Uncertainty Quantification of A Wind Farm Control Model: FLORIDyn

D.Shi

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Uncertainty Quantification of A Wind Farm Control Model: FLORIDyn

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

D.Shi

June 22, 2023

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Uncertainty Quantification of A Wind Farm Control Model: FLORIDyn

by

D.Shi

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: June 22, 2023

Supervisor(s):

Prof. Dr. Ir. J. W. van Wingerden Supervisor

Dr. Ir. Vinit V. Dighe Second Supervisor

Reader(s):

Abstract

Wind energy becomes more and more popular since it is environmentally friendly. Wind farm control is one of the most popular topics and it works on steering the wind farm to extract energy from wind as much as possible. Generally, the model capturing wake effects between turbines in the wind farm plays a role in wind farm control. The existing FLORIS model is considered suitable for wind farm control due to the fact that it has the ability or potential to capture wake features with reasonably computational costs. A drawback of the FLORIS model is the lack of dynamics, which is improved by developing the FLORIDyn model.

This thesis focuses on a Gaussian FLORIDyn model. The objective is to explore the possibility of improving the model accuracy by quantifying the associated uncertainty in the model parameters. Uncertainty quantification consisting of sensitivity analysis and Bayesian calibration is conducted based on a 3-Turbine case simulation using the UQLab software. Since a MCMC algorithm associated with Bayesian calibration requires to evaluate the FLORIDyn model multiple times, it can result in massive computational expenses when directly applying the computational model to the simulation. To deal with this, a surrogate model is first constructed to replace the original model. This thesis assesses two types of approaches for surrogate model construction which are the Kriging-based approach and the PCE-based approach. One approach is chosen after the comprehensive comparison in terms of accuracy and efficiency. The constructed surrogate model is then applied to the sensitivity analysis using Sobol' indices to investigate how each model parameter of interest affects the model output. Last, the high-fidelity SOWFA data are used as experimental data for Bayesian calibration. Compared to non-calibrated model outputs, calibrated model outputs are closer to the SOWFA data, which means that the accuracy of the FLORIDyn model is improved.

Keywords FLORIS, FLORIDyn, surrogate model, sensitivity analysis, Bayesian calibration

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor J. W. van Wingerden and my daily supervisor Vinit V. Dighe for their assistance during the writing of this thesis. I really appreciate it.

Delft, University of Technology D.Shi
June 22, 2023

"Solar power, wind power, the way forward is to collaborate with nature, it is the only way we are going to get to the other end of the 21st century."

— *Björk*

Chapter 1

Introduction

The context for the research conducted in this thesis is shown in this chapter. Section 1-1 describes the motivation for conducting this research by introducing the relevant background in the field of electricity generation and by investigating an offshore wind energy subject. Section 1-2 elaborates wind farm control in terms of control goals, methods and structures. Section 1-3 introduces several wind farm models related to this thesis. Section 1-4 first discusses existing studies and corresponding scientific gaps in Section 1-4-1. The objectives of the research are then summarized in Section 1-4-2 and the structure of this thesis is last shown in Section 1-4-3.

1-1 Motivation

The earth is being threatened by the increased emissions of greenhouse gases resulting from the excessive use of fossil fuels. Despite the fact that global CO_2 emissions reduced by 5.8% in 2020, global energy-related CO_2 emissions still remained $31.5GT$, which is approximately 50% higher than that in 1800 [2]. To deal with this situation, 2030 Climate Target Plan [1], one of proposals was put forward by European Commission. This plan is comprised of three main parts: cut greenhouse gas emissions by at least 55% by 2030 to make Europe be able to take the responsibility of achieving climate neutrality by 2050; create green jobs to reduce the conflict between cutting greenhouse gas emissions and developing the economy; inspire international partners to contribute to keeping the rise in global temperature to 1.5°C.

Replacing thermal power generation with renewable energy produced from wind, solar, hydro, tidal, geothermal and biomass is one of the most significant approaches to reduce CO_2 emissions. Figure 1-1 shows gross electricity production in 28 European Union member countries by energy sources [27]. Wind energy is expected to become the largest source of electricity for the European Union by 2050 [27]. In order to reduce the costs of installation and maintenance, a number of wind turbines are generally connected to the power system for producing electricity to form a cluster called as a wind farm (wind park). However, one issue that needs

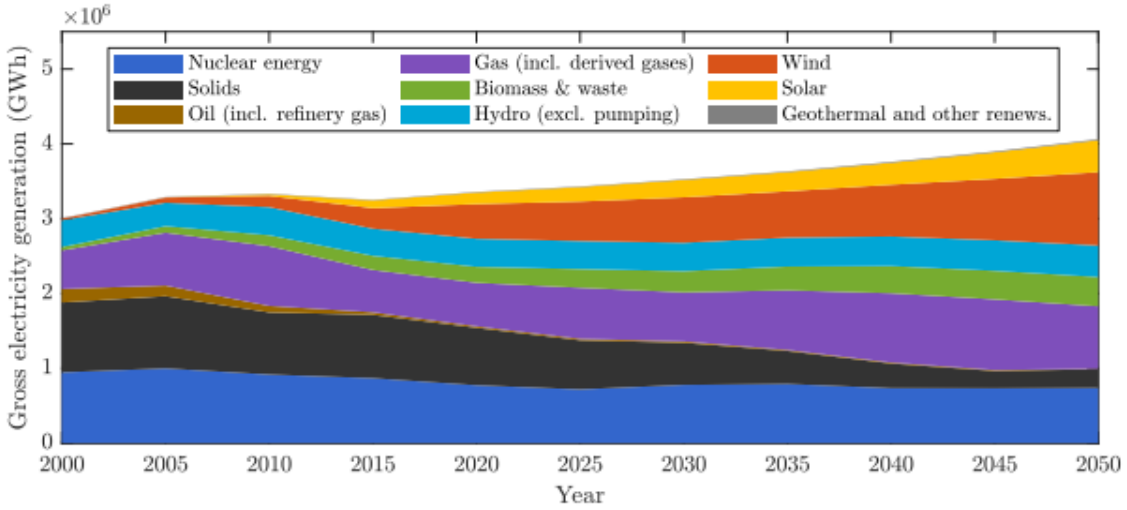


Figure 1-1: Gross electricity production in the EU-28 by energy sources from 2000 forecasted to 2050 [27]

to be addressed is the loss of efficiency for power generation due to wake interaction effects[1] between the wind turbines in the wind farm. Compared to the upstream wind turbines in the wind farm, the wake interaction effects are mainly exerted on the downstream wind turbines to result in the reduction in wind speeds and the increase in turbulence intensity[2] [58]. The turbulence leads to continuous changes not only in wind speeds, but also in wind directions over time and positions. Therefore, wake interaction effects cause the reduction in the power production of the wind farm. In turn, the operation states of wind turbines also have an influence on the wake. As a consequence, a proper wind farm controller with a better understanding of wake interaction effects can be expected to potentially improve the performance of a wind farm [33].

1-2 Wind farm control

Compared to conventional greedy control[3], wind farm control, a family of methods essentially operates the individual wind turbine within a wind farm in a coordinated way to achieve a common goal, which is basically to minimize the levelized costs of energy (LCOE) which can be computed by the sum of costs over lifetime divided by the sum of electricity production over lifetime [10]. There are three parts for achieving the goal which are minimizing the wear on the turbine structure, integrating with the electricity grid and maximizing the annual energy production [33]. Distributing the structure loads between wind turbines in a wind farm with the concept of wind farm control can decrease the wear experienced by wind turbines to make their structures degrade at a constant rate. The maintenance costs can be therefore reduced

[1] The aggregated influence on the energy production of the wind farm, which results from the changes in wind speeds caused by the impact of the different wind turbines [8].

[2] The turbulence intensity is a measure of abrupt fluctuations in wind speeds and can be computed by the standard deviation of the wind speed changes divided by the average wind speed.

[3] A control strategy of optimizing the power output of each turbine without considering other turbines.

in this way [69] and it also offers the possibility of achieving lighter designs which have more reasonable prices and longer lifetime [55]. The consumption of electricity in the grid and the generation of electricity to the grid mostly need to match each other. If the errors between two of them are not small enough, the power might be cut off and grid-connected machines can not work properly. The conventional power suppliers like hydraulic and thermal power plants are able to quickly adjust the supplied power to maintain the stability of the grid. However, power generated from wind or solar energy is free from such functionality [29]. These renewable technologies are therefore required to be able to regulate grid frequencies due to the increase of the proportion of wind and solar power production in total electricity yields [12]. Based on [35], such ancillary grid service can be also offered by the wind farm with the concept of wind farm control. Last, power maximization is the most intuitive goal of wind farm control and it has been smoothly conducted in high-fidelity simulations [44], wind tunnel experiments [26] and field experiments [31] [37] [38] [52]. The commercialization of the concept is also being researched in the industry field [5].

The wind farm control mainly uses two strategies which are axial induction control and wake steering. The former one focuses on affecting the axial induction factor defined as the reduction[4] from the freestream speed to the wind speed at the rotor divided by the freestream speed. The axial induction factor is relative to the thrust coefficient C_T which depends on the tip speed ratio[5] and the blade pitch angle[6] [67]. The latter one focuses on redirecting wind turbines' wakes away from downstream wind turbines by steering upstream wind turbines to be misaligned with the wind direction [80].

Wind farm control methods can be roughly divided into two types which are the model-based method and the model-free method. The former one makes predictions based on the mathematical model of the wind farm in the time domain. The latter one generally considers the wind farm as a black box and focuses on the measurements of system responses to approach the most reasonable control operation in the future. Besides, structures of wind farm controllers can be classified as closed-loop (feedback) and open-loop (feedforward). The former one takes into account the measurements of system responses to determine control settings, while the latter one only considers the prior information. More details are described next.

1-2-1 Open-loop model-free control

Figure 1-2 shows the industry-standard greedy wind farm control which is model-free and open-loop. Each wind turbine in the wind farm is individually controlled to maximize its own power generation without considering the other turbines. In addition, the interactions between turbines in the wind farm are also not taken into account [33].

1-2-2 Open-loop model-based control

Figure 1-3 presents the structure of open-loop model-based control, which mostly uses yaw-based wake steering algorithms [26] [37] [38] [44] [52]. A mathematical model is used to obtain

[4]The air passes through the rotor plane with a smaller speed than the freestream speed due to the fact that some air is deflected away when approaching the rotor.

[5]The ratio of the blade tip linear speed to the freestream speed.

[6]The angle between the chord line and the plane of rotation.

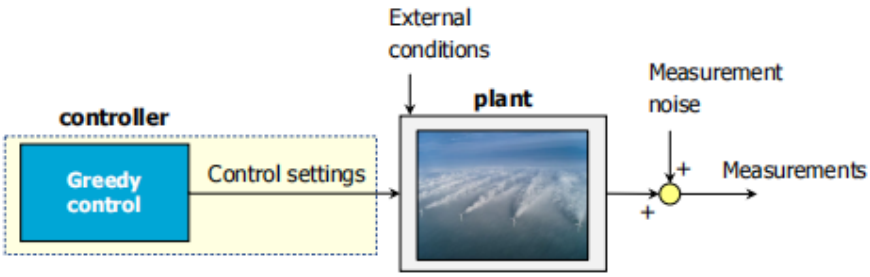


Figure 1-2: The structure of locally greedy wind farm control [33]

the wind field data for determining control settings. Additionally, this model can also capture the influence of control settings on the system performance. It is important to note that using the mathematical model directly in the algorithm can result in massive computational costs. Generally, a surrogate model[7] is first constructed to replace the original model, which can significantly improve computational efficiency by sacrificing the model accuracy. Since the control settings and the system responses will not be returned as feedback, it has high sensitivity to external conditions and the model accuracy, which means that it is easily disturbed [13] [32].

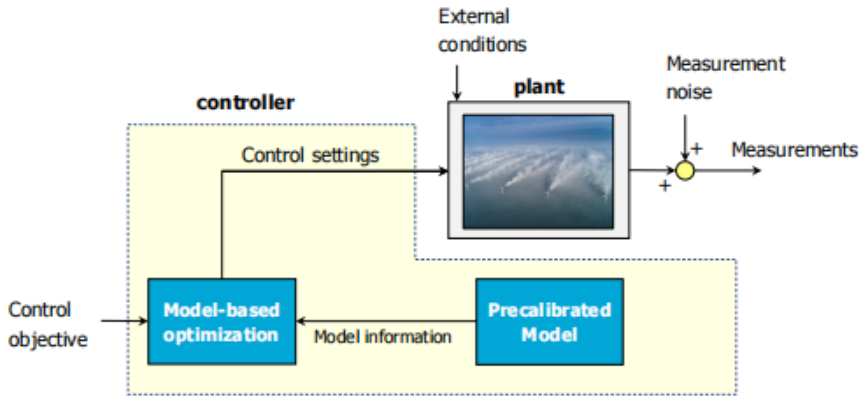


Figure 1-3: The structure of open-loop model-based wind farm control [33]

1-2-3 Closed-loop model-free control

The structure of closed-loop model-free control is shown in Figure 1-4 where the wind farm is considered as a black box system with inputs and outputs. In such methods, the system outputs are directly optimized without capturing temporal and spatial dynamics of the wind

[7] Surrogate modelling is an engineering method to replace the difficultly measured or computed outcomes of interest with outputs of an approximate model constructed by modelling the response to a limited number of intelligently chosen data points in the simulation.

farm. Campagnolo et al [26]. have achieved maximizing the power production by yaw-based wake steering closed-loop model-free control. However, due to neglecting the wake interactions, there is time delay for wake propagation, which results in that the effect of changing the yaw angle on the downstream wind turbine can be delayed a lot [24].

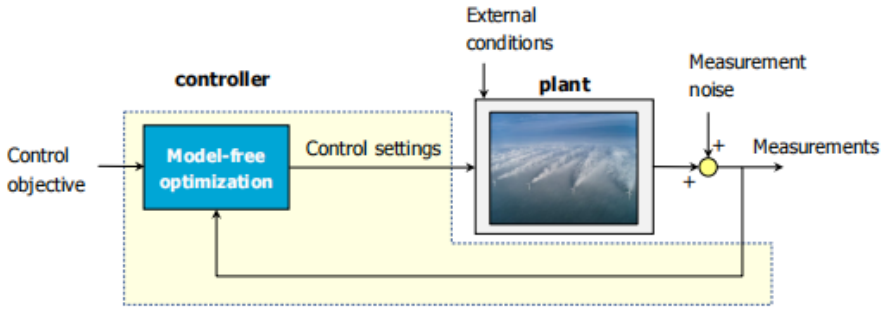


Figure 1-4: The structure of closed-loop model-free wind farm control [33]

1-2-4 Closed-loop model-based control

Figure 1-5 presents closed-loop model-based control, which is similar to open-loop model-based control in terms of low computational expenses and quick convergence and to closed-loop model-free control in terms of automatic corrections to process disturbances and resilience to model uncertainties [33]. The control settings and system responses will be returned as feedback to calibrate the mathematical model. Such closed-loop model-based methods have been applied to mitigate loads [25] [82], to regulate electricity grid frequencies [16] [25] [82] [56] [79] and to maximize power production [43] [70].

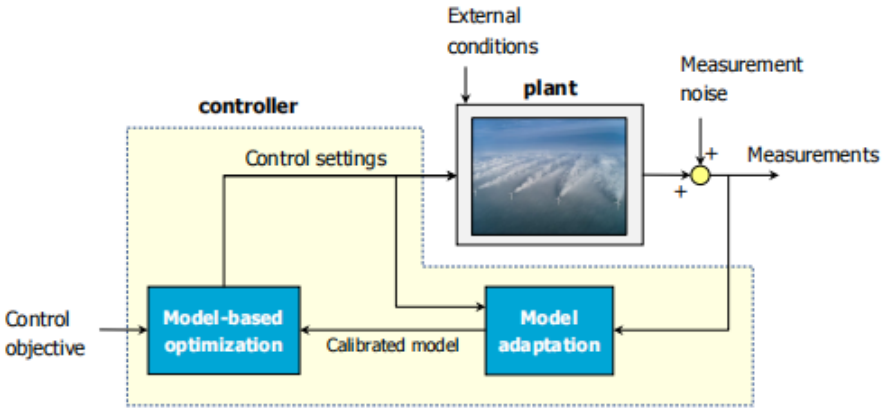


Figure 1-5: The structure of closed-loop model-based wind farm control [33]

1-3 Wind farm modelling

It is intuitive that the performance of model-based wind farm control depends on the accuracy and computational expenses of the mathematical model applied to the system. Basically, models can be divided into three categories which are high-fidelity models, medium-fidelity models and low-fidelity models. The offline applications like simulating wind farm flows and turbines require high-fidelity models to gain higher accuracy, while low-fidelity models are suitable for building controllers and analysing wind farms since low computational expenses make it feasible to optimize or calculate values in the real-time domain, even though the accuracy is sacrificed. Except Goit and Meyers [50] and Munters and Meyers [66], there are very few other research groups having directly used high-fidelity models in model-based wind farm control [33]. Some selected models are introduced next.

1-3-1 High-fidelity model

Computational Fluid Dynamics (CFD) simulations are high-fidelity methods which use fundamental nature laws to simulate liquids and gases in terms of the flow and interactions. Additionally, physical effects such as ground effects, turbulence effects and wind turbbine structure dynamics are also taken into account. The Simulator for Offshore Wind Farm Applications (SOWFA) is an open source tool for CFD simulations, which is developed with the OpenFOAM [4] software [3] by the National Renewable Energy Laboratory (NREL). It can capture most wind field features accurately and be applied to the validation for lower-fidelity models [89] [39] [45]. However, the computational expenses of the simulation with SOWFA are much higher than with lower-fidelity models.

1-3-2 Medium-fidelity model

Compared to high-fidelity simulations, the complexity of medium-fidelity simulations are reduced by making simplifications and assumptions to only reserve the necessary features of high-fidelity simulations. One example is a two-dimensional dynamic wind farm model, Wind Farm Simulator (WFSim), which is derived from the two-dimensional Navier-Stokes equations by Boersma et al [23]. The continuity equation in the standard Navier-Stokes model is altered to imitate wake depletion in the vertical dimension which is not modelled. This model is validated using PALM[8] and SOWFA data and it is fast enough to work as a part of a closed-loop control framework [41]. Its benefits are similar to high-fidelity models'. Moreover, the computational expenses and the number of states are approximately linearly related.

1-3-3 Low-fidelity model

In low-fidelity models, parametric equations are utilized to depict wake effects instead of differential equations applied to CFD models. It can significantly decrease computational expenses by using a set of adjustable parameters to capture wake features. Two examples related

[8] PArallelized LES Model (PALM): a reimplementation of an Large Eddy Simulation (LES) model. The LES model is a mathematical model for turbulence applied to CFD simulations. PALM is optimized to run on massively parallel computers [71].

to each other are introduced next and the second one is the main research object in this thesis.

FLORIS: FLOw Redirection and Induction in Steady-state (FLORIS) is a simplified control-oriented low-fidelity parametric model to predict the steady-state wake features in a wind farm in terms of the axial induction factor, turbulence intensity and the yaw angle. In FLORIS, due to lack of dynamics, a new wake shape directly results from changes at the rotor plane and there is no propagation related to the changes. The first FLORIS model named Zone FLORIS model was developed by Pieter Gebraad et al in 2014. The model utilized the Actuator Disc Model (ADM)[9] and the Jensen-Park model[10]. As shown in Figure 1-6, there are three divided zones in the Zone FLORIS model which are the near wake zone, the far wake zone and the mixing zone respectively and the wind speeds at outer zones recover faster than the wind speeds at inner zones. The Zone FLORIS model captures the wake features by considering the wind direction and speed, the yaw angle and the axial induction factor and without considering the turbulence intensity. Gebraad and van Wingerden [46] afterwards used it as a basis to develop the FLORIDyn model in the same year. Two years later, the Gaussian FLORIS model fitting Gaussian curves to the wake shape based on the experimental data was presented by Bastankhah and Porté-Agel [15]. The Gaussian FLORIS model is a part of the FLORIDyn model used in this thesis and more details are described in Chapter 2.

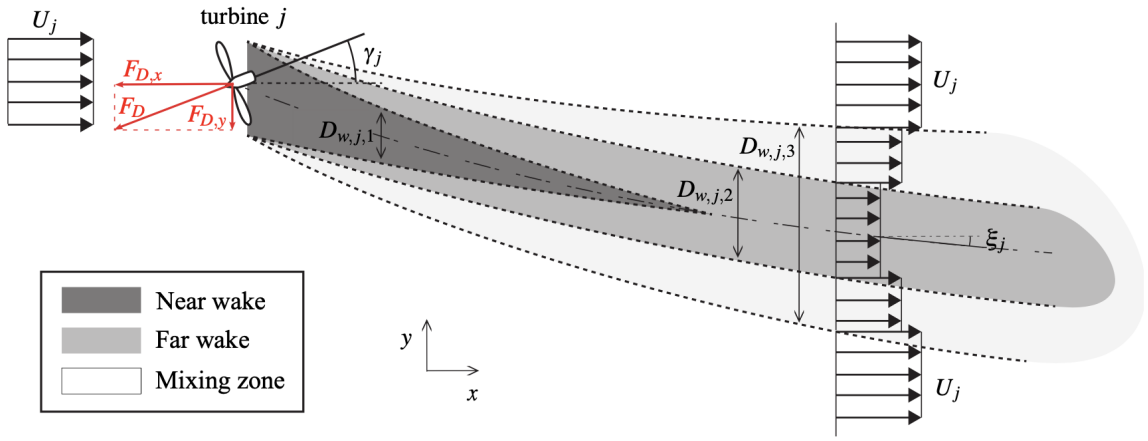


Figure 1-6: FLORIS model by Gebraad et al. [46]

FLORIDyn: FLOw Redirection and Induction Dynamics (FLORIDyn) is a novel control-oriented dynamic model to capture the wake interaction effects between wind turbines in a wind farm. Unlike there is lack of dynamic wake features in FLORIS, FLORIDyn captures not only wake effects at each wind turbine, but also the time delay between control-setting adjustments and responses of downstream wind turbines in the wind farm [43]. As mentioned above, the FLORIDyn model was initially presented by Gebraad and van Wingerden [46] based on the Zone FLORIS model in 2014. Afterwards, Becker worked on a Gaussian FLORIDyn model based on the Gaussian FLORIS model in 2020 [17]. More details of the

[9] It considers the rotor of the wind turbine as a uniform disc extracting energy from wind [19].

[10] It assumes that the wind speed recovers to the freestream speed in a constantly expanding cone formed by the wake which trails the wind turbine [53].

Gaussian FLORIDyn model are elaborated in Chapter 2.

1-4 Contributions

The research on maximizing power production using wind farm control is extremely popular in both academic and industrial fields [33]. However, there are still many related scientific gaps requiring to be bridged, which are discussed in Section 1-4-1. Next, the objectives of this thesis are described in Section 1-4-2 and the outline of this thesis is presented in Section 1-4-3.

1-4-1 Existing research and corresponding scientific gaps

Since the low-fidelity model is the research object in this thesis, several existing studies on Low-fidelity models are first reviewed to gain a better understanding and summarized as follows. The research conducted by Pieter Gebraad et al. [45] refers to the two-dimensional Zone FLORIS model whose wake shape depends on the wind speed and direction, the yaw angle and the axial induction factor. However, there are discontinuities in the wind field and this model does not consider ambient turbulence intensity, heterogeneous wind conditions or dynamics. Bastankhah and Porté-Agel [15] presented a three-dimensional Gaussian FLORIS model depicting the wake without structures of eddy or turbulence. Unlike the Zone FLORIS model, the influence of ambient turbulence on wake recovery is considered. However, heterogeneous wind conditions and dynamics are also missing. Afterwards, the FLORIS model including the heterogeneous conditions was proposed based on the Gaussian FLORIS model by Farrell et al. in 2020. It can present spatially heterogeneous wakes. It holds all of the benefits mentioned in the previous two papers but still lacks the dynamics [36]. To include dynamics, the first FLORIDyn model was developed based on the Zone FLORIS model by Gebraad and van Wingerden in 2014. Compared to the Zone FLORIS model, no benefits and drawbacks are changed but dynamics [46]. To combine heterogeneous wind conditions and dynamics, Becker [17] worked on a Gaussian FLORIDyn model based on the Gaussian FLORIS model in 2020. The developed FLORIDyn model keeps all of the benefits mentioned in previous FLORIS models. However, there is still room for improvement in model accuracy to approximate high-fidelity simulation data better.

To improve model accuracy, the intuitive approach is to calibrate the model. Basically, conventional calibration consists of expert-based calibration and optimization-based calibration. The former one focuses on minimizing the difference between the measurement data and model outputs by tuning model parameters [74]. The latter one has the same aim as expert-based calibration and achieves it using the optimization algorithms [42]. Conventional calibration assumes that there are no uncertainties in the measurement data, which means that all uncertainties in model outputs will be reduced in the calibration process [65]. However, uncertainties in model parameters do result in the existence of uncertainties in model outputs. As a consequence, the bias is introduced into the calibrated model inherently. Moreover, the modelling errors can not be dealt with in conventional calibration. In fact, the uncertainties in model outputs can be amplified by ignoring modelling errors and uncertainties in biased data. Since uncertainties can not be quantified or reduced, conventional calibration can not compare the calibrated parameter combinations that yield similar minimal errors.

To deal with the disadvantages mentioned above, the calibration problem is considered in a probabilistic way in this thesis. Several studies on quantifying wind farm model uncertainties are then reviewed also for a better understanding. Jincheng Zhang and Xiaowei Zhao [95] conducted the LES with the FLORIS model using different yaw angles and calibrated the FLORIS model based on SOWFA data in 2019. They concluded that considering model parameter uncertainties can improve the prediction of wind field and power production. More specifically, the posterior FLORIS model can predict proper uncertainty features in the mixing zone and can minimize the the power fluctuation. However, the dynamic features were not involved. In 2020, Pascal Richter et al. [75] found that the quasi-Monte Carlo method is suitable for propagating uncertainties in a wind farm simulation because of its ideal convergence rates, low implementation loads and acceptable errors. However, only sensitivity analysis of the wind farm model was conducted and model calibration was not involved. Next, M.T. van Beek et al. [88] worked on optimizing the annual energy production for the Lillgrund wind farm based on the yaw-based wake steering strategy by calibrating a FLORIS model with Supervisory Control Data Acquisition (SCADA) data [9] in 2021. Both sensitivity analysis and model calibration were involved. However, there was still lack of dynamic features. Additionally, the Lillgrund wind farm is denser than other general wind farms, which means that there was room for exploring more under the larger wind turbine spacing conditions.

1-4-2 Objectives

The Gaussian FLORIDyn model referred from the work of Becker [17] takes into account spatially and time-wise heterogeneous field conditions besides the wake interaction effects in a wind farm. However, there is still room for improvement in accuracy of this model. The objectives of this thesis are investigating the influence of model parameter uncertainties on the prediction of the Gaussian FLORIDyn model and calibrating model parameters based on high-fidelity simulation data. Both of them are expected to be conducted in a computationally reasonable way.

1-4-3 Outline

The structure of this thesis is described as follows. Chapter 2 elaborates the necessary information about the Gaussian FLORIDyn model referred from Becker's work [17] in terms of 10 model parameters on which this research focuses and a novelty parameter to regulate the temporal dynamics of the model. Next, the uncertainty quantification framework setup for the research objectives is described with regard to the methods of sensitivity analysis and model calibration using Bayesian inference in Chapter 3. Chapter 4 presents a specific simulation case where the 3-Turbine case of the model elaborated in Chapter 2 is applied to the framework described in Chapter 3. Finally, the summary and potential of this research are discussed in Chapter 5.

Chapter 2

FLORIDyn Model

This chapter elaborates the necessary information about a Gaussian FLORIDyn model referred from the work of Becker [17] in 2020 which is the presupposition of this thesis. Section 2-1 elaborates how the model parameters of interest affect the wind farm model which captures the wake effects and corresponding dynamic features. The summary of model parameters elaborated in the previous section is then presented in Section 2-2.

2-1 Model parameters

It is important to note that the steady-state model parameters are simply an adaptation from the Gaussian FLORIS model presented by Bastankhah and Porté-Agel [15]. On the other hand, a model parameter included to capture the temporal dynamics is the novelty of the dynamic FLORIDyn model. The information about maturing the FLORIS model to the FLORIDyn model in terms of relevant equations is hereinafter elaborated according to [17].

2-1-1 Wind direction effects

The reduced wind speed u_{red} at the downstream turbine in a wind farm, per say, can be derived from Equation 2-1 in the FLORIS model:

$$u_{red} = u\,(1 - r)\,, \tag{2-1}$$

where u is the free wind speed[1] and r is the reduction factor[2]. To make the wake model depend on the wind direction, a vector $\mathbf{u} = [u_{x_0} \quad u_{y_0}]^{\mathrm{T}}$ is substituted for the scalar u in Equation 2-1. As a consequence, the reduced wind speed vector $\mathbf{u}_{red}$ can be calculated by Equation 2-2:

[1] The wind speed without the influence of wakes.
[2] The wake property of upstream turbines causing a velocity reduction.

$$\mathbf{u}_{red} = \mathbf{u}\,(1 - r)\,. \tag{2-2}$$

Assume that $\mathbf{r}_0 = [x_0 \quad y_0 \quad z_0]^{\mathrm{T}}$ denotes the position in the world coordinate system[3] and $\mathbf{r}_1 = [x_1 \quad y_1 \quad z_1]^{\mathrm{T}}$ indicates the position in the wake coordinate system[4]. As shown in Figure 2-1, the transformation from the wake coordinate system to the world coordinate system can be written as Equation 2-3:

$$\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{R}_{01}\mathbf{r}_1 = \mathbf{t}_0 + \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{r}_1, \tag{2-3}$$

where $\mathbf{t}_0$ denotes the position of the respective turbine in the world coordinate system, $\mathbf{R}_{01}$ indicates the rotational matrix and φ is the wind angle.

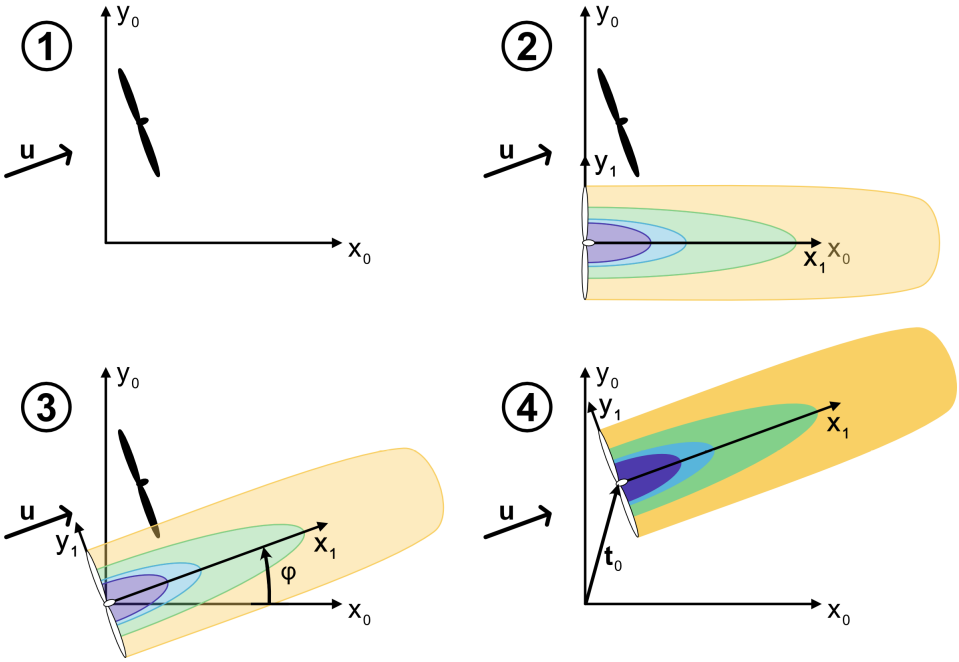


Figure 2-1: The transformation depicted in Equation 2-3: The first pane depicts that there is a turbine and a wind vector $\mathbf{u}$ in the world coordinate system. The second pane depicts there is a wake in the wake coordinate system required to be transformed into the world coordinate system. The third pane depicts all of the variables in the wake coordinate system are rotated by the wind angle φ. The fourth pane depicts the transformation is finished [17].

2-1-2 Estimation of reduction factors

The reduction factor r in Equation 2-2 can be estimated from the Gaussian FLORIS wake model consisting of the potential core zone, the near wake zone and the far wake zone [17].

[3]The grid is straight and the wake is bent [17].
[4]The wake is straight and the grid is bent [17].

As shown in Figure 2-2 [17], these three reduction zones blend into each other. Additionally, according to [15], the three reduction factors corresponding to the three reduction zones can be indicated by Equation 2-4, Equation 2-5 and Equation 2-6:

$$r_c = \left(1 - \sqrt{1 - C_T}\right), \tag{2-4}$$

$$r_{nw} = \left(1 - \sqrt{1 - C_T}\right) \exp\left[-\frac{1}{2}\left(\frac{|y_1 - \delta| - r_{pc_{y_1}}}{\sigma_{y,nw}}\right)^2\right] \exp\left[-\frac{1}{2}\left(\frac{|z_1 - z_h| - r_{pc_{z_1}}}{\sigma_{z,nw}}\right)^2\right], \tag{2-5}$$

$$r_{fw} = \left(1 - \sqrt{1 - C_T \frac{\cos\gamma}{8\sigma_y\sigma_z/D^2}}\right) \exp\left[-\frac{1}{2}\left(\frac{y_1 - \delta}{\sigma_{y,fw}}\right)^2\right] \exp\left[-\frac{1}{2}\left(\frac{z_1 - z_h}{\sigma_{z,fw}}\right)^2\right]. \tag{2-6}$$

In Equation 2-4, r_c denotes the potential core reduction factor and C_T is the thrust coefficient. In Equation 2-5, r_{nw} denotes the reduction factor acting on the transition from the potential core to the free stream in cross wind direction, δ is the deflection, z_h is the nacelle height, $r_{pc_{y_1}}$ is the potential core radius in y_1 direction and $r_{pc_{z_1}}$ is the potential core radius in z_1 direction. Additionally, $\sigma_{y,nw}$ and $\sigma_{z,nw}$ denote the standard deviations of the near wake in y_1 and z_1 directions. In Equation 2-6, r_{fw} denotes the far wake reduction factor and γ is the yaw angle[5]. Besides, $\sigma_{y,fw}$ and $\sigma_{z,fw}$ denote the standard deviations of the far wake in y_1 and z_1 directions.

Note that both of r_c and r_{nw} depict the near field characteristics and they are effective from the turbine position at $x_1 = 0$ to the end of the potential core at $x_1 = x_c$.

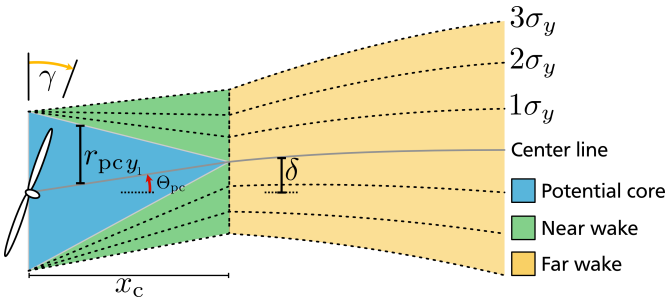


Figure 2-2: Wake shape depicted by wake zones and relevant parameters [17]

2-1-3 Potential core

The potential core is a cone shaped area with constant speed decrease [15]. The cone starts at the rotor plane with the y_1 width of $D\sqrt{u_R/u_c}\cos\gamma$ and the z_1 width of $D\sqrt{u_R/u_c}$ where D is the rotor diameter, u_R determined by Equation 2-7 denotes the wind speed at the rotor plane and u_c determined by Equation 2-8 denotes the wind speed in the potential core:

[5]Based on the definition of γ in the context of SOWFA simulations [17], the yaw angle is defined clockwise in contrast to all other angles which are defined counter-clockwise.

$$u_R = u\frac{C_T \cos\gamma}{2\left(1-\sqrt{1-C_T\cos\gamma}\right)}, \tag{2-7}$$

$$u_c = u\sqrt{1-C_T}. \tag{2-8}$$

The potential core radius constantly reduces from the turbine position at $x_1 = 0$ to the end of the potential core at $x_1 = x_c$. The radii in y_1 and z_1 direction of the cone shaped area can be deduced from Equation 2-9 and Equation 2-10:

$$r_{pc_{y_1}} = D\sqrt{\frac{C_T\cos\gamma}{2\left(1-\sqrt{1-C_T\cos\gamma}\right)\sqrt{1-C_T}}}\left(1-\frac{x_1}{x_c}\right)\cos\gamma, \tag{2-9}$$

$$r_{pc_{z_1}} = D\sqrt{\frac{C_T\cos\gamma}{2\left(1-\sqrt{1-C_T\cos\gamma}\right)\sqrt{1-C_T}}}\left(1-\frac{x_1}{x_c}\right). \tag{2-10}$$

Note that the constraint for these two equations is $0 \leq x_1 \leq x_c$ and x_c can be indicated in Equation 2-11:

$$x_c = \frac{\cos\gamma\left(1+\sqrt{1-C_T}\right)}{\sqrt{2}\left[\alpha^* I + \beta^*\left(1-\sqrt{1-C_T}\right)\right]}D. \tag{2-11}$$

2-1-4 Wake expansion

The standard deviation of Gaussian function is used to depict the wake expansion. Based on [15], the standard deviations of the near and far wake can be written as Equation 2-12 and Equation 2-13:

$$\left.\begin{array}{l}\sigma_{y,nw} = \frac{x_1}{x_c}\frac{D}{\sqrt{8}}\cos\gamma \\ \sigma_{z,nw} = \frac{x_1}{x_c}\frac{D}{\sqrt{8}}\end{array}\right\} \; x_1 < x_c, \tag{2-12}$$

$$\left.\begin{array}{l}\sigma_{y,fw} = (x_1 - x_c)\,k_y + \frac{D}{\sqrt{8}}\cos\gamma \\ \sigma_{z,fw} = (x_1 - x_c)\,k_z + \frac{D}{\sqrt{8}}\end{array}\right\} \; x_1 \geq x_c, \tag{2-13}$$

where k_y and k_z indicate the expansion factors in y_1 and z_1 directions.

Equation 2-12 and Equation 2-13 can be represented in a simplified way as follows:

$$\sigma_y = \max\left(x_1 - x_c,\ 0\right)k_y + \min\left(\frac{x_1}{x_c},\ 1\right)\frac{D}{\sqrt{8}}\cos\gamma, \tag{2-14}$$

$$\sigma_z = \max\left(x_1 - x_c,\ 0\right)k_z + \min\left(\frac{x_1}{x_c},\ 1\right)\frac{D}{\sqrt{8}}. \tag{2-15}$$

According to [36], the expansion factors k_y and k_z can be shown in Equation 2-16:

$$k_y = k_z = k_a I + k_b, \tag{2-16}$$

where I denotes the turbulence intensity given by Equation 2-17:

$$I = \sqrt{\sum_{i=1}^{n_T} I_{f,i}^2 + I_{amb}^2}. \tag{2-17}$$

Here I_{amb} denotes the ambient turbulence and I_f indicates the added turbulence by the n_T upstream turbines.

Consequently, $I_{f,i}$ can be written as Equation 2-18:

$$I_{f,i} = \frac{A_{overlap}}{A_{rotor,i}} \left[k_{f,a} a^{k_{f,b}} I_{amb}^{k_{f,c}} \left(\frac{x_1}{D} \right)^{k_{f,d}} \right], \tag{2-18}$$

where $A_{overlap}$ denotes the area of the rotor plane where a foreign wake overlaps, A_{rotor} indicates the area of the rotor plane and $k_{f,a}$, $k_{f,b}$, $k_{f,c}$ and $k_{f,d}$ are the weight factors of the foreign turbulence influence.

2-1-5 Wake deflection

According to [15], the total deflection for the near and far wake can be shown in Equation 2-19:

$$\begin{aligned} \delta = \theta_{pc} \max(x_1,\ x_c) + \frac{1}{2} \left[\operatorname{sign}(x_1 - x_c) + 1 \right] \frac{\theta_{pc}}{14.7} \sqrt{\frac{\cos\gamma}{k_y k_z C_T}} \left(2.9 + 1.3\sqrt{1 - C_T} - C_T \right) \\ \times \ln \left[\frac{\left(1.6 + \sqrt{C_T}\right) \left(1.6\sqrt{\frac{8\sigma_y \sigma_z}{D^2 \cos\gamma}} - \sqrt{C_T} \right)}{\left(1.6 - \sqrt{C_T}\right) \left(1.6\sqrt{\frac{8\sigma_y \sigma_z}{D^2 \cos\gamma}} + \sqrt{C_T} \right)} \right] D, \end{aligned} \tag{2-19}$$

where θ_{pc} denotes the deflection angle in the potential core.

Then, the deflection angle at the rotor θ can be estimated by Equation 2-20:

$$\theta \approx \frac{0.3\gamma}{\cos\gamma} \left(1 - \sqrt{1 - C_T \cos\gamma} \right). \tag{2-20}$$

It is important to note that θ_{pc} is equal to θ according to [15].

2-1-6 Power calculation

The power production can be estimated by Equation 2-21:

$$P = \eta \frac{1}{2} \rho u_{eff}^3 \pi \left(\frac{D}{2}\right)^2 C_P \cos\gamma^{p_p}. \tag{2-21}$$

The efficiency factor η was added in [45]. ρ is the air density and u_{eff} denotes the effective wind speed at the rotor plane. The power coefficient C_P[6] has been extended by [64] to also consider the yaw angle of the turbine. Besides, p_p denotes the factor for correcting the power coefficient under yawed operating conditions.

2-1-7 Temporal dynamics

To simulate the propagation of variations at the turbine through the wake over time, Observation Points (OPs) are introduced. A yaw angle variation of the turbine and its propagation through the wake with OPs are indicated in Figure 2-3 [17].

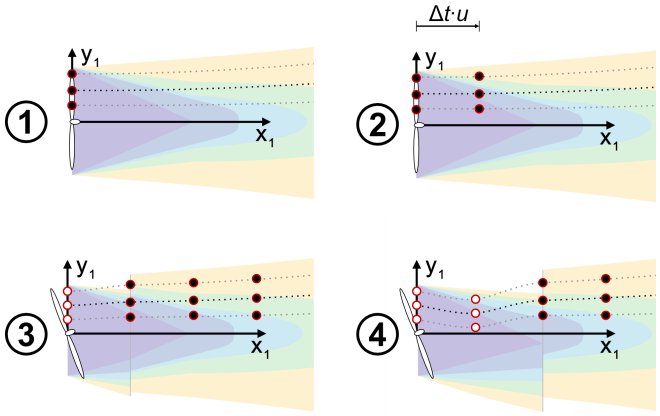


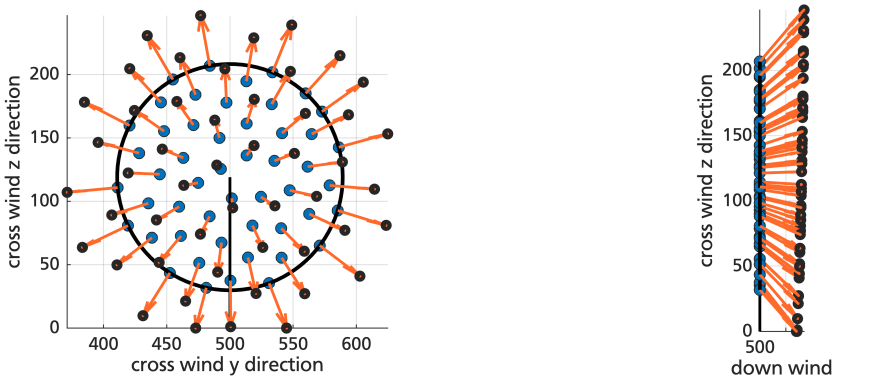
Figure 2-3: The processes that OPs transport state changes of the turbine through the wake: In the first pane, OPs store the information on the wake at the rotor plane. Their positions in the wake are determined according to the wake width. The dotted lines denote the paths determined by the wake width. A chain consists of OPs on the same dotted line. In the second pane, the new OPs (storing the same information as the original OPs at the rotor plane) and the original OPs have travelled further downstream at the wind speed. In the third pane, there is a variation of the wake shape resulting from the yaw angle of the turbine. As a consequence, OPs store different information and the color of these new OPs and the corresponding (background) wake are also different. In the fourth pane, new OPs storing the information after yawing the turbine follow the new paths [17].

Chain consisting of OPs Specifically, there are totally n_{OP} OPs introduced at the rotor plane at time step k and they travel downstream. Each OP has a relative cross wind position ν_y and ν_z which depends on the wake width to cover the wake plane regularly. According to

[6]The power coefficient C_P depends on the rotor axial induction factor a and the yaw angle of the rotor γ.

[17], to cover the majority of the reduced wind speed area, the wake widths are assumed as $6\sigma_y$ and $6\sigma_z$. The crosswind position is indicated in Equation 2-22:

$$\left.\begin{aligned} y_1 &= \nu_y \left[6\sigma_y + \max\left(r_{pcy_1},\ 0\right)\right] \\ z_1 &= \nu_z \left[6\sigma_z + \max\left(r_{pcz_1},\ 0\right)\right] \end{aligned}\right\}\ \nu_y,\ \nu_z \in [-0.5,\ 0.5]\,. \tag{2-22}$$



(a) $(y_1,\ z_1)$ plane (b) $(x_1,\ z_1)$ plane

Figure 2-4: OP distribution across the wake cross section: The blue dots denote the OPs initialized across the rotor plane depicted by the black circle. The orange arrows depict during one time step OPs move to their new positions which can be determined by Equation 2-22 [17].

The OP distribution across the wake cross section from the sunflower algorithm [90] is shown in Figure 2-4 [17]. The Voronoi method in Matlab [7] which can return polygons surrounding the OPs is used to obtain the approximation of the area indicated by an OP. The areas of the polygons are calculated, normalized and stored as weights in the vector w. Meanwhile, an outer circle is added for the calculation to limit the outer areas. Based on [17], the effective wind speed at the rotor plane is derived from the product of the vector w and the speed of the OPs at the rotor plane. Next, u_{eff} is determined by adding the products together. To be able to do this, Equation 2-23 is used to initialize the OPs at the rotor plane by regarding the rotor area as the wake width. After calculating the downstream wind step, Equation 2-22 is used to estimate their new cross wind position:

$$\begin{aligned} y_1 &= \nu_y D \cos\gamma, \\ z_1 &= \nu_z D. \end{aligned} \tag{2-23}$$

Under constant conditions, a chain consists of OPs created at the same relative position in the wake which have the same path. The computational load from the information on a chain is less than that from the information on each OP [17].

It was observed that the speed of an OP that affects the state change of the turbine arrive slower at the downstream turbine in the FLORIDyn model simulations [17]. Meanwhile, when a turbine state varies, the early recovering parts of the wake down field (the outer parts) response faster than the inner parts, which can result in the overlap parts in the wake. To deal with these two issues, Taylor's frozen turbulence hypothesis [85] is used by considering

that the OPs travel at the freestream wind speed rather than the speed they represent. The assumption describes large eddies can be regarded frozen as a whole and travel at the average wind speed. According to [78], a research conducted by Schlipf et al. in 2010 shows there is a good agreement between the hypothesis and measured data for eddies at a scale relevant to wind turbines.

There are two alternate ways for representing the speed of an OP. One is Equation 2-24 to denote the travelling speed of the OP is the same as the speed it represents in the wake, the other is Equation 2-25 to denote the travelling speed of the OP is u_{OP} while the speed it represents in the wake is $u_{OP,eff}$ [17].

$$u_{OP} = d * |\mathbf{u}|\,, \tag{2-24}$$

$$u_{OP} = d * |\mathbf{u}|\,(1 - r)\,, \tag{2-25}$$

where d is a factor added to regulate the temporal dynamics by scaling the advection speed.

2-2 Summary

As elaborated above, in this Gaussian FLORIDyn model, the wake effects are captured by the Gaussian FLORIS model and temporal dynamics are captured using OPs. As a consequence, eight wake model parameters adapted from the Gaussian FLORIS model presented by Bastankhah and Porté-Agel [15] are taken into account. Besides, the efficiency parameter η scaling the power outputs of the wind turbine and the power coefficient parameter p_p scaling the yaw angle γ are also included for uncertainty quantification. The mean values for both of them are tuned in [17]. Last, the temporal dynamics parameter d scaling the advection speed is also considered. Table 2-1 presents all model parameters for uncertainty quantification in this research and Table 2-2 shows the mean values for the model parameters and corresponding sources.

Parameter	Description
α^*, β^*	Weight constraints in the potential core length calculation
k_a, k_b	Weight constraints in the wake expansion calculation
$k_{f,a}$, ..., $k_{f,d}$	Parameters in the foreign turbulence effect calculation
η	Efficiency parameter scaling power outputs
p_p	Power coefficient parameter scaling the yaw angle
d	Temporal dynamics parameter scaling the advection speed

Table 2-1: Objective model parameters

	α^*	β^*	k_a	k_b	$k_{f,a}$	$k_{f,b}$	$k_{f,c}$	$k_{f,d}$	d	η	p_p
Mean value	2.32	0.154	0.38371	0.003678	0.73	0.8325	0.0325	-0.32	1	0.8572	2.2
Source	[15]	[15]	[15]	[15]	[87]	[87]	[87]	[87]	[30]	[17]	[17]

Table 2-2: Mean values for the model parameters

Chapter 3

Uncertainty Quantification

The Gaussian FLORIDyn model elaborated in Chapter 2 will be applied to uncertainty quantification by taking into account the model parameters summarized in Table 2-1. A major drawback of uncertainty quantification, further explained in the later sections, is high computational costs mainly caused by the associated Markov Chain Monte Carlo (MCMC) sampling algorithm. To deal with it, a surrogate model will be trained to replace the original FLORIDyn model before conducting the related uncertainty quantification research. Compared to a full MCMC simulation with the FLORIDyn model, the surrogate model lacks accuracy, which is a trade-off for simplicity and faster convergence. Besides, sensitivity analysis of the FLORIDyn model will be performed using the trained surrogate model to investigate how sensitive the model outputs are to each parameter of interest.

In this chapter, Section 3-1 first describes the basic information about surrogate modelling and then elaborates two types of approaches for constructing surrogate models assessed in the simulation conducted in Chapter 4. Section 3-2 first introduces the basic information about sensitivity analysis and next elaborates the method using Sobol' indices which is a global method used in the simulation conducted in Chapter 4. Section 3-3 first describes Bayesian inference and then elaborates the method used in the simulation conducted in Chapter 4 for calibrating the FLORIDyn model. Finally, the procedures to conduct uncertainty quantification of the FLORIDyn model are summarized in Section 3-4.

3-1 Surrogate modelling

Surrogate modelling is an engineering method to replace the outcomes of interest difficultly measured or computed with outcomes of an approximate model constructed by modelling the response to a limited number of intelligently chosen data in the simulation. Generally, experiments or simulations are necessary in most engineering problems. However, in many practical cases, even a single simulation is computationally expensive. Some kind of studies like design exploration, sensitivity analysis, and model calibration can be too time consuming to conduct since they may require hundreds or millions of simulation evaluations. Constructing

models which behave similarly to the original model in the simulation is an option to decrease computational expenses. Such models which can be computationally cheaper to evaluate are known as surrogate models [6].

In the current research, two types of approaches for constructing surrogate models which are Kriging and Polynomial Chaos Expansion (PCE) will be assessed and one of them will be selected to construct the surrogate model for the following sensitivity analysis and Bayesian calibration.

3-1-1 Kriging

Kriging, also known as Gaussian process regression, is a stochastic algorithm to construct new data points based on the Gaussian process governed by prior covariance [57]. To elaborate the principle of Kriging, some mathematical notations are introduced in the following content.

The model output $\mathcal{M}(\boldsymbol{x})$ is assumed as a realization of a Gaussian process indexed by the model inputs $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{X}} \subset \mathbb{R}^M$. Based on [77], the Kriging-based surrogate model can be presented as Equation 3-1:

$$\mathcal{M}^K(\boldsymbol{x}) = \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}) + \sigma^2 Z(\boldsymbol{x}, \zeta), \tag{3-1}$$

where $\mathcal{M}^K(\boldsymbol{x})$ represents the Kriging surrogate model evaluation. $\boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x})$ is named as the trend which is the mean value of the Gaussian process. It is comprised of P arbitrary functions $\boldsymbol{f} = \{f_1 f_2 ... f_P\}^T$ and corresponding coefficients $\boldsymbol{\lambda}^T = \{\lambda_1 \lambda_2 ... \lambda_P\}$. σ^2 is the variance of the Gaussian process. $Z(\boldsymbol{x}, \zeta)$ stands for a stationary Gaussian process whose mean value is 0 and variance is the unit-variance. ζ indicates the underlying probability space in terms of a correlation function R which will be elaborated later.

Trends

As described before, the trend represents the mean value of the Kriging-based surrogate model. In 2011, Dubourg [34] summed up 3 types of Kriging according to different trend formulas, which are simple Kriging, ordinary Kriging and universal Kriging respectively.

The trend of simple Kriging is described as Equation 3-2:

$$\boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}) + ... + f_P(\boldsymbol{x}), \tag{3-2}$$

where $\boldsymbol{f}(\boldsymbol{x})$ are arbitrary but fully specified functions and $\boldsymbol{\lambda}$ is the vector whose elements are all 1.

The trend of ordinary is indicated as Equation 3-3:

$$\boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}) = \lambda_0 f_0(\boldsymbol{x}) = \lambda_0, \tag{3-3}$$

where $f_0(\boldsymbol{x})$ is conventionally assumed as 1.

The trend of universal Kriging is assumed as a linear combination of prescribed arbitrary functions and is presented as Equation 3-4:

$$\boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}) = \lambda_0 f_0(\boldsymbol{x}) + \lambda_1 f_1(\boldsymbol{x}) + ... + \lambda_P f_P(\boldsymbol{x}). \tag{3-4}$$

From [57], besides the trends mentioned above, there are also other types of trends can be obtained using polynomial basis such as the linear trend indicated as $\lambda_0 + \sum_{i=1}^{M} \lambda_i x_i$ and the quadratic trend indicated as $\lambda_0 + \sum_{i=1}^{M} \lambda_i x_i + \sum_{i=1}^{M} \sum_{j=1}^{M} \lambda_{ij} x_i x_j$. The current research will use the linear trend.

Correlation functions

The correlation function multiplied by σ^2 in Equation 3-1 is to the covariance function [93]. The correlation function is used to depict how similar the new model outputs are to the original model outputs.

According to [57], the correlation function can be indicated as $R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ has the same dimensions as the model input and one element of $\boldsymbol{\theta}$ is for one model input dimension. There are 4 correlation families which are the linear correlation family, the exponential correlation family, the Gaussian correlation family and the Matérn correlation family respectively. To visualize how the new model outputs are parameterized by $\boldsymbol{\theta}$, all of 4 correlation families will be introduced in the one-dimensional space next.

The linear correlation family can be denoted as Equation 3-5[57]:

$$R(x, x'; \theta) = \max\left(0, 1 - \frac{|x - x'|}{\theta}\right). \tag{3-5}$$

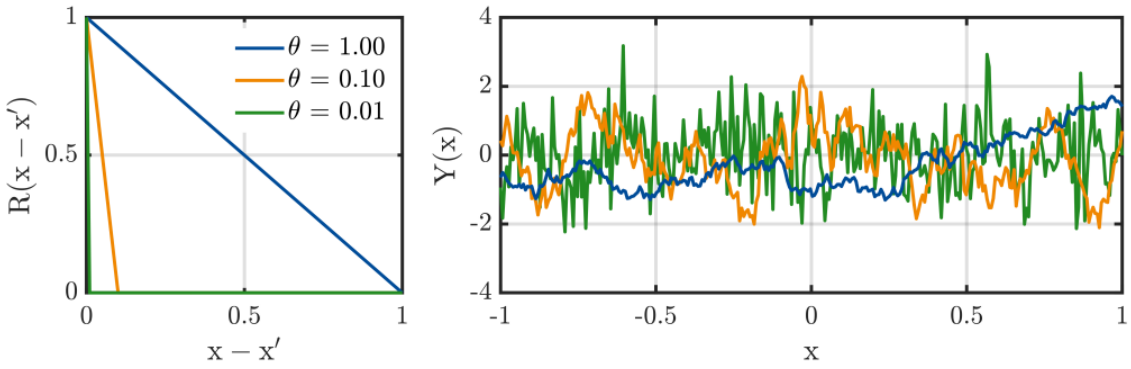


Figure 3-1: Equation 3-5 and sample paths obtained from the corresponding Gaussian process whose mean value is 0 and variance is the unit-variance with different θ [57].

The exponential correlation family can be described as Equation 3-6[57]:

$$R(x, x'; \theta) = \exp\left[-\frac{|x - x'|}{\theta}\right]. \tag{3-6}$$

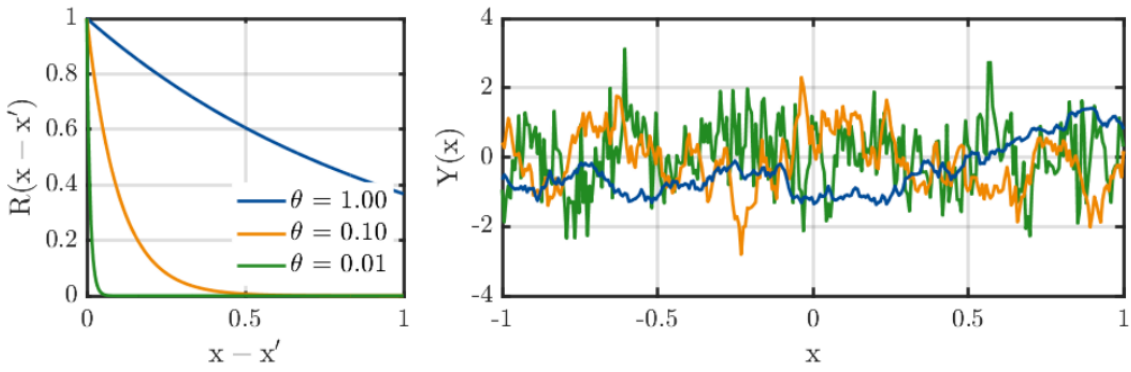


Figure 3-2: Equation 3-6 and sample paths obtained from the corresponding Gaussian process whose mean value is 0 and variance is the unit-variance with different θ [57].

The Gaussian correlation family can be presented as Equation 3-7[57]:

$$R\left(x, x'; \theta\right) = \exp\left[-\frac{1}{2}\left(\frac{|x - x'|}{\theta}\right)^2\right]. \tag{3-7}$$

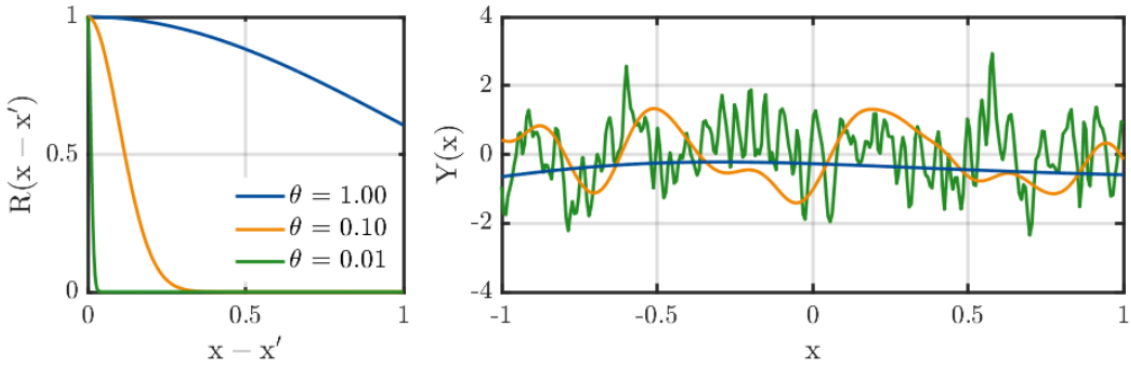


Figure 3-3: Equation 3-7 and sample paths obtained from the corresponding Gaussian process whose mean value is 0 and variance is the unit-variance with different θ [57].

The Matérn correlation family can be generally indicated as Equation 3-8[57]:

$$R\left(x, x'; \theta, v\right) = \frac{1}{2^{v-1}\Gamma\left(v\right)}\left(2\sqrt{v}\frac{|x - x'|}{\theta}\right)^v \mathcal{K}_v\left(2\sqrt{v}\frac{|x - x'|}{\theta}\right), \tag{3-8}$$

where $v \geq \frac{1}{2}$ represents the shape parameter, $\Gamma\left(v\right) = \int_0^\infty t^{v-1}e^{-t}dt$ and $\mathcal{K}_v$ denotes the modified Bessel function of the second kind [11].

According to [57], the Matérn correlation families with $v = \frac{3}{2}$ and $v = \frac{5}{2}$ are the most widely used and can be described as Equation 3-9 and Equation 3-10:

$$R\left(x, x'; \theta, v = \frac{3}{2}\right) = \left(1 + \sqrt{3}\frac{|x - x'|}{\theta}\right)\exp\left[-\sqrt{3}\frac{x - x'}{\theta}\right]. \tag{3-9}$$

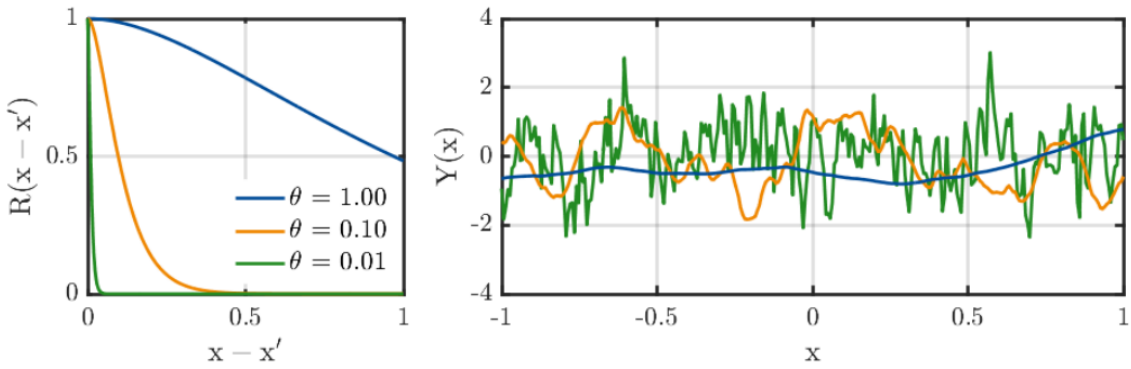


Figure 3-4: Equation 3-9 and sample paths obtained from the corresponding Gaussian process whose mean value is 0 and variance is the unit-variance with different θ [57].

$$R\left(x, x'; \theta, v\frac{5}{2}\right) = \left(1 + \sqrt{5}\frac{|x - x'|}{\theta} + \frac{5}{3}\left(\frac{|x - x'|}{\theta}\right)^2\right) \exp\left[-\sqrt{5}\frac{x - x'}{\theta}\right]. \tag{3-10}$$

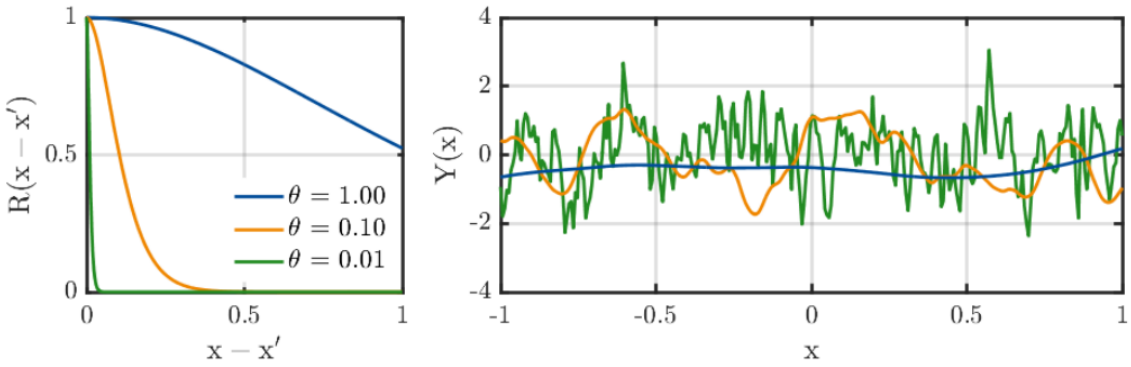


Figure 3-5: Equation 3-10 and sample paths obtained from the corresponding Gaussian process whose mean value is 0 and variance is the unit-variance with different θ [57].

When dealing with a multi-dimensional system, it is necessary to build the multi-dimensional correlation function using one of 4 one-dimensional correlation families introduced above [57]. The current research will use the exponential correlation family to build the eleven-dimensional ellipsoidal correlation function indicated as Equation 3-11[73]. To do so, the term $\frac{|x-x'|}{\theta}$ shown in Equation 3-6 needs to be substituted by h (same for the other correlation families when building the corresponding multi-dimensional ellipsoidal correlation function) according to [57].

$$R\left(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}\right) = R\left(h\right), \quad h = \sqrt{\sum_{i=1}^{M}\left(\frac{x_i - x_i'}{\theta_i}\right)^2}. \tag{3-11}$$

Estimation

The unknown $\boldsymbol{\theta}$ parametrizing the correlation function is necessary for training the Kriging surrogate model. To estimate $\boldsymbol{\theta}$, the maximum-likelihood method will be used in the current research. The related elaboration is shown next.

Maximum-likelihood (ML) estimation maximizes the likelihood of the model observations $\boldsymbol{\mathcal{Y}}$ by tuning the Kriging parameters $\boldsymbol{\lambda}$, σ^2 and $\boldsymbol{\theta}$. Based on [57], the likelihood function can be described as Equation 3-12:

$$\mathcal{L}_K\left(\boldsymbol{\lambda}, \sigma^2, \boldsymbol{\theta}; \boldsymbol{\mathcal{Y}}\right) = \frac{\left(\det \boldsymbol{R}\right)^{-\frac{1}{2}}}{\left(2\pi\sigma^2\right)^{\frac{N}{2}}} \exp\left[-\frac{1}{2\sigma^2}\left(\boldsymbol{\mathcal{Y}} - \boldsymbol{F}\boldsymbol{\lambda}\right)^T \boldsymbol{R}^{-1}\left(\boldsymbol{\mathcal{Y}} - \boldsymbol{F}\boldsymbol{\lambda}\right)\right]. \tag{3-12}$$

N represents the number of input points which can be used to describe the experimental design $\boldsymbol{\mathcal{X}} = \left[\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(N)}\right]^T$. Therefore, the corresponding model observations can be denoted as $\boldsymbol{\mathcal{Y}} = \left[y^{(1)} = \mathcal{M}\left(\boldsymbol{x}^{(1)}\right), ..., y^{(N)} = \mathcal{M}\left(\boldsymbol{x}^{(N)}\right)\right]^T$. $\boldsymbol{F}$ indicates the design matrix of the trend whose elements can be denoted as $F_{ij} = f_j\left(\boldsymbol{x}^{(i)}\right), \quad i = 1, ..., N; \quad j = 0, ..., P$. $\boldsymbol{R}$ is the correlation matrix whose elements can be denoted as $R_{ij} = R\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}; \boldsymbol{\theta}\right), \quad i, j = 1, ..., N$.

Based on [77] and [34], the estimates of $\boldsymbol{\lambda}$ and σ^2 described as Equation 3-13 and Equation 3-14 can provide the maximum value of $\mathcal{L}_K$.

$$\widehat{\boldsymbol{\lambda}} = \left(\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{\mathcal{Y}}, \tag{3-13}$$

$$\widehat{\sigma}^2 = \frac{1}{N}\left(\boldsymbol{\mathcal{Y}} - \boldsymbol{F}\boldsymbol{\lambda}\right)^T \boldsymbol{R}^{-1}\left(\boldsymbol{\mathcal{Y}} - \boldsymbol{F}\boldsymbol{\lambda}\right). \tag{3-14}$$

Next, $\boldsymbol{\theta}$ can be determined by solving the optimization problem presented as follows[57]:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}} \left[-\log \mathcal{L}_K\left(\boldsymbol{\theta}; \boldsymbol{\mathcal{Y}}\right)\right]. \tag{3-15}$$

Given Equation 3-12, Equation 3-13 and Equation 3-14, Equation 3-15 can be rewritten as following[57]:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}} \frac{1}{2}\left[\log\left(\det \boldsymbol{R}\right) + N \log\left(2\pi\sigma^2\right) + N\right]. \tag{3-16}$$

Posteriori error estimation

Based on [57], the leave-one-out (LOO) cross-validation (CV) error or the validation error will be used for investigating the accuracy of the Kriging-based surrogate model prediction. The former one can be determined according to the initial experimental design $\boldsymbol{\mathcal{X}}$ and the corresponding model output $\boldsymbol{\mathcal{Y}} = \mathcal{M}\left(\boldsymbol{\mathcal{X}}\right)$ and described as[57]:

$$\epsilon_{LOO} = \frac{1}{N}\left[\frac{\sum_{i=1}^{N}\left(\mathcal{M}\left(\boldsymbol{x}^{(i)}\right) - \mu_{\widehat{y},\backslash\boldsymbol{x}^{(i)}}\left(\boldsymbol{x}^{(i)}\right)\right)^2}{\mathrm{Var}\left[\boldsymbol{\mathcal{Y}}\right]}\right], \tag{3-17}$$

where $\mu_{\widehat{y},\backslash\boldsymbol{x}^{(i)}}\left(\boldsymbol{x}^{(i)}\right)$ indicates the mean of prediction at $\boldsymbol{x}^{(i)}$ produced by the surrogate model trained with the other points of $\boldsymbol{\mathcal{X}}$ except $\boldsymbol{x}^{(i)}$.

The latter one requires an independent set of observations $\boldsymbol{\mathcal{O}}_{\mathrm{val}} = \left\{\left(\boldsymbol{x}_{\mathrm{val}}^{(1)}, y_{\mathrm{val}}^{(1)}\right), ..., \left(\boldsymbol{x}_{\mathrm{val}}^{(N_{\mathrm{val}})}, y_{\mathrm{val}}^{(N_{\mathrm{val}})}\right)\right\}$ to be computed as[57]:

$$\epsilon_{\mathrm{val}} = \frac{N_{\mathrm{val}}-1}{N_{\mathrm{val}}}\left[\frac{\sum_{i=1}^{N_{\mathrm{val}}}\left(\mathcal{M}\left(\boldsymbol{x}_{\mathrm{val}}^{(i)}\right) - \mathcal{M}^{K}\left(\boldsymbol{x}_{\mathrm{val}}^{(i)}\right)\right)^2}{\sum_{i=1}^{N_{\mathrm{val}}}\left(\mathcal{M}\left(\boldsymbol{x}_{\mathrm{val}}^{(i)}\right) - \mu_{y_{\mathrm{val}}}\right)^2}\right], \tag{3-18}$$

where $y_{\mathrm{val}}^{(i)} = \mathcal{M}\left(\boldsymbol{x}_{\mathrm{val}}^{(i)}\right)$ is the original model output and the sample mean of the independent validation output set $\mu_{y_{\mathrm{val}}}$ can be denoted as follows[57]:

$$\mu_{y_{\mathrm{val}}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{M}\left(\boldsymbol{x}_{\mathrm{val}}^{(i)}\right). \tag{3-19}$$

3-1-2 Polynomial chaos expansion (PCE)

The polynomial chaos expansion (PCE) method is used to obtain the approximation according to the computational model's spectral representation on a properly constructed polynomial basis [62]. To do so, the variance of the computational model output needs to be finite:

$$\begin{aligned}\mathrm{Var}(y) &= \mathbb{E}\left[(y-\mathbb{E}(y))^2\right] \\ &= \mathbb{E}\left(y^2\right) - [\mathbb{E}(y)]^2 < +\infty,\end{aligned} \tag{3-20}$$

where $y = \mathcal{M}(\boldsymbol{x}),\ y \in \mathbb{R},\ \boldsymbol{x} \in \mathbb{R}^M$ is the computational model output and $\boldsymbol{x}$ is a random model input vector described by the joint probability density function $f_{\boldsymbol{x}}$.

Therefore, the condition shown in Equation 3-20 can be rewritten as:

$$\mathbb{E}\left(y^2\right) = \int_{\mathcal{D}_{\boldsymbol{x}}} \mathcal{M}(\boldsymbol{x})^2 f_{\boldsymbol{x}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} < +\infty. \tag{3-21}$$

Next, based on [62], the polynomial chaos expansion of the computational model can be defined as:

$$y = \mathcal{M}(\boldsymbol{x}) = \sum_{\boldsymbol{\kappa}\in\mathbb{N}^M} q_{\boldsymbol{\kappa}}\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}), \tag{3-22}$$

where $\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})$ denotes multivariate polynomials orthonormal w.r.t. $f_{\boldsymbol{x}}$, $\boldsymbol{\kappa} \in \mathbb{N}^M$ indicates the multi-index identifying the component of $\Psi_{\boldsymbol{\kappa}}$ and $q_{\boldsymbol{\kappa}} \in \mathbb{R}$ represents the corresponding coefficient.

In practice, the truncated polynomial chaos expansion is introduced to truncate the sum in Equation 3-22 into a finite sum corresponding to all polynomials in M input variables of total degree not more than p as follows[62]:

$$\mathcal{M}(\boldsymbol{x}) \approx \mathcal{M}^{PC}(\boldsymbol{x}) = \sum_{\boldsymbol{\kappa} \in \mathcal{A}^{M,p}} q_{\boldsymbol{\kappa}} \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}), \tag{3-23}$$

where $\mathcal{A}^{M,p} = \left\{\boldsymbol{\kappa} \in \mathbb{N}^M : |\boldsymbol{\kappa}| \leq p\right\}$ indicates the set of chosen multi-indices of $\Psi_{\boldsymbol{\kappa}}$ and its cardinality is indicated as follows:

$$\begin{aligned} \operatorname{card}\left(\mathcal{A}^{M,p}\right) \equiv P &= \binom{M+p}{p} \\ &= \frac{(M+p)!}{M!p!} \end{aligned} \tag{3-24}$$

Polynomial basis

Based on [62], to establish the polynomial basis $\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})$, a set of univariate orthonormal polynomials $\psi_k^{(i)}(x_i)$ are introduced and defined by an inner product as follows:

$$\left\langle \psi_j^{(i)}(x_i), \psi_k^{(i)}(x_i) \right\rangle \overset{\text{def}}{=} \int_{\mathcal{D}_{x_i}} \psi_j^{(i)}(x_i)\, \psi_k^{(i)}(x_i)\, f_{x_i}(x_i)\, \mathrm{d}x_i = \delta_{jk}, \tag{3-25}$$

where x_i represents one element of the input vector $\boldsymbol{x}$ of the computational model and x_i is orthogonal w.r.t. $\psi_j^{(i)}(x_i)$, $\psi_k^{(i)}(x_i)$, ..., which belong to the corresponding polynomial family, j and k indicate the corresponding polynomial degree, $f_{x_i}(x_i)$ denotes the probability density function describing x_i and the Kronecker delta δ_{jk} can be written as follows:

$$\delta_{jk} = \begin{cases} 0 & \mathrm{j} \neq \mathrm{k} \\ 1 & \mathrm{j} = \mathrm{k} \end{cases}. \tag{3-26}$$

Next, the tensor product of the univariate polynomials can be used to obtain the multivariate polynomial $\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})$ as follows:

$$\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) \overset{\text{def}}{=} \prod_{i=1}^{M} \psi_{\kappa_i}^{(i)}(x_i). \tag{3-27}$$

The UQLab software[63] used to conduct the simulation in the current research will by default select the mathematical expression for $\psi_{\kappa_i}^{(i)}(x_i)$ based on the type of $f_{x_i}(x_i)$ using the pairing table from [83]. For example, the random input variables used in the simulation in Chapter 4 are subject to the Uniform distribution, the corresponding univariate orthonormal polynomial family is then selected by default as Legendre whose details can be found in [94].

Coefficients estimation

In the current research, the experimental design set comprised of samples of random model inputs and the corresponding model outputs is post-processed to estimate $q_{\boldsymbol{\kappa}}$ given the relative $\Psi_{\boldsymbol{\kappa}}$. Based on [62], there are two types of methods to do so which are the projection method and the regression method respectively. The former one calculates the numerical integration according to the basis functions' orthogonality and the latter one uses the standard linear regression approaches to work out the system of linear equations which is derived by formulating Equation 3-22. The current research uses the Least Angle Regression (LAR) algorithm which is a kind of sparse regression approach belonging to the second type. The LAR algorithm also involves the Ordinary Least-Squares (OLS) regression approach. More details about the regression method used in Chapter 4 are elaborated in the following.

Least-squares regression first establishes a least-square problem by transforming Equation 3-22 into the sum of Equation 3-23 and a residual as indicated in Equation 3-28 according to [18]. Next, the vector $\boldsymbol{q}$ consisting of coefficients can be estimated by solving the least-square problem as shown in Equation 3-29[18].

$$\begin{aligned} y = \mathcal{M}(\boldsymbol{x}) &= \mathcal{M}^{PC}(\boldsymbol{x}) + \varepsilon_P \\ &= \sum_{\boldsymbol{\kappa} \in \mathcal{A}^{M,p}} q_{\boldsymbol{\kappa}} \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) + \varepsilon_P \\ &= \boldsymbol{q}^T \boldsymbol{\Psi}(\boldsymbol{x}) + \varepsilon_P, \end{aligned} \tag{3-28}$$

$$\widehat{\boldsymbol{q}} = \arg\min_{\boldsymbol{q} \in \mathbb{R}^P} \mathbb{E}\left[\left(\boldsymbol{q}^T \boldsymbol{\Psi}(\boldsymbol{x}) - \mathcal{M}(\boldsymbol{x})\right)^2\right], \tag{3-29}$$

where $\boldsymbol{q} = \underbrace{[q_{\boldsymbol{\kappa}}, ...]^T}_{P \text{ coefficients}}$, $\boldsymbol{\Psi}(\boldsymbol{x}) = \underbrace{[\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}), ...]^T}_{P \text{ multivariate polynomials}}$ and ε_P represents the truncation error.

To work out the least-square problem shown in Equation 3-29, the Ordinary Least Squares (OLS) approach is next introduced. Based on [62], the OLS solution can be obtained using a vector consisting of samples of the experimental design inputs $\boldsymbol{\mathcal{X}} = \left[\boldsymbol{x}^{(1)} ... \boldsymbol{x}^{(N)}\right]^T$ and a vector consisting of the corresponding experimental design outputs $\boldsymbol{\mathcal{Y}} = \left[y^{(1)} = \mathcal{M}\left(\boldsymbol{x}^{(1)}\right) ... y^{(N)} = \mathcal{M}\left(\boldsymbol{x}^{(N)}\right)\right]^T$, which is written as follows:

$$\widehat{\boldsymbol{q}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{\mathcal{Y}}, \tag{3-30}$$

where the element of the regression matrix $\boldsymbol{A}$ can be denoted as[62]:

$$A_{ij} = \Psi_j\left(\boldsymbol{x}^{(i)}\right), \; i = 1, ..., N; \; j = 0, ..., P-1. \tag{3-31}$$

It is important to note that the OLS approach can be only applied to the overdetermined system of linear equations, which means P can not be larger than N.

Sparse PCE: Least angle regression adds a penalty term $\|\boldsymbol{q}\|_1 = \sum_{j=0}^{P-1} |q_j|$ to the solution of the least-square problem shown in Equation 3-29 to work out the underdetermined system of linear equations [62]. The new solution is rewritten as follows:

$$\widehat{\boldsymbol{q}} = \arg\min_{\boldsymbol{q}\in\mathbb{R}^P} \left\{ \mathbb{E}\left[\left(\boldsymbol{q}^T \boldsymbol{\Psi}\left(\boldsymbol{x}\right) - \mathcal{M}\left(\boldsymbol{x}\right) \right)^2 \right] + \left\| \boldsymbol{q} \right\|_1 \right\}. \tag{3-32}$$

Based on [22], the penalty term can restrict the solution to be low-rank and the sparse PCE-based surrogate model can be trained using a small experimental design without sacrificing the accuracy. To achieve that, the LAR algorithm is widely used.

Based on [22], the main principle of the LAR algorithm is to iteratively move regressors from a candidate set to an active set. The next regressor is selected by considering its correlation with the current residual. For every iteration, every active regressor is imposed to be equicorrelated with the current residual to obtain the optimal set of regression coefficients for the current active set. The detailed procedures of the LAR algorithm are shown in Table 3-1.

Initialization	Coefficient: $\boldsymbol{q} = \mathbf{0}$; Candidate set: $\Psi_{\boldsymbol{\kappa}}$; Active set: $\varnothing$; Residual: $r_0 = y^{(i)}$.
Iterative Algorithm	Step 1: Search the regressor $\Psi_{\boldsymbol{\kappa}}$ which is the most correlated with the current residual; Step 2: Adjust all coefficients corresponding to the regressors in the current active set and the regressor at the first step towards their least-square values until their regressors have the same correlation with the residual as some other regressor in the candidate set has (This regressor in the candidate set will be the most correlated with the residual for the first step of the next iteration.); Step 3: Compute and store ϵ_{LOO} for the current iteration; Step 4: Update all active coefficients and move $\Psi_{\boldsymbol{\kappa}}$ at the first step from the candidate set to the active set; Step 5: Repeat the previous steps until the size of the active set equals $\min\left(P,\ N-1\right)$;

Table 3-1: Procedures of the LAR algorithm

Due to the fact that the number of iterations plays a role in the LAR algorithm and the regressors defined in the LAR algorithm are not constant because of the first step, it is more difficult to compute LOO errors than the OLS approach in which N surrogate models are not necessary based on Equation 3-37. To deal with it, the hybrid-LAR step is introduced at the end of every LAR iteration. The hybrid-LAR step adds a constant regressor to the chosen basis and uses the OLS approach to compute the corresponding coefficients and ϵ_{LOO}. Overall, a series of sets of multivariate polynomials are obtained by the LAR iterations, while the associated surrogate models are trained by the OLS approach. Finally, the surrogate model with the lowest ϵ_{LOO} is selected.

Additionally, an early stop criterion is introduced to deal with the situation with many regressors in practice. As shown in Figure 3-6, mostly, $\epsilon_{LOO}^{(j)}$ evolves smoothly and convexly as the number of LAR iterations increases. Therefore, the early stop criterion can be set up to stop adding regressors when ϵ_{LOO} exceeds its minimum. According to [62], the number

of iterations with the suitable early stop criterion is at least 10% of the potential maximum number of iterations.

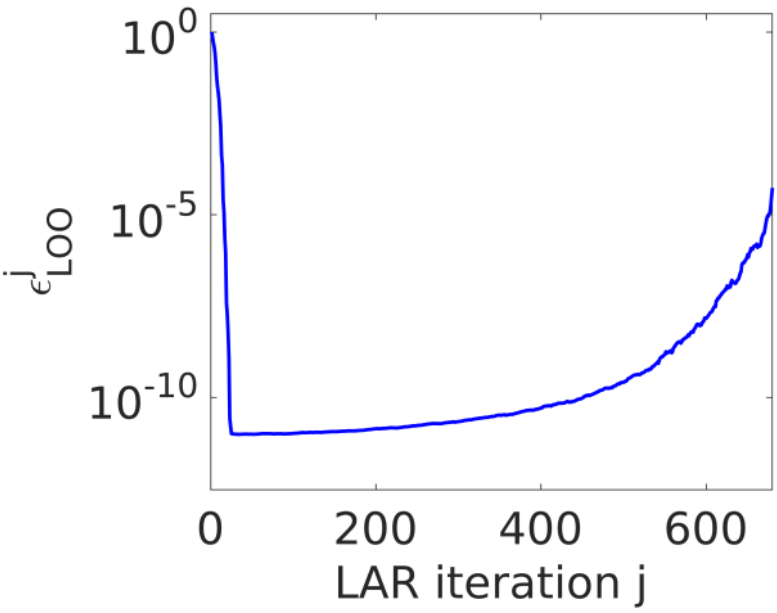


Figure 3-6: The typical behaviour of ϵ_{LOO} as the number of iterations increases in most practical cases [62].

Table 3-2 summarizes the advantages and disadvantages of the Kriging-based approach and the PCE-based approach.

	Pros	Cons
Kriging	1. The sampling errors have little influence on it; 2. The results can be unbiased when the sample size is large enough; 3. It can offer useful results for statistical inference;	1. It is computationally expensive; 2. It is possible that some optimal properties are not available when the sample size is too small; 3. Strong assumptions related to the data structure are usually necessary;
PCE	1. It is computationally efficient; 2. It is available to solve the underdetermined problem;	1. It may have limitations when dealing with high dimensional data; 2. It is easy to be influenced by the effects of noise;

Table 3-2: The advantages and disadvantages of Kriging and PCE

Posteriori error estimation

Based on [62], the accuracy of the PCE-based surrogate model prediction can be investigated using the related generalization error indicated as follows:

$$\epsilon_{gen} = \frac{\mathbb{E}\left[\left(\mathcal{M}\left(\boldsymbol{x}\right) - \mathcal{M}^{PC}\left(\boldsymbol{x}\right)\right)^2\right]}{\mathrm{Var}\left[\boldsymbol{\mathcal{Y}}\right]}. \tag{3-33}$$

Similarly, if there is an independent set of observations $\boldsymbol{\mathcal{O}}_{\mathrm{val}} = \left\{\left(\boldsymbol{x}_{\mathrm{val}}^{(1)}, y_{\mathrm{val}}^{(1)}\right), ..., \left(\boldsymbol{x}_{\mathrm{val}}^{(N_{\mathrm{val}})}, y_{\mathrm{val}}^{(N_{\mathrm{val}})}\right)\right\}$, the validation error for investigating the quality of the PCE surrogate model can be obtained as follows[62]:

$$\epsilon_{\mathrm{val}} = \frac{N_{\mathrm{val}} - 1}{N_{\mathrm{val}}}\left[\frac{\sum_{i=1}^{N_{\mathrm{val}}}\left(\mathcal{M}\left(\boldsymbol{x}_{\mathrm{val}}^{(i)}\right) - \mathcal{M}^{PC}\left(\boldsymbol{x}_{\mathrm{val}}^{(i)}\right)\right)^2}{\sum_{i=1}^{N_{\mathrm{val}}}\left(\mathcal{M}\left(\boldsymbol{x}_{\mathrm{val}}^{(i)}\right) - \mu_{y_{\mathrm{val}}}\right)^2}\right], \tag{3-34}$$

where $y_{\mathrm{val}}^{(i)} = \mathcal{M}\left(\boldsymbol{x}_{\mathrm{val}}^{(i)}\right)$ is the computational model output and the sample mean of the independent validation output set $\mu_{Y_{\mathrm{val}}}$ can be denoted as Equation 3-19.

However, it is expensive to obtain another set of independent input and output data for a computational model. As a consequence, the normalized empirical error and the leave-one-out (LOO) cross validation (CV) error are used to estimate the generalization error.

Normalized empirical error reuses the experimental design data and can be indicated as follows[62]:

$$\epsilon_{\mathrm{emp}} = \frac{\sum_{i=1}^{N}\left(\mathcal{M}\left(\boldsymbol{x}^{(i)}\right) - \mathcal{M}^{PC}\left(\boldsymbol{x}^{(i)}\right)\right)^2}{\sum_{i=1}^{N}\left(\mathcal{M}\left(\boldsymbol{x}^{(i)}\right) - \mu_y\right)^2}, \tag{3-35}$$

where μ_y represents the sample mean of the experimental design outputs.

Based on [62], the disadvantage of the normalized empirical error is that the over-fitting can be caused no matter what size the experimental design has. To deal with this, the LOO CV error is introduced.

Leave-one-out cross-validation error involves two parts according to [21]. First, there are N PCE-based surrogate models $\mathcal{M}^{PC\backslash i}$ trained and every one of them is trained using a reduced experimental design $\boldsymbol{\mathcal{X}}\backslash\boldsymbol{x}^{(i)} = \left\{\boldsymbol{x}^{(j)},\ j = 1, ..., N,\ j \neq i\right\}$. Next, the i-th surrogate model prediction at the corresponding excluded point $\boldsymbol{x}^{(i)}$ is compared with the corresponding output of the original model. The LOO CV error is described as follows:

$$\epsilon_{LOO} = \frac{\sum_{i=1}^{N}\left(\mathcal{M}\left(\boldsymbol{x}^{(i)}\right) - \mathcal{M}^{PC\backslash i}\left(\boldsymbol{x}^{(i)}\right)\right)^2}{\sum_{i=1}^{N}\left(\mathcal{M}\left(\boldsymbol{x}^{(i)}\right) - \mu_y\right)^2}. \tag{3-36}$$

Furthermore, according to [20], if the least-square minimization problem for estimating $q_{\boldsymbol{\kappa}}$ is solved, N PCE-based surrogate models will not be necessary and ϵ_{LOO} can be computed in another way shown as follows:

$$\epsilon_{LOO} = \frac{\sum_{i=1}^{N}\left(\frac{\mathcal{M}\left(\boldsymbol{x}^{(i)}\right) - \mathcal{M}^{PC}\left(\boldsymbol{x}^{(i)}\right)}{1 - h_i}\right)^2}{\sum_{i=1}^{N}\left(\mathcal{M}\left(\boldsymbol{x}^{(i)}\right) - \mu_y\right)^2}, \tag{3-37}$$

where h_i represents the i-th element of the vector $\boldsymbol{h} = \text{diag}\left(\boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\right)$ and $\boldsymbol{A}$ can be obtained from Equation 3-31.

3-2 Sensitivity analysis

The computational model $\mathcal{M}$ basically depends on the input variables, model parameters and scenarios. The current research focuses on investigating how sensitive model outputs are to each model parameter of interest, which can be worked out using the sensitivity analysis. Not only can the theoretical sense be provided, but it is also possible to explore spotting minor model parameters and reducing the dimensions of the relative problem. There are different kinds of methods for conducting the sensitivity analysis such as the sample-based methods, linearization methods and global methods. In the current research, Sobol' indices, one of the global methods, will be used to investigate the sensitivity of each model parameter of interest by decomposing the variance of the model outputs in terms of each model parameter's contribution and the combinations of these contributions.

3-2-1 Sensitivity analysis using Sobol' indices

Based on [81], sensitivity analysis using Sobol' indices is a form of global methods conducted on the probabilistic framework. It decomposes the variance of the model output into fractions attributed to model parameters of interest and the sets thereof. To elaborate its principle, the expansion of the computational model $y = \mathcal{M}(\boldsymbol{\theta_M})$ is defined as the sum with increasing dimensions under the condition of independent model parameters. For the simplification of notations in the following elaboration, the components of the vector $\boldsymbol{\theta_M} = (\theta_{M1}, ..., \theta_{MN_\theta})$ consisting of uncertain model parameters are assumed uniformly distributed within $[0, 1]$. It is important to note that this assumption has nothing to do with the sensitivity analysis conducted in Chapter 4. The assumption is only for making it convenient to introduce the theory and will not lead to lose the generality. The definition of the Sobol' decomposition is indicated as follows[61]:

$$y = \mathcal{M}(\boldsymbol{\theta_M}) = \mathcal{M}_0 + \sum_{i=1}^{N_\theta} \mathcal{M}_i(\theta_{Mi}) + \sum_{1\le i<j\le N_\theta} \mathcal{M}_{ij}(\theta_{Mi}, \theta_{Mj}) + ... + \mathcal{M}_{1,2,...,N_\theta}(\theta_{M1}, ..., \theta_{MN_\theta}), \tag{3-38}$$

where $\mathcal{M}_0 = \mathbb{E}[y]$, $\mathcal{M}_{ij}$ denotes the contribution made by θ_{Mi} and θ_{Mj} and other higher order terms are defined as $\mathcal{M}_{ij}$ analogously.

Meanwhile, all terms in the decomposition are orthogonal, which can be mathematically described as follows[61]:

$$\int_0^1 \mathcal{M}_{i_1,...,i_s}(\theta_{M1}, ..., \theta_{Ms})\, \mathrm{d}\theta_{Mik} = 0,\ 1 \le k \le s. \tag{3-39}$$

As a consequence, all terms in the decomposition can be defined in terms of conditional expected values[61]:

$$
\begin{aligned}
\mathcal{M}_0 &= \mathbb{E}\left(y\right) \\
&= \int_{\mathcal{D}_{\boldsymbol{\theta_M}}} \mathcal{M}\left(\boldsymbol{\theta_M}\right) \mathrm{d}\boldsymbol{\theta_M}, \\
\mathcal{M}_i\left(\theta_{Mi}\right) &= \mathbb{E}\left(y|\theta_{Mi}\right) - \mathcal{M}_0 \\
&= \int_0^1 ... \int_0^1 \mathcal{M}\left(\boldsymbol{\theta_M}\right) \mathrm{d}\boldsymbol{\theta}_{\boldsymbol{M}\sim i} - \mathcal{M}_0, \\
\mathcal{M}_{ij}\left(\theta_{Mi}, \theta_{Mj}\right) &= \mathbb{E}\left(y|\theta_{Mi}, \theta_{Mj}\right) - \mathcal{M}_0 - \mathcal{M}_i\left(\theta_{Mi}\right) - \mathcal{M}_j\left(\theta_{Mj}\right) \\
&= \int_0^1 ... \int_0^1 \mathcal{M}\left(\boldsymbol{\theta_M}\right) \mathrm{d}\boldsymbol{\theta}_{\boldsymbol{M}\sim i,j} - \mathcal{M}_0 - \mathcal{M}_i\left(\theta_{Mi}\right) - \mathcal{M}_j\left(\theta_{Mj}\right),
\end{aligned}
\tag{3-40}
$$

where $\mathbb{E}(\cdot)$ denotes the expected value operator and $\boldsymbol{\theta}_{\boldsymbol{M}\sim i}$ represents the vector consisting of all model parameters except θ_{Mi}.

The higher-order terms are analogously built. Therefore, the total variance of the model output V can be denoted as follows[61]:

$$
\begin{aligned}
V &= \mathrm{Var}(y) \\
&= \mathbb{E}\left[\left(y - \mathbb{E}(y)\right)^2\right] \\
&= \mathbb{E}\left(y^2\right) - \left[\mathbb{E}\left(y\right)\right]^2 \\
&= \int_{\mathcal{D}_{\boldsymbol{\theta_M}}} \mathcal{M}^2\left(\boldsymbol{\theta_M}\right) \mathrm{d}\boldsymbol{\theta_M} - {\mathcal{M}_0}^2 \\
&= \sum_{s=1}^{N_\theta} \sum_{1\le i_1<...<i_s\le N_\theta} \int_0^1 ... \int_0^1 \mathcal{M}^2_{i_1,...,i_s}\left(\theta_{Mi_1}, ..., \theta_{Mi_s}\right) \mathrm{d}\theta_{Mi_1} ... \mathrm{d}\theta_{Mi_s} \\
&= \sum_{i=1}^{N_\theta} V_i + \sum_{1\le i<j\le N_\theta} V_{ij} + ... + V_{1,...,N_\theta},
\end{aligned}
\tag{3-41}
$$

where $V_i = \mathrm{Var}\left[\mathbb{E}\left(y|\theta_{Mi}\right)\right]$, $V_{ij} = \mathrm{Var}\left[\mathbb{E}\left(y|\theta_{Mi}, \theta_{Mj}\right)\right] - V_i - V_j$ and higher-order terms are analogously determined.

Next, a nature definition for quantifying the sensitivity results from the variance decomposition shown as follows[61]:

$$
S_{i_1,...,i_s} = \frac{V_{i_1,...,i_s}}{V}. \tag{3-42}
$$

$S_{i_1,...,i_s}$ indicates how the set consisting of $\theta_{Mi_1}, ..., \theta_{Mi_s}$ contributes to the total variance. It is named as the first order Sobol' index S_i when only single one model parameter θ_{Mi} is involved. The second order Sobol' index S_{ij} indicates the interactive contribution made by two model parameter θ_{Mi} and θ_{Mj}. The analogous definitions can apply to the higher-order indices. It is important to note that the property shown in Equation 3-43 can hold [49]:

$$\sum_{i=1}^{N_\theta} S_i + \sum_{1\leq i<j\leq N_\theta}^{N_\theta} S_{ij} + ... + S_{1,...,N_\theta} = 1. \tag{3-43}$$

Furthermore, the sum of all Sobol' indices involving the model parameter θ_{Mi} is named as the total Sobol' index S_i^{Total} and can be practically calculated as follows[61]:

$$\begin{aligned} S_i^{Total} &= 1 - S_{\sim i} \\ &= 1 - \frac{\mathrm{Var}\left[\mathbb{E}\left(y|\boldsymbol{\theta}_{\boldsymbol{M}\sim i}\right)\right]}{V}. \end{aligned} \tag{3-44}$$

The property about the total Sobol' index is $\sum_{i=1}^{N_\theta} S_i^{Total} \geq 1$ [49].

3-3 Bayesian calibration

Most of calibration methods use all kinds of optimization algorithms to estimate the model parameters based on the inputs and outputs. Some of them also involve weighted least-square estimation [86] and best linear unbiased estimation [72]. These deterministic methods focus on minimizing the difference between the computed model outputs and the measured data, which can generate systematic bias and amplify uncertainty in the calibrated model output due to lack of prior information of model parameters. To deal with this, the idea of considering calibration from the probabilistic aspect is come up with.

In the current research, The FLORIDyn model is calibrated using Bayesian methods[48] which deal with so-called inverse problems[84][54]: estimate model parameters which can not be measured directly given the computational model providing the experimental data which are not directly related to model parameters. The inverse problems aim to obtain the insights on the model parameters based on the backward propagation of the observed information rather than the forward propagation of the model parameters' information through a computational model [91].

3-3-1 Bayesian inference

A computational model $\mathcal{M}$ is considered as follows:

$$y = \mathcal{M}\left(\boldsymbol{\theta}_{\boldsymbol{M}}\right),\ \boldsymbol{\theta}_{\boldsymbol{M}} \in \mathcal{D}_{\boldsymbol{\theta}_{\boldsymbol{M}}} \subset \mathbb{R}^{N_\theta},\ y \in \mathbb{R}, \tag{3-45}$$

where $\boldsymbol{\theta}_{\boldsymbol{M}}$ denotes the vector consisting of N_θ random model parameters $\theta_{M1}, ..., \theta_{MN_\theta}$ and y represents the one-dimensional output of $\mathcal{M}$.

Under the condition that N_θ model parameters $\theta_{M1}, ..., \theta_{MN_\theta}$ can not be directly measured and only the quantities of interest modelled by $\mathcal{M}$ are available to measure, a random vector $\boldsymbol{\Theta}$ associated with $\boldsymbol{\theta}_{\boldsymbol{M}}$ is introduced and the corresponding probability density function (PDF) $\pi(\cdot)$ is selected initially as the prior distribution:

$$\boldsymbol{\Theta_M} \sim \pi\left(\boldsymbol{\theta_M}\right). \tag{3-46}$$

Next, N independent measurements $\{y_i \in \mathbb{R},\ i = 1, ..., N\}$ of the model output Y are collected in a data set $\mathcal{Y} \overset{def}{=} \{y_1, ..., y_N\}$. Based on the classical Bayes' theorem [47], the posterior distribution conditional on the measured data $\mathcal{Y}$ can be indicated as follows:

$$\pi\left(\boldsymbol{\theta_M}|\mathcal{Y}\right) = \frac{\mathcal{L}\left(\boldsymbol{\theta_M};\mathcal{Y}\right)\pi\left(\boldsymbol{\theta_M}\right)}{Z_e}. \tag{3-47}$$

The term $\mathcal{L}\left(\boldsymbol{\theta_M};\mathcal{Y}\right)$ in Equation 3-47 denotes the likelihood function returning the related likelihood of observing the measured data $\mathcal{Y}$ for the given $\boldsymbol{\theta_M}$ and is defined as follows:

$$\mathcal{L}:\ \boldsymbol{\theta_M} \mapsto \mathcal{L}\left(\boldsymbol{\theta_M};\mathcal{Y}\right) \overset{def}{=} \prod_{i=1}^{N} \pi\left(y_i|\boldsymbol{\theta_M}\right). \tag{3-48}$$

The term Z_e represents the evidence, which is a factor normalizing that the integration in Equation 3-49 equals 1:

$$Z_e \overset{def}{=} \int_{\boldsymbol{\Theta}} \mathcal{L}\left(\boldsymbol{\theta_M};\mathcal{Y}\right)\pi\left(\boldsymbol{\theta_M}\right)\mathrm{d}\boldsymbol{\theta_M}. \tag{3-49}$$

3-3-2 Discrepancy

To depict the difference between the computational model output and the corresponding measurement, the discrepancy term $\varepsilon \in \mathbb{R}$ is introduced as follows:

$$y = \mathcal{M}(\boldsymbol{\theta_M}) + \varepsilon. \tag{3-50}$$

The current research assumes ε as an additive Gaussian discrepancy:

$$\varepsilon \sim \mathcal{N}\left(\varepsilon|0, \sigma^2\right), \tag{3-51}$$

where the mean of the discrepancy is assumed as 0 and $\sigma^2 \in \mathbb{R}$ is the unknown variance.

Based on Equation 3-50 and Equation 3-51, a measurement point $y_i \in \mathcal{Y}$ indicates a realization of a Gaussian distribution whose mean is $\mathcal{M}\left(\boldsymbol{\theta_M}\right)$ and variance is σ^2 described as follows:

$$\pi\left(y|\boldsymbol{\theta_M}\right) = \mathcal{N}\left(y|\mathcal{M}\left(\boldsymbol{\theta_M}\right), \sigma^2\right), \tag{3-52}$$

where the vector consisting of model parameters is considered as a random vector $\boldsymbol{\Theta} \sim \pi\left(\boldsymbol{\theta_M}\right)$.

Based on [91], given N independent measurements $y_1, ..., y_N$ collected in the data set $\mathcal{Y}$, the likelihood function can be derived as follows:

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta_M}, \sigma^2; \mathcal{Y}\right) &= \prod_{i=1}^{N} \mathcal{N}\left(y_i | \mathcal{M}\left(\boldsymbol{\theta_M}\right), \sigma^2\right) \\ &= \prod_{i=1}^{N_\theta} \frac{1}{\sqrt{\left(2\pi\sigma^2\right)^{N_\theta}}} \exp\left(-\frac{1}{2\sigma^2}\left(y_i - \mathcal{M}\left(\boldsymbol{\theta_M}\right)\right)^2\right).\end{aligned} \tag{3-53}$$

Given the prior distribution and the likelihood function, the posterior distribution can be then derived from Equation 3-47.

However, the above inference requires a specific value for the variance σ^2. When σ^2 is unknown, the strategy is to consider the discrepancy term as an additional parameter and to jointly infer it with model parameters. The new built parameter vector is defined as:

$$\boldsymbol{\theta} \overset{def}{=} \begin{pmatrix} \boldsymbol{\theta_M} \\ \theta_\varepsilon \end{pmatrix}, \tag{3-54}$$

where the discrepancy parameter θ_ε is priorly independent and assumed as $\theta_\varepsilon \equiv \sigma^2$. The new prior distribution can be indicated as:

$$\begin{aligned}\pi\left(\boldsymbol{\theta}\right) &= \pi\left(\boldsymbol{\theta_M}\right)\pi\left(\theta_\varepsilon\right) \\ &= \pi\left(\boldsymbol{\theta_M}\right)\pi\left(\sigma^2\right).\end{aligned} \tag{3-55}$$

Again, given the new prior distribution and the likelihood function with unknown σ^2, the corresponding posterior distribution can be derived as follows[91]:

$$\pi\left(\boldsymbol{\theta_M}, \sigma^2 | \mathcal{Y}\right) = \frac{\pi\left(\boldsymbol{\theta_M}\right)\pi\left(\sigma^2\right)\mathcal{L}\left(\boldsymbol{\theta_M}, \sigma^2; \mathcal{Y}\right)}{Z_e}. \tag{3-56}$$

Next, the distribution $\pi\left(\theta_{Mi} | \mathcal{Y}\right)$ for the individual computational model parameter and the distribution $\pi\left(\sigma^2 | \mathcal{Y}\right)$ for the variance can be extracted.

3-3-3 Inverse solution

According to [91], after the posterior distribution is obtained, a point estimate indicated by the posterior mean vector can be computed as:

$$\mathbb{E}\left(\boldsymbol{\Theta} | \mathcal{Y}\right) = \int_{\mathcal{D}_{\boldsymbol{\Theta}}} \boldsymbol{\theta_M} \pi\left(\boldsymbol{\theta_M} | \mathcal{Y}\right) \mathrm{d}\boldsymbol{\theta_M}. \tag{3-57}$$

To quantify the uncertainty of the estimation, the posterior covariance matrix is computed as:

$$\mathrm{Cov}\left(\boldsymbol{\Theta} | \mathcal{Y}\right) = \int_{\mathcal{D}_{\boldsymbol{\Theta}}} \left[\boldsymbol{\theta_M} - \mathbb{E}\left(\boldsymbol{\Theta} | \boldsymbol{\mathcal{Y}}\right)\right]\left[\boldsymbol{\theta_M} - \mathbb{E}\left(\boldsymbol{\Theta} | \mathcal{Y}\right)\right]^T \pi\left(\boldsymbol{\theta_M} | \mathcal{Y}\right) \mathrm{d}\boldsymbol{\theta_M}. \tag{3-58}$$

Additionally, the posterior distribution $\pi(\theta_{Mi}|\mathcal{Y})$ for the individual computational model parameter can be computed as:

$$\pi(\theta_{Mi}|\mathcal{Y}) = \int_{\mathcal{D}_{\boldsymbol{\Theta}_{\sim i}}} \pi(\boldsymbol{\theta_M}|\mathcal{Y})\, \mathrm{d}\boldsymbol{\theta}_{\boldsymbol{M}\sim i}, \tag{3-59}$$

where $\boldsymbol{\theta}_{\boldsymbol{M}\sim i}$ represents the vector consisting of all model parameters except θ_{Mi}.

In practice, $\pi(\boldsymbol{\theta_M}|\mathcal{Y})$ is usually used to compute the conditional expectation of the model output described as:

$$\mathbb{E}[\mathcal{M}(\boldsymbol{\Theta}|\mathcal{Y})] = \int_{\mathcal{D}_{\boldsymbol{\Theta}}} \mathcal{M}(\boldsymbol{\theta_M})\, \pi(\boldsymbol{\theta_M}|\mathcal{Y})\, \mathrm{d}\boldsymbol{\theta_M}. \tag{3-60}$$

3-3-4 Model predictions

Bayesian inference can be also used to work on the predictive distributions resulting from the prior and posterior estimations for Y and $Y|\mathcal{Y}$ respectively with considering the uncertainty in $\boldsymbol{\theta_M}$. According to [91], the prior predictive distribution is computed as:

$$\pi(y) = \int_{\mathcal{D}_{\boldsymbol{\Theta}}} \pi(y|\boldsymbol{\theta_M})\, \pi(\boldsymbol{\theta_M})\, \mathrm{d}\boldsymbol{\theta_M}. \tag{3-61}$$

Meanwhile, the posterior predictive distribution is computed as:

$$\pi(y|\mathcal{Y}) = \int_{\mathcal{D}_{\boldsymbol{\Theta}}} \pi(y|\boldsymbol{\theta_M})\, \pi(\boldsymbol{\theta_M}|\mathcal{Y})\, \mathrm{d}\boldsymbol{\theta_M}. \tag{3-62}$$

In practice, $\pi(y|\mathcal{Y})$ can be proposed by selecting a specific value $\widehat{\boldsymbol{\theta_M}}$ from $\pi(\boldsymbol{\theta_M}|\mathcal{Y})$ as follows:

$$\pi(y|\mathcal{Y}) \overset{def}{=} \pi\left(y|\widehat{\boldsymbol{\theta_M}}\right). \tag{3-63}$$

The specific value for $\widehat{\boldsymbol{\theta_M}}$ is mostly selected as the posterior mean [91].

3-3-5 Markov Chain Monte Carlo (MCMC)

According to [76] and [59], Markov Chain Monte Carlo (MCMC) methods, also known as samplers, comprise plenty of algorithms to work on inverse solutions by building a Markov chain $\left(\boldsymbol{\Theta}^{(1)}, ..., \boldsymbol{\Theta}^{(N_\theta)}\right)$ over the prior information $\mathcal{D}_{\boldsymbol{\Theta}}$ and setting a constant distribution of the Markov chain equal to the posterior distribution of $\boldsymbol{\Theta}$. The transition probability $\mathcal{K}\left(\boldsymbol{\theta_M}^{(t+1)}|\boldsymbol{\theta_M}^{(t)}\right)$ from the chain position $\boldsymbol{\theta_M}^{(t)}$ at the step t to the chain position $\boldsymbol{\theta_M}^{(t+1)}$ at the next step $t+1$ can be used to uniquely define the Markov chain with a constant distribution equal to $\pi(\boldsymbol{\theta_M}|\mathcal{Y})$ by satisfying the detailed balance requirement[91] indicated as follows:

$$\pi\left(\boldsymbol{\theta_M}^{(t)}|\mathcal{Y}\right)\mathcal{K}\left(\boldsymbol{\theta_M}^{(t+1)}|\boldsymbol{\theta_M}^{(t)}\right) = \pi\left(\boldsymbol{\theta_M}^{(t+1)}|\mathcal{Y}\right)\mathcal{K}\left(\boldsymbol{\theta_M}^{(t)}|\boldsymbol{\theta_M}^{(t+1)}\right). \tag{3-64}$$

Based on [91], the detailed balance requirement states that the probability of the move from $\boldsymbol{\theta_M}^{(t)}$ to $\boldsymbol{\theta_M}^{(t+1)}$ equals the probability of the move from $\boldsymbol{\theta_M}^{(t+1)}$ to $\boldsymbol{\theta_M}^{(t)}$.

The Markov chain's invariant distribution is truly the posterior distribution of $\boldsymbol{\Theta}$, which can be seen from the integral of the detail balance requirement over $\mathrm{d}\boldsymbol{\theta_M}^{(t)}$ shown as follows:

$$\pi\left(\boldsymbol{\theta_M}^{(t+1)}|\mathcal{Y}\right)=\int_{\mathcal{D}_{\boldsymbol{\Theta}}}\pi\left(\boldsymbol{\theta_M}^{(t)}|\mathcal{Y}\right)\mathcal{K}\left(\boldsymbol{\theta_M}^{(t+1)}|\theta_M^{(t)}\right)\mathrm{d}\boldsymbol{\theta_M}^{(t)}. \tag{3-65}$$

Most of MCMC algorithms require plenty of tuning to improve their performance if there is strong correlation between model parameters. To deal with this problem, the affine-invariant ensemble sampler (AIES) researched by [51] in 2010 is used in the current research. The AIES algorithm can perform equally well when sampling from the poorly-scaled, highly-anisotropic or well-scaled distributions which can be affine-transformed to each other [51]. The related working principles are explained next.

Affine-invariant ensemble sampler (AIES)

Based on [51], the AIES algorithm simultaneously evolves N_w MCMC chains named as N_w walkers by firstly generating N_w samples $\boldsymbol{\theta_M}_1^{(1)},...,\boldsymbol{\theta_M}_{N_w}^{(1)}$ from the given prior as the first position of each chain at once. After that, N_w walkers are sequentially updated at each step t. Each walker can be updated by generating a candidate and computing the a probability as the standard of accepting or rejecting the candidate to replace the original walker. In mathematical notation, $\boldsymbol{\theta_M}_k^*$ represents the candidate for the k-th walker at the step $t+1$, which can be indicated as follows:

$$\boldsymbol{\theta_M}_k^*=\tau\boldsymbol{\theta_M}_k^{(t)}+(1-\tau)\,\widetilde{\boldsymbol{\theta_M}}, \tag{3-66}$$

where $\widetilde{\boldsymbol{\theta_M}}$ represents a complementary walker randomly picked from the set $\left\{\boldsymbol{\theta_M}_1^{(t+1)},...,\boldsymbol{\theta_M}_{k-1}^{(t+1)},\boldsymbol{\theta_M}_{k+1}^{(t)},...,\boldsymbol{\theta_M}_{N_w}^{(t)}\right\}$ and τ denotes a real-valued proposal stretch factor which can be defined as a stochastic variable subject to a proposal probability density function $g(\tau)$ written as[60]:

$$g(\tau)=\begin{cases}\frac{1}{2\times\left(\sqrt{\nu}-\frac{1}{\sqrt{\nu}}\right)}\times\frac{1}{\sqrt{\tau}} & \tau\in\left[\frac{1}{\nu},\nu\right]\\ 0 & \text{otherwise}\end{cases}, \tag{3-67}$$

where ν is the step size of the AIES algorithm which can be set by the user and $\nu=2$ is chosen in most of cases based on [40].

Next, the candidate $\boldsymbol{\theta_M}_k^*$ is accepted to replace $\boldsymbol{\theta_M}_k^{(t+1)}$ with the acceptance probability P_k written as[51]:

$$P_k=\min\left(1,\ \tau^{N_\theta-1}\times\frac{\pi\left(\boldsymbol{\theta_M}_k^*|\mathcal{Y}\right)}{\pi\left(\boldsymbol{\theta_M}_k^{(t)}|\mathcal{Y}\right)}\right). \tag{3-68}$$

Generate a sample from the uniform distribution $u \sim U[0,1]$, if $P_k > u$, the candidate is accepted by setting $\boldsymbol{\theta}_{\boldsymbol{M}_k}^{(t+1)} = \boldsymbol{\theta}_{\boldsymbol{M}_k}^{*}$, otherwise, the candidate is rejected by setting $\boldsymbol{\theta}_{\boldsymbol{M}_k}^{(t+1)} = \boldsymbol{\theta}_{\boldsymbol{M}_k}^{(t)}$ instead.

Convergence assessment in MCMC methods

The chains consisting of samples constructed by MCMC methods will be subject to the posterior distribution in the end. However, the convergence quality of the chains can only be assessed from a finite number of samples in practice. In the current research, the trace plots will be used to assess the MCMC chains' convergence.

The trace plots can offer the visual insights about convergence for each dimension of the sample individually. To do so, the kernel density estimation (KDE) method is used to shape the posterior marginal. The kernel density estimator is indicated as follows[92]:

$$\begin{aligned} \hat{f}_h\left(\theta_{Mi}\right) &= \frac{1}{T}\sum_{t=1}^{T} K_h\left(\theta_{Mi} - \theta_{Mi,t}\right) \\ &= \frac{1}{Th}\sum_{t=1}^{T} K\left(\frac{\theta_{Mi} - \theta_{Mi,t}}{h}\right) \end{aligned}, \tag{3-69}$$

where $\theta_{Mi},\ i = 1,...N_\theta$ represents the i-th dimension of the sample, $\theta_{Mi,t}$ represents the sample of θ_{Mi} at the step t, h is a positive scalar named as the bandwidth, $K(\cdot)$ denotes the kernel function which is an even, non-negative function and whose integral over the range $(-\infty, +\infty)$ equals 1, $K_h(\cdot)$ is named as the scaled kernel and defined as $K_h(\cdot) = \frac{1}{h}K\left(\frac{\cdot}{h}\right)$.

The KDE of the posterior marginal will not have noticeable changes as the increase of the number of steps after the chain reaches its steady state [91].

3-4 Summary

The procedures of using the UQLab software [63] to conduct the uncertainty quantification of the FLORIDyn model are shown in Figure 3-7. First, the target model parameters summarized in Table 2-1 will be converted into probabilistic variables as their prior information. In the current research, each target model parameter will be subject to a uniform distribution by perturbing their mean value to obtain lower and upper bounds. Next, a surrogate model will be trained to replace the original FLORIDyn model by using the Kriging module [57] or the PCE module [62]. The trained surrogate model for the next procedure will be chosen based on the comprehensive comparison of the FLORIDyn model's Kriging-based surrogate model and the FLORIDyn model's PCE-based surrogate model in terms of accuracy and efficiency. The chosen surrogate model will be then applied to conduct sensitivity analysis with the sensitivity analysis module [61] and Bayesian calibration with the Bayesian inversion module [91]. The Sobol' indices will be used to depict each target parameter's sensitivity to the model output. The AIES algorithm will be used to sample from each target parameter's posterior distribution.

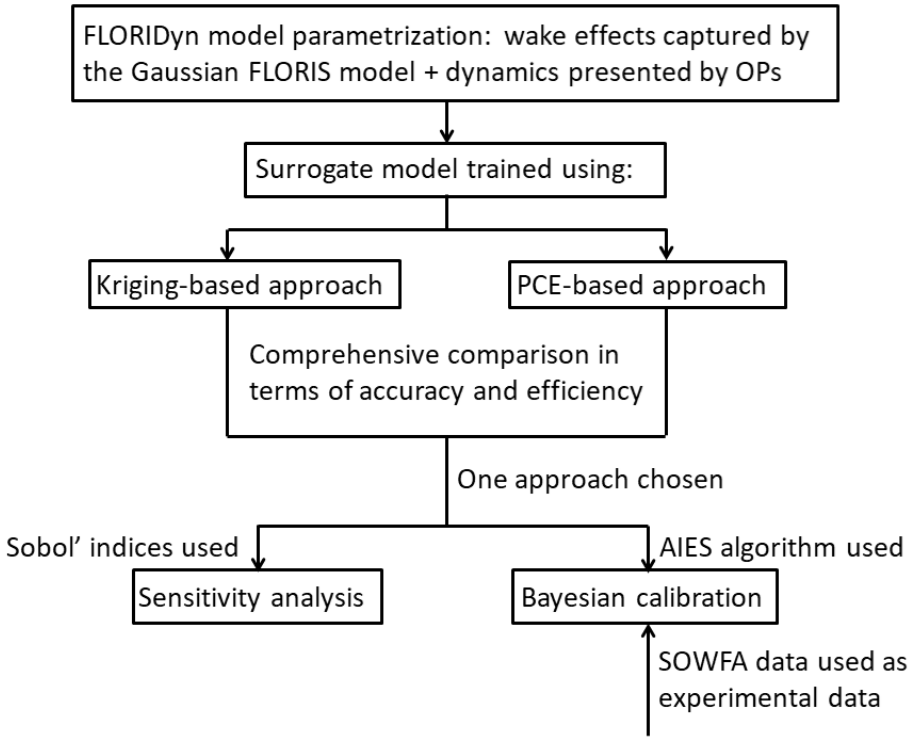


Figure 3-7: Flowchart for the procedures of uncertainty quantification.

Chapter 4

3-Turbine Case Study

The uncertainty quantification comprising of the surrogate model construction, sensitivity analysis and Bayesian calibration will be applied for the 3-Turbine case of the FLORIDyn model. SOWFA data will be used as the measurements for Bayesian calibration. The simulation conditions of the case are described in Section 4-1. Section 4-2 discusses the results of the case: Model parameters' prior distributions converted from their mean values and the comparison of the Kriging-based surrogate model and the PCE-based surrogate model are presented in Section 4-2-1. Section 4-2-2 analyses the sensitivity of the model output to each model parameter of interest. Last, the results of Bayesian calibration and the comparison with SOWFA data are shown in Section 4-2-3.

4-1 Simulation conditions

The simulation of the current research uses the DTU 10-MW reference wind turbine [14]. In FLORIDyn, the turbines are modelled using the Actuator Disc Model (ADM) [28]. There are 50 chains for each turbine with 200 OPs per chain, which can offer the enough OP density in the wake [17]. The time step is set to $4s$. The simulation scenario $\mathcal{S}$ comprised of 3 wind turbines is modified from [17]. The layout is shown in Figure 4-1. The three wind turbines are placed exactly along the downwind direction and the distance between two adjacent wind turbines equals $892m$, which is five times larger than the rotor diameter. The position coordinates of the three wind turbines are $(608m, 500m)$, $(1500m, 500m)$ and $(2392m, 500m)$ respectively. In SOWFA, the turbines are modelled using the Actuator Line Method (ALM) [68]. The time step is set to $0.04s$. The full SOWFA flow field domain consisting of the base cells of $10 \times 10 \times 10$ m is set to span $3000 \times 1000 \times 1000$ m. There are also two SOWFA flow field refinement areas without offset from the ground which are located at the center of the domain. The first one spans $2400 \times 800 \times 500$ m with the base cells of $5 \times 5 \times 5$ m and the second one spans $2200 \times 600 \times 350$ m with the base cells of $2.5 \times 2.5 \times 2.5$ m. In both FLORIDyn and SOWFA, the mean upstream wind speed is set to $8.2m/s$ at the hub height with the ambient turbulence intensity of around 6% and the time interval of the simulation

is set to $1200s$. The yaw angle γ of the upstream wind turbine is set to change with the rate of $0.3°/s$ from $0°$ to $10°$ at $200s$ and from $10°$ to $20°$ at $800s$. The positive value for the angle stands for clock-wise rotation. The other two wind turbines stay still during the entire simulation.

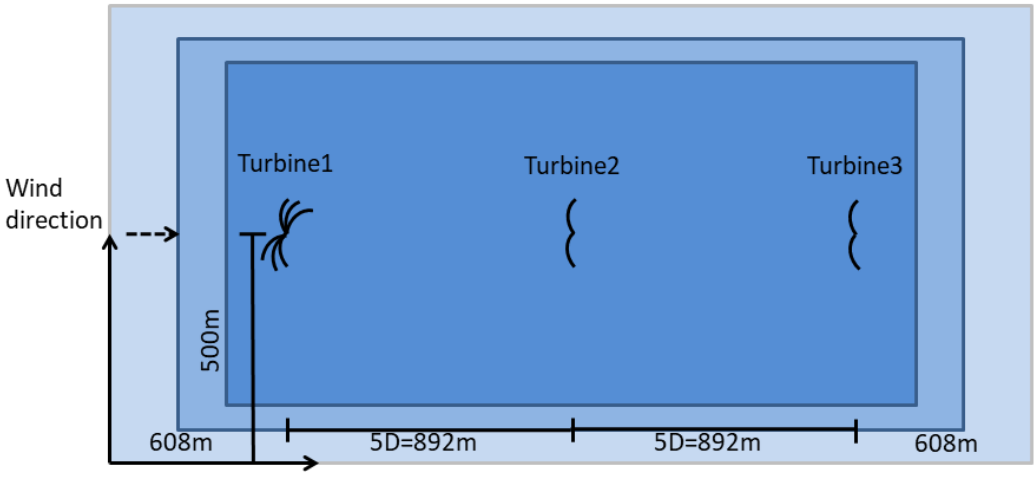


Figure 4-1: The scenario in terms of the 3-Turbine layout. Different cell refinement areas are marked with different colors. Three symbols for the upstream wind turbine represent $0°$, $10°$ and $20°$ yaw orientations respectively. The arrow on the left denoting the direction of wind is defined as $90°$.

4-2 Results and interpretation

Section 4-2-1 presents the model parameters' prior distributions converted from their mean values and two types of surrogate models trained for the 3-Turbine case of the FLORIDyn model. The results and the related interpretation of the sensitivity analysis are shown in Section 4-2-2. The model parameters' posterior distributions calibrated using SOWFA data as measurements and the comparison of the uncalibrated model, the calibrated model and SOWFA are presented in Section 4-2-3.

4-2-1 Surrogate models for the 3-Turbine case of FLORIDyn

To decrease the computational cost, the FLORIDyn model elaborated in Chapter 2 is replaced by the trained surrogate model described in Section 3-2. In mathematical notation, the vector comprised of model parameters is denoted as $\boldsymbol{\theta_M} = (\theta_{M1}, ..., \theta_{M11})$. An evaluation of the FLORIDyn model denoted as $\mathcal{M}$ at the discrete time step t returns wind turbine power outputs $y_i^{(t)}$, $i = 1, 2, 3$ forming a vector $\boldsymbol{Y}^{(t)} = \left(y_1^{(t)}, y_2^{(t)}, y_3^{(t)}\right)$. The FLORIDyn model evaluation $\mathcal{M}$ depends on the model parameters $\boldsymbol{\theta_M}$ and the simulation scenario $\mathcal{S}$ described in Section 4-1:

$$y_i^{(t)} = \mathcal{M}\left(\boldsymbol{\theta_M}, \mathcal{S}\right). \tag{4-1}$$

In the current research, the model parameters' prior distributions are defined as the uniform distribution $\mathcal{U}(\theta_{min}, \theta_{max})$ by perturbing their mean values summarized in Table 2-2 in Section 2-2 with $\pm 70\%$ to set lower and upper bounds. The specific values are indicated in Table 4-1.

	α^*	β^*	k_a	k_b	$k_{f,a}$	$k_{f,b}$	$k_{f,c}$	$k_{f,d}$	d	η	p_p
θ_{min}	0.696	0.0462	0.115	0.0011	0.219	0.250	0.0098	-0.544	0.3	0.257	0.66
θ_{max}	3.944	0.2618	0.652	0.0063	1.241	1.415	0.0553	-0.096	1.7	1.457	3.74

Table 4-1: Lower and upper bounds of the proposed prior uniform distributions for the model parameters

For the Kriging-based surrogate model elaborated in Section 3-1-1, the experimental design is obtained by sampling from the model parameters' prior distributions defined above with the Latin Hypercube sampling (LHS) method by default by the UQLab software [63]. The number of samples for the experimental design is set to 250. As described in Section 4-1, the time step in FLORIDyn is set to $4s$ and the simulation time is set to $1200s$. As a consequence, the FLORIDyn model evaluation returns a 301×3 matrix during the entire simulation. 903 original model outputs in total then need trained Kriging-based surrogates. The results for one of those outputs are indicated in Figure 4-2. As can be seen, the values for the trend, correlation function, estimation and optimization method are assigned by default by the UQLab software [63] for the $402nd$ output and the same way for the rest of outputs.

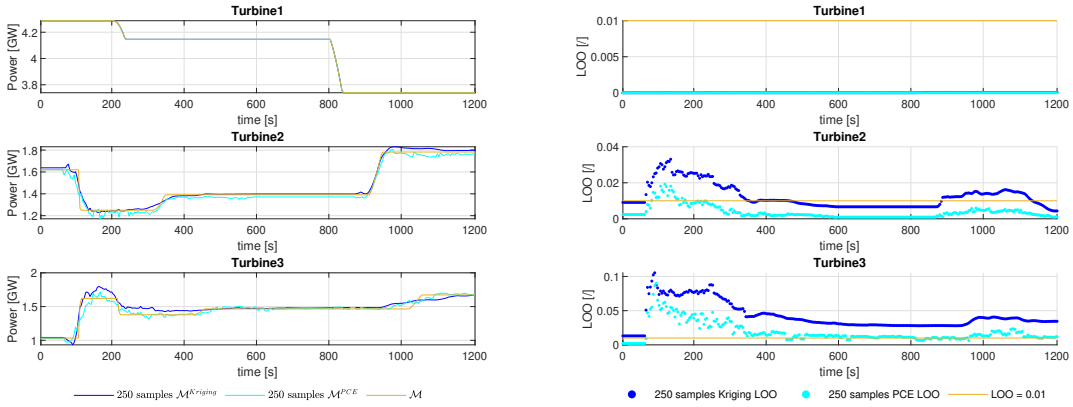
For the PCE-based surrogate model, the experimental design is obtained the same way as the Kriging-based surrogate model. The LARS sparse regression solver is used and 903 independent LARS-based surrogates in total then need to be built for the same reason mentioned above. To compare the accuracy of two types of obtained surrogate models, Figure 4-3(a) presents the outputs of the original FLORIDyn model and two types of surrogate models during the entire simulation. Meanwhile, all LOO errors of two types of surrogate models are shown in Figure 4-3(b).

As described in Section 4-1, the yaw angle γ of the upstream wind turbine (Turbine1) initially starts to be yawed with the angular velocity $0.3^\circ/s$ at $200s$. As shown in Figure 4-3(a), due to the fact that the influence of the yaw angle change needs time to arrive at the downstream wind turbine, the second wind turbine (Turbine2) then starts to be affected by it at about $312s$ and the third wind turbine (Turbine3) then starts to have a reaction at around $420s$. In other words, three wind turbines' first steady states reached after the first yaw angle change start at around $236s$, $348s$ and $456s$ respectively. In the current research, the simulation period from $456s$ to the end will be mainly taken into account.

It can be intuitively observed that both the Kriging-based surrogate model outputs and PCE-based surrogate model outputs almost overlap the FLORIDyn model outputs. However, the accuracy of these two types of surrogate models still requires to be investigated quantitatively. The LOO errors of theirs are next computed and 0.01 is used as the reference value for the LOO error modified from [30]. Even though the Kriging-based approach costs more time than the PCE-based approach when 250 model evaluations are used, Figure 4-3(b) indicates the PCE-based approach has better performance than the Kriging-based approach when using the same number of model evaluations in this case. However, not all LOO errors in these two approaches are smaller than 0.01. To improve the accuracy of the surrogate model more efficiently, the

```
%---------------- Kriging metamodel ----------------%
   Object Name:          Model 2
   Input Dimension:       11
   Output Dimension:      903
   Experimental Design
     X size:           [250x11]
     Y size:           [250x903]
     Sampling:           LHS
--- Output #402:
   Trend
     Type:             linear
     Degree:           1
     Beta:             [ 1.12554    0.00854    0.46729   -0.01026    0.12477   -0.02670
                        -0.20217    0.20887    0.03870    0.33301    0.11803    0.23650  ]

   Gaussian Process (GP)
     Corr. type:       ellipsoidal
     Corr. isotropy:    anisotropic
     Corr. family:     exponential
     sigma^2:          7.61912e-02
     Estimation method:  Maximum-likelihood (ML)

   Hyperparameters
     theta:            [ 2.39633    5.33886    2.62483    9.99995    3.69250    3.79732
                         9.99994    6.82289    9.99993    1.98890    9.99994  ]

     Optim. method:      Hybrid Genetic Alg.

   GP Regression
     Mode:             interpolation

   Error estimates
     Leave-one-out:    3.30787e-02
%-------------------------------------------------%
```

Figure 4-2: The Kriging-based surrogate results for the $402nd$ output of the original FLORIDyn model.



(a) Comparison of outputs of the FLORIDyn model, the Kriging-based surrogate model and the PCE-based surrogate model

(b) Comparison of LOO errors of the Kriging-based surrogate model and the PCE-based surrogate model

Figure 4-3: Comparison of the Kriging-based surrogate model and the PCE-based surrogate model in terms of outputs and LOO errors.

PCE-based approach is selected and the number of model evaluations is increased to 2000. As shown in Figure 4-4, during the simulation period of interest ($456s \sim 1200s$), all LOO errors are smaller than 0.01 except 3 out of 447 data. As a consequence, the PCE-based surrogate model trained with 2000 model evaluations will be selected to replace the original FLORIDyn model for the following sensitivity analysis and Bayesian calibration parts.

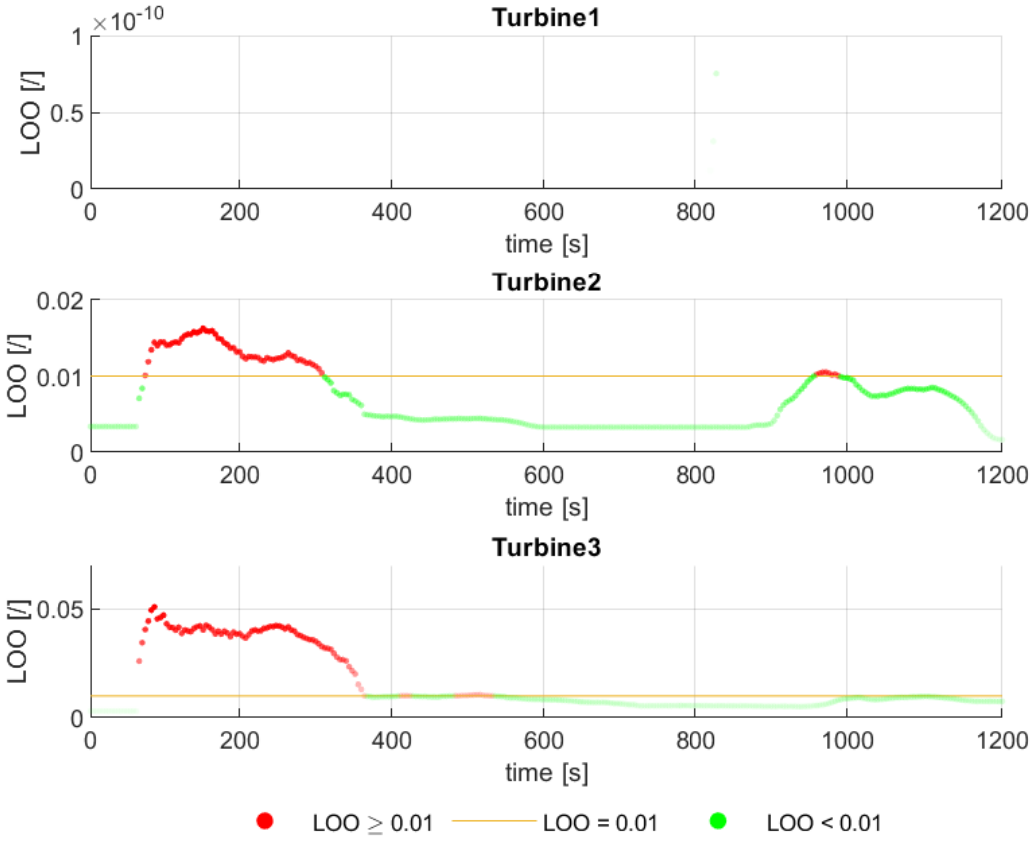


Figure 4-4: LOO errors during the entire simulation. Green indicates the LOO error is smaller than 0.01, while red indicates the opposite. The darker the color is, the larger the LOO error is.

4-2-2 Sensitivity analysis using Sobol' Indices

Total Sobol' indices elaborated in Section 3-3 are calculated to indicate the sensitivity of the PCE-based surrogate model to 11 model parameters summarized in Table 2-1. The results in terms of three wind turbines are time-dependent and presented in Figure 4-5, Figure 4-6 and Figure 4-7 respectively. The top edge line in each figure represents the sum of S_1^{Total} to S_{11}^{Total}, which can not be smaller than 1 since there are interaction effects between model parameters and these effects are counted more than once. The width of each color area depicts S_i^{Total}, $i = 1, ..., 11$ individually. The wider the color area is, the more sensitive the PCE-based surrogate model is to the corresponding model parameter.

As shown in Figure 4-5, it makes sense that the first eight model parameters describing the Gaussian FLORIS wake model in FLORIDyn do not have any influence on the upstream wind

turbine (Turbine1) which is located in the free stream due to the fact that no wake effects are captured. The advection scaling factor d scales the advection speed for transporting the FLORIDyn wind field data carried by OPs. Therefore, d can not be perceived in Turbine1 in the free stream either. Based on Equation 2-21 in Chapter 2-1-6, the power production is scaled by the efficiency factor η and the factor p_p is an exponent to correct the yaw angle γ. As a consequence, during the simulation period of interest ($456s \sim 1200s$), η mainly affects Turbine1 all the time and p_p can not be slightly perceived until $800s$ when the yaw angle γ starts to change with the angular velocity $0.3°/s$ from 10° to 20°.

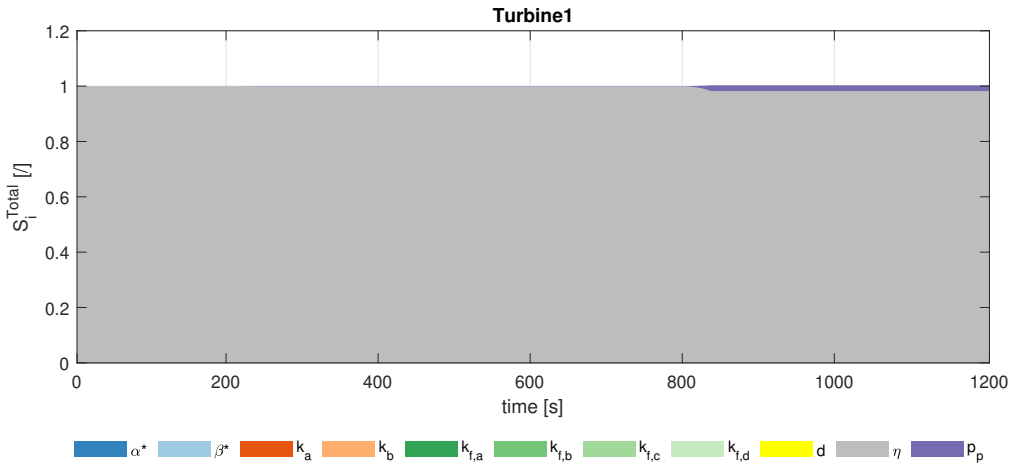


Figure 4-5: The sensitivity of the time-dependent power output of the upstream wind turbine with respect to 11 model parameters depicted by total Sobol' indices

Figure 4-6 shows that the wake effects resulting from the change of the yaw angle γ arrive at the second wind turbine (Turbine2), i.e., the first eight wake model parameters are perceived during the simulation period of interest ($456s$ $sim 1200s$). Since the wake effects are only from Turbine1 which is not far away from Turbine2, the near wake effects are mainly captured here. The near wake length x_c presented in Equation 2-11 in Section 2-1-3 depending on α^* and β^*, the wake expansion factors k_y and k_z presented in Equation 2-16 in Section 2-1-4 depending on k_a and k_b, the added turbulence I_f presented in Equation 2-18 in Section 2-1-4 depending on $k_{f,a}$, $k_{f,b}$, $k_{f,c}$ and $k_{f,d}$, jointly affect the reduction factor for computing the wind speed reduced by wake effects. Even though all eight wake model parameters indeed show up, except α^*, β^* and k_a are quite obvious to observe, the color areas corresponding to the rest are pretty small and not easy to find. The advection scaling factor d has an influence on when the wake information from Turbine1 and carried by OPs arrives at Turbine2. Therefore, d can be perceived in Turbine2. Additionally, η also has a significant influence here for the same reason as Turbine1 mentioned above. Last, p_p is not perceived because the yaw angle γ of Turbine2 always stays still with 0°.

Figure 4-7 presents that all model parameters of interest are perceived in the third wind turbine (Turbine3) except p_p because the yaw angle γ of Turbine3 also stays still with 0° all the time. Even though the color areas corresponding to k_b and $k_{f,c}$ are still inconspicuous, the other wake model parameters can be intuitively observed and especially $k_{f,a}$, $k_{f,b}$ and $k_{f,d}$ are much easier to find compared to Turbine2. This is because Turbine3 is affected by the wake

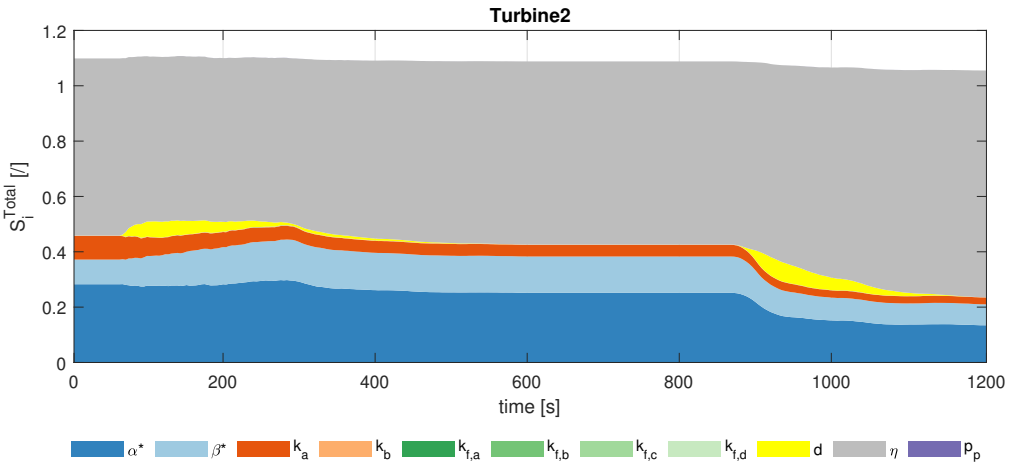


Figure 4-6: The sensitivity of the time-dependent power output of the middle wind turbine with respect to 11 model parameters depicted by total Sobol' indices

effects not only from relatively nearby Turbine2, but also from relatively far Turbine1, which means that there are more turbulence captured at Turbine3 than Turbine2. As a consequence, weight factors of the foreign turbulence influence can be perceived more easily. Additionally, as described above, d can be perceived because it affects the power production by scaling the speed for transporting the wind field data carried by OPs. Last, η also affects Turbine3 significantly for the same reason as the other two wind turbines.

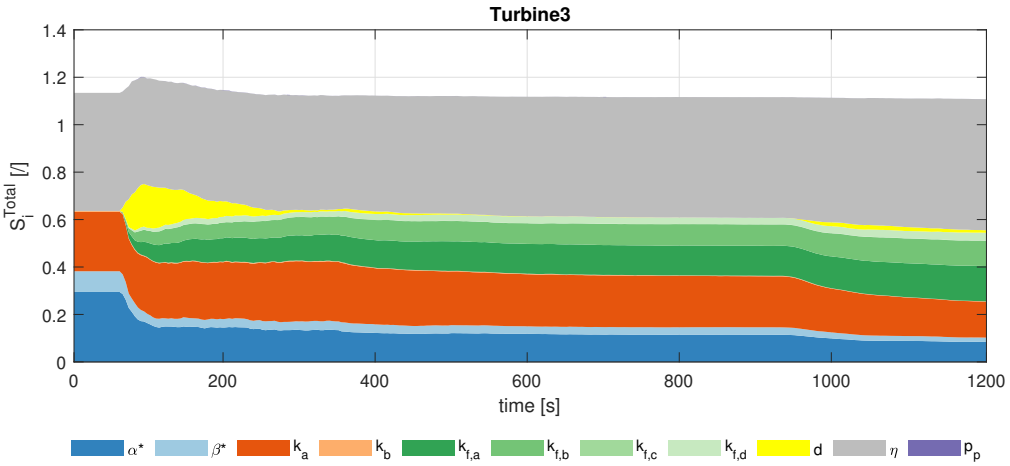


Figure 4-7: The sensitivity of the time-dependent power output of the downstream wind turbine with respect to 11 model parameters depicted by total Sobol' indices

Finally, based on the results of sensitivity analysis, the changes of all 11 model parameters of interest at different times contribute to the variations in the time-dependent power outputs of three wind turbines. Hence, all 11 model parameters summarized in Table 2-1 in Section 2-2 will be taken into account in the following Bayesian calibration part.

4-2-3 Bayesian calibration

The measurement data for the calibration is obtained from the SOWFA simulation. As described above, the model outputs are sensitive to all 11 model parameters of interest at different times. Therefore, all of these 11 model parameters are considered and the PCE-based surrogate model trained for sensitivity analysis can also be used here. Additionally, in this case, the variance σ^2 of the additive Gaussian discrepancy ε elaborated in Section 3-3-2 is assumed as 0.01. The AIES method is used by default by the UQLab software [63] to sample from the posterior distributions. The number of steps is set to 100 and the number of chains is set to 1000. The initial seeds for each chain are automatically sampled from the prior distributions. The trace plot and corresponding KDE of each model parameter are then obtained to check if the chain has arrived at its steady state.

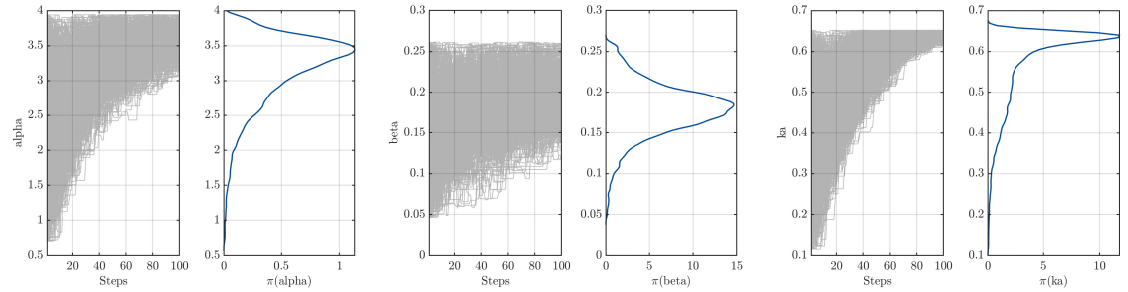
Figure 4-8 presents all trace plots and corresponding KDEs of 11 model parameters of interest. It can be observed that the KDE for each model parameter's posterior distribution will not change significantly with increasing the number of steps. Therefore, it can be concluded that the steady state of each model parameter has been reached [91].

The posterior distributions of all model parameters turn into Gaussian-like based on their Uniform-like prior distributions. The results of these Gaussian distributions are shown in Table 4-2 and visualized in Figure 4-9.

	α^*	β^*	k_a	k_b	$k_{f,a}$	$k_{f,b}$	$k_{f,c}$	$k_{f,d}$	d	η	p_p
μ	3.4	0.19	0.62	0.0043	0.7	0.94	0.035	-0.37	0.97	0.91	2.8
σ	0.22	0.026	0.027	0.00075	0.12	0.13	0.0066	0.061	0.17	0.013	0.38

Table 4-2: Mean values and standard deviations of the obtained posterior distributions for the model parameters.

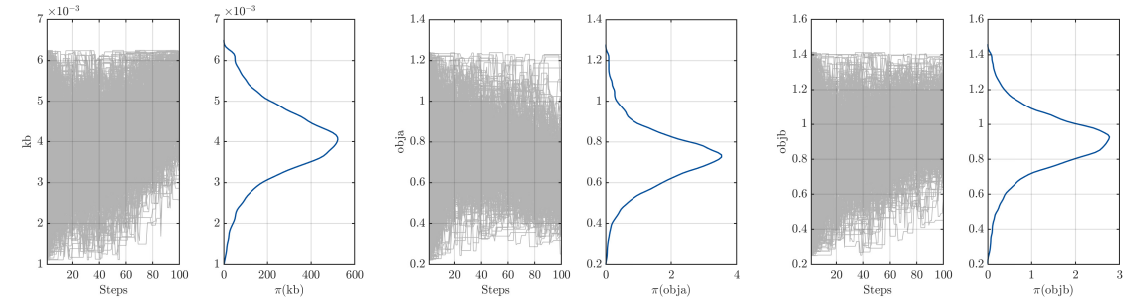
The probability of observing the model outputs depending on calibrated model parameters can be denoted by the posterior predictive distribution described in Section 3-3-4. Based on Equation 3-62, the posterior predictive distribution can be obtained with the likelihood estimated from the model evaluations and the samples of 11 model parameters' posterior distributions. Figure 4-10 presents the measurement data, the original FLORIDyn model outputs, the calibrated FLORIDyn model outputs computed with posterior mean values of 11 model parameters, the variance for the FLORIDyn model outputs estimated with 2σ, during the simulation time interval of interest ($456s \sim 1200s$). It can be observed that the variance almost encloses the SOWFA data and the calibrated FLORIDyn model outputs overlap the SOWFA data better compared to the original FLORIDyn model outputs, which means that the accuracy of the FLORIDyn model is indeed improved after being calibrated.



(a) Trace plot and corresponding KDE of α^*.

(b) Trace plot and corresponding KDE of β^*.

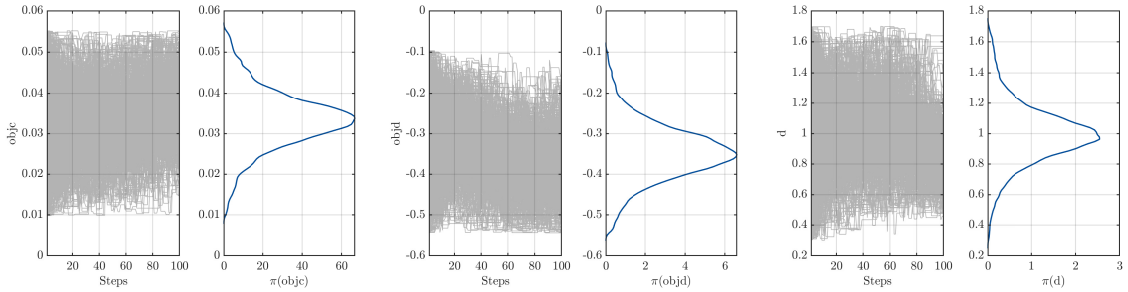
(c) Trace plot and corresponding KDE of k_a.



(d) Trace plot and corresponding KDE of k_b.

(e) Trace plot and corresponding KDE of $k_{f,a}$.

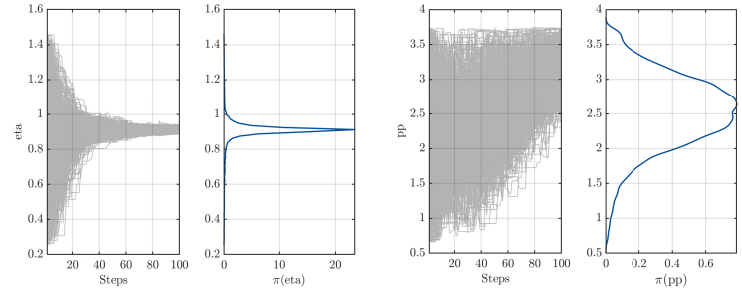
(f) Trace plot and corresponding KDE of $k_{f,b}$.



(g) Trace plot and corresponding KDE of $k_{f,c}$.

(h) Trace plot and corresponding KDE of $k_{f,d}$.

(i) Trace plot and corresponding KDE of d.



(j) Trace plot and corresponding KDE of η.

(k) Trace plot and corresponding KDE of p_p.

Figure 4-8: Trace plots and corresponding KDEs of all 11 model parameters of interest.

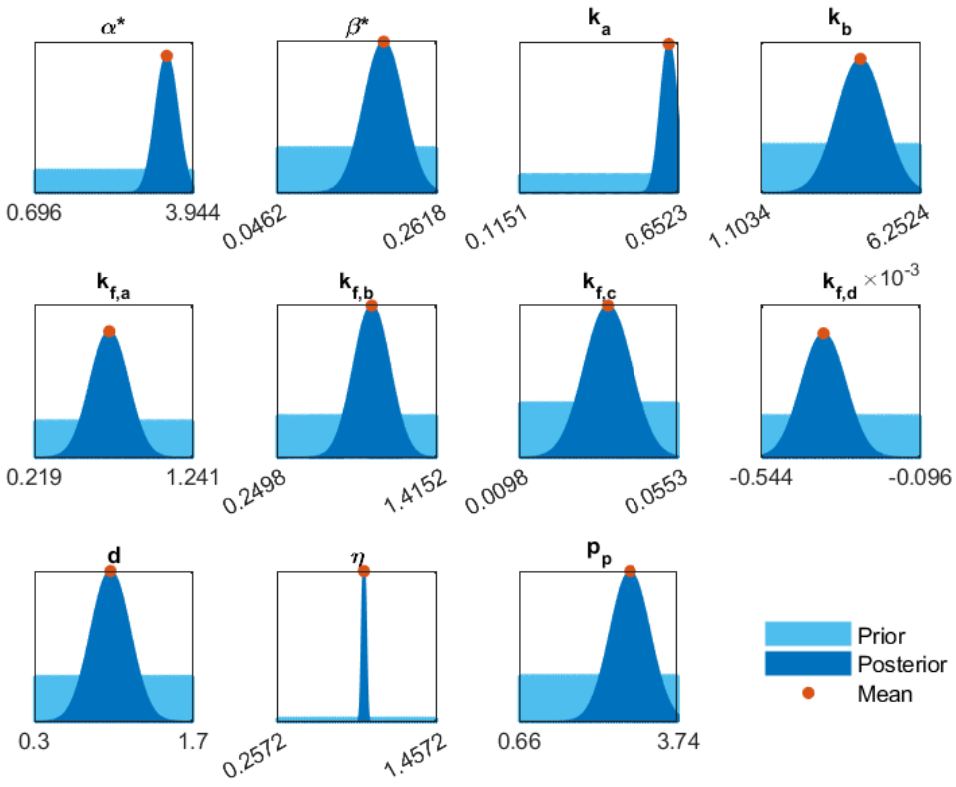


Figure 4-9: The prior and posterior distributions of 11 model parameters. The orange dots represent the mean values.

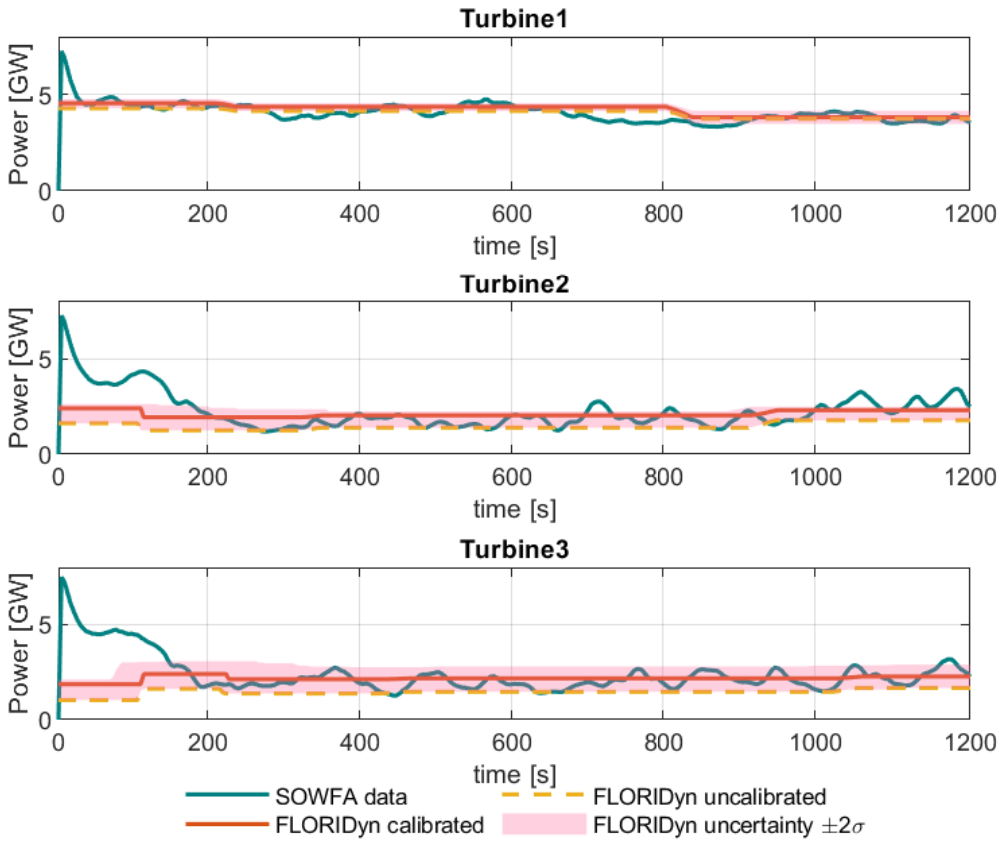


Figure 4-10: Comparison of the SOWFA data, the original FLORIDyn model outputs, the calibrated FLORIDyn model outputs and the variance for the FLORIDyn model outputs estimated with double standard deviation. The green, yellow dashed and red curves represent SOWFA, original FLORIDyn and calibrated FLORIDyn respectively. Besides, the transparent pink area indicates the variance for FLORIDyn.

Chapter 5

Summary and conclusion

This chapter first summarizes the achievements of this research. Next, the drawbacks appearing during the study and corresponding possible strategies to deal with them in the future are described. The overview of uncertainty quantification for the FLORIDyn model is presented in Section 5-1. The issues still not worked out are then described in Section 5-2. Last, the future work for dealing with those issues is discussed in Section 5-3.

5-1 Achievements

To improve the accuracy of the FLORIDyn model developed by [17], the uncertainty quantification was conducted using the UQLab software [63] in this research. As described in Chapter 2, in FLORIDyn, the wake effects are captured by the Gaussian FLORIS model [15] inherently depending on the first eight wake model parameters summarized in Table 2-1 in Section 2-2, d regulates the temporal dynamics by scaling the advection speed for transporting the wind field data carried by OPs, η and p_p scale the power outputs and the yaw angle γ respectively. As a consequence, these 11 model parameters were considered for uncertainty quantification and converted into probabilistic variables subject to uniform distributions by perturbing their mean values. However, it can result in high computational costs using the computational model directly in the research. The method to improve the efficiency of simulations is to construct a surrogate model to replace the original computational model. This research assessed two types of approaches for constructing the surrogate model which are the Kriging-based approach and the PCE-based approach respectively. The results show that the PCE-based approach can provide a more accurate surrogate model with similar efficiency compared to the Kriging-based approach in this case. The LOO error was used to quantify the accuracy and a reference value was set to 0.01 modified from [30]. A PCE-based surrogate model with almost all LOO errors smaller than 0.01 was then obtained to replace the original computational model for the following uncertainty quantification.

Sensitivity analysis uses Sobol' indices to indicate how sensitive the power outputs of each wind turbine are to each of 11 model parameters of interest at different times. Base on the

results that each of model parameters plays a role in calculating power outputs during the simulation time interval of interest, all 11 model parameters were considered in the Bayesian calibration part and the PCE-based surrogate model constructed for sensitivity analysis was therefore used for Bayesian calibration, too.

The AIES method was used to sample from the posterior distributions. The posterior predictive distribution was then obtained based on the samples of Gaussian-like posterior distributions and the likelihood estimated using model evaluations. Last, the improvement of accuracy can be observed by comparing the SOWFA data, the original FLORIDyn model outputs, the calibrated FLORIDyn model outputs and the variance for the FLORIDyn model outputs in Figure 4-10.

5-2 Drawbacks

As described in Section 4-1, the yaw angle γ of Turbine1 changes twice during the entire simulation. Therefore, there are two dynamic states captured by the FLORIDyn model which are at the time interval ($200s \sim 456s$) and the time interval ($800s \sim 1056s$) respectively. Based on Figure 4-3, for Turbine2 and Turbine3, both Kriging-based and PCE-based approaches with 250 experimental design points have better performance when applied to steady states than to dynamic states and have the worst performance when applied to the earlier dynamic state ($200s \sim 456s$). This situation keeps the same when applying the PCE-based approach with 2000 experimental design points as shown in Figure 4-4. This research focuses on the time interval ($456s \sim 1200s$) when most LOO errors are smaller than 0.01 to reduce the impact of surrogate model errors.

The trace plot and corresponding KDE of each model parameter of interest are used to approximate its posterior distribution and to check if its chain produce by the MCMC sampler has reached the steady state. Based on Figure 4-8 in Section 4-2-3, all chains of model parameters have reached their steady states, except the chain of α^* has not completely reached its steady state. Additionally, the lower and upper bounds of the prior distributions of α^* and k_a are not set properly.

5-3 Future work

Even though the accuracy of the FLORIDyn model has been improved after being calibrated, there is still room for further improvement in the future. The following items might be able to help further improve the model accuracy.

As described before, it can result in massive computational costs when applying the FLORIDyn model directly to uncertainty quantification. It is necessary to replace the FLORIDyn model with a surrogate model to reduce the computing costs, even though the errors between the surrogate model and the original model will have a negative influence on the follow-up study. Therefore, it is helpful to reduce the influence of errors as much as possible. In the current research, the LAR algorithm with default settings in the PCE module of the UQLab software [63] was used to train the surrogate model. Even though the user manual [62] for this module says the default settings are suitable for most cases, it is worth trying other

settings to search smaller LOO errors, or exploring introducing an early stop criterion to find the minimum LOO error which is similar to that shown in Figure 3-6.

Based on the results of sensitivity analysis, the power outputs are less sensitive to k_b and $k_{f,c}$ than to the other model parameters. Therefore, one possible method that sacrifices relatively little accuracy to improve efficiency is to reduce the dimension of the model parameter vector for Bayesian calibration. To do so, k_b and $k_{f,c}$ are defined as deterministic variables equal to their mean values instead of probabilistic variables subject to uniform distributions when constructing the surrogate model for Bayesian calibration.

Last, the number of steps set for the AIES algorithm can be further increased to ensure that the chain of each model parameter has completely reached its steady state. Additionally, another possible strategy to further improve the performance of Bayesian calibration is to jointly infer the discrepancy term ε with other model parameters by calibrating a new vector consisting of ε and other model parameters instead of assuming it as an additive Gaussian discrepancy. To do so, ε can be similarly defined as a probabilistic variable priorly subject to the uniform distribution. Adjusting lower and upper bounds of parameters' prior distributions to ensure that their mean values stay in the middle of the ranges as much as possible or directly expanding the intervals of their prior distributions might be also useful for the improvement.

Bibliography

[1] 2030 Climate Target Plan. https://ec.europa.eu/clima/eu-action/european-green-deal/2030-climate-target-plan_en Accessed Dec 16, 2021.

[2] CO2 Emissions. https://www.iea.org/reports/global-energy-review-2021/co2-emissions Accessed Dec 15, 2021.

[3] NREL. Simulator for Offshore Wind Farm Applications. https://github.com/dcsale/SOWFA Accessed Nov 26, 2021.

[4] OpenCFD Ltd. OpenFOAM. https://www.openfoam.com Accessed Nov 26, 2021.

[5] Siemens-Gamesa Renewable Energy and Wake Adapt solution. https://www.youtube.com/watch?v=UP2yJqGQ2FU Accessed Dec 20, 2021.

[6] Surrogate Model. https://deepchecks.com/glossary/surrogate-model/ Accessed Nov 08, 2022.

[7] Voronoi method. https://www.mathworks.com/help/matlab/ref/voronoi.html Accessed June 08, 2021.

[8] Wake Effect. https://www.wind-energy-the-facts.org/wake-effect.html Accessed May 30, 2021.

[9] What is SCADA? https://inductiveautomation.com/resources/article/what-is-scada Accessed May 11, 2021.

[10] What is the Levelized Cost of Energy (LCOE)? https://corporatefinanceinstitute.com/resources/knowledge/finance/levelized-cost-of-energy-lcoe/ Accessed Apr 21, 2021.

[11] Milton Abramowitz and Irena A Stegun. Handbook of mathematical functions, new york: Doverpublications, 1965.

[12] Jacob Aho, Andrew Buckspan, Jason Laks, Paul Fleming, Yunho Jeong, Fiona Dunne, Matthew Churchfield, Lucy Pao, and Kathryn Johnson. A tutorial of wind turbine control for supporting grid frequency through active power control. In *2012 American Control Conference (ACC)*, pages 3120–3131. IEEE, 2012.

[13] Jennifer Annoni, Pieter M.O. Gebraad, Andrew K. Scholbrock, Paul A. Fleming, and J.W. van Wingerden. Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model. *Wind Energy*, 19(6):1135–1150, 2016.

[14] Christian Bak, Frederik Zahle, Robert Bitsche, Taeseong Kim, Anders Yde, Lars Christian Henriksen, Morten Hartvig Hansen, Jose Pedro Albergaria Amaral Blasques, Mac Gaunaa, and Anand Natarajan. The dtu 10-mw reference wind turbine. In *Danish wind power research 2013*, 2013.

[15] Majid Bastankhah and Fernando Porté-Agel. Experimental and theoretical study of wind turbine wakes in yawed conditions. *Journal of Fluid Mechanics*, 806:506, 2016.

[16] Christopher J. Bay, Jennifer Annoni, Timothy Taylor, Lucy Pao, and Kathryn Johnson. Active power control for wind farms using distributed model predictive control and nearest neighbor communication. In *2018 Annual American Control Conference (ACC)*, pages 682–687. IEEE, 2018.

[17] M. Becker. Maturing FLORIS towards a Dynamic Wind Farm Model. 2020.

[18] Marc Berveiller, Bruno Sudret, and Maurice Lemaire. Stochastic finite element: a non intrusive approach by regression. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 15(1-3):81–92, 2006.

[19] Fernando D. Bianchi, Hernán De Battista, and Ricardo J. Mantz. *Wind turbine control systems: principles, modelling and gain scheduling design*. Springer Science & Business Media, 2006.

[20] Géraud Blatman. *Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis*. PhD thesis, Clermont-Ferrand 2, 2009.

[21] Géraud Blatman and Bruno Sudret. An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25(2):183–197, 2010.

[22] Géraud Blatman and Bruno Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of computational Physics*, 230(6):2345–2367, 2011.

[23] Sjoerd Boersma, Bart Doekemeijer, Mehdi Vali, Johan Meyers, and J.W. van Wingerden. A control-oriented dynamic wind farm model: Wfsim. *Wind Energy Science*, 3(1):75–95, 2018.

[24] Sjoerd Boersma, Bart M. Doekemeijer, Pieter M.O. Gebraad, Paul A. Fleming, Jennifer Annoni, Andrew K. Scholbrock, Joeri Alexis Frederik, and J.W. van Wingerden. A tutorial on control-oriented modeling and control of wind farms. In *2017 American control conference (ACC)*, pages 1–18. IEEE, 2017.

[25] Sjoerd Boersma, Vahab Rostampour, Bart Doekemeijer J.W. van Wingerden, and Tamás Keviczky. A model predictive wind farm controller with linear parameter-varying models. *IFAC-PapersOnLine*, 51(20):241–246, 2018.

[26] Filippo Campagnolo, Vlaho Petrović, Johannes Schreiber, Emmanouil M. Nanos, Alessandro Croce, and Carlo L. Bottasso. Wind tunnel testing of a closed-loop wake deflection controller for wind farm power maximization. In *Journal of Physics: Conference Series*, volume 753, page 032006. IOP Publishing, 2016.

[27] Pantelis Capros, A. De Vita, N. Tasios, P. Siskos, M. Kannavou, A. Petropoulos, S. Evangelopoulou, M. Zampara, D. Papadopoulos, Ch. Nakos, et al. Eu reference scenario 2016-energy, transport and ghg emissions trends to 2050. 2016.

[28] Francesco Castellani and Andrea Vignaroli. An application of the actuator disc model for wind turbine wakes calculations. *Applied Energy*, 101:432–440, 2013.

[29] Francisco Díaz-González, Melanie Hau, Andreas Sumper, and Oriol Gomis-Bellmunt. Participation of wind power plants in system frequency control: Review of grid code requirements and control methods. *Renewable and Sustainable Energy Reviews*, 34:551–564, 2014.

[30] Vinit V Dighe, Marcus Becker, Tuhfe Göçmen, Benjamin Sanderse, and Jan-Willem van Wingerden. Sensitivity analysis and bayesian calibration of a dynamic wind farm control model: Floridyn. In *Journal of Physics: Conference Series*, volume 2265, page 022062. IOP Publishing, 2022.

[31] Bart M. Doekemeijer, Stefan Kern, Sivateja Maturu, Stoyan Kanev, Bastian Salbert, Johannes Schreiber, Filippo Campagnolo, Carlo L. Bottasso, Simone Schuler, Friedrich Wilts, et al. Field experiment for open-loop yaw-based wake steering at a commercial onshore wind farm in italy. *Wind Energy Science*, 6(1):159–176, 2021.

[32] Bart M. Doekemeijer, J.W. van Wingerden, and Paul A. Fleming. A tutorial on the synthesis and validation of a closed-loop wind farm controller using a steady-state surrogate model. In *2019 American Control Conference (ACC)*, pages 2825–2836. IEEE, 2019.

[33] B.M. Doekemeijer. *Closing the loop in model-based wind farm control*. PhD thesis, Delft University of Technology, 2020.

[34] Vincent Dubourg. *Adaptive surrogate models for reliability analysis and reliability-based design optimization*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2011.

[35] Erik Ela, V. Gevorgian, P. Fleming, Y.C. Zhang, M. Singh, E. Muljadi, A. Scholbrook, J. Aho, A. Buckspan, L. Pao, et al. Active power controls from wind power: Bridging the gaps. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2014.

[36] Alayna Farrell, Jennifer King, Caroline Draxl, Rafael Mudafort, Nicholas Hamilton, Christopher J. Bay, Paul Fleming, and Eric Simley. Design and analysis of a spatially heterogeneous wake. *Wind Energy Sci. Discuss.(published online). https://doi.org/10.5194/wes-2020-57*, 2020.

[37] Paul Fleming, Jennifer Annoni, Jigar J. Shah, Linpeng Wang, Shreyas Ananthan, Zhijun Zhang, Kyle Hutchings, Peng Wang, Weiguo Chen, and Lin Chen. Field test of wake steering at an offshore wind farm. *Wind Energy Science*, 2(1):229–239, 2017.

[38] Paul Fleming, Jennifer King, Katherine Dykes, Eric Simley, Jason Roadman, Andrew Scholbrock, Patrick Murphy, Julie K. Lundquist, Patrick Moriarty, Katherine Fleming, et al. Initial results from a field campaign of wake steering applied at a commercial wind farm–part 1. *Wind Energy Science*, 4(2):273–285, 2019.

[39] Paul A. Fleming, Pieter M.O. Gebraad, Sang Lee, J.W. van Wingerden, Kathryn Johnson, Matt Churchfield, John Michalakes, Philippe Spalart, and Patrick Moriarty. Evaluating techniques for redirecting turbine wakes using sowfa. *Renewable Energy*, 70:211–218, 2014.

[40] Daniel Foreman-Mackey, David W Hogg, Dustin Lang, and Jonathan Goodman. emcee: the mcmc hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306, 2013.

[41] J.A. Frederik. Dynamic wind farm control using the wfsim flow model. 2017.

[42] Aaron Garrett, Joshua New, and Theodore Chandler. Evolutionary tuning of building models to monthly electrical consumption. *ASHRAE Transactions*, 119(2):89–100, 2013.

[43] Pieter M.O. Gebraad, Paul A. Fleming, and J.W. van Wingerden. Wind turbine wake estimation and control using floridyn, a control-oriented dynamic wind plant model. In *2015 American Control Conference (ACC)*, pages 1702–1708. IEEE, 2015.

[44] Pieter M.O. Gebraad, F.W. Teeuwisse, J.W. van Wingerden, Paul A. Fleming, S.D. Ruben, J.R. Marden, and L.Y. Pao. Wind plant power optimization through yaw control using a parametric model for wake effects—a cfd simulation study. *Wind Energy*, 19(1):95–114, 2016.

[45] Pieter M.O. Gebraad, F.W. Teeuwisse, J.W. van Wingerden, Paul A. Fleming, Shalom D. Ruben, Jason R. Marden, and Lucy Y. Pao. A data-driven model for wind plant power optimization by yaw control. In *2014 American Control Conference*, pages 3128–3134. IEEE, 2014.

[46] Pieter M.O. Gebraad and J.W. van Wingerden. A control-oriented dynamic model for wakes in wind plants. In *Journal of Physics: Conference Series*, volume 524, page 012186. IOP Publishing, 2014.

[47] A. Gelman, John B C., Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[48] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[49] Graham Glen and Kristin Isaacs. Estimating sobol sensitivity indices using correlations. *Environmental Modelling & Software*, 37:157–166, 2012.

[50] Jay P. Goit and Johan Meyers. Optimal control of energy extraction in wind-farm boundary layers. *Journal of Fluid Mechanics*, 768:5–50, 2015.

[51] Jonathan Goodman and Jonathan Weare. Ensemble samplers with affine invariance. *Communications in applied mathematics and computational science*, 5(1):65–80, 2010.

[52] Michael F. Howland, Sanjiva K. Lele, and John O. Dabiri. Wind farm power optimization through wake steering. *Proceedings of the National Academy of Sciences*, 116(29):14495–14500, 2019.

[53] Niels Otto Jensen. A note on wind generator interaction. 1983.

[54] Jari Kaipio and Erkki Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media, 2006.

[55] S.K. Kanev, F.J. Savenije, and W.P. Engels. Active wake control: An approach to optimize the lifetime operation of wind farms. *Wind Energy*, 21(7):488–501, 2018.

[56] Jonas Kazda and Nicolaos A. Cutululis. Model-optimized dispatch for closed-loop power control of waked wind farms. *IEEE Transactions on Control Systems Technology*, 28(5):2029–2036, 2019.

[57] C. Lataniotis, D. Wicaksono, S. Marelli, and B. Sudret. UQLab user manual – Kriging (Gaussian process modeling). Technical report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich,Switzerland, 2022. Report UQLab-V2.0-105.

[58] P.B.S. Lissaman. Energy effectiveness of arbitrary arrays of wind turbines. *Journal of Energy*, 3(6):323–328, 1979.

[59] Jun S Liu and Jun S Liu. *Monte Carlo strategies in scientific computing*, volume 75. Springer, 2001.

[60] Adolphus Lye, Alice Cicirello, and Edoardo Patelli. An efficient and robust sampler for bayesian inference: Transitional ensemble markov chain monte carlo. *Mechanical Systems and Signal Processing*, 167:108471, 2022.

[61] S. Marelli, C. Lamas, K. Konakli, C. Mylonas, P. Wiederkehr, and B. Sudret. UQLab user manual – Sensitivity analysis. Technical report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich,Switzerland, 2022. Report UQLab-V2.0-106.

[62] S. Marelli, N. Lüthen, and B. Sudret. UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich,Switzerland, 2022. Report UQLab-V2.0-104.

[63] S. Marelli and B. Sudret. UQLab: A framework for uncertainty quantification in Matlab, Proc. 2nd Int. Conf. onVulnerability. Technical report, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom, 2014. 2554-2563.

[64] Davide Medici. *Experimental studies of wind turbine wakes: power optimisation and meandering*. PhD thesis, KTH, 2005.

[65] Ralph T Muehleisen and Joshua Bergerson. Bayesian calibration-what, why and how. 2016.

[66] Wim Munters and Johan Meyers. An optimal control framework for dynamic induction control of wind farms and their interaction with the atmospheric boundary layer. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2091):20160100, 2017.

[67] Wim Munters and Johan Meyers. Dynamic strategies for yaw and induction control of wind farms based on large-eddy simulation and optimization. *Energies*, 11(1):177, 2018.

[68] Karl Nilsson, Wen Z Shen, Jens N Sørensen, Simon-Philippe Breton, and Stefan Ivanell. Validation of the actuator line method using near wake measurements of the mexico rotor. *Wind Energy*, 18(3):499–514, 2015.

[69] Tom Obdam, L.W.M.M. Rademakers, Henk Braam, and Peter Eecen. Estimating costs of operation & maintenance for offshore wind farms. In *Proceedings of European offshore wind energy conference*, pages 4–6, 2007.

[70] Steffen Raach, Bart Doekemeijer, Sjoerd Boersma, J.W. van Wingerden, and Po Wen Cheng. Feedforward-feedback wake redirection for wind farm control. *Wind Energy Science Discussions*, pages 1–18, 2019.

[71] Siegfried Raasch and Michael Schröter. P3. 13 a large-eddy simulation model performing on massively parallel computers. 2001.

[72] TJ Rao and S Sengupta. Calibrated linear unbiased estimators in finite population sampling. *Journal of statistical planning and inference*, 140(3):652–658, 2010.

[73] Carl Edward Rasmussen, Christopher KI Williams, et al. *Gaussian processes for machine learning*, volume 1. Springer, 2006.

[74] T Agami Reddy, Itzhak Maor, and Chanin Panjapornpon. Calibrating detailed building energy simulation programs with measured data—part i: General methodology (rp-1051). *Hvac&R Research*, 13(2):221–241, 2007.

[75] Pascal Richtera, Jannick Woltersa, and Martin Franka. Uncertainty quantification for the planning of offshore wind farms using monte carlo and sparse grid.

[76] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

[77] Thomas J Santner, Brian J Williams, William I Notz, and Brain J Williams. *The design and analysis of computer experiments*, volume 1. Springer, 2003.

[78] David Schlipf, Davide Trabucchi, Oliver Bischoff, Martin Hofsäß, Jakob Mann, Torben Mikkelsen, Andreas Rettenmeier, Juan José Trujillo, and Martin Kühn. Testing of frozen turbulence hypothesis for wind turbine applications with a scanning lidar system. 2011.

[79] Carl R. Shapiro, Johan Meyers, Charles Meneveau, and Dennice F. Gayme. Wind farms providing secondary frequency regulation: Evaluating the performance of model-based receding horizon control. *Wind Energy Science*, 3(1):11–24, 2018.

[80] Eric Simley, Paul Fleming, and Jennifer King. Field validation of wake steering control with wind direction variability. In *Journal of Physics: Conference Series*, volume 1452, page 012012. IOP Publishing, 2020.

[81] Ilya M Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001.

[82] Vedrana Spudić, Christian Conte, Mato Baotić, and Manfred Morari. Cooperative distributed model predictive control for wind farms. *Optimal Control Applications and Methods*, 36(3):333–352, 2015.

[83] Bruno Sudret. Uncertainty propagation and sensitivity analysis in mechanical models–contributions to structural reliability and stochastic spectral methods. *Habilitationa diriger des recherches, Université Blaise Pascal, Clermont-Ferrand, France*, 147:53, 2007.

[84] Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.

[85] Geoffrey Ingram Taylor. The spectrum of turbulence. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 164(919):476–490, 1938.

[86] Joel Tellinghuisen. Weighted least-squares in calibration: What difference does it make? *Analyst*, 132(6):536–543, 2007.

[87] TU Delft DCSC Data Driven Control Group. FLORISSE_M. https://github.com/TUDelft-DataDrivenControl/FLORISSE_M Accessed May 26, 2021.

[88] Maarten T. van Beek, Axelle Viré, and Søren J. Andersen. Sensitivity and uncertainty of the floris model applied on the lillgrund wind farm. *Energies*, 14(5):1293, 2021.

[89] Maarten J. van den Broek and J.W. van Wingerden. Dynamic flow modelling for model-predictive wind farm control. In *Journal of Physics: Conference Series*, volume 1618, page 022023. IOP Publishing, 2020.

[90] Helmut Vogel. A better way to construct the sunflower head. *Mathematical biosciences*, 44(3-4):179–189, 1979.

[91] P.-R. Wagner, J. Nagel, S. Marelli, and B. Sudret. UQLab user manual – Bayesian inversion for model calibration andvalidation. Technical report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich,Switzerland, 2022. Report UQLab-V2.0-113.

[92] Matt P Wand and M Chris Jones. *Kernel smoothing*. CRC press, 1994.

[93] Eric W. Weisstein. Statistical Correlation. MathWorld. https://mathworld.wolfram.com/StatisticalCorrelation.html Accessed Jan 04, 2023.

[94] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.

[95] Jincheng Zhang and Xiaowei Zhao. Quantification of parameter uncertainty in wind farm wake modeling. *Energy*, 196:117065, 2020.

Glossary

List of Symbols

Abbreviations

ADM	Actuator Disc Model
AEP	Annual Energy Production
AIES	Affine Invariant Ensemble Sampler
ALM	Actuator Line Method
BLUP	Best Linear Unbiased Prediction
CFD	Computational Fluid Dynamics
CV	Cross Validation
FLORIDyn	Flow Redirection and Induction Dynamics
FLORIS	Flow Redirection and Induction in Steady-State
KDE	Kernel Density Estimator
LAR	Least Angle Regression
LCOE	Levelized Costs Of Energy
LES	Large Eddy Simulation
LHS	Latin Hypercube Sampling
LOO	Leave One Out
MCMC	Markov Chain Monte Carlo
ML	Maximum Likelihood
NREL	National Renewable Energy Laboratory
OLS	Ordinary Least Squares
OP	Observation Point
PALM	PArallelized LES Model
PC	Polynomial Chaos
PCE	Polynomial Chaos Expansions
PDF	Probability Density Function
QoI	Quantity of Interest
SCADA	Supervisory Control and Data Acquisition
SOWFA	Simulator of On/Offshore Wind Farm Applications
UQ	Uncertainty Quantification
WFC	Wind Farm Control
WFSim	Wind Farm Simulator

Symbols

α^*, β^*	Weight constraints in the potential core length calculation	$[-]$
$\boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x})$	Trend in the Kriging-based approach	$[-]$
$\boldsymbol{\mathcal{X}}$	Experimental design vector	$[-]$
$\boldsymbol{\mathcal{Y}}$	Model observation vector	$[-]$
$\boldsymbol{\Theta}$	Random vector associated with the model parameter vector $\boldsymbol{\theta}_M$	$[-]$
$\boldsymbol{\theta}_M$	Model parameter vector	$[-]$
$\boldsymbol{x}$	Model input vector	$[-]$
δ	Deflection	$[m]$
δ_{jk}	Kronecker symbol	$[-]$
η	Efficiency factor	$[-]$
γ	Yaw angle	$[rad]$
λ	Tip speed ratio	$[-]$
$\mathbf{E}$	Expectation operator	$[-]$
$\mathbf{R}_{01}$	Rotational matrix	$[-]$

$\mathbf{r}_0$	Position in the world coordinate system	$[m]$
$\mathbf{r}_1$	Position in the wake coordinate system	$[m]$
$\mathbf{t}_0$	Position of the respective turbine in the world coordinate system	$[m]$
$\mathbf{u}$	Wind speed vector	$[m/s]$
$\mathbf{u}_{red}$	Reduced wind speed vector	$[m/s]$
$\mathcal{D}_{\boldsymbol{X}}$	Available integration domain	$[-]$
$\mathcal{K}$	Transition probability in the MCMC algorithm	$[-]$
$\mathcal{L}$	Likelihood function in the Bayesian inference	$[-]$
$\mathcal{L}_K$	Likelihood function in the Kriging-based approach	$[-]$
$\mathcal{M}$	Model output	$[-]$
$\mathcal{M}^K$	Kriging-based surrogate model	$[-]$
$\mathcal{M}^{PC}$	PCE-based surrogate model	$[-]$
$\mathcal{N}$	Gaussian distribution	$[-]$
$\mathcal{O}$	Observation set	$[-]$
$\mathcal{Y}$	Measurement data set	$[-]$
ν_y, ν_z	Relative y_1/z_1 direction	$[-]$
ρ	Air density	$[kg/m^3]$
Σ	Residual covariance matrix	$[-]$
σ^2	Variance	$[-]$
$\sigma_{y,fw}$	Standard deviation of the far wake in y_1 direction	$[m]$
$\sigma_{y,nw}$	Standard deviation of the near wake in y_1 direction	$[m]$
$\sigma_{z,fw}$	Standard deviation of the far wake in z_1 direction	$[m]$
$\sigma_{z,nw}$	Standard deviation of the near wake in z_1 direction	$[m]$
θ	Deflection angle at the rotor	$[rad]$
θ_{pc}	Deflection angle in the potential core	$[rad]$
ε	Scalar discrepancy term	$[-]$
φ	Wind angle	$[^\circ]$
a	Axial induction factor	$[-]$
$A_{overlap}$	Area of the rotor plane where a foreign wake overlaps	$[m^2]$
A_{rotor}	Area of the rotor plane	$[m^2]$
C_P	Power coefficient	$[-]$
C_T	Thrust coefficient	$[-]$
Cov	Covariance operator	$[-]$
D	Rotor diameter	$[m]$
d	Factor added to regulate the temporal dynamics	$[-]$
I	Turbulence intensity	$[-]$
I_{amb}	Ambient turbulence	$[-]$
I_f	Added turbulence	$[-]$
k	Index counting the time step	$[-]$
$K(\cdot)$	Kernel function	$[-]$
k_a, k_b	Weight constraints in the wake expansion calculation	$[-]$
$k_{f,a}, ..., k_{f,d}$	Weight factors of the foreign turbulence influence	$[-]$
$K_h(\cdot)$	Scaled Kernel function	$[-]$
k_y, k_z	Expansion factors in y_1/z_1 direction	$[-]$
N_θ	Number of model parameters	$[-]$
n_T	Number of turbines	$[-]$
N_w	Number of walkers (chains) in the AIES algorithm	$[-]$
P	Power production	$[W]$
P_k	Acceptance probability in the AIES algorithm	$[-]$
p_p	Factor for correcting the power coefficient under yawed operating conditions	$[-]$
R	Correlation function in the Kriging-based approach	$[-]$
r	Reduction factor	$[-]$
r_c	Potential core reduction factor	$[-]$
r_{fw}	Far wake reduction factor	$[-]$
r_{nw}	Reduction factor acting on the transition from the potential core to the free stream in cross wind direction	$[-]$
$r_{pc_{y_1}}$	Potential core radius in y_1 direction	$[m]$
$r_{pc_{z_1}}$	Potential core radius in z_1 direction	$[m]$
S_{ij}	Second order Sobol' index	$[-]$
S_i	First order Sobol' index	$[-]$
S_i^{Total}	Total Sobol' index	$[-]$
T	Number of time steps	$[-]$

t	Time step	$[-]$	y	Scalar output of $\mathcal{M}$	$[-]$
u	Free wind speed	$[m/s]$	y_0	y coordinate in the world coordinate system	$[m]$
u_c	Wind speed in the potential core	$[m/s]$	y_1	y coordinate in the wake coordinate system (cross wind)	$[m]$
u_{eff}	Effective wind speed	$[m/s]$	z_0	z coordinate in the world coordinate system	$[m]$
u_{OP}	Speed of an OP	$[m/s]$	z_1	z coordinate in the wake coordinate system (cross wind)	$[m]$
u_{red}	Reduced wind speed	$[m/s]$	Z_e	Evidence term	$[-]$
u_R	Wind speed at the rotor plane	$[m/s]$	z_h	Nacelle height	$[m]$
x_0	x coordinate in the world coordinate system	$[m]$			
x_1	x coordinate in the wake coordinate system (down wind)	$[m]$			
x_c	Potential core length	$[m]$			