Machine learning-based wind turbine control systems for demand-oriented scenarios

Tenghui Li

Submitted in fulfilment of the requirements for the

Degree of Doctor of Philosophy

James Watt School of Engineering

College of Science and Engineering

University of Glasgow

Abstract

When wind power has an increasing share towards a 100% renewable society, wind energy conversion systems (WECSs) need to consider a requirement of the grid generation-consumption equilibrium, i.e., wind turbines (WTs) should be able to adjust their output according to power demand. However, current WTs focus on maximum power capture, which has intrinsic disadvantages in power scheduling. Hence, this study aims for machine learning (ML) based control systems that realize flexible wind capture in demand-oriented scenarios.

First, this study reviews various turbine components and establishes corresponding control models. Second, aerodynamic modelling relies on an artificial neural network (ANN) to predict thrust, torque, and power from the turbine state. Subsequently, a novel online power strategy (OPS) based on an aerodynamic model solves the 2-degree-of-freedom (DOF) optimization of the rotor speed control (RSC) and pitch angle control (PAC), which has two implementations: power reference point tracking (PRPT) and reinforcement learning (RL). Besides, the OPS has a local linearization to estimate thrust and torque sensitivities for optimal control configuration. When a wind processing unit updates wind velocity and direction signals, the OPS receives the velocity signal to calculate the 2-DOF solution and unwraps the direction signal as the command of the yaw angle control (YAC), which achieves a complete 3-DOF regulation of rotation, pitch, and yaw. Besides, the OPS framework has four control implementations: one model-free controller and three model-based controllers. In addition, our turbine control system can integrate wind forecasting to enhance the capability of handling wind stochastics.

The case study verifies and proves the accuracy and reliability of the OPS-based control framework in four simulation cases. The proposed turbine control can track different power targets and ensure reliable output in stochastic winds. Therefore, this control framework can contribute to intelligent WTs for large-scale grid integration.

Table of Contents

List of Tables

List of Figures

List of Publications

Part of this research has been accepted and published in the following journal articles.

1. **Li, Tenghui**, Jin Yang, and Anastasia Ioannou. "Data-Driven Control of Wind Turbine under Online Power Strategy via Deep Learning and Reinforcement Learning." *Renewable Energy* 234 (August 2024): 121265. https://doi.org/10.1016/j.renene.2024.121265. (Impact Factor: 9.0, Quartile: Q1)
2. **Li, Tenghui**, Jin Yang, and Anastasia Ioannou. "Wind Forecasting-Based Model Predictive Control of Generator, Pitch, and Yaw for Output Stabilisation – a 15-Megawatt Offshore." *Energy Conversion and Management* 302 (February 2024): 118155. https://doi.org/10.1016/j.enconman.2024.118155. (Impact Factor: 9.9, Quartile: Q1)
3. **Li, Tenghui**, Xiaolei Liu, Zi Lin, Jin Yang, and Anastasia Ioannou. "A Linear Quadratic Regulator with Integral Action of Wind Turbine Based on Aerodynamics Forecasting for Variable Power Production." *Renewable Energy* 223 (March 2024): 119605. https://doi.org/10.1016/j.renene.2023.119605. (Impact Factor: 9.0, Quartile: Q1)
4. **Li, Tenghui**, Xiaolei Liu, Zi Lin, and Rory Morrison. "Ensemble Offshore Wind Turbine Power Curve Modelling – an Integration of Isolation Forest, Fast Radial Basis Function Neural Network, and Metaheuristic Algorithm." *Energy* 239 (January 2022): 122340. https://doi.org/10.1016/j.energy.2021.122340. (Impact Factor: 8.9, Quartile: Q1)

Declaration

I certify that the thesis presented here is the result of my own work, except where I clearly indicated that it is the work of others.

I declare that this thesis does not include work forming part of a thesis presented successfully for another degree.

Abbreviation

AC	alternating-current
Adadelta	adaptive learning rate method
Adagrad	adaptive gradient algorithm
Adam	adaptive moment estimation
ADMM	alternating direction method of multipliers
AHSE	aero-hydro-servo-elastic
ANFIS	adaptive neuro-fuzzy inference system
ANN	artificial neural network
APC	adaptive predictive control
API	application programming interface
BEM	blade element momentum
CDQN/C51	categorical deep Q-network
CFD	computational fluid dynamics
CNN	convolutional neural network
CVXOPT	convex optimization solver
DC	direct-current
DFIG	doubly fed induction generator
DL	deep learning
DNN	deep neural network
DOF	degree-of-freedom
DOWEC	Dutch Offshore Wind Energy Converter
DQN	deep Q-network
DTU	Technical University of Denmark
EKF	extended Kalman filter
ELM	extreme learning machine
FAST	fatigue, aerodynamics, structures, and turbulence
FCM	fuzzy c-means
FIFO	first-in-first-out
FILO	first-in-last-out
FLC	fuzzy logic control
Ftrl	follow the regularized leader
GPU	graphics processing unit
GW	gigawatt
GWh	gigawatt hour
HAWT	horizontal axis wind turbine
HDNN	hybrid deep neural network
IEA	International Energy Agency
iForest	isolation forest
IO	input-output
LQR	linear quadratic regulator
LSTM	long short-term memory
MAE	median absolute error
MDP	Markov decision process

MIMO	multiple-input-multiple-output
MISO	multiple-input-single-output
ML	machine learning
MLPN	multilayer perceptron network
MPC	model predictive control
MPS	maximum power strategy
MW	megawatt
Nadam	Nesterov-accelerated adaptive moment estimation
NP	polynomial-time
NREL	National Renewable Energy Laboratory
NSFM	nonsymmetric fuzzy means
NWP	numerical weather prediction
ODE	ordinary differential equation
OPS	online power strategy
OSQP	operator splitting solver for quadratic programs
PAC	pitch angle control
PID	proportional-integral-derivative
PLL	phase lock loop
PMSG	permanent magnet synchronous generator
ProxQP	proximal quadratic programming solver
PRPT	power reference point tracking
PSO	particle swarm optimization
QP	quadratic programming
QuadProg	quadratic programming solver
R^2	r-square
RBFN	radial basis function network
ReLU	rectified linear unit
RHC	receding horizon control
RL	reinforcement learning
RMSE	root mean squared error
RMSprop	root mean squared propagation
RNN	recurrent neural network
RSC	rotor speed control
SCS	splitting conic solver
SGD	stochastic gradient descent
TCP	Technology Collaboration Programme
TS	tabu search
VAWT	vertical axis wind turbine
VSC	voltage source converter
WECS	wind energy conversion system
WT	wind turbine
WTPC	wind turbine power curve
YAC	yaw angle control

Chapter 1 Introduction

The global wind industry is approaching the 1000 gigawatt (GW) milestone in cumulative install capacity with a year growth rate of 9% [1]. The International Energy Agency (IEA) Wind Technology Collaboration Programme (TCP) released the 2022 annual report of its member countries in October 2023 [1]. **Table 1.1** records the total installed wind power capacity by the end of 2022 in the IEA Wind TCP countries [1]. China kept on leading with the most noticeable deployment, close to reaching 400 GW of installed capacity (365 GW grid-connected) [1]. Europe had a significant increase in new wind installations with 19 GW and exceeded 250 GW, which ensured the progress of the clean energy targets set for 2030 and 2050 [1].

Table 1.1 Installed wind capacity in the IEA Wind TCP countries

Country	2021 Capacity (GW)	2022 Capacity (GW)	Country	2021 Capacity (GW)	2022 Capacity (GW)
China	346.7	387.2	Finland	3.6	5.7
USA	135	144.2	Portugal	5.6	5.7
Germany	63.9	66.3	Belgium	4.7	5.2
Spain	28.2	29.8	Norway	4.6	5.1
UK	25.7	28.7	Japan	4.6	4.8
France	18.9	20.9	Greece	4.5	4.7
Canada	14.3	15.3	Ireland	4.1	4.6
Sweden	12.1	14.3	Austria	3.4	3.6
Italy	11.1	11.6	South Korea	1.7	1.8
Netherlands	7.8	8.7	Switzerland	0.1	0.1
Denmark	6.9	7			

Figure 1.1 compares the installed wind capacity growth rate in 2022, in which nine countries exceeded 10% of annual growth. Finland surpassed other countries with an incredible increase of 58%. Offshore installations accounted for the UK’s added capacity (3 GW), indicating that offshore deployment has become a new trend. In addition, the UK built the first commercial floating wind farm (Hywind Scotland, **Figure 1.2**), which delivered electricity to the Scottish grid in 2017 [2].

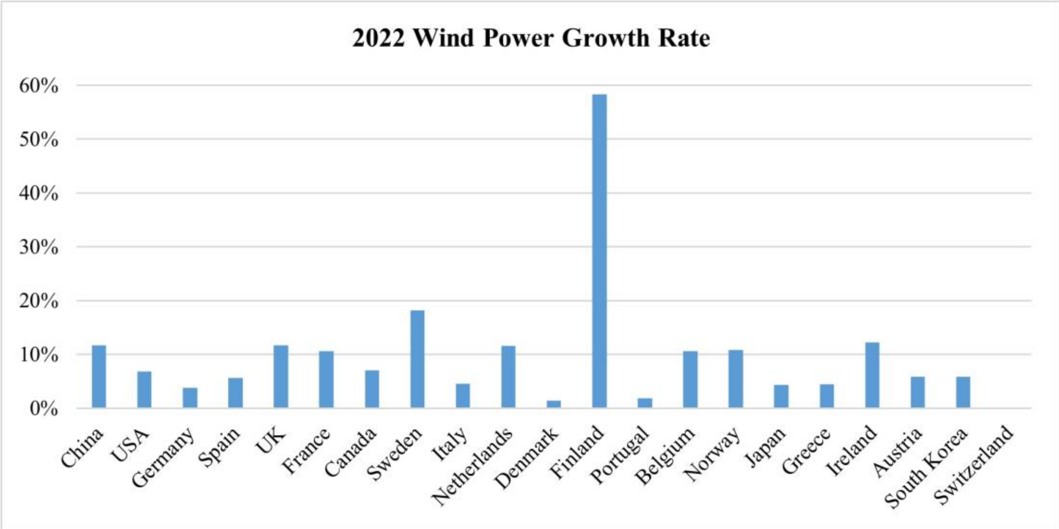

Figure 1.1 Comparison of the 2022 growth rate of the IEA Wind TCP countries

According to the latest data from Wind Europe, onshore supplies 1,709 GWh of daily wind generation and offshore contributes to 523 GWh [3]. Total wind energy ensures an electricity demand of 29.8% in Europe [3]. Wind energy has proven promising due to its continuous high deployment growth.

Figure 1.2 Hywind Scotland floating wind farm
(photo: Øyvind Gravås / Woldcam - Statoil ASA, source: Ref. [2])

With the expansion of wind energy worldwide, the capacity and size of a wind turbine (WT) have evolved for megawatt (MW) generation, even 10 MW or higher. For example, the average power rating of onshore WTs increased from 4 MW (2018) to 5.1 MW (2022), while offshore WTs rose to an average of 12.2 MW from 8 MW [1]. Especially offshore WTs have a range of rated power from 8 MW to 20 MW, in which the share of 15~20 MW has reached 38% [4]. Besides, rotor diameter has risen from 20 m to 160 m since 1980 [1]. Hence, WTs tend to be high power and move into the deep ocean.

Many countries devote themselves to a society of 100% renewable energy resources, which gives a particular role to wind power [5]. However, present turbine control systems cannot support this goal for the following reasons. Firstly, the aerodynamic results of a rotor disk (or propeller) lead to complexities and nonlinearities in the operation optimization of a WT [6]. Most WTs follow a conventional way to capture wind power as much as possible [7], which makes power scheduling extremely hard. On the other hand, current WTs cannot regulate their output power according to power demand. This study will investigate a novel method to achieve flexible power production rather than maximum. Secondly, large turbine size brings more efficient wind capture but challenges control design for stable and reliable power quality [8]. Therefore, this inspires us to develop more robust control systems for WTs to handle power fluctuation. Thirdly, wind sources have natural uncertainties (in velocity, direction, and shear) that cause unstable wind flow [9], so wind generation is less controllable than steam engines. Hence, foreseen wind information is necessary for a turbine system to correct its control policy in advance. With the motivation of flexible wind power, this study aims to combine cutting-edge machine learning (ML) technologies and classic control engineering theories. This research only involves software-level algorithms but does not change mechanical or structural designs. The main objectives of this study include:

- re-design the power strategy for turbine operation to suit demand-oriented power scenarios;
- develop more robust and intelligent turbine control systems to be compatible with the novel strategy;
- integrate wind forecasting to improve the performance of handling wind stochastics.

1.1 Modern Wind Turbines

To understand what causes the limitations of WTs, we first review the evolution of WTs. Modern WTs commonly seen onshore or offshore apply a primary type, namely horizontal axis wind turbine (HAWT), driven by lift forces [10]. In contrast to the HAWT, another type is vertical axis wind turbine (VAWT), where the main shaft is transverse to wind flow while other components are close to the ground base [11]. **Figure 1.3** compares HAWT and VAWT structures [12]. The VAWT has a simple structure and is insensitive to wind direction but cannot meet the same degree of success as the HAWT. The main factor is energy efficiency because power capture is proportional to rotor area [13]. Therefore, HAWTs will remain mainstream, and this study concentrates on this type (the term 'WT' points to HAWT in the following contents).

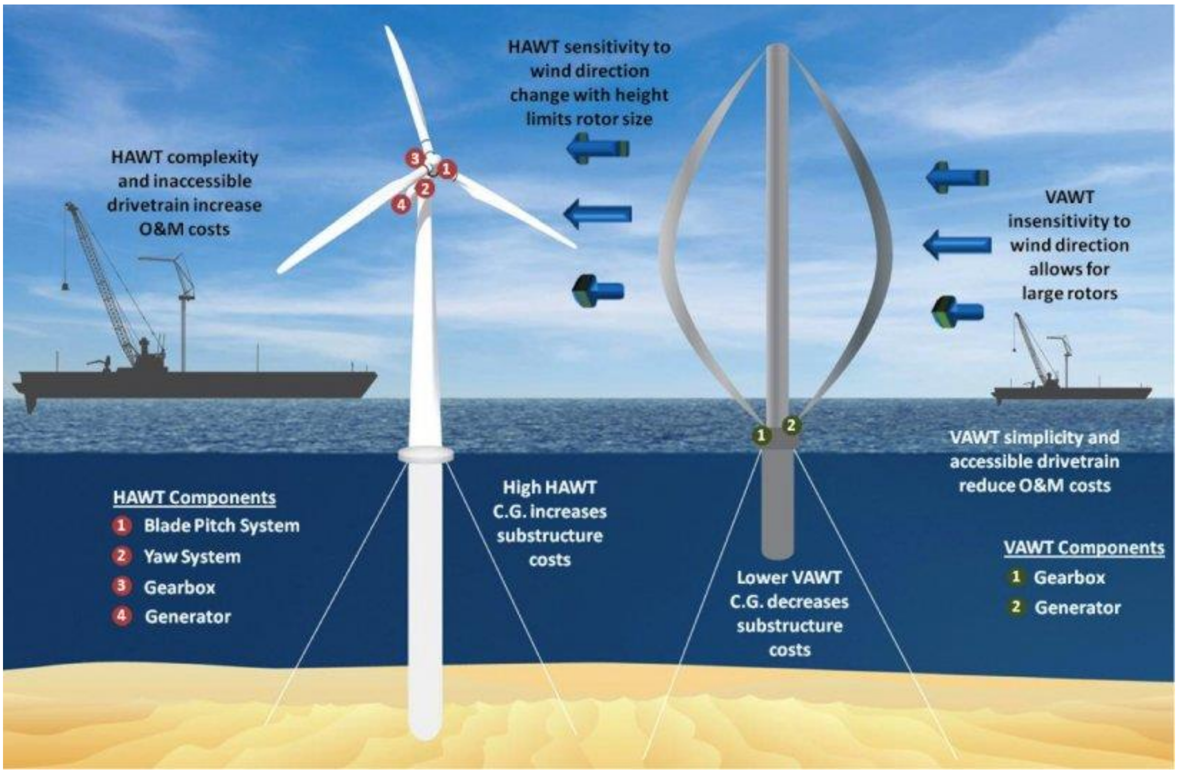

Figure 1.3 Comparison between horizontal and vertical layouts (source: Ref. [10])

Classification of modern turbines (**Figure 1.4**) usually relates to the number of blades (two, three, or more), blade regulation (stall or pitch), rotor orientation (upwind or downwind), drivetrain (gearbox or direct drive), and generator type (doubly fed induction generator or permanent magnet synchronous generator, abbreviated as DFIG or PMSG) [10]. The number of blades affects the capability of wind capture and depends on the mechanical and aerodynamic characteristics of a prototype design [14]. As the fluid medium is air, most WTs are three-bladed due to rotation balance and aerodynamic economy [14]. However, a control system has fewer concerns about specific blades because a control design usually requires an approximate turbine model that mathematically reflects most aerodynamic factors.

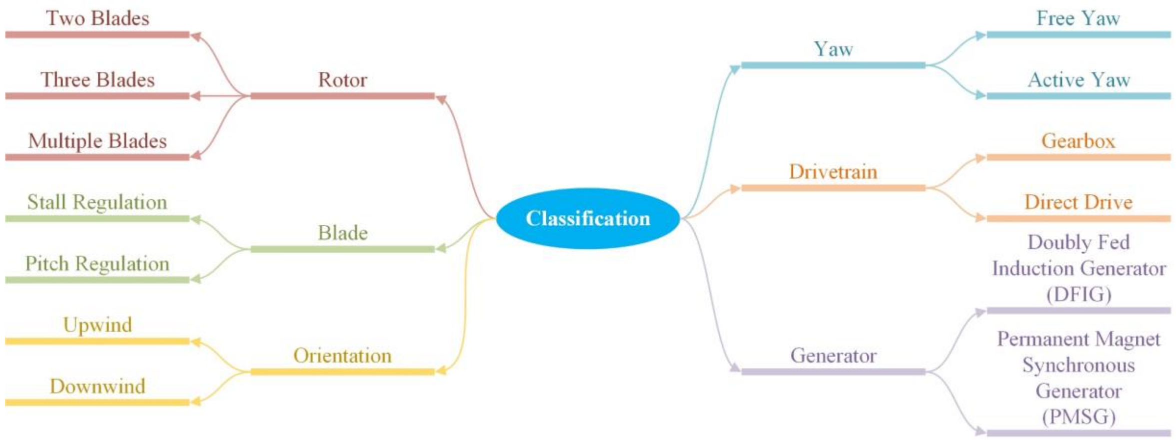

Figure 1.4 Six classifications of modern turbines

Old-type WTs rely on stall regulation to protect the system in heavy winds, which leads to an output curve of sudden drop after cut-off and forms a narrow generation region [10]. Modern WTs adopt pitch regulation for a wide output range, which offers more variable and adjustable aerodynamics [14]. Besides, pitch regulation is necessary for long blades due to economic benefits [7]. Hence, this study focuses on pitch-regulated WTs.

Rotor orientation can be upwind or downwind relative to tower position, as shown in **Figure 1.5** [15]. First, two configurations affect aerodynamic efficiency, where a downwind WT is lower since the nacelle and tower obstruct part of the airflow [16]. The second effect is related to the yaw system, i.e., how to navigate the nacelle direction. Upwind WTs must be active yaw, but downwind WTs can be self-align (free yaw) or active [10]. If upwind and downwind WTs share the same blade, there will be no difference to the control system due to the same descriptive model. The control system can add a constant of energy transfer efficiency to correct the influence of the blocked airflow. However, highly efficient WTs must consider yaw orientation because this affects the effective windward area [7]. Although downwind WTs are compatible with free-yaw systems, turbulences or stochastic flows may lead to short-term deviation at optimal nacelle position [14]. Therefore, the active yaw being conducive to alignment with wind direction will be one of the research objectives.

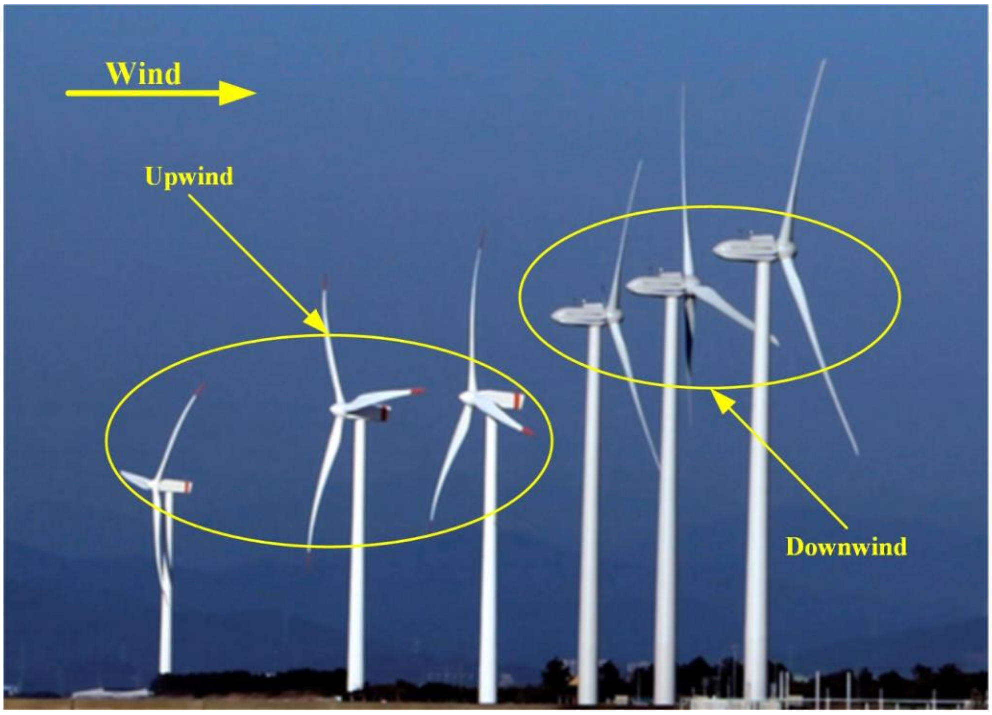

Figure 1.5 Upwind and downwind rotor orientations (photo: Toru Nagao, source: Ref. [15])

Early WTs directly adopted general-purpose generators, which required a transmission system to match the range of generator rotation with the range of rotor speed [17]. On the other hand, the generator operation should meet the requirement of the grid frequency, which is faster than normal rotor rotation due to mechanical and aerodynamic constraints [18]. This type relies on a gearbox to achieve speed and torque transmission. Recent advanced WTs apply a direct-drive design that combines the main shaft and the generator rotor [7]. The direct-drive turbine contributes to a more concise wind energy conversion system (WECS) and a more compact nacelle [8]. A gearbox-based turbine has a high gear ratio and fewer generator poles, while a direct-drive turbine commonly has a unit gear ratio and more poles. The essence is that a direct-drive generator increases its poles to match low rotation. Owing to both enormous market shares, this study will investigate both geared and direc-drive drivetrains to cover a variety of WTs.

Among various generators, the DFIG and PMSG are two of the most popular types. The DFIG relies on a back-to-back converter to achieve magnetizing at the rotor side of the generator, and its stator side directly outputs absorbed power [19]. The PMSG has a permanent magnet rotor for magnetizing, and its stator side places a back-to-back converter to change the electric frequency of output power [20]. The converter size of the DFIG is smaller than the PMSG since its magnetizing requires lower currents [21]. However, the PMSG is more suitable for high-power scenarios because of high torque endurance and no rotor loss. Since both types have widespread applications, this study will discuss their control systems separately.

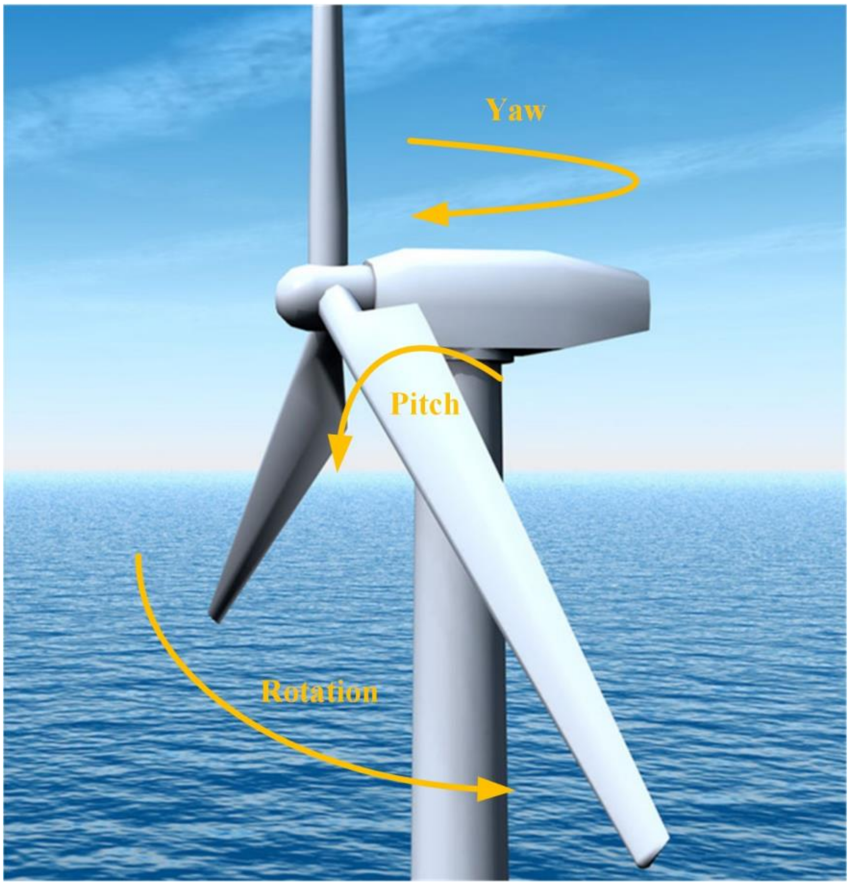

Figure 1.6 Sketch of the 3-DOF regulation of rotation, pitch, and yaw (source: Ref. [22])

According to the above analysis, modern WTs favour a three-degree-of-freedom (3-DOF) design, i.e., shaft rotation, pitch angle, and yaw angle. As a result, a turbine control system has three parts to realize the 3-DOF regulation (**Figure 1.6**), including rotor speed control

(RSC) [23], pitch angle control (PAC) [24], and yaw angle control (YAC) [17]. This study intends to coordinate the RSC, PAC, and YAC through an intelligent control system for flexible and reliable wind generation.

1.2 Control Review

As the most crucial part of a turbine control system, the RSC receives a rotor speed reference (speed reference) to manage generator torque, which aims to drive the main shaft at a nominated speed [23]. Meanwhile, the generator absorbs the kinetic energy of the shaft and outputs electric power. The PAC primarily tracks a pitch angle reference (pitch reference) if under collective blade policy, which utilizes the aerodynamic response of pitching blades to affect power capture [25]. Therefore, the PAC result affects the RSC process, and most turbine systems aim for better coordination of the RSC and PAC. Also, WTs equipped with the RSC and PAC are called variable speed and pitch-regulated, which refers to a 2-DOF regulation of speed and pitch [17]. By contrast, the YAC is more straightforward because it mainly depends on a forecasted wind direction (yaw reference) to navigate nacelle yaw [7].

The research community has put much effort into advanced turbine control systems. Gambier and Meng designed an integrated proportional integral derivative (PID) loop design for a 20-MW system considering torque control, pitch control, and fore-aft damping [26]. Junejo et al. introduced a physic-informed optimization into the PID of a VAWT, which improved power efficiency, disturbance suppression, and output stability [27]. Tu et al. coordinated the RSC and PAC, where the RSC command resulted from a reduced-load power curve and a frequency response signal, and the main objective of the PAC was to maintain rotor speed [28]. Lara et al. applied a torque and pitch control scheme to reduce tower vibration in the full load region and tuned control parameters through multi-objective optimization and multi-criteria decision [29]. Pan and Wang presented a repetitive fuzzy PID control in the PAC domain for the direct-drive PMSG-based WT to smooth power production [30]. Poureh et al. enhanced the gain-scheduled PAC system with four modifications to eliminate the effects of hidden terms caused by inaccurate realizations [31]. Hawari et al. applied a reduced-order model capturing low-frequency behaviour to establish a gain-tuning formula for the collective PAC [32]. Hu et al. implemented an open, modular, and adaptable controller for large-scale floating WTs, investigating the coupling effects between pitch regulation and platform motion [33]. Sahin and Yavrucuk introduced and verified an envelope protection system algorithm using a self-developed bladed simulator, which was adaptive to operational conditions and effectively reduced turbine excessive loads [34].

Essadki and Nasser presented an extended linear quadratic regulator (LQR) to promote control performance at various operational conditions [35]. Camblong et al. designed and analyzed an LQR for fatigue load reduction and grid primary frequency regulation [36]. Yao et al. introduced a novel 2-DOF control strategy and its model predictive control (MPC) for variable power regulation and tower load mitigation, which compared three schemes of the RSC and PAC coordination [37]. Routray and Hur compared feedback and feedforward MPCs to maximize energy capture over the entire envelope of operation regions [38]. Wakui et al. integrated an internal model identified from an aero-elastic-hydro simulation of previewed disturbances into the MPC to stabilize power output and platform motion [39]. Yao et al. upgraded the MPC of an individual turbine governing generator torque and pitch angle to the wind farm level through a hierarchical distribution architecture [40]. Sudharsan et al. developed a pseudo-adaptive MPC of the 2-DOF torque and pitch regulation to alleviate fatigue load [41]. Lin et al. implemented a variable-weight MPC to optimize power quality and load conditions [42].

Kelkoul and Boumediene proposed a sliding mode control (SMC) enhanced by a super-twisting algorithm to reduce generator chattering [43]. Yang et al. enhanced the SMC with a perturbation observer to deal with aerodynamic nonlinearities, generator parameter uncertainties, and wind stochastics, which enhanced wind production robustness and fault ride-through capability [44]. Zholtayev et al. developed an adaptive SMC for the PMSG, achieving the best trade-off between output performance and chattering reduction [45]. Baltag et al. presented a dynamic model that reflected speed and pitch effects on shaft rotation and designed an H-infinity (H-∞) controller to handle corresponding synthesis control [46]. Yin et al. applied an optimal loop-shaping H-∞ control in the hydraulic WT system for better tracking performance [47]. Huo and Xu investigated an automatic generation control scheme based on multi-event triggered mechanisms for networked wind systems to improve the utilization efficiency of network transmission resources [48]. Song et al. integrated a non-standard extended Kalman filter (EKF) estimator within the PAC loop for consistent optimal performance under global power regulation while avoiding wind measurement [49]. Soliman et al. proposed an adaptive fuzzy logic control (FLC) for the PMSG-based WT to enhance the performance of the grid-tied wind generator system [50]. Wang et al. presented a model-free adaptive predictive controller (APC) involving an ahead forecasting of wind speed, a multi-objective optimization of maximum output and minimum control input, and a one-step predictive control algorithm based on pseudo partial derivative [51]. **Table 1.2** summarizes control characteristics in the above literature.

Table 1.2 Summation of different turbine controls in the literature

Source	Internal Model	Power Strategy	Control Type	Technologies
Gambier and Meng [26]	power coefficient	maximum	PID	regional design, collective pitch, tower fore-aft damping, gain scheduling
Junejo et al. [27]	power coefficient	maximum	PID	VAWT control, torque control, physical model
Tu et al. [28]	power coefficient	maximum	PID	reduced-load design, frequency response
Lara et al. [29]	power coefficient	maximum	PID	tower fore-aft damping, yaw control, multi-objective optimization, multi-criteria decision
Pan and Wang [30]	power coefficient	maximum	PID	repetitive control, fuzzy logic, direct-drive system
Poureh et al. [31]	power coefficient	maximum	PID	gain scheduling, multiple configurations
Hawari et al. [32]	power coefficient	maximum	PID	reduced-order model, gain scheduling, frequency response
Hu et al. [33]	power coefficient	maximum	PID	gain scheduling, active stall control, platform motion optimization
Sahin and Yavrucuk [34]	power coefficient	maximum	PID	regional torque control, gain scheduling, gain correction, envelop protection
Essadki and Nasser [35]	power coefficient	maximum	LQR	DFIG system, disturbance rejection
Camblong et al. [36]	power coefficient	maximum	LQR	fatigue reduction, primary frequency regulation

Yao et al. [37]	power coefficient	variable	MPC	tower fatigue load optimization, torque-pitch coordination
Routray and Hur [38]	power coefficient	maximum	MPC	region switching, feedback and feedforward comparison
Wakui et al. [39]	power coefficient	maximum	MPC	disturbance preview, power and platform stabilization
Yao et al. [40]	power coefficient	adaptive	MPC	distribution control, tower fatigue reduction, active power control
Sudharsan et al. [41]	power coefficient	maximum	MPC	tower fatigue mitigation, torque-pitch coordination, pseudo-adaptive control
Lin et al. [42]	power coefficient	maximum	MPC	blade and shaft load reduction, torque-pitch coordination, adaptive weight matrix
Kelkoul and Boumediene [43]	power coefficient	maximum	SMC	super twisting algorithm, DFIG vector control, second-order sliding mode
Yang et al. [44]	power coefficient	maximum	SMC	perturbation observer, inaccurate model tolerance, real-time estimation
Zholtayev et al. [45]	power coefficient	maximum	SMC	super twisting algorithm, chattering reduction, second-order sliding mode
Baltag et al. [46]	power coefficient	maximum	H-∞	synthesis control, weighted process
Yin et al. [47]	power coefficient	maximum	H-∞	hybrid system with hydraulic transmission, torque control, pump control
Huo and Xu [48]	power coefficient	maximum	H-∞	multi-event triggered mechanism, network wind-integrated power system
Song et al. [49]	power coefficient	maximum	EKF	Kalmen estimator, torque control, nonlinear feedback
Soliman et al. [50]	power coefficient	maximum	FLC	fuzzy rule, grid-connected system, PMSG bilateral control
Wang et al. [51]	data prediction	maximum	APC	data-driven control, feedforward control, multi-objective optimization

According to the above review, most control systems target maximum wind power capture, i.e., under maximum power strategy (MPS). Aerodynamic complexities and nonlinearities result in conventional MPS systems. The nature of the MPS is to pre-measure several power points to simplify aerodynamic responses, called maximum power point tracking (MPPT) [52]. The MPPT-based MPS relies on a power coefficient approach as its internal mathematic model to describe aerodynamics, which is the fundamental principle of how a controller deals with nonlinearities. The MPS limits turbine operation to a narrow region, although a

3-DOF turbine has more output possibilities.

However, when WTs connect to the grid and have an increasing share of generation, the next-stage turbine control design has to consider the equilibrium between generation and consumption because the grid must follow the energy conservation law to ensure stability [53]. In other words, future WTs should actively change their output to meet power demand. This study will develop a novel online power strategy (OPS) to update the overall control policy to respond to user demand. Correspondingly, an OPS solution does not require any prior assumptions but is a real-time optimization of power command (or reference or target or demand), i.e., power reference point tracking (PRPT) [54]. It is hard for conventional mathematic tools to achieve this online optimization since a single or several equations cannot adequately explain aerodynamic nonlinearities. Fortunately, ML technologies provide extraordinary solutions for such problems, such as artificial neural networks (ANNs), deep learning (DL), and reinforcement learning (RL). This investigation will utilize ML to establish intelligent OPS-based turbine systems. **Figure 1.7** shows the upgrade from MPS-based to OPS-based. Meanwhile, this research will integrate artificial intelligence insides turbines and let turbines learn to control themselves.

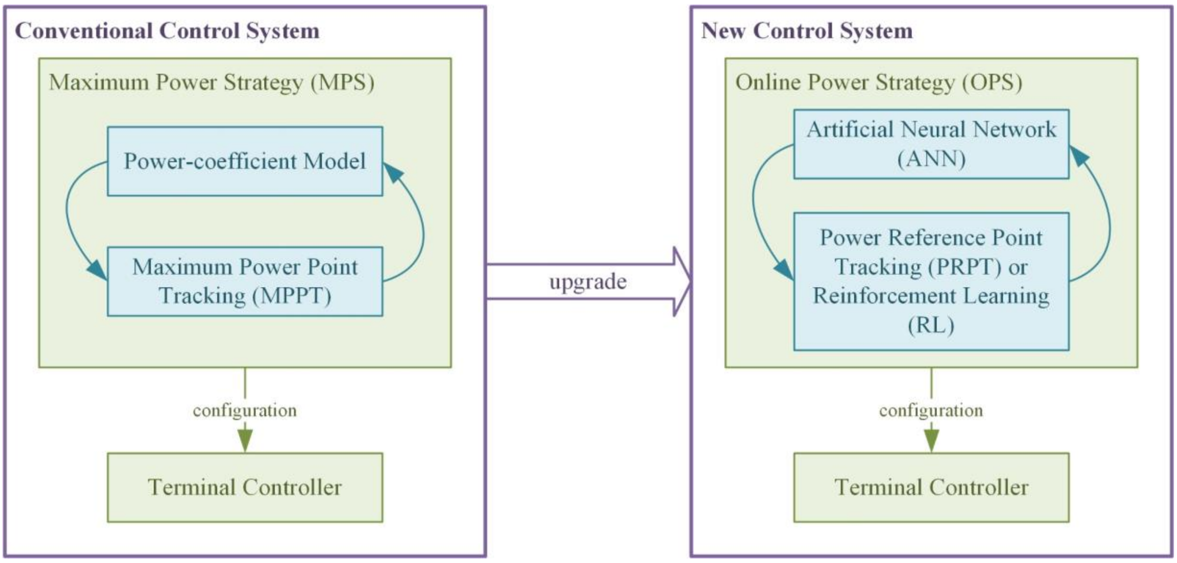

Figure 1.7 Upgrade from the MPS-based control to the OPS-based control

1.3 Machine Learning Review

ML has various applications in wind energy and forms three main research areas. The first is to model the wind turbine power curve (WTPC) to describe wind power across different operation regions [55]. The second is time series analysis for wind or output, which targets a series model based on historical data to handle wind uncertainties [56]. The above two topics focus on using ANNs to learn from data, drastically reducing dependence on physical models [57]. One of the most attractive advantages of ANNs is that they apply common frameworks to deal with all kinds of data without considering specific physical processes. ANNs have evolved into the era of DL, which enhances the learning capability for sophisticated data features and structures, especially nonlinear processes [57]. The last topic concerns RL, which promotes more intelligent decision-making systems [53]. This study tries to combine these three aspects in the co-control design of a WT.

1.3.1 Power Modelling

Before ML, parametric models are the most popular way to fit a WTPC model from sample data [58]. A parametric model derives output response by constructing a set of mathematical equations including a few parameters, potentially relying on underlying physical laws [59]. However, owing to the heavy nonlinearity of aerodynamics, obtaining accurate parameters for long blades becomes more challenging. Besides, most parametric methods can only ensure their accuracy for specific types. In contrast to parametric methods, ANNs, as nonparametric methods, do not require prior knowledge to derive the relationship from input to output, which ensures high reliability and strong resilience for all kinds of WTs [59].

Regarding WTPC modelling, many ANNs try to improve their accuracy and generalization by different network architectures, training algorithms, or data processing techniques. Jyothi and Rao introduced an adaptive wavelet neural network (WNN) for one-step-ahead wind power forecasting, which outperformed standard ANN and adaptive neuro-fuzzy inference system (ANFIS) methods [60]. Li et al. designed a three-layer (4-8-1) multilayer perceptron network (MLPN) with compressing functions to estimate output power from two sets of meteorological data with different sampling rates [61]. Mabel and Fernandez implemented a feedforward three-layer (3-4-1) MLPN to forecast monthly power generation [62]. Pelletier et al. developed a multi-stage modelling technique for two-layer MLPNs to reduce absolute and random errors, which surpassed parametric, nonparametric, and discrete methods [63]. Morshedizadeh et al. combined a feature selection method and a dynamic MLPN structure to improve power monitoring [64]. Zhao et al. presented a topology framework for day-ahead power forecasting, in which a Kalman filter processed wind speed (or velocity), a vector decomposition decomposed wind direction into sine and cosine components, and a three-layer MLPN was in charge of input-output mapping [65]. Lin et al. employed the isolation forest (iForest) as outlier (or anomaly) detection and fed filtered data into a five-layer deep neural network (DNN) for offshore power forecasting [66]. Zhou et al. designed two innovative structures of the physical process and DNN to forecast wind power under wake effects [67]. Lin and Liu investigated the nonlinear correlation between input features to reduce the size and scale of a DNN [68]. Shetty et al. developed a fast and efficient radial basis function network (RBFN) enhanced by a particle swarm optimization fuzzy c-means (PSO-FCM) clustering and an extreme learning machine (ELM) [69]. Karamichailidou et al. introduced a novel training algorithm for a thin plate spline-based RBFN, which combined the nonsymmetric fuzzy means (NSFM) and the tabu search (TS) [70]. **Table 1.3** lists the main attributes of the mentioned WTPC models.

Table 1.3 Summation of WTPC modelling methods in the literature

Source	Network	Algorithms	Input Features
Jyothi and Rao [60]	WNN	Morlet wavelet, backpropagation training	wind speed, wind direction, wind density, ambient temperature
Li et al. [61]	MLPN	compressing functions	wind speed, wind direction
Mabel and Fernandez [62]	MLPN	backpropagation training	wind speed, relative humidity, generation hour
Pelletier et al. [63]	MLPN	data filtering, logarithmic profile, power law, resampling	wind speed, wind direction, yaw error, air density, turbulence intensity, wind shear
Morshedizadeh et al. [64]	MLPN	Pearson correlation coefficient	wind speed, rotor speed, gear temperature, pitch angle
Zhao et al. [65]	MLPN	Kalman filter, direction decomposition, Levenberge-Marquardt training	wind speed, wind direction, temperature, pressure, humidity
Lin et al. [66]	DNN	iForest, general platform	wind speed, nacelle orientation, yaw error, pitch angle, ambient temperature
Zhou et al. [67]	DNN	parallel structure, data-driven transfer regression	wind speed, wind direction, distance, azimuth angle, yaw error
Lin and Liu [68]	DNN	outlier detection, correlation, general platform	wind speed, wind shear, pitch angle, nacelle orientation, yaw error, ambient temperature
Shetty et al. [69]	RBFN	PSO-FCM clustering, ELM	wind speed, wind direction, pitch angle, air density, rotor speed
Karamichailidou et al. [70]	RBFN	NSFM-TS	wind speed, wind direction, pitch angle, ambient temperature

Almost all ANNs about turbine modelling only focus on power but ignore other aerodynamic features. The analysis in **section 1.2** indicates the regulation objectives of modern WTs also need to consider the stability of shaft rotation and the fore-aft movement of tower vibration. Hence, a power model is not enough for a control system, and there is an urgent need for a comprehensive aerodynamic model to describe the relevant effects of rotor torque and hub thrust. Besides, simple ANNs cannot fully understand the aerodynamics of a long blade for high-power WTs. Thus, hybrid ANNs will be the solution for multi-aerodynamics modelling. A popular training way is to apply a general ML platform for parallel and accelerated computing, such as TensorFlow [71]. In conclusion, this study will investigate ANN-based aerodynamic modelling to predict thrust, torque, and power. Meanwhile, this study will use TensorFlow to build, train, and test models.

1.3.2 Wind Forecasting

Wind forecasting provides a necessary wind reference that determines the 3-DOF objective, so a reliable wind model is critical for output stability [72]. Conventional wind forecasting involves physical and statistical models [73]. Physical models, a part of numerical weather prediction (NWP), take multiple meteorological and topographical parameters as input and calculate potential wind variation through physical laws [57]. Statistical methods analyze historical data to estimate wind distribution and calculate the most potential wind [74]. However, physical models are commonly expensive and slow in computation due to fluid calculations, e.g., the Navier-Stokes equation [75]. Statistical methods require a process to be stationary, which is too strict for short-term time series to satisfy [76]. In contrast, ML provides an efficient calculation with no prerequisite for series distribution.

ML favours employing deep structures to identify complex features for time series forecasting, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [56]. Neshat et al. proposed an evolutionary decomposition and a generalized normal distribution optimization to enhance a bidirectional long short-term memory (LSTM) for wind speed prediction [77]. Chen et al. introduced a novel hybrid CNN-LSTM architecture for wind speed forecasting, which encoded high dimensional input into embedding vectors and decoded prediction results [78]. Ahmed et al. employed an ensemble-based LSTM to investigate seasonal and periodic characteristics over wide data segmentations (3-month~1-day) and time horizons (14-day~5-minute) [79]. Xiang et al. combined a self-attention temporal CNN and an LSTM to forecast ultra-short-term wind power [80]. Liu et al. applied error correction and variational model decomposition into a hybrid CNN-LSTM-MLP for hourly solar irradiance prediction [81]. Ewees et al. adopted a human-behaviour-based metaheuristic algorithm, a heap-based optimizer, to train the LSTM, thus improving the accuracy of the LSTM on wind power forecasting [82]. Agga et al. studied DL architectures in photovoltaic power forecasting and suggested that the CNN-LSTM suppresses standard ML and single DL models [83]. Fu et al. presented a framework of outlier processing, mode decomposition, subsequence reconstruction, and stacked generalization for short-term wind speed forecasting [84]. Also, data filtering and mode decomposition are helpful tools that can improve model training for complex series with the aid of nonlinearity identification, noise cleaning, and information extraction [85]. Although diverse state-of-the-art models continuously improve the reliability and accuracy of time series forecasting, their prediction windows are too long for a turbine control system to update the control objective. In addition, most wind forecasting methods focus on wind velocity but ignore direction. This work targets seconds-level wind series modelling to forecast velocity and direction simultaneously.

1.3.3 Reinforcement Learning

The essence of the OPS is equivalent to a 2-DOF optimization of rotor speed and pitch angle. Due to nonlinearities in rotor aerodynamics, it is hard for traditional mathematical tools to find optimal solutions, such as the Nelder-Mead [86] or the truncated Newton [87]. Besides, their convergence speed has no guarantee for a real-time update. Furthermore, the RSC has a higher regulation cost than the PAC because the transition of rotor speed requires the generator to absorb the variation of kinetic energy, resulting in excessive power fluctuation and extra stable time. The mentioned PRPT-based OPS still follows some fixed modes but does not weigh the RSC and PAC with different priorities. Thus, this study also provides an

RL-based OPS, aside from the PRPT-based approach.

RL intends to train an agent to perform optimal action in an environment to maximize a reward [88]. Regarding such nonlinear and complex decision-making problems, an RL agent initially attempts random actions and gradually knows the best ones, yielding an RL policy that maps optimal action from observation. RL has many applications in the wind power strategy. Xie et al. proposed a deep-network RL algorithm for torque and pitch control to adapt to real-time perturbations [89]. Wei et al. adopted a model-free Q-learning method to model from state observation to control action for variable speed control [90]. Kushwaha et al. presented a Q-learning-based maximum power extraction that identified unlearned maximum power points for wind variations [91]. Peng and Feng formulated the problem of sequential decision-making as a Markov decision process (MDP) and achieved an RL-based MPPT strategy for pitch control [92]. Mazare constructed an actor-critic-based RL approach to secure pitch control under false data injections and actuator faults [93]. Jiang et al. implemented an RL model of the deep Q-network (DQN) to optimize the short-term scheduling of a hydro-wind-solar system for total power maximization [94].

Present RL-based strategies roughly consist of two categories. The first sets maximum power capture as the RL objective for an agent to find the trace of power harvest. The second lets an agent correct the reference state to handle wind uncertainties or enhance fault tolerance. To our knowledge, existing RL methods are still extensions of the MPPT theory. This paper proposes a novel RL-based generation strategy where a turbine can autonomously determine the operation state to output the desired power. This RL-based OPS can improve power efficiency and output adaptivity, which upgrades WTs to be a part of the smart grid.

1.4 Thesis Overview

Most turbine simulations rely on a famous simulator, i.e., fatigue, aerodynamics, structures, and turbulence (FAST), founded by the National Renewable Energy Laboratory (NREL) [95]. In 2017, the NREL released a powerful version (OpenFAST) that included more extensions for aero-hydro-servo-elastic (AHSE) calculation [96]. Although the FAST has outstanding achievements, it pays more attention to mechanical analysis, such as natural frequencies, damping, mode shapes, aero-elastic instabilities, and multiple loads. Hence, the FAST contains a lot of extensions that power simulation never uses, which wastes a lot of simulation time and computation resources. In contrast, the FAST simplifies all kinds of generators as a first-order torque system, which ignores necessary characteristics in different generator responses. Besides, the FAST encapsulates simulation programs and does not provide convenient access interfaces, which are not friendly to control design. Considering the above limitations, this study will develop a new simulator coupling rotor, shaft, generator, pitch, yaw, and tower, in which a controller can easily access arbitrary measurable variables.

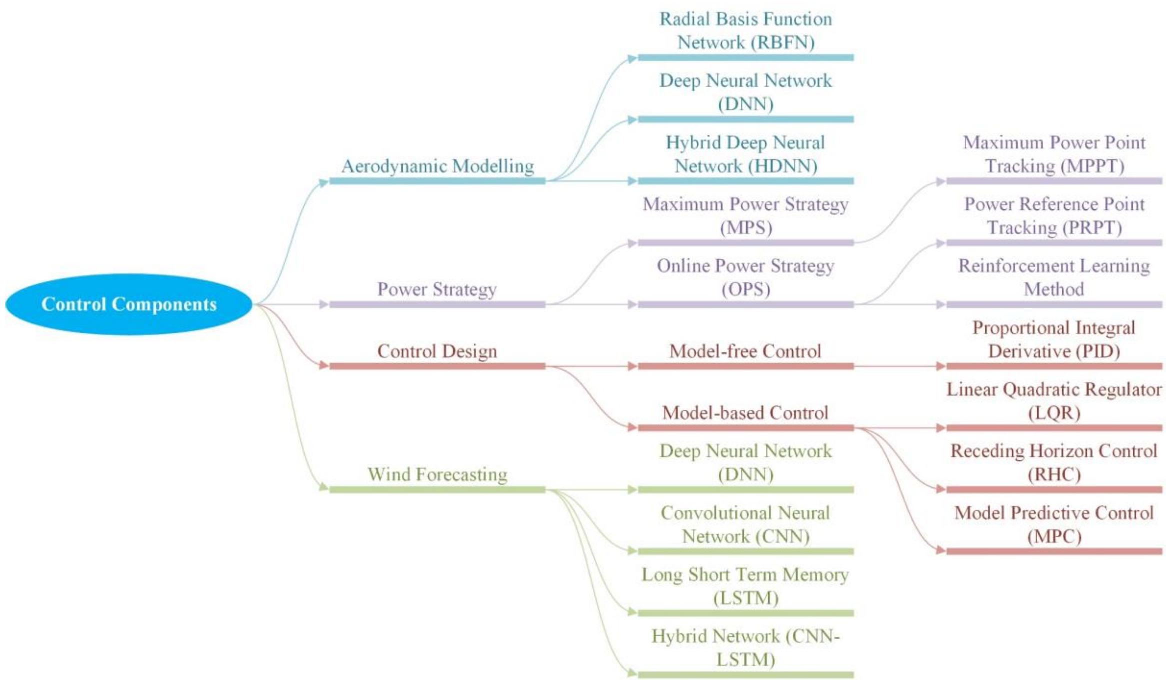

Figure 1.8 Control components developed and implemented in this study

According to the analysis of the disadvantages of present turbine systems, the main topic of this study is to upgrade WTs only for maximum capture to more intelligent systems for flexible wind generation. Meanwhile, our control systems improve the capability of handling nonlinear aerodynamics and uncertain wind stochastics. Our intelligent turbine control system consists of four components, i.e., an aerodynamic model, a power strategy, a specific controller, and a wind model. The wind model is optional for our systems, i.e., users can use their wind processing units to replace it. However, the other three must work together to perform control calculations. **Figure 1.8** categorizes models and algorithms developed in this study and includes the conventional MPS implemented for comparison.

First, this study upgrades classic power modelling to universal aerodynamic modelling and introduces three ANNs, including RBFN, DNN, and hybrid deep neural network (HDNN). Based on aerodynamic modelling, the novel OPS can replace the conventional MPS. The OPS implementation has two routes: ANN-based PRPT and RL-based decision-making. Under the OPS, this study designs model-free and model-based controllers, including PID, LQR, and MPC, which also involve a recursive MPC, namely receding horizon control (RHC) [97]. Besides, this study investigates four wind forecasting models to reduce control deviation caused by wind uncertainties, including DNN, CNN, LSTM, and hybrid CNN-LSTM. All involved ML technologies are not auxiliary tools for performance optimization after installation but work on the fundamental layer of the control algorithm, i.e., their saved model files are necessary for the first run. Therefore, this study develops an aerodynamic solver for ANN and RL training and collects historical wind data for wind forecasting model training. The following summarizes the main novelties of this study:

- Classic turbine controls follow a rule of fixed generation mode, which leads to unadjustable power output at each wind condition. This study focuses on applying ML technologies in turbine control to achieve flexible wind power, by which individual turbines can capture and convert needed power.
- ML plays roles in aerodynamic modelling, intelligent decision-making, and time-series analysis, which avoids sophisticated physical models and reduces difficulties in nonlinear optimization. Besides, with the help of ML, turbine operation optimization is entirely online and does not require prior knowledge about working points.
- The core control logic is a two-hierarchy from 3-DOF optimization to parameter update. After the power strategy determines the 3-DOF steady state according to wind conditions and power demand, the control system updates the dynamic parameters about local linearization. This logic decouples a complex problem regulating a nonlinear process into several non-complex tasks.
- This study upgrades conventional controllers to be compatible with the novel control

framework. Meanwhile, ML simplifies parameter calculations required by control configuration, especially force and sensitivity estimations.

Figure 1.9 Chapter relationship with research purposes

The following summarizes the main contents and contributions of each chapter (**Figure 1.9** draws the relationship between chapters):

- **Chapter 2** discusses physical models for turbine dynamics simulation, including rotor aerodynamics, shaft rotation (one-mass or two-mass), generator regulation (DFIG or PMSG), pitch and yaw servo responses, and tower fore-aft movement. These theoretical models construct a real-time simulator and derive two small-signal models for model-based control.
- **Chapter 3** introduced three turbine definitions, i.e., 5 MW, 10 MW, and 15 MW, which cover popular installed capacities. Hence, corresponding simulations can provide persuasive evidence to verify the adaptability and reliability of the proposed ML-based control.
- **Chapter 4** presents an overview of the control framework, which presents two routes (model-free and model-based) to realize a novel turbine control logic. This chapter also gives specific control schematics for DFIG and PMSG-based turbines and explains related time settings.

- **Chapter 5** discusses ANN-based aerodynamic modelling to predict wind power, hub thrust, and rotor torque instead of conventional power modelling. This chapter presents three networks for comparison, i.e., RBFN, DNN, and HDNN. The HDNN combines the RBF layer and the DL structure, which assures high fidelity of multiple aerodynamics.
- **Chapter 6** reviews the MPPT-based MPS and establishes the PRPT-based OPS. The PRPT introduces an ANN-based algorithm that calculates speed and pitch references as the regulation objective for demand-oriented power capture rather than maximum capture. Meanwhile, this chapter provides a convenient way to update force sensitivities for optimal control configuration.
- **Chapter 7** develops an RL-based OPS for intelligent decision-making. In addition to the PRPT-based OPS, an RL-based OPS can intelligently determine the reference state by weighing the RSC and PAC regulation costs. The RL agent uses a pseudo-Markov algorithm based on bisection and queue detection to achieve higher accuracy for discrete actions. The agent saves its learning results as an eager policy for lite implementation.
- **Chapter 8** proposes four specific controllers (PID, LQR, RHC, and MPC) to execute the command of an OPS. The topology of the PID loops is model-free, which applies gain scheduling to achieve optimal dynamic performance. The LQR, RHC, and MPC are model-based approaches which realize weighted multi-objective regulation.
- **Chapter 9** proposes four models (DNN, CNN, LSTM, and CNN-LSTM) for wind vector series forecasting, enhancing control accuracy and reliability under wind stochastics. The proposed wind forecasting can simultaneously process velocity and direction with the help of the compass-vector transformation.
- **Chapter 10** carries out the case study of the ML-based control systems. **Chapter 11** summarizes the main conclusions drawn from the case study.

Chapter 2 Wind Turbine System Modelling

Although there are some simulators for modelling turbine dynamics, such as the FAST (or OpenFAST) [96], QBlade [98], and Bladed [99], most of them focus on the responses of airflow and structure but simplify electrical devices and control systems. Their generator models are usually simple torque systems, which cannot reflect the dynamic process between the generator stator and rotor. In addition, most turbine simulators do not allow or accept an external control system to replace predefined ones. Hence, this research develops a dynamic simulator (**Figure 2.1**) for real-time turbine responses, which adopts fully-coupled modelling, provides a unified transduce and control bus, and allows the access of an external controller.

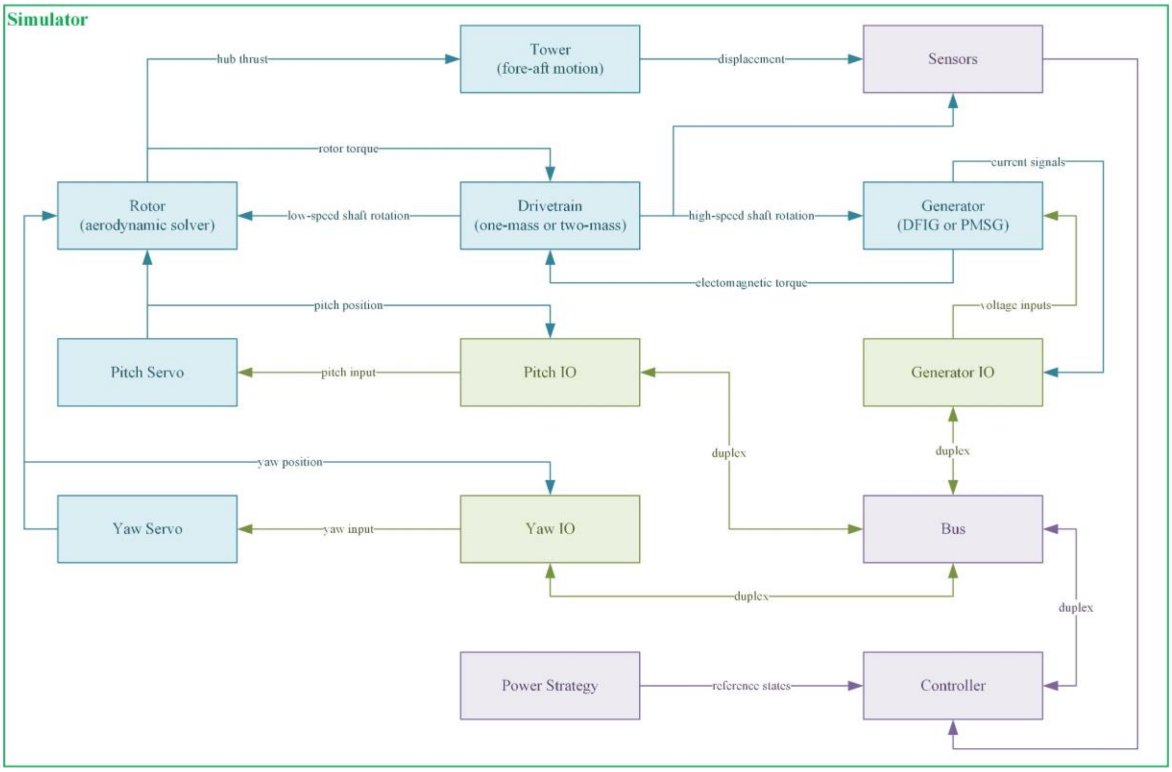

Figure 2.1 Simulator schematic for 3-DOF turbine dynamics

The developed simulator has six parts for the 3-DOF process, including rotor aerodynamics, drivetrain rotation, generator equivalent circuit, pitch servo, yaw servo, and tower top fore-aft motion. At the front of this section, it is necessary to clarify that two terminologies related to modelling have different meanings. The system model in this study refers to a set of subsystems that constitute the simulator. The control model represents the built-in model of a model-based control for trajectory prediction.

As the core component, the rotor aerodynamic solver updates the aerodynamic response of a rotor disk based on the operation state of wind speed, shaft rotation, pitch position, and yaw position, i.e., calculates rotor torque and hub thrust. The drivetrain module determines the rotation state of low-speed and high-speed shafts under the condition of rotor torque and electromagnetic torque. The drivetrain has two modelling methods: a one-mass model for direct drive or gearbox and a two-mass model for gearing, which updates rotational speeds on the low-speed end (rotor disk side) and high-speed end (generator side). The generator absorbs the kinetic energy of shaft rotation and converts it to electric power, which involves two widespread generators: DFIG and PMSG. The DFIG or PMSG adopts the same control logic that monitors current signals and adjusts voltage inputs to reach the desired generation state, i.e., output a given apparent power at a nominated speed. The pitch or yaw servo simulation relies on a first-order inertia model to obtain its angle position for a given input signal. The tower part contains a second-order model of top fore-aft motion to describe tower harmonic behaviour affected by hub thrust.

The simulator has three input-output (IO) interfaces for terminal actuators, i.e., generator, pitch servo, and yaw servo. The controller reads actuator states and calculates control inputs according to its control policy. Besides, the simulator measures tower displacement and rotational speed (at the high-speed end) for the controller to govern tower and shaft kinetic responses.

Along with system modelling inside the simulator, this section derives two small-signal control models according to generator type. They are necessary for model-based control due to trajectory calculation and optimization. Compared with existing control models, our small-signal models have the following advantages: consider tower top motion besides rotor rotation, establish the causal relationship between controllable variables and kinetic responses, and have direct access to terminal actuators.

2.1 Rotor Aerodynamics

The calculation of rotor aerodynamics intends to solve thrust perpendicular to the rotor hub and torque acting on the main shaft. This calculation is similar to a general propeller, which employs an iterative approach based on the blade element momentum (BEM) theory [100]. The BEM introduced by Glauert in 1935 computes thrust and torque for a set of wind speed, rotor angular speed and blade pitch angle [100]. The BEM consists of two assumptions:

- Each blade has several elements represented by 2-D airfoils subjected to local physical events [11]. Besides, there is no aerodynamic interaction between any two adjacent elements, i.e., no radial flow, and the forces on a blade element are determined independently by the lift and drag characteristics of the airfoil shape [101]. This paper only considers identical blade geometry and collective pitch regulation, so all blades share the same properties.
- As an extension and supplement of the Betz theory, the BEM theory assumes that a rotor of blades acts as an actuator disk transferring the kinetic energy of the inflow airflow to the rotational energy of the main shaft and thus reducing the airflow velocity, resulting in angular rotation to the wake flow, and making the streamlines diverge [10]. The rotor disk is considered frictionless, the flow is incompressible, and the momentum loss after passing through the rotor induces axial and angular velocities affecting thrust and torque imposed on the rotor [17].

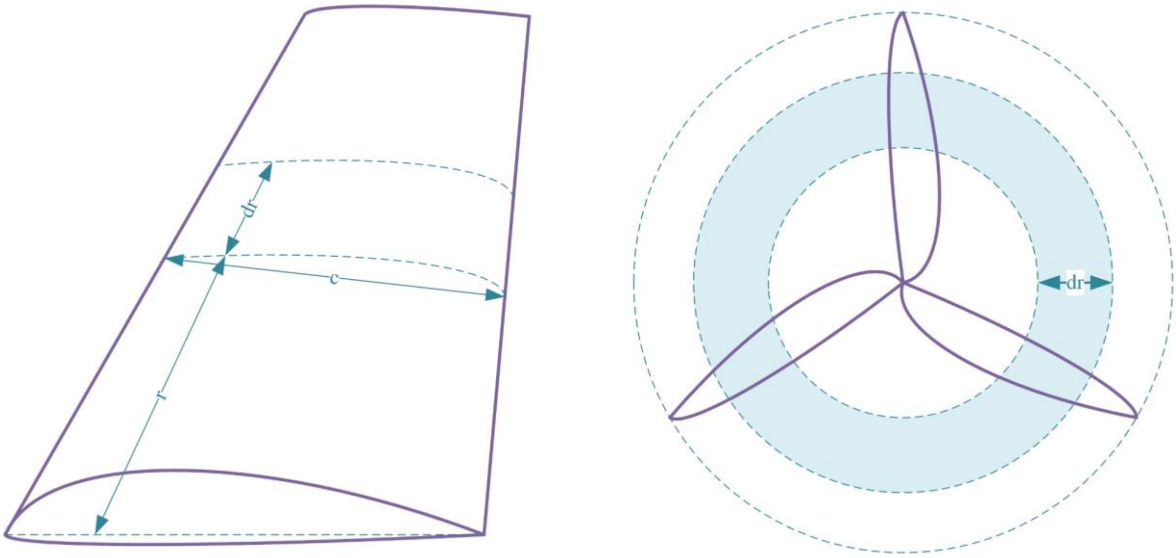

Figure 2.2 Element division and control volume of a three-bladed rotor

Figure 2.2 shows an example of a three-bladed rotor that the BEM theory divides into several elements. Each blade element is naturally an individual airfoil that physically corresponds to a definition of thickness and camber [7]. The blade description comprises several airfoil files with lift, drag, and moment coefficients [14]. It notes that the pitching moment mainly affects blade root load [10], which relates to the movement of the pitch servo. Given that a pitch servo can produce sufficient torque, the process of blade pitching is equivalent to a classic first-order problem, which ignores the effects of the pitching moment. Hence, the BEM calculation can skip calculations about the pitching moment. The BEM solver separately computes the response of each airfoil and later summates all elements. Individual airfoil calculation implies a parallel computation of the BEM solver, which synchronously calculates elements by multiprocessing to avoid time-consuming sequential solutions. The BEM solver provides an empirical and efficient way to estimate aerodynamics from the blade definition. This solver plays three fundamental roles in this study:

- A real-time simulator updates rotor forces that affect shaft rotation and tower motion.
- A data source generates synthetic data for an ANN-based model to establish the aerodynamic relation from state inputs to force outcomes.
- A wrapped environment executes and evaluates the action of an RL agent to reinforce the agent policy for the 2-DOF decision-making.

The core algorithm of the BEM solver is to find a proper relative angle for each element that meets the geometry constraints in the Newton-Raphson iteration [102]. **Figure 2.3** draws the geometric relation of a local blade element. **Eq. (1)** describes the relative wind angle for a given condition of wind velocity and rotor rotation [100].

$$\tan\varphi = \frac{v_n}{v_t} = \frac{v_i(1 - a_\perp)}{\omega_d r(1 + a_\angle)} \tag{1}$$

where

φ: the relative wind angle (rad)

$v_n,\ v_t$: the normal (or axial) and tangential winds (m/s)

$a_\perp,\ a_\angle$: the axial and angular induction factors (dimensionless)

v_i: the wind velocity of the inflow (m/s)

ω_d: the angular speed of the rotor (rad/s)

r: the distance from the centre of a selected element to the rotor hub (m)

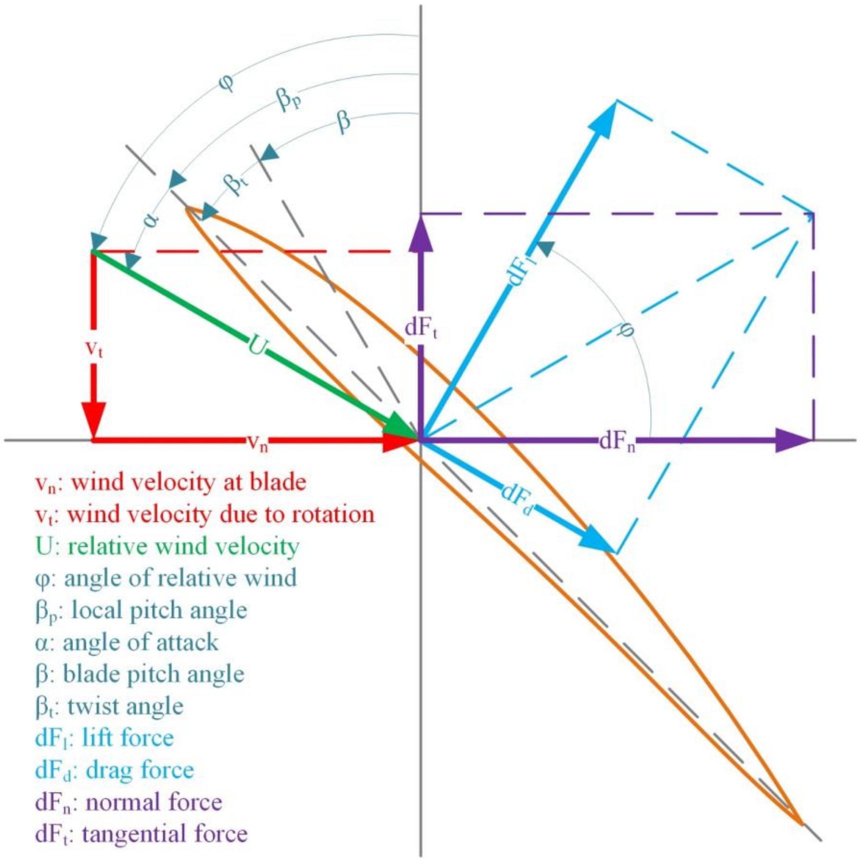

Figure 2.3 Blade geometry for analysis of a horizontal rotor disk

Eq. (1) considers the fractional decrease between the free stream and the rotor plane and includes the induced angular rotation across the flow disc, which accounts for the axial and angular induction factors, respectively. Meanwhile, **Eq. (1)** implies a termination criterion (**Eq. (2)**) to stop iteration, which indicates the convergence of the relative angle.

$$\mathrm{err}(\varphi) = \mathrm{abs}\left(\arctan\left(\frac{v_n}{v_t}\right) - \varphi\right) \tag{2}$$

Eq. (3) describes the influence of pitch regulation, which adjusts blade pitch to affect the angle of attack [10]. The nature of pitch regulation is to change the lift and drag forces on an airfoil by altering the angle of attack. Due to the blade design of twisting airfoils, **Eq. (4)** eliminates the mismatch between airfoil measurement and initial element position [7].

$$\alpha = \varphi - \beta_p \tag{3}$$

$$\beta_p = \beta + \beta_t \tag{4}$$

where

α: the angle of attack (rad)

β_p: the local pitch angle (rad)

β: the blade pitch angle (rad)

β_t: the initial twist angle (rad)

It notes that the angle of attack is the key to reading lift and drag coefficients from an airfoil file, which are usually discrete values along with sampled angles. Linear interpolation between two samples is necessary to derive an appropriate value. With airfoil read-write and linear interpolation, **Eqs. (5)(6)** compute the lift and drag forces of an element [7].

$$dF_l = C_l(\alpha)\frac{1}{2}\rho U^2 c dr \tag{5}$$

$$dF_d = C_d(\alpha)\frac{1}{2}\rho U^2 c dr \tag{6}$$

$$U = \sqrt{v_n^2 + v_t^2} \tag{7}$$

where

$F_l,\ F_d$: the lift and drag forces (N)

$C_l(\alpha),\ C_d(\alpha)$: the lift and drag coefficients, interpolated at α (dimensionless)

ρ: the air density (kg/m^3)

U: the magnitude of the relative wind (m/s)

c: the airfoil chord (m)

dr: the span of a blade element or the thickness of a control volume (m)

The geometry relation in **Figure 2.3** implies a force transformation from the airfoil plane to the rotation plane, which recomposes the element forces on two orthogonal axes, as **Eqs. (8)(9)** [11].

$$dF_n = dF_l\cos\varphi + dF_d\sin\varphi \tag{8}$$

$$dF_t = dF_l\sin\varphi - dF_d\cos\varphi \tag{9}$$

where

$F_n,\ F_t$: the normal (axial) and tangential forces (N)

For simplicity, the resultant orthogonal forces ($dF_n,\ dF_t$) have an alternative representation of the corresponding coefficients with a duplicated term $\frac{1}{2}\rho U^2 cdr$, as **Eqs. (10)(11)** [10].

$$dF_n = C_n \frac{1}{2}\rho U^2 cdr \tag{10}$$

$$dF_t = C_t \frac{1}{2}\rho U^2 cdr \tag{11}$$

where

C_n, C_t: the normal (axial) and tangential coefficients (dimensionless)

From the geometry analysis, **Eqs. (5)(6)(8)(9)(10)(11)** lead to a quick calculation from the interpolated lift and drag coefficients to the axial and tangential coefficients, as **Eqs. (12)(13)** [17].

$$C_n = C_l(\alpha)\cos\varphi + C_d(\alpha)\sin\varphi \tag{12}$$

$$C_t = C_l(\alpha)\sin\varphi - C_d(\alpha)\cos\varphi \tag{13}$$

According to the axial and rotational analysis, the orthogonal forces result in the local forces on a control volume, as **Eqs. (14)(15)** [7].

$$dT = BdF_n \tag{14}$$

$$dQ = BrdF_t \tag{15}$$

where

dT, dQ: the differential thrust and torque on a control volume (N, N·m)

B: the number of blades (dimensionless)

Also, the conservation of linear momentum derives the differential forces on a control volume, as **Eqs. (16)(17)**, in which the induced torque is equal but opposite to the corresponding airflow loss [10].

$$dT = 4a_{\perp}(1 - a_{\perp})\rho U^2 \pi r dr \tag{16}$$

$$dQ = 4a_{\angle}(1 - a_{\perp})\rho U \pi r^3 \omega_d dr \tag{17}$$

Eqs. (14)(15)(16)(17) update the induction factors from an iterative inflow angle as an aerodynamic consequence of wind flow and rotor rotation. **Eqs. (18)(19)** are an instantaneous equation set to compute the ultimate induction factors [11].

$$a_{\perp} = \left(\frac{4\sin^2\varphi}{\sigma' C_n} + 1\right)^{-1} \tag{18}$$

$$a_{\angle} = \left(\frac{4\sin\varphi\cos\varphi}{\sigma' C_t} - 1\right)^{-1} \tag{19}$$

$$\sigma' = \frac{Bc}{2\pi r} \tag{20}$$

where

σ': the local solidity (dimensionless)

During iteration, the inflow angle φ follows a procedure that updates the angle of attack α, interpolates the lift and drag coefficients (C_l, C_d), composes the axial and tangential coefficients (C_n, C_t), and calculates the induction factors (a, a'), which accounts for an iteration procedure of **Eqs. (3)(12)(13)(18)(19)** [10].

Since the suction side has a smaller pressure than the pressure side, the air has a trend of flowing around the tip from the lower to the upper surface [103]. Hence, the obtained geometric results require corrections to include tip and hub losses [14]. Prandtl introduced an empirical method to update the induction factors, as given in **Eqs. (21)~(23)** [102].

$$F = F_{tip} F_{hub} \tag{21}$$

$$F_{tip} = \frac{\pi}{2} \arccos\left(\exp\left(\frac{-B\left(r_{tip} - r\right)}{2r\sin\varphi}\right)\right) \tag{22}$$

$$F_{hub} = \frac{\pi}{2} \arccos\left(\exp\left(\frac{-B(r - r_{hub})}{2r\sin\varphi}\right)\right) \tag{23}$$

where

F: the tip-hub correction factor (dimensionless)

F_{tip}, F_{hub}: the tip and hub losses (dimensionless)

r_{tip}: the total blade length from root to tip (m)

r_{hub}: the hub radius (m)

Since the tip-hub correction from Prandtl's method directly affects the forces derived from the momentum theory, **Eqs. (18)(19)** are updated to **Eqs. (24)(25)** [10].

$$a_{\perp} = \left(F\frac{4\sin^2\varphi}{\sigma' C_n} + 1\right)^{-1} \tag{24}$$

$$a_{\angle} = \left(F\frac{4\sin\varphi\cos\varphi}{\sigma' C_t} - 1\right)^{-1} \tag{25}$$

If an axial induction factor exceeds 0.4, the BEM will lose most accuracy [11]. Spera further introduced an empirical relation (**Eq. (26)**) to fit with measurements [11]. The BEM solver only activates Spera's correction for a result over the threshold.

$$a_{\perp} = 0.5\left(2 + K_{\perp}(1 - 2a_c) - \sqrt{(K_{\perp}(1 - 2a_c) + 2)^2 + 4(K_{\perp}a_c^2 - 1)}\right) \tag{26}$$

$$K_{\perp} = \frac{4F\sin^2\varphi}{\sigma C_n} \tag{27}$$

where

a_c: the threshold of a valid axial induction factor, approximately 0.2

By combining **Eqs. (10)(11)(14)(15)(20)**, the lift and drag forces recompose the orthogonal forces on a control volume, which results in the differential thrust perpendicular to the rotor plane and the differential torque onto the rotation axis, as **Eqs. (28)(29)** [100].

$$dT = \sigma' \pi \rho U^2 C_n r dr \tag{28}$$

$$dQ = \sigma' \pi \rho U^2 C_t r^2 dr \tag{29}$$

	Algorithm 1 Aerodynamic BEM Solver (Newton-Raphson)
Input	airfoil measurements including β_t, C_l, and C_d
	wind speed v_i, rotor speed ω_d, pitch angle β
Output	local thrust dT, local torque dQ
1.	initialize a relative wind angle (φ) with an assumption ($a_{\perp} = a_{\angle} = 0$)
2.	**while** not converged ($\mathrm{err}(\varphi) \geq \Delta\varphi_{min}$):
3.	update the attack angle α by **Eq. (3)**
4.	read the airfoil coefficients ($C_l(\alpha)$, $C_d(\alpha)$)
5.	update the force coefficients (C_n, C_t) by **Eqs. (12)(13)**
6.	update the induction factors ($a_{\perp}$, $a_{\angle}$) by **Eqs. (18)(19)**
7.	renew the relative angle φ by **Eq. (1)**
8.	evaluate the angle error, $\mathrm{err}(\varphi)$, by **Eq. (2)**
9.	calculate the tip-hub loss F by **Eqs. (21)(22)(23)**
10.	correct the induction coefficients ($a_{\perp}$, $a_{\angle}$) by **Eqs.(24)(25)(26)**
11.	calculate the local thrust and torque (dT, dQ) by **Eqs. (28)(29)**

Algorithm 1 details the complete BEM iteration on a control volume. **Rows 1~8** employ the Newton-Raphson method to update the inflow angle, and **rows 9~11** determine the thrust local and torque with the empirical corrections. By executing **Algorithm 1** across all control volumes, **Eqs. (30)(31)** sum the calculated differentials to obtain the overall thrust and torque of the rotor disk. Meanwhile, the power capture equals the production of rotational speed and rotor torque, as given in **Eq. (32)**.

$$T_d = \sum_{i=1}^{L} dT_i \tag{30}$$

$$Q_d = \sum_{i=1}^{L} dQ_i \tag{31}$$

$$P_d = \omega_d Q_d \tag{32}$$

where

T_d: the hub thrust affecting the tower displacement (N)

Q_d: the rotor torque driving the shaft rotation (N·m)

P_d: the wind power captured by the rotor disk (W)

L: the number of blade elements (dimensionless)

Eq. (32) is only effective when the BEM solver acts as a data source or training environment. The simulator omits this and applies strict torque balancing to deliver power from the rotor to the generator. Due to some simplifications, the BEM solver cannot reach the same accuracy as the computational fluid dynamics (CFD) [14] but offers an efficient simulation to reflect the variation of aerodynamic forces, sufficient to verify a turbine control.

The BEM solver only solves the situation where wind flow is perpendicular to the rotor disk. On the other hand, the solving process ignores the yaw error that induces a crosswind parallel to the rotation plane. According to the blade analysis (**Figure 2.3**), this lateral wind does not contribute to the axial wind and only affects the tangential wind. However, no matter what position the rotor disk rotates at, its upper half has an opposite result to the lower half, which almost cancels each other out. Therefore, the supplementary influence of yaw angle only updates the orthogonal wind received by the BEM solver, as **Eq. (33)** [104].

$$v_i = v' \cos(\theta' - \gamma) \tag{33}$$

where

v': the actual wind velocity (m/s)

θ': the actual wind direction (rad)

γ: the nacelle yaw angle (rad)

2.2 Drivetrain Sytems

As rotor and generator torques act on shaft ends, a drivetrain system is responsible for energy delivery from the rotor side to the generator side [18]. A drivetrain model describes the corresponding process through torque balancing conditions. Although some WTs are direct-drive, i.e., no gearbox and synchronous rotation, drivetrain models still adopt the definition of low-speed (or main) and high-speed shafts for consistency.

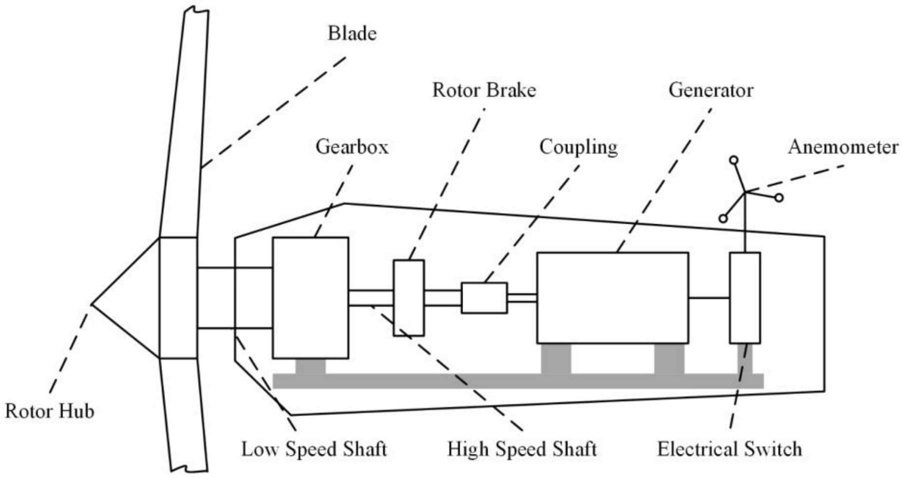

Figure 2.4a Geared system

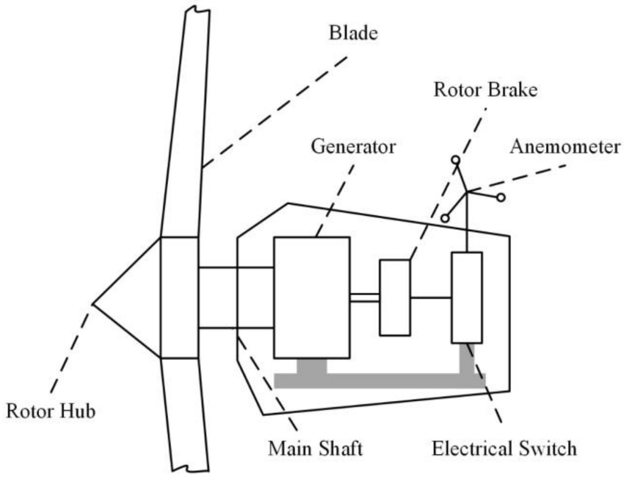

Figure 2.4b Direct-drive system

Figure 2.4 Layout of the drivetrain system

Figure 2.4 displays the layout of geared and direct-drive systems. A geared system has a gearbox to couple the turbine rotor and the generator, which scales up rotor rotation with higher speed because some generators have fewer poles and cannot accept low speed as their regular operation. However, a direct-drive system attaches the generator rotor to the turbine rotor for a generator with sufficient poles, which implies a unit gear ratio in drivetrain models. Large-capacity turbines favour the direct-drive design for a compact nacelle and no gear-torsion effect. This section introduces two commonly used drivetrain models to describe the dynamics of shafts, i.e., one-mass and two-mass [20]. The one-mass model works for both geared and direct-drive systems, while the two-mass model is only available to gear engagement between two shafts with a high gear ratio.

2.2.1 One-mass Model

The one-mass model is an ordinary differential equation (ODE) of drivetrain movement, where the drivetrain is a torsionally stiff body that terminal torques cannot twist. When the aerodynamic and electromagnetic torques act at both ends of the drivetrain, a resultant force drives the main shaft to rotate, as described by a spring model **Eq. (34)** [43]. The one-mass model synchronizes the high-speed shaft with the low-speed shaft by the gearbox, so the drivetrain has no gear slip.

$$J_t \dot{\omega}_d = Q_d - n_g Q_e - D_t \omega_d \tag{34}$$

where

J_t: the total moment of inertial including rotor disk and generator rotor (kg·m^2)

D_t: the total damping constant (N·m/(rad/s))

Q_e: the electromagnetic torque (N·m)

n_g: the gearbox ratio (dimensionless)

2.2.2 Two-mass Model

The two-mass model simultaneously considers stiffness and damping effects and describes the dynamics on the low-speed and high-speed shafts separately, as **Eqs. (35)(36)** [105]. Compared with the one-mass model, the two-mass model does not directly couple terminal torques on the main shaft but applies a breaking torque to involve gear engagement [106].

$$J_d \dot{\omega}_d = Q_d - Q_s - D_d \omega_d \tag{35}$$

$$J_m \dot{\omega}_m = \frac{Q_s}{n_g} - Q_e - D_m \omega_m \tag{36}$$

$$Q_s = D_s \left(\omega_d - \frac{\omega_m}{n_g} \right) + K_s \left(\theta_d - \frac{\theta_m}{n_g} \right) \tag{37}$$

$$\theta_d = \int_0^T \omega_d \, dt \tag{38}$$

$$\theta_m = \int_0^T \omega_m \, dt \tag{39}$$

where

ω_m: the rotational speed on the motor side (rad/s)

J_d: the moment of inertia of the rotor disk (kg·m^2)

D_d: the external damping on the roto disk side (N·m/(rad/s))

J_m: the moment of inertia of the motor (or generator) (kg·m^2)

D_m: the external damping on the motor side (N·m/(rad/s))

D_s: the mutual damping coefficient (N·m/(rad/s))

K_s: the elastic coefficient (N·m/rad)

Q_s: the breaking torque for shaft gearing (N·m)

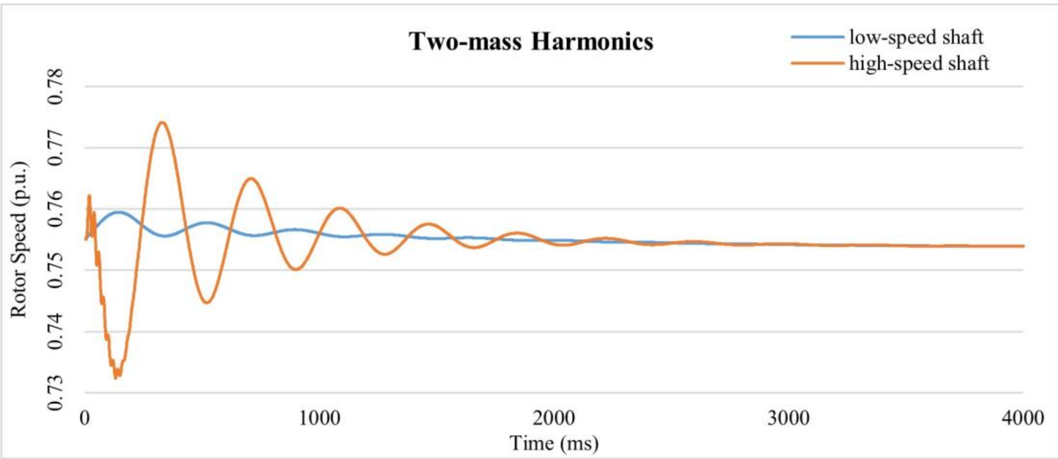

Figure 2.5 Example of the two-mass harmonic response

The two-mass model is closer to actual gearbox behaviours, which involves rotational speed difference and relative angular position between two shafts. On the other hand, two shafts are not synchronous until the drivetrain reaches a balanced state. **Figure 2.5** provides an extreme example of gearing a static high-speed shaft and a rotating low-speed shaft. For a gear ratio that leads to apparent elastic torsion, the two-mass model is more accurate than the one-mass model.

2.2.3 Discussion

If a geared drivetrain has properties such as $D_s \rightarrow 0$ and $K_s \rightarrow \infty$, **Eqs. (40)(41)** can eliminate the breaking torque between two shaft models. Meanwhile, this also implies that the geared system has no angular shift inside.

$$J_d \dot{\omega}_d + n_g J_m \dot{\omega}_m = Q_d - D_d \omega_d - n_g D_m \omega_m - n_g Q_e \tag{40}$$

$$\left(J_d + n_g^2 J_m\right) \dot{\omega}_d = Q_d - n_g Q_e - \left(D_d + n_g^2 D_m\right) \omega_d \tag{41}$$

Also, **Eqs. (40)(41)** infer a transformation from two-mass to one-mass, as **Eqs. (42)(43)**, which depends on treating two shafts as individual or whole.

$$J_t = J_d + n_g^2 J_m \tag{42}$$

$$D_t = D_d + n_g^2 D_m \tag{43}$$

The one-mass model can suit a direct-drive or geared system, but the two-mass model only simulates a geared system. Besides, two-mass applications should satisfy a condition ($n_g \geq 3$) as possible because a smaller gear ratio leads to unpredictable and unusual oscillation. Regarding a control model, most turbines only place a speed sensor at the high-speed end, and long-term sampling leads to cumulative error in the angle integrals, so the one-mass is superior in model simplification and control accuracy. Harmonics occurring on the high-speed end due to gear engagement can quickly decay and disappear. Therefore, a control design can only rely on the one-mass shaft model to estimate rotation trajectory.

2.3 Generator Sytems

A generator system has two regulation objectives: energy conversion and torque balancing. When the incoming flow drives the shaft to rotate, the generator absorbs kinetic energy and outputs electric power. The shaft can accelerate during this process until the electromagnetic torque balances the aerodynamic torque. The generator control is the most significant part of the RSC, which should regulate rotation at the nominated speed to capture the designed wind power. The DFIG and PMSG are the most popular types in the wind industry. This section will introduce their physical models on the equivalent circuit for simulation, which also derives their control models for model-free or model-based controllers.

2.3.1 Doubly Fed Induction Generator

Most WTs below 10 MW favour the DFIG due to its compact size and simple structure. The DFIG allows a reduced converter size, while synchronous or other generators require full-size converters and filters, contributing to a compact rotor design [107]. The DFIG belongs to the asynchronous motor that allows variable speed, whose stator connects to constant bus voltage and electric frequency no matter how rotor speed changes [108]. The DFIG has some advantages, such as high efficiency, variable rotor speed for output maximization, flexible power factor, and independent active and reactive power control [108].

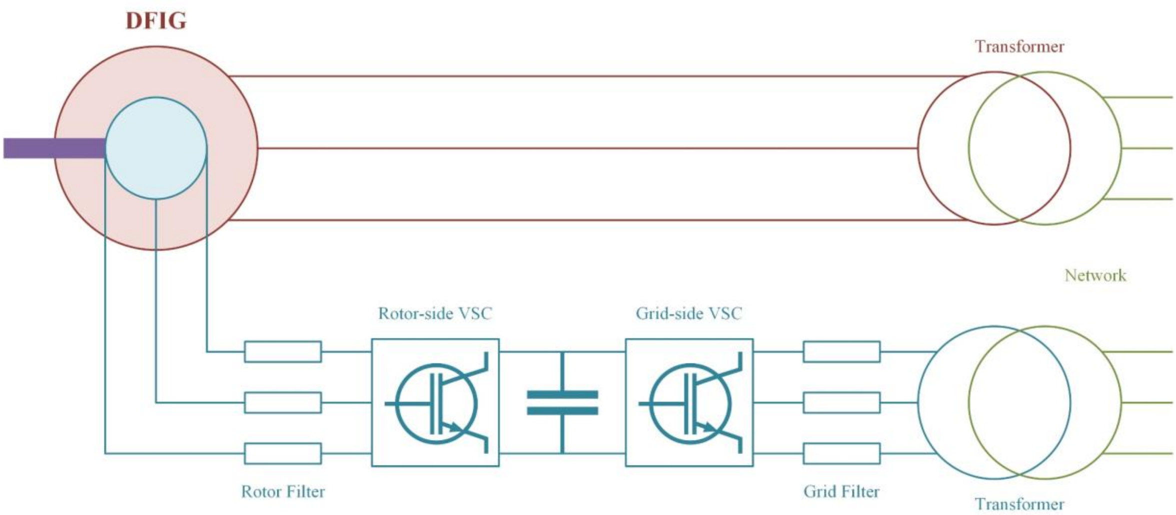

Figure 2.6 Layout of the stator and rotor sides of the DFIG

The DFIG consists of a stator directly connected to the grid and a rotor fed by a bidirectional converter, as shown in **Figure 2.6** [109]. The back-to-back converter includes two voltage source converters (VSCs) on the rotor and grid sides, which assures variable speed excitation at constant grid frequency and voltage [19]. The VSCs achieve the alternating-current (AC) re-modulation of the generator rotor through the direct-current (DC) link for speed matching, which changes voltage and frequency according to power and speed requirements.

The grid constantly fixes the stator voltage, while the rotor voltage depends on the intended output and rotation [110]. **Eqs. (44)(45)** compute the angular frequencies of the stator and rotor, respectively [19].

$$\omega_s = \omega_e \tag{44}$$

$$\omega_r = \omega_j - \omega_g \tag{45}$$

$$\omega_g = p\omega_m \approx pn_g\omega_d \tag{46}$$

where

ω_s: the angular frequency of the stator windings (rad/s)

ω_e: the synchronous frequency of the grid (rad/s)

ω_r: the effective frequency in the rotor-side equivalent circuit (rad/s)

ω_j: the injected frequency of the rotor windings for excitation (rad/s)

ω_g: the equivalent speed of the generator rotor (rad/s)

p: is the number of pairs of poles (dimensionless)

Another vital operation term is the slip, denoted as s_l, which defines the relationship between injected frequency and rotor rotation, as given by **Eq. (47)** [19].

$$s_l = \frac{\omega_j - \omega_g}{\omega_j} \tag{47}$$

The sign of the slip determines the operating modes for the machine:

- $\omega_j < \omega_g,\ s_l < 0$: hypersynchronous operation, generator mode
- $\omega_j = \omega_g,\ s_l = 0$: synchronous operation, no power flow
- $\omega_j > \omega_g,\ s_l > 0$: subsynchronous operation, motor mode

For a designated slip, **Eq. (48)** derives the injected frequency of the VSC modulation. **Eq. (49)** provides a way to estimate the rotor frequency according to the speed reference.

$$\omega_j = \frac{\omega_g}{1 - s_l} \tag{48}$$

$$\omega_r = \frac{s_l}{1 - s_l}\omega_g \tag{49}$$

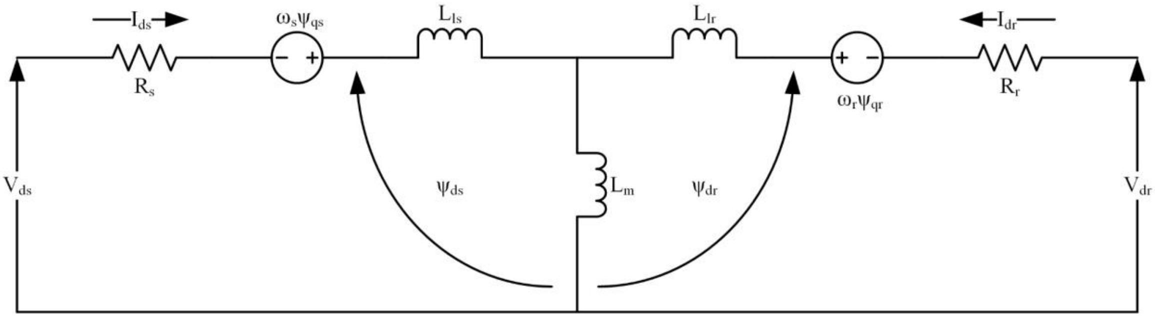

Figure 2.7a Equivalent d-axis circuit

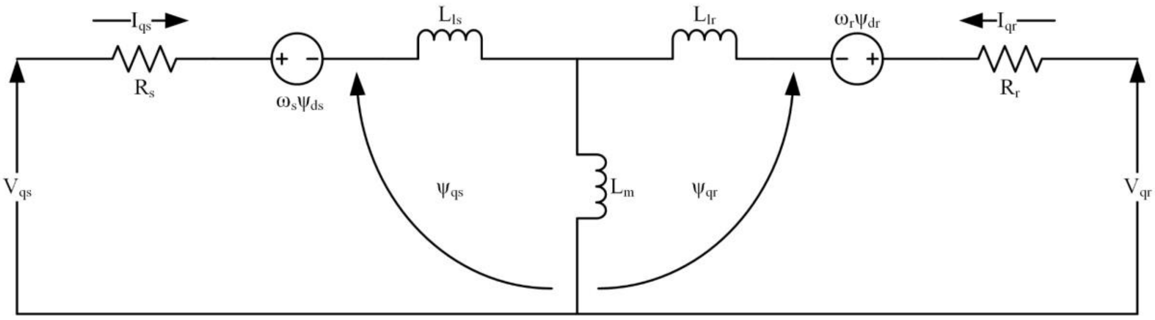

Figure 2.7b Equivalent q-axis circuit

Figure 2.7 Equivalent dq-axis circuits of the DFIG

According to the dq-axis analysis, DFIG modelling can be carried out on separate direct and quadrature axes, as shown in **Figure 2.7** [111]. The equivalent circuit implies a causal relation from the rotor side to the stator side through the flux linkages of the mutual inductance. In addition, the reference frame aligns with the d-axis, and the DFIG is assumed to be a star-connected balanced induction machine [112].

For simplicity, the above circuits can use a straight form of the equivalent inductances rather than the leakage inductances in the DFIG model, as **Eqs. (50)(51)** [113]. Meanwhile, **Eq. (52)** introduces a constant of the linkage coefficient σ [113] to simplify some expressions.

$$L_s = L_{ls} + L_m \tag{50}$$

$$L_r = L_{lr} + L_m \tag{51}$$

$$\sigma = 1 - \frac{L_m^2}{L_s L_r} \tag{52}$$

where

L_s, L_r: the equivalent inductances of the stator and rotor (H)

L_{ls}, L_{lr}: the leakage inductances of the stator and rotor (H)

L_m: the mutual inductance (H)

The dq-axis dynamic model assumes all sinusoidal variables in the stationary frame to be DC quantity in the synchronous frame. This results in a concise expression of the DFIG dynamics. **Eqs. (53)~(56)** are a set of differential equations representing the dq-axis components of the flux linkages of the stator and rotor [111].

$$\frac{d\psi_{ds}}{dt} = V_{ds} - R_s I_{ds} + \omega_s \psi_{qs} \tag{53}$$

$$\frac{d\psi_{qs}}{dt} = V_{qs} - R_s I_{qs} - \omega_s \psi_{ds} \tag{54}$$

$$\frac{d\psi_{dr}}{dt} = V_{dr} - R_r I_{dr} + \omega_r \psi_{qr} \tag{55}$$

$$\frac{d\psi_{qr}}{dt} = V_{qr} - R_r I_{qr} - \omega_r \psi_{dr} \tag{56}$$

where

ψ_{ds}, ψ_{qs}: the dq-axis components of the flux linkages of the stator (wb)

ψ_{dr}, ψ_{qr}: the dq-axis components of the flux linkages of the rotor (wb)

V_{ds}, V_{qs}: the dq-axis voltages of the stator (V)

V_{dr}, V_{qr}: the dq-axis voltages of the rotor (V)

I_{ds}, I_{qs}: the dq-axis currents of the stator (A)

I_{dr}, I_{qr}: the dq-axis currents of the rotor (A)

R_s, R_r: the stator and rotor resistances (Ω)

According to the expressions of the flux linkages in terms of the currents, **Eqs. (57)~(60)** determine the dq-axis currents of the stator and rotor [114].

$$I_{ds} = \frac{1}{\sigma L_s}\left(\psi_{ds} - \frac{L_m}{L_r}\psi_{dr}\right) \tag{57}$$

$$I_{qs} = \frac{1}{\sigma L_s}\left(\psi_{qs} - \frac{L_m}{L_r}\psi_{qr}\right) \tag{58}$$

$$I_{dr} = \frac{1}{\sigma L_r}\left(\psi_{dr} - \frac{L_m}{L_s}\psi_{ds}\right) \tag{59}$$

$$I_{qr} = \frac{1}{\sigma L_r}\left(\psi_{qr} - \frac{L_m}{L_s}\psi_{qs}\right) \tag{60}$$

Meanwhile, **Eqs. (61)(62)** record the stator and rotor phases, which are helpful to the inverse Park transformation.

$$\theta_s = \int_0^T \omega_s \, dt + \theta_{s_0} \tag{61}$$

$$\theta_j = \int_0^T \omega_j \, dt + \theta_{j_0} \tag{62}$$

where

θ_s, θ_j: the stator and rotor phases (rad)

θ_{s_0}, θ_{j_0}: the initial phase of the stator and injected frequencies (rad)

Each update of the DFIG states follows a sequence indicated in **Figure 2.8** to perform **Eqs. (53)~(56)** and **(57)~(60)**, where the flux linkages and currents are state variables, and the voltages are controller variables [110]. It notes that the grid connection enforces the stator voltages (V_{ds}, V_{qs}), which are external parameters for the induction machine [115]. When performing the Park transformation (abc-dq), a phase lock loop (PLL) in power electronics can eliminate phase differences caused by random initial angle positions.

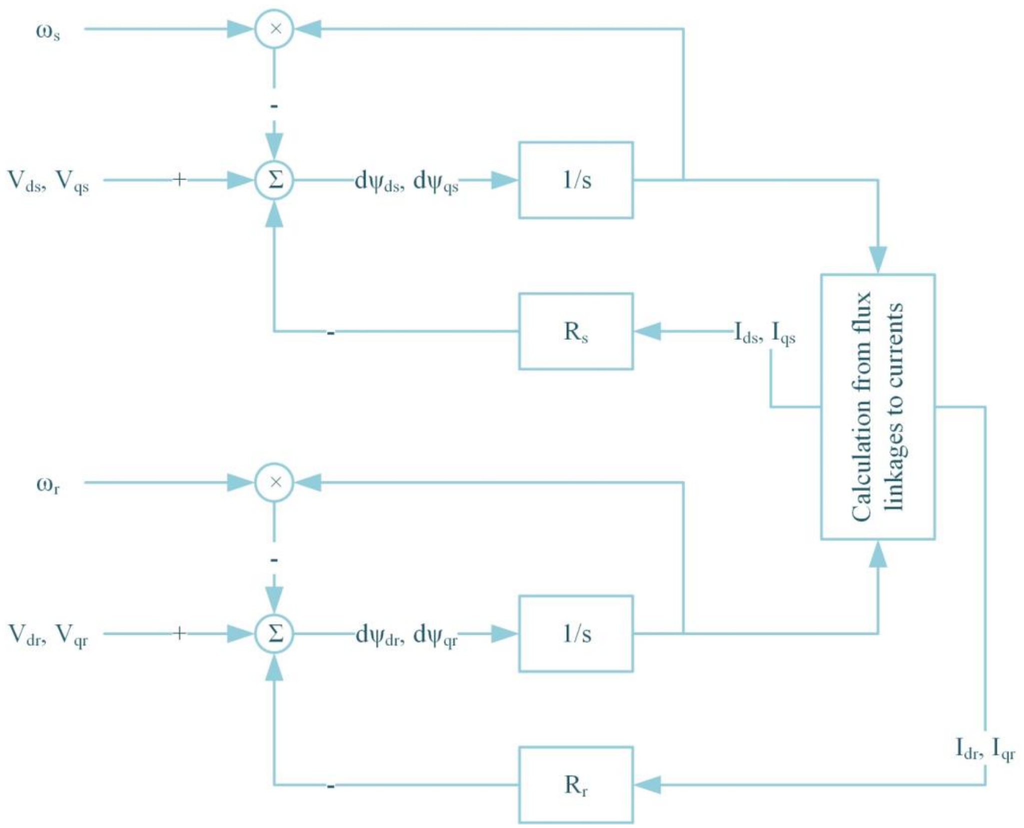

Figure 2.8 Procedures for updating the flux linkages and currents of the DFIG

At any arbitrary time, **Eqs. (63)~(67)** can calculate the electromagnetic torque and electric power of the DFIG [19]. The torque expression involves a constant generator efficiency to consider mechanical losses.

$$T_e = \frac{3p}{k_g} L_m \left(I_{qs} I_{dr} - I_{ds} I_{qr} \right) \tag{63}$$

$$P_s = 3\left(V_{ds} I_{ds} + V_{qs} I_{qs} \right) \tag{64}$$

$$Q_s = 3\left(V_{qs} I_{ds} - V_{ds} I_{qs} \right) \tag{65}$$

$$P_r = 3\left(V_{dr} I_{dr} + V_{qr} I_{qr} \right) \tag{66}$$

$$Q_r = 3\left(V_{qr} I_{dr} - V_{dr} I_{qr} \right) \tag{67}$$

where

k_g: the efficiency of power transfer (dimensionless)

P_s, Q_s: the active and reactive powers outputted by the stator windings (W, VAR)

P_r, Q_r: the active and reactive powers consumed by the rotor windings (W, VAR)

The injected frequency differs from the stator frequency, but **Eqs. (53)~(60)** naturally calculate the dynamics of the induction machine. Therefore, the DFIG needs **Eqs. (68)(69)** to update the rotor dq-axis currents in the equivalent circuit. The torque and power calculations and the control measurements take actual rotor currents. The DFIG simulator only stores actual currents or equivalent currents and applies **Eqs. (68)(69)** to recover the others.

$$I_{dr}^{equ} = K_f I_{dr}^{act} \tag{68}$$

$$I_{qr}^{equ} = K_f I_{qr}^{act} \tag{69}$$

$$K_f = \frac{\omega_s}{\omega_j} \tag{70}$$

where

I_{dr}^{equ}, I_{qr}^{equ}: the equivalent rotor currents used in **Eqs. (53)~(60)** (A)

I_{dr}^{act}, I_{qr}^{act}: the actual rotor currents used in **Eqs. (63)~(67)** (A)

As the DFIG normally draws the rotor currents and feeds the stator currents on the same grid, the output power of a DFIG should be the net value of stator output and rotor consumption, as **Eqs. (71)(72)**.

$$P_n = P_s - P_r \tag{71}$$

$$Q_n = Q_s - Q_r \tag{72}$$

where

P_n, Q_n: the net value of the active and reactive outputs (W, VAR)

Eqs. (53)~(56) (57)~(60) (63)~(67) constitute a real-time DFIG model that the simulator uses to update the DFIG state under the vector control of voltage. However, the turbine controller only cares about components that can affect the DFIG torque. Therefore, the DFIG control model can make some assumptions to reduce the number of variables. The stator voltage is usually constant due to grid connection, so the control model of a DFIG only pays attention to the rotor side, as **Eqs. (73)(74)** [116]. Meanwhile, the stator fluxes and rotor currents determine the electromagnetic torque of the DFIG, yielding **Eq. (75)** [117].

$$V_{dr} = R_r I_{dr} - \omega_r \sigma L_r I_{qr} + \sigma L_r \dot{I}_{dr} \tag{73}$$

$$V_{qr} = R_r I_{qr} + \omega_r \sigma L_r I_{dr} + \sigma L_r \dot{I}_{qr} + \omega_r \frac{L_m}{L_s} \psi_{ds} \tag{74}$$

$$Q_e = \frac{3p}{k_g} \frac{L_m}{L_s} \left(\psi_{qs} I_{dr} - \psi_{ds} I_{qr} \right) = -\frac{3p}{k_g} \frac{L_m}{L_s} \frac{V_b}{\omega_e} I_{qr} \tag{75}$$

where

V_b: the voltage rating or base voltage (V)

In the above, the d-axis flux linkage of the stator has a relation ($\psi_{ds} = \frac{V_b}{\omega_e}$) at the steady state. **Eqs. (73)(74)** describe the DFIG state when regulating the rotor voltages. The condition ($\psi_{qs} = 0$) is due to the flux orientation of the stator. Therefore, the q-axis current of the rotor entirely decides the generator torque in the control model.

Since the DFIG reference generally appears as an output target at a nominated speed, **Eqs. (76)(77)** estimate the power references of the DFIG stator according to power prediction [54].

$$P_s^{ref} = \frac{P_m}{1 - s} \approx \frac{\omega_e}{p n_g \omega_d^{ref}} k_g P_d^{ref} \tag{76}$$

$$Q_s^{ref} = K_{pf} P_s^{ref} \tag{77}$$

$$K_{pf} = \tan(\arccos(pf)) \tag{78}$$

where

P_s^{ref}, Q_s^{ref}: the reference of the active and reactive powers of the stator (W, VAR)

P_d^{ref}: the power prediction estimated by an aerodynamic model (W)

ω_d^{ref}: the speed reference of the main shaft (rad/s)

pf: the power factor (dimensionless)

The steady-state expression of the DFIG rotor currents derives from the power references (P_s^{ref}, Q_s^{ref}) with neglecting the stator resistance, which yields **Eqs. (79)(80)** [35].

$$I_{dr}^{ref} = \frac{L_s}{L_m}\left(\frac{V_b}{\omega_e L_s} - \frac{Q_s^{ref}}{3V_b}\right) \tag{79}$$

$$I_{qr}^{ref} = -\frac{L_s}{L_m}\frac{P_s^{ref}}{3V_b} \tag{80}$$

where

I_{dr}^{ref}, I_{qr}^{ref}: the reference of the dq-axis currents of the rotor (A)

Meanwhile, a small-signal model also requires the steady-state estimation of the rotor dq-axis voltages (V_{dr}^{ref}, V_{qr}^{ref}), which derives from inserting the current references (I_{dr}^{ref}, I_{qr}^{ref}) into **Eqs. (73)(74)** and neglecting the time-derivative terms.

In an actual operation, a slight difference between power estimation and output power can cause unstable rotation due to unbalanced torque conditions. It is impossible to thoroughly eliminate the model prediction error due to wind measurement and aerodynamic calculation. Therefore, a s-curve like compensator f_s [118] corrects the control input of the rotor q-axis current according to the normalized speed error, as shown in **Eq. (81)**.

$$I_{qr}^{sig} = \left(1 + f_s(\Delta\omega_d)\right) I_{qr}^{ref} \tag{81}$$

$$f_s(\Delta\omega_d) = L_{fs}\left(\frac{2}{1+\exp\left(-K_{fs}\Delta\omega_d\right)} - 1\right) \tag{82}$$

$$\Delta\omega_d = \frac{\omega_d - \omega_d^{ref}}{\omega_d^{ref}} \tag{83}$$

where

I_{qr}^{sig}: the corrected signal of the q-axis input reference of the DFIG (A)

$\Delta\omega_d$: the normalized error of the rotor speed of the main shaft (dimensionless)

L_{fs}: the upper and lower boundary for compensation (tuneable)

K_{fs}: the compensation slope (tuneable)

Figure 2.9 draws an example of f_s with $K_{sf} = 20$ and $L_{sf} = 0.5$. The compensator reduces generator production to accelerate rotor speed when below target or increases generator torque to slow down for overspeeding.

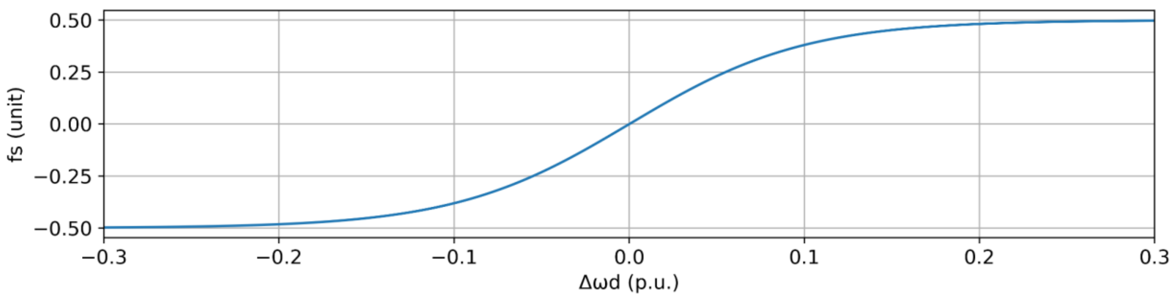

Figure 2.9 Example of the speed-current compensator

The DFIG allows a local transformer at the rotor side to boost rotor voltages and reduce rotor currents for higher efficiency of rotor excitation. To apply the conventional analysis of induction machines, we introduce two virtual transformers to convert the measured and scaled values. They are not physical devices but update the current and voltage signals with a relation of **Eq. (84)** [110].

$$n_r = \frac{I_{sca}}{I_{mea}} = \frac{V_{mea}}{V_{sca}} \tag{84}$$

where

n_r; the turns ratio of the DFIG rotor (dimensionless)

V_{mea}, I_{mea}: the measured voltage and current in the real circuit (V, A)

V_{sca}, I_{sca}: the scaled voltage and current in the equivalent circuit (V, A)

Since oversized rotor voltage inputs probably cause surge currents that damage the DFIG, **Eqs. (85)(86)**, as a limiter of lower and upper boundaries, constrain the rotor voltage inputs issued by the controller. **Eqs. (85)(86)** derives from the combination of **Eqs. (73)(74)(79)(80)** with ignoring the small and derivative items.

$$V_{dr}^{lim} = R_r \frac{V_b}{\omega_e L_m} \pm K_v R_r \frac{L_s}{L_m} \frac{K_{pf}^{max} P_b}{3V_b} \tag{85}$$

$$V_{qr}^{lim} = \pm K_v R_r \frac{L_s}{L_m} \frac{P_b}{3V_b} \tag{86}$$

where

V_{dr}^{lim}, V_{qr}^{lim}: the limiter of the rotor dq-axis voltages (V)

K_v: the coefficient of voltage tolerance (tuneable, ≥1)

K_{pf}^{max}: the maximum coefficient of reactive power (tuneable, ≥0)

P_b: the power rating or base power (W)

2.3.2 Permanent Magnet Synchronous Generator

Modern WTs are increasing capacity and size, but DFIGs cannot support such generation capacity due to intrinsic rotor excitation and complicated electrical structure. Besides, current WTs prefer direct-drive technologies to avoid the effects of gearbox transmission and shaft torsion. Compared with the DFIG, the PMSG excitation results from its permanent magnet, which gives rise to a compact generator rotor and allows more pole pairs [7]. Thus, the PMSG has a much easier implementation of direct drive, and modern large offshore WTs are almost PMSG-based. This study also involves the PMSG as an RSC actuator.

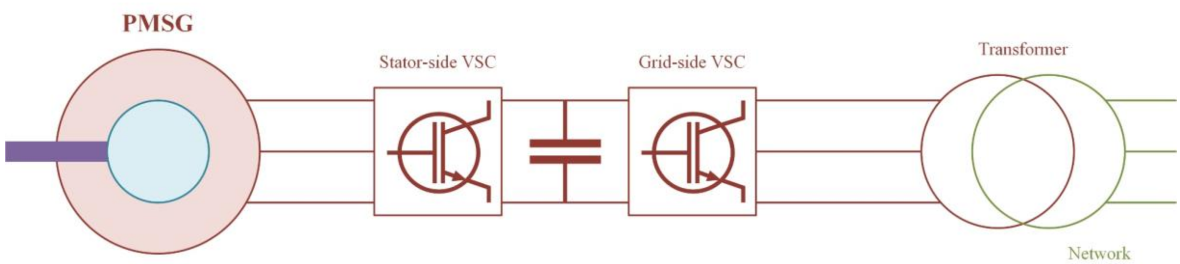

Figure 2.10 Layout of the stator side of the PMSG

The PMSG, known as the brushless DC motor, does not need to control its rotor side, so the PMSG regulation only relies on the stator-side control, as shown in **Figure 2.10** [110]. The PMSG also has a back-to-back converter that re-modulates the stator frequency for grid synchronization. Compared with the DFIG, the VSCs of the PMSG are full-scale due to heavy generation load, and the PMSG does not leak harmonic frequencies to the grid [17].

The stator-side frequency depends on rotor speed for synchronous generation, expressed in **Eq. (87)** [8]. Therefore, the stator-side VSC regulates AC voltage with speed variation and feeds DC voltage to the grid side. After that, the grid-side VSC re-modulates AC voltage from DC voltage with the grid frequency. Accordingly, the stator currents flow from the machine to the grid.

$$\omega_s = p\omega_m \tag{87}$$

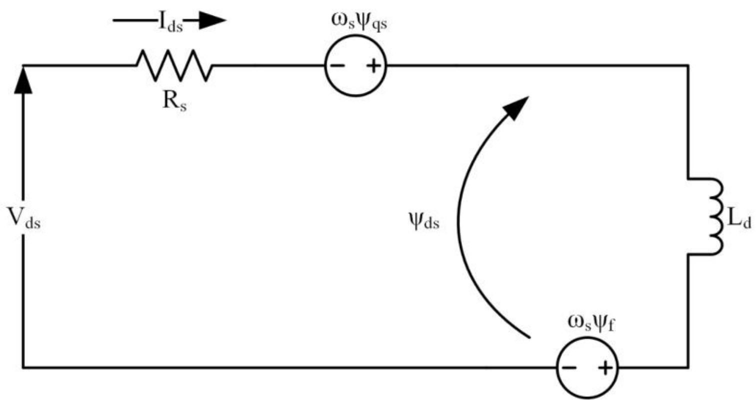

Figure 2.11a Equivalent d-axis circuit

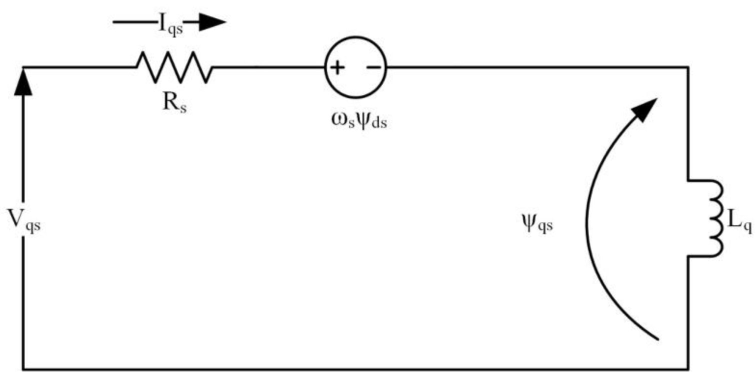

Figure 2.11b Equivalent q-axis circuit

Figure 2.11 Equivalent dq-axis circuits of the PMSG

The stator flux linkages result from the flux linkages of the permanent magnet and the mutual inductance currents [8]. The transformation from stationary to synchronous frame yields the equivalent dq-axis circuits of the PMSG on the stator side, as displayed in **Figure 2.11** [8]. The d-axis circuit reflects the variation of the flux linkages of the permanent magnet, and the rotor magnetic axis always aligns with the d-axis.

Similar to the circuit of the DFIG stator, **Eqs. (88)(89)** determine the dq-axis flux linkages of the PMSG stator [110].

$$\frac{d\psi_{ds}}{dt} = V_{ds} - R_s I_{ds} + \omega_s \psi_{qs} \tag{88}$$

$$\frac{d\psi_{qs}}{dt} = V_{qs} - R_s I_{qs} - \omega_s \psi_{ds} \tag{89}$$

Since the stator flux linkages impose on the dq-axis inductances, **Eqs. (90)(91)** determine the stator dq-axis currents [110].

$$I_{ds} = \frac{\psi_{ds} - \psi_f}{L_d} \tag{90}$$

$$I_{qs} = \frac{\psi_{qs}}{L_q} \tag{91}$$

where

L_d: the d-axis inductance (H)

L_q: the q-axis inductance (H)

ψ_f: the peak flux linkage produced by the permanent magnet (wb)

Meanwhile, **Eq. (92)** records the stator phase, which converts the dq-axis signals to the abc-axis signals by the inverse Park transformation.

$$\theta_s = \int_0^T \omega_s \, dt + \theta_{s_0} \tag{92}$$

The PMSG model takes the flowchart (**Figure 2.12**) to update **Eqs. (88)~(91)**, in which the stator voltages are solely controllable variables. Compared with the DFIG, the coupling complexity of the calculation of the PMSG is much lower due to the fixed magnet flux.

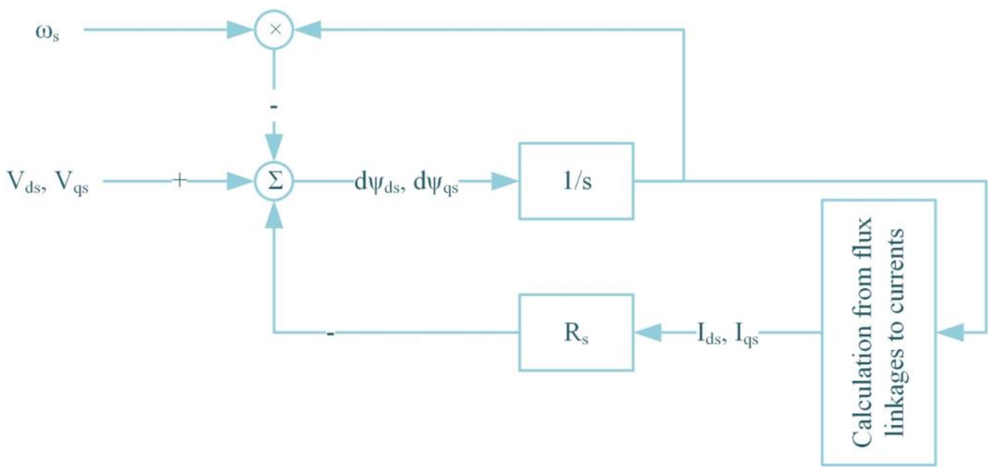

Figure 2.12 Procedures for updating the flux linkages and currents of the PMSG

Since the above equivalent model originally solves the motor dynamics, the torque and power of the PMSG should be negative to reverse motor mode to generator mode, as given in **Eqs. (93)~(95)** [20].

$$T_e = -\frac{3p}{k_g}\left(\psi_{ds}I_{qs} - \psi_{qs}I_{ds}\right) \tag{93}$$

$$P_s = -3\left(V_{ds}I_{ds} + V_{qs}I_{qs}\right) \tag{94}$$

$$Q_s = -3\left(V_{qs}I_{ds} - V_{ds}I_{qs}\right) \tag{95}$$

Since the PMSG has no rotor consumption for magnetizing, its net production equals the stator output, as **Eqs. (96)(97)**.

$$P_n = P_s \tag{96}$$

$$Q_n = Q_s \tag{97}$$

The control model of a PMSG only needs to consider the voltage control of the stator according to the stator currents [119], as given in **Eqs. (98)(99)**. Based on the current-voltage control, **Eq. (100)** estimates the resultant PMSG torque [45].

$$V_{ds} = R_s I_{ds} - \omega_s L_q I_{qs} + L_d \dot{I}_{ds} \tag{98}$$

$$V_{qs} = R_s I_{qs} + \omega_s L_d I_{ds} + L_q \dot{I}_{qs} + \omega_s \psi_f \tag{99}$$

$$Q_e = \frac{3p}{k_g}\left(\left(L_d - L_q\right)I_{ds} - \psi_f\right)I_{qs} = -\frac{3p}{k_g}\psi_f I_{qs} \tag{100}$$

This study only discusses symmetrical PMSGs that have equal dq-axis inductances, i.e.,

$L_d = L_q$, so their torque control entirely depends on the q-axis current.

Since the PMSG has no rotor loss for excitation, the PMSG can ideally convert the mechanical power received by the main shaft to the stator output, which accounts for the power estimation of **Eqs. (101)(102)**.

$$P_s^{ref} \approx k_g P_d^{ref} \tag{101}$$

$$Q_s^{ref} = K_{pf} P_s^{ref} \tag{102}$$

The normal operation of the PMSG requires the power factor to be a unit for two reasons. The permanent magnet is in charge of excitation, and the PMSG should ensure a minimum d-axis voltage.

Given that the d-axis voltage is negligible at steady state, **Eqs. (103)(104)** calculate the current references of the PMSG [120]. Therefore, the regulation objective of the active power control converts to an equivalent form of the q-axis current control, and the d-axis current control is responsible for the reactive power control.

$$I_{ds}^{ref} \approx -\frac{Q_s^{ref}}{3V_g} \tag{103}$$

$$I_{qs}^{ref} \approx -\frac{P_s^{ref}}{3V_g} \tag{104}$$

$$V_g = \psi_f \omega_s^{ref} \approx \psi_f p n_g \omega_d^{ref} \tag{105}$$

where

I_{ds}^{ref}, I_{qs}^{ref}: the reference of the stator currents (A)

As indicated by **Eq. (105)**, the stator voltage V_g is proportional to rotor speed, which means that an operation of high speed requires strong voltage tolerance. Therefore, this limits the PMSG operation to a low-speed range. Besides, substituting the current references and neglecting the time-derivative items in **Eqs. (98)(99)** can determine the voltage references (V_{ds}^{ref}, V_{qs}^{ref}).

The PMSG also needs a speed compensator for real-time correction, which can reduce the speed error caused by wind fluctuation and internal error. Since the q-axis current determines the PMSG torque, we place the speed compensator on the stator q-axis current as **Eq. (106)**.

$$I_{qs}^{sig} = \left(1 + f_s(\Delta\omega_d)\right) I_{qs}^{ref} \tag{106}$$

where

I_{qs}^{sig}: the corrected signal of the q-axis input reference of the PMSG (A)

The PMSG protection also needs a limiter to constrain the stator voltage inputs, as given in **Eqs. (107)(108)**. The PMSG limiter only considers voltage tolerance at the synchronous speed because of its linear increase with rotor speed.

$$V_{ds}^{lim} = \pm K_v K_{pf}^{max} \psi_f \omega_e \tag{107}$$

$$V_{qs}^{lim} = \pm K_v \psi_f \omega_e \tag{108}$$

where

V_{ds}^{lim}, V_{qs}^{lim}: the limiter of the stator dq-axis voltages (V)

2.4 Pitch and Yaw Systems

Most pitching [121] and yawing [122] systems in WECSs refer to the servo system that tracks an input signal. The pitch and yaw movements behave as a model of the dynamic system with amplitude and output limitations, which the first-order model can represent [123]. This simplification of pitch and yaw variations assumes an angle servo system has sufficient drive torque to eliminate external forcing influences. **Eqs. (109)(110)** describe the response of the pitch and yaw servos (β, γ), respectively. It notes that blades are under a collective pitch control and have an identical response [124].

$$\dot{\beta} = \tau_\beta^{-1}(\beta_i - \beta) \tag{109}$$

$$\dot{\gamma} = \tau_\gamma^{-1}(\gamma_i - \gamma) \tag{110}$$

where

τ_β: the time constant of the pitch servo (s)

τ_γ: the time constant of the yaw servo (s)

β_i: the pitch input (rad)

γ_i: the yaw input (rad)

One characteristic of pitch [30] and yaw [122] is that both have movement rate constraints due to maximum servo load. The pitch servo has double-side position boundaries that normally satisfy a condition ($\beta \in (0°, 90°)$). The yaw system can navigate at any angle around the compass circle.

2.5 Tower Motion

Hub thrust excites tower fore-aft oscillation, affecting tower fatigue and safety [125]. Since the BEM theory demonstrates that pitch and speed actions can suppress fore-aft motion [10], the control system has an extra objective of damping tower oscillation. Considering damped harmonics, the tower fore-aft motion behaves as a second-order system **Eq. (111)** [126].

$$M_{tm}\ddot{x}_m + D_{tm}\dot{x}_m + K_{tm}x_m = T_d \tag{111}$$

where

M_{tm}: the modal mass of the tower (kg)

D_{tm}: the modal damping of the tower (N·s/m)

K_{tm}: the model stiffness of the tower (N/m)

x_m: the fore-aft displacement of the tower top (m)

The analysis of tower fore-aft damping focuses on fore-aft velocity and ignores acceleration due to rapid change. Besides, the tower displacement has a steady-state bending position for a given thrust, as in **Eq. (112)**.

$$x_m^{ref} = \frac{T_d^{ref}}{K_{tm}} \tag{112}$$

where

x_m^{ref}: the steady-state position of the tower top bending (m)

T_d^{ref}: the thrust prediction of an aerodynamic model (N)

Regarding the control model of tower fore-aft damping, **Eq. (113)** can produce effective damping, in which the negative sign accounts for counteraction [127].

$$D_{tm}\dot{x}_m = -\Delta T_d \tag{113}$$

2.6 Small Signal Analysis

Model-based control requires a dynamic model for trajectory estimation, so the combination of **Eqs. (34)(73)(74)(75)(109)(110)(113)** yield a fully-coupled control model (**Eq. (114)**) for the DFIG-based WT, which simultaneously describes shaft rotation, circuit response, generator torque, pitch position, yaw movement, and tower displacement.

$$\begin{cases} J_t\dot{\omega}_d = Q_d - n_g Q_e - D_t\omega_d \\ V_{dr} = R_r I_{dr} - \omega_r \sigma L_r I_{qr} + \sigma L_r \dot{I}_{dr} \\ V_{qr} = R_r I_{qr} + \omega_r \sigma L_r I_{dr} + \sigma L_r \dot{I}_{qr} + \omega_r \dfrac{L_m}{L_s}\dfrac{V_b}{\omega_e} \\ Q_e = -\dfrac{3p}{k_g}\dfrac{L_m}{L_s}\dfrac{V_b}{\omega_e} I_{qr} \\ \dot{\beta} = \tau_\beta^{-1}(\beta_i - \beta) \\ \dot{\gamma} = \tau_\gamma^{-1}(\gamma_i - \gamma) \\ \dot{x}_m = -D_{tm}^{-1}\Delta T_d \end{cases} \tag{114}$$

By substituting **Eqs. (98)(99)(100)** into the above generator-related expressions, **Eq. (115)** represents the control model of the PMSG-based WT.

$$\begin{cases} J_t\dot{\omega}_d = Q_d - n_g Q_e - D_t\omega_d \\ V_{ds} = R_s I_{ds} - \omega_s L_q I_{qs} + L_d \dot{I}_{ds} \\ V_{qs} = R_s I_{qs} + \omega_s L_d I_{ds} + L_q \dot{I}_{qs} + \omega_s \psi_f \\ Q_e = -\dfrac{3p}{k_g}\psi_f I_{qs} \\ \dot{\beta} = \tau_\beta^{-1}(\beta_i - \beta) \\ \dot{\gamma} = \tau_\gamma^{-1}(\gamma_i - \gamma) \\ \dot{x}_m = -D_{tm}^{-1}\Delta T_d \end{cases} \tag{115}$$

However, **Eqs. (114)(115)** have two issues in applications: the symbol of rotor torque does not explicitly reveal aerodynamic nonlinearities, and the q-axis perturbation terms ($\omega_r \frac{L_m}{L_s}\frac{V_b}{\omega_e}$ or $\omega_s\psi_f$) result in a biased condition. Hence, we apply a small signal analysis [128] to obtain a linearized expression to address such issues. Supposing that a dynamic system has the characteristics of **Eq. (116)**, the Taylor expansion (**Eq. (117)**) performs the small signal analysis at a steady working point [37].

$$\dot{x} = f(x) \tag{116}$$

$$\dot{x} = f(x_0) + \left.\frac{\partial f}{\partial x}\right|_{x_0} (x - x_0) + \cdots \tag{117}$$

where

x_0: a balanced system state

For any balanced system, the constant term ($f(x_0)$) always equals zero, which leads to an incremental expression (**Eq. (118)**) ignoring high-order harmonic components, which approximates a linear system as **Eq. (119)** [37].

$$\Delta x = x - x_0 \tag{118}$$

$$\Delta \dot{x} = J_0 \Delta x \tag{119}$$

$$J_0 = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{120}$$

where

J_0: the Jacobian matrix of the system

It notes that the control system should update the Jacobian matrix when the working condition changes. Thus, we rewrite the nonlinear model of the DFIG-based WT (**Eq. (114)**) to a small-signal model as **Eq. (121)**.

$$\begin{cases} \Delta\dot{\omega}_d = J_t^{-1}\left(\Delta Q_d - n_g \Delta Q_e - D_t \Delta\omega_d\right) \\ \Delta Q_d = \dfrac{\partial Q_d}{\partial \omega_d}\Delta\omega_d + \dfrac{\partial Q_d}{\partial \beta}\Delta\beta \\ \Delta Q_e = -\dfrac{3p}{k_g}\dfrac{L_m}{L_s}\dfrac{V_b}{\omega_e}\Delta I_{qr} \\ \Delta\dot{I}_{dr} = -\dfrac{R_r}{\sigma L_r}\Delta I_{dr} + \omega_r \Delta I_{qr} + \dfrac{\Delta V_{dr}}{\sigma L_r} \\ \Delta\dot{I}_{qr} = -\omega_r \Delta I_{dr} - \dfrac{R_r}{\sigma L_r}\Delta I_{qr} + \dfrac{\Delta V_{qr}}{\sigma L_r} \\ \Delta\dot{\beta} = \tau_\beta^{-1}(\Delta\beta_i - \Delta\beta) \\ \Delta\dot{\gamma} = \tau_\gamma^{-1}(\Delta\gamma_i - \Delta\gamma) \\ \Delta x_m = -D_{tm}^{-1}\Delta T_d \\ \Delta T_d = -\dfrac{\partial T_d}{\partial \omega_d}\Delta\omega_d - \dfrac{\partial T_d}{\partial \beta}\Delta\beta \end{cases} \tag{121}$$

where

$\frac{\partial Q_d}{\partial \omega_d}$, $\frac{\partial Q_d}{\partial \beta}$: the torque sensitivities for rotor and pitch variations (N·m/(rad/s), N·m/rad)

$\frac{\partial T_d}{\partial \omega_d}$, $\frac{\partial T_d}{\partial \beta}$: the thrust sensitivities for rotor and pitch variations (N/(rad/s), N/rad)

By replacing the generator part in **Eq. (121)**, **Eq. (122)** represents the small-signal model of the PMSG-based WT.

$$\begin{cases} \Delta\dot{\omega}_d = J_t^{-1}\left(\Delta Q_d - n_g \Delta Q_e - D_t \Delta\omega_d\right) \\ \Delta Q_d = \dfrac{\partial Q_d}{\partial \omega_d}\Delta\omega_d + \dfrac{\partial Q_d}{\partial \beta}\Delta\beta \\ \Delta Q_e = -\dfrac{3p}{k_g}\psi_f \Delta I_{qs} \\ \Delta\dot{I}_{ds} = -\dfrac{R_s}{L_d}\Delta I_{ds} + \omega_s \dfrac{L_q}{L_d}\Delta I_{qs} + \dfrac{\Delta V_{ds}}{L_d} \\ \Delta\dot{I}_{qs} = -\omega_s \dfrac{L_d}{L_q}\Delta I_{ds} - \dfrac{R_s}{L_q}\Delta I_{qs} + \dfrac{\Delta V_{qs}}{L_q} \\ \Delta\dot{\beta} = \tau_\beta^{-1}(\Delta\beta_i - \Delta\beta) \\ \Delta\dot{\gamma} = \tau_\gamma^{-1}(\Delta\gamma_i - \Delta\gamma) \\ \Delta x_m = -D_{tm}^{-1}\Delta T_d \\ \Delta T_d = -\dfrac{\partial T_d}{\partial \omega_d}\Delta\omega_d - \dfrac{\partial T_d}{\partial \beta}\Delta\beta \end{cases} \tag{122}$$

The first three of **Eqs. (121)(122)** forms the linearized kinetic governing equations of shaft rotation (**Eqs. (123)(124)**) for DFIG-based and PMSG-based WTs, respectively. These governing equations calculate the rotor variation according to the pitch and q-axis current alternations. Compared with the indirect torque control [52], **Eqs. (123)(124)** avoid cascaded torque transfer functions, contributing to a simplified direct control of the generator and pitch servo.

$$\Delta\dot{\omega}_d = J_t^{-1}\left(\left(\frac{\partial Q_d}{\partial \omega_d} - D_t\right)\Delta\omega_d + \frac{\partial Q_d}{\partial \beta}\Delta\beta + n_g\frac{3p}{k_g}\frac{L_m}{L_s}\frac{V_b}{\omega_e}\Delta I_{qr}\right) \tag{123}$$

$$\Delta\dot{\omega}_d = J_t^{-1}\left(\left(\frac{\partial Q_d}{\partial \omega_d} - D_t\right)\Delta\omega_d + \frac{\partial Q_d}{\partial \beta}\Delta\beta + n_g\frac{3p}{k_g}\psi_f\Delta I_q\right) \tag{124}$$

Eqs. (121)(122) introduce four force sensitivities to be determined. A model-based controller requires specific values to initialize matrices. As one of the novelties, this study proposes an ANN-based approach to update their values online (later discussed in **Eq. (155)**), which conveniently and accurately reflects aerodynamic responses.

By separating state and control variables, **Eqs. (121)(122)** yields a general state-space model as **Eq. (125)**. Compared with fully partial derivative models, such as Ref. [37], our model has fewer variables by introducing the sensitivities. Meanwhile, a controller can access ultimate control inputs, i.e., generator voltages, pitch servo input, and yaw servo input.

$$\Delta\dot{x} = A\Delta x + B\Delta u \tag{125}$$

$$\Delta x = [\Delta x_m \quad \Delta\omega_d \quad \Delta I_{dx} \quad \Delta I_{qx} \quad \Delta\beta \quad \Delta\gamma]^T \tag{126}$$

$$\Delta u = [\Delta V_{dx} \quad \Delta V_{qx} \quad \Delta\beta_i \quad \Delta\gamma_i]^T \tag{127}$$

where

Δx: the state vector

Δu: the input vector

A: the state matrix

B: the input matrix

ΔI_{dx}, ΔI_{qx}: the measurement of the current deviations

ΔV_{dx}, ΔV_{qx}: the input of the voltage deviations

The DFIG and PMSG have the same control logic that observes currents and regulates voltages, so their controllers can share a general framework of the state-space model. The currents and voltages of the DFIG will be on the rotor side, while the PMSG will employ the stator-side variables. By transforming **Eqs. (121)** into the state-space form, **Eqs. (128)(129)** are the state and input matrices of the DFIG-based WT.

$$A = \begin{bmatrix} 0 & -D_{tm}^{-1}\dfrac{\partial T_d}{\partial \omega_d} & 0 & 0 & -D_{tm}^{-1}\dfrac{\partial T_d}{\partial \beta} & 0 \\ 0 & J_t^{-1}\left(\dfrac{\partial Q_d}{\partial \omega_d} - D_t\right) & 0 & \dfrac{n_g}{J_t}\dfrac{3p}{k_g}\dfrac{L_m}{L_s}\dfrac{V_b}{\omega_e} & J_t^{-1}\dfrac{\partial Q_d}{\partial \beta} & 0 \\ 0 & 0 & -\dfrac{R_r}{\sigma L_r} & \omega_r & 0 & 0 \\ 0 & 0 & -\omega_r & -\dfrac{R_r}{\sigma L_r} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\tau_\beta^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\tau_\gamma^{-1} \end{bmatrix} \tag{128}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ (\sigma L_r)^{-1} & 0 & 0 & 0 \\ 0 & (\sigma L_r)^{-1} & 0 & 0 \\ 0 & 0 & \tau_\beta^{-1} & 0 \\ 0 & 0 & 0 & \tau_\gamma^{-1} \end{bmatrix} \tag{129}$$

Similarly, **Eqs. (130)(131)** represent the state and input matrices of the PMSG-based WT.

$$A = \begin{bmatrix} 0 & -D_{tm}^{-1}\dfrac{\partial T_d}{\partial \omega_d} & 0 & 0 & -D_{tm}^{-1}\dfrac{\partial T_d}{\partial \beta} & 0 \\ 0 & J_t^{-1}\left(\dfrac{\partial Q_d}{\partial \omega_d} - D_t\right) & 0 & \dfrac{n_g}{J_t}\dfrac{3p}{k_g}\psi_f & J_t^{-1}\dfrac{\partial Q_d}{\partial \beta} & 0 \\ 0 & 0 & -\dfrac{R_s}{L_d} & \omega_s\dfrac{L_q}{L_d} & 0 & 0 \\ 0 & 0 & -\omega_s\dfrac{L_d}{L_q} & -\dfrac{R_s}{L_q} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\tau_\beta^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\tau_\gamma^{-1} \end{bmatrix} \tag{130}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ L_d^{-1} & 0 & 0 & 0 \\ 0 & L_q^{-1} & 0 & 0 \\ 0 & 0 & \tau_\beta^{-1} & 0 \\ 0 & 0 & 0 & \tau_\gamma^{-1} \end{bmatrix} \tag{131}$$

To handle small variations caused by inevitable errors, we expand the state vector to include some integral items for the actuators, which results in an update (**Eqs. (132)~(134)**) for the state-space model.

$$\Delta x = \begin{bmatrix} \cdots & \int \Delta I_{dx} & \int \Delta I_{qx} & \int \Delta\beta & \int \Delta\gamma \end{bmatrix}^T \tag{132}$$

$$A = \begin{bmatrix} 0 & a_{12} & 0 & 0 & a_{15} & 0 & 0 & 0 \\ 0 & a_{22} & 0 & a_{24} & a_{25} & 0 & 0 & 0 \\ 0 & 0 & a_{33} & a_{34} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{133}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b_{31} & 0 & 0 & 0 \\ 0 & b_{42} & 0 & 0 \\ 0 & 0 & b_{53} & 0 \\ 0 & 0 & 0 & b_{64} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{134}$$

The small-signal model only exists inside model-based controllers, which offer a potential trajectory estimation on the control horizon. Therefore, a controller can know the consequential results of control variables and consider these influences in multi-objective regulation. Besides, the state-space model directly separates state and control variables, contributing to concise and straightforward 3-DOF control designs.

2.7 Reliability Verification

This section gives a verification case to discuss simulation errors. The employed prototype and original data come from the FAST official release [129]. The simulator adopts the conventional control theories later introduced in **section 6.1** because the FAST only supports the MPS. **Figure 2.13** compares operating states as a function of wind speed. Most results can match the FAST data except for part results at about 6 m/s. The reason for these errors is that our control file follows the classic region division (**Figure 6.1**) and does not have special parameter settings for Region 1½. Thus, they arise from control differences but not system modelling. Besides, the thrust results have a certain degree of offset because the BEM solver ignores shaft tilt and blade pre-cone. Given that both are user-customized parameters for tower clearance and barely affect tower top dynamics, this study still considers strict orthogonal cases to calculate the thrust response.

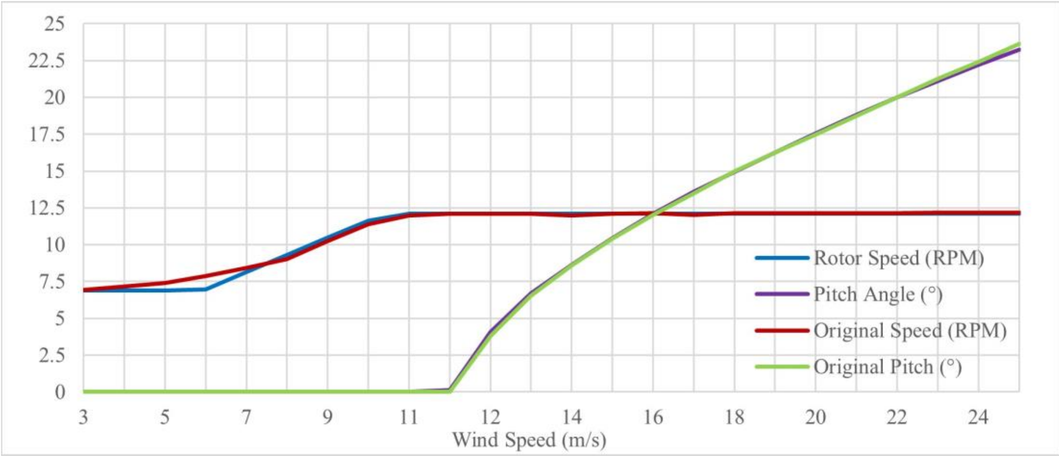

Figure 2.13a Comparison of 2-DOF control inputs

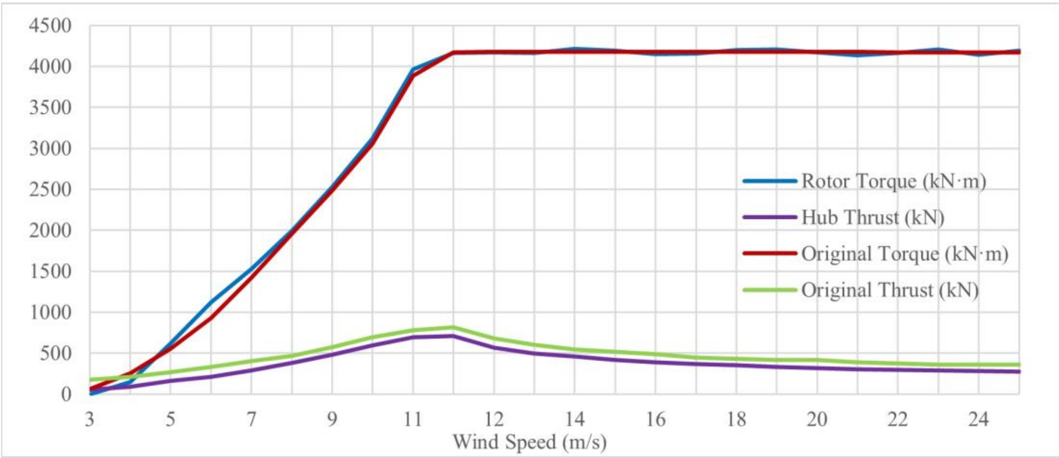

Figure 2.13b Comparison of aerodynamic forces

Figure 2.13 Comparison between simulated results and original FAST data

Table 2.1 lists two kinds of correlation coefficients to evaluate consistency with the FAST. The numerical results are close to or higher than 0.99, which verifies the accuracy of our simulator results. Thus, the proposed simulator is reliable and accurate for turbine dynamics.

Table 2.1 Correlation metrics of the operating curves

	Rotor Speed	**Pitch Angle**	**Rotor Torque**	**Hub Thrust**
R-square	0.989950	0.999833	0.998377	0.992246
Pearson's r	0.994962	0.999916	0.999188	0.996115

Chapter 3 Wind Turbine Definition

This section introduces three famous turbines (5-MW, 10-MW, and 15-MW) that already have wide applications in the wind industry and can provide persuasive simulation results. These turbines involve the geared, direct-drive, DFIG, and PMSG. Each target turbine comprises a prototype definition and an equivalent generator model. The definition includes airfoil descriptions, drivetrain details, pre-described regions, and tower modal characteristics. The generator specifications mainly consist of voltage, resistance, inductance, and operating frequency.

3.1 NREL 5-MW

The NREL issues a famous 5-MW definition for offshore and land-based WTs, named NREL 5-MW. The conceptual design calculations come from the Dutch Offshore Wind Energy Converter (DOWEC) project [129]. The NREL 5-MW is a three-blade variable speed turbine with pitch regulation and is initially compatible with a low-pole DFIG. **Table 3.1** provides the mechanical specifications of the NREL 5MW [129] with a virtual DFIG [19]. Since this study ignores shaft tilt and blade pre-cone, the rated wind speed slightly differs from the manual [129], which updates 11.4 m/s to 11.2 m/s. Besides, the gearbox ratio is adjusted to match the synchronous speed of the 6-pole DIFG.

Table 3.1 Parameters of the NREL 5-MW baseline

Parameter	Value	Comment
Wind regime	REpower 5M	-
Rotor rotation	clockwise, upwind	-
Control	variable speed, collective pitch	-
Cut-in wind speed v_i^{min}	3.0 m/s	-
Cut-off wind speed v_i^{max}	25.0 m/s	-
Rated wind speed v_i^{rtd}	11.2 m/s	-
Rated electrical power P_b	5 MW	-
Number of blades B	3	-
Rotor diameter	126 m	accounts for a tip length r_{tip} of 63 m
Airfoil series	DOWEC	17 individual elements
Minimum rotor speed ω_d^{min}	6.9 RPM	-
Maximum rotor speed ω_d^{max}	12.1 RPM	-

Design pitch β_{min}	0 deg	minimum pitch angle
Pitch time constant τ_β	1 s	
Pitch velocity β_v	8.0 deg/s	maximum velocity
Drivetrain	multi-stage gearbox	leads to a gear ratio n_g of 82.5
Low-speed shaft inertia J_d	38,759,228 kg·m^2	-
High-speed shaft inertia J_m	534.116 kg·m^2	-
Driveshaft spring constant K_s	867,637,000 N·m/rad	-
Driveshaft damping constant D_s	6,215,000 N·m/(rad/s)	-
Tower modal mass M_{tm}	34,7460 kg	-
Tower modal damping D_{tm}	107,423 N·s/m	-
Tower modal stiffness K_{tm}	369,024 N/m	-
Number of pole pairs p	3	matched for 6.9 RPM
Rated voltage V_b	398 V	-
Rotor turns n_r	10	boosts rotor voltage to 129 V
Stator resistance R_s	0.0026 Ω	equivalent circuit value
Rotor resistance R_r	0.0029 Ω	equivalent
Stator inductance L_s	0.002587 H	equivalent
Rotor inductance L_r	0.002587 H	equivalent
Magnetizing inductance L_m	0.0025 H	equivalent
Synchronous frequency f_e	50 Hz	-
Generator efficiency k_g	0.944	results in 5.23 MW P_d^{max}

3.2 IEA 10-MW

The IEA 10-MW offshore is part of the IEA Wind Task 37 outcome and upgrades from the Technical University of Denmark (DTU) 10-MW reference [4]. In this study, the rotation plane is strictly perpendicular to inflow wind. **Table 3.2** lists the technical details of the IEA-10MW with the DFIG equivalent circuit. This study applies a middle-voltage DFIG to match the IEA 10-MW, considering a smaller size and no worries about demagnetization for long-term operation. The number of poles is adjusted to 700 to match the rated speed. Compared with conventional geared DFIGs, the direct-drive DFIG does not require a gearbox and thus reduces rotation oscillation due to stiff gearing.

Table 3.2 Parameters of the IEA 10-MW offshore

Parameter	Value	Comment
Wind regime	IEC class 1A	-
Rotor rotation	clockwise, upwind	-
Control	variable speed, collective pitch	-

Cut-in wind speed v_i^{min}	4.2 m/s	-
Cut-off wind speed v_i^{max}	25.0 m/s	-
Rated wind speed v_i^{rtd}	10.5 m/s	-
Rated electrical power P_b	10 MW	-
Number of blades B	3	-
Rotor diameter	198 m	accounts for a tip length r_{tip} of 99 m
Airfoil series	FAA-W3	30 individual elements
Minimum rotor speed ω_d^{min}	6.0 RPM	-
Maximum rotor speed ω_d^{max}	8.68 RPM	-
Design pitch β_{min}	0 deg	minimum pitch angle
Pitch time constant τ_β	1 s	
Pitch velocity β_v	7.0 deg/s	maximum velocity
Drivetrain	direct-drive	leads to a unit gear ratio n_g
Blade mass	47,700 kg	leads to a total moment of inertia J_t of 155,835,900 kg·m^2
Tower modal mass M_{tm}	402,202 kg	-
Tower modal damping D_{tm}	181,859 N·s/m	-
Tower modal stiffness K_{tm}	913,662 N/m	-
Number of pole pairs p	350	matched for 8.68 RPM
Rated voltage V_b	3.3 kV	-
Rotor turns n_r	25	boosts rotor voltage to 0.33 kV
Stator resistance R_s	0.016 Ω	equivalent circuit value
Rotor resistance R_r	0.019 Ω	equivalent
Stator inductance L_s	0.01587 H	equivalent
Rotor inductance L_r	0.01587 H	equivalent
Magnetizing inductance L_m	0.015 H	equivalent
Synchronous frequency f_e	50 Hz	-
Generator efficiency k_g	0.944	results in 10.59 MW P_d^{max}

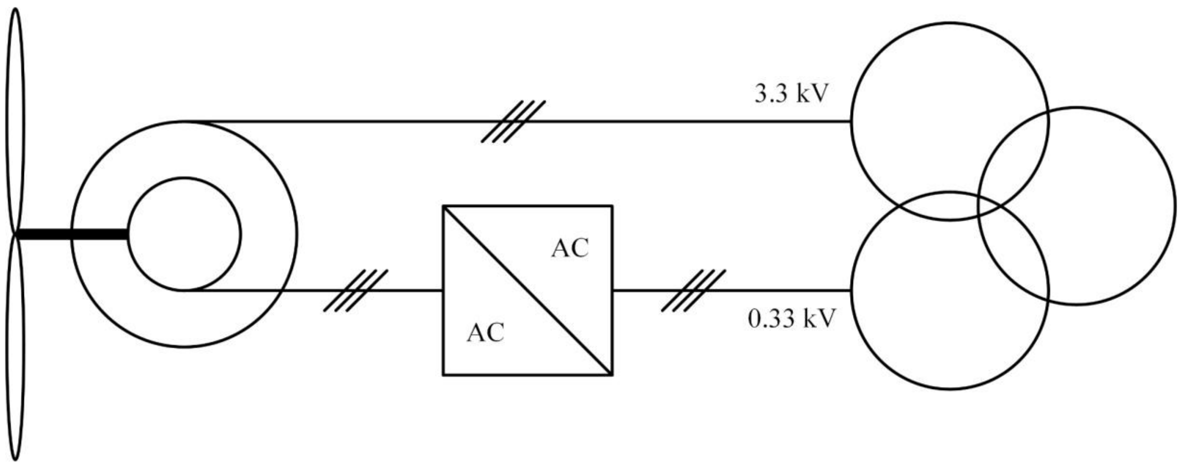

Figure 3.1 Stator and rotor configuration of the direct-drive middle-voltage DFIG

Figure 3.1 shows the DFIG configuration, in which the stator voltage needs to be as high as possible, and the rotor voltage tends to be lower, benefiting a low-voltage converter. Hence, there are individual voltage couplings of the stator and rotor connecting to the grid [21].

3.3 IEA 15-MW

The NREL and the DTU report a 15-MW turbine reference through the second work package of the IEA Wind Task 37, named IEA 15-MW [130]. The IEA 15-MW leaps ahead of the present generation of industry WTs for fixed-bottom offshore wind energy. The IEA 15-MW applies a monopile support structure, a Class IB direct-drive machine, a rotor diameter of 240 m, and a hub height of 150 m, as displaced in **Figure 3.2** [130].

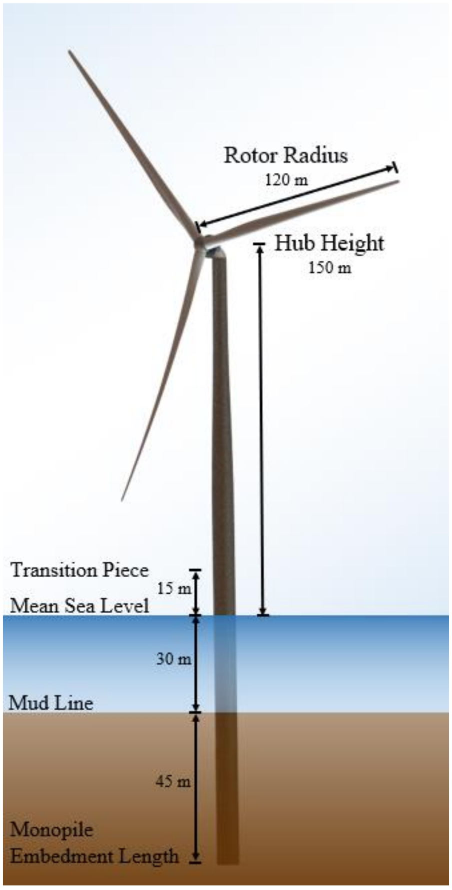

Figure 3.2 Structural layout of the IEA 15-MW offshore (source: Ref. [130])

Meanwhile, the IEA 15-MW adopts a highly integrated structure, where the main shaft joins the PMSG rotor, and the PMSG stator attaches to the yaw bedplate, as displayed in **Figure 3.3** [130]. Its layout discards the conventional nacelle that places individual devices, which reduces manufacturing costs and installation difficulties.

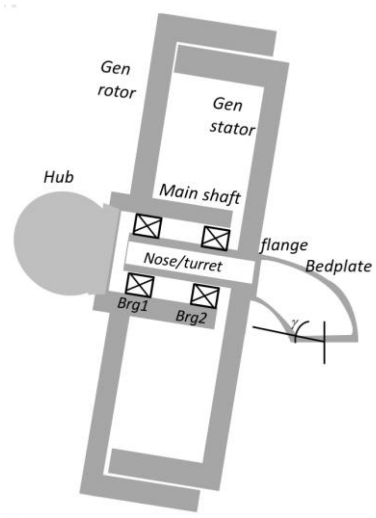

Figure 3.3a PMSG sketch

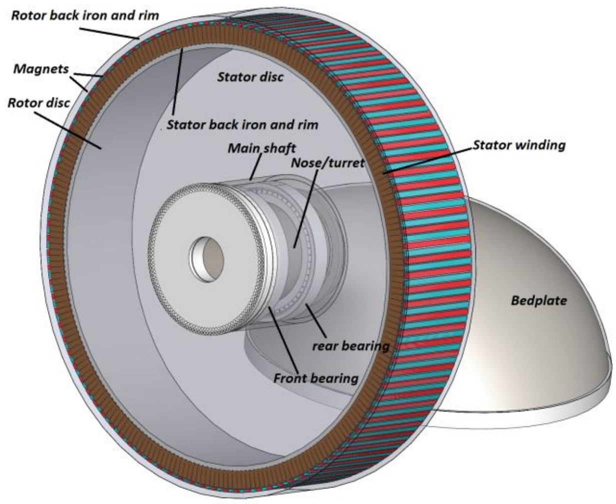

Figure 3.3b Nacelle CAD

Figure 3.3 Layout of the integrated PMSG and nacelle (source: Ref. [130])

Table 3.3 provides the necessary parameters of the IEA 15-MW. Each simulation step iterates on 50 blade elements to compute aerodynamic responses. The IEA 15-MW employs a 200-pole PMSG to achieve energy conversion. The power rating of the PMSG (15 MW) corresponds to a maximum of 15.54 MW absorption. Besides, the first-order model with rate constraint mathematically simulates the pitch or yaw servo.

Table 3.3 Parameters of the IEA 15-MW offshore

Parameter	Value	Comment
Wind turbine class	IEC class 1B	-
Rotor orientation	upwind	-
Control	variable speed, collective pitch nacelle rotation	-
Cut-in wind speed	4.4 m/s	optimized to avoid too low outputs
Cut-off wind speed	25.0 m/s	-
Rated wind speed	10.2 m/s	optimized for perpendicular rotation plane
Power rating P_b	15 MW	-
Number of blades B	3	-
Rotor diameter	240 m	accounts for a tip length r_{tip} of 120 m
Airfoil series	FAA-W3	50 blade elements
Minimum rotor speed ω_d^{min}	5.0 RPM	-
Maximum rotor speed ω_d^{max}	7.56 RPM	-
Design pitch β_{min}	0 deg	minimum pitch angle
Pitch time constant τ_β	1 s	first-order model

Pitch velocity β_v	7.0 deg/s	maximum velocity
Yaw time constant τ_γ	1.5 s	first-order model
Nacelle rotation limit γ_v	0.25 deg/s	maximum rotation rate
Blade mass	65,000 kg	leads to a total moment of inertia J_t of 936,000,000 kg·m^2
Drivetrain	direct-drive	-
Tower modal mass M_{tm}	550,400 kg	-
Tower modal damping D_{tm}	176,282 N·s/m	-
Tower modal stiffness K_{tm}	627,329 N/m	-
Number of pole pairs p	100	matched for 7.56 RPM
Rated voltage V_b	4.77 kV	-
Stator resistance R_s	0.16 Ω	-
Stator d-axis inductance L_d	0.01587 H	$L_d = L_q$ for symmetrical machines
Stator q-axis inductance L_q	0.01587 H	-
Permanent magnet flux ψ_f	19.49 wb	-
Synchronous frequency f_e	12.6 Hz	-
Generator efficiency k_g	96.5%	accounts for 15.54 MW P_d^{max}

Chapter 4 Control Framework

Present WT controls simplify the rules of turbine operation, so their frameworks are only suitable for a fixed generation mode, which means that any MPS-based frameworks cannot directly apply to the control implementation of the OPS. Thus, this study proposes two OPS-based control frameworks for terminal controller design. The developed frameworks combine multiple ML techniques and classic controllers, which adopt a general control logic (**Figure 4.1**).

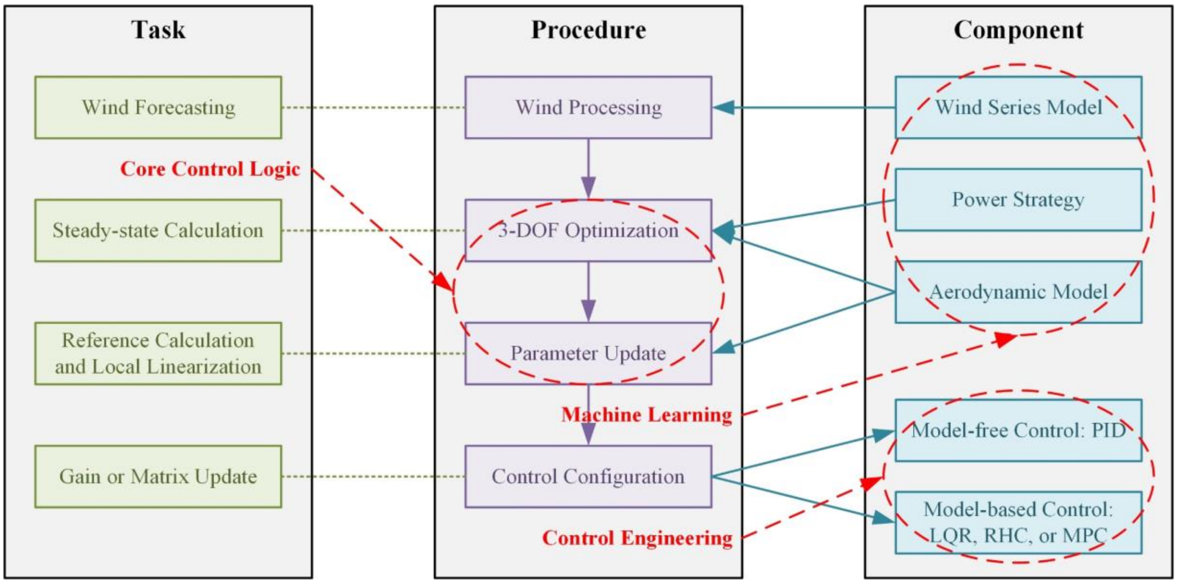

Figure 4.1 General control logic of the proposed ML-based turbine control

First, a wind processing unit yields wind speed and direction references for further calculations of the power strategy. This study employs a DL-based wind series model to achieve forecasting. However, wind forecasting is optional, and grid users can use other meteorological processing methods to deliver wind references. Once the power strategy receives the latest wind information, it will find the optimal 3-DOF steady state of rotation, pitch, and yaw to satisfy the requirement of power dispatching. This study develops two methods to implement the core functionality of the power strategy: a PRPT-based algorithm and an RL agent. It notes that both methods require an aerodynamic model to iterate. Subsequently, the control system utilizes the aerodynamic model to calculate other necessary references (i.e., power, thrust, voltages, currents, and tower displacement) and perform a local linearization to estimate force sensitivities. Once the control system collects all required configuration parameters, it updates gains for model-free control or system matrix for model-based control.

The terminologies about 'model-free' or 'model-based' refer to whether a controller for a nonlinear process needs a small-signal model as its internal control model to estimate system trajectory and compute control sequence. This study involves four specific controllers (PID, LQR, RHC, and MPC) in which the PID belongs to model-free, and the other three are model-based.

4.1 Model-free Control

The hierarchy of the model-free implementation is a five-layer architecture, as displayed in **Figure 4.2**. This model-free framework originates from the FAST control design [95] and adds some modules to be compatible with the OPS. The supplemented parts mainly include a generator loop, a tower damping loop, and actuator gain calculations.

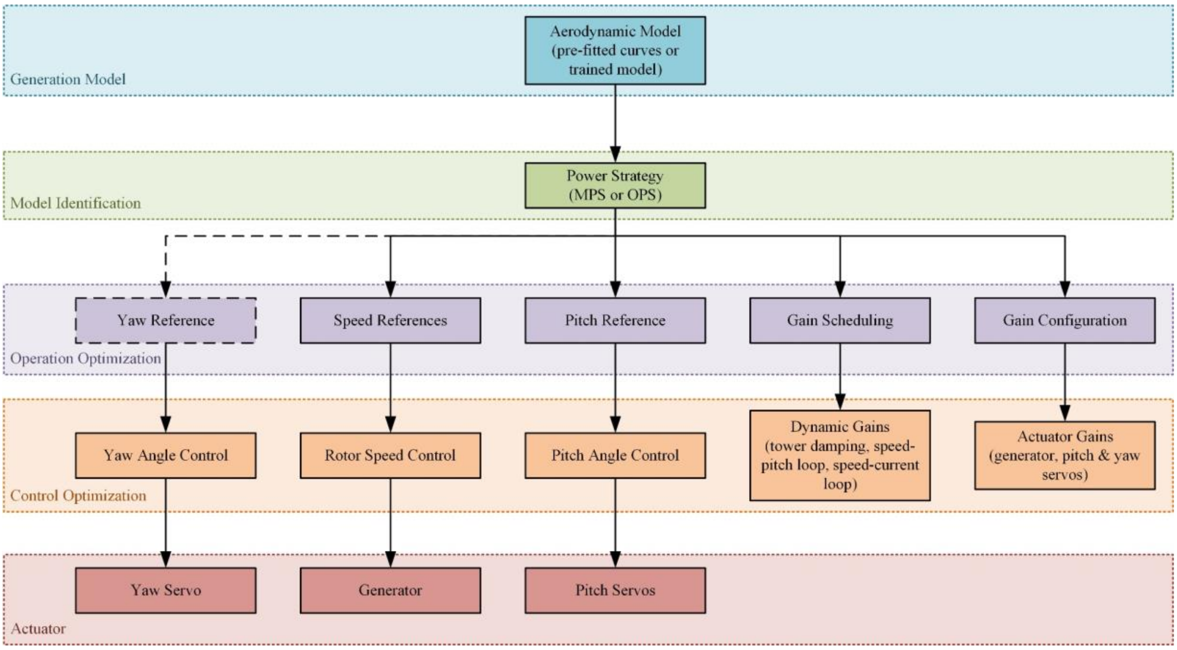

Figure 4.2 Hierarchy of the model-free framework

First, the control system contains an aerodynamic model that can be a set of pre-fitted curves or a trained ANN-based model. As the key to the control system, the power strategy is responsible for deriving essential parameters from the aerodynamic model. The first role of the power strategy is to solve the speed reference of the RSC and the pitch reference of the PAC through a built-in algorithm considering wind scenario or power demand, which accepts both the conventional MPS and the innovative OPS. The second role calculates aerodynamic parameters, i.e., torque and thrust sensitivities, in gain scheduling. Once the RSC and PAC receive input references, the generator and pitch servos take action to approach the reference state inferred by the power strategy. Besides, the power strategy unwraps the yaw reference from wind forecasting and orients the nacelle to turn the rotor disk perpendicular to wind flow. At the stage of control initialization, the power strategy only once calculates optimal gains for the generator and servos to eliminate actuator delay.

4.2 Model-based Control

Since **section 2.6** already establishes two turbine small-signal models, this study develops a corresponding model-based control framework to achieve trajectory prediction and multi-objective optimization. This framework consists of four objects: an aerodynamic model, an OPS algorithm, a periodic updated control model, and a state space-based controller. It is worth mentioning that the model-based framework is only available to the OPS since the MPS does not compute the necessary parameters for control configuration.

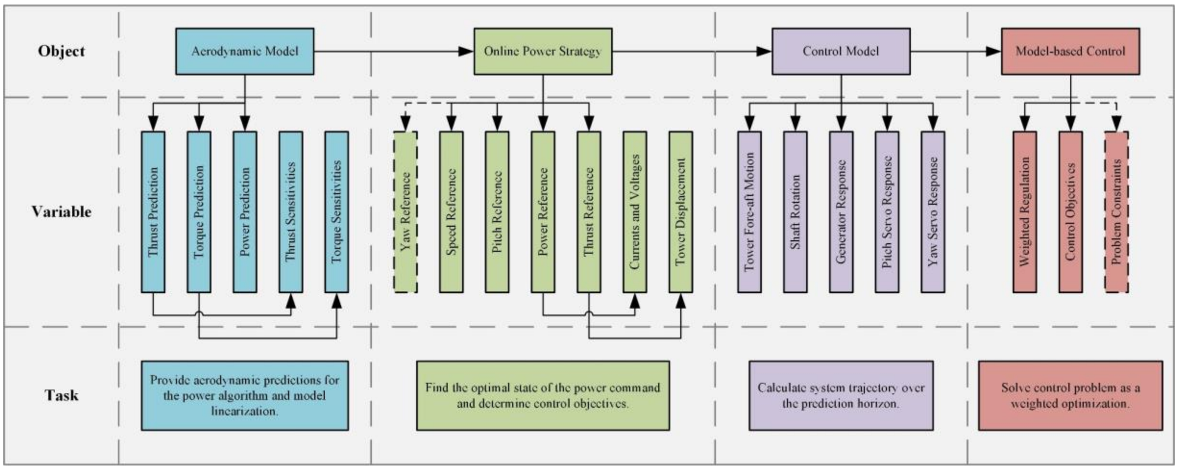

Figure 4.3 Flowchart of the model-based framework

Figure 4.3 illustrates the information flow between objects and critical variables to be updated. The OPS utilizes the aerodynamic model to decide regulation objectives and control parameters. Subsequently, the control model accepts the updated configuration for accurate trajectory estimation. The state space-based controller converts the control objective and control model to a weighted optimization that observes system states and computes control inputs.

4.3 Control Schematic

The small signal analysis in **section 2.6** indicates that signal acquisition and control output relate to the generator type. Therefore, this section presents the 3-DOF control schematic on the DFIG rotor and PMSG stator, respectively, as shown in **Figure 4.4**. The DFIG or PMSG can have an external controller to realize the grid-side control [109], so the generator control only considers the rotor-side or stator-side control.

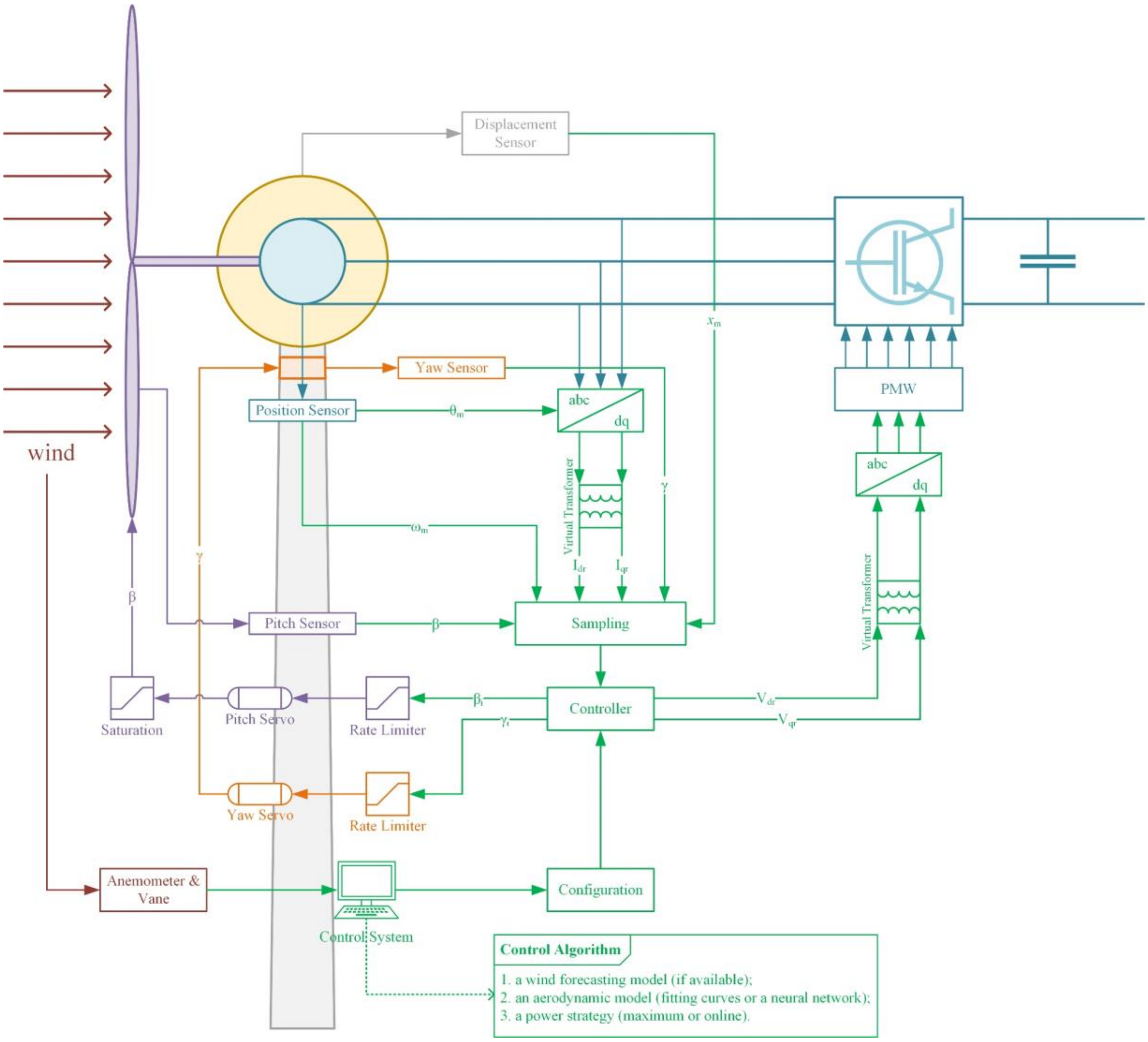

Figure 4.4a Deployment on the DFIG rotor side

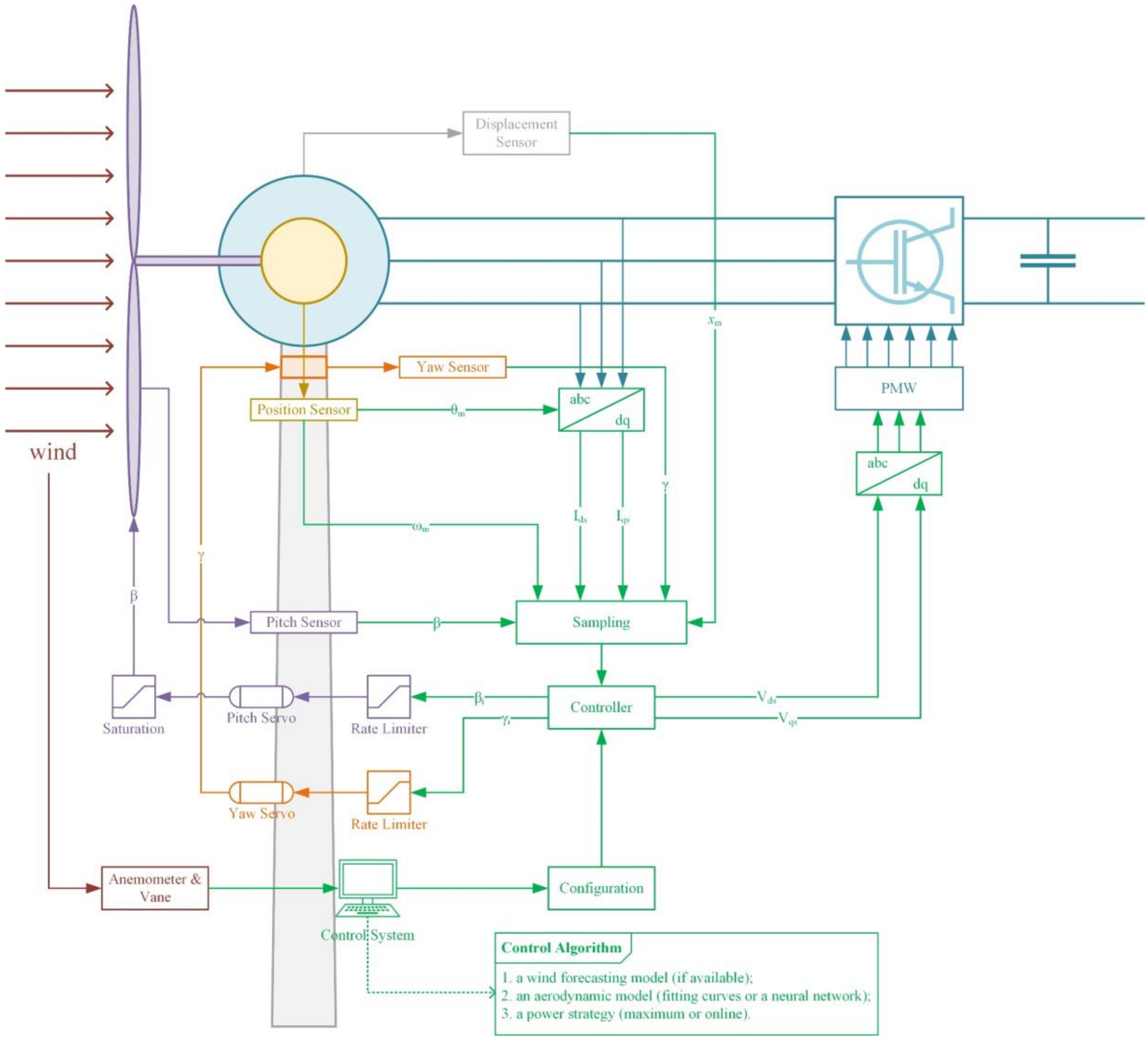

Figure 4.4b Deployment on the PMSG stator side

Figure 4.4 Control schematic of the 3-DOF turbine system

Firstly, the power strategy determines a reference point and necessary control parameters. Secondly, the control system updates the optimal gains or the small-signal control model. Meanwhile, the control system receives the control objective issued by the power strategy. After this, the controller monitors the system state and takes action to track the control objective. The system state includes tower displacement, rotor speed, generator currents, pitch position, and nacelle yaw. The control variables are generator voltages and servo inputs. Besides, if the control system has a wind forecasting model for wind series, the power strategy will use wind prediction to calculate the reference point to enhance accuracy.

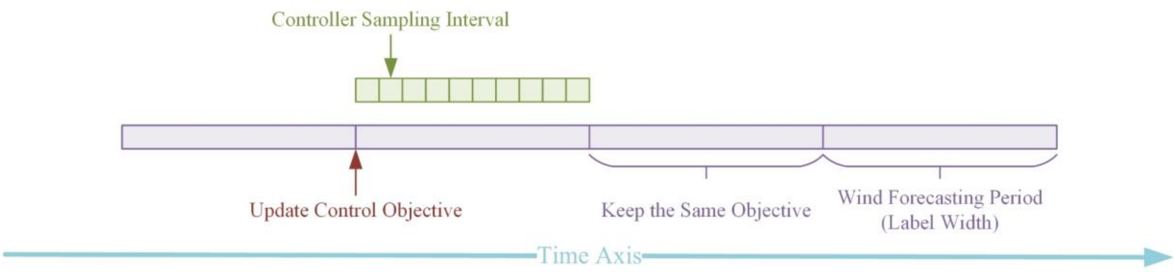

Figure 4.5 Time definition of the wind forecasting-enhanced controller

Our turbine control systems involve two time periods, as shown in **Figure 4.5**: a longer period for the power strategy (wind forecasting period) and a faster period for the controller (sampling interval). In each forecasting period, wind information passed into the power strategy remains the same, i.e., the 3-DOF calculation result also stays the same. Therefore, the control system only updates the power strategy at the beginning of each forecasting period. Given that wind information essentially comes from a wind forecasting model, this study names the updating period of a power strategy as the wind forecasting period (or prediction label width). Since a digital controller can only process discrete computations, it has to sample a system state and return a control result. A continuous controller needs a fast enough frequency to eliminate discrete effects, while a discrete controller allows a slower frequency if its discretization is valid. It notes that there is no physical relationship between forecasting and sampling. The forecasting period depends on the design of a time-series model, while the sampling interval relates to the characteristics of the system response.

Table 4.1 Recommended time settings

Name	Value
Wind Forecasting Period	20 s ~ 1 min
Sampling Interval (PID, LQR)	≤ 0.2 ms
Sampling Interval (RHC, MPC)	1 ms

Table 4.1 lists our recommended time settings. System inertia and wind characteristics affect the determination of a forecasting period. Generally speaking, the control system must reserve sufficient time for kinetic transition and state hold. Meanwhile, wind variation is a macroscopic result of airflow, so it is unlikely to have sudden changes in a short time. However, a too-long period may lead to a failure to track wind variation. Therefore, this

study recommends setting the forecasting period of the output of a wind model in a range of 20 s ~ 1 min. The rule of thumb is the most efficient way to determine the sampling interval of a turbine control because obtaining a specific transfer function for frequency analysis is extremely difficult due to nonlinear and complex aerodynamics. According to our experiments, a sampling period of less than 0.2 ms for continuous controllers (PID and LQR) can exceed the threshold of discrete effects. Considering the trade-off between fast response and computation burden, this study suggests an interval of 1 ms for discrete controllers (RHC and MPC).

Chapter 5 Aerodynamic Modelling

Since wind generation is not a simple process that several linear equations can express, turbine control requires a sophisticated aerodynamic model to determine steady states and perform local linearization. Aerodynamic variables include hub thrust, rotor torque, and captured power. This section introduces an ANN-based model forecasting the above aerodynamics for control configuration. **Figure 5.1** displays the relation between input variables and aerodynamic outcomes. Since the aerodynamic model is an indispensable control component, model training is a prerequisite for control deployment, which implies that initial training and testing can only use synthetic data. The aerodynamic solver developed in **section 2.1** can generate synthetic data. After turbine installation, users can collect and add measurements to the dataset for further model enhancement. However, this study only considers the scenario of generating synthetic data for the initial model.

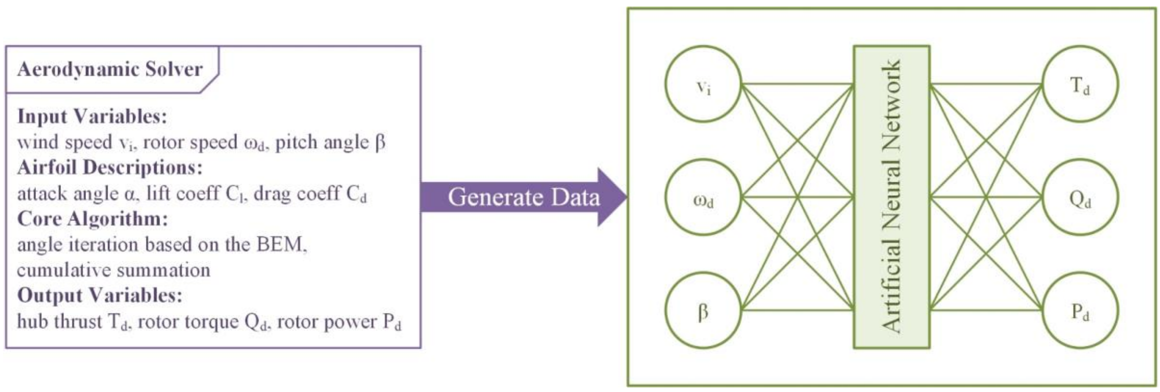

Figure 5.1 Input and output relation of the ANN-based aerodynamic model

Data generation takes a mesh-grid approach, i.e., the input of the entire data is a tuple of coordinate vectors in which each input variable has an even distribution. The resolution of an input variable is a critical factor of data precision that affects model accuracy and training efficiency. A smaller resolution contributes to a better description of turbine nonlinear kinetics but results in excessive data that consumes massive computational resources. **Table 5.1** lists the suggested resolutions based on our experience. Operating regions of modern turbines have many overlaps due to physical limitations, so the configuration of **Table 5.1** can suit most turbines.

Table 5.1 Recommended resolutions for data generation

	Wind Speed (m/s)	Rotor Speed (RPM)	Pitch Angle (deg)
Resolution	0.2	0.02	0.1

Figure 5.2 is the flowchart of data processing and model training. First, the aerodynamic solver generates necessary data, which consists of input and output. Subsequently, normalization and outlier detection preprocess data. The cleaned data are randomly split into training (with validation) and testing sets. TensorFlow [71] is responsible for model training and reloading.

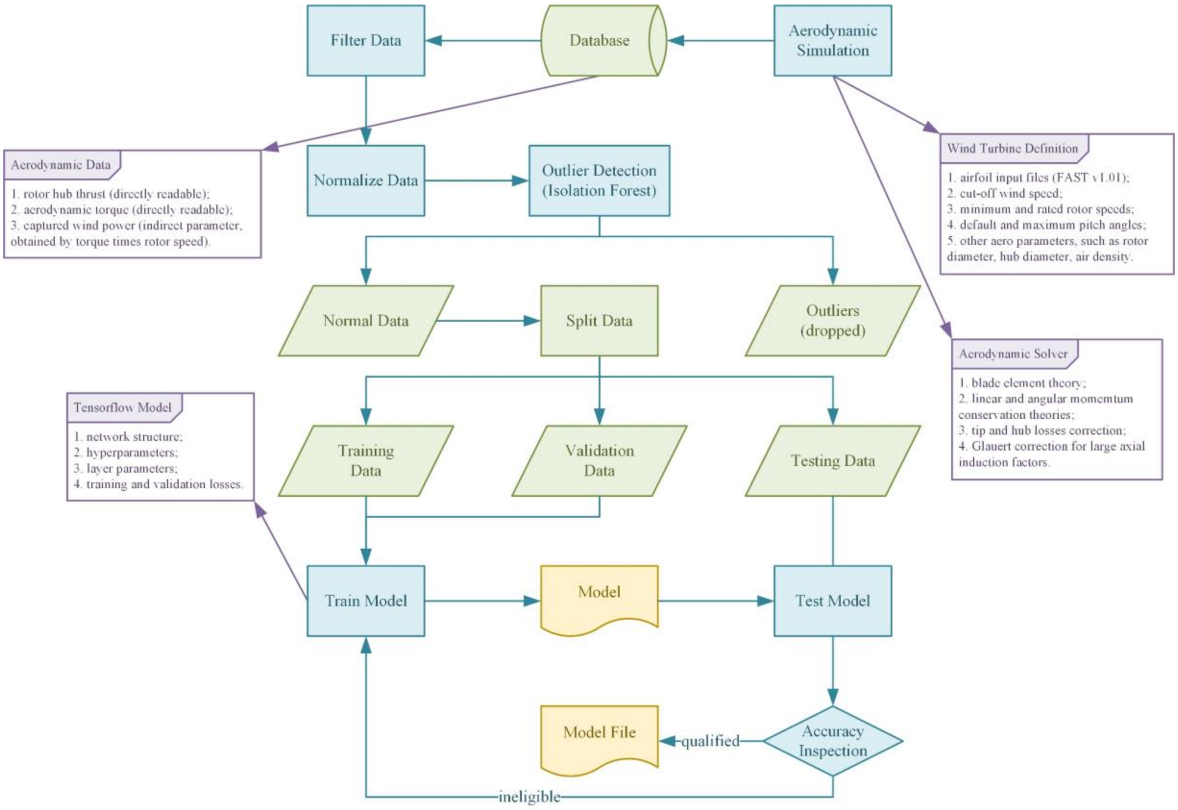

Figure 5.2 Flowchart of training and testing a TensorFlow-based ANN

5.1 Data Preprocessing

Figure 5.3 visualizes the IEA 10-MW aerodynamics (thrust, torque, and power) over the entire region and the corresponding available output with physical constraints. The raw data contain many points that the IEA 10-MW never uses. Some power exceeds 40 MW and even has negative production, but both are out of the design region and thus irrational to regular operation. It is necessary to filter meaningless data to accelerate training. A range of -10%~110% ensures rational operation and enough boundary data. It notes that the coarse filtering of a limited region cannot apply to the RL-based OPS since the RL training explores the entire region. For magnitude consistency among different dimensions, a min-max scaler normalizes (**Eq. (135)**) original data before further processing and later denormalizes (**Eq. (136)**) model predictions [131].

$$\tilde{x}' = \frac{\tilde{x} - \tilde{x}_{min}}{\tilde{x}_{max} - \tilde{x}_{min}} \tag{135}$$

$$\tilde{x} = \tilde{x}'(\tilde{x}_{max} - \tilde{x}_{min}) + \tilde{x}_{min} \tag{136}$$

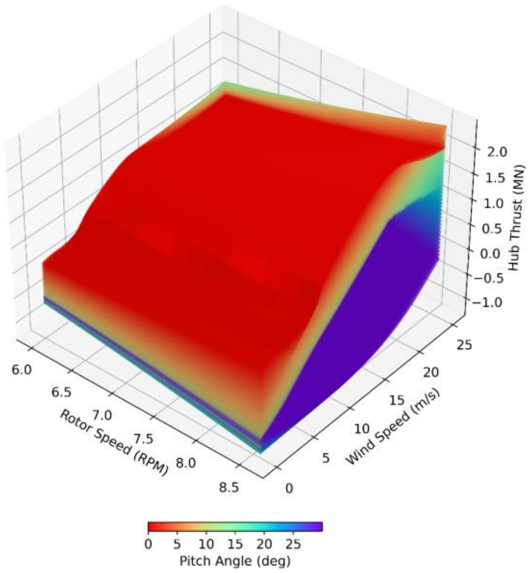

Figure 5.3a Hub thrust

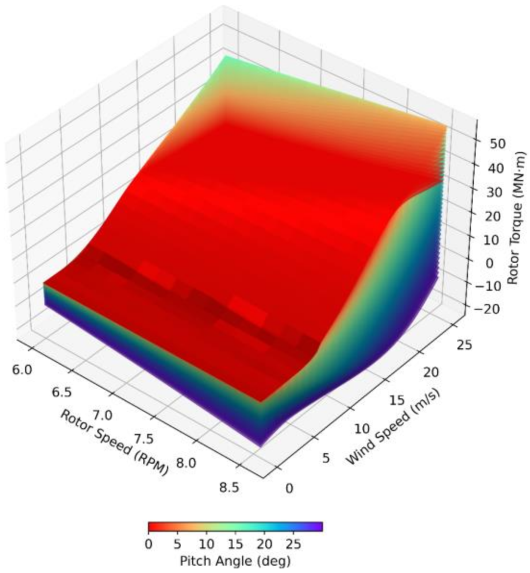

Figure 5.3b Rotor torque

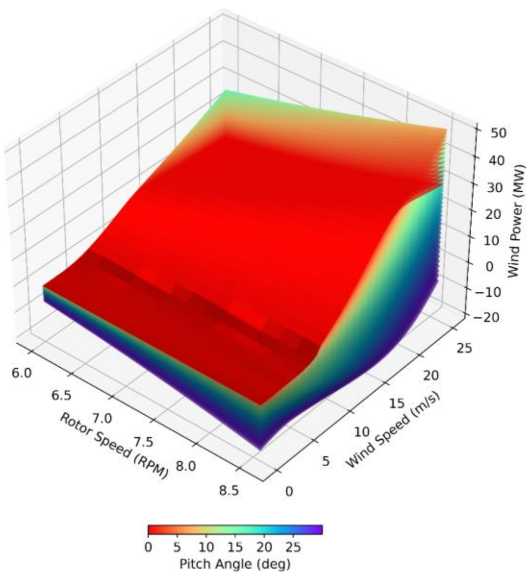

Figure 5.3c Rotor power

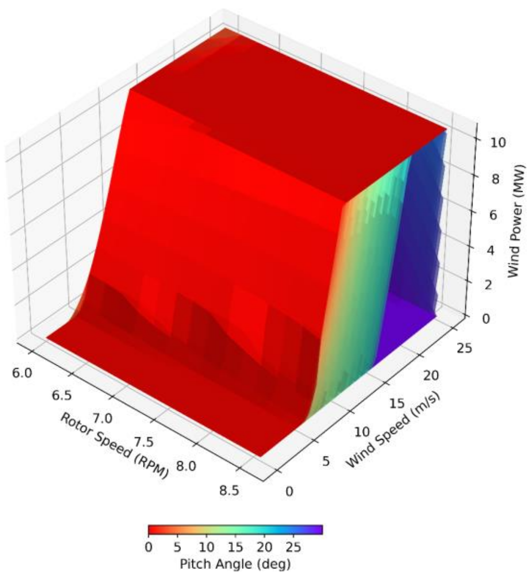

Figure 5.3d Available output

Figure 5.3 Aerodynamics of the IEA 10-MW in the entire operation region

Since the truncation of fixed boundaries leads to uneven data distribution in some training batches, an outlier detection (iForest [76]) detects these points to make the distribution more consistent. The iForest is a model-based approach that uses two outlier properties: they are a minority consisting of fewer samples and have attribute values that differ from those of normal instances [132]. The detecting process of the iForest includes generating isolation trees after subsampling, calculating the path length, and evaluating the anomaly score (**Eq. (137)**) to identify if an instance belongs to anomalies (it is considered as an anomaly when its score is close to 1) [132].

$$s(\boldsymbol{i},n) = 2^{-\frac{E(h(\boldsymbol{i}))}{c(n)}} \tag{137}$$

where

$s(\boldsymbol{i},n)$: the anomaly score of an instance

$\boldsymbol{i}$: a measurement instance of input and output

$h(\boldsymbol{i})$: the path length according to isolation trees

$E(h(\boldsymbol{i}))$: the average path length of $h(\boldsymbol{i})$ from a collection of isolation trees

$c(n)$: the average path length of an unsuccessful search in a binary search tree

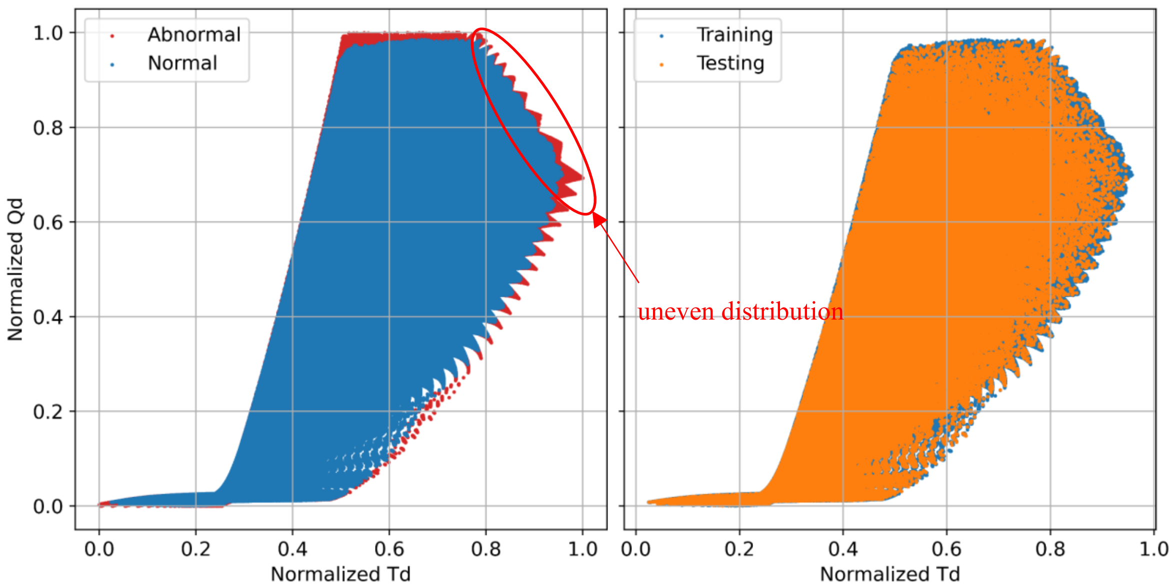

Figure 5.4 Outlier detection result and data splitting

The Python scikit-learn package provides an application programming interface (API) to call the iForest, which builds trees based on an ensemble of ExtraTreeRegressors [133]. The API of the iForest has three critical parameters affecting detection performance: the number of estimators, sample size, and contamination (drop) ratio. The first two mainly affect random error and solving speed, and a configuration of 200 estimators and 0.3 sample ratio can achieve the trade-off between accuracy and efficiency. The contamination ratio should be lower (1% in this study) to avoid missing too much information. **Figure 5.4** displays the thrust and torque projection of discarded data identified by the iForest. The highlighted data clusters probably cause unstable gradients if they concentrate on some batches. Detecting these outliers does not influence data integrity but contributes to stable training because of network generalization.

Table 5.2 provides the results of data preprocessing. The original data has 5,120,010 points (126×135×301 according to **Table 3.2** and **Table 5.1**), but only about 28% (71% is out of reasonable power range, and 1% belongs to anomalies) are normal data for training and testing ANNs.

Table 5.2 Data description for the IEA 10-MW aerodynamic modelling

Description	Size	Comment
Entire data	5,120,010	yielded by the BEM solver
Filtered data	3,689,192	about 71% of the entire, removed by coarse filtering
Outliers	14,609	about 1% of the entire, detected by the iForest
Training data	1,236,509	about 85% of the normal
Validation data	65,079	about 5% of the normal
Testing data	144,621	about 10% of the normal

5.2 Artificial Neural Network

Current WT models are multiple-input-single-output (MISO), i.e., modelling wind power through several inputs [134], but our target model is multiple-input-multiple-output (MIMO). Hence, this section proposes three networks, i.e., RBFN, DNN, and HDNN, for accurate MIMO forecasting. Also, since the OPS reliability depends on the aerodynamic model, it is necessary to investigate the impact of model differences on the OPS framework.

5.2.1 Radial Basis Function Network

The RBFN is a particular class of neural networks that evaluate distances from an input to RBF kernels [135]. Each RBF kernel learns to put its kernel at an appropriate position and returns the distance as the RBF output [70]. Compared with the commonly used perceptron, the RBF is naturally nonlinear and more suitable for nonlinear aerodynamic data. This study takes a multivariable Gaussian function (**Eq. (138)**) [136] as the RBF kernel.

$$\varphi_l(\tilde{x}) = \exp\left(-\frac{1}{2}\left\|\frac{\tilde{x}-\tilde{\mu}_l}{\tilde{\sigma}_l}\right\|^2\right) \tag{138}$$

where

φ_l: the RBF function of kernel l

$\tilde{\mu}_l$: the kernel centre

$\tilde{\sigma}_l$: the kernel spread

Table 5.3 lists the recommended RBFN configuration for aerodyanmic modelling. The hyperparameter tuning of an RBFN only increases the number of RBF kernels until training convergence has no improvement. An advantage of the RBFN is stable forecasting performance for sufficient kernels, i.e., the RBFN is unlikely to occur overfitting. However, the RBFN is very picky about the training optimizer, and some optimizers suffer from gradient explosion and disappearance.

Table 5.3 Configuration of the RBFN layers

Layer	Arguments	Value	Parameters
RBFunc	kernels	512	3,072
Dense	units	3	1,539

5.2.2 Deep Neural Network

DL has various networks, and one of the most famous types stacks multiple dense layers. Each layer consists of weights and biases that store information learned from training data and an activation function that nonlinearly connects two adjacent layers [68]. This study employs a classic DNN that stacks five dense layers, in which the first four layers are in charge of learning, and the last layer shapes the output of three variables. It is worth mentioning that the rectified linear unit (ReLU) [137], a widely-used activation, has compatibility issues with aerodynamic modelling due to overfitting. The DNN adopts the sigmoid function [55] to avoid the above problem. **Table 5.4** gives the DNN configuration.

Table 5.4 Configuration of the DNN layers

Layer	Arguments	Value	Parameters
Dense	units	64	256
	activation	sigmoid	
Dense	units	64	4,160
	activation	sigmoid	
Dense	units	64	4,160
	activation	sigmoid	
Dense	units	64	4,160
	activation	sigmoid	
Dense	units	3	195

5.2.3 Hybrid Deep Neural Network

The RBF kernel is efficient in feature extraction, and multiple dense layers can further enhance learning depth, which motivates us to combine the RBF and the multi-dense structure. **Figure 5.5** displays the architecture of the RBF-based hybrid DNN (HDNN).

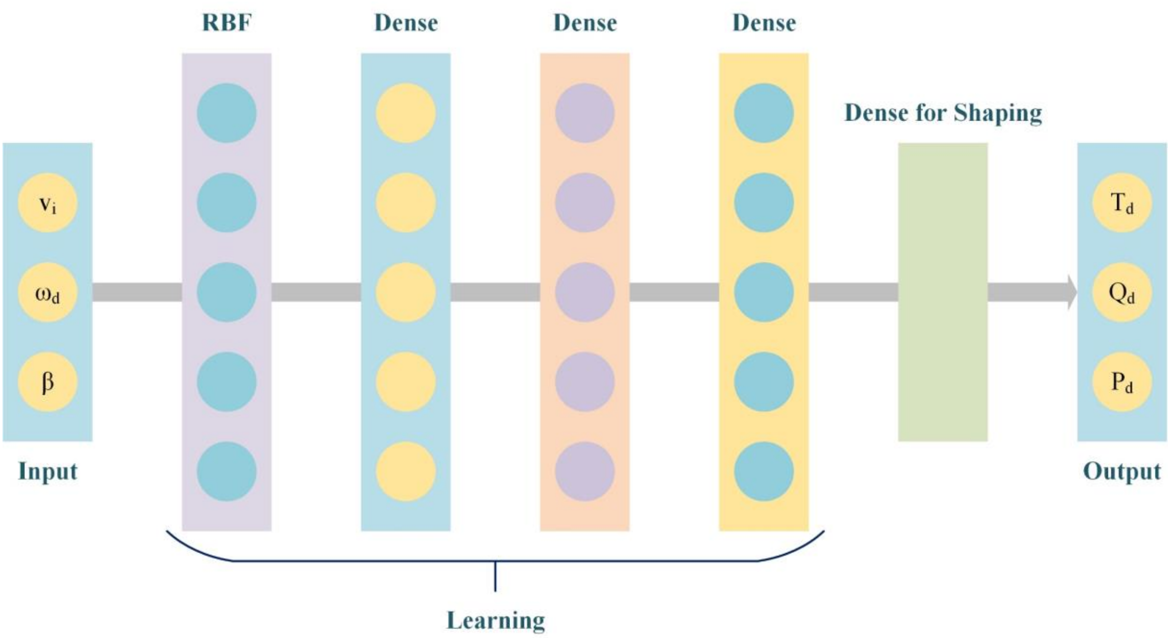

Figure 5.5 Network structure of the RBF-based HDNN

The HDNN cannot contain more than one RBF layer because two or more easily cause gradient disappearance. Meanwhile, similar to the DNN, the ReLU is incompatible with the RBF layer. The HDNN still applies the sigmoid activation. **Table 5.5** provides the configuration of the RBF-based HDNN.

Table 5.5 Configuration of the HDNN layers

Layer	Arguments	Value	Parameters
RBFunc	kernels	256	1,536
Dense	units	32	8,224
	activation	sigmoid	
Dense	units	32	1,056
	activation	sigmoid	
Dense	units	32	1,056
	activation	sigmoid	
Dense	units	3	99

5.3 TensorFlow Description

As a general ML platform, the quintessence of TensorFlow is its library of built-in layers and optimizers. The usage of the layer library depends on network structure, and the above content has given specific configurations. Therefore, the training optimizer is worthy of more attention because it predominately decides the accuracy of a final model. TensorFlow optimizers rely on the gradient descent to learn from data. First-order gradient algorithms [138] calculate gradients (or first derivatives) at the current point and move with a scaled step opposite the gradient, repeating until maximum iteration or minimum tolerance. The trade-off is to adjust the learning rate because a large value may skip over convex optima, and a small value may lead to slow convergence. Second-order gradient algorithms [139] add a momentum term with the second-order information to speed up learning and enhance search.

Table 5.6 lists eight state-of-the-art gradient training methods with their updating rules. The stochastic gradient descent (SGD) is a variant of the gradient descent that reduces redundant computation by updating one at a time but leads to heavy fluctuation of the objective function due to frequent updates [140]. The adaptive gradient algorithm (Adagrad) allows an adjustable learning rate, which performs large updates for parameters related to infrequent features and minor updates for frequent ones [141]. The adaptive learning rate method (Adadelta) is a more robust extension of the Adagrad that applies a moving window of gradient updates rather than the cumulative summation of all past gradients, which addresses the continual decay of the learning rate [142]. The root mean squared propagation (RMSprop) is an unpublished method of the adaptive learning rate that solves the decay of the learning rate in the Adagrad, which aims to maintain a moving average of squared gradients and divide the gradient by the root of its average [143]. The adaptive moment estimation (Adam) is another solution for updating the adaptive learning rate based on estimating the first and second order moments [144]. The Adamax is a variant of the Adam that uses the infinity norm as its adaptive term [144]. The Nesterov-accelerated adaptive moment estimation (Nadam) is an extension of the Adam that combines the Nesterov momentum [145]. The 'follow the regularized leader' (Ftrl) developed by Google initially works for advertisement click prediction, which has excellent sparsity and convergence properties [146]. According to our trials, the Adam optimizer has the most stable performance for training the proposed ANNs, and other optimizers have some issues, such as gradient disappearance, gradient exploration, and slow convergence.

Table 5.6 Gradient descent methods for training ANNs

Source	Name	Updating Rule	
Sutskever et al. [147]	SGD	$\theta_{t+1} = \theta_t - \eta g_t$	(139)
Duchi et al. [141]	Adagrad	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$	(140)
Zeiler [142]	Adadelta	$\theta_{t+1} = \theta_t - \frac{\sqrt{E[\Delta\theta^2]_{t-1}}}{\sqrt{E[g^2]_t + \epsilon}} g_t$	(141)
Hinton et al. [143]	RMSprop	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$	(142)
Kingma et al. [144]	Adam	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \widehat{m}_t$	(143)
Kingma et al. [144]	Adamax	$\theta_{t+1} = \theta_t - \frac{\eta}{\max(\beta_2 v_{t-1}, \lvert g_t \rvert)} \widehat{m}_t$	(144)
Dozat et al. [145]	Nadam	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \left(\beta_1 \widehat{m}_t + \frac{1-\beta_1}{1-\beta_1^t} g_t \right)$	(145)
McMahan et al. [146]	Ftrl	$\theta_{t+1} = \begin{cases} \theta_{t+1}, & \lvert z \rvert < \lambda_1 \\ \theta_{t+1} - \frac{\eta(z - sgn(z)\lambda_1)}{\alpha + \sqrt{E[g^2]_t} + \eta\lambda_2}, & \text{else} \end{cases}$	(146)

where (only for the above table)

θ_{t+1}: the unsolved parameter at the t-th training

η: the learning rate

$g_t = \nabla_{\theta_t} f(\theta_t)$: the gradient of the cost function

G_t: the diagonal matrix that contains the sum of the squares of the past gradients

ϵ: a smoothing term that avoids division by zero (usually on the order of 10^{-8})

$\widehat{m}$, $\hat{v}_t$: the bias-corrected first and second-moment estimate

β_1, β_2: the decay rate of the first and second moments

v_t: the estimate of the second moment
λ_1, λ_2: the L_1 and L_2 regularization strengths
α: the learning rate power

Regarding selecting a loss function, the mean squared error (MSE, **Eq. (147)**) [136] ensures better group fitting and thus is recommended. The number of epochs depends on data size, and this study recommends 600 epochs for sufficient learning. The batch size is a special hyperparameter related to the memory space of the graphics processing unit (GPU) and the learning depth of a minibatch. This study equips an Nvidia RTX 4090 (24 GB) to train models. Each model training adopts a batch size of 4,096, considering the trade-off between time consumption and training efficiency.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - y_p\right)^2 \tag{147}$$

where

y_i: the i-th measurement
y_p: the corresponding model prediction

It is worth mentioning that the network configurations from **Table 5.3** to **Table 5.5** are the results of the grid search optimization [148]. The hyperparameter tuning of the network size adopts a step of 128 kernels of the RBF layer and a step of 16 neurons of each learning dense layer. The listed configurations can suit the three turbines mentioned in **Chapter 3**. A robust network should have tolerance for its hyperparameter settings, i.e., similar and effective configurations should yield the same accuracy. Adding a dense layer will help improve learning depth if the training outcome diverges significantly.

5.4 Evaluation Criteria

Model evaluation adopts three indexes: the r-square (R^2, **Eq. (148)**) [136], the median absolute error (MAE, **Eq. (149)**) [82], and the root mean squared error (RMSE, **Eq. (150)**) [131]. The R^2 is a correlation coefficient that measures consistency between model predictions and measurements. The MAE focuses more on the central part of data distribution and eliminates the influence of some exceptional cases from the head and tail.

The RMSE is a classic statistical metric that indicates the divergence of the group error of forecasted values.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_i - y_p\right)^2}{\sum_{i=1}^{N} y_p^2} \tag{148}$$

$$\mathrm{MAE} = \mathrm{median}\left(\left|y_i - y_p\right|\right) \tag{149}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - y_p\right)^2} \tag{150}$$

Chapter 6 Wind Power Strategy

As the operation of a WT contains a lot of nonlinearities, the power strategy following some rules provides essential parameters describing these nonlinearities for a controller. Present WTs almost adopt the MPS, i.e., the MPPT-based strategy, to govern their wind generation. With increasing wind energy applications, the MPPT-based MPS no longer meets the grid requirements of scheduling and dispatching. This study develops a novel OPS for the 2-DOF regulation of the RSC and PAC to capture and output flexible power. WTs are usually the 3-DOF system that also includes the YAC, which controls nacelle yaw for upwind or downwind. The power strategy only unwraps the wind reference computed by a wind processing (or forecasting) unit as the YAC objective, which means yaw regulation is not a core mission of the power strategy. Therefore, the power strategy mainly focuses on coordinating the RSC and PAC.

6.1 Maximum Power Strategy

The MPS intends to harvest wind energy as much as possible and makes two hypotheses. First, rotor speed varies with wind speed until it reaches the rated speed, while pitch angle keeps the default position in the RSC region. Second, the PAC region keeps the rated speed and changes the pitch angle to maintain the rated output. The power curve explicitly draws the relation between wind speed and output power under standard test conditions as an auxiliary tool to understand the generation process. **Figure 6.1** provides a power curve that forms four regions representing different operations [58]:

- Region i is ideal zero output below cut-in, where the rotor disk is free to accelerate until entering the generation zone;
- Region ii has growing wind power from cut-in to rated wind, which bypasses the PAC for maximum capture;
- Region iii ensures rated output from rated wind to cut-off, where the PAC is activated to limit power production by reducing the aerodynamic torque;
- Region iv corresponds to shutdown over cut-off for safety considerations, which relies on stall control and shaft brake.

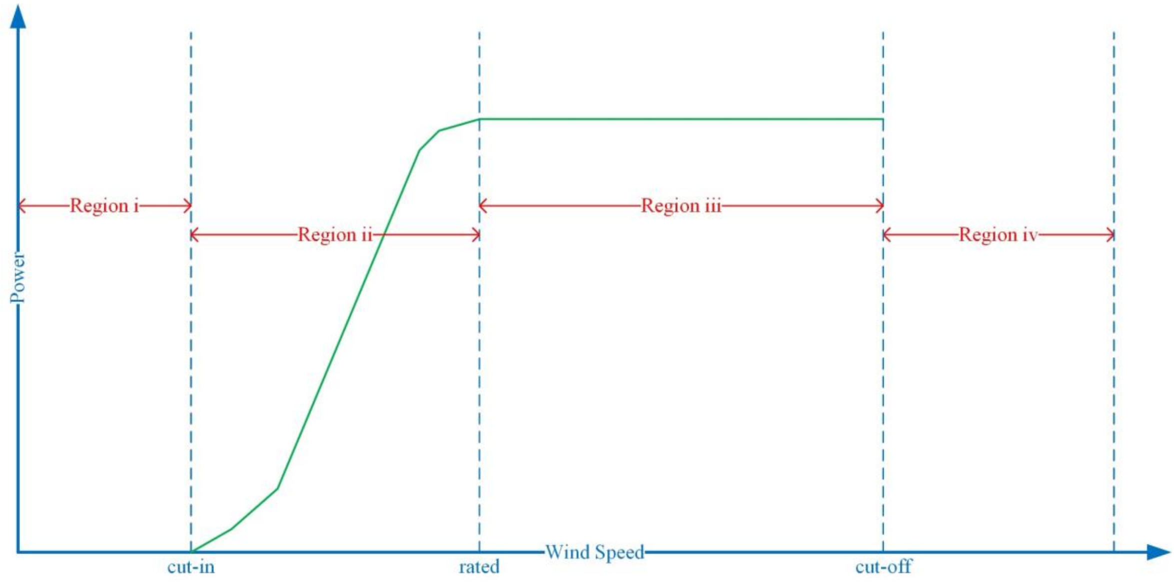

Figure 6.1 Region division of a variable speed and pitch-regulated turbine

Eq. (151) is the well-known power coefficient model of power extraction. The MPPT optimizes power coefficients to decide the system state for wind variation.

$$P_d = \frac{1}{2}\rho\pi R^2 C_p(\lambda, \beta) v_i^3 \tag{151}$$

$$\lambda = \frac{\omega_d R}{v_i} \tag{152}$$

where

$C_p(\lambda, \beta)$: the power coefficient (dimensionless)

λ: the tip speed ratio (TSR) (dimensionless)

Region ii commonly applies the optimal TSR that leads to a linear relation between speed reference and wind speed, represented by **Eq. (153)** [149].

$$\omega_d^{ref} = K_{op}^{tsr} v_i \tag{153}$$

where

K_{op}^{tsr}: the gain of the optimal TSR

Since Region ii increases wind power from zero, it is necessary to provide a power fitting to configure the generator current and voltage references. A popular way is the optimal torque control, which assumes that the received rotor torque is proportional to the squared rotor speed, as indicated in **Eq. (154)** [150].

$$Q_d^{ref} = K_{op}^{tor} \omega_d^2 \tag{154}$$

where

Q_d^{ref}: the torque reference (N)

K_{op}^{tor}: the gain of the optimal torque

However, the optimal torque control has a drawback that does not cover the region of lower power limited by the minimum rotor speed. Therefore, this study applies high-order curve fitting to provide corresponding power points in Region ii. **Figure 6.2** illustrates the MPPT-based RSC of the NREL 5-MW, which adopts the optimal TSR and high-order power fitting.

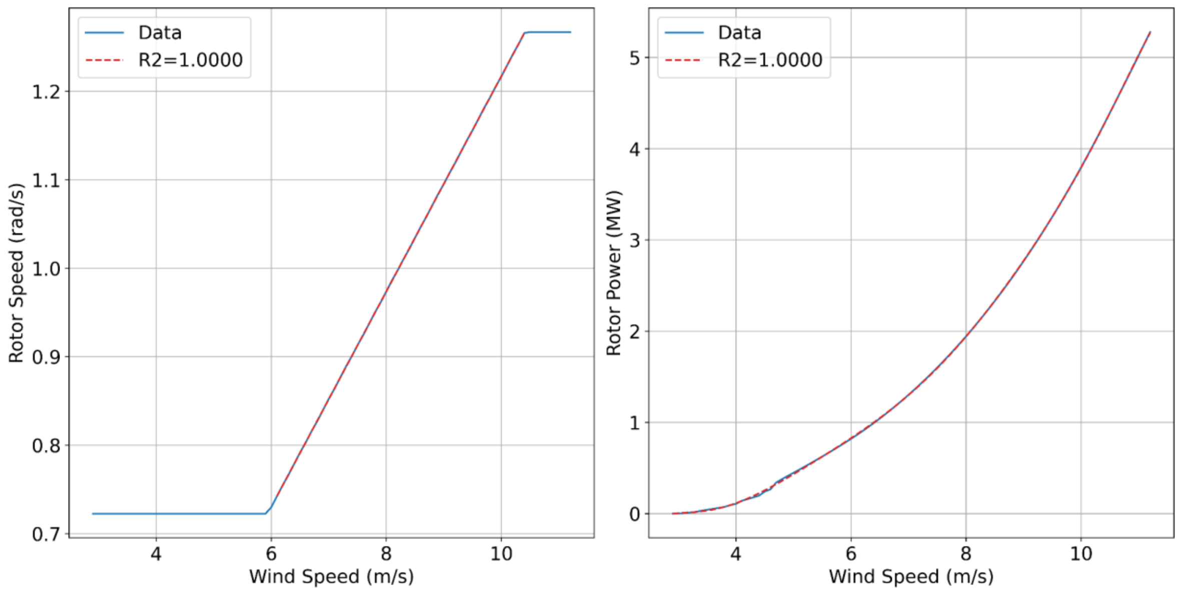

Figure 6.2 MPPT fittings of the NREL 5-MW in Region ii

The MPPT-based PAC is more complex due to gain scheduling and tower fore-aft damping. The basic idea of the PAC is to find a fitting relation between wind speed and pitch angle through several power points, which also find other fittings, such as sensitivities and tower displacement. **Figure 6.3** plots the PAC fittings regarding β_{ref}, $\frac{\partial Q_d}{\partial \beta}$, $\frac{\partial T_d}{\partial \beta}$, and x_m^{ref} in Region iii. Gain scheduling relis on $\frac{\partial Q_d}{\partial \beta}$ and $\frac{\partial T_d}{\partial \beta}$, and x_m^{ref} provides the steady-state bending position of the tower top.

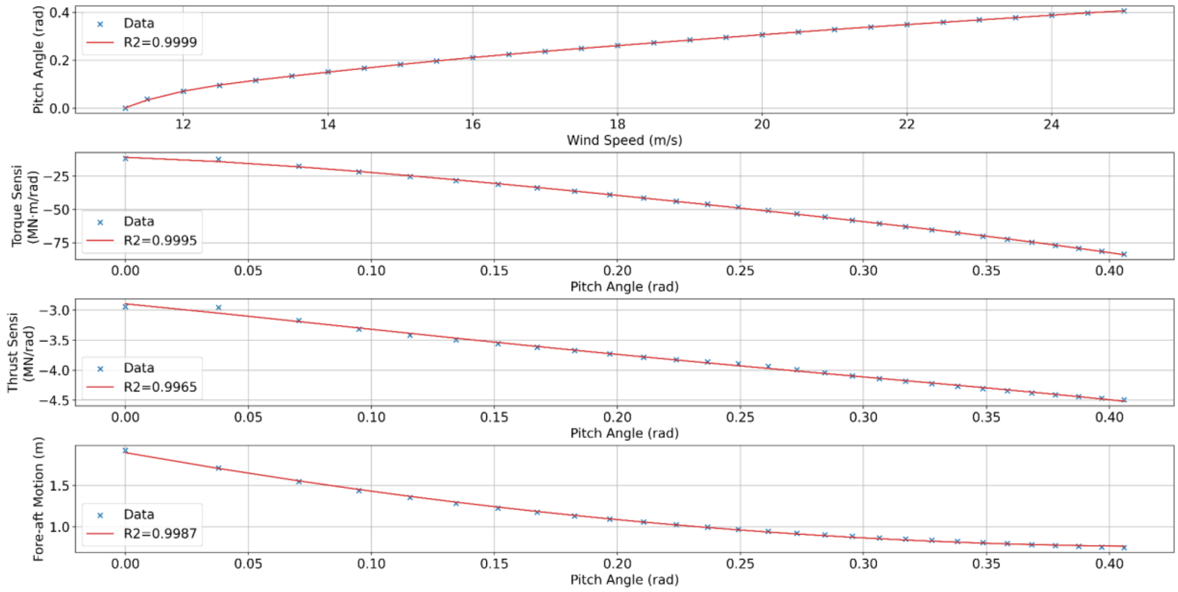

Figure 6.3 MPPT fittings of the NREL 5-MW in Region iii

Combining the MPPT fittings in Regions ii and iii, **Figure 6.4** coordinates the RSC and PAC to achieve the MPS of the NREL 5-MW. The main disadvantage of the MPS is a single output curve, in which each wind speed has a unique power. The following section will propose a novel power strategy to overcome this shortage.

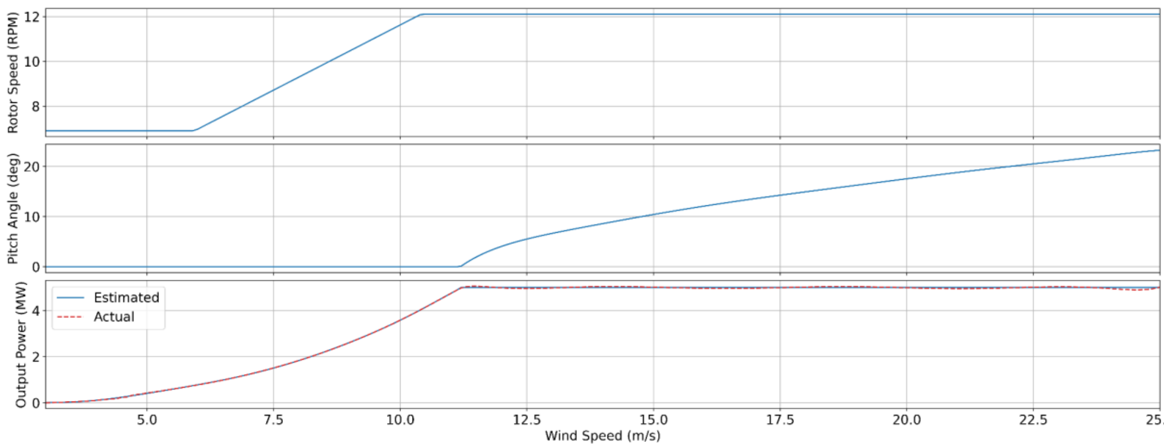

Figure 6.4 Coordination of the RSC and PAC of the NREL 5-MW

6.2 Online Power Strategy

The essence of the OPS is to find a proper 2-DOF state of rotor speed and pitch angle under a given wind speed, which intends to reduce the error between the power target (command) and the power estimation of an ANN-based aerodynamic model. Since the 2-DOF regulation needs to be continuous and smooth with wind variation, the OPS algorithm adopts a PRPT-based approach [54] to solve this online optimization. The PRPT uses the same division of the MPPT regions, as shown in **Figure 6.5**. The PRPT follows a similar rotor regulation of the optimal TSR to ensure enough power capture. The MPPT only adjusts the pitch angle of blades in Region iii, but the PRPT activates pitch regulation in Region ii for flexible output. It implies that the PRPT is an online extension of the MPPT. Based on the above, the OPS algorithm aims for a solution of the optimal speed and pitch (ω_d^{ref}, β_{ref}) to approach power target in the PRPT mode.

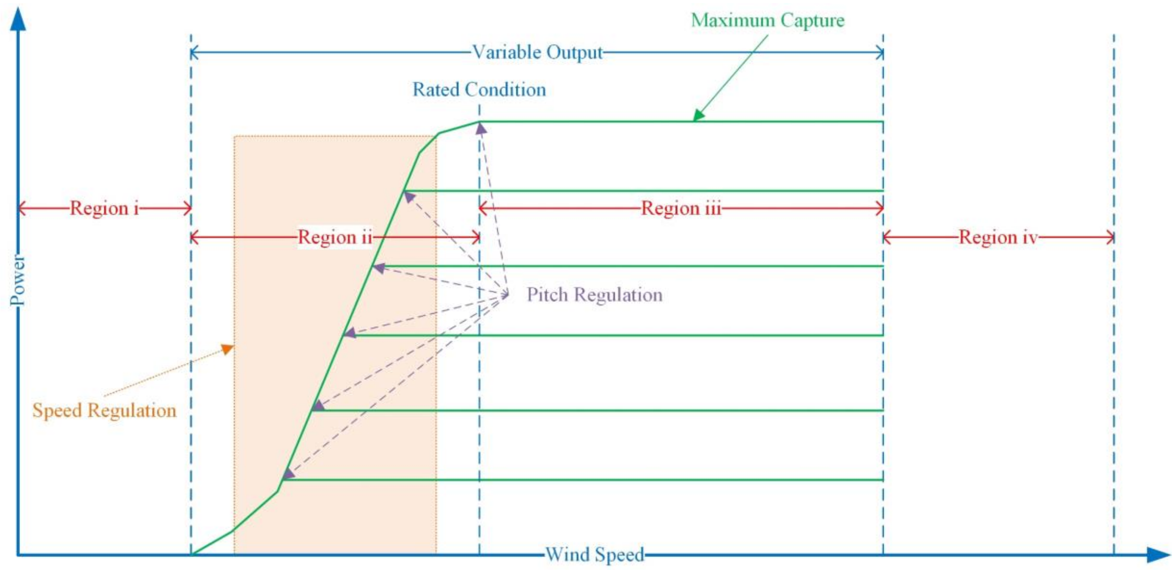

Figure 6.5 Region division of the MPPT and PRPT

Figure 6.6 explains the search procedure of the OPS algorithm. Firstly, the algorithm selects the output curve of zero pitch and finds the maximum output at this curve. Secondly, the algorithm tries to increase the pitch angle to lower power capture until it reaches the vicinity of a target power. Each step runs on a set of evenly distributed attempts for rotor speed or pitch angle, so the final answer has a limited resolution. The algorithm is essentially a variant of the grid search [151] that allows an attempt batch for group calculation. TensorFlow, or other ML platforms, has hardware (GPU-based) acceleration for batch input, which is much more efficient than a single update.

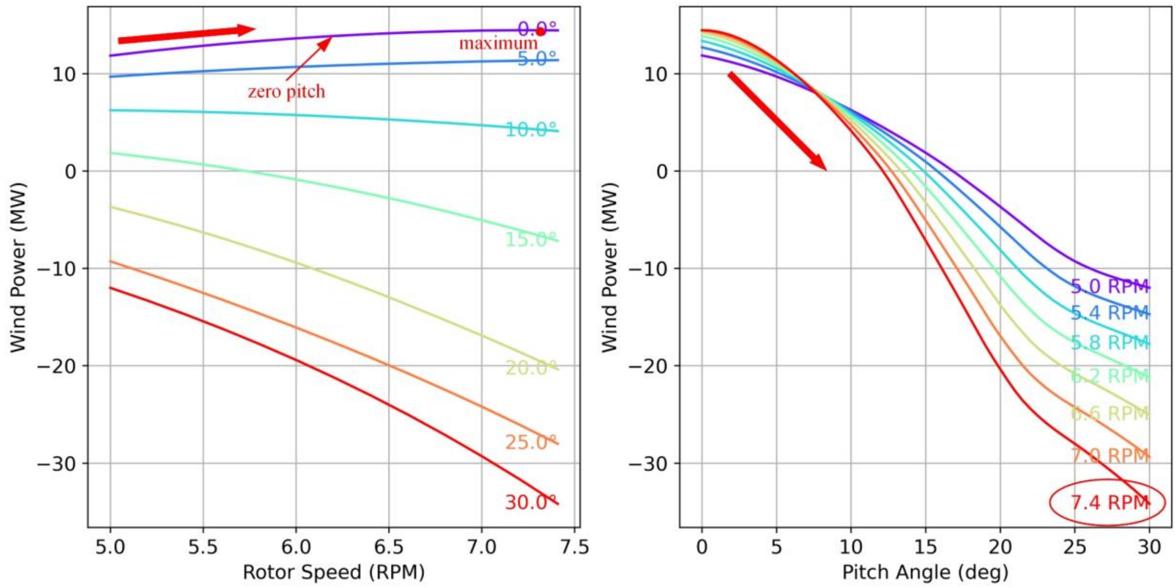

Figure 6.6 Grid search on the IEA 15-MW at a wind speed of 10 m/s

Although manufacturers provide rated wind in turbine specifications, its precision cannot meet the requirement of decimal places. **Algorithm 2** is a subalgorithm of the OPS that determines the rated wind and only needs to be executed in initialization. The basic idea is to compare model output with rated power under zero pitch. The returned value can divide Region ii and Region iii on an ANN model for the core algorithm (**Algorithm 3**).

	Algorithm 2 Rated Wind Determination
Input	variable $\tilde{x} \leftarrow [0 \quad \omega_d^{max} \quad \beta_{min}]$,
	aerodynamic model $\tilde{y} = \mathcal{M}(\tilde{x})$
Output	rated wind speed v_i^{rtd}
1.	allocate a grid sequence $d\tilde{x}_{v_i}$ for wind search
2.	update the input batch $\tilde{x} = \tilde{x} + d\tilde{x}$
3.	find $\tilde{x}_i$ where $\mathcal{J}(\tilde{x}) = \min\left(\left\lvert \tilde{y}\rvert_{P_\mathrm{d}} - P_d^{max}\right\rvert\right)$
4.	return v_i^{rtd} from $\tilde{x}_i$

Algorithm 3 contains a maximum capture estimation (**rows 1~9**) and a target matching (**rows 11~15**). The upper limit estimation sets the maximum power capture as a temporary power target when applying two procedures in **Figure 6.6**. This part aims to find the conventional maximum load and ensure the rotor power is in a reasonable range in case of irrational targets. Subsequently, the target matching increases the pitch angle to reduce the power prediction until it reaches the target power. If overloaded, the program will immediately terminate searching and return the solution of maximum capture, which

accounts for the logical condition of **rows 10 and 17**.

Algorithm 3 Batch-based Grid Search	
Input	average or forecasted wind speed v_i
	target output power P_t
	aerodynamic model $\tilde{y} = \mathcal{M}(\tilde{x})$
Output	speed and pitch references ω_d^{ref}, β_{ref}
	model output power P_d^{ref}
1.	initialize an origin $\tilde{x} \leftarrow [v_i \quad \omega_d^{max} \quad \beta_{min}]$
2.	**if** in Region ii ($v_i < v_i^{rtd}$):
3.	allocate a grid sequence $d\tilde{x}_{\omega_d}$ for speed search
4.	**else** (Region iii)
5.	allocate a grid sequence $d\tilde{x}_{\beta}$ for pitch search
6.	update the input batch $\tilde{x} = \tilde{x} + d\tilde{x}$
7.	make the aerodynamic prediction $\tilde{y} = \mathcal{M}(\tilde{x})$
8.	find $\tilde{x}_i$ where $\mathcal{J}(\tilde{x}) = \min(\lvert \tilde{y}\vert_{P_d} - P_d^{max} \rvert)$
9.	save the maximum capture $P_{max} = \tilde{y}\vert_{P_d}$ at $\tilde{x}_i$
10.	**if** the target is valid ($P_{max} > P_t$):
11.	reallocate a pitch search sequence $d\tilde{x}_{\beta}$
12.	update the input batch $\tilde{x} = \tilde{x}_i + d\tilde{x}$
13.	forecast again $\tilde{y} = \mathcal{M}(\tilde{x})$
14.	find new $\tilde{x}_i$ where $\mathcal{J}(\tilde{x}) = \min(\lvert \tilde{y}\vert_{P_d} - P_t \rvert)$
15.	accept the variable capture $P_d^{ref} = \tilde{y}_i\vert_{P_d}$ at $\tilde{x}_i$
16.	**else**
17.	accept the maximum capture $P_d^{ref} = P_{max}$
18.	return ω_d^{ref} and β_{ref} from $\tilde{x}_i$
19.	return T_d^{ref} and P_d^{ref} from $\tilde{y}_i = \mathcal{M}(\tilde{x}_i)$

Along with speed and pitch references, **Algorithm 3** also returns power and thrust predictions that estimate the steady state of the generator and tower. The speed and power references are not direct generator signals and require the current and voltage computations in **section 2.3**, while the pitch reference is a direct control input of the pitch servo.

After **Algorithm 3** determines a reference point, torque and thrust sensitivity pairs $(\frac{\partial Q_d}{\partial \omega_d}, \frac{\partial Q_d}{\partial \beta})$ and $(\frac{\partial T_d}{\partial \omega_d}, \frac{\partial T_d}{\partial \beta})$ can be known. The torque sensitivity pair provides a local linearized estimation of rotation trajectory, and the thrust sensitivity pair predicts an approximate variation of tower displacement. Both pairs are obligatory parameters for control configuration. With an ANN model, an approximation method [54] conveniently estimates four sensitivities near a reference point, expressed as **Eq. (155)**.

$$\left.\frac{\partial y}{\partial \Delta}\right|_{\tilde{x}_{ref}} = \frac{\mathcal{M}|_y(\tilde{x} - \Delta_l) - \mathcal{M}|_y(\tilde{x} + \Delta_r)}{\Delta_l + \Delta_r} \tag{155}$$

where

$\mathcal{M}|_y$: the model prediction for thrust T_d or torque Q_d

$\tilde{x}_{ref}$: the reference point determined by **Algorithm 3**

Δ_l, Δ_r: the left and right increments that represent speed ($\Delta\omega$) or pitch ($\Delta\beta$) linearization (tuneable)

Figure 6.7 sketches the linearization of **Eq. (155)**. In the middle of the operation region, this estimation strictly follows $\Delta_l = \Delta_r = \Delta$. However, the linearization needs to adjust the increaments as $\Delta_l + \Delta_r = 2\Delta$ when close to the variable boundary.

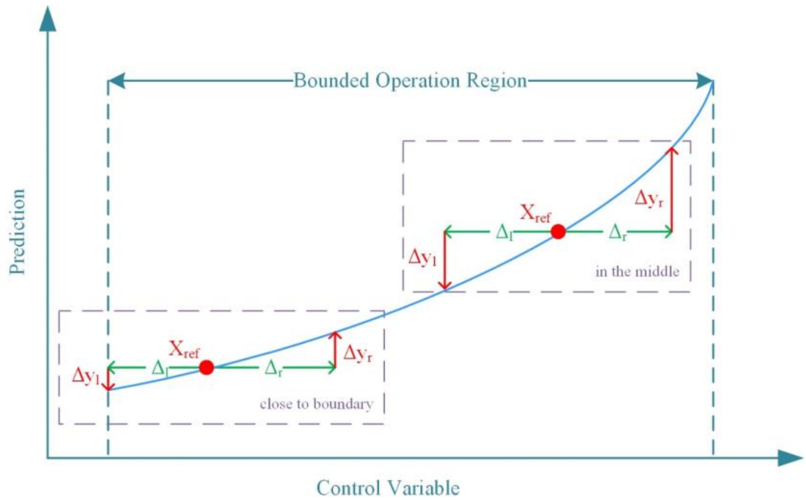

Figure 6.7 Local linearization of an aerodynamic prediction at the reference point

6.3 Wind Prediction Processing

After a wind model generates a series of wind vector predictions, a weighted average tool determines the average wind speed and direction of N predictions as wind reference. **Eqs. (156)(157)** compute the velocity and direction references. **Eqs. (158)(159)** weights wind components by assigning more importance to a recent prediction.

$$\bar{v} = \sqrt{\bar{v}_x^2 + \bar{v}_y^2} \tag{156}$$

$$\bar{\theta} = \arctan \frac{\bar{v}_x}{\bar{v}_y} \tag{157}$$

$$\bar{v}_x = \frac{\sum_{i=0}^{N-1}(N-i)v_x^i}{\sum_{n=1}^{N} n} \tag{158}$$

$$\bar{v}_y = \frac{\sum_{i=0}^{N-1}(N-i)v_y^i}{\sum_{n=1}^{N} n} \tag{159}$$

where

$\bar{v}$: the wind velocity prediction to update the power strategy (m/s)

$\bar{\theta}$: the wind direction prediction to determine the yaw reference (rad)

$\bar{v}_x$, $\bar{v}_y$: the averaged xy-axis wind components (m/s)

Since WTs need to avoid lateral force caused by yaw error, nacelle orientation expects to face wind flow, which infers yaw reference equals predicted direction. **Eq. (160)** unwraps the yaw angle for continuous yaw orientation. **Eqs. (161)(162)** calculate the clockwise or counterclockwise nacelle rotation, and **Eq. (160)** selects the minimum rotating angle. After the above, the power strategy obtains the yaw reference and directly passes it to the control system.

$$\gamma_{ref} = \begin{cases} \gamma'_{ref} + \gamma^+, \text{if } \gamma^+ < |\gamma^-| \\ \gamma'_{ref} + \gamma^-, \text{else} \end{cases} \tag{160}$$

$$\gamma^+ = \mathrm{mod}\left(\bar{\theta} - \gamma'_{ref}, 2\pi\right) \tag{161}$$

$$\gamma^- = \gamma^+ - 2\pi \tag{162}$$

where

γ_{ref}: the present yaw reference (rad)

γ'_{ref}: the last yaw reference (rad)

Chapter 7 Intelligent Power Strategy

Section 6.2 proposes a PRPT-based OPS solution, but it is an extension of the conventional theories and does not weigh priorities between speed and pitch regulation. Speed regulation might lead to long rotation settlement due to large system inertia and extra power fluctuation due to kinetic energy variation, which affects kinetic stability and power quality. Although pitch regulation does not result in such kinetic and power costs, a turbine system cannot only rely on pitch regulation for a sufficient range of adjustable power capture. Fortunately, RL is a powerful tool to address such multi-factor optimization problems [152]. Hence, this study also presents an RL-based OPS for more intelligent 2-DOF coordination that can consider the costs of regulating speed and pitch when responding to the power command. **Figure 7.1** draws the flowchart from decision-making to control actuation. Two networks serve aerodynamic modelling and decision-making, respectively. Essentially, the eager policy plays a role in a weighted 2-DOF optimization of speed and pitch on the turbine model.

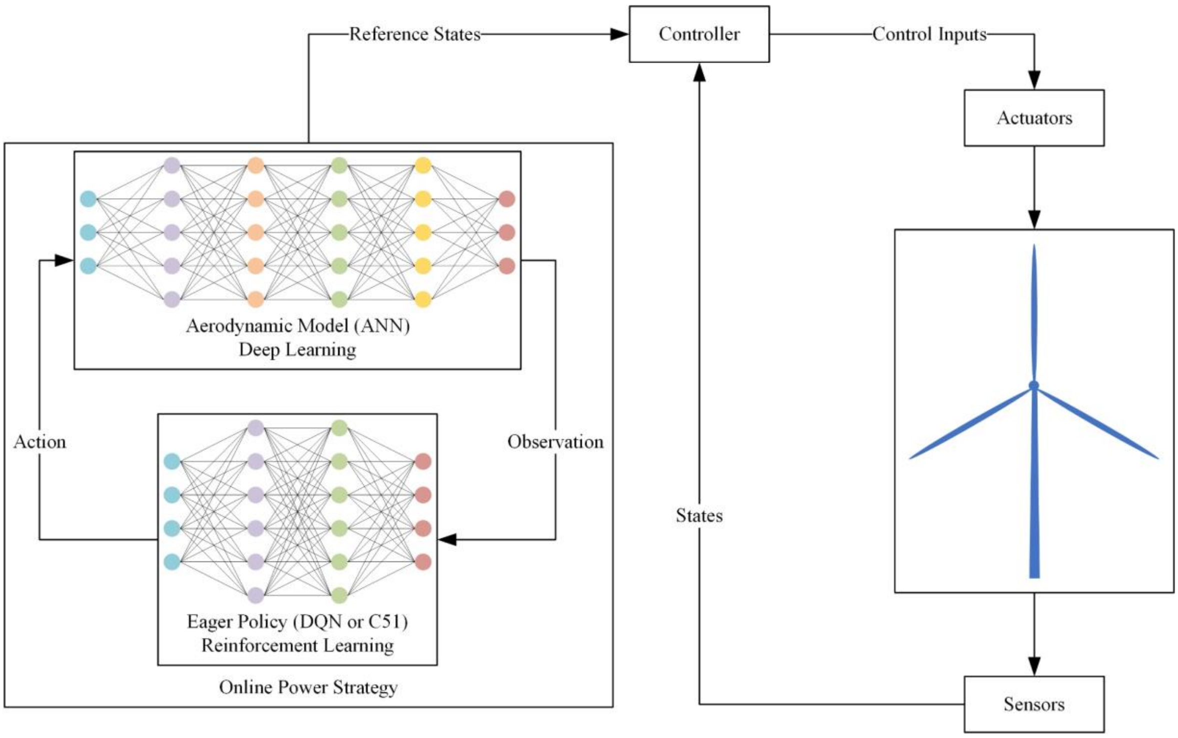

Figure 7.1 Flowchart of the RL-based OPS with model-policy interaction

7.1 Learning Framework

Since the coordinated regulation of speed and pitch aims to minimize the error between power capture and target power, the RL objective is to obtain an RL policy that derives the speed and pitch references considering the state transition cost from present to next. This study performs RL training with TF-Agents, an extension of TensorFlow [153]. TF-Agents provides open-source APIs to train agents without considering any general RL frameworks. The learning framework of TF-Agents includes an environment, an agent, a driver, and a replay buffer, as shown in **Figure 7.2**. The environment wraps and batches the aerodynamic solver within the maximum episode to execute and evaluate actions. The agent module provides a variety of agent APIs, which build multiple networks as their RL policies and accept external optimizers for training. This study imports the Adam optimizer from TensorFlow to train the agent network. The policy here corresponds to an RL terminology that maps an environment observation to an action or a distribution over actions. The driver module observes the environment and drives the agent to execute its policy, which the replay buffer collects as a trajectory to feed the network training. After training, the agent saves its policy as an eager policy that directly returns optimal action for current observation.

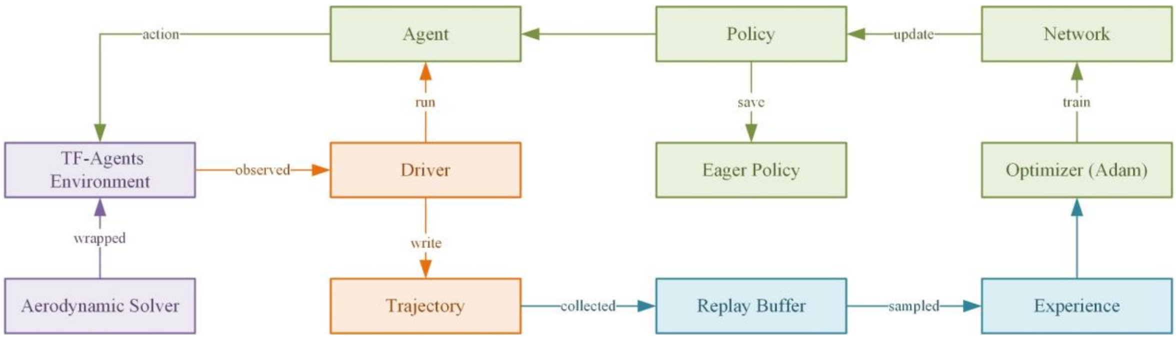

Figure 7.2 TF-Agents learning framework for the OPS

This study defines each observation ($s \in \mathcal{S}$) that contains four variables: power target P_t, rotor speed ω_d, pitch angle β, and power capture P_d, given as **Eq. (163)**. In this definition, the rotor speed and pitch angle are control variables. The power target is an external variable that the user sets. The power capture indicates the current state of generation.

$$s = [P_t \quad \omega_d \quad \beta \quad P_d] \tag{163}$$

The action space (**Table 7.1**) affecting wind capture consists of discrete steps. There are two problems when using these discrete actions. Firstly, action resolution has a dominant influence on final accuracy. Secondly, shorter steps benefit accuracy but slow searching or get trapped in local optima. On the contrary, longer steps can accelerate computation and avoid traps but cannot ensure satisfactory precision. The following section will introduce a novel algorithm for the bisected update of actions to overcome the resolution problem caused by discrete steps.

Table 7.1 Action space for controlling rotor speed and pitch angle

Code	Action ($a \in \mathcal{A}$)	Comment
0	$+\Delta_{\omega_d}$	bisected variable (tuneable)
1	$-\Delta_{\omega_d}$	same as above
2	$+\Delta_{\beta}$	same as above
3	$-\Delta_{\beta}$	same as above

7.2 Bisected Action

This section proposes a novel bisected movement instead of fixed steps to address the above resolution issues. The basic principle is that an RL agent steps back and forth if it is close to but does not reach an optimum. This characteristic inspires us to reduce (or bisect) the current step length when going by the optimum again.

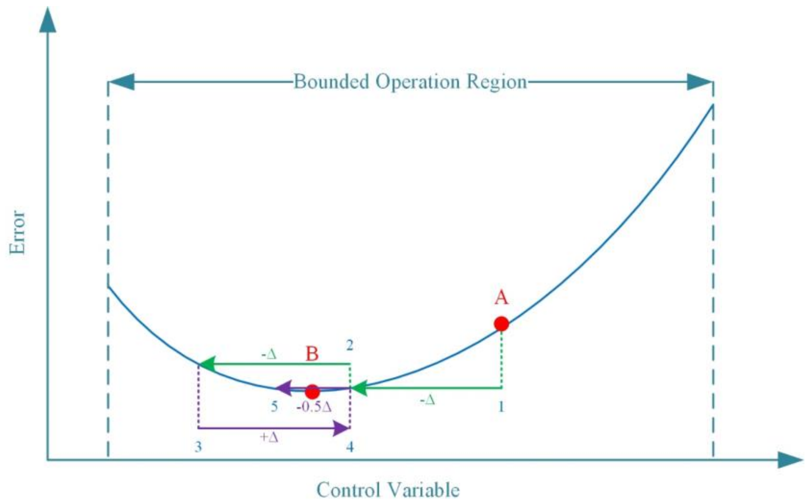

Figure 7.3 Process of the bisected action approaching an optimum

Figure 7.3 illustrates a process of the bisected action. The agent starts at the A site and moves towards the B optimum. Breakpoints 2 and 4 are at the same position, so the agent bisects its step when revisiting this position. Therefore, the agent stops moving around and infinitely approaches the B site. The bisection process is not a strict MDP, but a broad sense of forward and backward movements is a pseudo-MDP. The reason is that approaching or going away from an optimum still obtains the deserved rewards. Considering this pseudo-MDP process, the discount of a collected trajectory has to be a small value to ensure the training convergence, in which 0.1~0.3 can balance experience and exploration.

The bisection realization requires a similarity examination if the agent revisits a state. For this purpose, we introduce a first-in-first-out (FIFO) queue, denoted as L_f, to record the history of the explored state. The front queue members are more likely to repeat, which is the advantage of the FIFO queue over the first-in-last-out (FILO) stack. Besides, each action is assigned a local coefficient (K_a) to perform bisection. **Algorithm 4** is a pseudo program traversing the FIFO and updating the action step. The comparison (**row 2**) between the present state and a FIFO member asserts both are similar if each variable difference is less than the threshold ε. If the FIFO has a similar member, then the action coefficient K_a^m at the m-th movement recursively multiplies the bisection coefficient K_b (**rows 1~3**). Subsequently, the FIFO removes the last member and appends the current state in the first place (**rows 4~5**). Lastly, **Algorithm 4** updates the action step after the bisection search.

	Algorithm 4 FIFO-based Bisection
Input	FIFO queue $L_f = \{l_1, l_2, \dots, l_f\}$
	agent action code a, rotor speed ω_d, pitch angle β
Output	action step Δ'_a
1.	**for** l_i in L_f:
2.	**if** $a = a_i$ and $\lvert\omega_d - \omega_d^i\rvert < \varepsilon_{\omega_d}$ and $\lvert\beta - \beta_i\rvert < \varepsilon_\beta$:
3.	update the individual action coefficient recursively
	$K_a^m = K_b K_a^{m-1}$
4.	pop out the last member l_f
5.	amend a new member in the first place
	$l_0 = [a \quad \omega_d \quad \beta]$
6.	update the action step as
	$\Delta'_a = K_a^m \Delta_a$
7.	return and execute Δ'_a

Since any turbine operation has control boundaries for speed and pitch, it is necessary to consider the agent's behaviour at the boundaries. This study adopts a circular movement when the agent exceeds a boundary, as shown in **Figure 7.4**. If an action step causes going outside a lower or upper limit, the agent will come out from the opposite side. This moving method ensures that the agent will not get stuck at the boundaries but may cause frequent traversal. Therefore, the reward should add a penalty if the agent crosses a boundary.

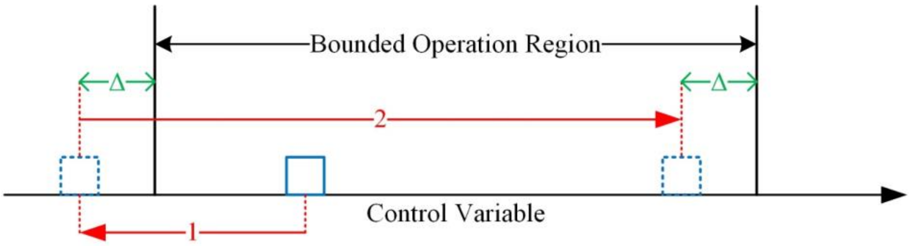

Figure 7.4 Update of the agent position when exceeding a region boundary

7.3 Reward

Regarding the reward of an action, a straightforward criterion is if the agent reduces the output error ($|P_d - P_t|$). However, it is insufficient to reflect the cost of an action because the alternation of the RSC is much more expensive than that of the PAC. The generator has to change its output to affect the electromagnetic torque for acceleration or brake, which leads to significant power fluctuation. Therefore, pitch regulation has a higher priority than speed regulation if both achieve positive rewards, which the action reward should consider. **Table 7.2** is a reward table that offers preference for pitch regulation.

Table 7.2 Reward table for evaluating the actions

Action	Reward Value	Comment
$\pm\Delta_{\omega_d}$	+7	closer to target
$\pm\Delta_{\omega_d}$	-3	away from target
$\pm\Delta_{\beta}$	+10	closer to target
$\pm\Delta_{\beta}$	-1	away from target
exceed boundary	-5	accumulated

7.4 Agent Network

This study implements two agents for discrete action. The first agent is the well-known DQN, which combines RL with ANNs and overcomes the limitations of selected features or fully observed low-dimensional state spaces [154]. The DQN observes the state of the controllable object and computes the Q-values of all actions as output, which assists the agent in selecting the optimal action [155]. **Eq. (164)** transforms a parameter vector ($\theta^{s,a}$) from the table function family into a step of the gradient optimization, where $\theta^{s,\,a}$ has a relation of $\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ [156].

$$Q^*(s, a, \theta) = \theta^{s,\,a} \tag{164}$$

An instance of TF-Agents uniformly draw samples or mini-batches of experience from the pool in the replay buffer, i.e., $s' \sim p(s'|s, a)$. For the sample batch at each timestep, **Eq. (165)** presents the regression target the model tries to predict [156].

$$y_t = r(s') + \gamma \max_{a'} Q^*(s', a', \theta^-) \tag{165}$$

Therefore, the DQN update at each iteration uses the loss function (**Eq. (166)**) and proceeds a gradient descent step of **Eq. (167)** [154].

$$\mathcal{L}_{s,\,a}(\theta) = \mathbb{E}_t[(Q^*(s, a, \theta) - y_t)^2] \tag{166}$$

$$\nabla_\theta = \frac{\mathcal{L}_{s,\,a}(\theta)}{\partial\theta} \tag{167}$$

The above DQN models the Q-value of the state-action pair, which an agent commonly receives. Considering the distribution of the random return, a distributional Q-learning uses Bellman's equation to model the probability distribution of the Q-value [157]. Due to restrictions of parametric distributions and only available samples from $p(s'|s, a)$ for each update [156], a categorical DQN (CDQN or C51) is introduced to achieve a practical algorithm design [158]. The C51 corresponds to a parameter that sets distributional atoms to 51 [158]. Owing to the Q-value distribution rather than expectation, the C51 assures more stable training and thus improves model performance.

Assuming the distribution of random variables as $Z_\theta^*(s, a)$ and the undetermined parameters as $\zeta_\theta(s, a)$, each transition computes the projection of y_t onto the support $\{z_i\}$, which yields **Eq. (168)** [156].

$$\mathcal{P}(y_t = r' + \gamma z_i) = \zeta_i^*\left(s',\ \max_{a'} \sum_i z_i \zeta_i^*(s', a', \theta^-), \theta^-\right) \tag{168}$$

Target and model output are distributions so that the loss function can be a form of divergence $\mathcal{D}$ between y_t and $Z_\theta^*(s, a)$ [157]. As $\mathcal{D}$ is an unbiased estimation of the gradient of the KL divergence for full distribution, the sample loss $\mathcal{L}_{s,\,a}(\theta)$ of the CDQN is the cross-entropy term of the KL divergence, represented by **Eq. (169)** [157].

$$\mathcal{L}_{s,\,a}(\theta) = \mathbb{E}_t\left[\mathcal{D}_{KL}\left(y_t || Z^*(s, a, \theta)\right)\right] \tag{169}$$

Table 7.3 summarizes the configuration of two agents and their networks. The DQN receives an external sequential dense network with the ReLU activation as the Q-network. The C51 adopts a categorical Q-network from TF-Agents [153]. Both agent training adopts the dynamic step driver provided by TF-Agents [153]. According to debugging observations, a single layer often leads to a lack of learning, but multiple layers (≥3) have difficulties in convergence. The two-layer network can make a great trade-off between learning depth and training efficiency. Meanwhile, both layers should have the same number of neurons, considering network input and output dimensions. Based on the above, hyperparameter tuning gradually increases neurons with a step of 128 neurons until the agent has no improvement in random tests.

Table 7.3 Summary of two agents and networks

Agent	Network	Layers	Driver Type	Comment
DQN	fully-coupled network	512, 512	step	ReLU activation
C51	categorical Q-network	512, 512	step	51 atoms

7.5 Control Configuration

An RL policy and an ANN model assemble an intelligent turbine system that relies on its experience and prediction to search for the 2-DOF solution. **Figure 7.5** illustrates how the RL eager policy uses the ANN model to update control configuration.

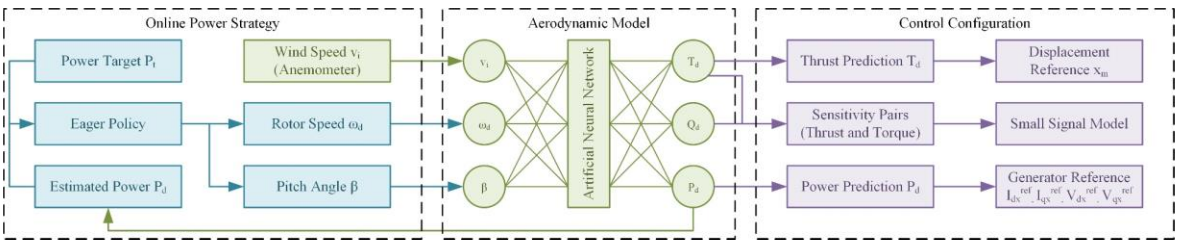

Figure 7.5 Flowchart of the RL-based OPS for each control update

The RL-based OPS utilizes the eager policy to find the 2-DOF solution on the aerodynamic model for a given output target and a measured wind speed. Subsequently, the RL-based

OPS also applies **Eq. (155)** to update the sensitivities of torque and thrust. Finally, the control system receives the 2-DOF answer as its next regulation objective and updates the loop gains or the small-signal model according to the selected control mode.

Chapter 8 Control Design

When a power strategy infers the reference state (speed, pitch, thrust, power, sensitivities, and yaw), the controller governs the generator, pitch servo, and yaw servo for the 3-DOF regulation. First, the generator adjusts the electromagnetic torque and absorbs the shaft's kinetic energy, which drives the main shaft at the speed reference, which is the principle of the RSC [117]. Second, the pitch servo tracks the pitch reference and moderately changes the angle position according to rotation and power output, which is the purpose of the PAC [28]. Third, the generator and pitch servo cooperatively ensure a smooth variation of thrust acting on the rotor disk to alleviate the vibration of tower fore-aft motion. Fourth, the yaw servo ensures the windward operation of the rotor disk to reduce the lateral component of wind, which is the primary function of the YAC [104]. This section introduces four terminal controllers: PID, LQR, RHC, and MPC. The PID design refines existing PID loops, while the other three adopt innovative small-signal models (**Eqs. (121)(122)**).

8.1 Proportional Integral Derivative

The PID is a classic controller that continuously calculates an error value as the difference between setpoint and process variable and applies a correction based on proportional, integral, and derivative terms [159]. Since a native PID only relies on the error to make a control decision and does not require a predictive model, the PID is a typical model-free approach to realize turbine control.

This study investigates present turbine PIDs and takes their most attractive features to establish a comprehensive PID control system, as displayed in **Figure 8.1**. The fundamental PID loop comes from the FAST baseline, which involves a generator torque loop, a pitch servo loop, and a speed-to-pitch loop [96]. The FAST also introduces a method of pitch-gain scheduling [31] to allow the optimal gain configuration of the speed-pitch loop. A DFIG loop [110] or PMSG loop [30] replaces the FAST's first-order torque control for real-world generators. The regulation of the q-axis current of the generator has a speed-to-current loop to compensate for the speed error caused by wind uncertainties and other perturbations. The PID design also includes a motion-to-pitch loop for tower damping control [26]. As a supplement to the YAC, the PID design has an additional yaw loop to change nacelle orientation.

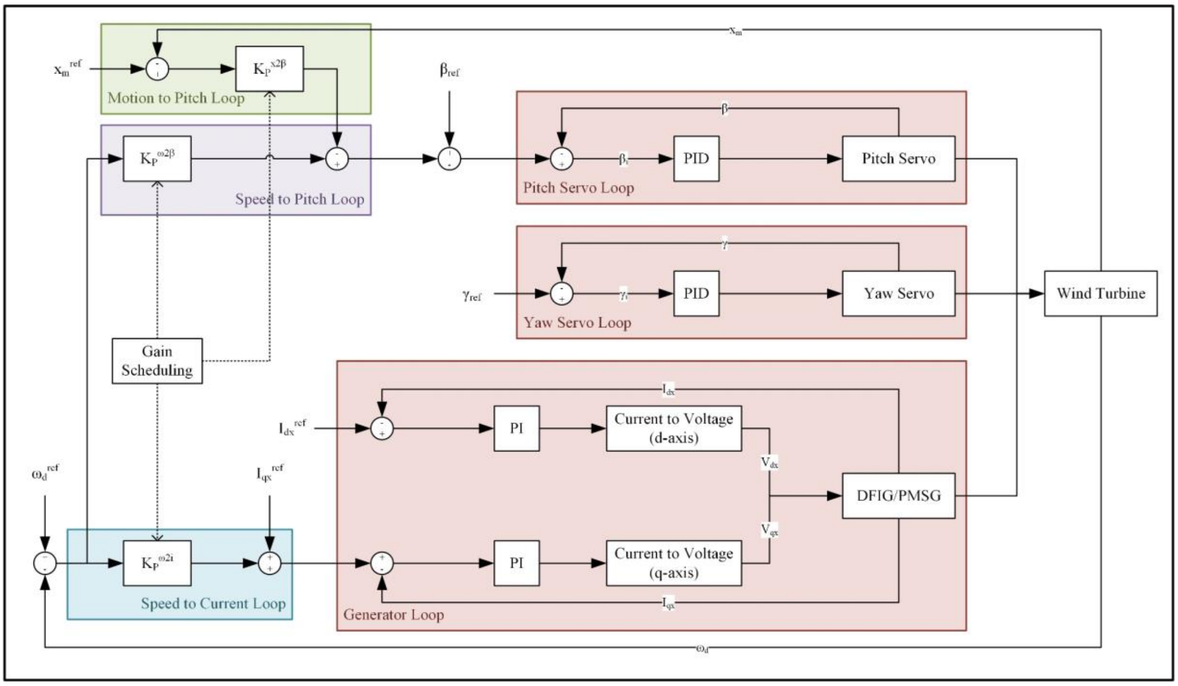

Figure 8.1 PID loop design for model-free control

The generator loop has two independent PI controllers to minimize the dq-axis current errors (ΔI_{dx}, ΔI_{qx}) according to the pre-calculated current references (I_{dx}^{ref}, I_{qx}^{ref}), as given in **Eqs. (170)(171)**. The DFIG takes the result of **Eqs. (79)(80)** as input and uses **Eqs. (73)(74)** to convert the current signals to the final voltage inputs. The PMSG receives the result of **Eqs. (103)(104)** as setpoint and updates the voltage variables with **Eqs. (98)(99)**.

$$\Delta I_{dx} = \left(K_P^{ix} + \frac{K_I^{ix}}{s}\right)\left(I_{dx}^{ref} - I_{dx}\right) \tag{170}$$

$$\Delta I_{qx} = \left(K_P^{ix} + \frac{K_I^{ix}}{s}\right)\left(I_{qx}^{ref} + dI_{qx} - I_{qx}\right) \tag{171}$$

where

K_P^{ix}: the proportional gain of the dq-axis currents

K_I^{ix}: the integral gain of the dq-axis currents

dI_{qx}: the correction item resulted from the speed-to-current loop

The optimal gain for the generator current control results from a second-order analysis [160] of the shaft-generator system. **Eqs. (172)(173)** computes the corresponding proportional and integral gains. The dq-axis current loops apply the same gain configuration, considering the synchronous regulation of apparent power.

$$K_P^{ix} = K_{ix}\frac{2J_tK_{ds}K_{\omega s}}{-n_g\dfrac{\partial Q_e}{\partial I_x}} \tag{172}$$

$$K_I^{ix} = K_{ix}\frac{J_tK_{\omega s}^2}{-n_g\dfrac{\partial Q_e}{\partial I_x}} \tag{173}$$

where

K_{ds}: the damping ratio of the main shaft (tuneable)

$K_{\omega s}$: the natural frequency of the main shaft (tuneable)

K_{ix}: the intensity of the current regulation (tuneable)

$\frac{\partial Q_e}{\partial I_x}$: the sensitivity of the generator torque regarding the current change

The generator sensitivity is an intrinsic constant that depends on electric characteristics. **Eqs. (174)(175)** estimate its value for the DFIG and PMSG, respectively, which come from the derivative of **Eqs. (75)(100)**.

$$\frac{\partial Q_e}{\partial I_x} = -\frac{3p}{k_g}\frac{L_m}{L_s}\frac{V_g}{\omega_e} \tag{174}$$

$$\frac{\partial Q_e}{\partial I_x} = -\frac{3p}{k_g}\psi_f \tag{175}$$

The pitch servo loop involves three items due to a large time constant, as given in **Eq. (176)**. Meanwhile, the classic analysis of the first-order closed loop yields the optimal PI gains for the pitch servo, as **Eqs. (177)(178)** [159]. Its derivative gain requires manual tuning to balance response and stability.

$$\Delta\beta = \left(K_P^\beta + \frac{K_I^\beta}{s} + K_D^\beta s\right)\left(\beta_{ref} - d\beta - \beta\right) \tag{176}$$

$$K_P^\beta = 2\tau_\beta K_{d\beta}K_{\omega\beta} - 1 \tag{177}$$

$$K_I^\beta = \tau_\beta K_{\omega\beta}^2 \tag{178}$$

where

K_P^β: the proportional gain of the pitch servo

K_I^β: the integral gain of the pitch servo

K_D^β: the derivative gain of the pitch servo (tuneable)

$d\beta$: the correction item resulted from the speed-to-pitch and motion-to-pitch loops

$K_{d\beta}$: the damping ratio of the pitch servo (tuneable)

$K_{\omega\beta}$: the natural frequency of the pitch servo (tuneable)

By applying a similar analysis of the pitch servo on the yaw servo, **Eq. (179)** is in charge of the nacelle navigation, and **Eqs. (180)(181)** computes the optimal PI gains.

$$\Delta\gamma = \left(K_P^\gamma + \frac{K_I^\gamma}{s} + K_D^\gamma s\right)\left(\gamma_{ref} - \gamma\right) \tag{179}$$

$$K_P^\gamma = 2\tau_\gamma K_{d\gamma} K_{\omega\gamma} - 1 \tag{180}$$

$$K_I^\gamma = \tau_\gamma K_{\omega\gamma}^2 \tag{181}$$

where

K_P^γ: the proportional gain of the yaw servo

K_I^γ: the integral gain of the yaw servo

K_D^γ: the derivative gain of the yaw servo (tuneable)

$K_{d\gamma}$: the damping ratio of the yaw servo (tuneable)

$K_{\omega\gamma}$: the natural frequency of the yaw servo (tuneable)

The q-axis current loop cascades a speed-to-current loop to compensate for the speed error, which yields a pure proportional term (**Eq. (182)**).

$$dI_{qx} = K_P^{\omega i} \Delta\omega_d \tag{182}$$

$$K_P^{\omega i} = K_{\omega i} \frac{K_{pi} P_d^{ref}}{\omega_d^{ref}} \tag{183}$$

$$K_{pi} = \frac{k_g}{3V_g(1-s)} \frac{L_s}{L_m} \tag{184}$$

$$K_{pi} = \frac{k_g}{3V_g} \tag{185}$$

where

$K_P^{\omega i}$: the gain of the speed-to-current loop

K_{pi}: a constant of converting the power reference to the q-axis current reference (**Eq. (184)** for the DFIG, **Eq. (185)** for the PMSG)

$K_{\omega i}$: the intensity of the speed-to-current loop (tuneable)

Aside from the main task of varying power capture, the PAC also plays a role in alleviating the rotor speed error and damping the tower top oscillation, as indicated in **Eq. (186)**.

$$d\beta = d\beta_{\omega_d} - d\beta_{x_m} \tag{186}$$

$$d\beta_{\omega_d} = K_P^{\omega\beta}\Delta\omega_d \tag{187}$$

$$d\beta_{x_m} = K_P^{x\beta}\Delta x_m \tag{188}$$

$$K_P^{\omega\beta} = \frac{2J_t K_{ds} K_{\omega s}}{-\frac{\partial Q_d}{\partial \beta}} \tag{189}$$

$$K_P^{x\beta} = K_{xb}\frac{D_{tm}}{-\frac{\partial T_d}{\partial \beta}} \tag{190}$$

where

$d\beta_{\omega_d}$: the pitch compensation for the error of rotor speed

$d\beta_{x_m}$: the pitch compensation for the displacement of tower fore-aft motion

$K_P^{\omega\beta}$: the gain of the speed-to-pitch loop

$K_P^{x\beta}$: the gain of the motion-to-pitch loop

K_{xb}: the intensity of the motion-to-pitch loop (tuneable)

The control system sets the gains of **Eqs. (172)(173)(177)(178)(180)(181)** at the stage of controller initialization and updates the gains of **Eqs. (183)(189)(190)** for any wind change. The gain update of the speed-to-current, speed-to-pitch, and motion-to-pitch loops is named gain scheduling, which aims for optimal dynamic response under nonlinear aerodynamics.

8.2 Model-based Control

When a power strategy issues reference states, including speed, pitch, yaw, power, and thrust, the steady-state calculation of the DFIG (**Eqs. (76)(77)(79)(80)(73)(74)**) or PMSG (**Eqs. (101)(102)(103)(104)(98)(99)**) converts a power reference to the current and voltage signals, and **Eq. (112)** gives the steady-state position of tower bending. Simultaneously, the power strategy updates force sensitivities in a small-signal model (**Eq. (121) or (122)**). After these, a model-based controller can optimize response trajectory and take control actions.

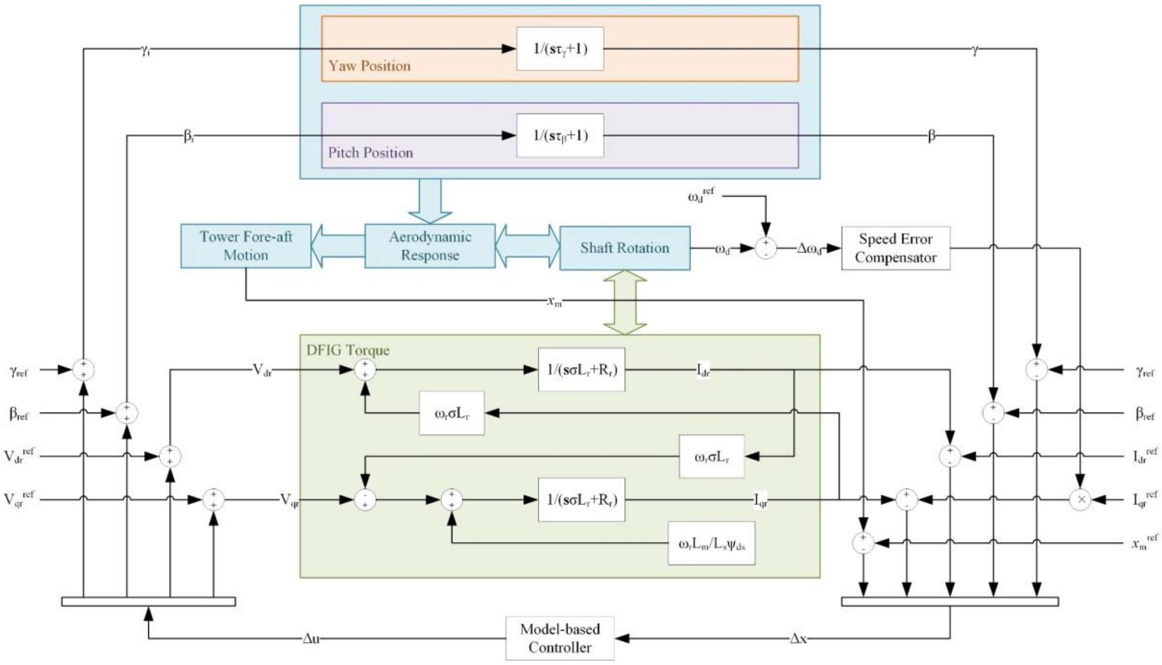

Figure 8.2a DFIG rotor-side synthesis control

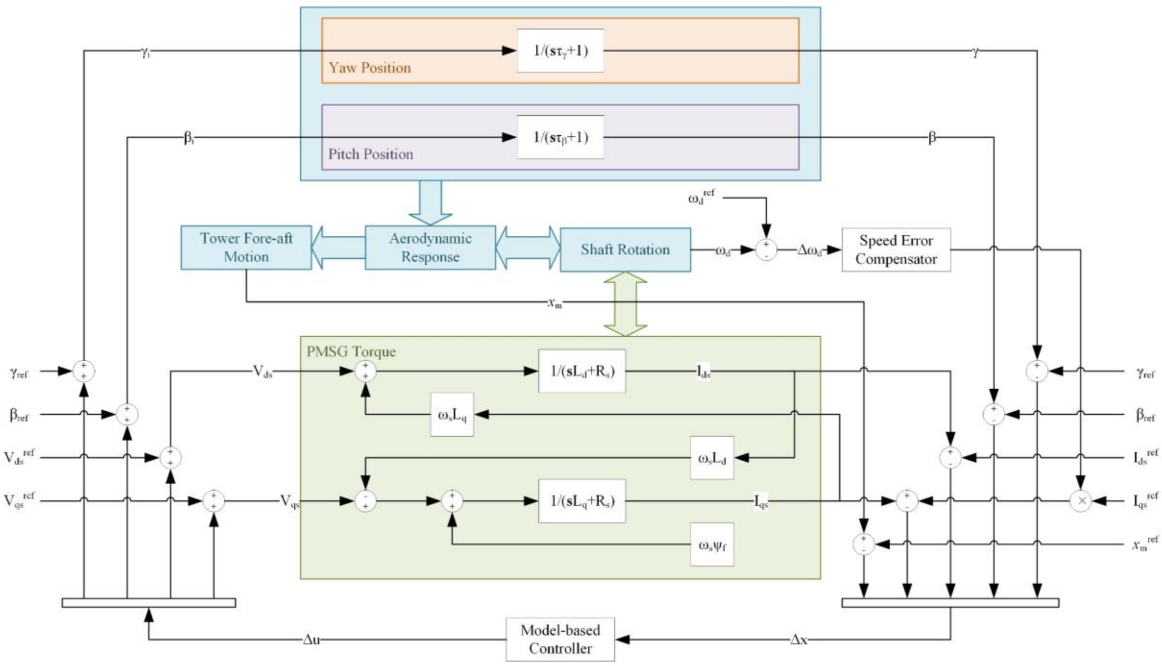

Figure 8.2b PMSG stator-side synthesis control

Figure 8.2 Feedback design for model-based control

Figure 8.2 shows the model-based feedback loop for the DFIG-based or PMSG-based system. The control configuration remains unchanged until the power strategy gets updated. This study proposes three compatible controllers, i.e., LQR, RHC, and MPC, and their main difference is the calculation method of control actions. Compared with the PID, a model-based controller coordinates the generator, pitch servo, and yaw servo, considering the consequent influence of shaft rotation and tower movement. However, the PID regards these two objectives as minor corrections inside the PID loop. Therefore, the model-based approach is more suitable for the multi-objective regulation of the turbine system, especially for intrinsic nonlinearities caused by aerodynamics.

8.3 Linear Quadratic Regulator

The LQR computes a state feedback gain considering a quadratic cost function of the system state and control input [161]. The LQR is a gain-optimized controller that has the following advantages. Firstly, given the quadratic cost minimization, the LQR can consider future responses from present to infinite and thus optimize its control policy by updating the optimal gain. Secondly, the LQR requires less computational resources and is friendly to low-memory devices regarding hardware implementation. The LQR only updates its feedback gain when the power strategy decides to move to a new state. In a dynamic transition, the LQR realizes regulation by a matrix production of the gain and measured state. Besides, the LQR design offers a flexible input regulation that sets input variables with different priorities.

A natural objective of regulation is system stabilization so that the state converges quickly to zero without spending too many control efforts [162]. As the system feedback of the proposed model-based turbine control is full-state, i.e., the output vector is the same as the state vector, the LQR uses a quadratic cost function of **Eq. (191)** to balance the aggressive regulation with the cost of control [163].

$$\mathcal{J} = \min\left(\int_0^\infty (\Delta x^T Q \Delta x + \Delta u^T R \Delta u)\, dt\right) \tag{191}$$

where

Q: the weighting matrix for the state cost (tuneable)

R: the weighting matrix for the actuation cost (tuneable)

According to the stability requirements, Q and R are positive semi-definite and positive definite, respectively, and are often diagonal [35]. The diagonal elements assign different priorities to state and input variables, which leads to adjustable system behaviours. Typically, the relative ratios of elements in the weighting matrices follow a rule of powers of ten [162].

The LQR gain follows the control law of **Eq. (192)** and satisfies the condition of **Eq. (193)** when optimal. A common solution for this gain is to form the extended Hamiltonian matrix pencil [164] and use an appropriate set of the Schur vectors [165].

$$\Delta u = K_{LQR}\Delta x \tag{192}$$

$$K_{LQR} = -R^{-1}B^{T}P \tag{193}$$

$$A^{T}P + PA - PBR^{-1}B^{T}P + Q = 0 \tag{194}$$

where

K: the optimal feedback gain

P: the solution to the algebraic Riccati equation (**Eq. (194)**)

8.4 Receding Horizon Control

The terminology 'RHC' refers to a recursive solution of the MPC [97], and this study uses this terminology to distinguish it from the MPC based on quadratic programming (QP) [166] in **section 8.5**. The RHC or MPC predicts a sequence of future system responses and minimizes the objective function of a control sequence [97]. Considering the full-state feedback in a finite linear-quadratic RHC or MPC, an objective function (**Eq. (195)**) for the prediction of N steps evaluates the state deviation and the input cost [166].

$$\mathcal{J} = \min\left(\sum_{k=0}^{N}\left(x_k - x_{obj}\right)^{T} Q\left(x_k - x_{obj}\right) + \sum_{k=0}^{N-1} u_k^{T} R u_k\right) \tag{195}$$

$$s.t. \begin{cases} x_{k+1} = A_d x_k + B_d u_k \\ x_{min} \le x_k \le x_{max} \\ u_{min} \le u_k \le u_{max} \\ x_0 = \bar{x} \end{cases} \tag{196}$$

$$A_d = e^{At_s} \tag{197}$$

$$B_d = \int_0^{t_s} e^{A(t_s - t)} dt\, B \tag{198}$$

where

x_{obj}: the regulation objective

x_k: the discrete state at the k time step

u_k: the discrete input at the k time step

A_d: the discrete state matrix

B_d: the discrete input matrix

$\bar{x}$: the sampled state

The number of prediction steps determines the span of the control horizon and thus affects final control results. Fewer steps cannot fully use trajectory planning, but excessive steps are meaningless for current control consequences and lead to a large scale of the target problem. This study suggests 10~20 steps for such discrete controllers, which can balance control performance and computation speed. This study adopts 20 steps in both RHC and MPC settings.

The RHC applies the recursion formula (**Eq. (199)**) to update the augmented states with a control sequence [40]. Thus, its control sequence corresponds to the local optimum for the recurred states. Meanwhile, the constructed matrices ($\bar{A}$, $\bar{B}$) are not sparse, so the RHC consumes plenty of memory to store the model of a long trajectory.

$$\begin{bmatrix} x_{k+1|k} \\ x_{k+2|k} \\ x_{k+3|k} \\ \vdots \\ x_{k+N|k} \end{bmatrix} = \bar{A} x_k + \bar{B} \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+N-1} \end{bmatrix} \tag{199}$$

$$\bar{A} = \begin{bmatrix} A_d \\ A_d^2 \\ A_d^3 \\ \vdots \\ A_d^N \end{bmatrix},\ \bar{B} = \begin{bmatrix} B_d & 0 & 0 & \cdots & 0 \\ A_d B_d & B_d & 0 & \cdots & 0 \\ A_d^2 B_d & A_d B_d & B_d & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_d^{N-1} B_d & A_d^{N-2} B_d & \cdots & A_d B_d & B_d \end{bmatrix} \tag{200}$$

where

$\bar{A}$: the recursive state matrix

$\bar{B}$: the recursive input matrix

The recursive optimization also needs three auxiliary matrices (**Eqs. (201)~(203)**) to establish the objective of **Eq. (199)**, which yields an equivalent form of **Eq. (195)** [167]. Also, these additional matrices can weigh the trajectory on the prediction horizon.

$$\bar{Q} = \begin{bmatrix} \underbrace{\begin{matrix} Q & 0 & 0 & 0 \\ 0 & Q & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q \end{matrix}}_{N} \end{bmatrix} \tag{201}$$

$$\bar{R} = \begin{bmatrix} \underbrace{\begin{matrix} R & 0 & 0 & 0 \\ 0 & R & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{matrix}}_{N} \end{bmatrix} \tag{202}$$

$$\bar{I} = \begin{bmatrix} \underbrace{\begin{matrix} I_{n_u} & 0 & 0 & \cdots & 0 \\ -I_{n_u} & I_{n_u} & 0 & \cdots & 0 \\ 0 & -I_{n_u} & I_{n_u} & \cdots & 0 \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & -I_{n_u} & I_{n_u} \end{matrix}}_{N} \end{bmatrix} \tag{203}$$

Since the strict optimum of **Eq. (195)** should meet the first-order necessary condition and the second-order sufficiency condition [168], the RHC gain (**Eq. (204)**) ensures optimal control sequence for input small signal [167]. Only the first answer takes effect at each discrete time step under a greedy control policy, yielding the final input as **Eq. (205)** [97].

$$K_{RHC} = (\bar{B}^T\bar{Q}\bar{B} + \bar{I}^T\bar{R}\bar{I})^{-1}\bar{B}^T\bar{Q}\bar{A} \tag{204}$$

$$\Delta u_k = \begin{bmatrix} \underbrace{\begin{matrix} I_{n_u} & 0 & \cdots & 0 \end{matrix}}_{N} \end{bmatrix} K_{RHC}\Delta x_k \tag{205}$$

8.5 Model Predictive Control

Another MPC implementation supposes the MPC objective of **Eq. (195)** as a QP problem that general QP solvers can process [169]. The QP-based approach first establishes an equivalent target with trajectory prediction. Subsequently, a QP solver finds an optimal control sequence that meets state and input constraints. Compared with the RHC, the QP-based MPC can achieve global optimization with the constraints because the QP solver iterates over the entire sequence of state and input. The terminology 'QP' is a set of problems numerically solved using active-set, interior-point, or augmented-Lagrangian methods [170]. **Eqs. (206)** gives the standard form of a QP problem involving costs, subject to a set of constraints (**Eq. (207)**) [169].

$$\mathcal{J} = \min\left(\frac{1}{2}x^T\mathcal{P}x + q^T x\right) \tag{206}$$

$$s.t. \begin{cases} \mathcal{G}x \le h \\ \mathcal{A}x = b \\ \ell \le x \le u \end{cases} \tag{207}$$

where

$\mathcal{P}$: the symmetric cost

q: the linear cost

$\mathcal{G}$, h: the inequality constraints

$\mathcal{A}$, b: the equality constraints

ℓ, u: the variable constraints

Since the MPC optimization is not a straightforward form of the QP optimization, a state-constrained method [118] updates **Eq. (195)** to a sparse form of **Eq. (206)**. The transformation from control problem to QP optimization follows **Eqs. (208)~(216)** [171].

$$x = [x_0 \quad \cdots \quad x_N \quad u_0 \quad \cdots \quad u_{N-1}]^T \tag{208}$$

$$\mathcal{P} = \begin{bmatrix} \overbrace{\begin{matrix} Q & \cdots & 0_{n_x} \\ \vdots & \ddots & \vdots \\ 0_{n_x} & \cdots & Q \end{matrix}}^{N+1} & \cdots & 0_{n_x \times n_u} \\ \vdots & \ddots & \vdots \\ 0_{n_u \times n_x} & \cdots & \underbrace{\begin{matrix} R & \cdots & 0_{n_u} \\ \vdots & \ddots & \vdots \\ 0_{n_u} & \cdots & R \end{matrix}}_{N} \end{bmatrix} \tag{209}$$

$$q = \begin{bmatrix} \overbrace{-Qx_{ref} \quad \cdots}^{N+1} & \underbrace{0_{n_u} \quad \cdots}_{N} \end{bmatrix}^T \tag{210}$$

$$\mathcal{G} = 0 \tag{211}$$

$$h = \begin{bmatrix} \overbrace{\infty_{n_x} \quad \cdots}^{N+1} & \underbrace{\infty_{n_u} \quad \cdots}_{N} \end{bmatrix}^T \tag{212}$$

$$\mathcal{A} = \begin{bmatrix} \underbrace{\begin{matrix} -I_{n_x} & 0_{n_x} & \cdots & 0_{n_x} \\ A_d & -I_{n_x} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{n_x} \\ 0_{n_x} & \cdots & A_d & -I_{n_x} \end{matrix}}_{(N+1)\times(N+1)} & \underbrace{\begin{matrix} 0_{n_u} & 0_{n_u} & \cdots & 0_{n_u} \\ B_d & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{n_u} \\ 0_{n_u} & \cdots & B_d & 0_{n_u} \end{matrix}}_{(N+1)\times N} \end{bmatrix} \tag{213}$$

$$\mathcal{b} = \begin{bmatrix} -x_0 & \underbrace{0_{n_x} \quad \cdots}_{N} \end{bmatrix}^T \tag{214}$$

$$\mathcal{l} = \begin{bmatrix} \overbrace{x_{min} \quad \cdots}^{N+1} & \underbrace{u_{min} \quad \cdots}_{N} \end{bmatrix}^T \tag{215}$$

$$\mathcal{u} = \begin{bmatrix} \overbrace{x_{max} \quad \cdots}^{N+1} & \underbrace{u_{max} \quad \cdots}_{N} \end{bmatrix}^T \tag{216}$$

where

n_x, n_u: the number of state variables or input variables

x_{min}, x_{max}: the state constraints

u_{min}, u_{max}: the input constraints

The inequality constraint ($\mathcal{G}x \leq \mathcal{h}$) is out of use, and **Eqs. (211)(212)** deliver obligatory parameters for a QP solver. The cost matrix $\mathcal{P}$ must be semi-positive so that the QP problem is not non-deterministic polynomial-time (NP) hard. The QP solver uses the equality constraint ($\mathcal{A}x = \mathcal{b}$) to formulate the system dynamics on the finite horizon, which accounts for global trajectory planning.

The MPC objective can be written as an incremental expression given a linearized system at a reference point. An essential purpose of the linearized MPC is to eliminate the state deviation, i.e., $\Delta x_{obj} = 0$, which yields **Eqs. (217)(218)**.

$$\mathcal{J} = \min\left(\sum_{k=0}^{N} \Delta x_k^T Q \Delta x_k + \sum_{k=0}^{N-1} \Delta u_k^T R \Delta u_k\right) \tag{217}$$

$$s.t.\begin{cases} \Delta x_{k+1} = A_d \Delta x_k + B_d \Delta u_k \\ x_{min} \leq x_{ref} + \Delta x_k \leq x_{max} \\ u_{min} \leq u_{ref} + \Delta u_k \leq u_{max} \\ \Delta x_0 = \Delta \bar{x} \end{cases} \tag{218}$$

where

x_{ref}: the reference of the system state

u_{ref}: the reference of the input state

Eqs. (217)(218) essentially establish the small-signal MPC, which is now compatible with the model-based framework. For this conversion, the small signal deviation of **Eqs. (219)(220)(221)(222)(223)** replaces the original form of **Eqs. (208)(210)(214)(215)(216)**.

$$\mathscr{x} = [\Delta x_0 \quad \cdots \quad \Delta x_N \quad \Delta u_0 \quad \cdots \quad \Delta u_{N-1}]^T \tag{219}$$

$$\mathscr{q} = \left[\overbrace{0_{n_x} \quad \cdots}^{N+1} \quad \underbrace{0_{n_u} \quad \cdots}_{N}\right]^T \tag{220}$$

$$\mathscr{b} = \left[-\Delta x_0 \quad \underbrace{0_{n_x} \quad \cdots}_{N}\right]^T \tag{221}$$

$$\mathscr{l} = \left[\overbrace{x_{min} - x_{ref} \quad \cdots}^{N+1} \quad \underbrace{u_{min} - u_{ref} \quad \cdots}_{N}\right]^T \tag{222}$$

$$\mathscr{u} = \left[\overbrace{x_{max} - x_{ref} \quad \cdots}^{N+1} \quad \underbrace{u_{max} - u_{ref} \quad \cdots}_{N}\right]^T \tag{223}$$

The combination of **Eqs. (219)(209)(220)(211)(212)(213)(221)(222)(223)** establish an equivalent QP problem of **Eq. (195)**, in which the MPC samples the system state and updates **Eq. (221)** to track the OPS command. **Eqs. (222)(223)** are only renewed when the reference point shifts. For each time step, the MPC only returns the first result Δu_0 as the control input.

Table 8.1 lists compatible QP solvers with the algorithm, matrix API, and warm-start support. The warm-start means a solver accepts an external condition as the initial solution, which can accelerate computation in small perturbation cases. If applicable, the MPC receives the last solution as the current warm start.

Table 8.1 Compatible solvers for the QP-based MPC design

Solver	Algorithm	Matrix API	Warm-start
QuadProg [172]	active set	dense	no
CVXOPT [173]	interior point	dense	yes
SCS [174]	augmented Lagrangian	sparse	yes
Clarabel [175]	interior point	sparse	no
OSQP [171]	augmented Lagrangian	sparse	yes
ProxQP [169]	augmented Lagrangian	sparse	yes

The quadratic programming solver (QuadProg) is a dual algorithm taking unconstrained minimum as the initial point, which is efficient when a primal feasible point is hard to find [172]. The convex optimization solver (CVXOPT) is an interior-point method for cone programming to optimize the non-sparse structure of linear equations [173]. The splitting conic solver (SCS) based on the alternating direction method of multipliers (ADMM) is a first-order QP algorithm towards large-size problems [174]. The Clarabel is an interior point numerical solver that applies a novel homogeneous embedding for convex optimization problems [175]. The operator splitting solver for quadratic programs (OSQP) is an ADMM-based method that requires only a single matrix factorization and implements custom sparse linear algebra routines, efficiently solving parametrized optimizations [171]. The proximal quadratic programming solver (ProxQP) is a variation of the augmented Lagrangian method enhanced by a preconditioning strategy, an initialization for primal and dual variables, and a cold restart strategy [169].

Table 8.2 Solver benchmark regarding solving speed

Solver	Solving Count	Total (s)	Min. (ms)	Max. (ms)	Mean (ms)
QuadProg	30000	653.00	19.743	25.506	21.767
CVXOPT	30000	477.84	7.500	26.581	15.928
SCS	29989	30.39	0.053	3.004	1.013
Clarabel	29904	24.88	0.280	4.007	0.832
OSQP	29819	18.41	0.303	2.000	0.617
ProxQP	24991	12.72	0.225	1.504	0.509

As **Table 8.1** introduces six QP solvers, it is necessary to investigate their performance on the problem of the 3-DOF turbine regulation. **Table 8.2** is a benchmark where six QP solvers address 30,000 samples. The benchmark runs on an Intel Core i9 14900k. The computation of the QuadProg and CVXOPT is too slow compared to the other sparse methods. **Eqs. (209)(213)** indicates a native sparse QP programming, so we recommend sparse methods to achieve high-performance computing. The SCS, Clarabel, OSQP, and ProxQP can ensure a sampling rate of 1 ms, which also meets the minimum requirement of governing a generator. The ProxQP exhibits outstanding efficiency with the lowest solving count and average time. Hence, the following simulation selects the ProxQP as the MPC solver.

8.6 Discussion

This study proposes four controllers, and **Table 8.3** compares their fundamental characteristics. The PID is the only model-free design with no trajectory prediction. The LQR, RHC, and MPC achieve the weighted multi-objective regulation of shaft rotation, tower motion, generator response, pitch position, and yaw orientation.

Table 8.3 Comparison of four turbine controllers

Controller	Type	Characteristics
PID	model-free, continuous	• its gain scheduling gets configuration from the power strategy; • the main objective is rotor speed, and the other objectives are minor corrections; • it has the lowest requirement for computation.
LQR	model-based, continuous	• the feedback gain gets updated when the power strategy makes a new decision; • the trajectory horizon is from present to infinite; • it consumes less memory and computational resources since the control system only stores the gain matrix.
RHC	model-based, discrete	• its prediction horizon has limited steps; • its gain matrix is greater than the LQR in most cases.
MPC	model-based, discrete	• it has the highest requirements for memory and computation due to the QP optimization; • its control result is more accurate for nonlinear systems due to the dynamic solution of the QP solver; • it natively considers the problem constraints for each solution.

8.7 Generator Buffer

Since wind variation and operation adjustment result in frequent power fluctuation, two Savizky-Golay filters [176] process the signal of the dq-axis voltages outputted by the controller to reduce oscillation as an output buffer of the generator, as shown in **Figure 8.3**. Another consideration of the output buffer is that the generator response is faster, which allows waiting for mechanical settlements. This study chooses the Savizky-Golay because it is a moving-average filter with zero phase and does not delay any frequencies.

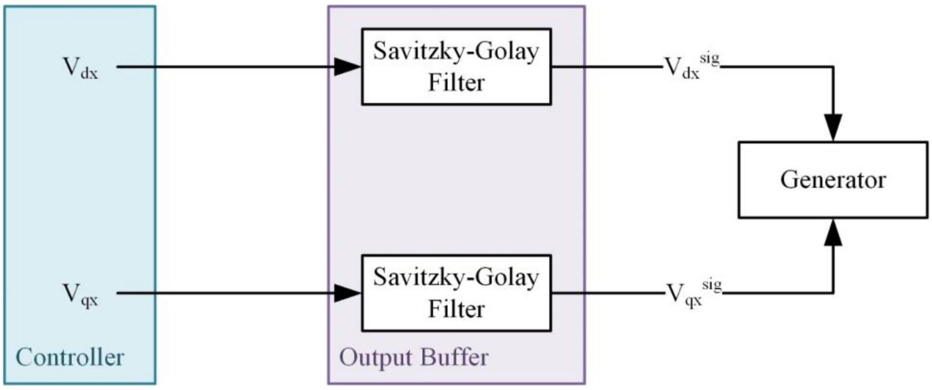

Figure 8.3 Output buffer based on the Savizky-Golay filter

The Savizky-Golay filter is a kind of least-squares polynomial smoothing (**Eq. (224)**) that minimizes the error function for the group of $2M + 1$ input samples centred on $n = 0$ [176].

$$\mathcal{E}_N = \sum_{n=-M}^{M} (p(n) - x[n])^2 = \sum_{n=-M}^{M} \left(\sum_{k=0}^{N} \alpha_k n^k - x[n] \right)^2 \tag{224}$$

where

$\mathcal{E}_N$: the mean-squared approximation error

$x[n]$: the n-th input sample

α_k: the k-th polynomial coefficient

The Savizky-Golay window M determines the length of samples, and the Savizky-Golay order N affects the degree of smoothness. Increasing the window and order can improve filtering performance. However, a long window can lead to destructive signal delay, and a high order significantly increases computational costs.

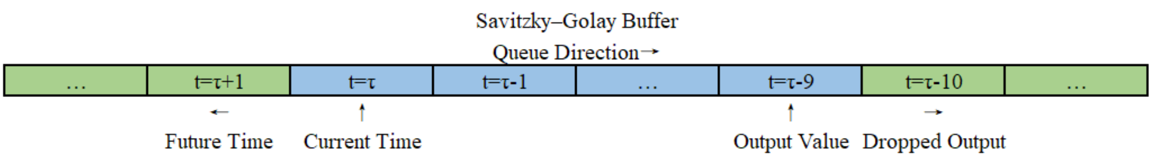

Figure 8.4 Movement of the Savizky-Golay buffer

Figure 8.4 explains the movement of the Savizky-Golay filter on a voltage sequence with a buffer size of 10. The filtered output is a smoothed value after 9 buffer samples, so this moving buffer lags the voltage signal. More buffer samples improve smoothness, but a long buffer can ruin stability due to signal lag. Therefore, the filter tuning should consider the kinetic and electric characteristics of the generator response to make the trade-off between output smoothness and control loss.

Chapter 9 Wind Forecasting

A wind forecasting module processes the sequences of wind velocity and direction measured by the anemometer and vane. Meanwhile, it infers the most likely wind for the power strategy to update the control configuration. **Figure 9.1** explains the procedures of applying wind forecasting to update the power strategy for geared and direct-drive systems. The forecasting module takes wind measurements from an anemometer and a vane mounted on the nacelle. Subsequently, the vector series converted from the measurements is passed to a wind vector series model to predict future wind vectors. The inverse vector transformation restores velocity and direction predictions. The power strategy will use the restored wind prediction to determine the working point (rotation, pitch, yaw, and power capture). Meanwhile, the control module updates the regulation objective and optimal control parameters at the working point. After that, the controller receives the latest configuration in the prediction period and executes the 3-DOF turbine regulation.

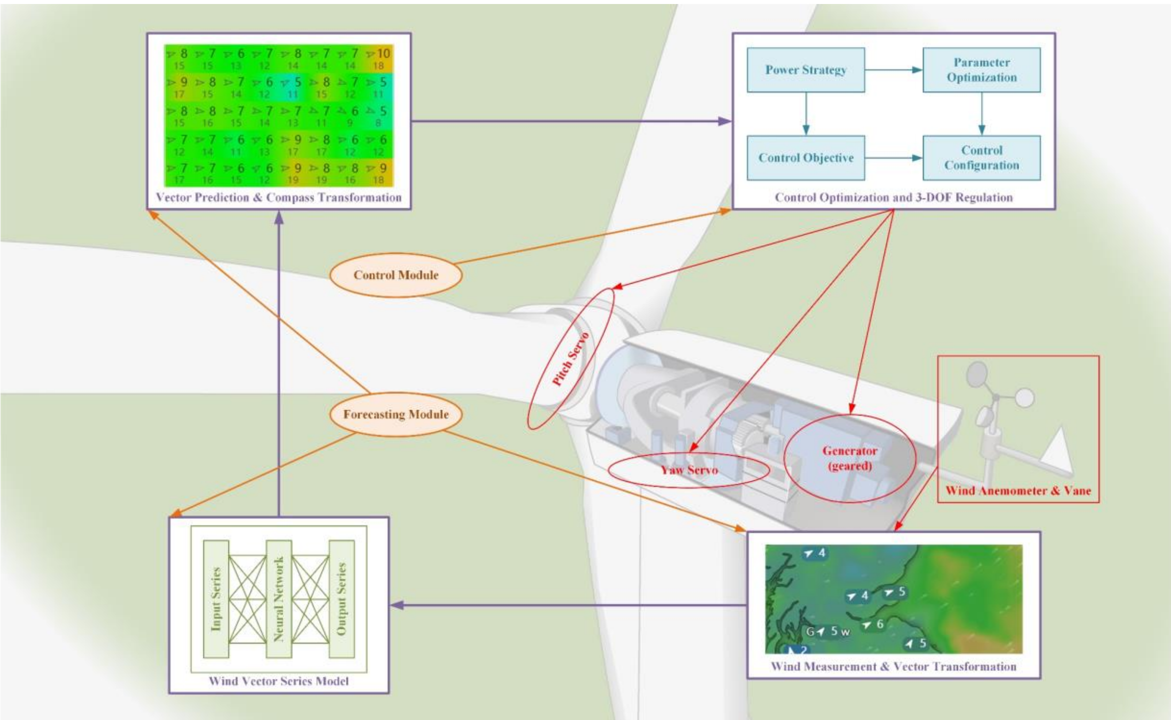

Figure 9.1a Deployment on the geared turbine

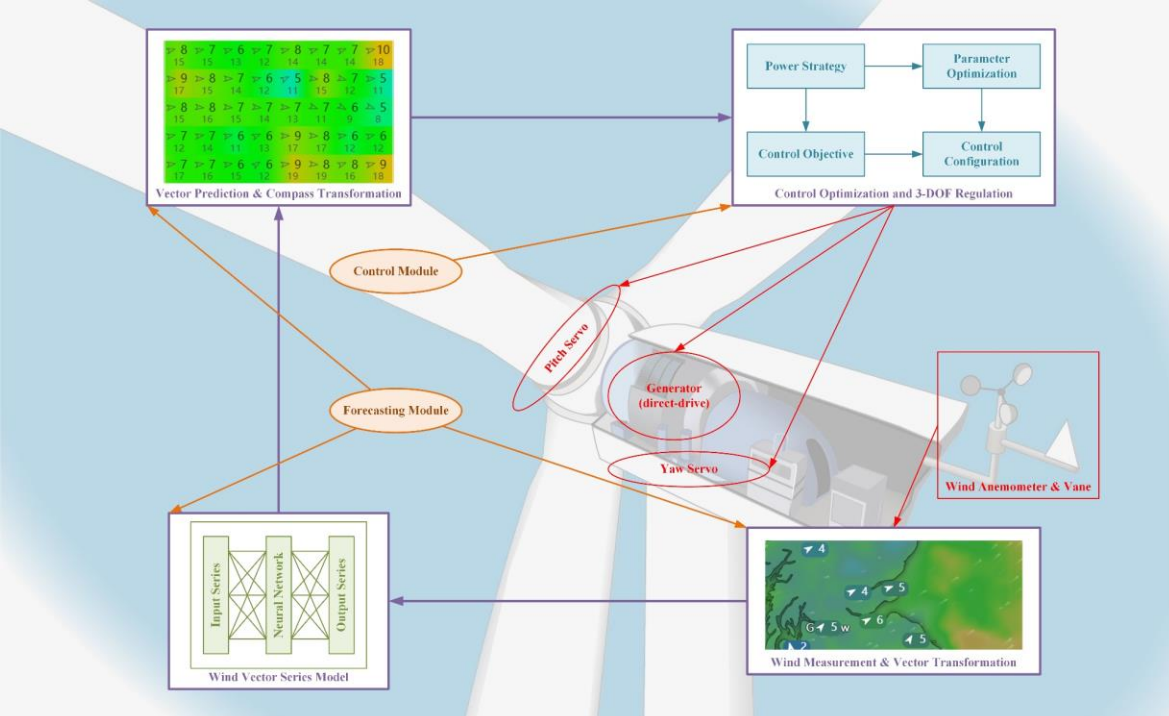

Figure 9.1b Deployment on the direct-drive turbine

Figure 9.1 Flowchart from wind forecasting to control optimization

Wind forecasting uses several prior wind samples to predict future wind trends as time series forecasting. An accurate forecasting model ensures optimal control configuration for higher system stability and power quality. This section will build four DL networks to predict wind vector series through data preprocessing and compass-vector transformation, as shown in **Figure 9.2**.

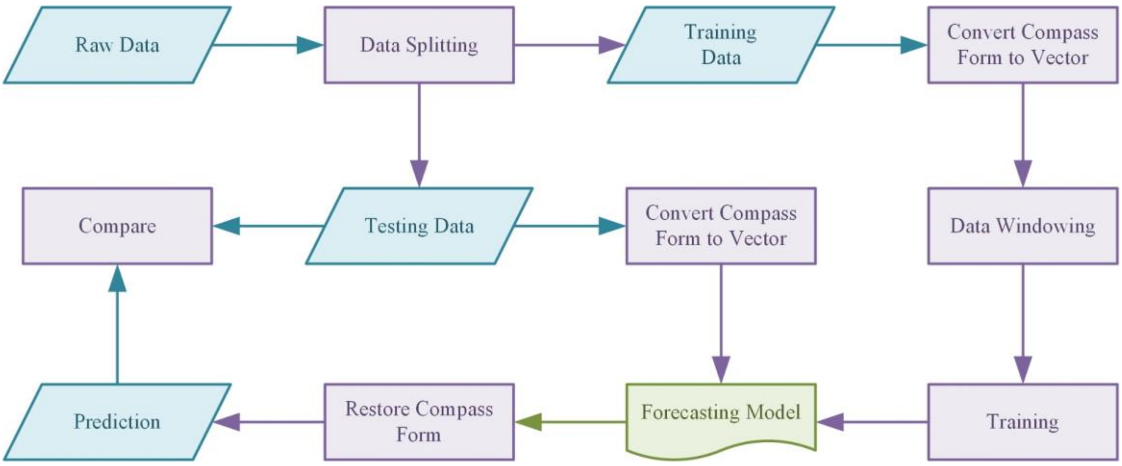

Figure 9.2 Procedures for training and testing a wind series model

9.1 Data Description

Since multiple variables might affect wind prediction, it is necessary to perform a feature selection to select proper features for time series modelling [177]. Considering the employed dataset gathered from multi-meteorological sensors has 13 variables, the spatial correlation [73] between these features evaluates their dependencies. A negative correlation means an opposite variation between two series candidates, but its magnitude indicates the same correlation intensity as a positive. Thus, feature selection only needs to consider absolute correlations.

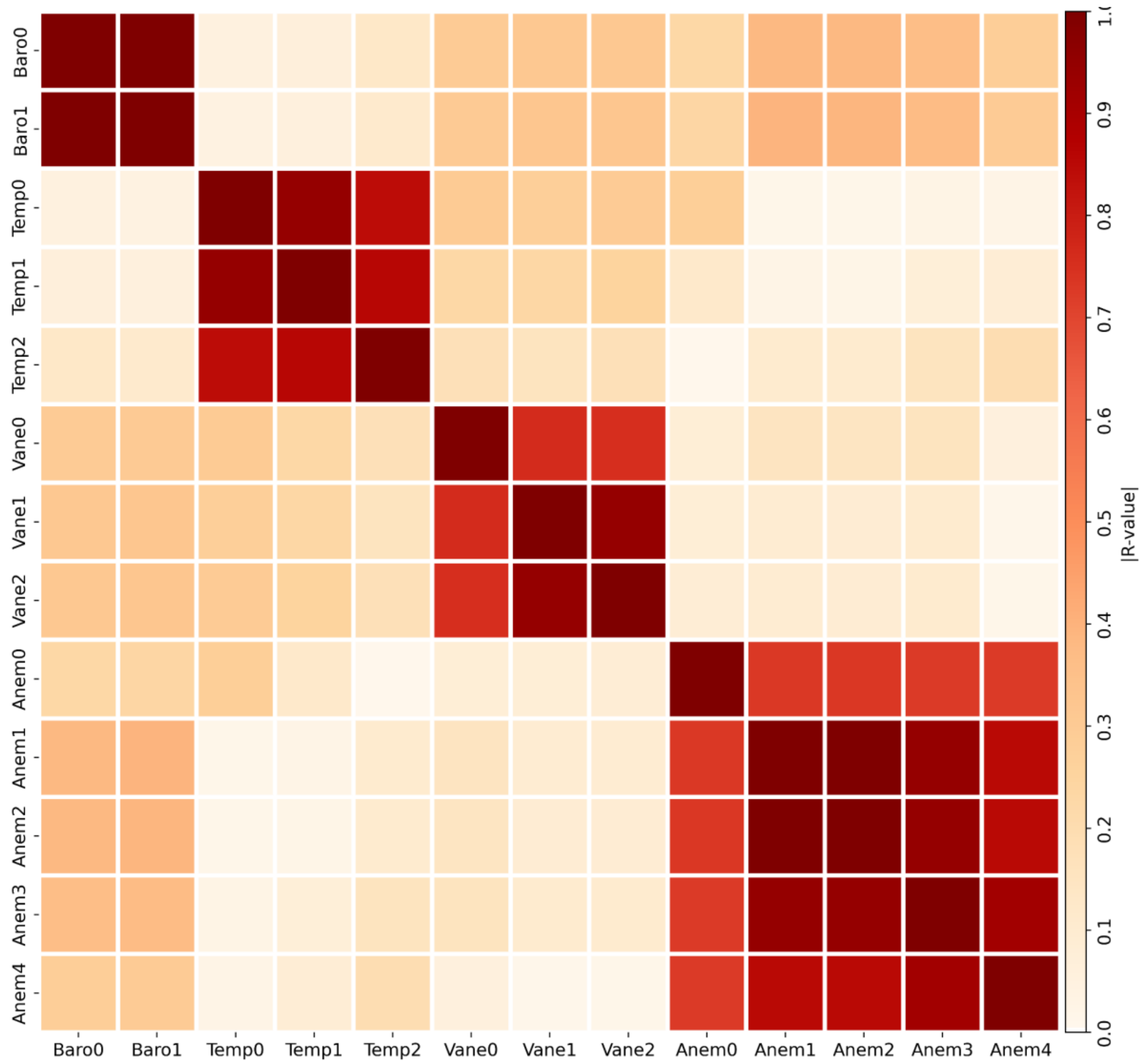

Figure 9.3 Heatmap of the spatial correlation between sampled features (sensors: barometer, thermometer, wind vane, and wind anemometer)

Figure 9.3 calculates the absolute correlation for feature analysis, in which a value close to the unit shows a higher correlation. Air pressure (barometer measurement) correlates lightly with wind velocity and direction, and temperature potentially affects direction, but both correlations are less than 0.5. Only wind measurements at different heights have strong correlations. Hence, the time series modelling ignores air pressure and temperature and only considers wind velocity and direction at the hub height for simplicity.

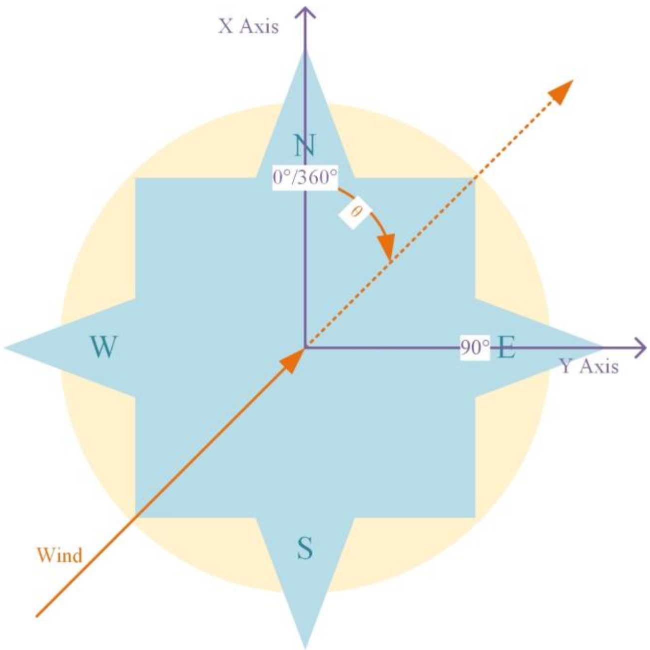

Figure 9.4 Compass wind of reporting velocity and direction (the north refers to 0° / 360°)

Figure 9.4 defines the representation of wind measurement in the compass direction, which is clockwise and represents the direction blowing from [74]. Angles are not explicit model input due to the unsuccessive alternation across 0° and 360°. A vector decomposition transforms wind velocity and direction for better learning, as **Eqs. (225)(226)**, as well as its inverse transformation as **Eqs. (227)(228)**.

$$v_x = v' \cos\theta' \tag{225}$$

$$v_y = v' \sin\theta' \tag{226}$$

$$v' = \sqrt{v_x^2 + v_y^2} \tag{227}$$

$$\theta' = \arctan\frac{v_x}{v_y} \tag{228}$$

where

v_x, v_y: the xy-axis components of a wind vector (m/s)

Figure 9.5 compares the distributions of compass wind and decomposed vector. It clearly shows wind vectors are much simpler for a model to learn. Therefore, wind forecasting converts compass wind to wind vector and later recovers compass wind after forecasting. Meanwhile, a forecasting model no longer needs to weigh input features because an orthogonal pair have equal importance.

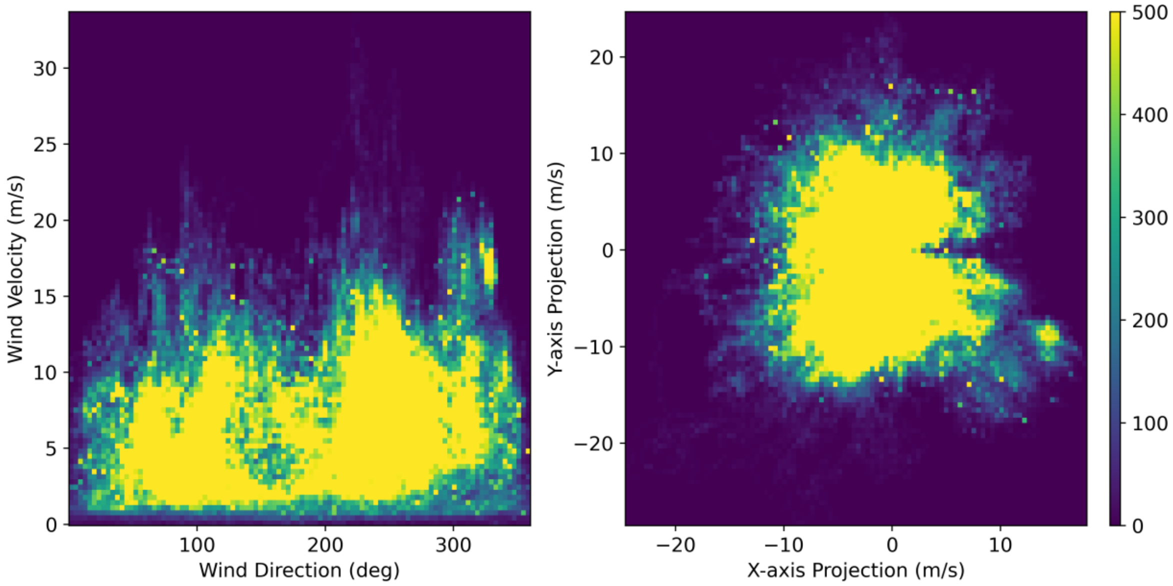

Figure 9.5 Wind distribution and vector decomposition (uneven relation between velocity and direction vs. uniform relation)

The training set has 2,281,610 samples sampled at a frequency of 0.1-Hz. Two reasons account for selecting this frequency. First, faster sampling does not improve accuracy because natural non-compressed airflow cannot alter suddenly during such a period. Second, this considers the recommended forecasting period in **Table 4.1**. **Table 9.1** summarizes the statistics of the training set about compass and vector. Velocity concentrates on 4.28~9.62 m/s, and direction focuses on about 90° and 250° affected by seasonal winds. Model testing is on a successive 30-day set of 259,200 samples.

Table 9.1 Data description of the 0.1-Hz wind training set

	Mean	Std.	Min.	25%	50%	75%	Max.
v' (m/s)	7.30	3.97	-0.10	4.28	6.89	9.62	33.71
θ' (°)	192.97	87.40	0.01	110.36	218.62	259.84	359.99
v_x (m/s)	-1.19	4.80	-24.64	-3.95	-1.24	1.23	17.98
v_y (m/s)	-1.64	6.47	-28.53	-6.66	-2.16	3.19	24.68

9.2 Data Preprocessing

For passing correct information into a model, the min-max scaler (**Eqs. (135)(136)**) performs input normalization and prediction restoration. $\tilde{x} = [v_x \quad v_y]^T$ is an orthogonal vector, $\tilde{x}_{min}$ and $\tilde{x}_{max}$ can be found in **Table 9.1**. After preprocessing, data needs to be windowed so that an ML platform can perform training on inputs and predictions to make a time series model. Meanwhile, a windowing definition determines the time axis of successive predictions. **Figure 9.6** visualizes the operation of windowing on training and testing sets. A complete window consists of input and shift subwindows, in which a shift subwindow includes an offset and a label. The green offset in **Figure 9.6** is necessary for practical purposes, such as reserving time for model calculation. It notes that the proposed windowing has different movements on training and testing sets. Each window from the training set is independent and has no overlap with adjacent windows. However, the testing set moves the window according to the label width to make successive predictions. In this study, an input subwindow of 6 samples and a label subwindow of 3 predictions constitute a window with zero shift.

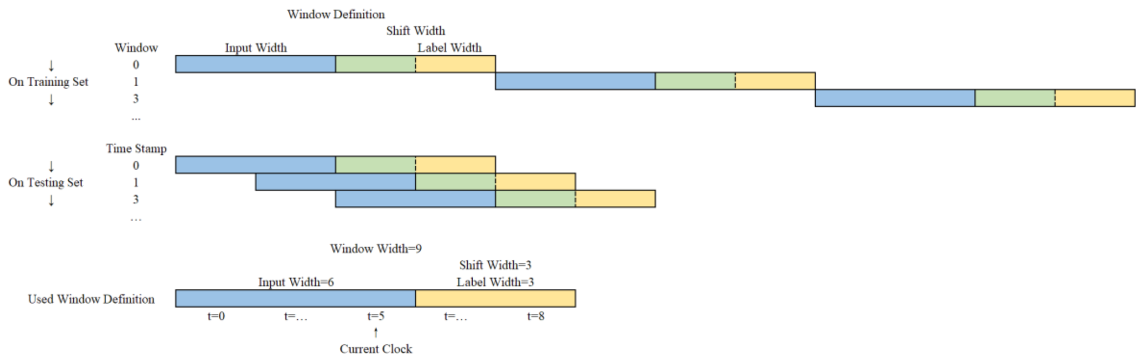

Figure 9.6 Window definition for sampling and forecasting

9.3 Deep Learning Models

DL models rely on complex and multiple network architectures that exploit nonlinear mapping capabilities [178]. Convolutional and recurrent structures are two of the most popular DL methods in time series forecasting [81]. This section will introduce CNN and LSTM-based models to predict 2-D wind vector series. Meanwhile, this wind modelling includes an MLP-based DNN and a hybrid CNN-LSTM for comparison. Under the window definition, four models have the same input shape of 6×2 and the output shape of 3×2, corresponding to a 60-s input series and a 30-s prediction series. In other words, a wind forecasting model predicts half a minute for control optimization. Wind forecasting also adopts the Adam optimizer and the MSE loss function, which are the same as aerodynamic modelling. In addition, the batch size is 4,096, and the maximum epoch is 400.

9.3.1 Deep Neural Network

The DNN, another name for deep MLPN, is a classic type that stacks regular layers in TensorFlow [179]. The developed DNN has a seven-layer architecture, as illustrated in **Figure 9.7**. The hidden part between input and label layers undertakes two tasks: one is for learning the underlying information of time series, and the other is for shaping so that a prediction has the expected output shape, which yields a learning-shaping mechanism.

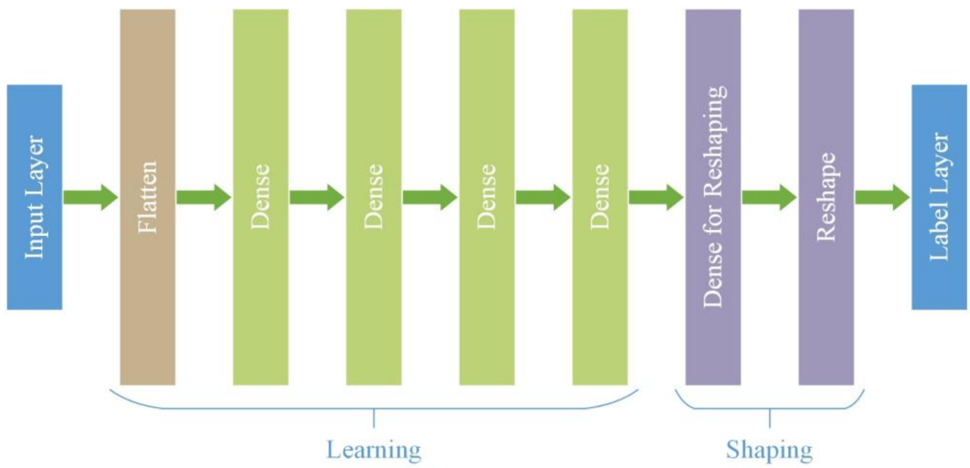

Figure 9.7 Structure of the DNN with stacked dense layers

Since the native dense layer only works for 1-D vectors, it requires flattening the 2-D input layer to pass data flow correctly. All dense layers on the learning side take the ReLU [80] as their activation function to introduce nonlinearity between connected layers. In the shaping structure, the dense layer whose neurons equal the label size applies a linear operation to output the result directly, and the reshaping layer transforms the hidden vector into the label shape. It notes that the shaping structure is reused in the following networks to achieve 2-D prediction.

Table 9.2 gives the specific layer configuration of the DNN in TensorFlow. The hyperparameter tuning of the DNN neurons only adjusts four dense layers on the learning side. Their neurons can be equally increased or decreased with a step of 16 units until the training loss converges smoothly. The tuning of the other models also follows the same principle. The configuration of **Table 9.2** can ensure steady training through multiple trials. It is worth mentioning that a robust network should not be sensitive to hyperparameters. It should consider adding a learning layer if similar hyperparameters lead to enormous prediction discrepancies.

Table 9.2 Configuration of the DNN layers

Layer	Arguments	Value	Parameters
Flatten	-	-	-
Dense	units	32	416
	activation	ReLU	
Dense	units	32	1,056
	activation	ReLU	
Dense	units	32	1,056
	activation	ReLU	
Dense	units	32	1,056
	activation	ReLU	
Dense	units	6	198
Reshape	-	(3, 2)	-

9.3.2 Convolutional Neural Network

The CNN has become a primary DL method in renewable energy forecasting [81]. It relies on feature extraction to interpret the topological structure of sophisticated data, which has achieved success in image processing [81]. Compared with 2-D images, the only difference in applying the CNN in wind series forecasting is that a convolution kernel convolves the input layer on a single spatial (or temporal) dimension [78]. **Figure 9.8** shows the network structure of the 1-D CNN. The 1-D convolution layer moves the kernel window along with the time axis of the input series. Since the Conv1D layer has parsed underlying time information, the unfolded Conv1D output cascades a dense layer for learning.

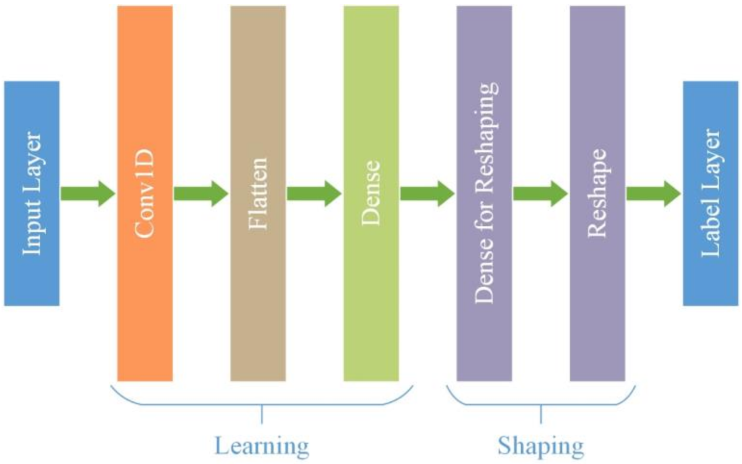

Figure 9.8 Structure of the 1-D CNN

Table 9.3 details the optimized structure of **Figure 9.8**. The convolution and flattening layers perceive time series features and output a character vector so that a dense layer can understand temporal characteristics. The feature extraction of a convolutional-flattened structure results in plenty of parameters in the dense layer on the learning side.

Table 9.3 Configuration of the CNN layers

Layer	Arguments	Value	Parameters
Conv1D	filters	32	224
	kernel size	3	
	activation	ReLU	
	padding	same	
Flatten	-	-	-
Dense	units	32	6176
	activation	ReLU	
Dense	units	6	198
Reshape	-	(3, 2)	-

9.3.3 Long Short Term Memory

The LSTM is one of the most popular extensions of the RNN that predicts various time series [81]. The fundamental mechanism of a recurrent structure is a sequential model that can arrange time-series data as input vectors and supply output results through its internal cells [82]. Therefore, the input series goes through cells in a sequential vector, and the output label concatenates with the next time series at each movement [79]. The LSTM uses memory cells (**Figure 9.9**) to replace conventional RNN cells, thus avoiding the gradient explosion and disappearance of the RNN.

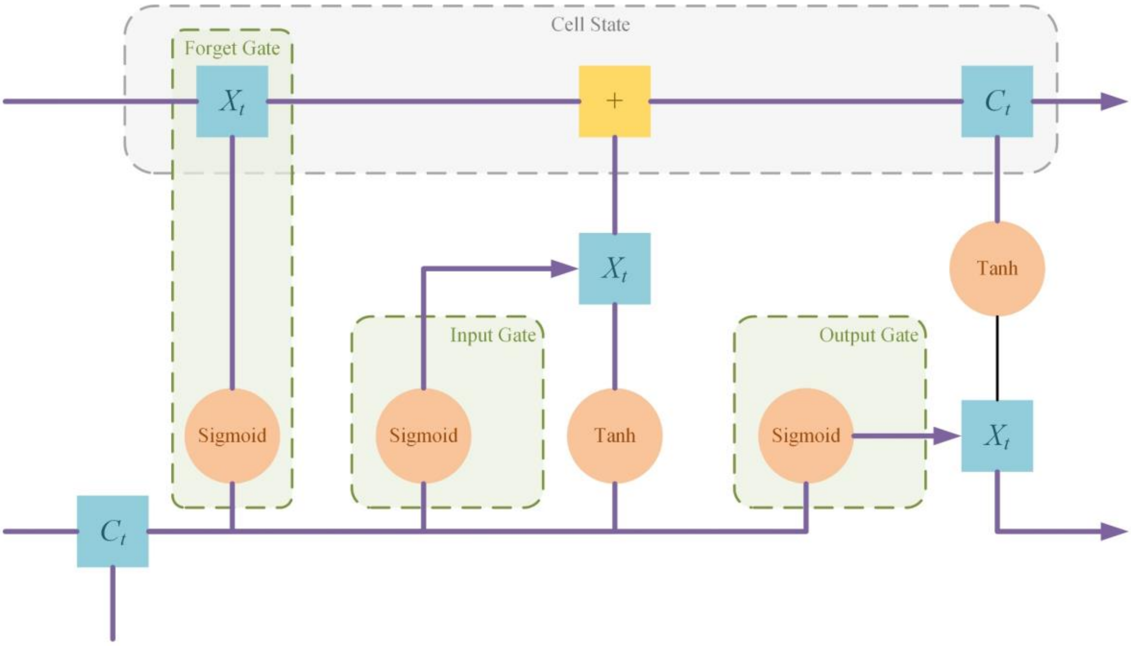

Figure 9.9 Structure of the LSTM memory cell

The information flow of the gates and cell state in **Figure 9.9** implies the calculation procedure of the LSTM at each step, as **Eqs. (229)~(234)** [78]. The input gate controls the information received by a memory cell, the output gate controls the forecasted information of the memory cell, and the forget gate determines the information to be removed [74]. The memory cell is responsible for recording cell states [74].

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{229}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{230}$$

$$C_t' = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{231}$$

$$C_t = f_t * C_{t-1} + i_t * C_t' \tag{232}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{233}$$

$$h_t = o_t * \tanh(C_t) \tag{234}$$

where

f_t: the forget gate

i_t: the input gate

C_t: the cell state

o_t: the output gate

W, b: the weight and bias of a gate/cell

The LSTM optimization principally adjusts the number of memory cells to ensure sufficient and economical recurrent units. In this wind forecasting problem, we do not recommend adding more than one LSTM layer because a stack of LSTMs results in training difficulties. The LSTM with the shaping structure adopts the settings of **Table 9.4**.

Table 9.4 Configuration of the LSTM layers

Layer	Arguments	Value	Parameters
LSTM	units	32	4480
Dense	units	6	198
Reshape	-	(3, 2)	-

9.3.4 Hybrid Network

This section combines a CNN and an LSTM to build a hybrid CNN-LSTM model. **Figure 9.10** illustrates the framework of this hybrid network. In the hybrid structure, the CNN plays a role in feature extraction, and the LSTM works as a temporary forecasting layer to calculate such extracted features along the time axis. Therefore, the LSTM undertakes the functionality of forecasting for the extracted features rather than raw time series. The CNN-LSTM structure is supposed to be an encoder-decoder mechanism, where the CNN is the encoder and the LSTM is the decoder [78].

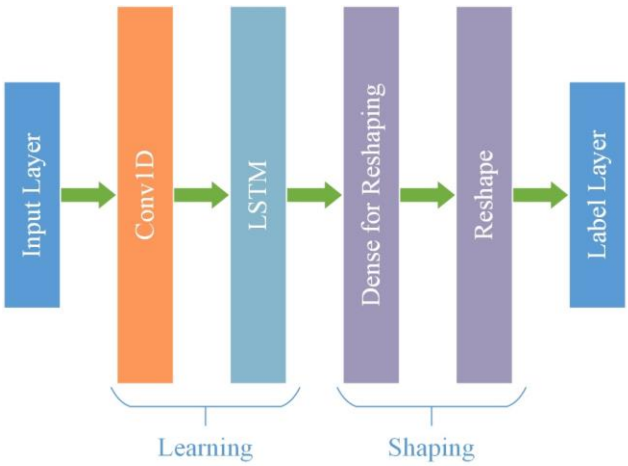

Figure 9.10 Structure of the hybrid CNN-LSTM

Table 9.5 lists the layer configuration of the CNN-LSTM. Owing to recurrent prediction, the

CNN-LSTM still has a concise layer structure. However, the LSTM layer requires more network parameters because the extracted features increase the dimension of the LSTM input.

Table 9.5 Configuration of the CNN-LSTM layers

Layer	Arguments	Value	Parameters
Conv1D	filters	32	224
	kernel size	3	
	activation	ReLU	
	padding	same	
LSTM	units	32	8320
Dense	units	6	198
Reshape	-	(3, 2)	-

Chapter 10 Case Study

This case study consists of several simulations that perform the OPS control. The first simulation will test the accuracy and reliability of the OPS in flexible wind generation. The second case will compare ANN-based aerodynamic models regarding model accuracy and control performance. The third case will discuss the advantages of wind forecasting in turbine control. The last case will compare the RL-based OPS and the PRPT-based OPS to enhance turbine intelligence. Considering that the YAC depends on wind direction, only **case 10.3** involves actual yaw influences, and the rest cases are unyawed. **Table 10.1** summarizes the simulation purposes of the above cases. If a turbine control can track each power strategy command, it will ensure long-term stability and reliability. Thus, this case study mainly tests control performance in periodic updates. The duration of a stochastic simulation depends on whether random wind velocities can cause enough kinetic changes for statistical analysis. The update span in stochastic tests is related to system inertia, as discussed in **section 4.3**.

Table 10.1 Brief description of four simulation cases

Case	Turbine	Related Chapters	Purposes
10.1	NREL 5-MW	3.1, 4.1, 4.2, 5.2, 6.1, 6.2, 8.1, 8.3, 8.4, 8.5	• examine the effectiveness of the PRPT-based OPS; • compare the model-free and model-based controllers.
10.2	IEA 10-MW	3.2, 5.1, 5.2, 5.4, 6.2, 8.5	• test three ANN-based aerodynamic models; • discuss the influence of different models on control.
10.3	IEA 15-MW	3.3, 6.2, 6.3, 8.5, 9.1, 9.2, 9.3	• test four DL-based models about wind forecasting; • compare control performance without and with wind forecasting.
10.4	IEA 15-MW	3.3, 7.1, 7.2, 7.3, 7.4, 7.5, 8.5	• test two agents for the RL-based OPS; • examine the effectiveness of the RL-based OPS.

10.1 Power Strategy

This case employs the NREL 5-MW to simulate the control response of the turbine under the MPS and OPS. If the PID proposed in **section 8.1** runs under the MPS, it naturally would be an extension of the FAST baseline that adds actual generator control and tower fore-aft damping. The case study still uses the terminology 'FAST' to represent the MPS-based PID (MPS-PID) to follow convention. This section includes a comparison between the FAST and the OPS-based LQR (OPS-LQR), an examination of the OPS with different power targets, an investigation of anti-disturbance, and a survey of the model-based controllers.

10.1.1 FAST Comparison

This simulation runs a wind velocity series of several steps to compare the FAST and the LQR, as shown in **Figure 10.1**. First, there are some variations in rotation at wind speeds of 6 and 8 m/s. These rotation differences come from two aspects: the OPS algorithm (**Algorithm 3**) solves the 2-DOF optimization with finite precision, and the built-in aerodynamic model is not 100% accurate. However, their steady outputs do not exhibit significant differences because long blades have a tolerance for a slight divergence of rotor speed. Besides, determining the optimal TSR ratio relies on a set of measurements, and its resolution is also limited. Hence, both speed references are effective and accurate. The LQR has smaller output fluctuations at 8 and 10 m/s steps than the FAST. The LQR uses a small-signal model to optimize control response, while the FAST regulation passively changes control inputs according to measured errors.

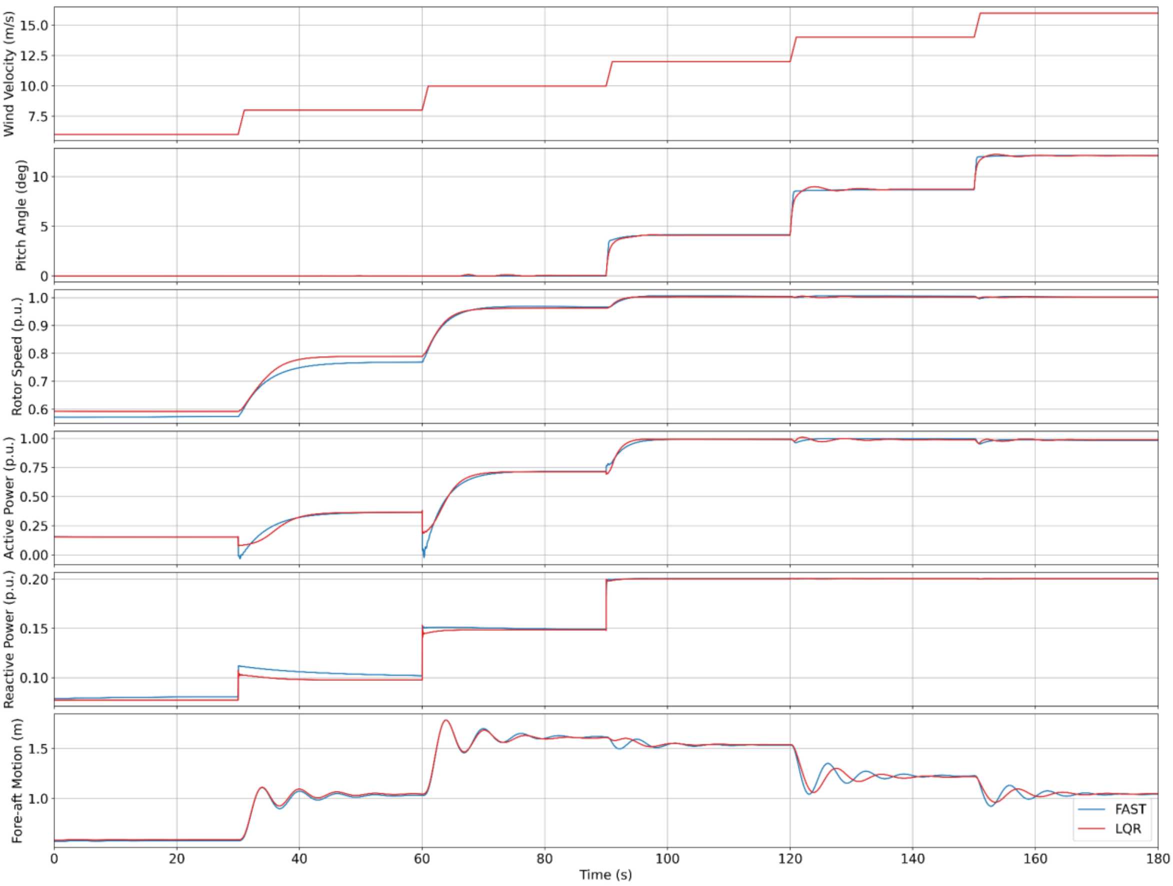

Figure 10.1 Comparison of the step response of the FAST and LQR (active/reactive power is the net value of stator output and rotor consumption)

Besides, the PAC of the LQR has more frequent regulations, especially for wind speeds over 10 m/s, which results in more effective tower fore-aft damping. Besides, the test region (60~90 s) of the LQR has additional pitch regulation to reduce tower oscillation. Tower damping can lead to some output fluctuations but prolong the mechanical lifetime of the tower. Therefore, the OPS-LQR better coordinates the RSC and PAC for multi-objective control. The NREL 5-MW has a gearbox to couple the low-speed and high-speed shafts. However, any model-based control in this study applies the one-mass model to predict the rotation of the main shaft. The simulation of **Figure 10.1** indicates that the one-mass model ignoring gearing harmonics is sufficient for control purposes.

Table 10.2 Comparison of the transient response of the FAST and LQR

Wind Speed (m/s)		Rotor Speed				Output Power		
		Rise Time (s)	Settling Time (s)	Overshoot	Error (p.u.)	Active Power (p.u.)	Power Factor	Stable Time (s)
8	FAST	9.45	13.22	0.08%	0.0007	0.36	0.96	20.16
	LQR	7.47	10.29	0.00%	-0.0012	0.36	0.97	16.39
10	FAST	6.15	8.31	0.76%	0.0059	0.72	0.98	14.16
	LQR	5.27	7.36	0.00%	0.0017	0.71	0.98	11.41
12	FAST	3.48	5.10	4.55%	0.0047	0.99	0.98	6.61
	LQR	2.85	4.10	0.24%	0.0031	0.99	0.98	4.54
14	FAST				0.0051	1.00	0.98	2.63
	LQR				0.0026	0.99	0.98	6.41
16	FAST				0.0027	0.98	0.98	2.24
	LQR				0.0024	0.99	0.98	5.39

Note: settling time is for step magnitude within ±5%, and stable time is for value magnitude within ±1%.

Table 10.2 provides the metrics of the transient response in **Figure 10.1**. The LQR contributes faster rise time and settling time and has fewer overshoots in rotor speed, in which the cases of 8 and 10 m/s are zero-overshoot. Hence, the LQR surpasses the FAST in accelerating rotation. Besides, the FAST results in a speed error of about 0.005 p.u. in the range of 10~14 m/s, while the speed error of the LQR falls in a region of -0.001~0.003 p.u. The FAST and LQR have similar steady outputs, but the output stable time is not the same due to the effective tower damping of the LQR. Therefore, the OPS-based LQR has better dynamic performance while ensuring the same power capture as the MPS-PID. Meanwhile, this simulation proves the advantage of the model-based control for the multi-objective regulation of a WT.

10.1.2 Variable Output

Aside from maximum output, the OPS-LQR allows the NREL 5-MW to capture and output user-needed power. **Figure 10.2** displays an example of the LQR tracking targets of 25%, 50%, 75%, and 100% rated power. If a power target exceeds the upper limitation of power capture, the OPS will automatically adjust to capture maximum power. For example, the bounded wind capture is responsible for the output overlaps of 6~10 m/s. The constrained generator load affects the acceleration of the main shaft when targeting a lower output. The reason is the reduced electromagnetic torque and constant system inertia. Therefore, when applying the OPS-based control, users should allow more stable time for lower output levels in Region ii. Region iii eliminates this effect due to a constant reference of the rated speed.

Figure 10.2 Variable power production of targeting 25%, 50%, 75%, and 100%

Table 10.3 measures the steady-state output power at each wind speed. For a target output within boundaries, the absolute error of active power falls in a range of 0.001~0.014 p.u., while the average error achieves 0.006 p.u. Hence, the OPS-LQR is reliable and accurate in meeting the requirements of demand-oriented applications. It notes that lower output scenarios lead to the deviation of the power factor because of the rotor loss of the DFIG. Thus, low-load generation should consider placing a power factor compensator to ensure grid operation.

Table 10.3 Summary of the variable output of OPS-LQR with four targets

Wind Speed (m/s)	Target	Active Power (p.u.)	Power Factor	Wind Speed (m/s)	Target	Active Power (p.u.)	Power Factor
7	25%	0.24	0.94	12	25%	0.25	0.98
	50%	0.24	0.94		50%	0.49	0.98
	75%	0.24	0.94		75%	0.74	0.98
	100%	0.24	0.94		100%	0.99	0.98
8	25%	0.25	0.96	13	25%	0.25	0.98
	50%	0.36	0.97		50%	0.50	0.98
	75%	0.36	0.97		75%	0.74	0.98
	100%	0.36	0.97		100%	0.99	0.98
9	25%	0.25	0.97	14	25%	0.25	0.98
	50%	0.50	0.97		50%	0.49	0.98
	75%	0.52	0.97		75%	0.74	0.98
	100%	0.52	0.97		100%	0.99	0.98
10	25%	0.25	0.98	15	25%	0.25	0.98
	50%	0.50	0.98		50%	0.49	0.98
	75%	0.71	0.98		75%	0.74	0.98
	100%	0.71	0.98		100%	0.99	0.98
11	25%	0.25	0.98	16	25%	0.25	0.98
	50%	0.50	0.98		50%	0.49	0.98
	75%	0.74	0.98		75%	0.74	0.98
	100%	0.94	0.98		100%	0.99	0.98

Note: underlined values indicate that target power exceeds maximum capture.

10.1.3 Anti-disturbance Test

Figure 10.3 shows an anti-disturbance simulation for different targets under a random wind velocity of 11.9~12.1 m/s. This test configures the OPS using a base wind velocity of 12 m/s. Intuitively, the FAST and LQR (100%) have similar anti-disturbance performance. With the decrease of targets, rotor speed gets lower but is still around the rated speed, and tower bending gets apparent alleviation.

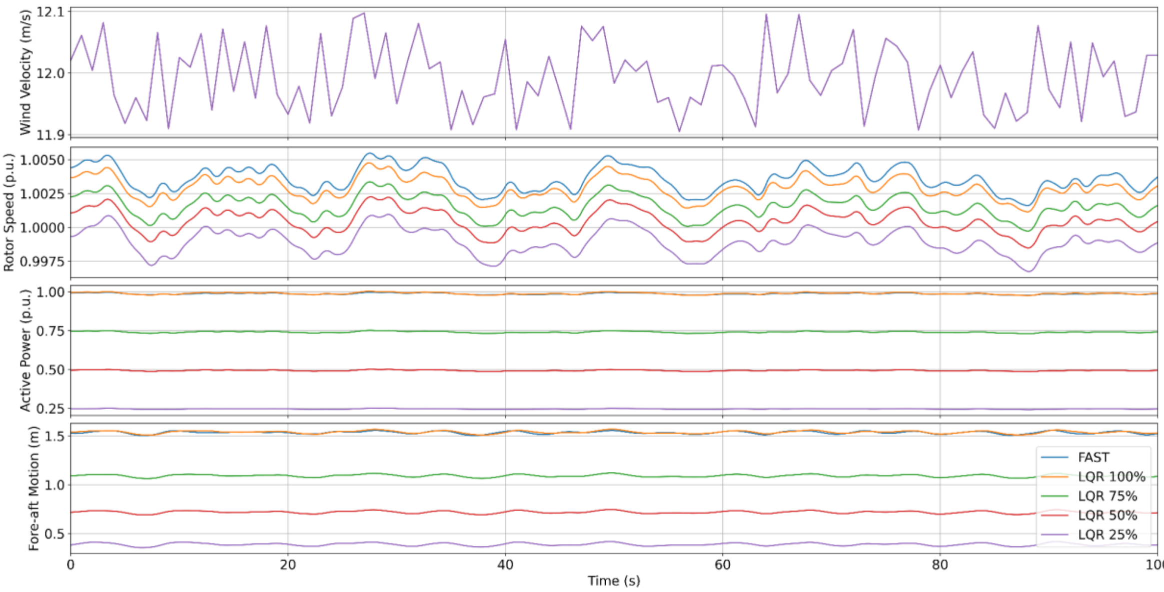

Figure 10.3 Anti-disturbance test of targeting 25%, 50%, 75%, and 100%

Table 10.4 provides the statistics of anti-disturbance about the FAST and variable-output LQR. First, the LQR has better RAC results than the FAST, and its average power is closer to the power rating. Second, a lower target reduces output fluctuation and tower bending. Therefore, the OPS-LQR is proven robust in tracking a target, and there is no worry about system stability when applying the OPS.

Table 10.4 Statistics of the anti-disturbance performance of the FAST and LQR

		FAST	**LQR 100%**	**LQR 75%**	**LQR 50%**	**LQR 25%**
ω_d (p.u.)	Mean	1.0036	1.0029	1.0016	1.0004	0.9989
	Std.	0.0009	0.0008	0.0008	0.0008	0.0009
P_n (p.u.)	Mean	0.9878	0.9899	0.7414	0.4940	0.2464
	Std.	0.0048	0.0062	0.0047	0.0034	0.0019
x_m (m)	Mean	1.5342	1.5386	1.0929	0.7202	0.3909
	Std.	0.0130	0.0133	0.0127	0.0127	0.0140

10.1.4 Controller Comparison

Chapter 8 proposes four controllers (PID, LQR, RHC, and MPC) to realize the OPS control. This part provides a brief example (**Figure 10.4**) of their dynamic differences. Since the PID always tries to eliminate the largest error in the loops, the PAC of the PID quickly converges to the vicinity of the pitch reference and subsequently considers the coordination of rotor stabilizing and tower damping. This inherent shortage leads to a sluggish recovery of rotation and output. Therefore, the model-free control is not ideal for the multi-objective regulation of the 3-DOF turbine.

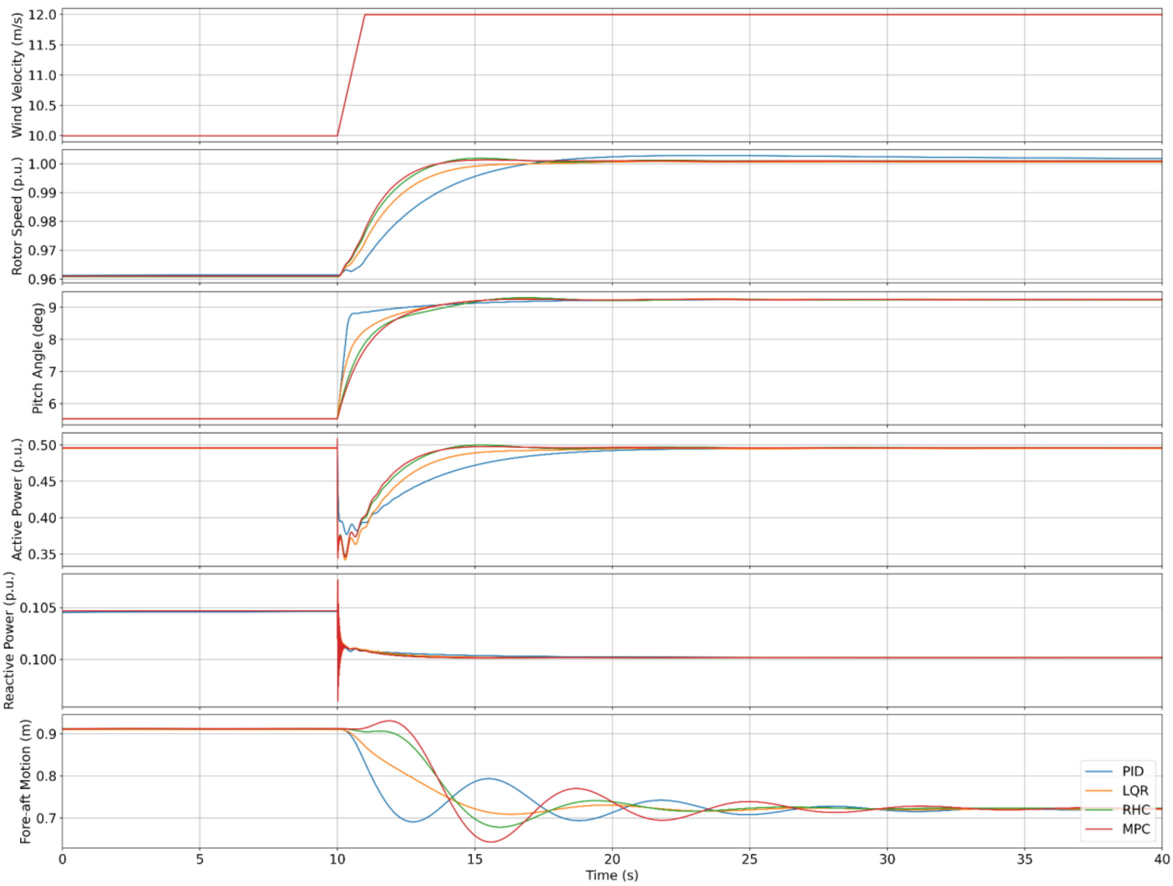

Figure 10.4 Comparison of four OPS-compatible controllers

In contrast, a model-based control can achieve better multi-objective regulation owing to its trajectory estimation, but the LQR, RHC, and MPC have different control policies. The LQR calculates its gain on an infinite horizon, which accounts for better tower damping. However, the LQR is sensitive to the weighting matrices, and the Raccati solver cannot find a finite solution for some weights. Besides, the LQR supposes a process to be strictly linear, but WTs are heavily nonlinear due to aerodynamics. Therefore, some weights cause the eigenvalues of the associated Hamiltonian pencil to be close to the imaginary axis. The tuning of the LQR should pay attention to the above issues. Compared with the LQR, the RHC and MPC have more tolerance for the weighting matrices owing to their finite prediction horizon. The RHC and MPC generally have similar rotation and power recoveries, but the MPC sacrifices part tower damping for the recoveries. In conclusion, the LQR benefits tower damping, and the RHC and MPC prefer to recover rotation and power first.

Table 10.5 compares the transient response of four controllers. First, the time consumption of accelerating rotation results in a ranking (PID>LQR>RHC>MPC) considering rise time and settling time. Second, the stable time of power also derives the same ranking. Therefore, the MPC surpasses the RHC, LQR, and PID in terms of prompt response and stable output. The rest of the case study will employ the MPC as the sole controller.

Table 10.5 Comparison of the transient response of four controllers

	Rotor Speed				**Output Power**		
	Rise Time (s)	**Settling Time (s)**	**Overshoot**	**Error (p.u.)**	**Active Power (p.u.)**	**Power Factor**	**Stable Time (s)**
PID	4.89	6.89	2.43%	0.0018	0.50	0.98	8.81
LQR	3.27	4.60	0.29%	0.0007	0.49	0.98	5.30
RHC	2.64	3.40	2.26%	0.0010	0.50	0.98	3.62
MPC	2.44	3.24	1.04%	0.0009	0.50	0.98	3.53

10.1.5 Summary

This case employs the model-based LQR to implement the PRPT-based OPS. According to the comparison with the FAST, the OPS algorithm (**Algorithm 3**) ensures the reliability of maximum capture if targeting 100% output. Meanwhile, the LQR is significantly superior to the FAST in accelerating the main shaft rotation and stabilizing the DFIG output power. In addition to a target of 100%, this study also provides a set of targets (25%, 50%, and 75%) to examine the accuracy of variable output. The LQR can achieve an average error of 0.006 p.u. for tracking a power command, proving an output accuracy of over 99%. Therefore, the OPS can find a reliable and accurate 2-DOF solution (speed and pitch) to capture the desired power. Besides, the anti-disturbance test indicates that the OPS-LQR is robust in tracking different targets. Along with the LQR, this case also discusses other OPS-compatible controllers (PID, RHC and MPC). Model-based control (LQR, RHC, and MPC) can achieve better multi-objective regulation than model-free control (PID). In addition, the MPC has advantages in the responses of speed and power compared to the LQR and RHC.

10.2 Aerodynamic Modelling

This section applies the OPS-based MPC (OPS-MPC) on the IEA 10-MW. First, this study trains and tests three ANN-based aerodynamic models to compare the accuracy of the MIMO prediction. Subsequently, full-load and half-load simulations investigate the OPS-MPC performance on different ANNs.

10.2.1 Model Comparison

This part trains each model 30 times to avoid random effects caused by the training optimizer and random seeds. **Figure 10.5** displays the scatter plot and error distribution of predictions on the testing data (10%). The experimental results indicate each model has desirable linear fittings, but they show different characteristics regarding error distribution.

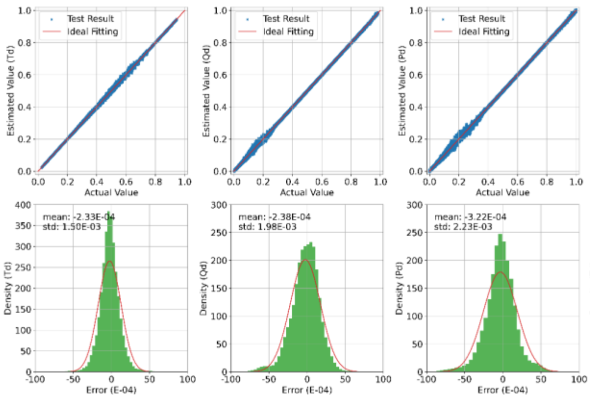

Figure 10.5a RBFN prediction results (high linear fitting, medium error divergence)

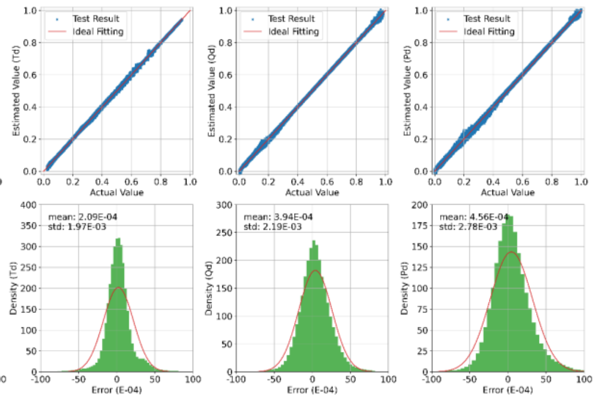

Figure 10.5b DNN prediction results (medium linear fitting, high error divergence)

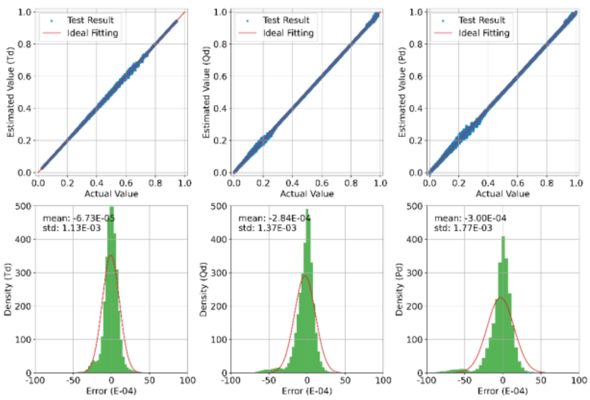

Figure 10.5c HDNN prediction results (high linear fitting, low error divergence)

Figure 10.5 Testing results of the RBFN, DNN, HDNN

The maximum density of the HDNN can reach about 500, 500, and 400 for thrust, torque, and power, respectively, which outperforms the RBFN and DNN. Besides, the error deviation intuitively derives a ranking of centralizing (HDNN>RBFN>DNN). Therefore, the RBF kernel is more efficient than the stack of dense layers for aerodynamic modelling. Besides, the HDNN can improve model accuracy by combining feature extraction and deep network.

Table 10.6 summarizes the statistics of 30 trainings regarding the trusted region, R^2, MAE, and RMSE. When examining these metrics, their accuracies are delusive due to incredible values. For example, the R^2 reaches at least 0.999, even in some cases exceeding 0.9999. First, aerodynamic data generated by the aerodynamic solver are noiseless and unbiased. Besides, an R^2 of 0.999 can cause a 10-kW mismatch of 10-MW output, which is still a non-negligible error for the control system.

Table 10.6 Statistics of the predictions on the testing data

Model	Variable	-Δδ	Δδ	R^2	MAE	RMSE
RBFN	T_d	-10028.77	9045.07	0.999894	0.000755	0.001521
	Q_d	-114619.74	105778.11	0.999938	0.001123	0.001995
	P_d	-81628.16	74126.30	0.999943	0.001138	0.002251
DNN	T_d	-12088.87	12972.83	0.999819	0.000851	0.001986
	Q_d	-114603.31	129208.62	0.999923	0.001209	0.002226
	P_d	-91848.79	102477.44	0.999911	0.001471	0.002817
HDNN	T_d	-7321.51	7036.87	0.999941	0.000525	0.001133
	Q_d	-81475.28	70941.46	0.999970	0.000589	0.001399
	P_d	-65225.35	58239.97	0.999964	0.000712	0.001791

Note: Δδ is the trusted region of the 3δ rule (T_d: N, Q_d: N·m, P_d: Watt), R^2, MAE, and RMSE take normalized values.

The R^2, MAE, and RMSE of each variable of the HDNN are the best, which is entirely superior to the RBFN and DNN. These metrics numerically verify the ranking (HDNN>RBFN>DNN). In addition, the symmetric radii of the 3δ rule are more of interest since the sensitivity estimation depends on the trusted region. The HDNN has more concentrated radii than the RBFN and DNN, and the HDNN and RBFN have lower radii, which implies that the RBF kernel can further improve prediction stability.

10.2.2 Full-load Test

The full-load test naturally corresponds to rated output in Region iii and mainly evaluates anti-disturbance performance [54]. This test employs a random wind series that ranges from 12~13 m/s for enough wind drive. The control system receives wind prediction per 20-s to execute the OPS and update control configuration. Due to rated generation in Region iii, this test also includes the FAST baseline for comparison.

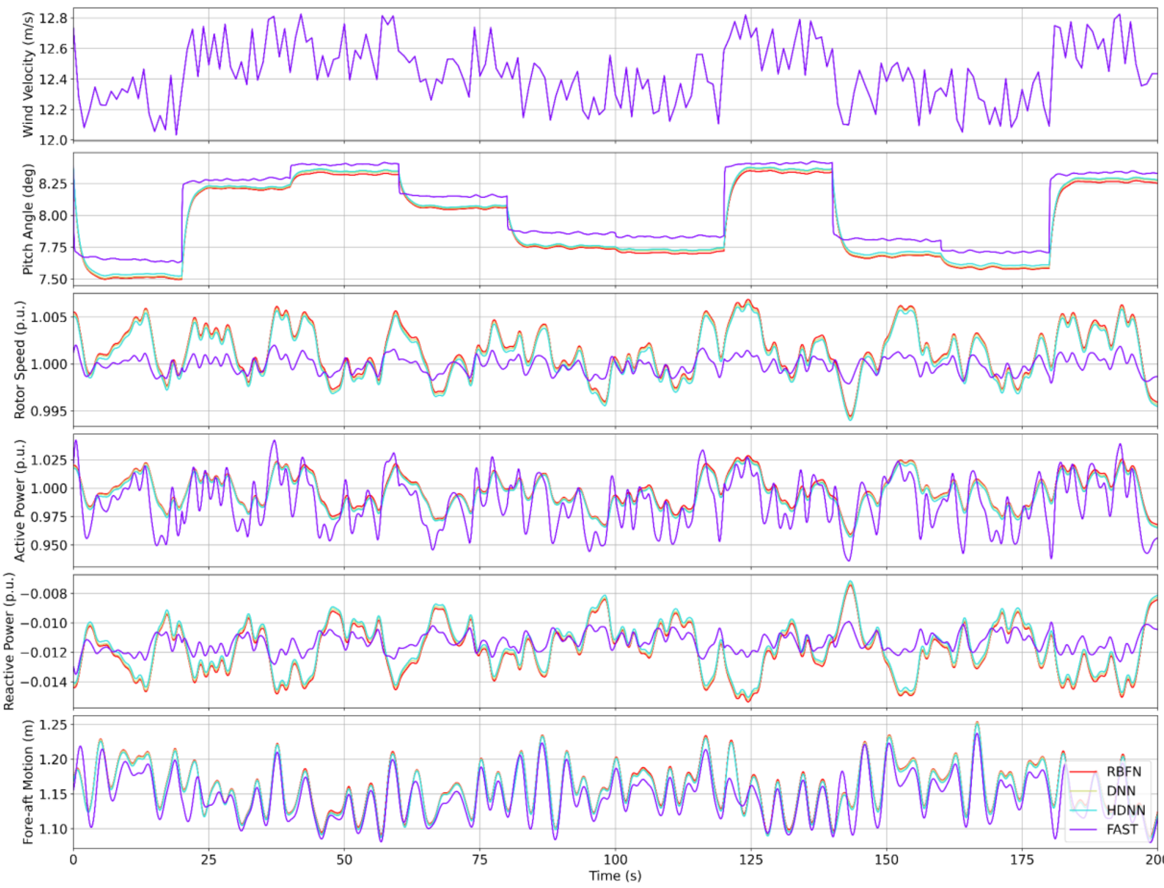

Figure 10.6 Simulation result for the full-load generation (active/reactive power is the net value)

Figure 10.6 shows that the RBFN, DNN, and HDNN responses are almost identical but have slight PAC differences. Consequently, each model has a similar PAC result for the same speed reference. The FAST relies on the speed loop that sets stable rotation as the primary regulation objective [26], so the generator needs to absorb most of the power variation, which leads to a smooth speed line. The FAST does not involve the trajectory estimations

(**Eqs. (121)(122)**) to predict rotor speed or fore-aft motion. Therefore, the FAST passively waits for changes in rotation and motion, aggravating output and tower oscillations. In addition, the reactive output of the FAST has fewer fluctuations due to the independent dq-axis loops. However, the MPC knows the dq-axis interaction of voltage and current and adjusts the d-axis magnetizing for smoother active power. Compared with the FAST, the OPS-MPC can balance multiple objectives, which benefits more stable active production.

10.2.3 Half-load Test

The full-load scenario mainly relies on pitch regulation to output rated power, so there is no intrinsic difference between the MPS and the OPS. Therefore, this case sets the output target to 50% (0.5 p.u.) to illustrate the advantage of power demand tracking. The half-load test applies a wind time series of 8.5~9.5 m/s that can cause speed variation, which requires the coordination of the RSC and PAC.

According to **Figure 10.7**, different models result in various speed and pitch references, but the OPS-MPC reliably tracks 50%, no matter how wind changes. Hence, the MPC is not model-sensitive and is compatible with different models. However, some noticeable power hills occur at 40, 60, 100, 120, 160, and 180 seconds because rotor transition unavoidably leads to sudden output change to absorb or supply kinetic energy. Therefore, energy storage [53], as an auxiliary device, helps absorb or fill such huge power fluctuations. Since Region ii requires considerable time due to system inertia, the power target and updating period are better to hold as long as possible. Although the MPC suffers such wind stochastic variations, no situation is out of control, which proves robustness and anti-disturbance. **Figure 10.7e** provides the FAST simulation for comparing the MPPT and the PRPT. In conclusion, this test indicates that the OPS-MPC can accurately output 0.5 p.u. as a reliable energy source.

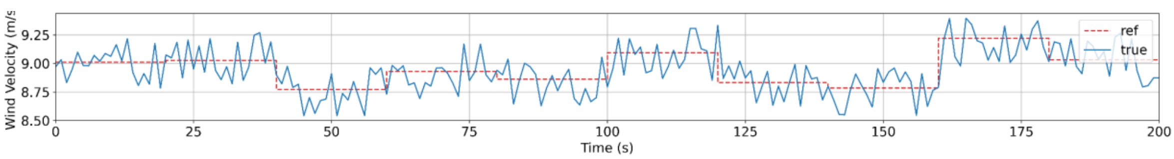

Figure 10.7a Wind series (8.5~9.5 m/s in Region ii)

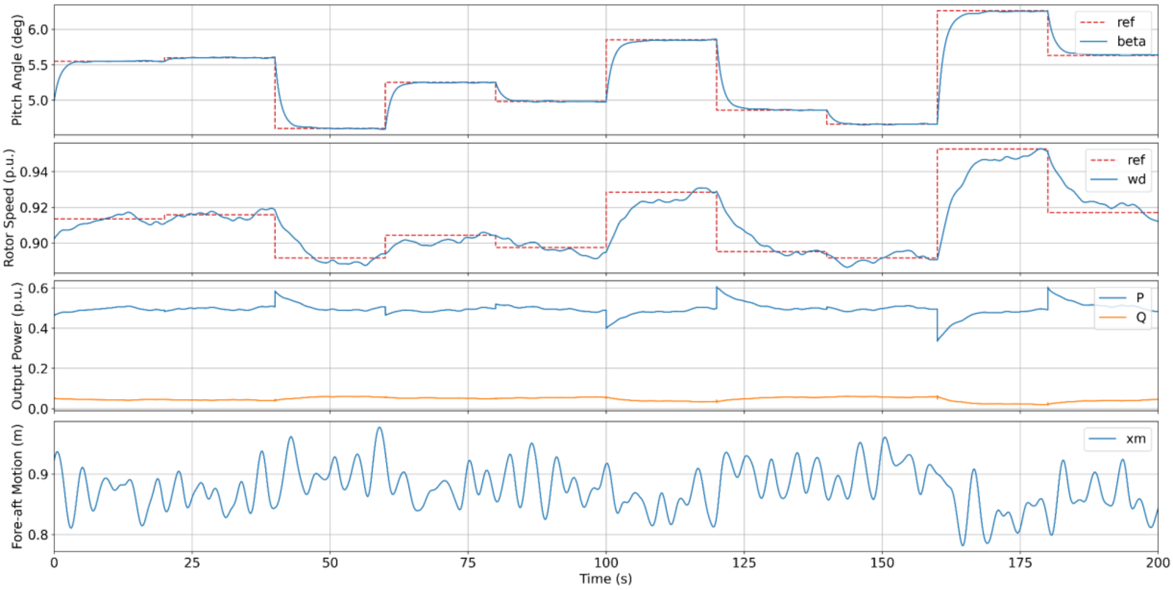

Figure 10.7b RBFN (medium power fluctuation, 0.34~0.61 p.u.)

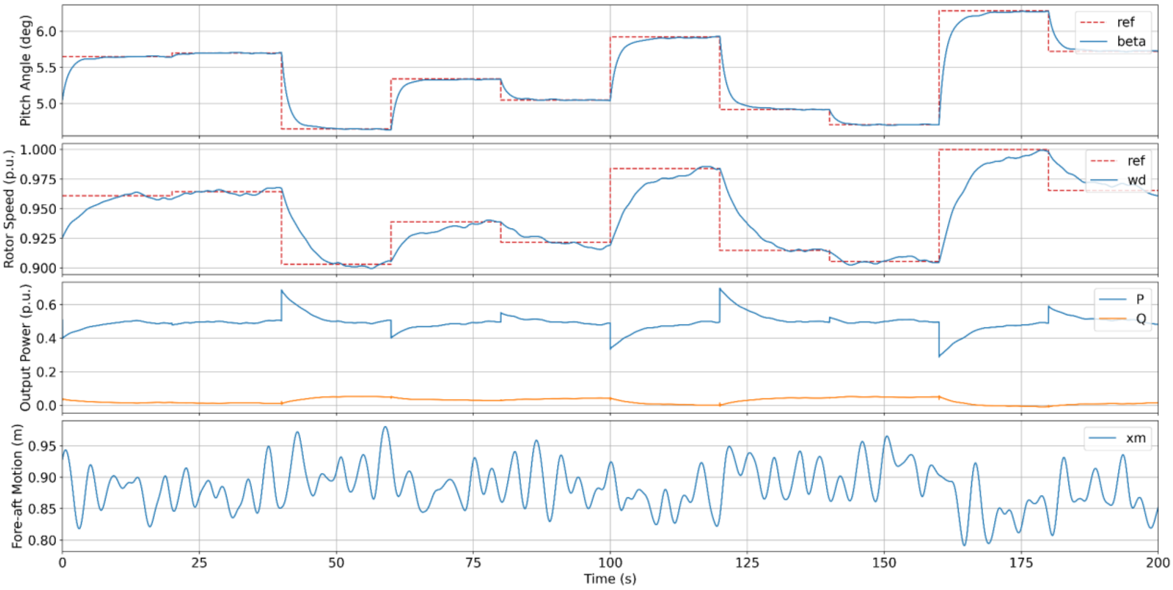

Figure 10.7c DNN (largest power fluctuation, 0.29~0.70 p.u.)

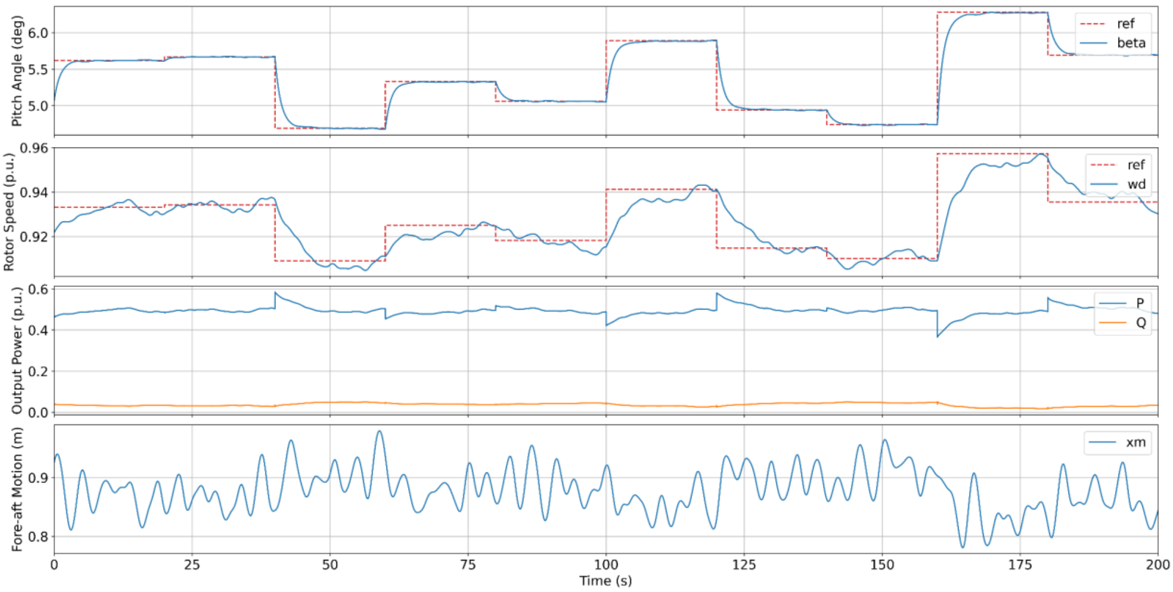

Figure 10.7d HDNN (lowest power fluctuation, 0.37~0.59 p.u.)

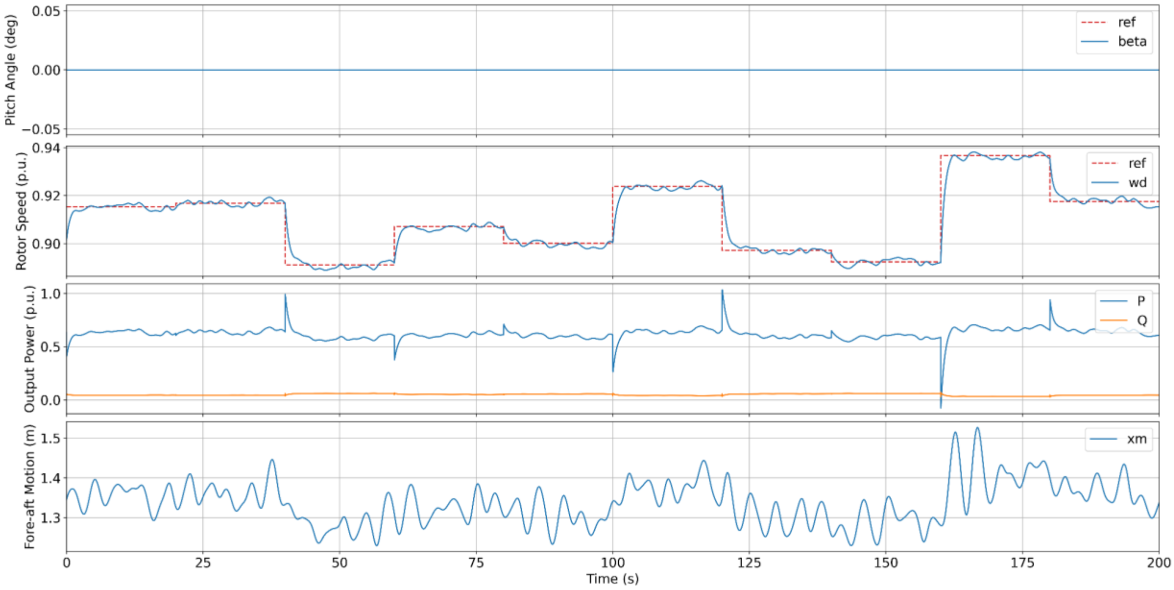

Figure 10.7e FAST (unable to track 0.5 p.u.)

Figure 10.7 Simulation result for tracking a target of 50%
(P / Q is the net value)

Table 10.7 gives the statistical results of **Figure 10.6** and **Figure 10.7**. In the full-load test, the MPC achieves a mean rotor speed of 1.001 ± 0.003 p.u., an active output of 0.997 ± 0.014 p.u., and a fore-aft motion of 1.159 ± 0.033 m, in which the net active production reaches an accuracy of over 99%. Meanwhile, different models do not give rise to significant statistical differences. In contrast, the FAST output varies with 0.987 ± 0.022 p.u. whose fluctuation range is 1.57 times the MPC.

Table 10.7 Statistics of the 100% and 50% load simulations

Load Level		100%				50%			
		Mean	**Std.**	**Min.**	**Max.**	**Mean**	**Std.**	**Min.**	**Max.**
RBFN	ω_d (p.u.)	1.00147	0.00261	0.99444	1.00687	0.90992	0.01649	0.88653	0.95286
	P_n (p.u.)	0.99815	0.01450	0.95964	1.02866	0.49563	0.02687	0.33533	0.60645
	x_m (m)	1.16050	0.03293	1.08898	1.25404	0.87681	0.03542	0.78155	0.97754
DNN	ω_d (p.u.)	1.00127	0.00261	0.99423	1.00662	0.94430	0.02843	0.89957	0.99943
	P_n (p.u.)	0.99690	0.01446	0.95851	1.02684	0.49377	0.04535	0.28712	0.69851
	x_m (m)	1.15889	0.03308	1.08661	1.25289	0.88433	0.03457	0.79125	0.98038
HDNN	ω_d (p.u.)	1.00100	0.00261	0.99399	1.00639	0.92682	0.01322	0.90473	0.95716
	P_n (p.u.)	0.99566	0.01443	0.95696	1.02570	0.49451	0.02181	0.36546	0.58577
	x_m (m)	1.15727	0.03289	1.08546	1.25044	0.87817	0.03626	0.78063	0.97980
FAST	ω_d (p.u.)	0.99987	0.00086	0.99788	1.00201	0.90960	0.01383	0.88892	0.93818
	P_n (p.u.)	0.98713	0.02233	0.93587	1.04230	0.62126	0.05564	-0.07516	1.03410
	x_m (m)	1.14681	0.03163	1.08036	1.23710	1.33738	0.05288	1.22923	1.52665

Note: $P_n = P_s - P_r$, due to the rotor consumption of the DFIG.

In the half-load scenario, the MPC obtains a rotation of 0.927 ± 0.019 p.u., a power of 0.495 ± 0.031 p.u., and a motion of 0.880 ± 0.035 m. Also, the output accuracy is over 99%, while the FAST cannot respond to the 50% target due to its intrinsic MPPT. However, the speed and power variations of the DNN are almost twice the HDNN and 1.7 times the RBFN. It indicates that the HDNN contributes to more robust control performance when coordinating the RSC and PAC. In conclusion, the HDNN-MPC, as the optimal OPS scheme, is more reliable and accurate in producing constant output, ensuring safe rotation, and damping tower motion under stochastic winds.

10.2.4 Summary

This case implements an OPS-MPC framework to adapt to demand-oriented wind power scenarios. This framework consists of an ANN-based aerodynamic model, an OPS algorithm with local linearization, a real-time updated control model, and a QP-based MPC. This study investigates three ANNs (RBFN, DNN, and HDNN) for integration with the OPS. According to extensive experiments, the HDNN-MPC scheme outperforms the others in model accuracy, multi-objective regulation, control performance, and power quality. Proven by the full-load and half-load tests, the OPS-MPC reliably tracks power targets, which also considers shaft and tower behaviours in the coordinated regulation of the DFIG and pitch servo. Therefore, the OPS-MPC can replace conventional turbine control to achieve more flexible wind generation.

10.3 Wind Forecasting

The case utilizes actual wind velocity and direction to test the 3-DOF turbine system under the OPS, which enables the YAC aside from the coordinated RSC and PAC. A comparison first discusses the forecasting accuracy of four wind models (DNN, CNN, LSTM, and CNN-LSTM). After model verification, a simulation will test the 3-DOF OPS-MPC of the IEA 15-MW with and without wind forecasting.

10.3.1 Model Comparison

Wind forecasting provides critical velocity and direction information for the power strategy to determine the reference state. **Figure 10.8** displays compass wind forecasted by four time-series models. Given a 30-s prediction length, there will be 86,400 predictions for a 30-day wind series. According to the figures, predicted wind speed aligns with actual velocity and achieves high accuracy. Predicted yaw angle often matches wind direction and keeps track of rapid fluctuation.

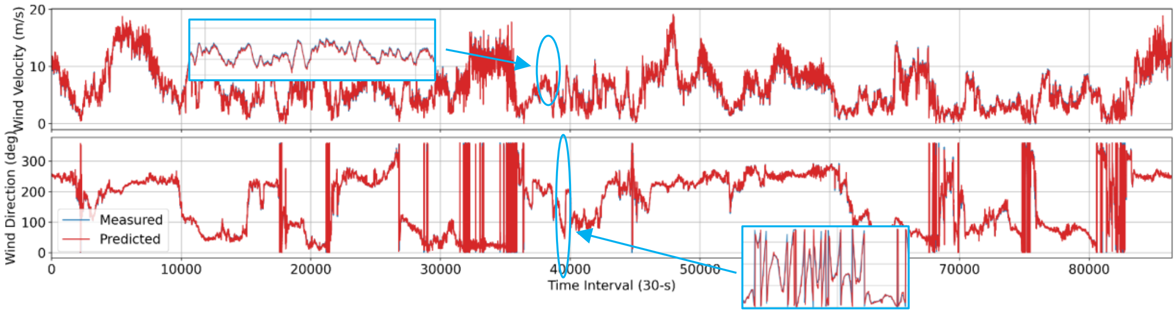

Figure 10.8a DNN prediction series

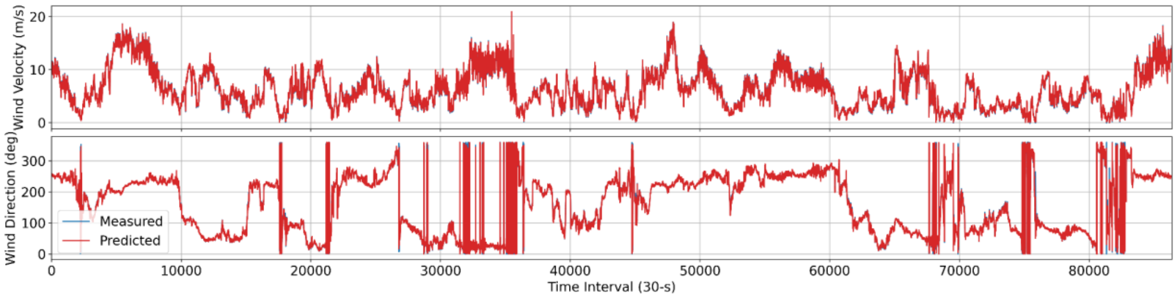

Figure 10.8b CNN prediction series

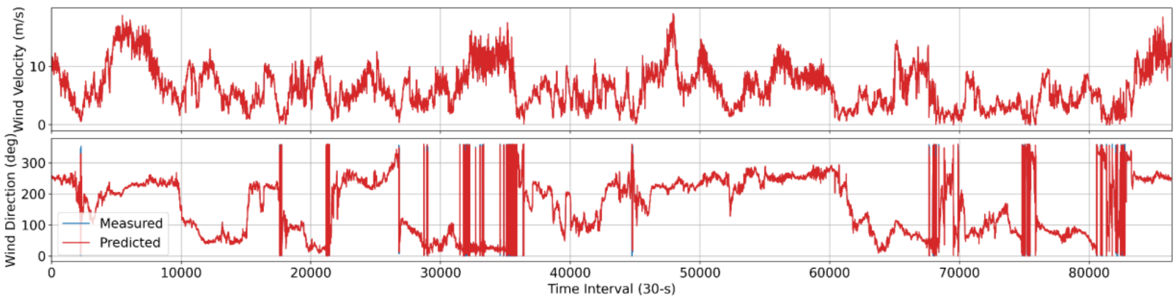

Figure 10.8c LSTM prediction series

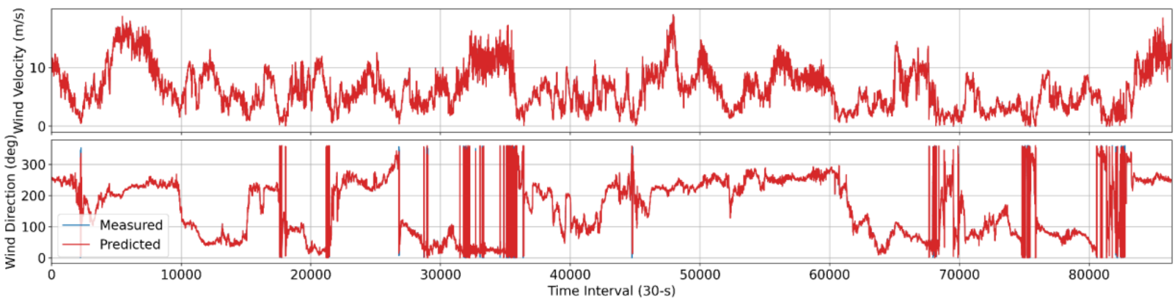

Figure 10.8d CNN-LSTM prediction series

Figure 10.8 Example of wind series forecasting on one-month data

Table 10.8 summarizes the evaluation indexes of 30 training results. Four models achieve an average R^2 of over 0.996 in wind speed and direction, indicating the wind modelling frame of the compass-vector transformation and vector series model is feasible and reliable. However, there are some noticeable model differences. Firstly, the MLPN has the worst direction prediction, considering a max MAE of 1.95° and a max RMSE of 6.97°. Therefore, a conventional stacked-layer network is weaker than a network containing feature extraction or recurrent structure. The CNN enhances the direction accuracy compared with the MLPN, but its stability is slightly worse than that of the LSTM and CNN-LSTM. The reason is that the CNN primarily relies on the dense layer to learn features extracted by the convolution. Therefore, a model with a recurrent structure is more suitable for wind vector series.

Table 10.8 Statistics of the model accuracy over 30 tests

			Mean	**Std.**	**Min.**	**25%**	**50%**	**75%**	**Max.**
MLPN	v' (m/s)	R^2	0.9969	0.0006	0.9952	0.9966	0.9970	0.9974	0.9978
		MAE	0.0696	0.0359	0.0278	0.0417	0.0610	0.0881	0.1778
		RMSE	0.1961	0.0188	0.1644	0.1816	0.1945	0.2051	0.2453
	θ' (°)	R^2	0.9969	0.0009	0.9938	0.9966	0.9970	0.9976	0.9979
		MAE	0.6272	0.3784	0.2467	0.3976	0.5345	0.7547	1.9528
		RMSE	4.8753	0.6284	4.0581	4.3771	4.8631	5.1295	6.9666
CNN	v' (m/s)	R^2	0.9977	0.0004	0.9968	0.9976	0.9977	0.9980	0.9982
		MAE	0.0494	0.0196	0.0286	0.0336	0.0434	0.0579	0.0999
		RMSE	0.1688	0.0133	0.1500	0.1597	0.1675	0.1725	0.1994
	θ' (°)	R^2	0.9974	0.0004	0.9961	0.9973	0.9975	0.9977	0.9980
		MAE	0.4468	0.1859	0.2593	0.3214	0.4188	0.4891	1.1263
		RMSE	4.4847	0.3353	3.9698	4.2824	4.4312	4.6185	5.5511
LSTM	v' (m/s)	R^2	0.9979	0.0002	0.9975	0.9978	0.9979	0.9980	0.9984
		MAE	0.0394	0.0096	0.0277	0.0329	0.0371	0.0443	0.0635
		RMSE	0.1624	0.0074	0.1434	0.1589	0.1628	0.1675	0.1755
	θ' (°)	R^2	0.9973	0.0003	0.9966	0.9971	0.9972	0.9974	0.9979
		MAE	0.3577	0.0767	0.2567	0.3084	0.3410	0.3991	0.5361
		RMSE	4.6320	0.2287	4.0804	4.4830	4.6702	4.7458	5.1869
CNN-LSTM	v' (m/s)	R^2	0.9977	0.0002	0.9971	0.9976	0.9977	0.9978	0.9982
		MAE	0.0451	0.0133	0.0270	0.0352	0.0433	0.0499	0.0762
		RMSE	0.1689	0.0086	0.1518	0.1653	0.1684	0.1714	0.1890
	θ' (°)	R^2	0.9974	0.0003	0.9970	0.9972	0.9974	0.9976	0.9980
		MAE	0.4187	0.0830	0.2929	0.3558	0.4259	0.4792	0.5757
		RMSE	4.4910	0.2498	3.9378	4.3335	4.5116	4.6770	4.8721

Considering most metric distributions, the LSTM and CNN-LSTM have similar performance and surpass the MLPN and CNN. Nevertheless, the LSTM has tiny advantages except for the RMSE of the direction, which manifests that the encoder-decoder design has no promotion. The CNN-LSTM theoretically can increase the learning efficiency of complex and diverse features, but a wind vector has only two projection features. Therefore, the encoder-decoder cannot further improve the LSTM. In conclusion, the LSTM has accurate predictions, stable training results, and economic parameters. The following control simulation will use the LSTM as the wind forecasting model.

10.3.2 Synthesis Control

Since the turbine operation depends on wind reference in each forecasting period, this section will investigate the 3-DOF OPS-MPC with or without forecasting for high and medium wind intensities. There will be four generation scenarios:

1. 100% power production at a wind speed range of 14.5~16 m/s,
2. 50% power production with the above series,
3. 50% power production at a range of 9.4~9.8 m/s,
4. 25% power production with the above series.

Since **scenario 1** satisfied the rated output condition of the MPPT-based MPS, this event also includes the FAST baseline. The non-forecasted and FAST simulations average the input wind series as the most frequent observation to update their power strategies [10]. **Scenario 1** compares and discusses the performance and reliability of the OPS-MPC, and **scenarios 2~4** examine the effects of wind forecasting on control performance.

10.3.3 High-velocity Scenario

Wind speed over 10.2 m/s can ensure sufficient wind capture for the rated power production, to which the high-velocity scenario of 14.5~16 m/s corresponds. **Figure 10.9** runs the full-load operation of the IEA 15-MW under three controls. The FAST and non-forecasted controls share the same wind data processing, but their rotation and production differ significantly. Firstly, the FAST sets rotor rotation as the main-loop objective, which accounts for fewer rotation fluctuations (about 0.0015 p.u.) than the non-forecasted MPC. On the contrary, the MPC weighing multiple objectives allows moderate speed variation for better power quality. Therefore, the active output of the MPC is intuitively closer to the installed capacity, especially when updating wind reference.

The forecasted control has more stable rotor speed and power output than the non-forecasted. The rotor speed of the forecasted control distributes from 0.996 to 1.003 p.u. However, the non-forecasted fluctuates from 0.993 p.u. to 1.003 p.u., and its fluctuations look more aggressive. As one of the most critical stability indexes, rotor speed reflects the capability of suppressing incoming wind uncertainties of a control design. Since the wind forecasting model provides a reliable wind estimation, the control unit updates an accurate reference point and a reliable small-signal model for the MPC. Thus, the results of the forecasted control are close to expectations. When examining velocity and direction predictions, a wind forecasting model behaves as sampling future winds with the zero-hold method. In contrast, the non-forecasted method cannot reach such accuracy. For example, the zone of 570~780 s has an apparent gap between wind reference and actual wind.

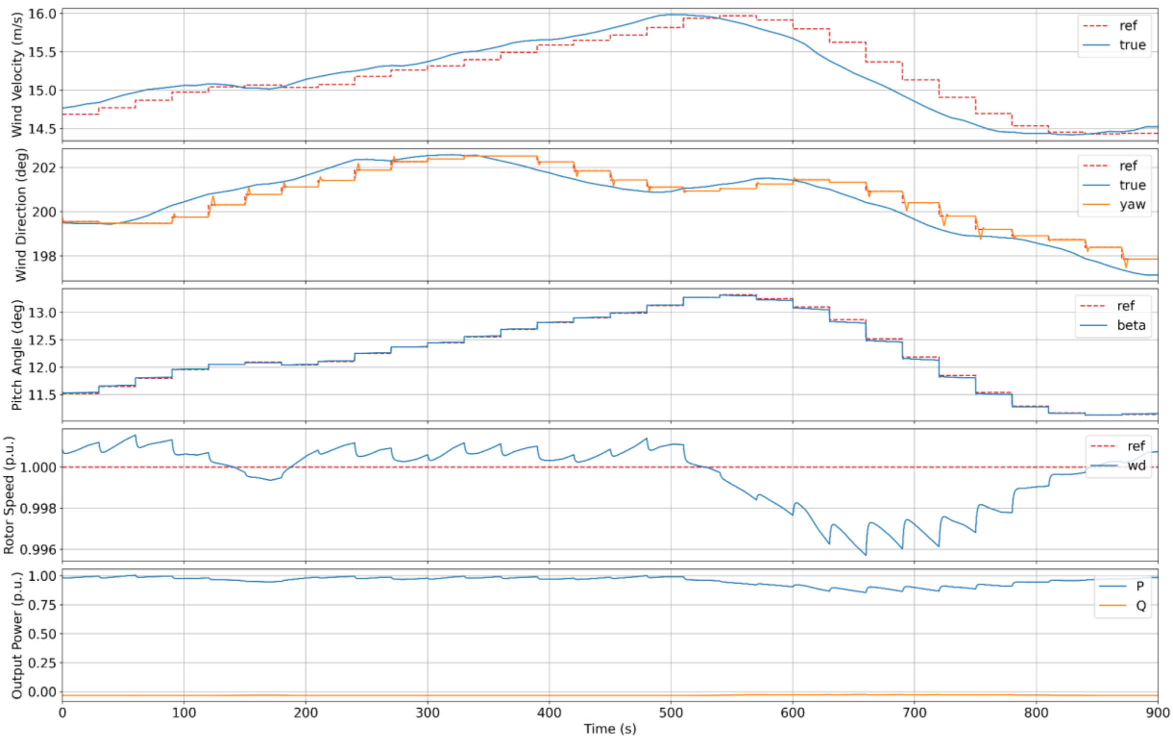

Figure 10.9a FAST control

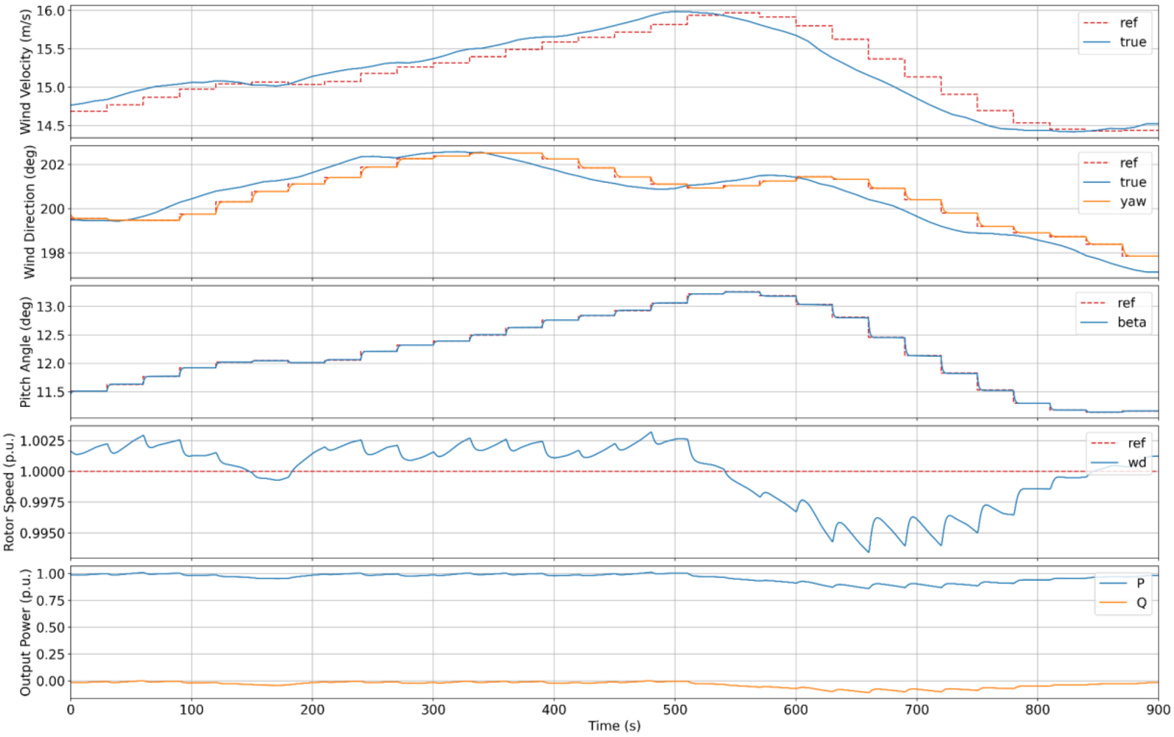

Figure 10.9b Disable wind forecasting

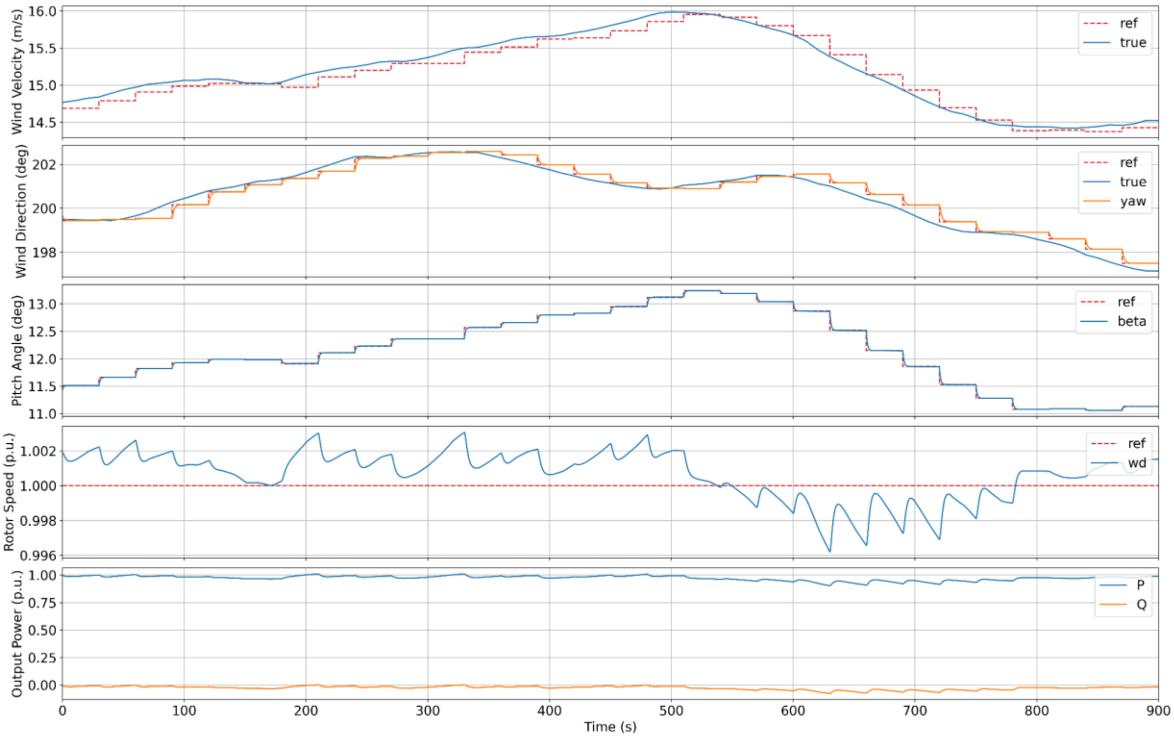

Figure 10.9c Enable wind forecasting

Figure 10.9 Scenario 1, 100% power production in high wind velocities

When lowering the power target to 50%, both rotor speeds in **Figure 10.10** fluctuate less because a lower target reduces the rotor load. Thus, the MPC can more easily handle fluctuation. Like the 100% case, the forecasted control performs better in stabilizing rotation. Regarding active power, the lowest output of the non-forecasted almost touches 0.4 p.u., which is farther than the forecasted.

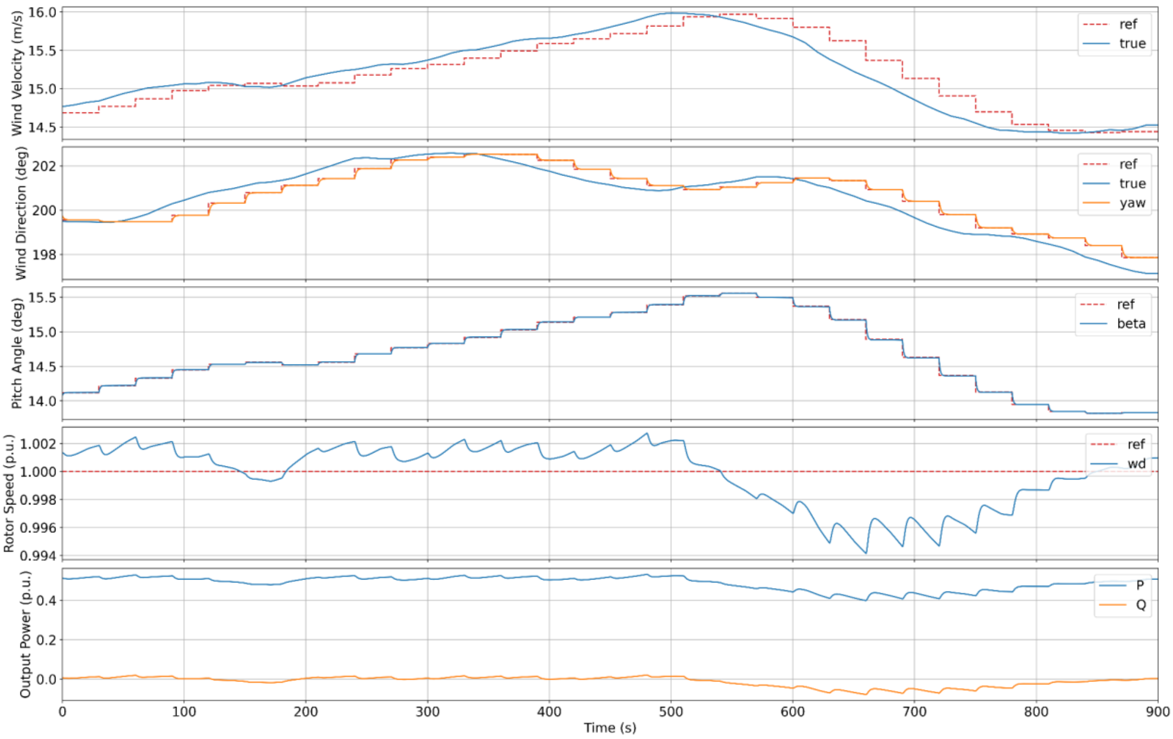

Figure 10.10a Disable wind forecasting

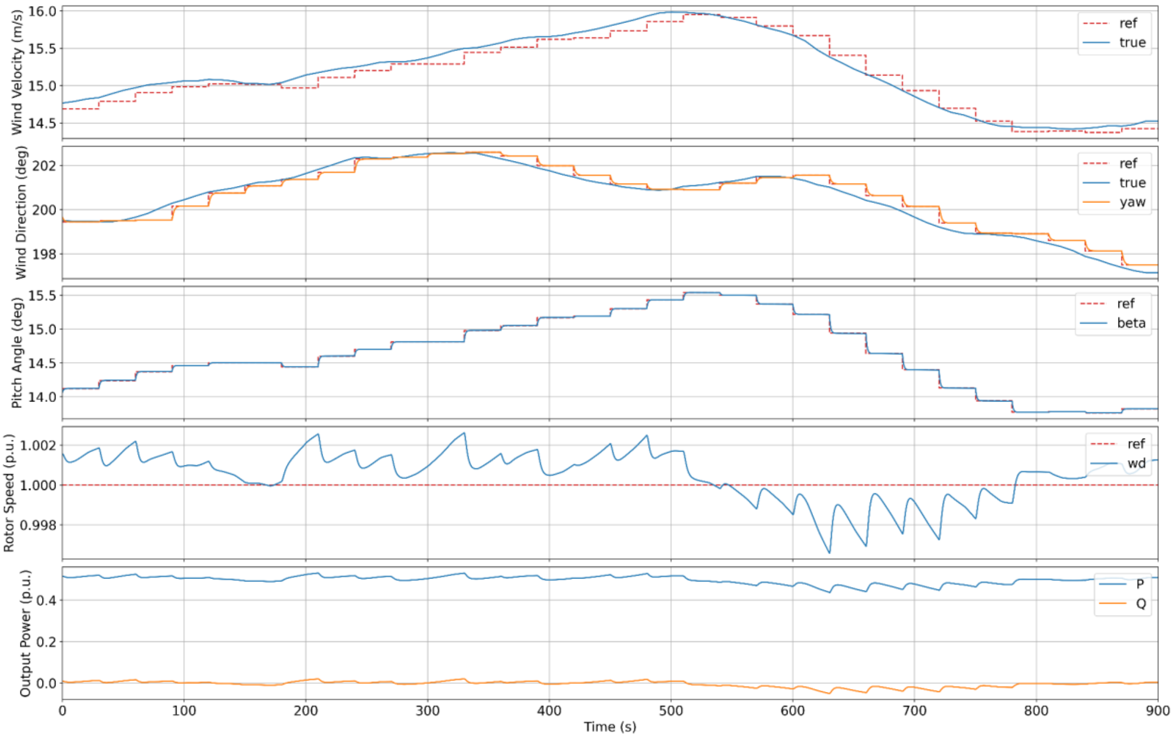

Figure 10.10b Enable wind forecasting

Figure 10.10 Scenario 2, 50% power production in high wind velocities

10.3.4 Medium-velocity Scenario

As given in **Figure 9.5**, wind distribution concentrates on medium wind speeds. Thus, control performance under medium winds is more important. Besides, medium-velocity scenarios often require speed regulation, so a control system faces a challenge in smoothing power production. Hence, this section focuses more on power variation during rotor kinetic transition.

According to **Figure 10.11**, the forecasted speed and direction closely track the upcoming wind. However, the non-forecasted wind has obvious mismatches, e.g., wind speed before 210 s and wind direction in 420~720 s. Due to the variable speed operation in Region ii, there are conspicuous power spikes before 210 s and after 750 s. It is hard for a turbine to eliminate these spikes because rotor transition and power smoothness are contradictory. A fast rotor regulation requires the PMSG to change the electromagnetic torque immediately, resulting in a power spike. It is helpful to place a circuit filter or energy storage [53] to absorb these spikes.

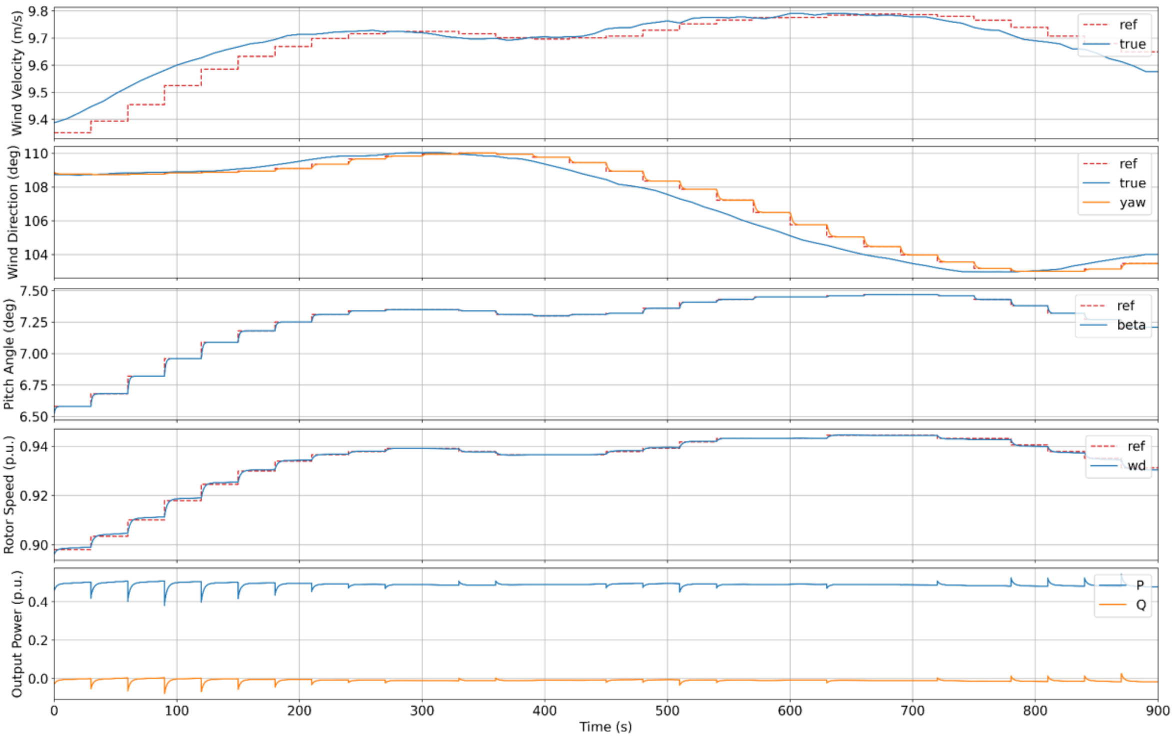

Figure 10.11a Disable wind forecasting

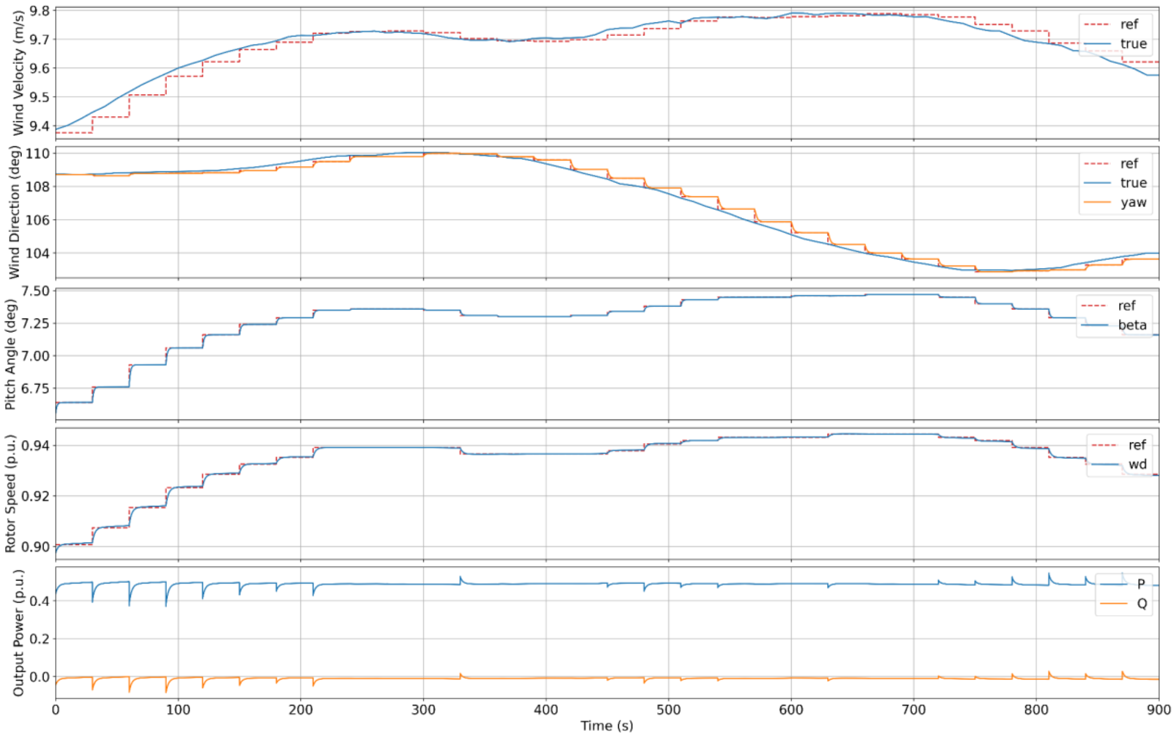

Figure 10.11b Enable wind forecasting

Figure 10.11 Scenario 3, 50% power production in medium wind velocities

Figure 10.12 decreases the output target to 25% for the same wind series in **Figure 10.11**. Although non-forecasted wind reference drifts away from measurement, both controls' rotor speed and output power do not exhibit noteworthy differences compared to the high-velocity cases. The MPC's internal linearization is enough to handle minor aerodynamic variations, so wind forecasting has less effect in medium-velocity cases.

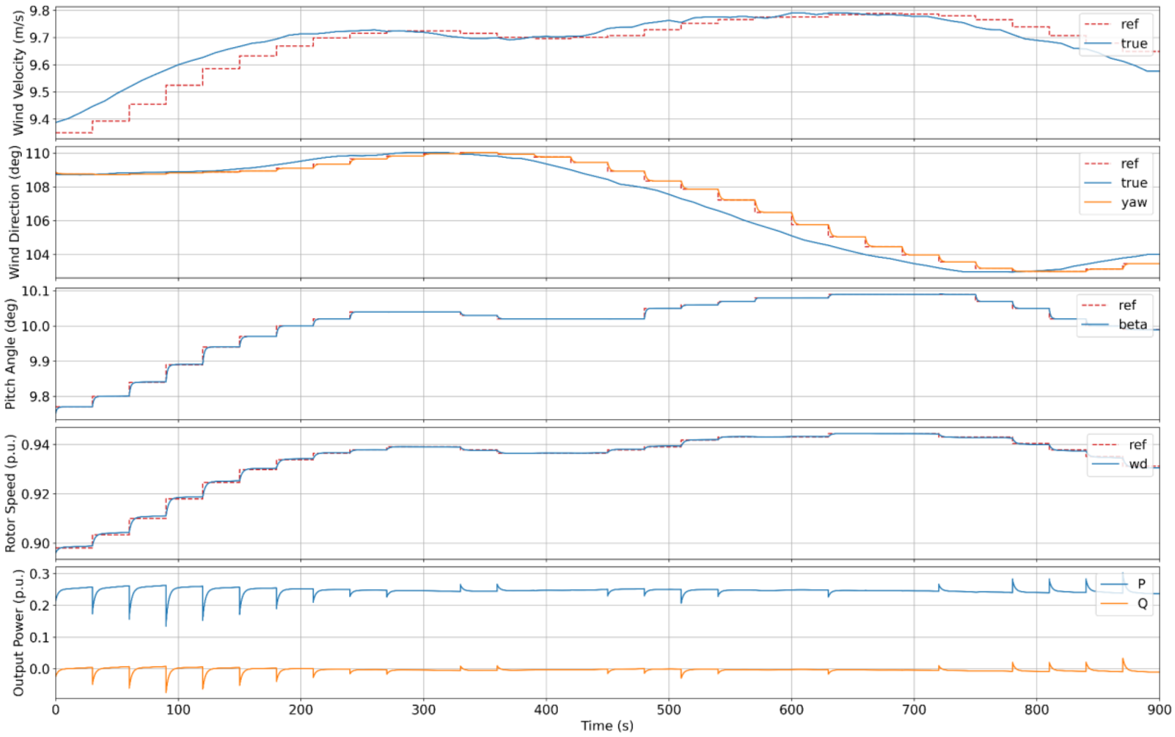

Figure 10.12a Disable wind forecasting

Figure 10.12b Enable wind forecasting

Figure 10.12 Scenario 4, 25% power production in medium wind velocities

Table 10.9 summarizes the statistical results of rotor speed and output power in the four scenarios. Firstly, the FAST ensures stable rotation with minimum speed deviation, which gives rise to more output fluctuation for fast speed response. Compared with the FAST, the MPC can improve an average output of 0.006 p.u., contributing to a higher power efficiency.

With wind forecasting, the power production has a mean promotion of 0.01 p.u., and the rotation and output fluctuation remarkably reduces at least 44% in **scenarios 1~2**. However, **scenarios 3~4** only diminish about 10% of rotational fluctuation but improve nothing for power quality. Wind forecasting is less meaningful in a low-load scenario because the MPC's robustness is sufficient to compensate for wind error. In conclusion, the wind forecasting-enhanced OPS-MPC has advantages in stabilizing rotation and effectively improves power quality at heavy loads.

Table 10.9 Statistics of the forecasted and non-forecasted scenarios

Velocity	Target	Control	Variable	Mean	Std.	Min.	Max.
High	100% **scenario 1**	FAST	ω_d (p.u.)	0.9997	0.0015	0.9957	1.0016
		Figure 10.9a	P_n (p.u.)	0.9568	0.0363	0.8555	1.0052
		non-forecasted	ω_d (p.u.)	0.9999	0.0025	0.9934	1.0032
		Figure 10.9b	P_n (p.u.)	0.9628	0.0386	0.8609	1.0136
		forecasted	ω_d (p.u.)	1.0006	0.0014	0.9962	1.0031
		Figure 10.9c	P_n (p.u.)	0.9739	0.0214	0.9036	1.0113
	50% **scenario 2**	non-forecasted	ω_d (p.u.)	0.9999	0.0022	0.9942	1.0027
		Figure 10.10a	P_n (p.u.)	0.4880	0.0337	0.3982	0.5319
		forecasted	ω_d (p.u.)	1.0005	0.0012	0.9966	1.0026
		Figure 10.10b	P_n (p.u.)	0.4977	0.0187	0.4354	0.5304
Medium	50% **scenario 3**	non-forecasted	ω_d (p.u.)	0.9344	0.0116	0.8961	0.9445
		Figure 10.11a	P_n (p.u.)	0.4897	0.0089	0.3804	0.5434
		forecasted	ω_d (p.u.)	0.9349	0.0105	0.8973	0.9445
		Figure 10.11b	P_n (p.u.)	0.4888	0.0087	0.3715	0.5466
	25% **scenario 4**	non-forecasted	ω_d (p.u.)	0.9344	0.0117	0.8960	0.9445
		Figure 10.12a	P_n (p.u.)	0.2472	0.0087	0.1350	0.3024
		forecasted	ω_d (p.u.)	0.9349	0.0105	0.8972	0.9445
		Figure 10.12b	P_n (p.u.)	0.2464	0.0088	0.1272	0.3063

Note: $P_n = P_s$, the PMSG has no rotor consumption.

10.3.5 Summary

This study proposes a wind forecasting-enhanced OPS-MPC that uses wind prediction to optimize control configuration and integrates the generator and servo control for constant output. The proposed wind forecasting achieves the prediction of wind direction series on the foundation of the conventional wind speed model. With a wind vector series model, the power strategy interprets velocity and direction predictions to a 3-DOF working point for demand tracking. The forecasting section comprehensively investigates feature analysis, compass-vector transformation, series windowing, learning-shaping structure, and deep network optimization. The model evaluation indicates that the convolutional or recurrent network surpasses the simple layer stack, and the recurrent layer is more efficient, which accounts for a model ranking (LSTM≥CNN-LSTM>CNN>DNN). The average accuracy of the LSTM reaches an R^2 of at least 0.9973. This trustworthy performance can help the OPS-MPC perceive wind stochastic variation in advance.

The OPS-MPC coordinates the PMSG, pitch servo, and yaw servo for the 3-DOF regulation of rotor, pitch, and yaw, which enhances the IEA 15-MW for flexible and reliable offshore generation. Proven by the baseline comparison, the MPC with the state-of-the-art QP solver and sparse-matrix QP construction has a persuasive control performance of multiple objectives. Also, four generation scenarios verify that wind forecasting strengthens rotation stability and power smoothness. In particular, wind forecasting can lower about 44% of rotor speed and output power oscillations at high wind speed. Integrating wind series forecasting and 3-DOF turbine control reinforces wind energy's controllability and promotes wind power's stability.

10.4 Reinforcement Learning

This case investigates the performance of the RL-based OPS. This case assumes an unyawed condition and uses the MPC framework to execute the OPS command. The first part trains the DQN and C51 agents and discusses their performance regarding the OPS solution. The second part examines the energy efficiency of the RL-based OPS for maximum output. The last part verifies the reliability of the RL-MPC in outputting desired power. **Table 10.10** lists two RL-MPCs and two competitors in the simulation.

Table 10.10 Summary of four turbine controllers

Name	Power Strategy	Controller	Comment
FAST	MPPT-based MPS	model-free PID	maximum capture only
PRPT	PRPT-based OPS	model-based MPC	able to respond to power command
DQN	RL-based OPS	same as the above	same as the above
C51	same as the above	same as the above	same as the above

10.4.1 Agent Comparison

Each RL training episode randomly initializes the power target and environment state to explore the operation of the IEA 15-MW. **Figure 10.13** records the training history of the DQN and C51. The training loss plays a role in assessing convergency, while the evaluation monitors whether training is abnormal or fails. The training loss is the element-wise-squared loss [153] recommended in the TF-Agents tutorial. The evaluation is the median error of 10 random tests in the evaluation environment.

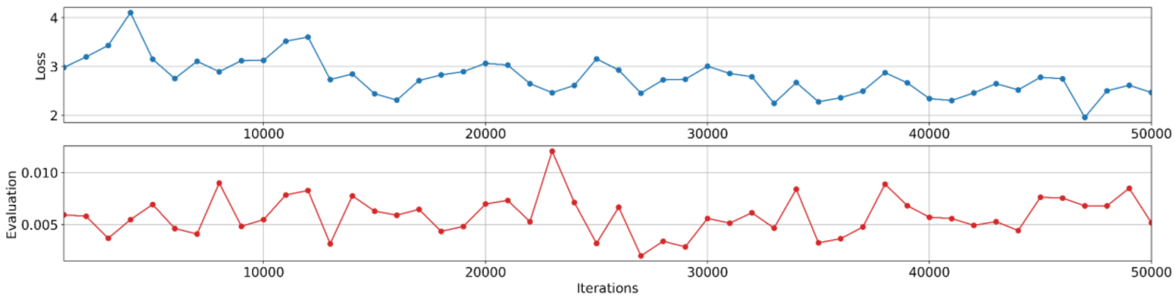

Figure 10.13a DQN training history

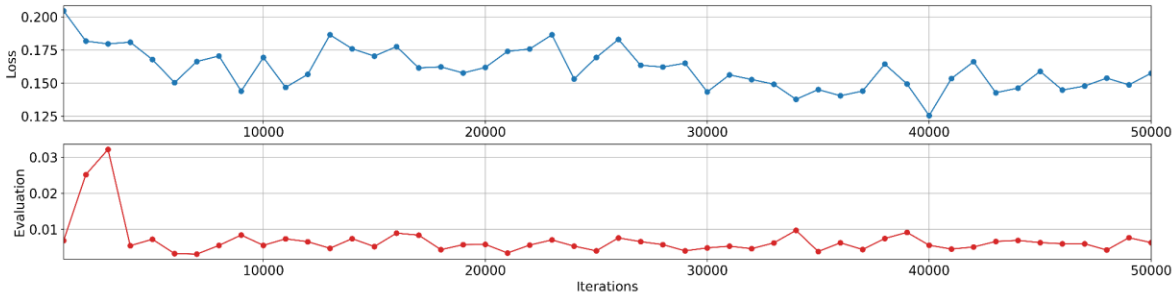

Figure 10.13b C51 training history

Figure 10.13 Training loss and evaluation of the DQN and C51

Since each training episode randomly sets the agent's initial state and final target, the training loss always has a certain degree of fluctuation. However, when an agent gains enough experience, it can find the optimal path from the starting point to the destination. Thus, the training loss will finally fall into a small fluctuation region. According to **Figure 10.13**, an RL training of at least 30,000 iterations can ensure a successful convergence and collect adequate experience. For better training efficiency, a termination criterion stops an episode if the error is less than 0.01 p.u. because the bisected action only needs the agent to locate the vicinity of an optimum. Compared with the DQN, the C51 has more aggressive attempts before 3,000 iterations but quickly converges to a steady RL policy afterwards.

Figure 10.14 shows the solving iteration and error distribution of the DQN and C51 on 300 random tests. It notes that oversized targets account for most large errors due to random trials. The DQN and C51 can find the OPS solution within 75 iterations and solve 84.7% of cases with negligible error. For the remaining 15.3% of cases, the agent aims for maximum capture, which leads to similar error distributions.

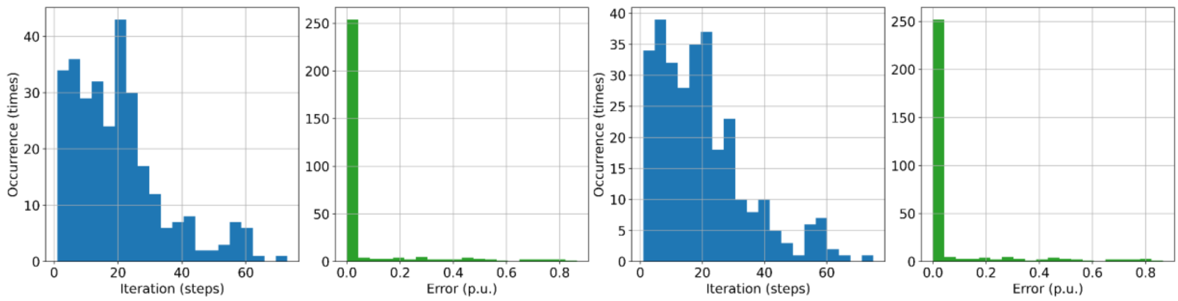

Figure 10.14a DQN test results **Figure 10.14b C51 test results**

Figure 10.14 Comparison of the DQN and C51 on 300 random tests

Figure 10.15 shows examples of the calculation contour of the RL-based OPS. Although the training of the DQN and C51 are in the same environment, they evolve into different decision-making behaviours. The DQN tends to find a unidirectional path to reach the final result, while the C51 prefers to travel a broad region and even obviously turns back on rotor speed, as shown in **Figure 10.15d**. Hence, the DQN has a stronger memory for experience, and the C51 seems greedy to explore. The presented examples achieve a solving accuracy of over 98%, which proves the effectiveness of the FIFO-based bisection algorithm (**Algorithm 4**). It notes that the 2-DOF contour is the calculation process of the OPS, and the control system only takes the final state to update reference states and control parameters.

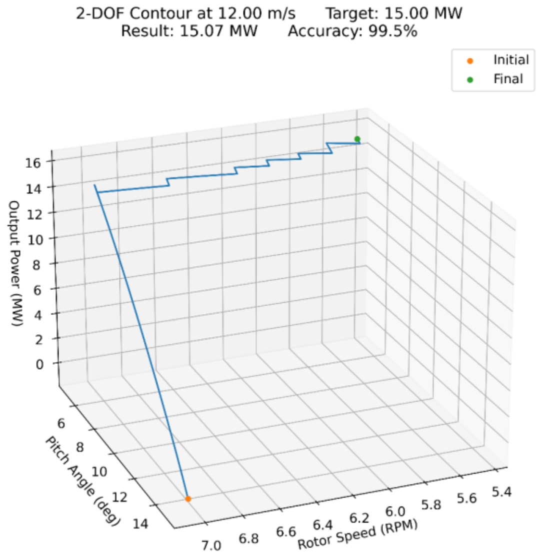

Figure 10.15a DQN, targeting 100% capacity

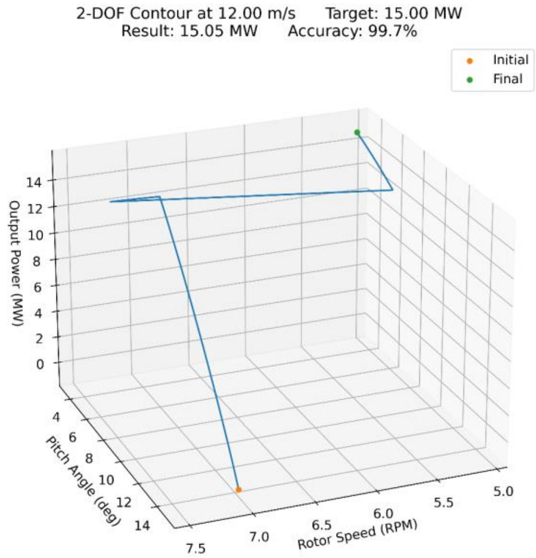

Figure 10.15b C51, targeting 100% capacity

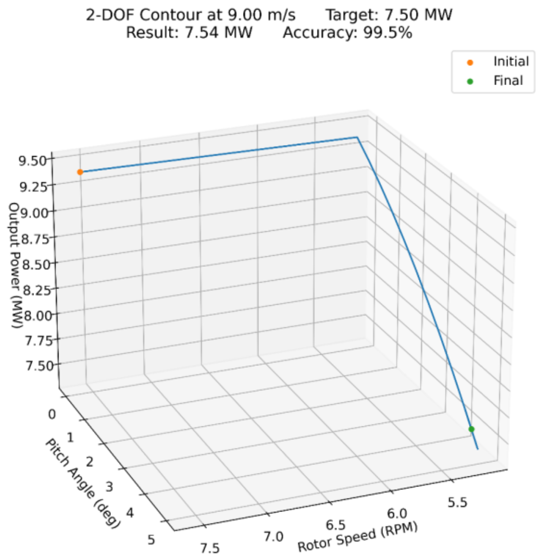

Figure 10.15c DQN, targeting 50% capacity

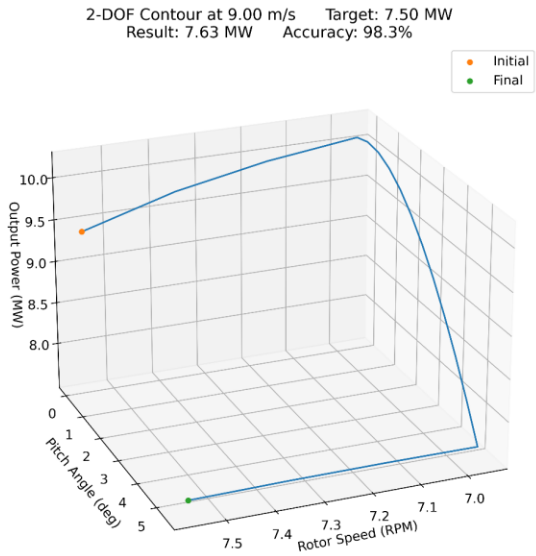

Figure 10.15d C51, targeting 50% capacity

Figure 10.15 Examples of the contour of the RL-based OPS with the eager policy

10.4.2 Maximum Power Simulation

Since maximum power evaluates the capability of wind energy conversion, examining the output potential of the RL-based OPS is necessary. This section sets the power target of the PRPT, DQN, and C51 to rated power. **Figure 10.16** exhibits a simulation of four controllers in Region ii, in which the system updates the reference state per 20-s. Firstly, different power strategies do not offer the same references but can ensure the same steady-state power at each period end. This evidence indicates that the RL-based OPS can secure maximum capture and energy efficiency. Secondly, the MPC has advantages in output stability over the

FAST, considering the FAST has noticeable active power spikes at 40, 60, 140, 160, and 180 seconds.

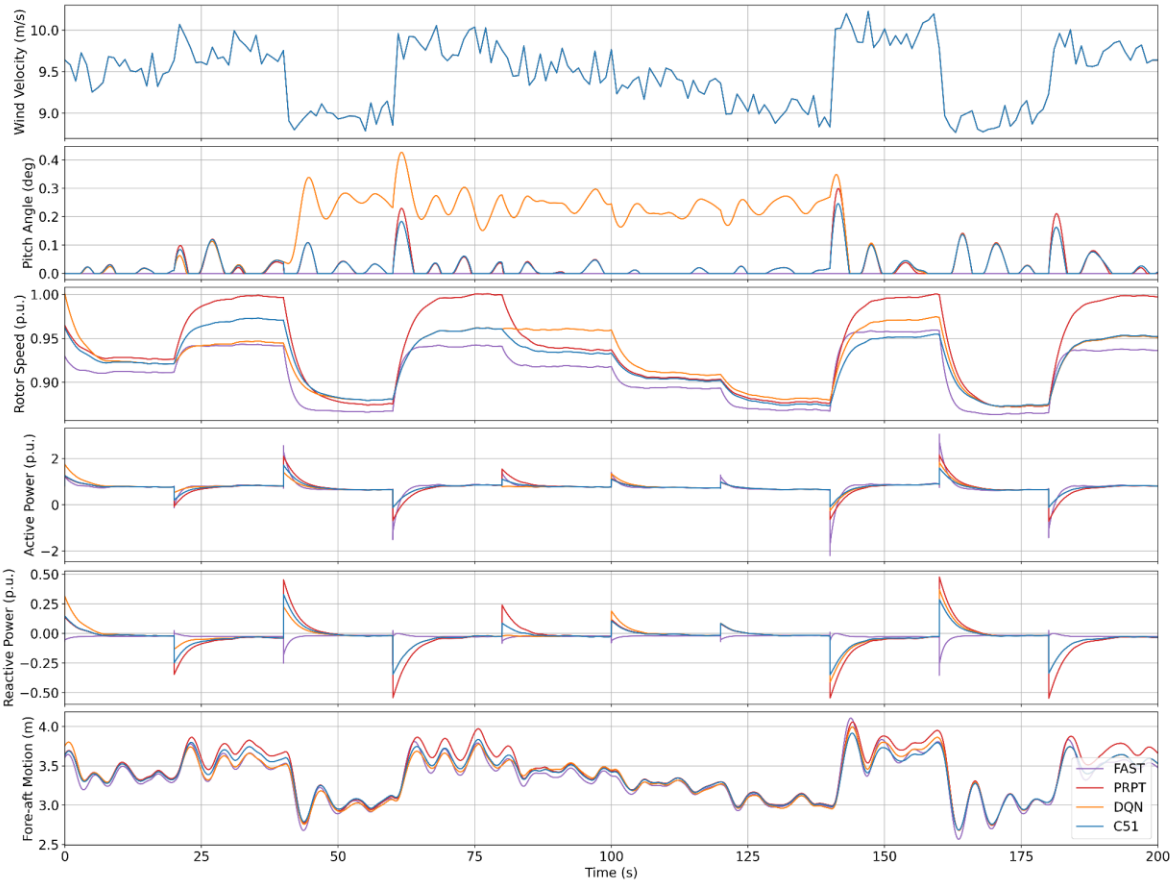

Figure 10.16 Simulation of four controllers in Region ii

Figure 10.17 compares four controllers with a random wind series in Region iii. The FAST, PRPT, and DQN identify rated rotor speed as optimal, while the C51 determines multiple speed references in Region iii. Since the C51 agent models the probability distribution of the Q-value, the C51 tends to regulate rotation in a small range. Therefore, the C51 causes more output spikes because each rotor speed transition needs the PMSG to absorb the variation of kinetic energy. Although four controllers have different 2-DOF references, they all ensure rated output. Considering Regions ii and iii, the proposed RL-based OPS is reliable for maximum capture in Regions ii and iii.

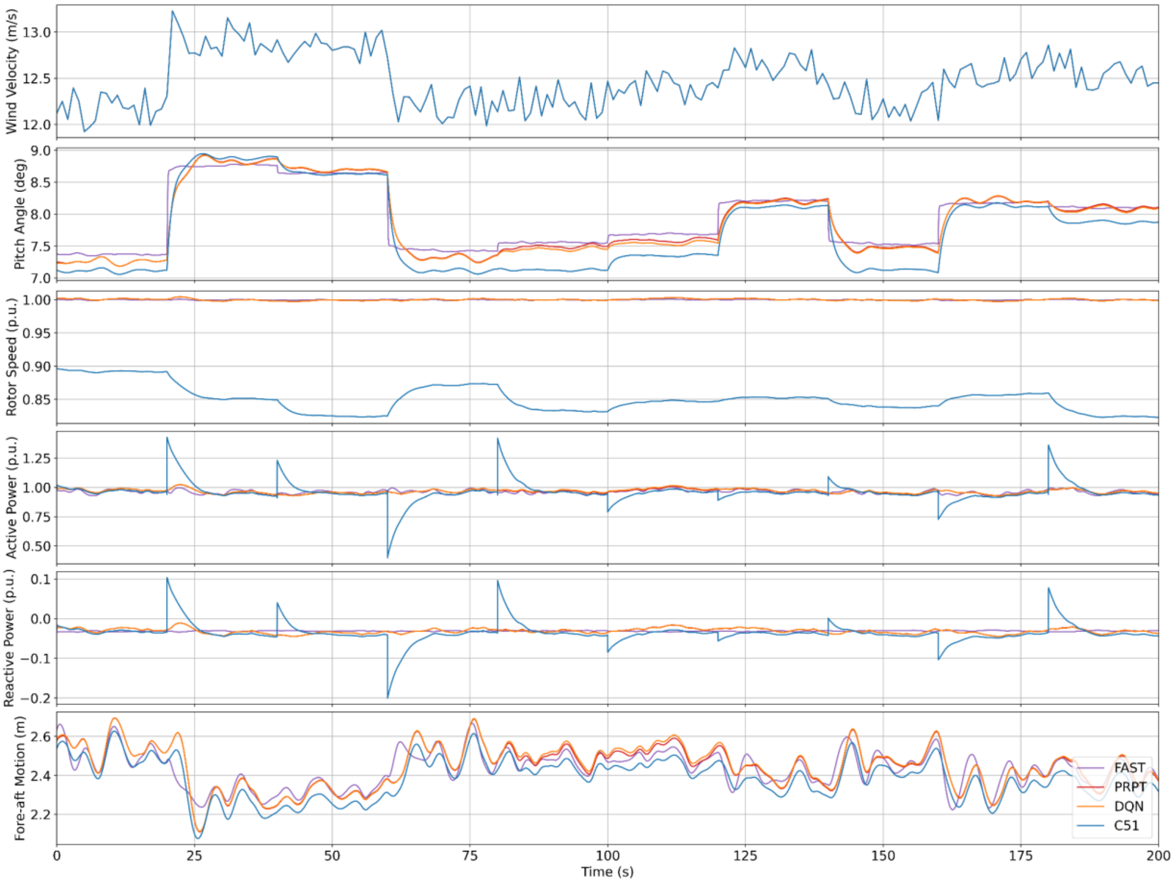

Figure 10.17 Simulation of four controllers in Region iii

Table 10.11 compares the statistical results of turbine performance in **Figure 10.16** and **Figure 10.17**. Since the FAST sets stable rotation as the main objective, it has the maximum power range (-2.193~3.056 p.u.) in **Figure 10.16**, and these transient overloads probably damage the generator. However, the MPC can weigh multiple objectives and predict corresponding trajectories, which avoids excessive generator regulation. Compared to the MPPT and PRPT, the RL methods realize higher power quality in Region ii, considering power extraction, fluctuation, and magnitude. Regarding Region iii, the FAST, PRPT, and DQN do not exhibit apparent differences. The C51 reaches the same average power level but has the largest fluctuation (0.086 p.u.). In addition, the MPC can produce effective tower fore-aft damping in Region ii because of the supplementary model of tower motion. However, four controls result in different fore-aft damping in Region iii due to different 2-DOF reference states. In conclusion, the DQN surpasses the other three on multi-objective balance and power quality for maximum wind capture.

Table 10.11 Comparison of four controls in Regions ii and iii

		Figure 10.16				Figure 10.17			
		Mean	**Std.**	**Min.**	**Max.**	**Mean**	**Std.**	**Min.**	**Max.**
FAST	ω_d (p.u.)	0.910	0.031	0.864	0.960	1.000	0.001	0.999	1.001
	P_n (p.u.)	0.762	0.266	-2.193	3.056	0.962	0.017	0.928	1.004
	x_m (m)	3.330	0.274	2.565	4.109	2.431	0.090	2.224	2.667
PRPT	ω_d (p.u.)	0.938	0.046	0.872	1.001	1.000	0.001	0.997	1.005
	P_n (p.u.)	0.755	0.277	-0.712	2.162	0.967	0.018	0.928	1.023
	x_m (m)	3.410	0.307	2.677	4.058	2.443	0.108	2.111	2.695
DQN	ω_d (p.u.)	0.926	0.032	0.872	1.001	1.000	0.001	0.997	1.005
	P_n (p.u.)	0.773	0.195	-0.256	1.843	0.968	0.019	0.927	1.024
	x_m (m)	3.351	0.270	2.680	3.995	2.448	0.110	2.113	2.694
C51	ω_d (p.u.)	0.923	0.032	0.873	0.973	0.850	0.020	0.823	0.897
	P_n (p.u.)	0.766	0.179	-0.124	1.729	0.962	0.086	0.399	1.429
	x_m (m)	3.363	0.261	2.683	3.916	2.384	0.103	2.076	2.628

Note: $P_n = P_s$, the PMSG has no rotor consumption.

10.4.3 Constant Power Simulation

Constant power output is more valuable than maximum output, considering power dispatching and grid safety. This section excludes the FAST since it cannot react to any power command. **Figure 10.18** sets a power target of 0.5 p.u. for the PRPT, DQN, and C51. The PRPT follows a similar rule of the optimal TSR, i.e., the PRPT-based OPS always changes rotor speed according to wind speed in Region ii, which accounts for significant power transitions. The DQN and C51 have fewer output fluctuations than the PRPT, which benefits from the RL reward prioritizing pitch regulation.

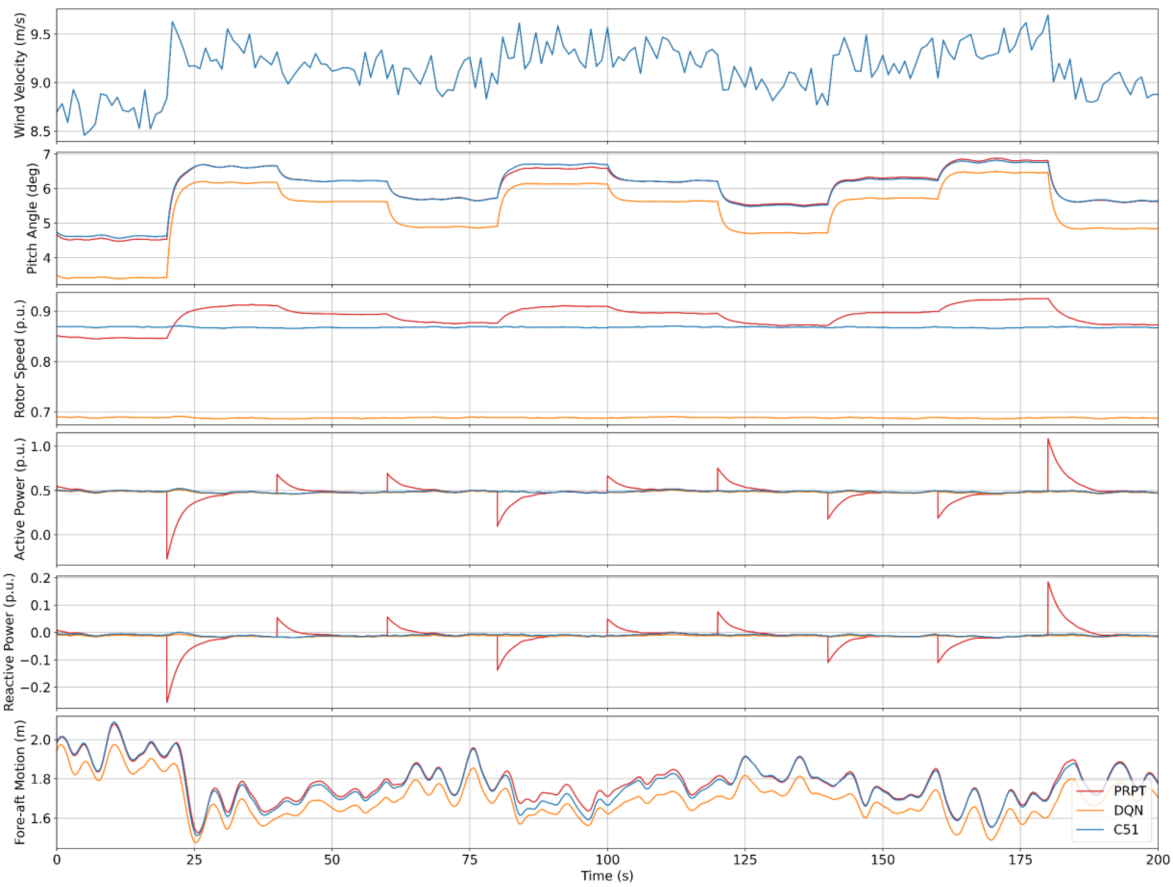

Figure 10.18 Simulation of the PRPT, DQN, and C51 targeting an output of 50%

Figure 10.19 uses the same wind series but changes the power target to 25%. Firstly, the speed regulation of the PRPT is almost the same as that of the 50% case due to its RSC policy. Secondly, the DQN-MPC adjusts rotor speed from 0.688 p.u. to 0.742 p.u. and maintains this speed. Thirdly, the C51 also changes its rotation to approach the target, but its speed variation (0.008 p.u.) is less than that of the PRPT (0.021 p.u.). **Figure 10.18** and **Figure 10.19** indicate that the DQN learns to put a speed reference to cover broader generation scenarios, which accounts for less generator regulation and more stable output.

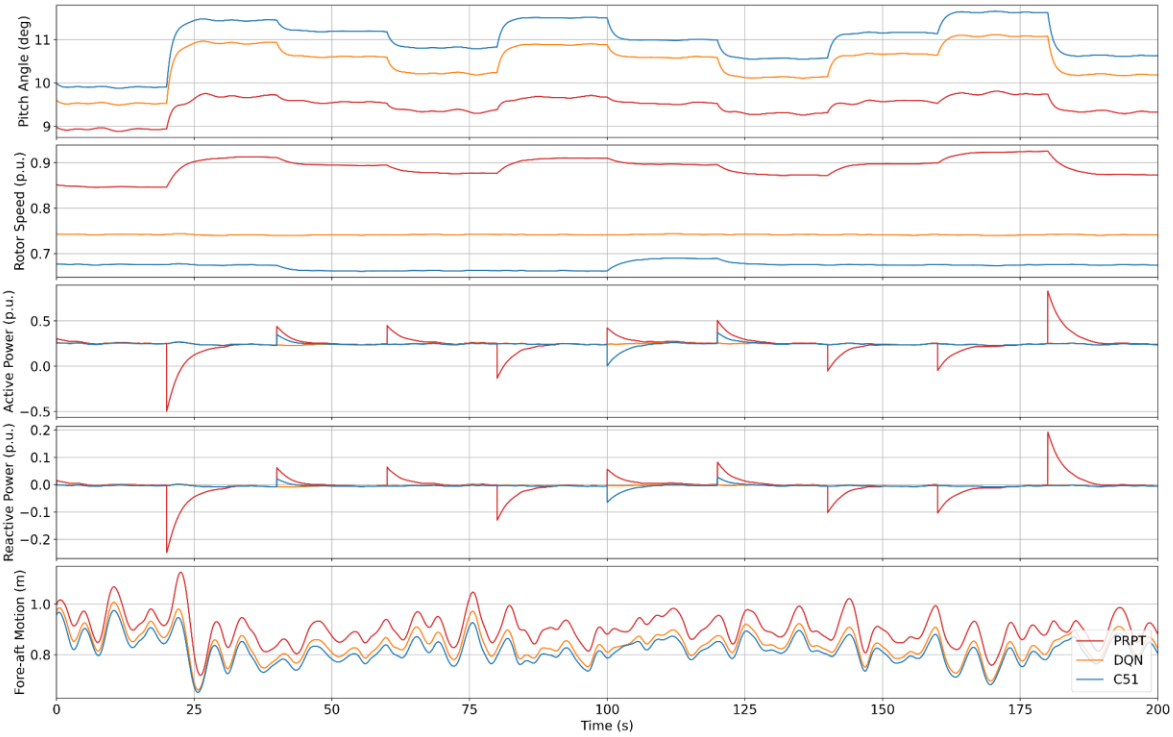

Figure 10.19 Simulation with the previous wind series for a target of 25%

Table 10.12 Comparison of tracking 50% and 25% power targets

		Figure 10.18				Figure 10.19			
		Mean	**Std.**	**Min.**	**Max.**	**Mean**	**Std.**	**Min.**	**Max.**
PRPT	ω_d (p.u.)	0.890	0.021	0.845	0.925	0.890	0.021	0.846	0.925
	P_n (p.u.)	0.484	0.100	-0.274	1.087	0.244	0.099	-0.497	0.831
	x_m (m)	1.780	0.096	1.526	2.079	0.904	0.058	0.719	1.126
DQN	ω_d (p.u.)	0.688	0.001	0.686	0.691	0.742	0.001	0.740	0.744
	P_n (p.u.)	0.481	0.009	0.460	0.511	0.246	0.008	0.229	0.269
	x_m (m)	1.695	0.093	1.475	1.975	0.838	0.054	0.661	1.008
C51	ω_d (p.u.)	0.868	0.001	0.866	0.871	0.673	0.008	0.661	0.690
	P_n (p.u.)	0.488	0.010	0.463	0.523	0.245	0.024	0.004	0.372
	x_m (m)	1.768	0.103	1.511	2.089	0.815	0.051	0.651	0.975

Table 10.12 compares the performance of constant output in **Figure 10.18** and **Figure 10.19**. Compared with the PRPT, the DQN and C51 contribute to less power and rotation fluctuations owing to the agent experience on the 2-DOF solution. The DQN and C51 identify trade-off speed references, so they do not need to alter the PMSG load frequently. However, the PRPT follows a unique speed curve, so each wind change affects the PMSG to accelerate or decelerate. In the 50% case, the RL can reduce power oscillation by about 90%. In the 25% case, the DQN and C51 can lessen it by 92% and 75%, respectively. Therefore, the RL-based OPS is more intelligent in the 2-DOF regulation of speed and pitch.

Additionally, compared with the probability modelling of the C51, the straightforward training of the DQN is more effective in deciding the optimal reference state.

10.4.4 Summary

An intelligent turbine control under an RL-based OPS consists of two components. Firstly, an ANN-based aerodynamic model provides necessary thrust, torque, and power predictions, which supports convenient iteration for the RL eager policy and local linearization for the control model. Secondly, an RL-based 2-DOF optimization intelligently decides the reference of the system state by which a turbine can actively respond to power command. Meanwhile, a novel FIFO-based bisection algorithm (**Algorithm 4**) for discrete agent actions improves the accuracy of the 2-DOF solution.

The simulation proves the effectiveness and reliability of the proposed RL-MPC. Firstly, the RL-based OPS has the same capability to capture maximum power as the MPPT and PRPT. Secondly, compared with the PRPT-MPC, the RL-MPC significantly reduces power fluctuation for constant output. Thirdly, regarding agent selection, the DQN that directly models the solving experience yields better 2-DOF answers than the C51 for turbine operation. Hence, the proposed DQN-MPC can contribute to more flexible wind generation and higher power quality. Also, the hybrid RL framework can help a turbine decide control objectives and optimize control parameters, which offers a pathway to next-generation intelligent WECSs.

Chapter 11 Conclusion

As wind energy has an increasing share of the electricity market, modern WTs are also growing in capacity. However, the single-output mode of present turbines challenges power dispatching for demand-generation equilibrium. In other words, the conventional turbine control for maximum capture cannot support the upcoming 100% renewable electricity. Hence, new turbine control for flexible wind generation is urgently needed.

Previous turbine control designs only look for maximum load due to aerodynamic complexity. With the development of computer science, ML brings many revolutionary tools to address such problems and optimizations related to aerodynamics. This study develops a novel control framework that combines ML and control engineering to achieve adjustable turbine output to suit demand-oriented generation. This framework includes an ANN that offers quick aerodynamic calculations, an OPS algorithm that solves online optimization for power tracking, a terminal controller that governs the turbine system, and a wind model that predicts wind series according to historical data.

First, this study proposes three ANN-based aerodynamic models (RBFN, DNN, and HDNN) predicting thrust, torque, and power from wind speed, rotor speed, and blade pitch. Second, a PRPT-based OPS algorithm can solve the 2-DOF reference of the RSC and PAC on an aerodynamic model to track a power target. Besides, the OPS provides beneficial thrust and torque sensitivities for optimal control configuration. In addition to the PRPT-based OPS, this study also develops an RL-based OPS for the weighted 2-DOF solution, which can assign the RSC and PAC with different priorities.

This study designs four controllers to implement the OPS, including PID, LQR, RHC, and MPC. The PID is a typical model-free controller without any trajectory estimation. By comparison, the other three are model-based controllers with a built-in control model to predict system trajectory on the control horizon. In addition, the OPS can receive velocity and direction predictions from wind forecasting to reduce the effects of wind uncertainties on control. The velocity prediction assists in solving the 2-DOF reference of the RSC and

PAC, and the wrapped direction prediction straightly governs the YAC. Therefore, the OPS achieves a complete 3-DOF regulation of rotor, pitch, and yaw.

Straightforwardly, the involved ML methods, including aerodynamics models, decision-making agents, and wind models, collaborate to find the solution for outputting the desired power. Meanwhile, the local linearization inside the OPS updates thrust and torque sensitivities for the controller to handle aerodynamic nonlinearities for optimal dynamic response. After the OPS determines overall regulation objectives and configures necessary parameters, a specific controller executes the OPS command for state transition and disturbance suppression.

In addition to the novel designs, this study develops a comprehensive simulator for variable WTs. Also, this research runs a series of simulations to test the accuracy and reliability of the OPS. The conclusions based on the case summaries include:

- The PRPT-based OPS can ensure the same maximum output as the MPPT-based MPS. Also, this OPS reaches an average output accuracy of over 99%.
- Model-based control can achieve better multi-objective regulation owing to the small-signal control model. The MPC is preferable due to better dynamics.
- The HDNN has better aerodynamic predictions than the RBFN and DNN regarding error distribution and control accuracy.
- The LSTM and CNN-LSTM ensure more accurate results than the DNN and CNN in wind vector series forecasting. Given fewer network parameters, the LSTM is more economical in training and deploying than the CNN-LSTM.
- Wind forecasting notably reduces at least 44% of rotation and output fluctuations for the 3-DOF OPS-based turbine, which enhances system stability and power quality.
- The RL-based OPS contributes to a more intelligent 2-DOF solution of the RSC and PAC than the PRPT-based OPS, which reduces power spikes caused by rotor state transition. Furthermore, the DQN is superior to the C51 in the 2-DOF decision-making process.

Although our innovative methods bring attractive features, there are some disadvantages and limitations that may cause control failure:

- The aerodynamic solver has a mandatory prerequisite, i.e., a blade must follow the classic BEM theories. Thus, the developed simulator only suits classic BEM blades. Meanwhile, the BEM solver affects synthetic data for ANN-based modelling and RL agent training. Therefore, ML training should consider other simulated or experimental methods if a blade has sophisticated geometry.
- Since training data for aerodynamic modelling are synthetic, they cannot avoid some mismatches with actual measurements. Similarly, the RL environment cannot entirely replace real-world turbine operation. Hence, data collection and environment setup may require wind tunnel results for correction and reconciliation.
- ML cannot guarantee that a training result will be 100% successful and effective. Besides, random seeds heavily affect ML, so training results are always varied. Hence, an industrial application should train a model or agent several times and conduct a quality check to avoid accidental training faults.

Our extensive analyses and simulations verify the feasibility of the proposed ML-based OPS control framework for flexible and reliable wind generation. Numerical and statistical results prove its effectiveness, accuracy, reliability, and robustness. Further, the proposed control systems provide a feasible technical route for intelligent WTs, which can contribute to the grid integration of wind power and promote the development and application of wind energy for a clean energy society. For better industrial applications, we plan to have the following contents in future work: investigate data fusion between theoretical values and actual measurements for model enhancement; strengthen the learning experience of an RL agent through interaction with real-world turbine operation; add fault diagnostics in our designs to improve fault tolerance; develop the cluster coordination of our turbine control for upgrade to wind farm control.

References

[1] H. Holttinen and B. H. Jørgensen, "IEA Wind TCP Annual Report 2022," Oct 2023.

[2] Equinor. "World's first floating wind farm has started production." https://www.equinor.com/news/archive/worlds-first-floating-wind-farm-started-production (accessed Apr, 2024).

[3] WindEurope. "How much wind was in Europe's electricity yesterday?" windeurope.org. https://windeurope.org/about-wind/daily-wind/ (accessed Dec, 2022).

[4] P. Bortolotti *et al.*, "IEA Wind TCP Task 37: Systems engineering in wind energy-WP2. 1 Reference wind turbines," National Renewable Energy Lab.(NREL), Golden, CO (United States), 2019.

[5] P. Bórawski, A. Bełdycka-Bórawska, K. J. Jankowski, B. Dubis, and J. W. Dunn, "Development of wind energy market in the European Union," *Renewable Energy,* vol. 161, pp. 691-700, 2020, doi: 10.1016/j.renene.2020.07.081.

[6] S. Yang, K. Lin, and A. Zhou, "An ML-based wind turbine blade design method considering multi-objective aerodynamic similarity and its experimental validation," *Renewable Energy,* vol. 220, 2024, doi: 10.1016/j.renene.2023.119625.

[7] T. Letcher, *Wind energy engineering: A handbook for onshore and offshore wind turbines*. Elsevier, 2023.

[8] V. Yaramasu and B. Wu, *Model predictive control of wind energy conversion systems*. John Wiley & Sons, 2016.

[9] S. Sun, Y. Liu, Q. Li, T. Wang, and F. Chu, "Short-term multi-step wind power forecasting based on spatio-temporal correlations and transformer neural networks," *Energy Conversion and Management,* vol. 283, 2023, doi: 10.1016/j.enconman.2023.116916.

[10] J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind Energy Explained: Theory, Design and Application, Second Edition*. John Wiley & Sons, 2010.

[11] M. O. L. Hansen, *Aerodynamics of Wind Turbines Second Edition*. James & James, 2000.

[12] W. E. T. Office. "Vertical-Axis Wind Turbines Could Reduce Offshore Wind Energy Costs." https://www.energy.gov/eere/wind/articles/vertical-axis-wind-turbines-could-reduce-offshore-wind-energy-costs (accessed May, 2024).

[13] E. Hau, second, Ed. *Wind Turbines Fundamentals, Technologies, Application, Economics*. Berlin, Heidelberg: Springer, 2006.

[14] D. Zhao, N. Han, E. Goh, J. Cater, and A. Reinecke, *Wind turbines and aerodynamics energy harvesters*. Academic Press, 2019.

[15] I. W. TCP. "Task 40 Work Plan and Objectives." https://iea-wind.org/task40/t40-work-plan-and-objectives/ (accessed Apr, 2024).

[16] S. Boersma *et al.*, "A tutorial on control-oriented modeling and control of wind farms," in *2017 American control conference (ACC)*, 2017: IEEE, pp. 1-18.

[17] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Wind energy handbook*. John Wiley & Sons, 2011.

[18] S. Heier, *Grid Integration of Wind Energy*. John Wiley & Sons, 2014.

[19] G. Abad, J. s. Lo´pez, M. A. Rodrı´guez, L. Marroyo, and G. Iwanski, *Doubly Fed*

Induction Machine Modeling and Control for Wind Energy Generation Applications. Hoboken, New Jersey: John Wiley & Sons, 2011.

[20] W. Du, H. Wang, and S. Bu, *Small-signal stability analysis of power systems integrated with variable speed wind generators*. Springer, 2018.

[21] V. D. Colli, F. Marignetti, and C. Attaianese, "Feasibility of a 10 MW doubly fed induction generator for direct-drive wind turbines," in *2009 IEEE PES/IAS Conference on Sustainable Alternative Energy (SAE)*, 2009: IEEE, pp. 1-5.

[22] E. Review. "Why are wind turbines being switched off?" https://electricalreview.co.uk/2023/09/06/why-are-wind-turbines-switched-off/ (accessed Sep, 2024).

[23] L.-R. Chang-Chien, C.-C. Sun, and Y.-J. Yeh, "Modeling of Wind Farm Participation in AGC," *IEEE Transactions on Power Systems,* vol. 29, no. 3, pp. 1204-1211, 2014, doi: 10.1109/tpwrs.2013.2291397.

[24] A. Hwas and R. Katebi, "Wind turbine control using PI pitch angle controller," *IFAC Proceedings Volumes,* vol. 45, no. 3, pp. 241-246, 2012.

[25] M. Rahimi and M. Asadi, "Control and dynamic response analysis of full converter wind turbines with squirrel cage induction generators considering pitch control and drive train dynamics," *International Journal of Electrical Power & Energy Systems,* vol. 108, pp. 280-292, 2019, doi: 10.1016/j.ijepes.2019.01.018.

[26] A. Gambier and F. Meng, "Control system design for a 20 MW reference wind turbine," in *2019 IEEE Conference on Control Technology and Applications (CCTA)*, 2019: IEEE, pp. 258-263.

[27] A. R. Junejo, N. U. Gilal, and J. Doh, "Physics-informed optimization of robust control system to enhance power efficiency of renewable energy: Application to wind turbine," *Energy,* vol. 263, 2023, doi: 10.1016/j.energy.2022.125667.

[28] G. Tu, Y. Li, and J. Xiang, "Coordinated Rotor Speed and Pitch Angle Control of Wind Turbines for Accurate and Efficient Frequency Response," *IEEE Transactions on Power Systems,* vol. 37, no. 5, pp. 3566-3576, 2022, doi: 10.1109/tpwrs.2021.3136822.

[29] M. Lara, J. Garrido, M. L. Ruz, and F. Vázquez, "Multi-objective optimization for simultaneously designing active control of tower vibrations and power control in wind turbines," *Energy Reports,* vol. 9, pp. 1637-1650, 2023, doi: 10.1016/j.egyr.2022.12.141.

[30] L. Pan and X. Wang, "Variable pitch control on direct-driven PMSG for offshore wind turbine using Repetitive-TS fuzzy PID control," *Renewable Energy,* vol. 159, pp. 221-237, 2020, doi: 10.1016/j.renene.2020.05.093.

[31] A. Poureh, M. Chamani, and A. Bahri, "Nonlinear analysis of gain scheduled controllers for the NREL 5-MW turbine blade pitch control system," *International Journal of Electrical Power & Energy Systems,* vol. 145, 2023, doi: 10.1016/j.ijepes.2022.108578.

[32] Q. Hawari, T. Kim, C. Ward, and J. Fleming, "A robust gain scheduling method for a PI collective pitch controller of multi-MW onshore wind turbines," *Renewable Energy,* vol. 192, pp. 443-455, 2022, doi: 10.1016/j.renene.2022.04.117.

[33] R. Hu, C. Le, Z. Gao, H. Ding, and P. Zhang, "Implementation and evaluation of control strategies based on an open controller for a 10 MW floating wind turbine,"

Renewable Energy, vol. 179, pp. 1751-1766, 2021, doi: 10.1016/j.renene.2021.07.117.

[34] M. Sahin and I. Yavrucuk, "Adaptive envelope protection control of wind turbines under varying operational conditions," *Energy,* vol. 247, 2022, doi: 10.1016/j.energy.2022.123544.

[35] A. Essadki and T. Nasser, "LQR controller design for a nonlinear, doubly fed induction generator model," in *2018 6th International Renewable and Sustainable Energy Conference (IRSEC)*, 2018: IEEE, pp. 1-6.

[36] H. Camblong, S. Nourdine, I. Vechiu, and G. Tapia, "Control of wind turbines for fatigue loads reduction and contribution to the grid primary frequency regulation," *Energy,* vol. 48, no. 1, pp. 284-291, 2012, doi: 10.1016/j.energy.2012.05.035.

[37] Q. Yao, Y. Hu, H. Deng, Z. Luo, and J. Liu, "Two-degree-of-freedom active power control of megawatt wind turbine considering fatigue load optimization," *Renewable Energy,* vol. 162, pp. 2096-2112, 2020, doi: 10.1016/j.renene.2020.09.137.

[38] A. Routray and S.-H. Hur, "Full Operational Envelope Control of a Wind Turbine Using Model Predictive Control," *IEEE Access,* vol. 10, pp. 121940-121956, 2022, doi: 10.1109/access.2022.3223087.

[39] T. Wakui, A. Nagamura, and R. Yokoyama, "Stabilization of power output and platform motion of a floating offshore wind turbine-generator system using model predictive control based on previewed disturbances," *Renewable Energy,* vol. 173, pp. 105-127, 2021, doi: 10.1016/j.renene.2021.03.112.

[40] Q. Yao, Y. Hu, T. Zhao, Y. Guan, Z. Luo, and J. Liu, "Fatigue load suppression during active power control process in wind farm using dynamic-local-reference DMPC," *Renewable Energy,* vol. 183, pp. 423-434, 2022, doi: 10.1016/j.renene.2021.10.069.

[41] G. Srinivasa Sudharsan, K. Natarajan, S. G. Rahul, and A. Kumar, "Active power control in horizontal axis wind turbine considering the fatigue structural load parameter using psuedo adaptive- model predictive control scheme," *Sustainable Energy Technologies and Assessments,* vol. 57, 2023, doi: 10.1016/j.seta.2023.103166.

[42] Z. Lin, Z. Chen, J. Liu, and Q. Wu, "Coordinated mechanical loads and power optimization of wind energy conversion systems with variable-weight model predictive control strategy," *Applied Energy,* vol. 236, pp. 307-317, 2019, doi: 10.1016/j.apenergy.2018.11.089.

[43] B. Kelkoul and A. Boumediene, "Stability analysis and study between classical sliding mode control (SMC) and super twisting algorithm (STA) for doubly fed induction generator (DFIG) under wind turbine," *Energy,* vol. 214, 2021, doi: 10.1016/j.energy.2020.118871.

[44] B. Yang, T. Yu, H. Shu, J. Dong, and L. Jiang, "Robust sliding-mode control of wind energy conversion systems for optimal power extraction via nonlinear perturbation observers," *Applied Energy,* vol. 210, pp. 711-723, 2018, doi: 10.1016/j.apenergy.2017.08.027.

[45] D. Zholtayev, M. Rubagotti, and T. D. Do, "Adaptive super-twisting sliding mode control for maximum power point tracking of PMSG-based wind energy conversion systems," *Renewable Energy,* vol. 183, pp. 877-889, 2022, doi: 10.1016/j.renene.2021.11.055.

[46] A. Baltag, G. Livint, A. G. Baciu, and S. Bellarbi, "Design of H∞ Robust Controllers for Wind Turbines Based on DFIG," presented at the 2022 International Conference and Exposition on Electrical And Power Engineering (EPE), 2022.

[47] X. Yin, X. Tong, X. Zhao, and A. Karcanias, "Maximum Power Generation Control of a Hybrid Wind Turbine Transmission System Based on H∞ Loop-Shaping Approach," *IEEE Transactions on Sustainable Energy,* vol. 11, no. 2, pp. 561-570, 2020, doi: 10.1109/tste.2019.2897549.

[48] Z. Huo and C. Xu, "Distributed cooperative automatic generation control and multi-event triggered mechanisms co-design for networked wind-integrated power systems," *Renewable Energy,* vol. 193, pp. 41-56, 2022, doi: 10.1016/j.renene.2022.05.011.

[49] D. Song *et al.*, "A novel wind speed estimator-integrated pitch control method for wind turbines with global-power regulation," *Energy,* vol. 138, pp. 816-830, 2017, doi: 10.1016/j.energy.2017.07.033.

[50] M. A. Soliman, H. M. Hasanien, H. Z. Azazi, E. E. El-Kholy, and S. A. Mahmoud, "An Adaptive Fuzzy Logic Control Strategy for Performance Enhancement of a Grid-Connected PMSG-Based Wind Turbine," *IEEE Transactions on Industrial Informatics,* vol. 15, no. 6, pp. 3163-3173, 2019, doi: 10.1109/tii.2018.2875922.

[51] S. Wang, J. Li, Z. Hou, Q. Meng, and M. Li, "Wind power compound model-free adaptive predictive control based on full wind speed," *CSEE Journal of Power and Energy Systems,* 2020, doi: 10.17775/cseejpes.2019.02170.

[52] J. Liu, H. Meng, Y. Hu, Z. Lin, and W. Wang, "A novel MPPT method for enhancing energy conversion efficiency taking power smoothing into account," *Energy Conversion and Management,* vol. 101, pp. 738-748, 2015, doi: 10.1016/j.enconman.2015.06.005.

[53] X. Wang, J. Zhou, B. Qin, and L. Guo, "Coordinated control of wind turbine and hybrid energy storage system based on multi-agent deep reinforcement learning for wind power smoothing," *Journal of Energy Storage,* vol. 57, 2023, doi: 10.1016/j.est.2022.106297.

[54] T. Li, X. Liu, Z. Lin, J. Yang, and A. Ioannou, "A linear quadratic regulator with integral action of wind turbine based on aerodynamics forecasting for variable power production," *Renewable Energy,* 2023, doi: 10.1016/j.renene.2023.119605.

[55] H. Sun, C. Qiu, L. Lu, X. Gao, J. Chen, and H. Yang, "Wind turbine power modelling and optimization using artificial neural network with wind field experimental data," *Applied Energy,* vol. 280, 2020, doi: 10.1016/j.apenergy.2020.115880.

[56] B. Yang *et al.*, "State-of-the-art one-stop handbook on wind forecasting technologies: An overview of classifications, methodologies, and analysis," *Journal of Cleaner Production,* vol. 283, 2021, doi: 10.1016/j.jclepro.2020.124628.

[57] Y. Wang, R. Zou, F. Liu, L. Zhang, and Q. Liu, "A review of wind speed and wind power forecasting with deep neural networks," *Applied Energy,* vol. 304, 2021, doi: 10.1016/j.apenergy.2021.117766.

[58] V. Sohoni, S. C. Gupta, and R. K. Nema, "A Critical Review on Wind Turbine Power Curve Modelling Techniques and Their Applications in Wind Based Energy Systems," *Journal of Energy,* vol. 2016, pp. 1-18, 2016, doi: 10.1155/2016/8519785.

[59] S. Hanifi, X. Liu, Z. Lin, and S. Lotfian, "A Critical Review of Wind Power

Forecasting Methods—Past, Present and Future," *Energies,* vol. 13, no. 15, 2020, doi: 10.3390/en13153764.

[60] M. N. Jyothi and P. V. R. Rao, "Very-short term wind power forecasting through Adaptive Wavelet Neural Network," presented at the 2016 Biennial International Conference on Power and Energy Systems: Towards Sustainable Energy (PESTSE), 2016.

[61] S. Li, D. C. Wunsch, E. A. O'Hair, and M. G. Giesselmann, "Using neural networks to estimate wind turbine power generation," *IEEE Transactions on Energy Conversion,* vol. 16, no. 3, pp. 276-282, Sep 2001, doi: 10.1109/60.937208.

[62] M. Carolin Mabel and E. Fernandez, "Analysis of wind power generation and prediction using ANN: A case study," *Renewable Energy,* vol. 33, no. 5, pp. 986-992, 2008, doi: 10.1016/j.renene.2007.06.013.

[63] F. Pelletier, C. Masson, and A. Tahan, "Wind turbine power curve modelling using artificial neural network," *Renewable Energy,* vol. 89, pp. 207-214, 2016, doi: 10.1016/j.renene.2015.11.065.

[64] M. Morshedizadeh, M. Kordestani, R. Carriveau, D. S. K. Ting, and M. Saif, "Improved power curve monitoring of wind turbines," *Wind Engineering,* vol. 41, no. 4, pp. 260-271, 2017, doi: 10.1177/0309524x17709730.

[65] P. Zhao, J. Wang, J. Xia, Y. Dai, Y. Sheng, and J. Yue, "Performance evaluation and accuracy enhancement of a day-ahead wind power forecasting system in China," *Renewable Energy,* vol. 43, pp. 234-241, 2012, doi: 10.1016/j.renene.2011.11.051.

[66] Z. Lin, X. Liu, and M. Collu, "Wind power prediction based on high-frequency SCADA data along with isolation forest and deep learning neural networks," *International Journal of Electrical Power & Energy Systems,* vol. 118, 2020, doi: 10.1016/j.ijepes.2020.105835.

[67] H. Zhou, Y. Qiu, Y. Feng, and J. Liu, "Power prediction of wind turbine in the wake using hybrid physical process and machine learning models," *Renewable Energy,* vol. 198, pp. 568-586, 2022, doi: 10.1016/j.renene.2022.08.004.

[68] Z. Lin and X. Liu, "Wind power forecasting of an offshore wind turbine based on high-frequency SCADA data and deep learning neural network," *Energy,* vol. 201, 2020, doi: 10.1016/j.energy.2020.117693.

[69] R. P. Shetty, A. Sathyabhama, P. P. Srinivasa, and A. A. Rai, "Optimized Radial Basis Function Neural Network model for wind power prediction," presented at the 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP), Mysore, 2016.

[70] D. Karamichailidou, V. Kaloutsa, and A. Alexandridis, "Wind turbine power curve modeling using radial basis function neural networks and tabu search," *Renewable Energy,* vol. 163, pp. 2137-2152, 2021, doi: 10.1016/j.renene.2020.10.020.

[71] M. Abadi *et al.*, "TensorFlow: a system for Large-Scale machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265-283.

[72] D. Song *et al.*, "New perspectives on maximum wind energy extraction of variable-speed wind turbines using previewed wind speeds," *Energy Conversion and Management,* vol. 206, 2020, doi: 10.1016/j.enconman.2020.112496.

[73] S. Wen, Y. Li, and Y. Su, "A new hybrid model for power forecasting of a wind farm

using spatial–temporal correlations," *Renewable Energy,* vol. 198, pp. 155-168, 2022, doi: 10.1016/j.renene.2022.08.044.

[74] K. U. Jaseena and B. C. Kovoor, "Decomposition-based hybrid wind speed forecasting model using deep bidirectional LSTM networks," *Energy Conversion and Management,* vol. 234, 2021, doi: 10.1016/j.enconman.2021.113944.

[75] M. Richmond, A. Sobey, R. Pandit, and A. Kolios, "Stochastic assessment of aerodynamics within offshore wind farms based on machine-learning," *Renewable Energy,* vol. 161, pp. 650-661, 2020, doi: 10.1016/j.renene.2020.07.083.

[76] W. Zhang, Z. Lin, and X. Liu, "Short-term offshore wind power forecasting - A hybrid model based on Discrete Wavelet Transform (DWT), Seasonal Autoregressive Integrated Moving Average (SARIMA), and deep-learning-based Long Short-Term Memory (LSTM)," *Renewable Energy,* vol. 185, pp. 611-628, 2022, doi: 10.1016/j.renene.2021.12.100.

[77] M. Neshat *et al.*, "A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the Lillgrund offshore wind farm," *Energy Conversion and Management,* vol. 236, 2021, doi: 10.1016/j.enconman.2021.114002.

[78] Y. Chen *et al.*, "2-D regional short-term wind speed forecast based on CNN-LSTM deep learning model," *Energy Conversion and Management,* vol. 244, 2021, doi: 10.1016/j.enconman.2021.114451.

[79] R. Ahmed, V. Sreeram, R. Togneri, A. Datta, and M. D. Arif, "Computationally expedient Photovoltaic power Forecasting: A LSTM ensemble method augmented with adaptive weighting and data segmentation technique," *Energy Conversion and Management,* vol. 258, 2022, doi: 10.1016/j.enconman.2022.115563.

[80] L. Xiang, J. Liu, X. Yang, A. Hu, and H. Su, "Ultra-short term wind power prediction applying a novel model named SATCN-LSTM," *Energy Conversion and Management,* vol. 252, 2022, doi: 10.1016/j.enconman.2021.115036.

[81] J. Liu, X. Huang, Q. Li, Z. Chen, G. Liu, and Y. Tai, "Hourly stepwise forecasting for solar irradiance using integrated hybrid models CNN-LSTM-MLP combined with error correction and VMD," *Energy Conversion and Management,* vol. 280, 2023, doi: 10.1016/j.enconman.2023.116804.

[82] A. A. Ewees, M. A. A. Al-qaness, L. Abualigah, and M. A. Elaziz, "HBO-LSTM: Optimized long short term memory with heap-based optimizer for wind power forecasting," *Energy Conversion and Management,* vol. 268, 2022, doi: 10.1016/j.enconman.2022.116022.

[83] A. Agga, A. Abbou, M. Labbadi, Y. E. Houm, and I. H. Ou Ali, "CNN-LSTM: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production," *Electric Power Systems Research,* vol. 208, 2022, doi: 10.1016/j.epsr.2022.107908.

[84] W. Fu, Y. Fu, B. Li, H. Zhang, X. Zhang, and J. Liu, "A compound framework incorporating improved outlier detection and correction, VMD, weight-based stacked generalization with enhanced DESMA for multi-step short-term wind speed forecasting," *Applied Energy,* vol. 348, 2023, doi: 10.1016/j.apenergy.2023.121587.

[85] M. Neshat *et al.*, "Wind turbine power output prediction using a new hybrid neuro-evolutionary method," *Energy,* vol. 229, 2021, doi: 10.1016/j.energy.2021.120617.

[86] F. Gao and L. Han, "Implementing the Nelder-Mead simplex algorithm with adaptive

parameters," *Computational Optimization and Applications,* vol. 51, no. 1, pp. 259-277, 2010, doi: 10.1007/s10589-010-9329-3.

[87] A. Caliciotti, G. Fasano, S. G. Nash, and M. Roma, "An adaptive truncation criterion, for linesearch-based truncated Newton methods in large scale nonconvex optimization," *Operations Research Letters,* vol. 46, no. 1, pp. 7-12, 2018, doi: 10.1016/j.orl.2017.10.014.

[88] Z. Deng, C. Xu, X. Han, Z. Cheng, and F. Xue, "Decentralized yaw optimization for maximizing wind farm production based on deep reinforcement learning," *Energy Conversion and Management,* vol. 286, 2023, doi: 10.1016/j.enconman.2023.117031.

[89] J. Xie, H. Dong, and X. Zhao, "Data-driven torque and pitch control of wind turbines via reinforcement learning," *Renewable Energy,* vol. 215, 2023, doi: 10.1016/j.renene.2023.06.014.

[90] C. Wei, Z. Zhang, W. Qiao, and L. Qu, "Reinforcement-Learning-Based Intelligent Maximum Power Point Tracking Control for Wind Energy Conversion Systems," *IEEE Transactions on Industrial Electronics,* vol. 62, no. 10, pp. 6360-6370, 2015, doi: 10.1109/tie.2015.2420792.

[91] A. Kushwaha, M. Gopal, and B. Singh, "Q-Learning based Maximum Power Extraction for Wind Energy Conversion System With Variable Wind Speed," *IEEE Transactions on Energy Conversion,* vol. 35, no. 3, pp. 1160-1170, 2020, doi: 10.1109/tec.2020.2990937.

[92] S. Peng and Q. Feng, "Data-driven optimal control of wind turbines using reinforcement learning with function approximation," *Computers & Industrial Engineering,* vol. 176, 2023, doi: 10.1016/j.cie.2022.108934.

[93] M. Mazare, "Adaptive optimal secure wind power generation control for variable speed wind turbine systems via reinforcement learning," *Applied Energy,* vol. 353, 2024, doi: 10.1016/j.apenergy.2023.122034.

[94] W. Jiang, Y. Liu, G. Fang, and Z. Ding, "Research on short-term optimal scheduling of hydro-wind-solar multi-energy power system based on deep reinforcement learning," *Journal of Cleaner Production,* vol. 385, 2023, doi: 10.1016/j.jclepro.2022.135704.

[95] J. M. Jonkman and M. L. B. Jr., "FAST User's Guide," National Renewable Energy Laboratory, Golden, Colorado, 2005.

[96] N. R. E. Laboratory. "OpenFAST Documentation, User Documentation." https://openfast.readthedocs.io/en/main/source/user/index.html# (accessed Nov, 2023).

[97] E. F. Camacho and C. B. Alba, *Model predictive control.* Springer science & business media, 2013.

[98] QBlade.org. "QBlade - Next Generation Wind Turbine Simulation." https://qblade.org/ (accessed Sep, 2024).

[99] D. N. Veritas. "Bladed - Wind turbine design software." https://www.dnv.com/publications/bladed-149137/ (accessed Sep, 2024).

[100] J. Morgado, M. Â. R. Silvestre, and J. C. Páscoa, "Validation of New Formulations for Propeller Analysis," *Journal of Propulsion and Power,* vol. 31, no. 1, pp. 467-477, 2015, doi: 10.2514/1.B35240.

[101] L. Cottura, R. Caradonna, A. Ghigo, R. Novo, G. Bracco, and G. Mattiazzo,

"Dynamic Modeling of an Offshore Floating Wind Turbine for Application in the Mediterranean Sea," *Energies,* vol. 14, no. 1, 2021, doi: 10.3390/en14010248.
[102] S. A. Ning, "A simple solution method for the blade element momentum equations with guaranteed convergence," *Wind Energy,* vol. 17, no. 9, pp. 1327-1345, 2013, doi: 10.1002/we.1636.
[103] Y. E. khchine and M. Sriti, "Tip Loss Factor Effects on Aerodynamic Performances of Horizontal Axis Wind Turbine," *Energy Procedia,* vol. 118, pp. 136-140, 2017, doi: doi.org/10.1016/j.egypro.2017.07.028.
[104] S. Wan, L. Cheng, and X. Sheng, "Effects of Yaw Error on Wind Turbine Running Characteristics Based on the Equivalent Wind Speed Model," *Energies,* vol. 8, no. 7, pp. 6286-6301, 2015, doi: 10.3390/en8076286.
[105] V. P. Singh, N. Kishor, P. Samuel, and N. Singh, "Small-signal stability analysis for two-mass and three-mass shaft model of wind turbine integrated to thermal power system," *Computers & Electrical Engineering,* vol. 78, pp. 271-287, 2019, doi: 10.1016/j.compeleceng.2019.07.016.
[106] B. Boukhezzar and H. Siguerdidjane, "Nonlinear Control of a Variable-Speed Wind Turbine Using a Two-Mass Model," *IEEE Transactions on Energy Conversion,* vol. 26, no. 1, pp. 149-162, 2011, doi: 10.1109/tec.2010.2090155.
[107] V. D. Colli, F. Marignetti, and C. Attaianese, "2-D mechanical and magnetic analysis of a 10 MW doubly fed induction generator for direct-drive wind turbines," in *2009 35th Annual Conference of IEEE Industrial Electronics*, 2009: IEEE, pp. 3863-3867.
[108] F. Mazouz, S. Belkacem, Y. Harbouche, R. Abdessemed, and S. Ouchen, "Active and Reactive Power Control of a DFIG For Variable Speed Wind Energy Conversion," in *Proceedings of the 6th International Conference on Systems and Control*, Batna, Algeria, 2017.
[109] P. K. Gayen, D. Chatterjee, and S. K. Goswami, "Stator side active and reactive power control with improved rotor position and speed estimator of a grid connected DFIG (doubly-fed induction generator)," *Energy,* vol. 89, pp. 461-472, 2015, doi: 10.1016/j.energy.2015.05.111.
[110] N. Mohan, *Advanced electric drives analysis, control, and modeling using MATLAB Simulink*. New Jersey: John Wiley & Sons, 2014.
[111] B. K. Bose, *Modern Power Electronics and AC Drives*, 1st ed. Prentice-Hall, 2001.
[112] C. HAMON, "Doubly-fed Induction Generator Modeling and Control in DigSilent Power Factory," Master, KTH School of Electrical Engineering, KTH, 2010.
[113] W. Qiao, "Dynamic Modeling and Control of Doubly Fed Induction Generators Driven by Wind Turbines," presented at the 2009 IEEE/PES Power Systems Conference and Exposition, 2009.
[114] R. Fadaeinedjad, M. Moallem, and G. Moschopoulos, "Simulation of a Wind Turbine With Doubly Fed Induction Generator by FAST and Simulink," *IEEE Transactions on Energy Conversion,* vol. 23, no. 2, pp. 690-700, 2008, doi: 10.1109/tec.2007.914307.
[115] B. Ozpineci and L. M. Tolbert, "Simulink implementation of induction machine model - a modular approach," presented at the IEEE International Electric Machines and Drives Conference, 2003. IEMDC'03., 2003.
[116] Y. Mousavi, G. Bevan, I. B. Kucukdemiral, and A. Fekih, "Sliding mode control of

wind energy conversion systems: Trends and applications," *Renewable and Sustainable Energy Reviews,* vol. 167, 2022, doi: 10.1016/j.rser.2022.112734.

[117] D.-C. Phan and S. Yamamoto, "Rotor speed control of doubly fed induction generator wind turbines using adaptive maximum power point tracking," *Energy,* vol. 111, pp. 377-388, 2016, doi: 10.1016/j.energy.2016.05.077.

[118] T. Li, J. Yang, and A. Ioannou, "Wind forecasting-based model predictive control of generator, pitch, and yaw for output stabilisation – A 15-megawatt offshore," *Energy Conversion and Management,* vol. 302, 2024, doi: 10.1016/j.enconman.2024.118155.

[119] J. Chen, W. Yao, C.-K. Zhang, Y. Ren, and L. Jiang, "Design of robust MPPT controller for grid-connected PMSG-Based wind turbine via perturbation observation based nonlinear adaptive control," *Renewable Energy,* vol. 134, pp. 478-495, 2019, doi: 10.1016/j.renene.2018.11.048.

[120] M. K. K. Prince, M. T. Arif, M. E. Haque, A. Gargoom, and A. M. T. Oo, "Design and implementation of finite control set MPC with an LCL filter for grid-tied PMSG based wind turbine," *International Journal of Electrical Power & Energy Systems,* vol. 152, 2023, doi: 10.1016/j.ijepes.2023.109197.

[121] Y. Ren, L. Li, J. Brindley, and L. Jiang, "Nonlinear PI control for variable pitch wind turbine," *Control Engineering Practice,* vol. 50, pp. 84-94, 2016, doi: 10.1016/j.conengprac.2016.02.004.

[122] D. Song *et al.*, "Deep optimization of model predictive control performance for wind turbine yaw system based on intelligent fuzzy deduction," *Expert Systems with Applications,* vol. 221, 2023, doi: 10.1016/j.eswa.2023.119705.

[123] X. Zhao, Z. Lin, B. Fu, and S. Gong, "Research on frequency control method for micro-grid with a hybrid approach of FFR-OPPT and pitch angle of wind turbine," *International Journal of Electrical Power & Energy Systems,* vol. 127, 2021, doi: 10.1016/j.ijepes.2020.106670.

[124] S. Hur and W. E. Leithead, "Collective control strategy for a cluster of stall-regulated offshore wind turbines," *Renewable Energy,* vol. 85, pp. 1260-1270, 2016, doi: 10.1016/j.renene.2015.07.087.

[125] N. J. Abbas, J. Jasa, D. S. Zalkind, A. Wright, and L. Pao, "Control co-design of a floating offshore wind turbine," *Applied Energy,* vol. 353, 2024, doi: 10.1016/j.apenergy.2023.122036.

[126] Z. Zhu, J. Zhang, S. Zhu, and J. Yang, "Digital twin technology for wind turbine towers based on joint load–response estimation: A laboratory experimental study," *Applied Energy,* vol. 352, 2023, doi: 10.1016/j.apenergy.2023.121953.

[127] E. A. Bossanyi, "The Design of closed loop controllers for wind turbines," *Wind Energy,* vol. 3, no. 3, pp. 149-163, 2001, doi: 10.1002/we.34.

[128] M. Eremia and M. Shahidehpour, *Handbook of Electrical Power System Dynamics: Modeling, Stability, and Control.* Wiley-IEEE Press, 2013.

[129] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW Reference Wind Turbine for Offshore System Development," National Renewable Energy Laboratory, 2009.

[130] E. Gaertner *et al.*, "IEA wind TCP task 37: definition of the IEA 15-megawatt offshore reference wind turbine," National Renewable Energy Lab.(NREL), Golden, CO (United States), 2020.

[131] L. Liu, J. Wang, J. Li, and L. Wei, "An online transfer learning model for wind turbine power prediction based on spatial feature construction and system-wide update," *Applied Energy,* vol. 340, 2023, doi: 10.1016/j.apenergy.2023.121049.
[132] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," presented at the 2008 Eighth IEEE International Conference on Data Mining, 2008.
[133] scikit-learn. "Isolation Forest Algorithm." https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html (accessed Mar, 2024).
[134] A. P. Marugán, F. P. G. Márquez, J. M. P. Perez, and D. Ruiz-Hernández, "A survey of artificial neural network in wind energy systems," *Applied Energy,* vol. 228, pp. 1822-1836, 2018, doi: 10.1016/j.apenergy.2018.07.084.
[135] C. S. K. Dash, A. K. Behera, S. Dehuri, and S.-B. Cho, "Radial basis function neural networks: a topical state-of-the-art survey," *Open Computer Science,* vol. 6, no. 1, pp. 33-63, 2016, doi: 10.1515/comp-2016-0005.
[136] T. Li, X. Liu, Z. Lin, and R. Morrison, "Ensemble offshore Wind Turbine Power Curve modelling – An integration of Isolation Forest, fast Radial Basis Function Neural Network, and metaheuristic algorithm," *Energy,* vol. 239, 2022, doi: 10.1016/j.energy.2021.122340.
[137] Y.-Y. Hong and C. L. P. P. Rioflorido, "A hybrid deep learning-based neural network for 24-h ahead wind power forecasting," *Applied Energy,* vol. 250, pp. 530-539, 2019, doi: 10.1016/j.apenergy.2019.05.044.
[138] S. E. Fahlman, *An empirical study of learning speed in back-propagation networks.* Carnegie Mellon University, Computer Science Department Pittsburgh, PA, USA, 1988.
[139] N. Ampazis and S. J. Perantonis, "Two highly efficient second-order algorithms for training feedforward networks," *IEEE Transactions on Neural Networks,* vol. 13, no. 5, pp. 1064-1074, 2002.
[140] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747,* 2016.
[141] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of machine learning research,* vol. 12, no. 7, 2011.
[142] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701,* 2012.
[143] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on,* vol. 14, no. 8, p. 2, 2012.
[144] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," presented at the 3rd International Conference for Learning Representations, San Diego, 2015.
[145] T. Dozat, "Incorporating nesterov momentum into adam," 2016.
[146] H. B. McMahan *et al.*, "Ad click prediction: a view from the trenches," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1222-1230.
[147] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning,*

2013: PMLR, pp. 1139-1147.
[148] scikit-learn. "GridSearchCV." https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (accessed Sep, 2024).
[149] B. Wu, Y. Lang, N. Zargari, and S. Kouro, *Power conversion and control of wind energy systems*. John Wiley & Sons, 2011.
[150] R. Cardenas, R. Peña, S. Alepuz, and G. Asher, "Overview of control systems for the operation of DFIGs in wind energy applications," *IEEE Transactions on Industrial Electronics,* vol. 60, no. 7, pp. 2776-2798, 2013.
[151] A. Kisvari, Z. Lin, and X. Liu, "Wind power forecasting – A data-driven method along with gated recurrent neural network," *Renewable Energy,* vol. 163, pp. 1895-1909, 2021, doi: 10.1016/j.renene.2020.10.119.
[152] T. Li, J. Yang, and A. Ioannou, "Data-driven control of wind turbine under online power strategy via deep learning and reinforcement learning," *Renewable Energy,* vol. 234, 2024, doi: 10.1016/j.renene.2024.121265.
[153] S. Guadarrama *et al.*, "TF-Agents: A library for reinforcement learning in tensorflow," 2018. [Online]. Available: https://github.com/tensorflow/agents.
[154] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature,* vol. 518, no. 7540, pp. 529-533, 2015, doi: 10.1038/nature14236.
[155] A. K. Shakya, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," *Expert Systems with Applications,* vol. 231, 2023, doi: 10.1016/j.eswa.2023.120495.
[156] S. Ivanov and A. D'yakonov, "Modern deep reinforcement learning algorithms," *arXiv preprint arXiv:1906.10025,* 2019.
[157] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *International conference on machine learning*, 2017: PMLR, pp. 449-458.
[158] M. Hessel *et al.*, "Rainbow: Combining improvements in deep reinforcement learning," in *Proceedings of the AAAI conference on artificial intelligence*, 2018, vol. 32, no. 1.
[159] K. Ogata, *Modern control engineering*. Prentice hall Upper Saddle River, NJ, 2010.
[160] M. H. Hansen, A. Hansen, T. J. Larsen, S. Øye, P. Sørensen, and P. Fuglsang, "Control design for a pitch-regulated, variable speed wind turbine," Forskningscenter Risoe, Denmark.
[161] G. R. Goncalves da Silva, A. S. Bazanella, C. Lorenzini, and L. Campestrini, "Data-Driven LQR Control Design," *IEEE Control Systems Letters,* vol. 3, no. 1, pp. 180-185, 2019, doi: 10.1109/lcsys.2018.2868183.
[162] T. Duriez, S. L. Brunton, and B. R. Noack, *Machine learning control-taming nonlinear dynamics and turbulence*. Springer, 2017.
[163] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.
[164] P. Van Dooren, "A generalized eigenvalue approach for solving Riccati equations," *SIAM Journal on Scientific and Statistical Computing,* vol. 2, no. 2, pp. 121-135, 1981.
[165] A. Laub, "A Schur method for solving algebraic Riccati equations," *IEEE Transactions on automatic control,* vol. 24, no. 6, pp. 913-921, 1979.

[166] L. Wang, *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, 2009.
[167] A. Haber, A. Polo, I. Maj, S. F. Pereira, H. P. Urbach, and M. Verhaegen, "Predictive control of thermally induced wavefront aberrations," *Opt Express,* vol. 21, no. 18, pp. 21530-41, Sep 9 2013, doi: 10.1364/OE.21.021530.
[168] S. H. Zak, *Systems and control*. Oxford University Press New York, 2003.
[169] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "Prox-qp: Yet another quadratic programming solver for robotics and beyond," in *RSS 2022-Robotics: Science and Systems*, 2022.
[170] S. Caron. "Optimality conditions and numerical tolerances in QP solvers." https://github.com/qpsolvers/qpsolvers_benchmark (accessed Nov, 2023).
[171] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation,* vol. 12, no. 4, pp. 637-672, 2020, doi: 10.1007/s12532-020-00179-2.
[172] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical programming,* vol. 27, no. 1, pp. 1-33, 1983.
[173] L. Vandenberghe. (2010). The CVXOPT linear and quadratic cone program solvers. Available: http://cvxopt.org/documentation/coneprog.pdf
[174] B. O'Donoghue, "Operator splitting for a homogeneous embedding of the linear complementarity problem," *SIAM Journal on Optimization,* vol. 31, no. 3, pp. 1999-2023, 2021.
[175] P. Goulart and Y. Chen. "Interior Point Conic Optimization for Julia." https://oxfordcontrol.github.io/ClarabelDocs/stable/ (accessed May, 2023).
[176] R. Schafer, "What Is a Savitzky-Golay Filter? [Lecture Notes]," *IEEE Signal Processing Magazine,* vol. 28, no. 4, pp. 111-117, 2011, doi: 10.1109/msp.2011.941097.
[177] C. Zhang, J. Zhou, C. Li, W. Fu, and T. Peng, "A compound structure of ELM based on feature selection and parameter optimization using hybrid backtracking search algorithm for wind speed forecasting," *Energy Conversion and Management,* vol. 143, pp. 360-376, 2017, doi: 10.1016/j.enconman.2017.04.007.
[178] Y. Zhang, Y. Li, and G. Zhang, "Short-term wind power forecasting approach based on Seq2Seq model using NWP data," *Energy,* vol. 213, 2020, doi: 10.1016/j.energy.2020.118371.
[179] TensorFlow.org. "TensorFlow Core Tutorials - Time series forecasting." https://www.tensorflow.org/tutorials/structured_data/time_series (accessed Aug, 2023).