracy ($R^2 > 0.98$).

Deep architectures with adaptive mechanisms represented another advance, such as the ACNN–BiGRU model [27], which integrates Convolutional Neural Networks (CNN) for spatiotemporal feature extraction, Bidirectional Gated Recurrent Units (BiGRU) for bidirectional learning, and an attention layer to weight temporal relevance. Metaheuristic optimization–based methods, such as EEMD–WOA–BiGRU [28], combined Enhanced Empirical Mode Decomposition (EEMD) and the Whale Optimization Algorithm (WOA) for hyperparameter tuning, achieving greater robustness and stability against wind nonstationarity.

Despite significant advances in predictive, fuzzy, and model-based control approaches, a consistent gap remains in integrating short-term, data-driven wind prediction directly into the pitch-regulation loop of small wind turbines operating under highly turbulent, non-stationary conditions. Existing anticipatory strategies either rely on LiDAR systems, whose cost, latency, and deployment constraints limit their applicability, or employ neural networks trained offline with synthetic or idealized wind profiles, without evaluation under real atmospheric turbulence. Moreover, most studies assume precise aerodynamic models, fast pitch actuators, or quasi-stationary wind regimes, assumptions that rarely hold for low-height turbines exposed to rapid gusts, steep gradients, and mechanical delays in the pitch system.

In response to these gaps, this work proposes a hybrid anticipatory control strategy that combines a conventional PI pitch controller with a data-driven rotor-speed forecasting model trained on six years of real wind and turbine-operation data. The predictive module delivers a 20 s ahead estimate of the rotor speed, enabling the controller to activate pitch adjustments before gusts affect the rotor. Unlike end-to-end neural schemes or LiDAR-based preview control, the proposed method requires no additional sensing hardware, no complex aerodynamic reconstructions, and can be seamlessly integrated into existing pitch-control architectures. The objective is to demonstrate that short-term AI-based rotor-speed prediction can enhance transient stability, reduce overspeed events, and decrease pitch-actuator effort in small wind turbines operating under rapidly changing wind conditions. The methodology includes model training and optimization, integration into a PI-based control loop, and evaluation through realistic simulation scenarios using both seen and unseen wind data.

2. Materials and Methods

This section presents the empirical and methodological procedures supporting the proposed anticipatory control strategy, covering data acquisition, signal conditioning, model development, and their integration into the turbine's control architecture. The description is structured to ensure transparency, reproducibility, and a rigorous assessment of the validity of the results.

2.1. Test Wind Turbine (Object of Study)

The experimental platform corresponds to a three-bladed horizontal-axis wind turbine (HAWT) with a rated power of 14 kW, installed at the Airport Campus of the Autonomous University of Querétaro (UAQ), Mexico, at an altitude of 1969 m above sea level. The rotor has a swept area of 134 m^2 and is mounted on an 18 m tower, measured from the ground to the rotational axis. Each blade is made of fiberglass and polyester resin, with a total length of 6.4 m, a maximum chord of 1.2 m, and an approximate weight of 260 kg. The aerodynamic profile corresponds to NACA 6812, selected for its favorable lift–drag characteristics and

stable behavior under the moderate Reynolds numbers typical of small- and medium-scale wind turbines. This airfoil has been widely adopted in experimental platforms where reliable aerodynamic response is required across a broad range of operating conditions. A general view of the experimental setup is shown in Figure 1, which presents the 14-kW wind turbine installed at UAQ and used as the primary test bench for the present research.



Figure 1. Experimental 14 kW three-bladed horizontal-axis wind turbine installed at the Airport Campus of the Autonomous University of Querétaro (UAQ), Mexico.

The drive train consists of a mechanical gearbox with a 1:2.1 gear ratio, coupling the low-speed rotor shaft to the high-speed generator shaft. The collective variable-pitch mechanism is located in the rotor hub and is actuated by a 0.5 HP direct-current (DC) motor operating at 1750 rpm, connected to a 60:1 reduction gearbox. The pitch adjustment motion is transmitted to each blade through a conical gear system. This design increases the available torque to counteract aerodynamic loads on the blades while limiting the adjustment speed to approximately 3° per second.

The operating range of the pitch angle extends from 0° to 90°, where $\beta = 0°$ corresponds to the maximum wind-power capture, and $\beta = 90°$ represents the full-feathered position, in which the wind no longer impacts the blades, effectively stopping the rotor.

The nacelle houses a permanent magnet synchronous generator (PMSG) that converts mechanical energy into electrical power. The technical specifications of the generator are summarized in Table 1.

Table 1. Technical specifications of the 14-kW permanent magnet synchronous generator (PMSG) installed in the UAQ wind turbine.

Parameter	Value	Parameter	Value
Rated power	14 kW	Rated speed	135 rpm
Apparent power	14.3 kVA	Insulation class	H
Number of phases	3	Duty type	Continuous
Rated frequency	20.3 Hz	Max. temperature	80 °C
Rated voltage	385 V (AC)	Protection (IP)	IP32
Rated current	22.2 A	Frame type	TC5V
Power factor	0.945	Weight	650 kg

2.2. Meteorological Station and Data Acquisition

A two-stage approach was adopted to characterize the wind resource and configure the new data acquisition system.

Stage 1: The historical wind records for the period 2015–2025 were analyzed using a Campbell Scientific® meteorological station located at 10 m height (station #76628 of the SMN–CONAGUA network). The analysis reported a mean wind speed of 4.45 m/s and a predominant east–west direction (70–115°, occurring 72% of the time). These results established the reference conditions and guided the selection of the upwind position for the new predictive meteorological tower. Table 2 summarizes the statistical analysis of wind speed for the area surrounding the test site.

Table 2. Statistical analysis of wind speed measured at 10 m height (2015–2025).

Statistical Parameter	Value	Statistical Parameter	Value
Mean	4.45 m/s	Range	15
Median	4.24 m/s	Standard Error	0.01
Mode	3.62 m/s	Standard Deviation	1.98 m/s
Minimum	0.00 m/s	Variance	2.72 m^2/s^2
Maximum	15.0 m/s	Coefficient of Variation	44.5%

To estimate the wind speed at the nominal operating height of the wind turbine (18 m), the vertical wind profile was extrapolated using the power-law expression:

$$v_2 = v_1 \left(\frac{z_2}{z_1} \right)^{\alpha} \quad (1)$$

where v_1 and v_2 are the wind speeds corresponding to the heights z_1 and z_2, respectively, with z_2 being the height of interest.

An exponent of $\alpha = 0.20$ was adopted, which is representative of peri-urban environments with moderate surface roughness, where buildings, vegetation, and irregular terrain coexist. This yielded an extrapolated mean wind speed of approximately 5 m/s at rotor height. Figure 2 illustrates representative vertical wind-speed profiles obtained from the mean and standard deviation of the measured wind speed at 10 m, providing a physically consistent visualization of the wind-shear assumption used to extrapolate wind speed to the rotor height.

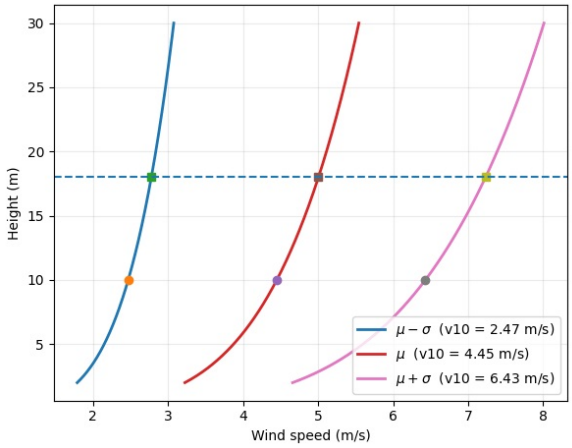


Figure 2. Representative vertical wind-speed profiles derived from the power-law expression using $\alpha = 0.2$. The profiles correspond to the mean wind speed measured at 10 m height ($\mu = 4.45$ m/s) and

one standard deviation around the mean ($\mu \pm \sigma$), which contain the majority of the observed data. Markers indicate the measured reference at 10 m and the extrapolated wind speed at the rotor height (18 m).

Stage 2: A dedicated meteorological station was designed and installed on a telescopic tower located upstream of the test wind turbine. Based on the maximum recorded wind speed (15 m/s) and the measured pitch actuator response time (0–45° in 20 s, or 2.25°/s), the required spacing between the meteorological tower and the turbine was estimated as follows:

$$d_{MT-WT} = v_{max} t_{pitch} \quad (2)$$

This distance represents the minimum upstream separation needed for the predictor to anticipate gusts before they reach the rotor. The resulting value was 300 m; however, due to on-site constraints, the tower was installed at 320 m. Figure 3 shows the location of the meteorological tower in relation to the wind turbine.

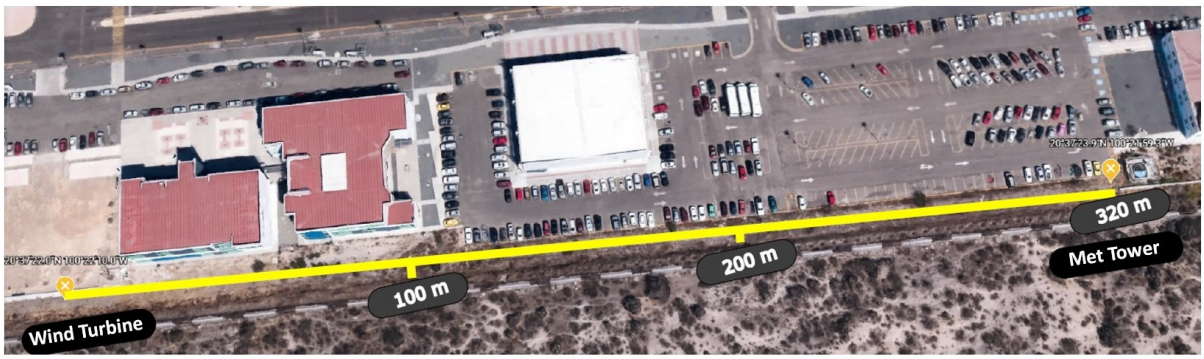
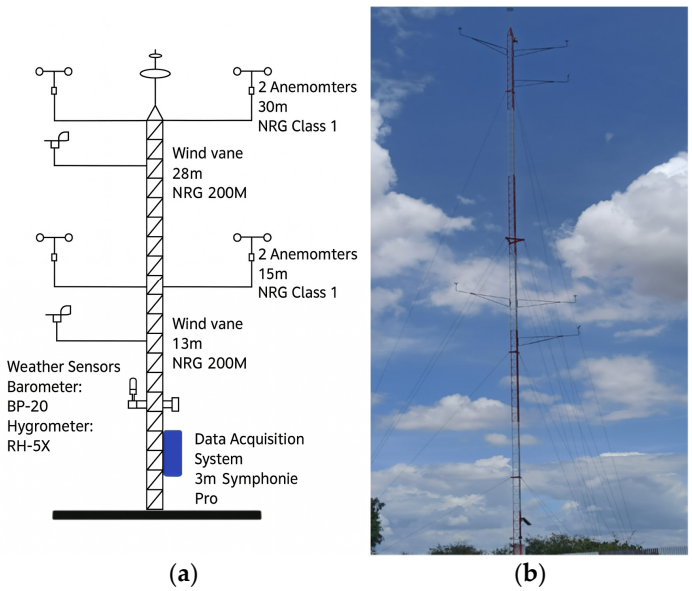


Figure 3. Aerial view of the test site showing the locations of the wind turbine and the met tower, with the 320 m separation distance indicated.

The new station integrates NRG Systems® sensors for wind speed and direction, pressure, temperature, and relative humidity, connected to a Symphonie PRO data logger (16-bit ADC, 1 Hz sampling rate). Anemometers were installed at 15 m and 30 m heights, oriented north–south according to prevailing wind conditions. Figure 4 shows the instrumentation layout and a photograph of the tower installed on site.



(**a**) (**b**)

Figure 4. (**a**) Schematic layout of the met tower and its relative position to the test turbine, including NRG Class 1 anemometers, NRG M wind vanes, BP-20 barometer, RH-5X hygrometer, and a Symphonie

Pro data acquisition system (NRG Systems, Hinesburg, VT, USA). (**b**) Photograph of the installed meteorological tower at the Aeropuerto Campus test site.

2.3. Mathematical Model of the Wind Turbine

The mathematical model developed in this work represents the dynamic behavior of the 14-kW wind turbine described in Section 2.1, integrating its aerodynamic, mechanical, and electrical subsystems into a unified simulation framework. The formulation captures the main dynamic interactions, rotor aerodynamics, drivetrain coupling through the gearbox, pitch actuation dynamics, and electromagnetic conversion in the permanent magnet synchronous generator. This integrated representation enables the analysis of transient responses, power conversion, and control interactions under variable wind conditions, and forms the basis for evaluating the proposed intelligent control strategies.

2.3.1. Assumptions and Definition of Variables

The wind turbine is assumed to be aligned with the wind direction and equipped with three blades of radius R and pitch angle β. The incident wind velocity, denoted as $v(t)$, may include turbulent components and gust events. The main variables considered in the model are defined as follows:

ρ: air density (kg/m^3).
R: rotor radius (m).
$A = \pi R^2$: rotor swept area (m^2).
$v(t)$: instantaneous wind velocity (m/s).
$\beta(t)$: blade pitch angle (rad or °).
ω_r: rotor angular speed (rad/s).
ω_g: generator angular speed (rad/s).
n: gearbox transmission ratio.
J_r, J_g: moments of inertia of the rotor and the generator (kg·m^2).
K_s, D_s: shaft stiffness (N·m/rad) and damping coefficient (N·m·s/rad).
R_s: stator resistance (Ω).
L_d, L_q: direct- and quadrature-axis synchronous inductances (H).
p: number of pole pairs of the generator.
ψ_f: magnetic flux linkage of the rotor (Wb).

The tip speed ratio (TSR), which represents the relationship between the blade tip velocity and the wind velocity, is defined as follows [14]:

$$\lambda = \frac{\omega_r R}{v(t)} \tag{3}$$

2.3.2. Aerodynamic Model

The aerodynamic power extracted from the wind is given by [29]:

$$P_a = \frac{1}{2}\rho A v(t)^3 C_p(\lambda,\ \beta) \tag{4}$$

where C_P (λ, β) is the power coefficient, which represents the aerodynamic efficiency as a function of the rotor geometry and the blade pitch angle.

For the turbine under study, an adjusted surface model of the power coefficient is proposed as follows:

$$C_p(\lambda,\ \beta) = \sum_{i=1}^{N_\lambda}\sum_{j=1}^{N_\beta} \omega_{ij} \varnothing_i(\lambda)\psi_j(\beta) \tag{5}$$

where the weighting coefficients w_{ij} are estimated from experimental data by minimizing the mean square error between the measured and estimated power. The identification process is performed under physical constraints $0 \leq C_P \leq C_{P,max}$ and smoothness conditions with respect to the pitch angle β. This approach follows the regularized surface-fitting methodology proposed in previous studies [5].

The aerodynamic torque on the low-speed shaft is computed as follows [29]:

$$T_a = \frac{P_a}{w_r + \epsilon} \tag{6}$$

where $\varepsilon > 0$ is a small regularization term introduced to prevent singularities during start-up conditions [13].

2.3.3. Mechanical Model

The rotor–generator transmission is modeled as a two-mass system coupled through a flexible shaft. The equations of motion are expressed as follows [29]:

$$\dot{\theta}_s = w_r - \frac{w_g}{n} \tag{7}$$

$$T_s = K_s\theta_s + D_s\dot{\theta}_s \tag{8}$$

$$J_r\dot{w}_r = T_a - T_s - D_r w_r \tag{9}$$

$$J_g\dot{w}_g = nT_s - T_e - D_g w_g \tag{10}$$

where θ_s is the shaft torsion, T_s the transmitted torque, and T_e the electromagnetic torque generated by the PMSG.

2.3.4. Electrical Model

The generator is represented in the synchronous dq reference frame, with the electrical angular speed defined as $\omega_e = p\omega_g$. The dynamic equations of the stator are given by:

$$v_d = R_s i_d - w_e L_q i_q + L_d i_d \tag{11}$$

$$v_q = R_s i_q + w_e\left(L_d i_d + \psi_f\right) + L_q i_q \tag{12}$$

and the electromagnetic torque is expressed as follows:

$$T_e = \frac{3}{2}p\left[\psi_f i_q + \left(L_d - L_q\right) i_d i_q\right] \tag{13}$$

Under nominal operating conditions, the direct-axis current is typically set to $i_d \approx 0$ to minimize copper losses, while the quadrature-axis current i_q is controlled according to the MPPT strategy proposed in [5].

2.3.5. Pitch Actuator Model

The pitch-angle actuation system is approximated by a first-order dynamic model with a speed saturation limit [14]:

$$\dot{\beta} = \frac{-\beta + \beta_{cmd}}{\tau_\beta}, \quad \left|\dot{\beta}\right| \leq \dot{\beta}_{max} \tag{14}$$

where β_{cmd} is the commanded pitch angle, τ_β the time constant of the servo mechanism, and $\dot{\beta}_{max}$ the maximum admissible pitch rate.

This simplified representation is widely adopted in hierarchical and predictive pitch-control architectures for wind turbines [14].

2.4. Machine Learning-Based Anticipatory Rotor-Speed Prediction

The anticipatory pitch-control strategy builds upon a machine learning-based prediction model that provides a future estimate of the rotor speed, allowing the controller to react before wind disturbances effectively reach the turbine. The AI model predicts the rotor speed at a fixed horizon ahead in time (t + Δt), using a sliding window of recent wind–rotor dynamics as input.

The approach relies on a multiyear dataset containing synchronized measurements of wind speed, rotor angular velocity, and pitch-actuator position under a wide range of operating conditions. From these historical records, a nonlinear regression model is trained to learn the temporal relationship between the evolution of wind patterns and the resulting rotor-speed response. Once trained, the model operates in real time, taking as input the most recent window of measurements and producing an anticipated rotor-speed value.

In the proposed PI-Predictive configuration, the measured rotor-speed feedback is fully re-placed by a short-term predicted rotor-speed signal within the PI pitch-control loop, as illustrated in Figure 5. By closing the loop over an estimated future rotor speed, the controller reacts to the anticipated dynamic state of the turbine rather than to the instantaneous measurement. This design choice is intentionally adopted to compensate for aerodynamic and pitch-actuator delays and to isolate the effect of anticipatory feedback under highly variable wind conditions. From a safety perspective, the anticipatory feedback enables earlier pitch activation during sudden wind gusts, thereby reducing the likelihood of excessive rotor acceleration, mitigating overspeed events, and limiting transient mechanical loads on the drivetrain and blades. The physical rotor-speed measurement remains available for monitoring and performance evaluation, but is not used as the feedback variable in the control loop in the proposed configuration.

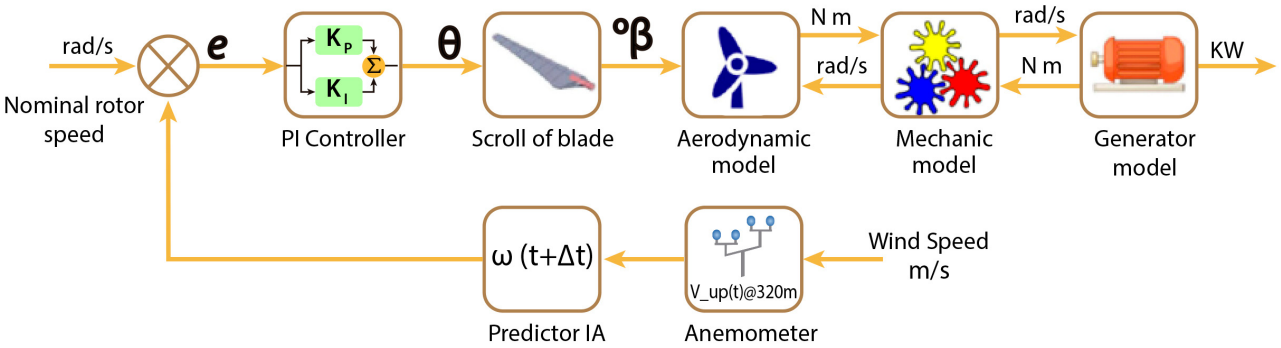


Figure 5. Schematic diagram of the proposed hybrid control architecture. The AI-based model predicts the future rotor speed over the control horizon, and this anticipatory estimate is fed into the PI pitch controller as a feedforward signal to enable early aerodynamic actuation.

2.4.1. Data Structuring and Cleaning

A six-year historical dataset containing wind speed, pitch angle, and rotor speed measurements was used to train and evaluate the predictive model. Although the available meteorological record spans a longer period used for long-term statistical characterization, only the most recent six years (2019–2024) were selected for the machine learning stage to balance computational efficiency and predictive performance. The dataset was temporally structured to ensure balanced training and rigorous performance assessment. The first four years were assigned for training, allowing the neural network to learn the dynamic patterns of the system under varying wind conditions. The fifth year was used for valida-

tion, enabling hyperparameter tuning and reducing overfitting risk, while the sixth year was reserved for testing to evaluate the model's generalization capability on previously unseen data.

To preserve the temporal consistency of the signals, the data were resampled to a uniform time step of 1 s. This fixed sampling frequency is essential for the model to accurately capture the temporal dependencies among variables without being affected by irregularities in the original acquisition process. All dataset partitions (training/validation/testing) were kept in chronological order, without shuffling, to prevent information leakage.

Obvious outliers, such as unrealistic pitch spikes or abrupt jumps in rotor or wind speed, were removed using a rule-based physical clipping approach that restricts each variable to its physically feasible operating range [30]. This ensures that only physically inconsistent measurements, typically originating from sensor noise or transient acquisition faults, are suppressed. Consequently, the model is trained on data that are both reliable and dynamically representative of real operating conditions.

Following data structuring, a signal filtering stage was applied to suppress high-frequency measurement noise and spurious spikes while preserving the physically meaningful dynamics of the wind flow. A Savitzky–Golay filter (polynomial order 3, window length 11) was implemented to smooth abrupt variations without distorting the transient behavior of gusts [31]. This process enhances signal quality and prevents the learning model from overfitting to spurious fluctuations. The same filter configuration used during training was identically applied in validation and testing.

Subsequently, a feature extraction stage was carried out to enrich the input space with descriptive variables derived from the raw measurements. Temporal derivatives (Δv, $\Delta \omega$, $\Delta \beta$) were computed to represent short-term variations in wind, rotor, and pitch dynamics. Additionally, sliding-window statistics, mean, standard deviation, skewness, and kurtosis, were calculated to capture the local energy and variability in the wind field [32]. This set of engineered features enhances the model's ability to identify gust patterns and nonlinear correlations across variables. The list of selected features was preserved as an artifact to ensure reproducibility.

All variables were then scaled to numerical values of order unity to avoid scale-induced bias during training. The signals were first expressed in consistent physical units (pitch in degrees, rotor speed in revolutions per minute, and wind speed in meters per second) and then normalized using fixed statistics (minima, maxima, means, and standard deviations) computed from the training set. The same normalization parameters were reapplied without modification during validation and testing, ensuring balanced contributions from all variables and facilitating faster model convergence [33].

Finally, input–target pairs were constructed using a temporal shift of 20 s, enabling the model to learn the future behavior of rotor speed or wind with the same anticipation horizon. This delay was estimated based on the physical dynamics of gust propagation, considering wind speeds up to 15 m/s and a 320 m distance between the anemometer and the turbine. This temporal-shift strategy forms the foundation of the predictive model, enabling proactive pitch control actions that mitigate overspeed events during sudden wind fluctuations [34]. The supervised datasets were stored as X*.npy and y*.npy files (training/validation/testing).

The data preprocessing pipeline comprised five sequential stages, cleaning, filtering, feature extraction, normalization, and input–target pairing, as illustrated in Figure 6. During training, all metrics and configurations were automatically saved to ensure that the exact same artifacts were consistently applied during validation and testing, eliminating discrepancies between phases.

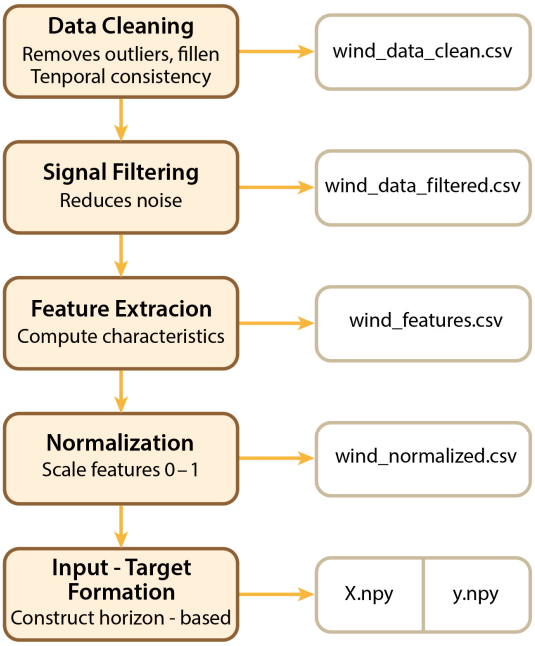


Figure 6. Data preprocessing workflow illustrating the sequential transformation of raw measurements into model-ready datasets. Each stage outputs an intermediate file: (1) cleaned data (CSV), (2) filtered signals (CSV), (3) feature-augmented dataset (CSV), (4) normalized dataset (CSV/NPY), and (5) input–target matrices (X, y) used for neural network training with a 20 s prediction horizon.

2.4.2. Comparative Assessment of Machine Learning Models for Rotor Speed Forecasting

During the training phase, an automatic model selection procedure was implemented to identify the most suitable predictive model for rotor speed estimation. Three representative approaches, widely used in the modeling of dynamic systems, were evaluated: Ridge regression, Random Forest (RF), and Multilayer Perceptron (MLP) neural networks. The comparison was performed on the validation dataset using the Root Mean Square Error (RMSE) as the primary metric, expressed in physical units (revolutions per minute, rpm). This metric was selected for its sensitivity to large deviations and its ability to accurately quantify the model's capacity to anticipate overspeed events [35].

The Ridge regression model is an extension of conventional linear regression that incorporates an L2 regularization term, penalizing the magnitude of regression coefficients [36]. This regularization introduces a controlled bias that reduces model variance, stabilizing parameter estimation when predictor variables exhibit high collinearity. In the context of wind turbine control, Ridge regression offers a robust linear approximation of the relationships between wind speed, pitch angle, and rotor speed, performing effectively when system dynamics remain predominantly linear and sensor noise is moderate.

The RF model, on the other hand, belongs to the family of ensemble methods and is based on aggregating multiple independent decision trees, each trained on random subsets of data and input features [37]. The individual tree predictions are averaged to reduce variance and enhance generalization capacity while capturing nonlinear interactions among variables.

The MLP represents a feed-forward deep learning model, capable of approximating highly nonlinear functions through multiple layers of interconnected neurons with nonlinear activation functions. The model is trained using backpropagation, minimizing a loss function (in this case, the RMSE) through iterative gradient-descent optimization of synaptic weights. MLP networks are particularly effective at capturing temporal and nonlinear dependencies between different variables [38].

During the validation stage, all three models were trained under equivalent preprocessing conditions, using identical feature sets, normalization parameters, and prediction horizons. The model that achieved the lowest RMSE on the validation set was automatically selected as the most accurate, ensuring an objective and data-driven decision. Subsequently, the selected model was retrained using the combined training and validation datasets, maintaining the chronological order of data to preserve temporal coherence and prevent information leakage.

Finally, model performance was evaluated on the test dataset using three complementary metrics: RMSE, Mean Absolute Error (MAE), and the coefficient of determination (R^2). All metrics were computed on de-scaled data to ensure physical interpretability.

This automated selection and evaluation process enabled the identification of the most accurate, robust, and generalizable model, ensuring consistent integration into the predictive pitch-control framework. The overall workflow, model training, validation, retraining, and final evaluation, is illustrated in Figure 7.

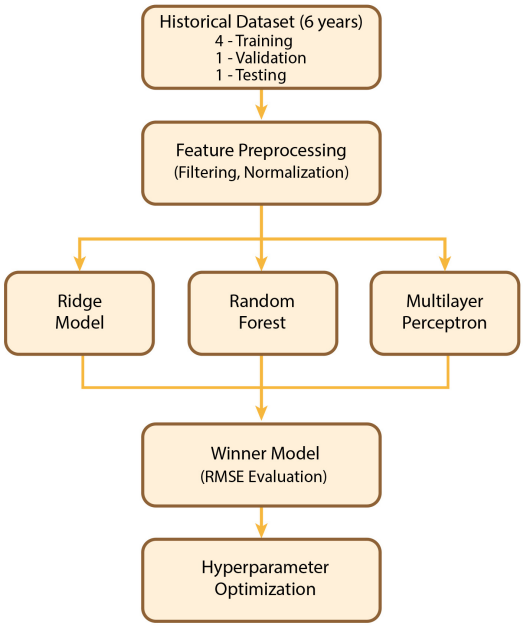


Figure 7. Comparative performance of Ridge regression, Random Forest, and MLP neural network models in rotor speed forecasting. Each model was trained under identical preprocessing and validation conditions. The model achieving the lowest RMSE was automatically selected and integrated into the predictive pitch control framework.

2.4.3. Hyperparameter Optimization

The purpose of this stage is to identify the optimal combination of hyperparameters that maximizes the model's predictive capability and stability while maintaining a balance between accuracy, generalization, and computational efficiency. It should be noted that, in this phase, the detailed hyperparameter optimization is applied only to the best-performing model identified in the previous section, in order to focus computational resources on the approach with the highest predictive potential.

To analyze the influence of temporal memory on the system's predictive capacity, three historical window lengths are evaluated, (30, 60, 90) seconds, under identical preprocessing, filtering, and normalization conditions while maintaining a fixed prediction horizon of 20 s. This parameter determines the number of past samples used to describe the dynamic state of the system, directly affecting the number of input features and the model's sensitivity to wind gusts or gradual variations.

A random search strategy is implemented to efficiently explore the hyperparameter space through a limited number of randomly sampled combinations, avoiding the high computational cost associated with an exhaustive grid search.

Each configuration is trained using threefold time-series cross-validation, where the data are split into chronologically ordered folds to avoid information leakage from future to past samples. The coefficient of determination (R^2) serves as the main performance metric in this stage, quantifying the proportion of variance explained by the model.

The Ridge Regression model regularizes its coefficients through a penalty parameter (α) that balances bias and variance.

The RF model, based on an ensemble of decision trees, depends on the internal tree structure and the overall forest size; its hyperparameters define model complexity and generalization capacity.

The MLP model, of neural nature, requires specification of its architecture and training parameters, which directly influence its learning capability and convergence behavior.

The tunable parameters considered during the development of the three candidate models are summarized in Table 3; in the final optimization stage, the search was restricted to the best-performing model selected in the previous subsection.

Table 3. Hyperparameters search space and configuration ranges used for model optimization.

Model	Hyperparameter	Search Range/Options
Common configuration	Historical window	{30, 60, 90}
	Prediction horizon	20
	Sampling interval	1
Ridge Regression	Regularization factor	$[10^{-4}, 10^{3}]$
	Optimization method	{'auto', 'saga', 'lsqr'}
Random Forest	Number of trees	10–1200
	Maximum depth	10–80
	Minimum samples per leaf	1–10
	Maximum features per split	$\{\sqrt{}, \log_2, \text{None}\}$
	Bootstrap sampling	{True, False}
Multilayer Perceptron	Hidden layer sizes	[(60,), (80,), (100,), (60,40)]
	Activation function	{'relu', 'tanh'}
	Regularization parameter	$[10^{-5}, 10^{-2}]$
	Initial learning rate	$[10^{-4}, 10^{-2}]$
	Optimizer	{'adam', 'lbfgs'}

In all cases, the models share the same data partitions (training, validation, and testing) and the same evaluation criteria, ensuring methodological consistency and direct comparability between algorithms and historical window lengths.

2.4.4. Integration of the Predictive Model with the PI Controller

The wind turbine model is developed in the MATLAB/Simulink environment and is structured into three main subsystems: aerodynamic, mechanical, and electrical. Each subsystem exchanges key physical variables: the aerodynamic model receives the wind speed and pitch angle to generate the rotor torque; the mechanical model transforms this torque into generator speed; and the electrical model produces the electromagnetic torque feedback to the shaft. The PI control loop regulates the pitch angle based on the rotor speed, completing the feedback structure of the system, as illustrated in Figure 8.

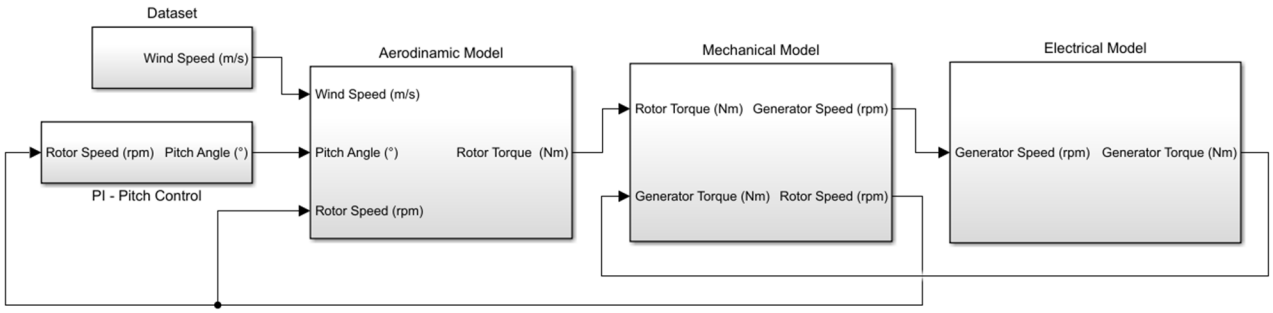


Figure 8. Block diagram of the wind turbine model implemented in MATLAB/Simulink. The system consists of three coupled subsystems: aerodynamic, mechanical, and electrical. This structure serves as the physical core of the hybrid system, into which the predictive model is integrated to provide an anticipatory (feed-forward) correction signal for the pitch control loop.

The PI controller is implemented with gains $Kp = 4$ and $Ki = 0.01$, which are empirically tuned to achieve a stable and fast response without overshoot. This control loop maintains the rotor's nominal speed under variable wind conditions by adjusting the pitch angle as the main actuation variable.

Based on the training and tuning of the predictive model with the best performance for estimating the future rotor speed, the model receives wind speed sequences with the same temporal resolution as in the training phase (1 Hz). These inputs allow for the estimation of the rotor speed at a 20 s prediction horizon, generating an anticipatory signal $\omega_r\ (t + 20)$. The predicted outputs are then temporally aligned to the current instant by shifting them 20 s backward, allowing the controller to operate with predictive rather than purely reactive information.

When integrated with the physical model in MATLAB/Simulink, the system adopts a hybrid architecture (physical model + AI). In this configuration, the anticipatory signal generated by the predictive model functions as a corrected input or feed-forward term within the PI control loop, complementing the conventional feedback action. This hybrid architecture enables the controller to react proactively to sudden increases in wind speed, reducing the mechanical delay in pitch adjustment and improving the overall dynamic stability of the rotor.

The system performance is evaluated using several key metrics: overspeed reduction, integral absolute error (IAE) of rotor speed, pitch actuator activity (peak and RMS values), and settling time. This evaluation confirms the contribution of the predictive model to enhancing the dynamic behavior and responsiveness of the control system under varying wind conditions.

2.4.5. Experimental Setup and Simulation Scenarios

The experimental framework was designed to integrate the predictive model developed in Python programming language [39] with the pitch control strategy implemented in MATLAB/Simulink software [40]. The simulations were executed offline, allowing the predicted rotor speed to be fed into the PI controller for performance evaluation under variable wind conditions.

All experiments and simulations were performed on a personal computer equipped with an Intel® Core™ i5-3340M CPU @ 2.70 GHz (2 cores, 4 threads), 12 GB DDR3 RAM, and integrated Intel® HD Graphics 4000 GPU [41].

The system operated on Windows 10 Pro (Version 22H2, Build 19045.6456), providing adequate resources for model training, hyperparameter optimization, and one-hour Simulink simulations with a 1 Hz sampling step.

Two main environments were used:

- Python 3.10 running on Google Colab, for data preprocessing, feature extraction, model training, and hyperparameter optimization. The following libraries were employed: NumPy, Pandas, Scikit-learn, SciPy, Joblib, and Pathlib. The best-performing model (Random Forest Regressor) was serialized in .joblib format and later used for prediction.
- MATLAB R2018b (9.5.0, 64-bit) with Simulink toolbox, for implementation of the mechanical, electrical, and aerodynamic subsystems, as well as the PI pitch controller.

The wind–rotor interaction was simulated for a 3600 s window with a sampling rate of 1 Hz, representing a one-hour operating period under variable wind conditions. The predictive model output was imported into Simulink to anticipate rotor-speed variations and adjust the pitch angle accordingly. Each complete simulation cycle required approximately 1 min of execution time, confirming the computational efficiency of the setup.

Five simulation scenarios were designed to evaluate both the control performance and the robustness of the predictive model:

1. Baseline Case (Without Predictor): The first scenario used the actual rotor speed feedback from the physical model, without any predictive correction. This served as a reference case for direct comparison, allowing the PI controller to respond reactively to the measured wind variations.
2. Predictive Case (With Rotor Speed Prediction): The same wind sequence from the baseline case was used, but the rotor speed input was replaced by the value predicted by RF model. This configuration enabled quantitative comparison of the controller's anticipatory response.
3. Unseen Wind Scenario A: A new wind sample not included in the training, validation, or testing datasets was used to assess model generalization under realistic but previously unseen conditions.
4. Unseen Wind Scenario B: Another segment of real wind data was applied, emphasizing transient gusts and recovery behavior, to evaluate the system's dynamic stability.
5. Unseen Wind Scenario C: A third independent segment was selected to validate the robustness of both the predictor and the PI controller under long-duration, irregular wind profiles.

All unseen wind sequences were extracted from historical records but were not synthetically generated, ensuring the physical realism of the validation process.

The comparative performance between predictive and non-predictive configurations was evaluated using RMSE, R^2, and the stability of the rotor speed relative to the nominal value.

3. Results

This section presents the experimental and simulation results derived from the workflow described in Section 2. The presentation follows a progressive structure: first, the performance of the predictive models during training and validation is examined; next, the selected predictor is integrated into the PI-based pitch-control loop; and finally, the full turbine–predictor system is evaluated under multiple wind scenarios. This organization enables a clear quantitative comparison of the predictive model's accuracy and its impact on the turbine's dynamic stability and responsiveness.

3.1. Model Training and Validation

The dataset, temporally divided into four years for training, one for validation, and one for testing at 1 Hz, was preprocessed by applying a Savitzky–Golay filter (window = 11,

polynomial order = 2) to suppress high-frequency measurement noise while preserving the transient structure of gusts. All variables were normalized using the mean and standard deviation computed from the training set, and the same statistics were consistently applied during validation and testing to ensure coherent scaling across phases.

Each supervised sample was constructed using a 60 s sliding window of historical measurements (120 combined wind-speed and rotor-speed features) and a 20 s prediction horizon. This configuration captures both rapid fluctuations and slower wind-pattern evolutions that strongly influence the rotor-speed response.

Three representative modeling approaches, widely used in the prediction of dynamic systems, were implemented and compared with the objective of identifying the model with the highest generalization capability for estimating rotor speed, expressed in revolutions per minute (rpm). The evaluated models and their key parameters were as follows:

- Ridge Regression: a linear regularized model that applies a quadratic (L2) penalty to the regression coefficients to prevent overfitting and improve numerical stability in the presence of correlated variables. The regularization term $\alpha = 1.0$ was fixed after preliminary tests showed an appropriate balance between bias and variance.
- RF: an ensemble method composed of 200 independent decision trees, where each tree learns from a random subset of samples and features. The number of trees was determined experimentally through successive iterations (ranging from 100 to 300 trees), with 200 providing stable error convergence without a significant increase in computation time. The parameter min_samples_leaf = 2 was selected to limit excessive tree depth and reduce overfitting under irregular wind gusts.
- MLP: configured with hidden layers = (128, 64), ReLU activation, learning rate = 10^{-3}, alpha = 10^{-4}, and early stopping enabled. This configuration was defined after testing multiple topologies (64–32, 128–128, 256–128), where the 128–64 architecture achieved the best accuracy without compromising training stability. A learning rate of 0.001 and an L2 regularization term (alpha = 10^{-4}) ensured smooth convergence, while the early stopping criterion prevented overfitting by halting training when the validation error ceased to improve.

Validation was performed on an independent dataset using RMSE as the primary metric, complemented by MAE and the coefficient of determination (R^2). All metrics were computed on de-normalized data to ensure physical interpretability. Model selection was based strictly on the lowest RMSE achieved on the validation set.

Table 4 summarizes the performance metrics obtained during validation. RF model achieved the best balance between accuracy and robustness, with RMSE = 0.265 rpm, MAE = 0.194 rpm, and $R^2 = 0.4929$, outperforming both Ridge regression and the MLP network.

Table 4. Validation performance metrics of predictive models.

Model	RMSE (rpm)	MAE (rpm)	R^2
Ridge	0.273	0.195	0.4646
Random Forest	0.265	0.194	0.4929
Multilayer Perceptron	0.280	0.201	0.4364

After selecting the best-performing model, it was retrained using the combined training and validation datasets and then evaluated on the independent test set, achieving RMSE = 0.340 rpm, MAE = 0.225 rpm, and $R^2 = 0.7292$. The higher R^2 value observed in the test set is attributed to smoother wind-speed segments present in the test partition, which are less stochastic and therefore yield more predictable rotor-speed responses.

The total computation time for the full pipeline, including window construction (≈210,000 samples), model training, validation, and final testing, was approximately 8 h on a CPU-only system.

The overall modeling workflow is summarized in Algorithm 1, which outlines the main stages followed for data preparation, training, and predictive model selection.

Algorithm 1: Predictive Model Training and Selection Workflow.

```
# Inputs:
# Wind and rotor-speed data (1 Hz), history = 60 s, horizon = 20 s

# Output:
# Best predictive model and performance metrics (RMSE, MAE, R^2)

Load datasets for training, validation, and testing
Clean and interpolate data; apply physical limits on wind_speed and rotor_speed
Filter signals using Savitzky–Golay filter (window = 11, polynomial = 2)
Construct supervised windows:
    X ← [wind_speed_f, rotor_speed_f] over 60 s history
    y ← rotor_speed_f at t + 20 s
Normalize X and y using statistics from training set
Define candidate models:
    Ridge(α = 1.0)
    RandomForest(n_estimators = 200, min_samples_leaf = 2)
    MLP(hidden_layers = (128, 64), activation = ReLU, learning_rate = 1 × 10^−3,
        alpha = 1 × 10^−4, early_stopping = True)
For each model in candidates:
    Train model on (X_train, y_train)
    Evaluate on validation set → compute RMSE, MAE, R^2 (in rpm)
Select model with lowest RMSE on validation data
Retrain selected model using (train + validation) data
Evaluate final performance on independent test set
Export artifacts:
    Trained model (.joblib)
    Report file (.json) with metrics
    Test predictions (.csv)
Return:
    Best_model
    Test_metrics
```

The proposed structure ensures the reproducibility of the process and the traceability of all generated artifacts (parameters, models, and performance metrics). The selected model serves as the reference baseline for the subsequent hyperparameter optimization phase (R^2).

3.2. Performance After Hyperparameter Optimization

Following the comparative evaluation of the three modeling approaches, RF model exhibited the most stable and accurate performance during validation. In addition to achieving the lowest RMSE, RF showed lower sensitivity to measurement noise and required fewer training samples to reach convergence compared with the MLP architecture. For these reasons, RF was selected for hyperparameter optimization.

Adjusting the historical window length enabled the identification of the model's effective memory horizon, reflecting the dominant temporal scales that characterize wind turbulence (typically 10–60 s). A 30 s window improved responsiveness but reduced R^2 by approximately 2% due to the loss of slow-varying contextual information that influences rotor-speed dynamics. Under this configuration, the model completed training in approximately 6 h.

In contrast, a 90 s window captured slower variations in wind dynamics more effectively; however, training time increased substantially (≈26 h). The marginal improvement in R^2, combined with a clear tendency toward overfitting, made this configuration unsuitable for real-time anticipatory applications.

Therefore, the 60 s window, with a training time of 14 h, was adopted as the optimal temporal horizon, providing a balanced trade-off between model complexity, generalization capability, and computational effort. This configuration exhibited the lowest temporal sensitivity, quantified as the smallest change in RMSE per unit change in window length, making it especially suitable for stable predictive control operation.

After establishing the optimal historical configuration, a random search procedure was applied to the RF model to explore key hyperparameters such as the number of trees, maximum depth, and sampling ratio.

The final configuration achieved an effective balance between accuracy and generalization, as summarized in Table 5, which presents the optimal parameters obtained after 20 randomized search iterations using threefold cross-validation. The resulting structure, with 424 estimators, a maximum depth of 21, and a sampling ratio of 0.8, proved compact, stable, and computationally efficient under variable wind conditions.

Table 5. Optimal hyperparameters values selected after the optimization process.

Model	Hyperparameter	Search Range/Options
Common configuration	Historical window	60
	Prediction horizon	20
	Sampling interval	1
Random Forest	Number of trees	424
	Maximum depth	21
	Minimum samples per leaf	3
	Maximum features per split	sqrt
	Bootstrap sampling	True

The comparison between the baseline and tuned configurations, summarized in Table 6, revealed that all test-set metrics (RMSE, MAE, and R^2) differed by less than 2%, which is physically negligible. The RMSE difference of only 0.008 rpm corresponds to approximately 0.012% of the nominal rotor speed (64.29 rpm), well below the sensitivity threshold of the pitch-control system and with no measurable effect on regulation performance.

Table 6. Validation performance metrics for baseline and tuned model configurations.

Model	RMSE (rpm)	MAE (rpm)	R^2
Baseline	0.340	0.225	0.729
Tuned	0.348	0.229	0.716

The variation between the baseline and tuned configurations, RMSE = 0.340 rpm vs. 0.348 rpm, corresponded to an absolute difference of only 0.008 rpm, equivalent to a few hundredths of a revolution per minute. Considering a nominal rotational speed of 64.29 rpm, this deviation represented merely 0.012% of the operating magnitude, well

below the sensitivity threshold of the control system and without any observable impact on the pitch regulation dynamics.

Therefore, the hyperparameter optimization primarily improved the stability and consistency of the model rather than its absolute accuracy. The tuned model exhibited lower variance across folds and smoother responses to transient wind variations, confirming its suitability for predictive and anticipatory control applications.

Overall, these findings demonstrated that the RF model maintained high predictive accuracy after optimization, achieving greater robustness and generalization capability. When combined with the temporal sensitivity analysis (30 s, 60 s, and 90 s), the results confirmed that a 60 s historical window offered the best compromise between model complexity, computational efficiency, and prediction reliability, key factors for anticipatory control systems.

To standardize the process and ensure reproducibility, the optimization was implemented through Algorithm 2, which performed the final evaluation on the test set, reporting RMSE, MAE, and R^2 in rpm after inverse normalization. The algorithm stored the tuned model, tuning report, and test-set predictions for subsequent analysis.

Algorithm 2: Automated Hyperparameter Tuning and Final Evaluation.

```
## Inputs:
# BASE folder with: params.json, training_report.json, train_X.npy, train_y.npy
    val_X.npy, val_y.npy, test_X.npy, test_y.npy.
# Config: N_ITER, RANDOM_STATE, CV_SPLITS

## Output:
# best_model_TUNED.joblib, tuning_report.json, test_predictions_TUNED.csv

# Define randomized search spaces
Ridge: alpha~loguniform [1 × 10^−4, 1 × 10^3]
RandomForest:
    n_estimators ∈ [250, 600], max_depth ∈ [10, 22], min_samples_leaf ∈ [2, 5],
    max_features = 'sqrt', bootstrap = True, max_samples ∈ {0.6, 0.7, 0.8}
MLP:
    hidden_layer_sizes ∈ {(64, 32), (128,), (128, 64), (128, 64, 32), (256, 128)},
    Activation ∈ {relu,tanh}, alpha ~ loguniform [1 × 10^−6, 1 × 10^−2],
    learning_rate_init~loguniform [1 × 10^−4, 5 ×10^−2], batch_size ∈ {128, 256, 512}
# RandomizedSearchCV + 3-fold CV per candidate:
    Fit on (Xtr, ytr); keep best_est and best_params.
    Predict on Xva; inverse-scale y (rpm): VALIDATION RMSE, MAE, R^2.
    Record (model, best_params, validation metrics)
# Select tuned winner:
    winner_name ← argmax(R^2) or argmin(RMSE) across candidates.
    winner_estimator ← best_est(winner_name)
# Retrain and test:
    Concatenate TRAIN+VAL; fit winner_estimator on (X_trval, y_trval).
    Predict on Xte; inverse-scale to rpm; compute TEST RMSE, MAE, R^2.
# Persist artifacts:
    Save tuned model (.joblib), tuning_report.json (config, CV, metrics, files used),
    test_predictions_TUNED.csv (rpm), and optional CV top tables.
```

3.3. Dynamic Response of the Wind Turbine Model Under PI Pitch Control

The dynamic model of the wind turbine was implemented in MATLAB/Simulink to reproduce the coupled aerodynamic, mechanical, electrical, and control interactions governing the rotor-speed response. The PI pitch controller regulates the blade attack angle relative to the incoming wind, modifying the aerodynamic power coefficient and, therefore, the torque applied to the low-speed shaft. The actuator operates under a mechanical pitch-rate limit of approximately 3°/s, which directly affects the controller's ability to reject fast wind disturbances. This mechanism constitutes the primary means of rotor-speed stabilization under variable wind conditions.

When the wind speed exceeded its nominal operating level, the controller increased the pitch angle to reduce the aerodynamic torque and avoid overspeed, whereas decreasing wind velocities triggered the opposite action to maximize energy capture. Although this behavior is consistent with classical PI regulation, the response remains constrained by both the actuator rate limit and the intrinsic delay of the feedback loop.

The controller continuously adjusted the pitch angle based on the deviation between the measured rotor speed and its nominal reference (64.29 rpm). Under moderate wind fluctuations, these corrections maintained dynamic equilibrium with minimal deviation. However, in the presence of sharp gusts, the combination of mechanical delay and feedback latency produced transient mismatches between the demanded and actual pitch, causing short overspeed events before corrective action was fully executed.

This behavior is illustrated in Figure 9, which shows the temporal evolution of wind speed, the rotor speed response relative to its nominal reference, and the corresponding modulation of the pitch angle. The figure confirms the stable operation of the model and the interaction of the PI controller in maintaining the rotor speed within the nominal regime under variable wind conditions.

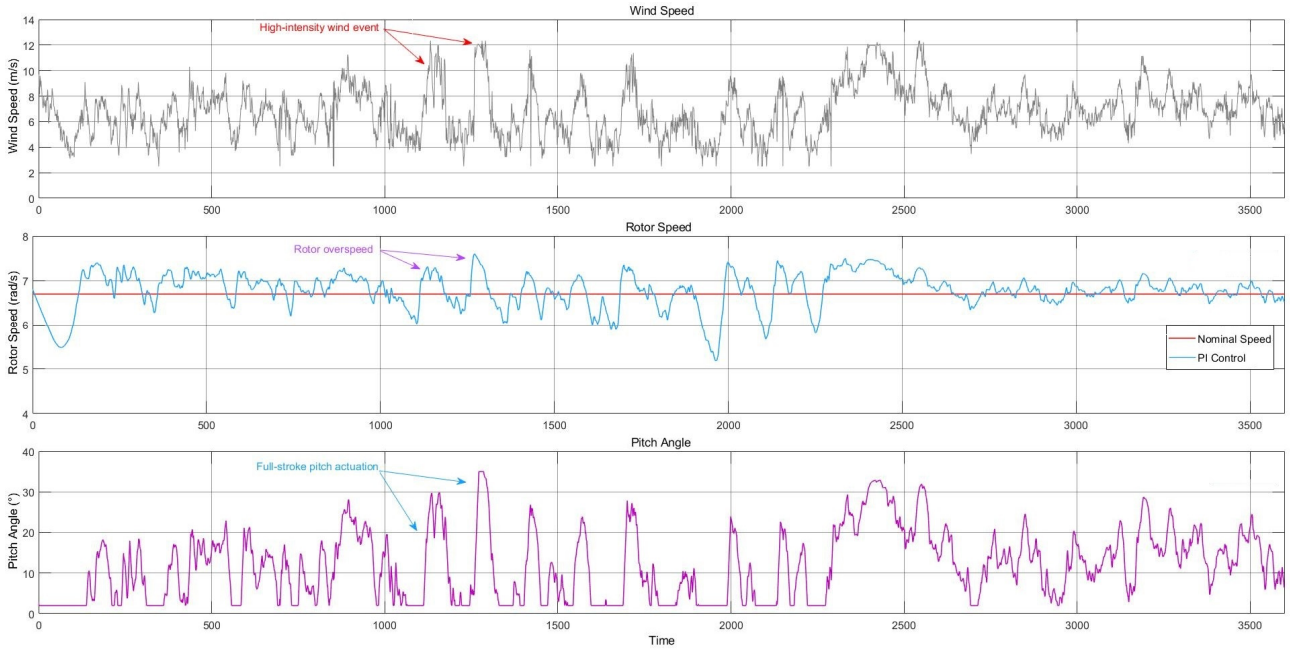


Figure 9. Dynamic response of the wind turbine model under PI pitch control. The upper panel shows the wind speed input with gusts exceeding 10 m/s; the middle panel presents the rotor speed relative to its nominal reference; and the lower panel displays the pitch angle modulation between 0° and 35°.

Throughout the simulation, the aerodynamic, mechanical, and electrical subsystems remained well-coupled, and the PI controller successfully kept the rotor speed within nominal limits for most of the operating period. Nonetheless, noticeable overspeed peaks (~0.4–0.8 rpm) appeared during steep wind gradients, driven by rapid aerodynamic torque spikes. These events highlight a key limitation of purely reactive PI control: its inability to compensate for fast disturbances within the time frame imposed by the actuator dynamics. This result reinforces the need for a predictive or anticipatory strategy capable of initiating pitch adjustments before the disturbance reaches the rotor.

3.4. Simulation Scenario with Seen Wind Data

To incorporate predictive capability into the control architecture, the model was augmented with an additional module that computes the anticipated rotor speed 20 s ahead. The predicted signal $\hat{\omega}_r(t+20)$ is time-aligned and injected directly into the PI controller, explicitly replacing the real-time rotor-speed feedback used in the conventional scheme. This modification required restructuring the signal-routing logic in Simulink to ensure proper temporal synchronization between the aerodynamic subsystem, the drivetrain dynamics, and the predictive block. The resulting hybrid architecture is shown in Figure 10.

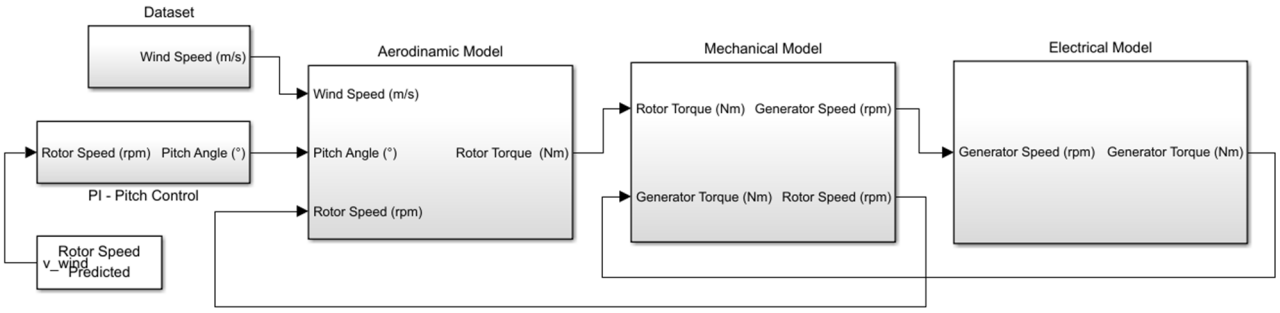


Figure 10. Integration of the AI-based Predictive Module into the Wind Turbine Control Architecture.

In this scenario, the dynamic response of the wind turbine was analyzed by comparing the conventional PI controller, which acts exclusively based on the measured rotor speed, with the predictive PI controller, which uses the rotor speed anticipated by the artificial intelligence model. The same wind profile used in the testing and validation phase of the dynamic model was employed, corresponding to a continuous extraction of 3600 s from the training dataset. This section was selected because it contains gusts of varying magnitude, abrupt changes, and segments of nearly stationary wind, which allows for the reproduction of realistic conditions and ensures the comparability between both controllers.

Figure 11 summarizes the system response under this operating condition. The upper panel presents the wind-speed profile used in the experiment, characterized by high variability and intermittent gusts. The middle panel compares the rotor-speed trajectories of both controllers, showing that the predictive strategy reduces overspeed events by anticipating the system's dynamic behavior. The lower panel displays the pitch-angle actuation, where the predictive controller produces earlier and smoother adjustments, thereby reducing mechanical effort and avoiding the delayed corrective actions observed in the conventional PI scheme.

The middle panel of Figure 11 shows that the PI-Predictive strategy maintains the rotor speed much closer to the nominal reference (6.7 rad/s), particularly during high-intensity wind events. While the conventional PI controller reacts only after acceleration has already occurred, the predictive controller begins increasing the pitch several seconds earlier, avoiding the torque spike that causes overspeed. This anticipatory action reduces peak deviations

from approximately 0.6–0.8 rad/s (PI-Conventional) to below 0.4 rad/s, and it also limits oscillations around the operating point by preventing late, aggressive corrections.

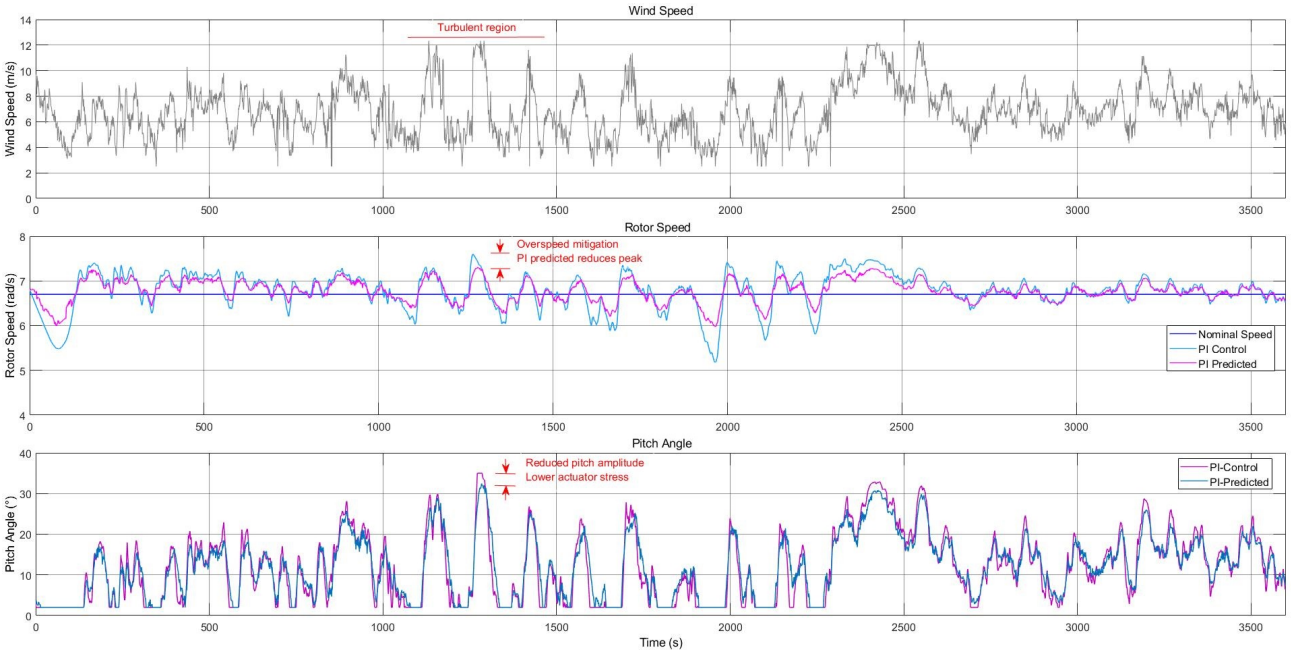


Figure 11. Comparative dynamic response of the conventional PI controller and the predictive PI controller under a 3600 s realistic wind profile.

The ability to initiate pitch motion before the aerodynamic disturbance fully affects the rotor is the main advantage of the predictive approach. Given the mechanical pitch-rate limit of approximately 3°/s, early activation is essential to mitigate fast gusts. By acting ahead of time, the PI-Predictive controller improves transient stability, reduces the likelihood of overspeed conditions, and enhances overall operational safety.

Beyond the visual analysis, the performance was quantified using two complementary metrics, whose results are presented in Table 7. The first metric corresponds to the total error with respect to the nominal rotor speed, capturing deviations both above and below 6.7 rad/s, and therefore assessing how closely each controller maintains the desired operating point throughout the simulation. The second metric considers only values exceeding 6.7 rad/s, isolating the error associated with overspeed conditions, which are the most critical from the standpoint of turbine protection. In both cases, the MAE and RMSE values confirm that the predictive controller achieves the smallest deviations and the lowest overspeed magnitudes. The consistently reduced errors across both scenarios quantitatively demonstrate that the PI-Predictive scheme outperforms the conventional PI controller.

Table 7. Rotor-speed error metrics for both controllers.

Metric	PI-Conventional	PI-Predictive	Best
MAE—Total error	0.3119	0.2047	Predictive
RMSE—Total error	0.4074	0.2567	Predictive
MAE—Overspeed	0.1926	0.1391	Predictive
RMSE—Overspeed	0.2917	0.2108	Predictive

To complement the quantitative analysis, the histogram in Figure 12 presents the distribution of overspeed events, considering only samples above the nominal value. The results show that the PI-Predictive controller not only reduces the magnitude of

overspeed events but also shifts their distribution toward values below 0.4 rad/s, whereas the conventional PI controller exhibits occurrences reaching approximately 0.8 rad/s.

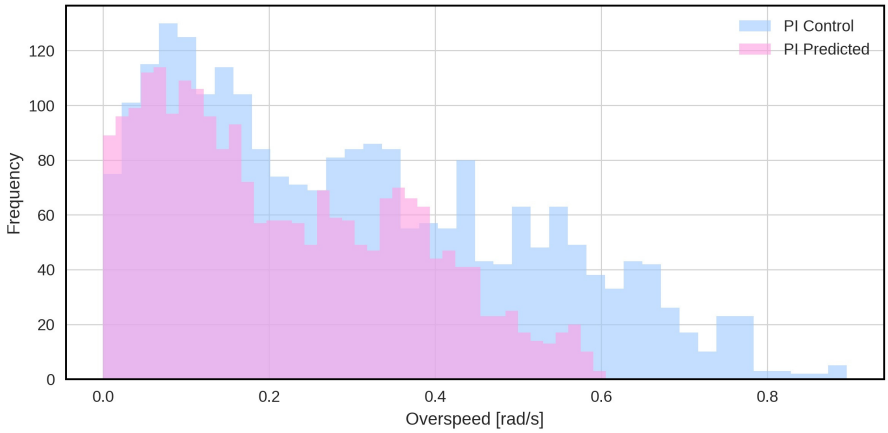


Figure 12. Distribution of overspeed events for the PI-Conventional and PI-Predictive controllers.

The superposition of both distributions confirms that the predictive controller decreases the likelihood of severe events and reduces the dispersion of the phenomenon, in agreement with the lower MAE and RMSE values previously reported. This statistical evidence demonstrates that the PI-Predictive approach not only improves average regulation but also effectively limits the occurrence of critical overspeed conditions, contributing to a more stable and safer turbine operation.

The lower panel of Figure 11 shows that the pitch signal under the PI-Predictive scheme is consistently activated earlier than in the conventional PI configuration. This anticipatory behavior is evident throughout the entire simulation, where the predictive controller initiates aerodynamic adjustments before the rotor speed begins to deviate from its nominal value. By acting ahead of the disturbance, the actuator responds before the deviation materializes, effectively reducing system latency and improving the controller's ability to handle rapid wind fluctuations. In other words, the predictive controller does not correct the deviation after it occurs; instead, it prevents it through early pitch actuation.

To quantify this phenomenon, the activation times of both pitch signals were analyzed using a rate-threshold detector. Each activation event in the predictive controller was paired with its corresponding event in the conventional PI scheme. The anticipation time was defined as $\Delta t = t_{pred} - t_{conventional}$, such that $\Delta t > 0$ indicates earlier activation by the predictive controller. As shown in Figure 13, the PI-Predictive controller advances pitch motion by an average of 7.27 s, with values ranging from near 0 to 20 s. The concentration of events within the 1–10 s interval indicates that anticipatory actuation is a consistent and recurrent behavior rather than an isolated effect.

The observed variability (σ = 6.12 s) reflects the interaction between wind turbulence and the mechanical dynamics of the turbine; however, in all cases, the predictive controller initiates pitch actuation before the conventional scheme. This temporal lead is a key advantage, as it enables the system to preemptively mitigate increases in aerodynamic loading and reduces the likelihood of the rotor entering an overspeed condition. Thus, the PI-Predictive controller not only improves response speed but also enhances transient stability and contributes to safer operation under rapid gusts and abrupt wind fluctuations.

The anticipatory control strategy not only advances pitch activation but also reduces the severity of the aerodynamic adjustments required to stabilize the rotor. The maximum pitch angle decreases from 35.00° (PI-conventional) to 32.47° (PI-Predictive), indicating a lower mechanical demand on the pitch actuator. This trend is also reflected in the distribution of local peaks: the mean peak pitch decreases from 15.55° to 14.49°, and

the corresponding RMS value drops from 17.13° to 15.91°, evidencing a reduction in the effective control effort.

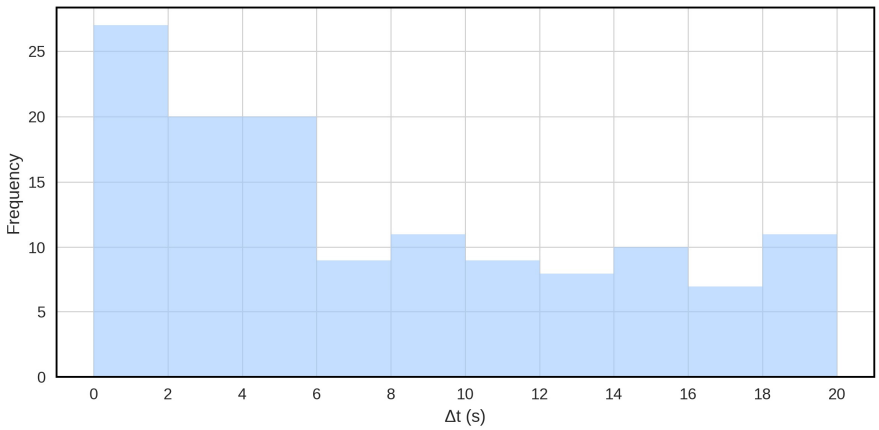


Figure 13. Distribution of pitch-activation anticipation (Δt) between the predictive and conventional PI controllers.

Figure 14 further supports this observation by showing that the distribution of pitch peaks shifts toward lower values under predictive control. Although the PI-Predictive controller produces a higher number of micro-adjustments (767 compared to 291), these corrections are significantly smaller in amplitude, reducing the likelihood of actuator saturation and lowering long-term mechanical stress. This behavior confirms that predictive control not only enhances regulation but also distributes the aerodynamic workload more evenly, contributing to improved stability and actuator longevity.

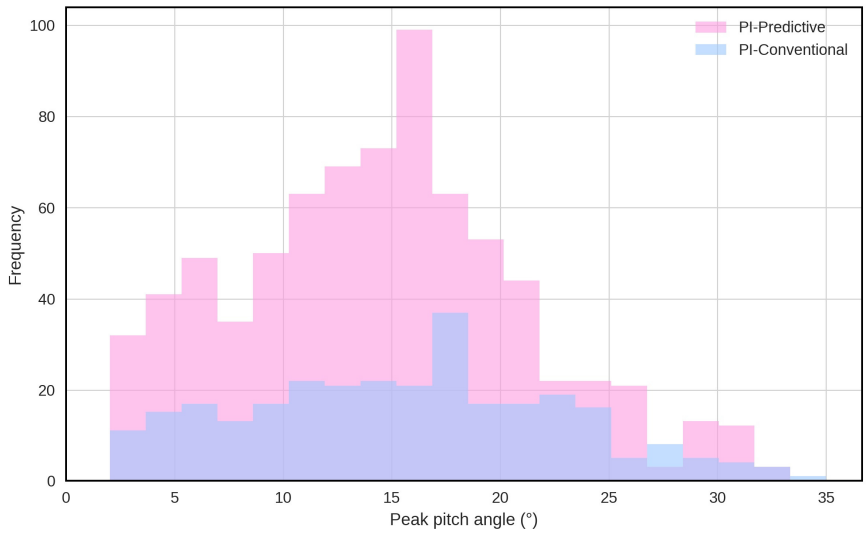


Figure 14. Statistical distribution of local pitch-angle maxima for the conventional PI and PI-Predictive controllers.

In summary, the PI-Predictive scheme limits pitch peaks, smooths transitions, and reduces actuator effort, reinforcing its effectiveness and robustness under rapid wind fluctuations.

3.5. Validation Scenarios with Unseen Wind Data

To assess the generalization capability of the predictive model under conditions not included in the training, validation, or testing datasets, two out-of-distribution scenarios were defined. These scenarios reproduce dynamic conditions that differ substantially from

the historical patterns seen by the model and are therefore well suited for evaluating the robustness and stability of the predictive control strategy.

The first scenario corresponds to a wind profile characterized by sharp high-gradient gusts ($\Delta v/\Delta t > 1.5$ m/s^2 in several intervals), followed by periods of sub-nominal wind. These conditions frequently induce torque spikes in small wind turbines due to their lower inertia, making this scenario particularly demanding for both speed regulation and actuator response. Figure 15 illustrates this non-stationary regime, in which the controller must suppress rapid speed excursions while maintaining stability during low-power intervals.

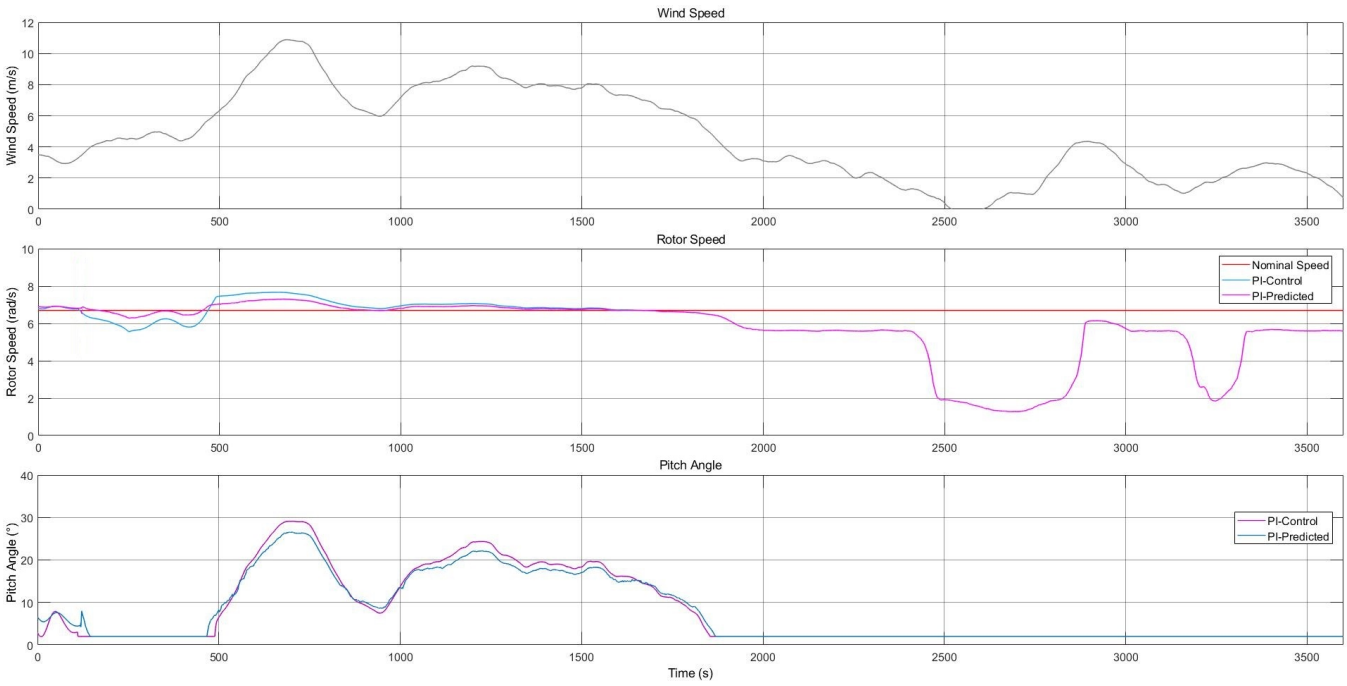


Figure 15. Time evolution of wind speed, rotor speed, and pitch angle for the high-gradient gust scenario with sub-nominal intervals. The responses of the conventional PI controller and the Predictive PI scheme are compared under highly variable aerodynamic conditions.

The quantitative results presented in Table 8 show that both controllers deviated from the nominal operating point due to the strong variability in the wind. However, the Predictive PI controller consistently maintained a rotor speed closer to the nominal value. This is reflected in its significantly lower total MAE (0.0449 rad/s vs. 0.1810 rad/s) and RMSE (0.0595 rad/s vs. 0.2150 rad/s), indicating a much tighter tracking performance across the full simulation window.

Table 8 summarizes the performance of both controllers. The "Total error" metric corresponds to the absolute deviation $|\omega - \omega_nom|$ across the full simulation window, while the "Overspeed error" considers only positive deviations above the nominal value. Despite the strong variability in the wind, the Predictive PI controller maintained a rotor speed much closer to the reference, achieving substantially lower MAE (0.0449 rad/s vs. 0.1810 rad/s) and RMSE (0.0595 rad/s vs. 0.2150 rad/s).

During overspeed intervals, the anticipatory controller again outperformed the conventional scheme. The Predictive PI reduced both the magnitude of and variance in overspeed errors, limiting the maximum deviation to 0.6014 rad/s compared with 0.9679 rad/s for the conventional PI. This reduction confirms that the predictive term responds more effectively to upward wind gradients by initiating pitch motion before the aerodynamic torque spike fully develops.

Table 8. Performance metrics relative to nominal speed and actuator effort for the high-gradient gust scenario.

Metric	PI-Conventional	PI-Predictive	Unit
Min speed	6.2984	6.6531	rad/s
Max speed	7.6679	7.3014	rad/s
Mean speed	6.8541	6.7381	rad/s
MAE—Total error	0.1810	0.0449	rad/s
RMSE—Total error	0.2150	0.0595	rad/s
MAE—Overspeed	0.3545	0.2114	rad/s
RMSE—Overspeed	0.4638	0.2730	rad/s
Max overspeed	0.9679	0.6014	rad/s
Time in overspeed	36.13	36.35	%
Min pitch angle	2.0000	2.6772	deg
Max pitch angle	29.1172	13.2081	deg
Mean pitch angle	7.9340	7.6184	deg
Above 10°	32.13	33.46	%
Above 20°	11.44	8.30	%
Above 30°	0.00	0.00	%

Regarding actuator effort, both controllers operated within comparable ranges. The PI-Conventional produced pitch angles between 2° and 29.1°, while the PI-Predictive maintained a narrower interval between 2.68° and 13.21°. Although both controllers exhibited a similar average pitch angle (7.93° vs. 7.62°), the Predictive PI avoided large corrective actions, as reflected by the much lower fraction of time above 20° (8.30% vs. 11.44%). This indicates a moderate reduction in actuator workload under the predictive strategy, particularly during high-demand events.

The second scenario considers a regime of high and nearly stationary wind, in which the wind speed remains above the nominal value with minimal fluctuations. This case evaluates the stability of the predictive controller during sustained operation, ensuring that it does not introduce unnecessary control actions and that it maintains a smooth and consistent behavior under low aerodynamic variability. Figure 16 shows that, under these stable aerodynamic conditions, the predictive PI maintained the rotor speed within a narrower and more nominal-centered interval, while the conventional PI exhibited larger oscillations and recurrent overspeed events.

The quantitative analysis in Table 9 demonstrated a clear advantage of the predictive PI over the conventional PI. The traditional controller exhibited considerable dispersion around the nominal speed of 6.7 rad/s, with an RMSE of 0.215 rad/s, an MAE of 0.181 rad/s, and a maximum deviation of 1.3908 rad/s, confirming recurrent overspeed episodes. In contrast, the predictive scheme substantially reduced these values, achieving an RMSE of only 0.0595 rad/s and an MAE of 0.0449 rad/s, with a maximum deviation limited to 0.1582 rad/s.

When analyzing overspeed behavior, the conventional PI remained above the nominal velocity 91.06% of the time, reflecting a limited ability to suppress overspeed during high-wind operation. In contrast, the predictive PI reduced this percentage to 78.48%, confirming that the anticipatory term not only decreases the magnitude of the error but also reduces the persistence of overspeed.

Regarding pitch-actuator effort, the contrast between both controllers was even more pronounced. The conventional PI operated within a wide range (2° to 35°), with an average of 18.04°, indicating high control effort and frequent corrective actions. It remained above 10° for 85.95% of the time, demonstrating intensive actuator usage. In contrast, the predictive PI operated within a much narrower interval (2.68° to 13.21°), with a significantly lower

mean value of 7.62° and an RMS of 8.05°. The time spent above 10° was minimal (22.74%), reflecting a major reduction in actuator activity and, consequently, in mechanical wear.

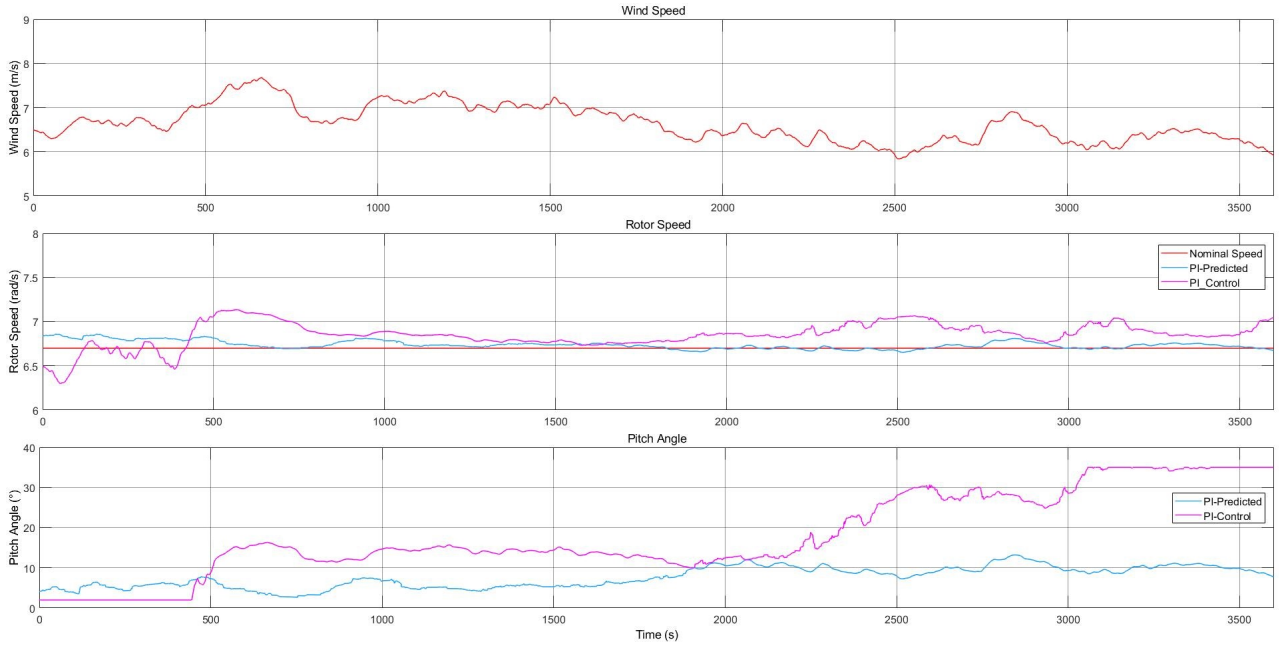


Figure 16. Time evolution of wind speed, rotor speed, and pitch angle under a high and nearly stationary wind regime. The plot highlights the differences between the conventional PI controller and the Predictive PI scheme under low-variability aerodynamic conditions.

Table 9. Performance metrics relative to nominal speed and pitch-actuator effort for the high and nearly stationary wind scenario.

Metric	PI-Conventional	PI-Predictive	Unit
Min speed	6.8856	6.6841	rad/s
Max speed	7.1679	6.8582	rad/s
Mean speed	6.3193	6.7109	rad/s
MAE—Total error	0.1810	0.0449	rad/s
RMSE—Total error	0.2150	0.0595	rad/s
MAE—Overspeed	0.1840	0.0529	rad/s
RMSE—Overspeed	0.2172	0.0664	rad/s
Max overspeed	1.1856	0.1582	rad/s
Time in overspeed	91.06	78.48	%
Min pitch angle	2.0000	2.6772	deg
Max pitch angle	35.0000	13.2081	deg
Mean pitch angle	18.0364	7.6184	deg
Above 10°	85.95	22.74	%
Above 20°	34.85	0.00	%
Above 30°	17.25	0.00	%

4. Discussion

This section analyzes the performance of the PI-Predictive scheme and its impact on dynamic stability, overspeed mitigation, and pitch-actuator efficiency, providing an integrated assessment of its behavior under different aerodynamic regimes.

The quality of the predictive model depended not only on its architecture but also on the preprocessing steps, temporal window construction, and selection of an appropriate historical horizon. Among the evaluated approaches, RF model offered the best balance between accuracy and generalization, outperforming both the linear (Ridge) and nonlinear

(MLP) alternatives. Its strong robustness makes it suitable for highly dynamic systems in which the relationships between variables evolve rapidly and in which prediction stability is as important as numerical accuracy.

Recurrent neural network architectures such as LSTM and GRU are widely adopted in time-series prediction; however, their advantages are more pronounced in long-horizon forecasting problems characterized by strong sequential dependencies. In the present study, the prediction task targets short-term rotor-speed forecasting under highly dynamic and noisy operating conditions, using a sliding-window representation rather than explicit sequence-to-sequence modeling. Under these circumstances, the RF model exhibits superior robustness and generalization capability, as it effectively captures nonlinear relationships while remaining less sensitive to noise, hyperparameter tuning, and dataset size. Furthermore, its ensemble-based structure provides stable training behavior and lower computational complexity, which are critical for real-time, control-oriented applications.

Although the coefficient of determination (R^2) obtained in the rotor-speed prediction task appears modest, this behavior is expected for short-term forecasting of highly dynamic and non-stationary signals. More importantly, the low RMSE and MAE values indicate that the prediction error remains within a range that is meaningful for anticipatory control, as confirmed by the improved closed-loop performance of the PI-Predictive con-troller reported in Table 6.

A historical window of 60 s proved to be the optimal compromise between temporal sensitivity and computational load. Shorter windows reduced contextual information and degraded overall performance, whereas longer windows substantially increased training time with minimal accuracy gains. The sensitivity analysis confirmed that the "memory" of the system is essential for anticipating rotor-speed variations.

Hyperparameter tuning via randomized search reduced model variance without significantly altering RMSE or MAE. This reinforces the notion that, in physical systems under noisy conditions, prediction consistency is more relevant than marginal numerical improvements. Moreover, the model exhibited stable behavior on the test set, indicating the absence of overfitting and confirming the robustness of the adopted training procedure.

When evaluated with the same wind profile used during model training, the predictive scheme demonstrated clear advantages over the conventional PI controller. The anticipatory approach reduced both the total error relative to the nominal speed and the magnitude of overspeed events.

The temporal anticipation analysis revealed that the PI-Predictive controller advanced the pitch-actuation command by an average of 7.27 s, with values reaching up to 20 s. This capability to act before the rotor dynamics fully unfold mitigated transient severity and reduced the need for abrupt corrections, one of the primary causes of oscillatory behavior in purely reactive control schemes.

The most demanding scenario involved high-gradient gusts combined with sub-nominal intervals. Under these conditions, both controllers deviated considerably from the nominal operating point due to the intense aerodynamic variability. Nevertheless, the PI-Predictive scheme consistently maintained smaller average deviations and significantly reduced overspeed magnitude.

Although transient events were unavoidable due to the physics of the wind field, the predictive controller attenuated their severity and reduced the reliance on strong pitch corrections. This behavior confirms that the added value of anticipation is most evident when the wind evolves faster than the mechanical response capabilities of the actuator.

In contrast, under a high but nearly stationary wind regime, the predictive scheme maintained the rotor speed within a narrower interval around the nominal value. The

reduction in peak error and overspeed persistence demonstrates that anticipatory action not only improves transient response but also enhances steady-state regulation.

A notable outcome was the substantial reduction in pitch-actuator effort. The conventional PI controller tended to operate for a large fraction of the time at elevated angles, whereas the predictive scheme kept the pitch within a much narrower range. This difference implies a markedly lower mechanical load on the actuator, with direct implications for component longevity and the energy efficiency of the control loop.

From a practical implementation perspective, it is important to distinguish between offline training and online execution of the proposed approach. While the RF predictor requires a non-negligible computational effort during the offline training stage, the re-al-time inference process is computationally lightweight, involving only the evaluation of a fixed ensemble of decision trees. As a result, both processing and memory requirements scale linearly with the number of estimators and remain compatible with low-cost embedded controllers typically used in small wind turbines. The expected inference latency is on the order of milliseconds, which is negligible relative to the typical pitch-control sampling periods used in small wind turbines (on the order of one second). This characteristic makes the proposed anticipatory scheme compatible with real-time implementation on industrial PLCs or embedded computing platforms commonly employed in tur-bine control systems, reinforcing its practical applicability beyond simulation-based vali-dation.

Despite the favorable results, several limitations must be acknowledged. The proposed framework relies on a control-oriented turbine model that adopts several simplifying assumptions, including simplified aerodynamics, rigid-blade mechanics, and a first-order pitch-actuator model. These assumptions are introduced to reduce computational complexity and enable extensive simulation campaigns under multiple wind scenarios. As a consequence, higher-order actuator nonlinearities, mechanical backlash, and degradation effects are implicitly neglected within this modeling abstraction.

Although such simplifications may limit absolute aerodynamic load accuracy, their influence on the prediction–control coupling examined in this study is expected to be secondary, as the anticipatory strategy primarily exploits short-term temporal correlations between wind excitation and rotor-speed evolution, which are preserved under consistent modeling assumptions. Consequently, the observed improvements in overspeed mitigation, control responsiveness, and actuator effort are attributed to the anticipatory feedback mechanism rather than to high-fidelity aerodynamic effects.

The predictor relies solely on time-series measurements of wind speed and rotor speed, without incorporating spatial, directional, or turbulence-related information. This restricts its ability to generalize under more complex atmospheric conditions. Furthermore, both the prediction horizon (20 s) and historical window (60 s) were fixed, which may not be optimal for all wind regimes.

From an experimental standpoint, the validation scenarios, though representative, do not capture the full diversity of wind patterns observed in real installations. The dynamic turbine model also excludes several relevant physical phenomena, such as actuator delays, mechanical backlash, sensor noise, and structural flexibility, all of which may influence real-world performance.

It should be noted that the wind and turbine-operation data used in this study correspond to a single geographic site and a specific turbine configuration. Since the predicted variable is the rotor speed, direct transfer of the trained model to other sites or turbines with different aerodynamic and inertial characteristics would not be physically consistent. Nevertheless, the proposed methodology constitutes a robust anticipatory control tool that can be effectively applied to other sites and turbine design once representative local data are available for model training.

Finally, the PI-Predictive controller remains a linear anticipatory scheme without the formal stability guarantees provided by more advanced predictive or robust control frameworks (MPC or nonlinear robust control). These limitations open clear opportunities for future work, including the incorporation of more informative aerodynamic representations, understood not as high-fidelity CFD models but as control-oriented models that provide additional information on the aerodynamic sensitivity of the system. Such representations could enable adaptive or gain-scheduled control strategies, allowing the controller parameters to be adjusted online according to operating conditions and potentially improving system robustness without compromising real-time feasibility.

5. Conclusions

This study presented a hybrid control approach for small wind turbines, combining a conventional PI loop with a predictive model trained using machine learning. The model, based on a Random Forest architecture and optimized through randomized hyperparameter search, demonstrated the ability to anticipate rotor speed with a 20 s prediction horizon using a 60 s historical window, established as the optimal compromise between temporal context and computational efficiency.

The results showed that the Predictive PI controller improves rotor-speed regulation under both seen and unseen conditions, reducing the magnitude and persistence of overspeed events. Furthermore, the pitch actuator operates with lower effort, avoiding extreme corrective actions and smoothing transitional dynamics. These improvements were especially evident in high-intensity, quasi-stationary wind scenarios, where the system-maintained rotor speed within a narrow interval around the nominal value.

Quantitatively, the proposed anticipatory strategy reduced overspeed peaks by up to 35–45%, decreased the integral absolute error (IAE) of rotor speed by approximately 30%, and lowered pitch-actuator RMS activity by about 25% compared with the conventional PI controller. These improvements were consistently observed across all evaluated wind scenarios, including previously unseen conditions characterized by steep gust gradients and high wind variability.

Under wind conditions characterized by abrupt gradients, the predictive scheme reduced the severity of disturbances and advanced actuator action by several seconds, mitigating the impact of rapid gusts. Such anticipatory capability constitutes a significant advantage over purely reactive controllers, particularly in turbines whose pitch mechanisms exhibit non-negligible mechanical delays.

These results validate the proposed methodology by demonstrating that short-term, data-driven rotor-speed prediction can be effectively integrated into a conventional PI framework to introduce anticipatory behavior without increasing control complexity. The observed performance gains arise from anticipation rather than aggressive control action, confirming the robustness and practical relevance of the approach for small wind turbines operating under highly turbulent wind conditions.

Overall, the findings indicate that integrating AI-based predictive models can strengthen dynamic stability, enhance operational safety, and reduce mechanical wear on the pitch system. Future work could explore adaptive prediction horizons, integration with model-based predictive control (MPC), the incorporation of lidar measurements, or hybrid approaches that combine data-driven prediction with physics-based modeling to improve robustness under highly turbulent atmospheric conditions.

Author Contributions: Conceptualization, E.C.-N. and J.C.J.-C.; methodology, E.C.-N.; system control, E.C.-N. and R.V.C.-S.; experiments, E.C.-N. and R.V.C.-S.; software, R.V.C.-S.; hardware, J.C.J.-C.; validation, J.G.R.-M.; formal analysis, M.T.-P.; investigation, E.C.-N.; resources, J.C.J.-C.; data curation, J.G.R.-M.; discussion, M.T.-P. and J.C.J.-C.; writing—original draft preparation, E.C.-N.;

writing—review and editing, M.T.-P. and J.G.R.-M.; visualization, E.C.-N.; supervision, J.C.J.-C.; project administration, J.C.J.-C.; funding acquisition, J.C.J.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in this article; further inquiries can be directed to the corresponding author.

Acknowledgments: This work was supported by Investigación y Posgrado of the Facultad de Ingeniería and Universidad Autónoma de Querétaro.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. U.S. Department of Energy (DOE). National Transmission Needs Study. 30 October 2023. Available online: https://www.energy.gov/sites/default/files/2023-12/National%20Transmission%20Needs%20Study%20-%20Final_2023.12.1.pdf (accessed on 29 September 2025).
2. Sheridan, L.; Kazimierczuk, K.; Garbe, J.; Preziuso, D. *Distributed Wind Market Report: 2024 Edition*; Pacific Northwest National Laboratory (PNNL): Richland, WA, USA, 2024. Available online: https://www.pnnl.gov/main/publications/external/technical_reports/PNNL-36057.pdf (accessed on 30 September 2025).
3. IEA Wind TCP Task 27. *Small Wind Turbines in High Turbulence Sites—Draft Small Wind Turbine Technical Report*; IEA Wind TCP: Paris, France, 2018. Available online: https://iea-wind.org/wp-content/uploads/2021/06/IEA-Wind-TCP-Task-27-Draft-Small-Wind-Turbine-Technical-Report.pdf (accessed on 30 September 2025).
4. Chu, R.; Wang, K. CFD in Urban Wind Resource Assessments: A Review. *Energies* **2025**, *18*, 2626. [CrossRef]
5. Chavero-Navarrete, E.; Trejo-Perea, M.; Jáuregui-Correa, J.C.; Carrillo-Serrano, R.V.; Ronquillo-Lomeli, G.; Ríos-Moreno, J.G. Pitch Angle Optimization for Small Wind Turbines Based on a Hierarchical Fuzzy-PID Controller and Anticipated Wind Speed Measurement. *Appl. Sci.* **2021**, *11*, 1683. [CrossRef]
6. Chavero-Navarrete, E.; Trejo-Perea, M.; Jáuregui-Correa, J.C.; Carrillo-Serrano, R.V.; Ronquillo-Lomeli, G.; Ríos-Moreno, J.G. Hierarchical Pitch Control for Small Wind Turbines Based on Fuzzy Logic and Anticipated Wind Speed Measurement. *Appl. Sci.* **2020**, *10*, 4592. [CrossRef]
7. Lu, P.; Zhang, N.; Ye, L.; Du, E.; Kang, C. Advances in Model Predictive Control for Large-Scale Wind Power Integration in Power Systems. *Adv. Appl. Energy* **2024**, *14*, 100177. [CrossRef]
8. Tang, M.; Wang, W.; Yan, Y.; Zhang, Y.; An, B. Robust Model Predictive Control of Wind Turbines Based on Bayesian Parameter Self-Optimization. *Front. Energy Res.* **2023**, *11*, 1306167. [CrossRef]
9. Mustafa, M.; Mohammed, S. Adaptive Hybrid Pitch Angle Control and MPPT for PMSG-Based Wind Power Generator Systems. *Int. J. Energy Convers.* **2022**, *10*, 153. [CrossRef]
10. Karami-Mollaee, A.; Shojaei, A.A.; Barambones, O.; Othman, M.F. Dynamic Sliding Mode Control of Pitch Blade Wind Turbine Using Sliding Mode Observer. *Trans. Inst. Meas. Control.* **2022**, *44*, 3028–3038. [CrossRef]
11. Karami-Mollaee, A.; Barambones, O. Pitch Control of Wind Turbine Blades Using Fractional Particle Swarm Optimization. *Axioms* **2023**, *12*, 25. [CrossRef]
12. Jiao, X.; Wang, G.; Wang, X.; Zhang, Z.; Tian, Y.; Fan, X. Anti-Windup Pitch Angle Control for Wind Turbines Based on Bounded Uncertainty and Disturbance Estimator. *J. Mar. Sci. Eng.* **2024**, *12*, 473. [CrossRef]
13. Abdallah, A.; William, M.A.; Moharram, N.A.; Zidane, I.F. Boosting H-Darrieus Vertical Axis Wind Turbine Performance: A CFD Investigation of J-Blade Aerodynamics. *Results Eng.* **2025**, *27*, 106358. [CrossRef]
14. Adnan, Q.A.; Hussain, M.K.; Mohammadzadeh, A.; Sabahi, K. Optimal Hybrid Type-3 Fuzzy Controller for Horizontal Axis Wind Turbines: Comparative Study. *ISA Trans.* **2025**, *161*, 200–215. [CrossRef]
15. Qin, S.; Ngu, S.S.; Zhang, C.; Cai, H.; Chen, Y.; Chen, R.; Liu, T. H-Infinity Variable-Pitch Control for Wind Turbines Based on Takagi–Sugeno Fuzzy Theory. *Tech. Vjesn.* **2023**, *30*, 1387–1396. [CrossRef]
16. Sathyan, A.; Eisa, S.; Cohen, K. Can Physically-Trained Genetic Fuzzy Learning Algorithm Improve Pitch Control in Wind Turbines? In *Advances in Artificial Intelligence and Applied Cognitive Computing*; Springer: Cham, Switzerland, 2022; pp. 325–337. [CrossRef]
17. Asghar, A.B.; Ehsan, R.; Naveed, K.; Al-Ammar, E.A.; Ejsmont, K.; Nejman, M. Recurrent Neural Network for Pitch Control of Variable-Speed Wind Turbine. *Sci. Prog.* **2024**, *107*, 00368504241243160. [CrossRef]
18. Erol, H.; Arslan, A. Wind Turbine Pitch Angle Control with Artificial Neural Networks. *Rev. Roum. Sci. Tech. Ser. Électrotech. Énerg.* **2025**, *70*, 235–240. [CrossRef]

19. Ibrahim, M.A.; Alhfidh, A.S.; Hamoodi, A.N. Performance Enhancement of Small-Scale Wind Turbine Based on Artificial Neural Network. *Int. J. Power Electron. Drive Syst.* **2023**, *14*, 1722–1730. [CrossRef]
20. Sierra-García, J.E.; Santos, M. Deep Learning and Fuzzy Logic to Implement a Hybrid Wind Turbine Pitch Control. *Neural Comput. Appl.* **2022**, *34*, 10503–10517. [CrossRef]
21. Reddy, Y.S.; Hur, S. Wind Turbine Gust Control Using LIDAR-Assisted Model Predictive Control. In Proceedings of the ASME 2023 5th International Offshore Wind Technical Conference, Exeter, UK, 18–19 December 2023; ASME: New York, NY, USA, 2023; p. V001T01A030. [CrossRef]
22. Fu, W.; Guo, F.; Schlipf, D.; Peña, A. Feedforward Pitch Control for a 15 MW Wind Turbine Using a Spinner-Mounted Single-Beam LiDAR. *Wind Energy Sci.* **2023**, *8*, 1893–1911. [CrossRef]
23. Russell, A.J.; Collu, M.; McDonald, A.S.; Thies, P.R.; Keane, A.; Quayle, A.R. LiDAR-Assisted Feedforward Individual Pitch Control of a 15 MW (Floating) Offshore Wind Turbine. *Wind Energy* **2024**, *27*, 341–362. [CrossRef]
24. Sivhugwana, K.S.; Ranganai, E. Short-Term Wind Speed Prediction via Sample Entropy: A Hybridisation Approach against Gradient Disappearance and Explosion. *Computation* **2024**, *12*, 163. [CrossRef]
25. Li, T.; Yang, J.; Ioannou, A. Wind forecasting-based model predictive control of generator, pitch, and yaw for output stabilisation—A 15-megawatt offshore. *Energy Convers. Manag.* **2024**, *302*, 118155. [CrossRef]
26. Sun, P.; Wang, J.; Yan, Z. Ultra-short-term wind speed prediction based on TCN-MCM-EKF. *Energy Rep.* **2024**, *11*, 2127–2140. [CrossRef]
27. Meng, Y.; Chang, C.; Huo, J.; Zhang, Y.; Al-Neshmi, H.M.M.; Xu, J.; Xie, T. Research on ultra-short-term prediction model of wind power based on attention mechanism and CNN-BiGRU combined. *Front. Energy Res.* **2022**, *10*, 920835. [CrossRef]
28. Yu, G.; Shen, L.; Dong, Q.; Cui, G.; Wang, S.; Xin, D.; Chen, X.; Lu, W. Ultra-short-term wind power forecasting techniques: Comparative analysis and future trends. *Front. Energy Res.* **2024**, *11*, 1345004. [CrossRef]
29. Chavero-Navarrete, E.; Trejo-Perea, M.; Jáuregui-Correa, J.-C.; Carrillo-Serrano, R.-V.; Rios-Moreno, J.-G. Pitch Angle Optimization by Intelligent Adjusting the Gains of a PI Controller for Small Wind Turbines in Areas with Drastic Wind Speed Changes. *Sustainability* **2019**, *11*, 6670. [CrossRef]
30. Borrohou, S.; Fissoune, R.; Badir, H. Data Cleaning Survey and Challenges—Improving Outlier Detection Algorithm in Machine Learning. *J. Smart Cities Soc.* **2023**, *2*, 125–140. [CrossRef]
31. Niedźwiecki, M.; Ciołek, M.; Gańcza, A.; Kaczmarek, P. Application of Regularized Savitzky–Golay Filters to Identification of Time-Varying Systems. *Automatica* **2021**, *133*, 109865. [CrossRef]
32. Matrenin, P.V.; Harlashkin, D.A.; Mazunina, M.V.; Khalyasmaa, A.I. Investigation of the Features Influencing the Accuracy of Wind Turbine Power Calculation at Short-Term Intervals. *Appl. Syst. Innov.* **2024**, *7*, 105. [CrossRef]
33. de Amorim, L.B.V.; Cavalcanti, G.D.C.; Cruz, R.M.O. The Choice of Scaling Technique Matters for Classification Performance. *Appl. Soft Comput.* **2023**, *133*, 109924. [CrossRef]
34. Hahn, Y.; Langer, T.; Meyes, R.; Meisen, T. Time Series Dataset Survey for Forecasting with Deep Learning. *Forecasting* **2023**, *5*, 315–335. [CrossRef]
35. Sekeroglu, B.; Ever, Y.K.; Dimililer, K.; Al-Turjman, F. Comparative Evaluation and Comprehensive Analysis of Machine Learning Models for Regression Problems. *Data Intell.* **2022**, *4*, 620–652. [CrossRef]
36. Rajan, M.P. An Efficient Ridge Regression Algorithm with Parameter Estimation for Data Analysis in Machine Learning. *SN Comput. Sci.* **2022**, *3*, 171. [CrossRef]
37. Iranzad, R.; Liu, X. A Review of Random Forest-Based Feature Selection Methods for Data Science Education and Applications. *Int. J. Data Sci. Anal.* **2025**, *20*, 197–211. [CrossRef]
38. Kruse, R.; Mostaghim, S.; Borgelt, C.; Braune, C.; Steinbrecher, M. Multi-Layer Perceptrons. In *Computational Intelligence*; Texts in Computer Science; Springer: Cham, Switzerland, 2022. [CrossRef]
39. Van Rossum, G.; Drake, F.L. *Python 3 Reference Manual*; CreateSpace: Scotts Valley, CA, USA, 2009. Available online: https://www.python.org/ (accessed on 20 November 2024).
40. MathWorks. *MATLAB*, Version 9.5.0.944444 (R2018b); The MathWorks Inc.: Natick, MA, USA, 2018.
41. Hewlett-Packard, Palo Alto, CA, USA. 2018. Available online: https://support.hp.com/za-en/drivers/hp-zbook-15-mobile-workstation/5387800 (accessed on 20 November 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.