WIND FARM MODELING FOR DESIGN AND ACTIVE POWER CONTROL OF WIND FARMS

by
Genevieve M. Starke

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
March, 2022

Abstract

As the interest in reducing carbon emissions continues, wind energy has become a prominent component of a carbon-free electrical supply. However, as wind continues to grow its presence in the electrical grid, a better understanding of its potential contributions is needed. One of the components that is required for this understanding is deeper knowledge of the physics of wind farms to improve physics-based modeling and more accurately predict the wind farm power output. Another component is discovering the extent to which wind farms can dynamically respond to the needs of the grid through power tracking where the wind farm tracks a power reference signal controlled by the grid operator.

In this work, we first present a coupled physics-based model for a wind farm to predict the total power out of wind farms of arbitrary geometry. The model combines two reduced order models, the first with a smaller scale on the order of individual turbines, and the other with a larger scale on the order of the scale of the whole farm. By coupling these two models, and the physics that they capture, the coupled model is able to provide insight and information on the behavior of the wind farm and predict the power output for multiple wind farm configurations. The results are verified with multiple large eddy simulation (LES) wind farm codes.

This work also presents a control-oriented wind farm model and its application to active power control. This control-oriented model uses a graph theory approach to represent the wind farm and the interactions between the turbines. This enables the model to represent dynamic changes that affect the wind farm, such as a change in

wind direction or a dynamic yaw change. The effect of such changes is challenging to represent with conventional wake models. The model was validated for both representing a dynamic wind direction change and a dynamic yaw change. This model was incorporated into an inner and outer loop control framework where the outer loop consists of a yaw model-constrained optimal control and the inner loop consists of a pitch control. The controller was applied to an LES wind farm plant and showed the potential of dynamic yaw control in different wind farm operating conditions.

Thesis Readers

Dr. Dennice Gayme (Primary Advisor)
Associate Professor and Carol Croft Linde Faculty Scholar
Department of Mechanical Engineering
Johns Hopkins University

Dr. Charles Meneveau (Secondary Advisor)
Louis M. Sardella Professor
Department of Mechanical Engineering
Johns Hopkins University

Dr. Benjamin Hobbs
Theodore M. and Kay W. Schad Professor of Environmental Management
Department of Environmental Health and Engineering
Johns Hopkins University

Dedicated to my loving family

and all those whose love and support brought me here

Thank you.

Acknowledgments

I so grateful to the countless people who have supported me and enabled me to pursue and complete a doctoral degree. I would like to first thank my advisors Dennice Gayme and Charles Meneveau for their consistent support, encouragement, and guidance through this process. Thank you to Jennifer King for her technical guidance and encouragement through our collaborations. I would also like to thank Professor Benjamin Hobbs for serving on my thesis committee, as well as Professors Louis Whitcomb, Enrique Mallada, and Harihar Rajaram who served on my GBO committee, and the many wonderful professors who made my education possible. I would also like to thank Professor Jacques Lewalle for fostering the first flickers of my interest in research. I would like to thank Melissa Green and Gina Lee-Glauser for their wonderful encouragement and support. I would also like to thank Carl Shapiro, who was an invaluable mentor at Hopkins and paved the way for my project to exist. This dissertation was also made possible with support from the National Science Foundation Graduate Research Fellowship Program and computational resources from the Maryland Advanced Research Computer Center (MARCC).

I would like to thank all of my colleagues and friends at Hopkins, who carried me through this degree: Ben, Srivaths, Karthik, Aditya, Ghanesh, Mitchell, Chang, Chengda, Rajni, Pengcheng, Xiaowei, Kristofer, Aaron, Giray, Yue, Aishwarya, and Manuel. Finally, I am so very grateful for the boundless love and support from my parents, Don and Sally, from my sister Natalie and my brother Peter, who have enabled my achievements from the start and have never faltered in all the years in

between. And also thank you to Fletcher for being himself, and thus the best cat one could ask for.

Contents

List of Figures

Chapter 1

Introduction

In the past few years, the installed capacity of renewable energy sources, including solar, wind, and hydro-power, has grown significantly [1]. Additionally, the demand for these renewable sources continues to grow as low or carbon free energy sources. These sources are attractive to not only replace current sources of electricity, but also to aid in electrifying other sectors, such as transportation [2–4]. As a prominent renewable energy source, wind energy will play a large role in this shift. Some estimates have wind energy growing by as much as a factor of 10 by 2050 [5]. Indeed, wind energy has already grown from a niche provider of electricity to providing about 5% of the global electricity demand [6]. With the opportunity for growth and low emissions, wind energy is poised to become a much larger part of our electricity supply.

However, the variable nature of wind energy could pose a challenge in incorporating more wind energy sources in the electrical grid. The current electrical grid in the United States operates at a frequency of about $60Hz$, which is an artifact of AC power currents produced by electromagnetic generators, and needs to stay very close to this operating frequency [7]. Conventional generators not only meet the demand for electricity, but also actively participate in frequency regulation to ensure the grid is adequately balanced and avoid blackouts. Unlike conventional generators, the output of a wind farm is heavily dependent on the current weather conditions and also not always predicable in advance, adding uncertainty to the power output [8]. As wind

energy becomes an increasingly larger portion of the electricity supply, a wind farm must also be able to provide some generator services to enable grid stability [7, 9]. While some of these services can be achieved by curtailing the output of wind farms so they contain a reserve of power, it is against the economic incentives for a wind farm to operate below maximum power [8]. The goal of this research is then to understand the physics of wind farms in more depth so that the power predictions become more accurate, and also to enable wind farms to provide generator services with as little curtailment as possible.

1.1 Understanding and Modeling Wind Energy Physics

The first goal mentioned above is improving our understanding of the physics inside a wind farm. The physics of a wind farm start with the atmospheric boundary layer (ABL), which is the layer in the earth's atmosphere that interacts with the ground. The kinetic energy in the atmosphere is a result of the uneven heating provided by the sun and the Coriolis forces from the earth's rotation, which prompts the movement of the air in the atmosphere [4]. The ABL typically has a height of 1-2 kilometers and changes throughout the day in response to the heating of the sun [10]. Depending on current conditions, the ABL can promote more mixing in the air (convective), or depress mixing (stable), with the intermediate regime between the two being called the neutral ABL [11]. The velocity profile for a neutrally stable ABL most often follows the logarithmic or Monin-Obukhov similarity law [10]. This neutrally stable ABL will be the atmospheric condition addressed in this work.

Wind turbines exist within this turbulent layer and extract energy from the flow when the wind turns the blades of the wind turbine. This extraction results in an area of reduced velocity behind the turbine called the wake of the turbine, which is convected and eventually dissipated by the ABL. Turbines in wind farms interact with

each other through these wakes, which results in aerodynamic coupling throughout the wind farm. One of the main goals of physics-based modeling is to better understand the behavior and the interactions between the wakes in the wind farm, as well as the interactions between the wakes and the ABL, in order to better predict the future behavior and power output of the wind farm.

1.1.1 Physics-Based Analytical Modeling

Analytical modeling of wind farms has traditionally fallen into two main categories: (1) wake models that focus on the velocity deficits (wakes) of turbines and the interactions among the turbine wakes and (2) top-down representations that examine the overall effect of the farm on the ABL. Wake models describe the magnitude, profile and evolution of the velocity deficits arising from the mean-flow kinetic energy and momentum extraction by individual turbines. This velocity deficit profile and wake growth law is then combined with a superposition law, which is used to capture wake interactions between upstream turbines in order to predict the mean velocity seen by each turbine in the array. There is a vast literature proposing different velocity deficit representations that range from analytical functions [12–16] to experimental [17, 18] or data-driven parameterizations [19]. The wake growth is assumed to be linear to represent turbulent diffusion. A number of different superposition laws for the deficits [20–23] have also been proposed that rely on a the physics principles of kinetic energy, mass balance and momentum balance, most commonly leading to quadratic or linear superposition of the wakes. For example, the widely used Jensen-Katic model [12, 24] assumes a top-hat velocity profile, linear wake growth and a square superposition of wake velocity deficits.

On the other hand, top-down representations do not model detailed array velocity distributions but instead consider large scale interactions of the wind farm with the

ABL. When the ABL encounters a wind farm, a new developing boundary layer forms where the wind farms can be thought of as a new roughness encountered by the ABL flow. Many of these models are based on the notion that in the 'fully developed' region of a large wind farm, the boundary layer is no longer growing and the wind farm is in balance with the ABL. This concept was introduced by Templin in 1974 [25] and a complete model was proposed by Frandsen [26], where they specified the last rows (latter portions) of the farm as the fully developed region, where the farm causes a modification of the mean velocity profile in the ABL. A number of subsequent models have built upon the Frandsen top-down model. Such models incorporated a more detailed description of the mean velocity profile in the turbine array boundary layer [11], while the theory of developing boundary layers [27] enabled the model to be applied in both fully developed and developing regions of the wind farm. Top-down models have also been extended to include various atmospheric stratification conditions [28, 29] and used in larger atmospheric simulations [30].

In general, wake models are applied to predict the power of individual turbines and wake behavior in the farm, while top-down models provide an average view that gives information on the physics of the ABL flow and the large-scale behaviors that begin to affect large wind farms (deep array effects). Coupled models that seek to leverage the relative strengths of both approaches have been shown to improve the accuracy of the hub height velocity estimates and total wind farm power output predictions over either type of model in isolation. The Frandsen model [31] represents a precursor to this approach in that it linked the top-down model in the fully developed region to a wake model at the front of a farm through an intermediate region. While this model recognized the benefits of both wake and top-down models for different wind farm flow regimes, the two models were not integrated and the information exchange between the regions was limited to the transition from one to the other over the intermediate region.

The Coupled Wake Boundary Layer (CWBL) model proposed by Stevens et al. [32, 33] demonstrated that two-way coupling of a wake and top-down model led to further improvements in power output predictions. This model and subsequent extensions showed excellent agreement with LES of operational wind farms [33]. As with the Fransden [31] model this CWBL required *a priori* identification of developing and fully developed regions of the farm. In addition, the CWBL model [33] assumed a regular rectangular array of wind turbines since regular streamwise (s_x) and spanwise (s_y) spacings needed to be prescribed, precluding its direct application to wind farms with arbitrary turbine layouts.

Many wind farm sites have topographical features that dictate irregular wind turbine arrangements, which complicates the a-priori identification of fully developed regions and further motivates the development of models that can be applied to a wide-range of turbine configurations. In Shapiro et al. [22], another coupled model was proposed that built on the ideas of the CWBL model. This model took important steps toward full generalization to arbitrary layouts by localizing the top-down portion of the model to each turbine instead of averaging over an entire region of the wind farm. A local top-down model for the developing boundary layer [27] then provides a description of the growth of internal boundary layers for each turbine. This method enables the model to be applied in both the developing and fully developed region of the farm and eliminates the need for an empirical specification of the evolution of the wake growth parameter as a function of distance into the farm. The Shapiro et al. model also improved upon the CWBL implementation through enhancements of the wake model including a super Gaussian velocity deficit (wake) profile that approximates a top-hat profile near the turbine, and then transitions smoothly towards a Gaussian profile [13, 22, 34] further downstream. Linear superposition was used to account for wake interactions.

Coupled models have shown great promise in leveraging multiple approaches to

wind farm modeling to incorporate more of the physics into the modeling and have the potential to create models with less parameterization. This approach is thus worthwhile investigating both for use in modeling irregular wind turbine arrangements and for use in learning more about wind farm dynamics in general.

1.1.2 Control-Oriented Dynamic Modeling

In the previous section, we discussed models that focus on the in-depth physics of the wind farm and attempt to capture key aspects of the mechanics of the wind turbine interactions. Recent work has made significant strides in refining these models in combination with a variety of wake superposition approaches to generate accurate estimates of the total farm power output over a range of turbine layouts for fixed wind directions, see e.g., [20, 22, 23, 32]. Changes in total wind farm power output for different wind directions have also been characterized in terms of fixed wind inlet angles to a given wind farm [35]. However, another aspect to consider in the study of the behavior of the wakes in a wind farm is the dynamic response of a wind farm to disturbances such as wind direction or dynamic yaw changes. Few studies address the dynamic behavior of the farm as the wind direction changes, although accounting for the effect of these changes has been shown to improve power output estimates [36] and produce more effective wake steering control versus approaches that assumed a static wind direction [37, 38].

Prevailing methods typically account for small changes in wind direction as an uncertainty that is included in simulations or models by taking a weighted average of the results over a range of wind inlet angles surrounding a desired value. Prior work has demonstrated that averaging the results of Reynolds-averaged Navier-Stokes (RANS) wind farm simulations in this manner improves predictions of the velocity deficit in several wind farms when compared to wind farm experimental observations [39]. This approach also improves the agreement between wake model power output

predictions and field data [40].

Taking the further step of accounting for dynamically changing wind direction or sweeps over a range of wind directions is challenging and computationally intensive. Previous approaches include incorporating the directional changes within a high-fidelity precursor simulation that then generates inflow conditions for a second LES of the wind farm [36, 41] as well as simulations with a dynamically changing reference frame [42]. Other work employs a nested simulation framework, where weather phenomena are modeled in the larger domains, and the wind farm is located in the smallest domain [43]. These detailed studies have provided a greater understanding of the phenomena and highlighted the difficulties of modeling the associated dynamic changes in wind farm power. For example, the LES studies of Munters et al. [36] show that a wind farm can experience a sharper drop in power output during a dynamic wind direction sweep than would be predicted through a series of static simulations at each different but constant wind inlet direction. This unforeseen reduction in power output can have a number of impacts in terms of forecasting the power available from the wind farm for the power grid. Control approaches require capturing the effect of these dynamics in real-time, which limits the applicability of the computationally expensive prior approaches based on LES or nested models.

In terms of yaw, while static wind turbine yaw has been studied extensively, resulting in many models for static yaw modeling, both in two dimensions and three dimensions [44–50], there are few models that account for time-dependent changes in yaw. The formulation outlined in [44] can represent dynamic changes as it is based on a partial differential equation (PDE), however the dynamic formulation has not been validated. The FLOw Redirection and Induction in Steady State (FLORIS) model from National Renewable Energy Laboratory (NREL) has been adjusted to take dynamic changes into account between turbines by implementing time delays in the travel of information between turbines [51]. A similar approach is also employed

by [52] along with an ensemble Kalman filter to incorporate measurements. Although the time scale of yaw changes it usually slow compared to pitch changes, accurate representations of how dynamic yaw changes propagate are important for accurate wind farm power predictions.

In order to characterize the complete capability of wind, both a deep physical understanding of wind farm interactions and an understanding of the transient response of the wind farm to changing conditions are needed.

1.2 Error Correction and State Estimation in Wake Modeling

Though great progress has been made in modeling the wake interactions in wind farms, both in static and dynamic situations, all reduced order models must leave out details of the flow in order to increase the efficiency of the model. These details can be important, particularly when designing controllers for wind farms. It has been shown that even as little as a 5% error in, for example, the tip speed ratio, which is the ratio between the tangential speed of the tip of the turbine blade and the freestream incoming wind velocity, of a turbine can result in energy loss of 1-3%, which can contribute to significant energy losses over a year of energy production [53]. In light of this, work is also being done on how to incorporate measurements taken from a wind farm to increase the accuracy of the reduced order models.

Wind farms collect an array of data while operating, some of which is called SCADA (Supervisory Control And Data Acquisition) data. Each turbine collects measurements of the rotor speed and the power output of the turbine. Additionally, each turbine typically has an anemometer to measure the wind speed and a wind vane to measure wind direction. These instruments usually sit on top of the nacelle of the turbine. While rotor speed and power measurements are typically accurate, the wind speed and wind direction measurements are prone to errors due to the disturbances

in the flow that the the rotor and nacelle structure create. Finally, meteorological masts measure wind conditions in specific locations, and lidar scans can give wind conditions over an area. [53, 54]

Many research studies have focused on how to use these measurements to improve wind farm predictions and supplement the wake models used in control. In pursuit of this, SCADA data and lidar scans have been used to reconstruct the flow conditions and estimate parameters over the farm in an operational wind farm [55–59], as well as used to improve the predictions made by wake models. One way of incorporating measurements is to filter the power measurements from the freestream turbines in the farm to find an inflow profile for the wake model [22]. This is an improvement over a uniform inflow since it can represent the varying wind conditions that can occur over wind farms that span multiple kilometers of space. This is a beneficial correction, but complex wake interactions in the farm can also necessitate more involved methods to update the reduced order models.

One of the most common updating methods using wind farm measurements is the Kalman filter, which updates the estimated states using an optimal gain [60]. When used in wind farm applications, there are a number of variations on the original Kalman filter formulation that are applied, such as including a Bayesian optimization scheme [61]. Other forms of the Kalman filter include the extended, unscented, and ensemble Kalman filters [56]. The ensemble Kalman filter, in particular, is used extensively in problems with larger state space variables, as would arise in large wind farms, since it approximates the error covariance matrix using ensembles of the model to reduce the necessary computations [62]. Ensemble Kalman filters have been shown to be successful in updating the parameters of wake models to more closely track the wind farm plant conditions [56, 63–65]. Kalman filtering, particularly ensemble Kalman filtering, is a useful tool to incorporate wind farm measurements into the wake model to update the model in real time and remains a popular topic of research

in this area.

1.3 Active Power Control in Wind Farms

The predominant goal of wind farm control is to increase the capacity factor for a wind farm. In addition, it is hoped to both increase power and decrease loading on individual turbines, when possible, and to better integrate wind farms into the electricity grid [66]. An important part of these goals is being able to provide ancillary services, which are the services necessary to ensure reliable transmission of power, to the grid. These services include things like scheduling and dispatch, reactive power and voltage control, operating reserves, and frequency regulation. In this thesis, we focus on frequency regulation, which involves using electricity producers to balance the grid frequency. Since wind farms are compensated for their power production, one of the focuses of active power control of wind farms is enabling the participation of wind farms in frequency regulation while minimizing the reduction in the bulk power of the wind farm. The control actuation of turbines occurs through changing the nature of the turbine wakes and how much power is extracted from the flow. Three ways that turbines are controlled are the thrust coefficient, which would be implemented using generator torque or blade pitch control in the field, yaw, which rotates the turbine in both directions around the tower of the wind turbine to affect the angle of the incoming flow, and tilt, which moves the turbine to tilt upwards or downwards, using the nacelle as the axis of rotation [67, 68]. In a dynamic framework, the thrust coefficient and turbine yaw are the two most common control strategies. These approaches have been implemented both separately and in tandem. Studies have been done examining the effect of tilt [68, 69], however it remains a sparser area of study. This lesser interest could be due to the structural considerations involved with tilt, which could include considering the tower of the wind turbine when determining the direction of the tilt angle and needing the turbine to be able to withstand a relatively large angle of tilt

for years, as noted in [70].

The following studies employ the thrust coefficient of a wind turbine, which modulates how much power the turbine is extracting from the flow at any given moment, as the control variable in active power control. There have been studies focusing on a power tracking using a distributed optimization framework in the [51, 71], others that additionally attempt to minimize the loads on the turbines in the farm [72] or maximize the available power while following a signal [73]. However, all of these studies use a simplified model for the wind farm, which could reduce their effectiveness. Studies have also been done using LES as the wind plant, such as in [74], where model predictive control (MPC) was used to implement a row-based controller on an LES wind farm plant. The control was effective, but dependent on the turbines being arranged in a regular array. Another study uses LES as the model in MPC, which enables full knowledge of the flow to use pitch control to maximize the power output [75]. Alternatively, PI controllers have also been designed to work on a wind-farm level, however, in some studies the farm was severely derated in order to provide enough power for power tracking [76].

Yaw control, or wake steering, has also been used in wind farm control. Due to the slower timescale of yawing, i.e., the fact that the turbine can only be yawed at a finite speed, and yawing causes wear on the material components of the wind turbine, most work has been done on static yaw optimization. In these studies, it has been shown that yawing can increase the overall output of the wind farm, both in simulations [77, 78], in experiments [79, 80], and more recently in field tests [47, 81, 82]. In a dynamic setting, yaw control has mostly been used for power maximization rather than power tracking. It has been implemented using a model free approach, such as in [83], in a wind tunnel test. Other approaches include using a heuristic wake model or LES to generate set points for the yaw controller [82]. Steps have also been taken to implement a more dynamic wake model in yaw control, which takes into

account the delay between turbines [52]. While much work has been done in adjacent areas, dynamic yaw is not typically applied to power tracking on its own due to its slower timescale. It was applied in one study using a linear model derived from data and performed well. However, since the model is derived from a specific data set, application of the model to operating conditions outside of the initial data set could reduce the effectiveness of the control [84].

In addition to being applied separately, yaw and pitch control have also been combined in several applications. Here, we will focus specifically on studies where LES was used as the wind farm plant, since these studies provide more realistic wind farm conditions than less detailed wind farm models. The study in [85] is a successor to the pitch power maximization study [75] and uses LES as the model in model-based optimal control. By using adjoint equations to find the gradients, this study was able to use both yaw and the thrust coefficient to dynamically maximize the power output, finding gains of $25\% - 34\%$ in the power output over a control period of 30 minutes. This paper is an excellent proof of concept for the potential of combined yaw and pitch control. However, currently, LES is too computationally expensive and time consuming to use for real-time control applications. Both control methods were also used in [66] for power tracking, where the yaw control was only activated if there was not enough power in the farm to meet the power reference signal. This study used an inner control loop for pitch and an outer control loop for the yaw, both based on MPC. However, the yaw optimization was performed only every fifteen minutes and only considered static yaw configurations. Using both yaw and thrust control together remains a challenging and open area or research that needs further study.

1.4 Outline

This thesis works towards achieving a better understanding of wind farm physics in order to attain more accurate power prediction as well as towards developing a

dynamic control-oriented model capable of providing insight into the dynamic response of wind farms to changing conditions. Furthermore, it works towards including active yaw control in power tracking using yaw and pitch actuation. The subsequent chapters include (1) the development and analysis of a physics-based coupled model for arbitrary wind farm geometries, (2) building a control-oriented model based on graph theory to capture the dynamic response of the wind farm, (3) incorporating measurements available to operating wind farms to correct modeling errors, and (4) developing a pitch and yaw controller that provides power tracking of a power reference signal. The rest of the thesis is organized as follows.

In Chapter 2, we discuss the physics-based coupled model and validate it on several nonuniform wind farm configurations. In Chapter 3, we present the control-oriented graph based model. Chapter 4 covers the incorporation of wind farm measurements into the wake models for error correction. In Chapter 5, a yaw and pitch controller is presented and applied to an LES wind farm plant. Finally, conclusions and future work are presented in Chapter 6.

Chapter 2

Wake Model Coupling: The Area Localized Model

In this chapter, we present a coupled, physics-based model, the Area Localized Coupled (ALC) model, that seeks to improve our understanding of the wake interactions in the wind farm and to predict the power output of the wind farm. The approach taken also further advances the goal of increasing model applicability to a wider range of wind farm geometries and inflow conditions.

The localized approach employed in the ALC model and the model described in Ref. [22] have the added advantage of providing significantly more information about the flow field than conventional wake models, top-down models, and the original CWBL implementation [33]. For example, these prior approaches do not typically furnish information about the velocity field such as turbulence conditions (friction velocities above and below turbine layer) within each cell. They instead provide only mean velocities or use empirical correlations for turbulence intensities, which make it more difficult to quantify loading characteristics for the turbines.

In this work, the ALC model is applied to two wind farms and its predictions are compared to LES. The first wind farm is arranged in a circular configuration that provides an example of a non-rectangular array with streamwise and spanwise spacings between turbines that vary throughout the farm for any given wind direction. We

provide a detailed analysis of the velocity predictions for each turbine in this circular farm over a range of wind inflow directions to illustrate the use of the local turbine area in the computations and to evaluate the accuracy of the predictions of the model at both the turbine and farm level. A second wind farm consisting of two regions, one with regular and the other with random turbine placements, is also considered over a smaller range of wind directions. The results illustrate the benefits of the area localized approach in coupling physics-based models for both the turbine wakes and the ABL and to provide good predictions of the local velocity field and power produced by individual turbines as well as the total wind farm production.

2.1 Area Localized Coupled (ALC) Model

The ALC model couples a wake model with a super-Gaussian wake profile and linear wake superposition, with a top-down model for a developing wind farm. Each turbine has its own instance of the top-down model that is localized to its own "area" and described by the local planform thrust coefficient. The planform thrust coefficient represents the momentum extracted from the flow in a developing internal boundary layer upstream of each of the turbines by accounting for their axial flow resistance (the localized internal boundary layer concept distinguishes the ALC model from the [22] approach). The coupling is imposed through a minimization of the difference between the area average velocities computed from the two models, see Fig. 2-1. This matching condition is used to determine the wake expansion coefficients for each local turbine cell. The following subsections summarize the wake model and the top-down model that are coupled in the ALC model, see Fig. 2-1.

2.1.1 Wake Model

The aim of the wake model is to calculate the streamwise velocity at the hub height of each turbine in the wind farm. The wake model used here is developed in [22]. When

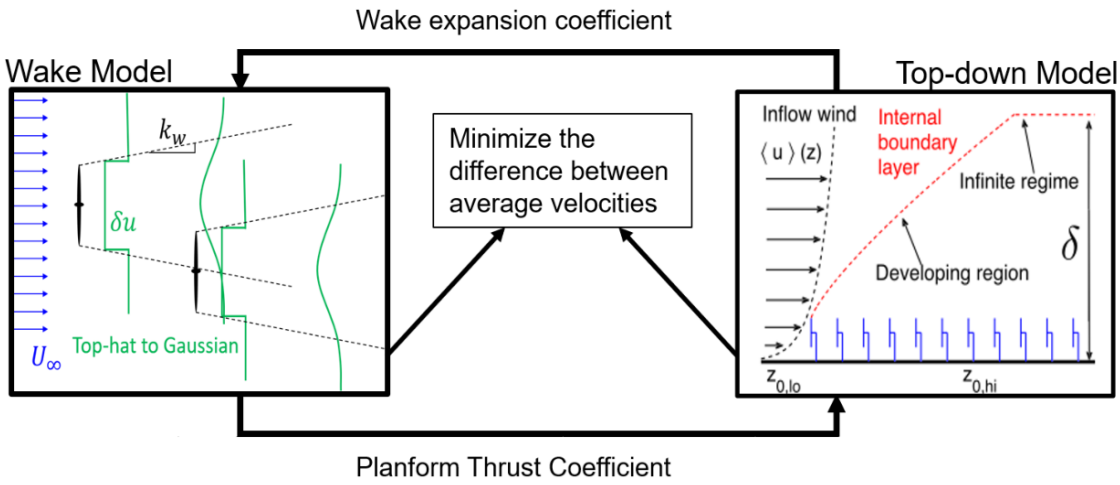

Figure 2-1. Schematic diagram illustrating the ALC model, where the wake model (left, top view) and top-down model (right, side view) are connected with the wake expansion coefficient k_w and the planform thrust coefficient c_{ft} (describing the momentum loss per unit horizontal area). δu is the wake velocity deficit, U_∞ the inlet velocity at hub-height, $\langle u \rangle(z)$ is the mean velocity profile as function of height z from the boundary layer (top-down) description, δ is the height of the ABL, $z_{0,lo}$ is the ground surface roughness length while $z_{0,hi}$ represents the effective surface roughness height including effects of the wind farm.

evaluated at turbine hub locations, the velocity can be used to predict the power generated by each turbine. This flow field entering the turbine rotor area is described as a superposition of individual wakes by the equation

$$u(x, y, z) = U_\infty(y, z) - \sum_n \delta u_n(x)\, W_n(x', r'). \tag{2.1}$$

where x is the streamwise coordinate (in the direction of the incoming freestream wind), $x' = x - s_{n,x}$ is the coordinate relative to the position of the turbine, $s_{n,x}$, $r' = [(y - y_n)^2 + (z - z_n)^2]^{1/2}$ is the radial distance from the center position of the n^{th} turbine. $U_\infty(y, z)$ is the incoming freestream mean velocity which can vary across the inflow plane (y is the horizontal coordinate transverse to the incoming wind), and in general also as function of vertical coordinate z. In the wake model used in this paper we will neglect the variations of $U_\infty(y, z)$ in the vertical direction for simplicity and set $U_\infty(y) = U_\infty(y, z_h)$, where z_h is the hub-height. $\delta u_n(x)$ is the deficit velocity

of the n^{th} turbine at downstream position x, and $W_n(x,r)$ is the shape function of wake n. The velocity deficit $\delta u_n(x)$ is calculated as the steady-state solution to a one-dimensional partial differential equation that describes the more generally valid time-dependent behavior of the wake:

$$\frac{\partial \delta u_n}{\partial t} + U_\infty \frac{\partial \delta u_n}{\partial x} = -\frac{2U}{(d_{w,n}(x))^2}\frac{d(d_{w,n})}{dx}\delta u_n(x,t) + [U_\infty \delta u_{0,n}]\left[\frac{1}{\Delta\sqrt{2\pi}}\exp\left(\frac{-x^2}{2\Delta^2}\right)\right].\infty \tag{2.2}$$

The equation describes the rate of change of δu_n when moving downstream at speed U_∞ in response to two effects: (1) the transverse diffusion expressed as a wake expansion term (first term on the equation's right-hand-side) leading to a decrease of δu_n, and (2) the creation of a wake deficit at and near the turbine locations due to the axial induction (second term). Under the simplifying assumption of steady state flow conditions, the solution to Eq. 2.2 has the form

$$\delta u_n(x) = \frac{\delta u_{0,n}}{[d_{w,n}(x)]^2}\ \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\Delta\sqrt{2}}\right)\right], \tag{2.3}$$

where $\delta u_{0,n}$ is the initial velocity deficit at the turbine location, $d_{w,n}(x)$ is the wake expansion function (the normalized location-dependent wake diameter), and Δ is the characteristic width of the Gaussian function used to smooth the forcing in Eq. 2.2 along the streamwise direction. The wake expansion function is assumed to be of the following form

$$d_{w,n}(x) = 1 + k_{w,n}\ln(1 + e^{x/R}), \tag{2.4}$$

which at large x/R, where R is the radius of the turbine, tends to the classical linear growth $d_{w,n}(x) = k_{w,n}(x/R)$ but smoothly merges to $d_{w,n}(x) = 1$ in the near-turbine region thus preventing values that are not physical. The parameter $k_{w,n}$ is the turbine specific wake expansion coefficient for turbine n that determines the rate at which the wake expands as it travels downstream. The expansion of each individual turbine wake

depends on local flow conditions. For instance, one would expect it should increase with increasing local turbulence intensity. In the ALC model, the coefficient will be calculated with an expression of the form

$$k_{w,n} = \alpha \frac{u_{*,n}}{u_{\infty,n}}. \tag{2.5}$$

where $u_{\infty,n}$ is the y-dependent upstream velocity at each turbine and $u_{*,n}$ is related to the local friction velocit. The precise form for the local friction velocity to be used in the model is derived from the top-down model as described in §2.1.2. The coefficient α is a model parameter that will be determined through the coupling of the wake and top-down models.

We incorporate effects of spanwise variations in the inflow velocity by determining the "upstream" velocity $u_{\infty,n}$ of each turbine. We first define the turbines that are not in the wake of others for a given wind direction as "freestream turbines", to which we ascribe a y dependent freestream velocity $u_{\infty,n} = U_\infty(y_n)$. Here $U_\infty(y_n)$ denotes the average velocity across the turbine disk based on the incoming flow field. Then, the nonuniform velocity inflow profile is propagated through the farm to the waked turbines. In order to find the disk velocity, and thus the power, of the waked turbines, we need to determine the the disk velocity that would have existed at wind turbine n without the turbine there. This depends on the wakes of the upstream turbines, which are accounted for through wake superposition. Wake superposition represents the combined effect of wakes interacting with each other in the flow. To calculate $u_{\infty,n}$, i.e., the upstream velocity for the n^{th} turbine, we use the following expression:

$$u_{\infty,n} = \int_0^R \left[U_\infty(y) - \sum_{m \neq n} \delta u_m(s_{n,x}) W_m(s_{n,x} - s_{m,x}, r') \right] 2\pi r' dr', \tag{2.6}$$

where $s_{n,x}$ is, again, the position of the turbine in the streamwise direction and $r' = [(y - y_n)^2 + (z - z_n)^2]^{1/2}$ is the radial coordinate of point (y, z) for turbine n with rotor center at (y_n, z_n). The superposition sum is then performed over all turbines

upstream of turbine n. The nonzero terms in the sum are determined by the wake shape function W_n, which takes the form of a super-Gaussian

$$W(x,r) = C(x)\exp\left(-2\left(\frac{2r}{D\ d_w(x)}\right)^{p(x)}\right) \tag{2.7}$$

where $r^2 = y^2 + z^2$. This function smoothly transitions from a top-hat wake profile immediately following the turbine into a Gaussian wake profile downstream. The following functional form for $p(x)$ was proposed to reproduce the shape and rate of transition from a top-hat to a Gaussian seen in experimental wake data:

$$p(x) = 2\left(1 + \frac{D}{\max(x,0)}\right). \tag{2.8}$$

The function $C(x)$ can then be shown [22] to have the following form, assuming mass (or linearized momentum) conservation

$$C(x) = \frac{p(x)}{2\Gamma\left(2/p(x)\right)} 2^{(2/p(x))}, \tag{2.9}$$

where Γ is the gamma function. Finally, the initial velocity deficit is obtained from an inviscid model and has the traditional form based on actuator disk momentum theory [22, 67]:

$$\delta u_{0,n} = u_{\infty,n}\left(1 - \sqrt{1 - C_{T,n}}\right) = \frac{1}{2} C'_{T,n}\, u_{d,n}, \tag{2.10}$$

where $C_{T,n}$ is the coefficient of thrust for the turbine, which determines the amount of thrust the turbine extracts from the flow. $C'_{T,n}$ is the local coefficient of thrust and is related to the coefficient of thrust through the relation $C_{T,n}U^2_{\infty,n} = C'_{T,n}u^2_{d,n}$, where $u_{d,n}$ is the average velocity over the turbine disk area, which is defined later in Eq. 2.23.

In the present work, we use the stead state solution shown in Eq. 2.3 to the PDE shown in Eq. 2.2, and we do not take into account the time-dependence. However, if $\delta u_n(x,t)$ were to be determined from the time-dependent PDE, the ALC model can be readily extended to time-varying applications.

Returning to the steady state case, the wake model is now complete as long as a value for the parameter α can be specified unambiguously. This is accomplished through matching with the top-down model described in the next section.

2.1.2 Top-Down Model

As previously discussed, the top-down model [11, 31] regards the wind farm as a homogenized surface roughness in the atmospheric boundary layer. Assuming a fully developed (horizontally homogeneous) wind turbine array boundary layer (WTABL) [11] the velocity profile can be shown to consist of two constant shear stress regions, one below the turbines and one above the turbines. Each layer has friction velocities, $u_{*,lo}$ and $u_{*,hi}$, and roughness heights, $z_{0,lo}$ and $z_{0,hi}$, respectively. The model connects the two regions through the wind turbine layer, which is defined as $z_h - R \leq z \leq z_h + R$ where R is the radius of the turbines [11]. A momentum balance results in the following relationship between the two friction velocities and the planar average velocity $\bar{u}_h$ at hub-height z_h of the turbines:

$$u_{*,hi}^2 = u_{*,lo}^2 + \frac{1}{2} c_{ft} \bar{u}_h^2, \tag{2.11}$$

where c_{ft} is the planform thrust coefficient that represents the momentum extracted from the flow by the turbines' axial flow resistance per unit horizonatal area. This coefficient is obtained from the ALC model coupling, as described in §2.1.3.

In the wind turbine layer, the turbulent flow is assumed to include the effects of an additional eddy viscosity ν_{w*}. The total eddy viscosity in this region is defined in terms of the friction velocities in the top and bottom halves of the turbine region. In the lower region, where $z_h - R \leq z \leq z_h$, the eddy viscosity is represented as $(1+\nu_{w*})\kappa u_{*,lo}$, and in the upper region, $z_h \leq z \leq z_h + R$, it is represented as $(1+\nu_{w*})\kappa u_{*,hi}$. The extra eddy viscosity needs to be modeled and it was assumed to have the form $\nu_{w*} = 28\sqrt{\frac{1}{2}c_{ft}}$ in [11]. By using the known value of the lower surface roughness height $z_{0,lo}$, and the

momentum balance (Eq. 2.11), and enforcing velocity continuity across the vertical profile, one may derive the roughness height due to the wind farm as [11]:

$$z_{0,hi} = z_h \left(1 + \frac{R}{z_h}\right)^{\beta} \exp\left(-\left[\frac{c_{ft}}{2\kappa^2} + \zeta^{-2}\right]^{-1/2}\right), \tag{2.12}$$

where $\zeta = \ln\left[(z_h/z_{0,lo})\left(1 - R/z_h\right)^{\beta}\right]$ and $\beta = \nu_{w*}/(1+\nu_{w*})$. Using this result, we can also calculate the friction velocities in both of the layers:

$$u_{*,hi} = u_* \frac{\ln(\delta/z_{0,lo})}{\ln(\delta/z_{0,hi})}, \tag{2.13}$$

$$u_{*,lo} = u_{*,hi} \frac{\ln\left(\frac{z_h}{z_{0,hi}}\left(1 + \frac{R}{z_h}\right)^{\beta}\right)}{\ln\left(\frac{z_h}{z_{0,lo}}\left(1 + \frac{R}{z_h}\right)^{\beta}\right)}, \tag{2.14}$$

where u_* is the friction velocity of the incoming ABL and δ is the overall ABL height.

Note that the top-down model is based on the planform-averaged momentum equation and therefore requires averaging over extended horizontal areas. In order to apply the model in a more localized fashion, horizontal areas associated to each of the turbines must be defined. We define these areas using Voronoi tessellation (also used in the model in [22]), which naturally associates cells and their areas to each turbine n. In particular, each turbine is a node, and the vertices are defined as points that are equidistant from three separate nodes. Figure 2-2 shows the local areas resulting from a Voronoi tessellation procedure applied to several different configurations of turbines: a regular array, a staggered array, a circular configuration, and an entirely random distribution. The edge cells are defined by using ghost points projected outside of the array [22].

The Voronoi cells separate the wind farm into planform areas belonging to each turbine in a way that can be generalized to any wind farm layout. The ALC model exploits this local turbine area definition to compute friction velocities and roughness heights individually in each Voronoi cell, which enables the model to provide information on flow conditions in the areas around each turbine. Information related to the

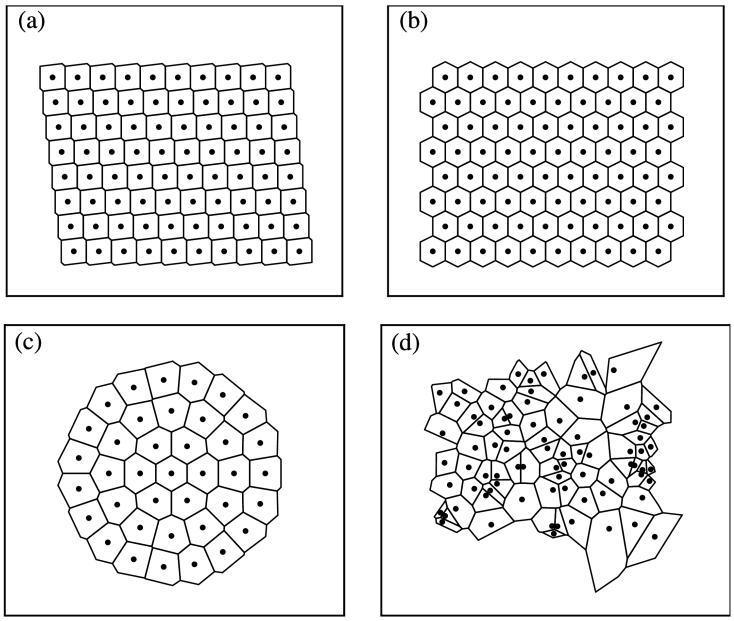

Figure 2-2. Voronoi cells drawn for different farm configurations including (a) a uniform farm, (b) a staggered farm, (c) a circular farm and (d) a random farm

turbulence of the local flow field is then used for each turbine and cell to determine appropriate wake expansion coefficients for the wake model.

In order to compute the local friction velocities, we use the notion of a developing internal boundary layer (IBL) over a wind farm [27, 86] that depends on the streamwise position x from the start of the farm. The IBL height is modeled according to:

$$\delta_{ibl}(x) = \min\left[z_h + z_{0,hi}\left(\frac{x - x_0}{z_{0,hi}}\right)^{4/5},\ \delta\right], \tag{2.15}$$

where $x - x_0$ represents the distance to the beginning of the IBL, i.e., the distance to the start of the farm directly upstream to turbine n and δ is the height of the overall ABL. This implies that the friction velocities evolve as a function of x through the farm, until δ_{ibl} reaches the final boundary layer height δ in the fully-developed region. Analysis of the developing wind farm internal boundary layer [27, 87] shows that the friction velocity evolves similarly to that of a boundary layer flow over a surface with a smooth-to-rough transition, i.e., the friction velocity increases sharply

at the transition (the front of the wind farm) but then gradually decreases again as the IBL grows and the mean velocity gradient decreases. The analysis in [27] shows that the position-dependent friction velocity $u_{*,hi}(x)$ can be obtained from Eq. 2.13 by replacing the overall δ by $\delta_{ibl}(x)$ for turbine n. That is to say, we will evaluate $u_{*,hi,n}$ for each individual turbine by replacing δ in Eq. 2.13 by $\delta_{ibl}(s_{n,x})$ evaluated at the position of turbine n.

When dealing with an irregular wind farm, special care must be taken to properly define the starting position of the internal boundary layer. Here we choose to define the IBL start to be located at the freestream turbines, i.e., the turbines in the farm that only see the upstream incoming velocity and are not affected by the wakes of other turbines. The process to identify freestream turbines for a given wind inlet direction is described in the next section. Figure 2-3 shows how we model this as a boundary layer initialization, defined for the case of a circular wind farm. In this case the start ("trip line") of the IBL is defined at the first set of turbines to see the freestream velocity. Let $x_{n,0}$ be the x-location on the trip-line directly upstream of turbine n. The streamwise distance between any given turbine and the boundary layer "trip line" for turbine n is thus given by $s_{n,x} - x_{n,0}$, shown with the blue arrow for a sample turbine in the farm.

In the localized top-down model, calculations can now be performed for each turbine in its own turbine cell, enabling the localization needed to represent both the spatial non-uniformities due to arbitrary turbine placements as well as nonuniform (y-dependent) velocity inflow. In the case of the freestream turbines, the freestream values of the flow are used. For the remaining turbines, we use the developing boundary layer framework. For a standard boundary layer, the friction velocity of a boundary layer flow over a rough surface with roughness $z_{0,lo}$ and inflow velocity U_∞ (mean velocity at hub-height $z = z_h$ upstream of the wind farm) would be given by $u_* = U_\infty \kappa / \ln(z_h/z_{0,lo})$. If the inflow is y-dependent, and if the freestream

inflow velocity corresponding to turbine n is denoted as $U_{\infty,n}$ (it can be computed by averaging the inflow over the cell), we can write $u_{*,n} = U_{\infty,n}\kappa/\ln(z_h/z_{0,lo})$. Replacing in Eq. 2.13 to obtain the actual friction velocity for turbine n, we can write

$$u_{*,hi,n} = U_{\infty,n}\frac{\kappa}{\ln(z_h/z_{0,lo})}\frac{\ln(\delta_{ibl}(x_n)/z_{0,lo})}{\ln(\delta_{ibl}(x_n)/z_{0,hi})}. \tag{2.16}$$

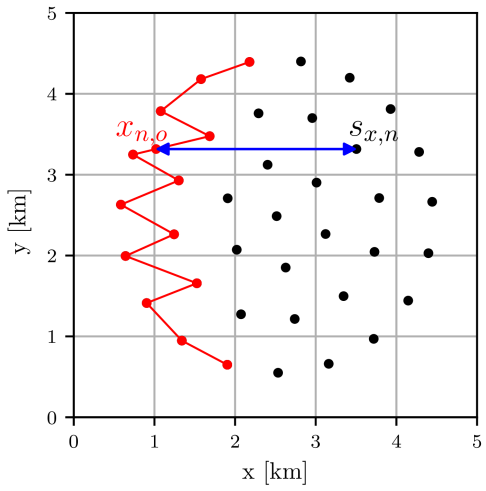

Figure 2-3. Definition of the start of the boundary layer (IBL "trip-line" in red) for the top-down model starting at the freestream turbines for a wind direction flowing from left to right in the figure. The blue arrow shows the distance between the trip line and a sample (the n-th) turbine in the farm. This distance is used to determine the IBL height for the n-th turbine according to Eq. 2.15.

Once the friction velocity is known, the top-down model provides a prediction for the mean velocity at hub height [11]. Once the model is applied individually to each cell, the mean velocity is given by

$$\bar{u}^{td}_{h,n} = \frac{u_{*,hi,n}}{\kappa}\ln\left(\frac{z_0}{z_{0,hi}}\left(1+\frac{R}{z_h}\right)^{\beta}\right). \tag{2.17}$$

Note that in order to evaluate $\bar{u}^{td}_{h,n}$ we require $z_{0,hi}$, which according to Eq. 2.12 depends on the planform thrust coefficient c_{ft}. This value can differ from turbine to turbine since the total turbine forces and momentum exchanges affecting the development of the local internal boundary layer may differ across the wind farm. In order to determine $c_{ft,n}$ for each individual turbine n, information from the wake

model (§2.1.1) is required as described in the next section §2.1.3 where the coupling of the constituent models is presented.

2.1.3 Coupling of wake and top-down models: ALC Model

The wake model and the top-down model are coupled by comparing their respective predictions of the average planar velocity values in each waked Voronoi cell. In the top-down model, these are the velocities $\bar{u}_{h,n}^{td}$ calculated in each Voronoi cell according to Eq. 2.17. The output of the wake model is a velocity field $u(x, y, z_h)$ from Eq. 2.1. In order to compare the wake model to the top-down model predictions, a cell-averaged wake model velocity is defined by averaging the velocity field predicted by Eq. 2.1 at $z = z_h$ over each cell for each turbine n. We denote this cell averaged wake model velocity as $\bar{u}_{h,n}^{wm}$. The steps are illustrated in Fig. 2-4. The average planar velocities in each cell are illustrated in Fig. 2-4(c).

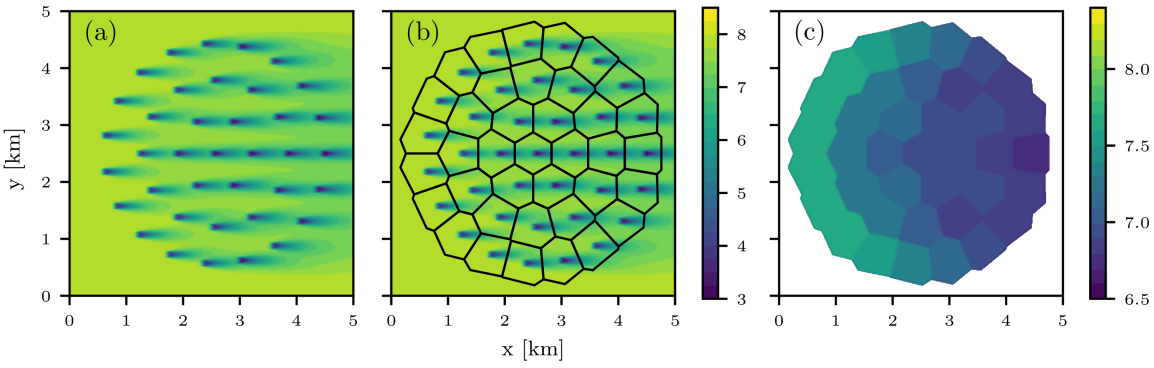

Figure 2-4. The planar velocity from the wake model is calculated by taking (a) the velocity field given by the wake model and considering the planar average in each Voronoi cell, shown in (b). The average velocity from each cell is shown in (c). (Note the use of a different color scale.)

Ideally one would want the velocities predicted by the top-down and wake methods to yield the same cell averaged velocities, for each turbine. Free model parameters are adjusted to minimize the least-square error between both predictions from the two approaches. Specifically, recall that the parameter α required to specify the wake

expansion parameter k_w in Eq. 2.5 was left unspecified. We now follow the basic idea of the CWBL approach [32] to determine the wake expansion coefficient by iterating on parameter values until the velocities at hub height predicted by each model match. Since in the ALC model has many cells, we find the α that minimizes the square difference between the average planar velocities from each model over the cells of turbines that are in the wake of upstream turbines (called "waked cells"), according to

$$\min_{\alpha} \sum_{n \in wake} \left(\bar{u}_{h,n}^{wm} - \bar{u}_{h,n}^{td} \right)^2 . \tag{2.18}$$

where $wake$ defines the set of waked cells. The summation does not include free-stream turbines because for those turbines the top-down and wake models would use the same value of k_w which is unaffected by wakes and expected to be different from those inside the farm. The free-stream turbines are found by setting a threshold on the velocity deficit of 1%. Specifically, turbine n is denoted as a freestream turbine if

$$1 - u_{\infty,n}/U_{\infty,n} < 0.01, \tag{2.19}$$

where $U_{\infty,n}$ is the average freestream velocity over the disk for the n^{th} turbine according to the inflow profile and $u_{\infty,n}$ is the inflow to the n^{th} turbine defined by Eq. 2.6 from the wake model, which takes into account the wakes of upstream turbines. The procedure is iterative and begins by treating only turbines that fall along the convex hull of the farm as free-stream turbines. After a first iteration (see below), $u_{\infty,n}$ can be determined and additional freestream turbines can be identified (those complying with the condition in Eq. 2.19) that may be located further inside the farm and yet be unaffected by upstream turbines. The minimization is carried out using the scipy.optimize.minimize function for unconstrained minimization. It uses the default BFGS optimization method with bounds of $0 \leq \alpha \leq 100$.

For the waked turbines, we compute a turbine-specific wake expansion coefficient $k_{w,n}$, to be used in Eq. 2.4, in terms of the arithmetic mean of the high and low

cell-specific friction velocities (found from Eq. 2.16 and Eq. 2.14), according to

$$k_{w,n} = \alpha \frac{1}{2} \frac{u_{*,hi,n} + u_{*,lo,n}}{\bar{u}^{td}_{h,n}}. \tag{2.20}$$

The denominator ($\bar{u}^{td}_{h,n}$) represents the advection velocity for which we use the average hub height velocity in the cell from the top-down model.

For the freestream turbines, we use the same expression but since they are subjected to the incoming freestream advection and friction velocities, their expansion coefficient is evaluated according to Eq. 2.5, or simply

$$k_{w,n} = \alpha \frac{\kappa}{\ln(z_h/z_{0,lo})} \tag{2.21}$$

and using the same α as for the entire wind farm.

Evaluation of the top-down model still requires specification of the planform thrust coefficient c_{ft} in Eq. 2.12, which in turn determines the evolution of $\delta_{ibl}(x)$ as well as the top-down mean velocity from Eq. 2.17. The planform thrust coefficient c_{ft} is defined as the total force per unit horizontal area. We argue that the relevant area affecting the turbulence at turbine n is the area of all the Voronoi cells upstream of the turbine, as illustrated in the sketch in Fig. 2-5. This is the region over which the IBL evolves until it reaches turbine n and we therefore consider the vertical momentum flux averaged over this entire "upstream" area to characterize the evolution of the boundary layer reaching turbine n.

The local planform area thrust coefficient is then given by

$$c_{ft,n} = \frac{\pi R^2 \sum\limits_{i \in line} C'_{T,i} u^2_{d,i}}{\sum\limits_{i \in line} A_{pf,i} (\bar{u}^{wm}_{h,i})^2}, \tag{2.22}$$

where $A_{pf,n}$ is the planar area of the Voronoi cell for the n^{th} turbine, and *line* refers to the cells that would be crossed if a line were drawn from the current cell to the front of the farm. This area is illustrated in Fig. 2-5, where the line is denoted in red and the cells included in the calculation are shaded. Moreover, $u_{d,i}$ is the wake-model

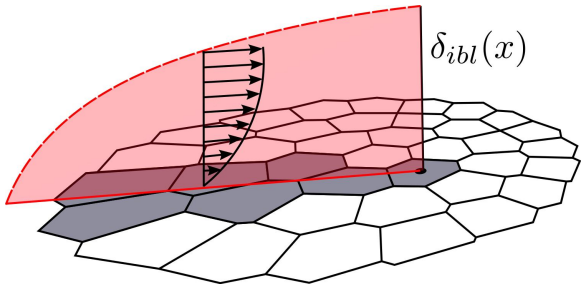

Figure 2-5. Illustration of how the cells are selected for inclusion in the calculation of the planform thrust coefficient, where the grey cells are included in the calculation for this particular turbine.

velocity field $u(x, y, z_h)$ (obtained from Eq. 2.1) averaged over the rotor disk area A_d for turbine i:

$$u_{d,i} = \frac{1}{\pi R^2}\left(1 - \frac{C_T'}{4 + C_T'}\right) \iint_{A_d} u(x = s_{i,x}, y, z)\, dydz. \tag{2.23}$$

The coefficient $c_{ft,n}$ can be thought of as the ratio of the total force applied by the turbines in the cells upstream including turbine n that affect the local boundary layer development, and the total horizontal momentum flux associated to the horizontally averaged mean velocity over the same planform area. The addition of the cells in front of the current cell is meant to represent the effect of the developing boundary layer over the farm and links the downstream cells to those upstream, since the turbines in the upstream cells influence those in their wakes. The coefficient $c_{ft,n}$ is then used to determine a local roughness height $z_{0,hi,n}$ that is then used in the evaluation of $\delta_{ibl}(x_n)$ and $u_{*,hi,n}$ according to Eqs. 2.15 and 2.16.

Figure 2-6 summarizes the coupling in the ALC model using an example of the planar average velocity view from each model. Using the developing boundary layer approach for the entire farm enables the model to find one α value that includes input from all the waked turbine cells. The top-down model provides the information required to determine the wake expansion coefficient in the wake model, while the wake model provides the planform thrust coefficient to be used in the top-down model.

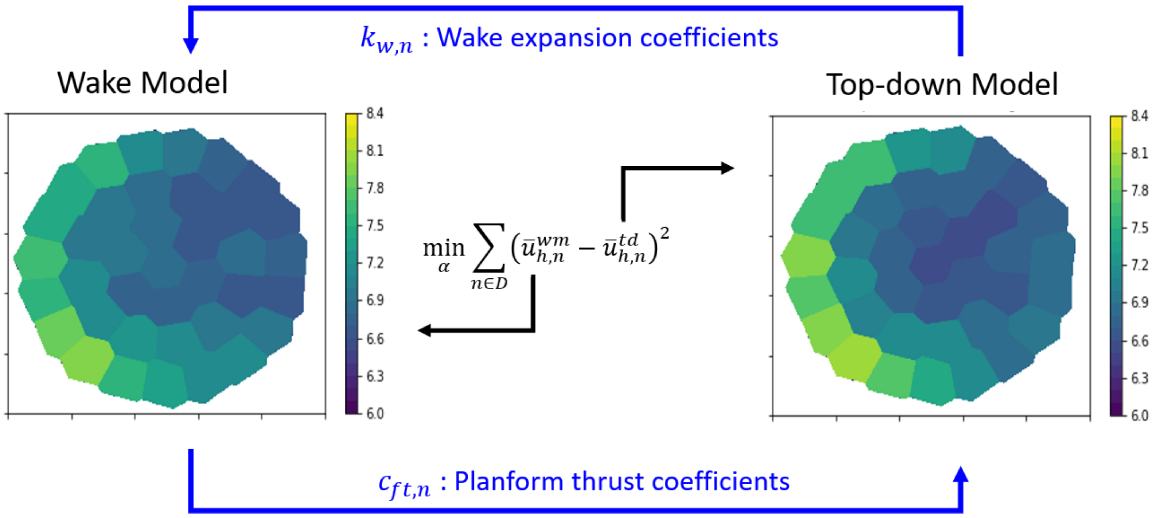

Figure 2-6. A summary of the ALC model that shows how the planar velocities between the two models are compared and the difference is minimized.

After the determination of α, the power at each turbine is found by using the velocity field produced by the wake model with the optimized wake expansion coefficients from the coupling. In order to compute the power from turbine n, we use

$$P_n = \frac{1}{2}\rho C'_P \pi R^2 u_{d,n}^3, \tag{2.24}$$

where ρ is the density of the air, C'_P is the local power coefficient and $u_{d,n}$ is the average disk velocity of the turbine calculated using Eq. 2.23.

From the model description it is apparent that the ALC model provides significantly more information about the flow than conventional wake models. The ALC model calculates friction velocities for each of the cells which gives local flow information regarding turbulence conditions (assuming that turbulent root-mean-square velocities are proportional to the friction velocity). This local information can be used in combination with operating conditions to predict other quantities such as the unsteady loading characteristics for each individual turbine. Additionally, each cell is associated to an individual wake expansion coefficient, even though only one α is selected for the

entire farm. Having individual k_w values for wakes emanating from each turbine can represent, for example, situations in which the expansion rate for turbines close to the entrance is lower than the wake expansion rate further downstream, where turbulence levels are enhanced.

The next two sections demonstrate the application of the ALC model to two wind farms that are not arranged in a rectangular lattice pattern. In §2.2.2 we consider a circular farm while in §2.2.3 we examine a wind farm that includes random turbine spacings to demonstrate the ability of the model to capture a range of farm geometries.

2.2 Model Validation with LES

2.2.1 LES of Wind Farms

Many of the cases provided in this thesis are validated using LES, so the following will provide a brief explanation of this simulation method. The flow through a wind farm, in the neutrally-buoyant case that exists in a neutrally stable ABL, is governed by the incompressible Navier-Stokes equations

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2.25}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} - \nu \frac{\partial^2 u_i}{\partial x_i \partial x_j} + f_i \tag{2.26}$$

where u_i is the i^{th} component of the velocity, ρ is the density of the air, p is the pressure, ν is the kinematic viscosity, and f_i is the i^{th} component of the force per unit mass applied on the flow from the turbines. These equations also employ Einstein's summation convention in regards to repeated indices. These equations can be simulated directly using Direct Numerical Simulation (DNS). However, the range of scales necessary (from an ABL height of 1000 m, to resolving the individual turbine blades) that must be resolved to simulate a wind farm in the atmosphere makes the use of DNS impossible due to current computing limitations.

Instead, the method of large-eddy simulation is often employed, as it retains much of the accuracy of DNS while also reducing the computational burden to achievable levels. This enables LES to serve as a validation tool for more simple analytical models of wind farms, and also as a plant model to test wind farm control strategies. Reynolds Averaged Navier-Stokes (RANS) can also be used for validation, but has the trade off of lesser accuracy with faster computation. In LES, a filtering operation, denoted as $\tilde{\cdot}$, is applied to the Navier-Stokes equations to remove the scales smaller than a prescribed filter width Δ. The Navier-Stokes equations after the filter is applied have the form

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0 \tag{2.27}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\widetilde{u_j u_i} = -\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_i} - \nu\frac{\partial^2 \tilde{u}_i}{\partial x_i \partial x_j} + \tilde{f}_i \tag{2.28}$$

Although filtering is not synonymous with averaging, it does share the property that enables the decomposition of the advection term in a similar way as the Reynolds stress $\overline{u_j' u_i'} = \overline{u_j u_i} - \overline{u}_j \overline{u}_i$:

$$\widetilde{u_j u_i} = \tilde{u}_j \tilde{u}_i + (\widetilde{u_j u_i} - \tilde{u}_j \tilde{u}_i) = \tilde{u}_j \tilde{u}_i + \sigma_{ij} \tag{2.29}$$

In this case, σ_{ij} represents the subgrid stress tensor. Once this decomposition is applied to the filtered equation, we find the following equations

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0 \tag{2.30}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \tilde{u_j}\frac{\partial \tilde{u}_i}{\partial x_j} = -\frac{\partial \sigma_{ij}}{\partial x_j} - \frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_i} - \nu\frac{\partial^2 \tilde{u}_i}{\partial x_i \partial x_j} + \tilde{f}_i. \tag{2.31}$$

After the filtering, the viscous term is negligible due to the gradient of the filtered velocity being smooth. Additionally, the advective term can be rewritten into a rotational form by adding a "Bernoulli" term, which has the form

$$-\tilde{u}_j \frac{\partial \tilde{u}_j}{\partial x_i} = -\frac{\partial}{\partial x_i}\left(\frac{1}{2}\tilde{u}_j \tilde{u}_j\right). \tag{2.32}$$

Finally, the subgrid stress tensor can be decomposed into a deviatoric part by removing its trace $\sigma_{ij} = \tau_{ij} + \frac{1}{3}\sigma_{kk}\partial_{ij}$. Now the trace of the subgrid stress and the Bernoulli term can be absorbed into a modified pressure term that has the form

$$\tilde{p}^* = \tilde{p}/\rho + \frac{1}{3}\sigma_{kk} + \frac{1}{2}\tilde{u}_j\tilde{u}_j. \tag{2.33}$$

Substituting these expressions into the filtered equations gives the filtered Navier-Stokes equations

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0 \tag{2.34}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \tilde{u_j}\left(\frac{\partial \tilde{u}_i}{\partial x_j} - \frac{\partial \tilde{u}_j}{\partial x_i}\right) = -\frac{\partial \tau_{ij}}{\partial x_j} - \frac{\partial \tilde{p}^*}{\partial x_i} + \tilde{f}_i, \tag{2.35}$$

which is the set of equations solved in LES.

The models in this thesis are validated using two different LES codes. The first is the Simulator for Wind Farm Applications (SOWFA), with is an LES code developed by the the National Renewable Energy Laboratory (NREL) [88, 89]. The second is our in-house open-source JHU LESGO code, which descends from the LES code of Albertson [90], and simulates Cartesian domains using pseudo-spectral numerics [91]. The JHU LESGO code is used for validation for the ALC model and the control-oriented model presented in Chap. 3, as well as for a wind farm plant for the control framework presented in Chap. 5.

The two general simulation techniques for representing wind turbines within a wind farm simulation are the actuator disk model (ADM), where the turbine forcing is distributed over the turbine disk area and applied to the flow, and actuator line model (ALM), where the forcing is distributed along each turbine blade to be applied to the flow. The LES simulations in this thesis employ the actuator disk model. Velocity wake profiles from LES using ADM have been shown to be in good agreement with those using (ALM) [92]. Simulations employing ADM can be accomplished using

coarser LES grid-spacings as compared to those using ALM, and thus the choice of ADM enables us to run many more cases (inflow angles) and run simulations for longer times (aiming for statistical convergence).

2.2.2 Circular Wind Farm

In this section we compare ALC model predictions to LES results generated using the NREL SOWFA code [88, 89] for the circular wind farm configuration shown in Fig. 2-7. Comparisons include both total wind farm power and power from individual wind turbines for 12 wind inflow directions, 30 degrees apart, spanning a full 360 degrees. Figure 2-7 indicates the orientation of the wind directions; the 0° direction corresponds to a wind inflow direction from the north. Angles are then measured in the clockwise direction (i.e., at 90° the flow is going from right to left, and for 270° the flow is going left to right). Given this orientation, the wind farm is symmetric around the east-west axis but has more complicated wake interactions than a rectangular lattice as the streamwise and spanwise spacings between turbines are not uniform for any given wind direction.

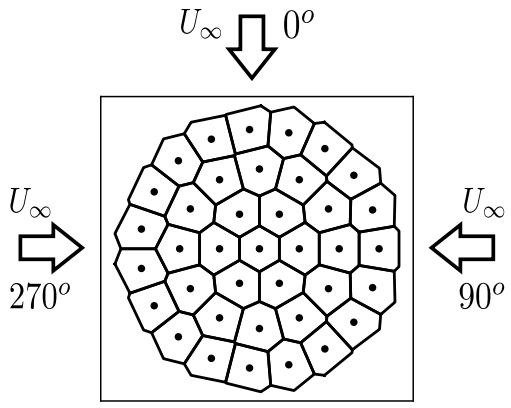

Figure 2-7. Orientation of wind direction for the circular farm

The LES of the farm comprises a set of 38 NREL 5-MW reference turbines [93] represented by actuator disk models. The simulation domain used $N_x \times N_y \times N_z =$

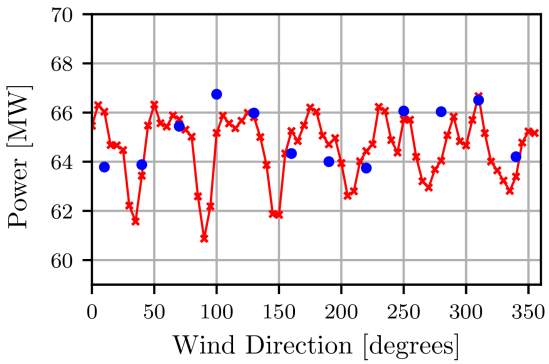

Figure 2-8. Average power for the circular wind farm as a function of wind direction obtained from the NREL LES (•) compared to the ALC model-predicted power with the model run in 5° increments (red line -x-).

$500 \times 500 \times 100$ grid-points. The inflow has a mean velocity of 8 m/s and was generated by a precursor simulation. The simulation uses a roughness height of $z_0 = 0.15$ m. The details of this simulation are available in [94].

Figure 2-8 shows the average power as a function of wind direction obtained from LES at 30° intervals beginning at 10°, denoted by the blue markers. The model data is also on the figure, represented by the red line, and will be discussed in more detail later. The LES data points were found by averaging the total power output of the LES farm over a time interval spanning approximately 0.75 hours of real time farm operation, which represents approximately five flow-through times. One flow-through time represents the time it would take a fluid parcel in the flow to travel from the inlet to the exit of the farm.

To provide an accurate comparison, the parameters used to evaluate the ALC model are selected to closely match those of the LES. We therefore use the values for the NREL 5-MW reference turbine [93], which has a hub height of $z_h = 90$ meters and a diameter of $D = 126$ meters. The inflow velocity distribution $U_\infty(y)$ is found using average inflow velocity data from the LES. All of the LES have very similar inflow profiles, so an average over the 12 simulations was used as the ALC inflow profile for

all wind directions. For the lower roughness height in the model, we use the roughness value from the LES: $z_{0,lo} = 0.15$ meters. The maximum boundary layer height was set to $\delta = 750$ meters, since the LES had a temperature inversion layer at this height, which caps the growth of the boundary layer.

In addition to the turbine parameters and flow conditions, we match the local thrust and power coefficients (C'_T and C'_P, respectively) of the turbines in the LES for the ALC model evaluation. Through this process, we determine one value for the thrust and power coefficients, respectively, over all ALC implementations. In general, the thrust and power coefficients are determined by the turbine type and its operating conditions. In our case, we have the power produced as well as the velocity at the disk for the wind directions $10° - 190°$ (in $30°$ increments for a total of 7 directions) for each turbine from the LES. We extract the coefficients by rearranging Eq. 2.24:

$$C'_{P,n} = \frac{2P_n}{\pi R^2 u_{d,n}^3 \rho}. \tag{2.36}$$

Using this expression, we first average over the time series to find an average local coefficient of power for each turbine. Figure 2-9 shows the local coefficients of power calculated for all seven directions, plotted by turbine number. The red triangles represent unwaked (or freestream) turbines in that orientation and the blue circles represent the waked turbines. There is significant spread in the local coefficients of power for the turbines. However, all of the freestream turbines are located at the bottom of the plot, around $C'_P \approx 1.4$. Since all of the spread occurred in the waked turbines, where turbulent fluctuations may skew calculations of the local power coefficient, the waked turbines were judged to be a less accurate measure for this quantity. Therefore, we use the average value from the unwaked turbines, $C'_P = 1.387$, for the local power coefficient in the ALC model.

Once the average local power coefficient was calculated, Blade Element Momentum

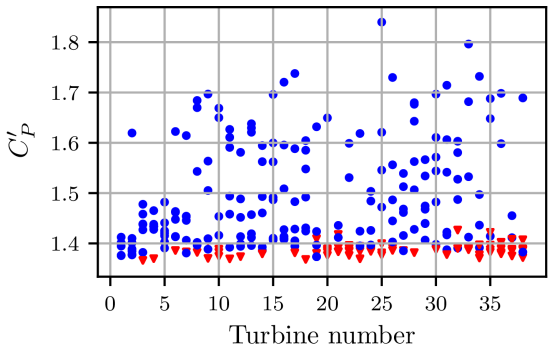

Figure 2-9. The unwaked (▼) and waked (•) local power coefficients calculated from the data for wind directions $10° - 190°$ (in $30°$ increments)

theory is used to find the relationship between C'_P and C'_T:

$$C'_P = \gamma C'_T \tag{2.37}$$

where the prefactor $\gamma = 0.9032$ was determined using the specifications of the NREL 5MW turbine (we are grateful to Dr. Carl Shapiro (personal communication) for performing this calculation). In the ALC model, C'_T is used to calculate the planform thrust coefficient and the local forcing of velocity deficit in the wake model, while C'_P is used to calculate the power from the velocity computed at the disk from Eq. 2.24.

The ALC model was run using the parameters outlined above for the cases where the wind is approaching from $0° - 360°$ in increments of $5°$. Figure 2-8 compares the ALC model power output predictions (red line) to the LES results (blue dots). The farm is symmetric about the east-west axis, but due to the nonuniform inflow used in the simulation, the model results are not quite symmetric. We can see that in most cases the ALC model results match those of the LES, i.e., they fall reasonably close to the blue symbols. The one noticeable outlier is for the directions $10°$. In this case the model over-predicts the power, however, the ALC prediction for a $15°$ inflow direction, which is a small change in inlet angle, provides a closer match to this LES point. This highlights how small changes in angle can significantly impact the power output of the wind farm. Such changes are especially apparent, for example, over the range

90°-110°.

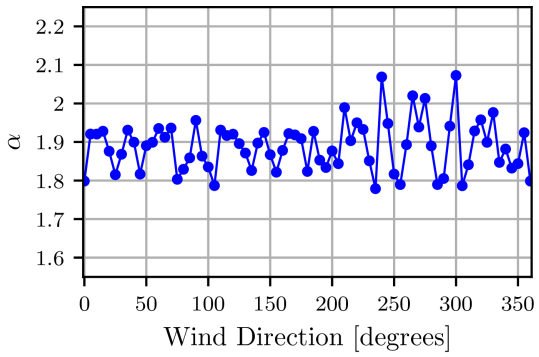

Figure 2-10. Model parameter α obtained from the error minimization in the ALC model as a function of wind direction for the circular wind farm.

Figure 2-10 shows the free parameter α obtained from the error minimization over the waked portion of the farm, as a function of wind direction. The fact that the results are on the order of unity (around 2) confirms the validity of the scaling assumptions underlying Eq. 2.20. However, a non-trivial dependence on angle can be observed, which is a result of the complicated relationships involved in the model, the turbine geometric layout and the transverse variations in inflow velocity.

We now compare the post-optimization hub-height velocity field obtained from the ALC model with the average velocity field from the LES. Figure 2-11 shows (a) the average velocity field from the LES and (b) the velocity field calculated by the ALC model when the incoming wind direction is 190°. We can see that allowing nonuniform inflow enables the ALC model to reproduce the variations of higher and lower velocities seen across the LES, which continue back through the wind farm. However, the ALC model also captures wake superpositions, as can be seen by examining the wakes at the back of the farm. The LES shows some flow acceleration in between the wakes which the ALC model does not capture.

We next examine the velocity averages in each Voronoi cell as well as the turbine power output prediction for a 190° inflow direction. The top row of Fig. 2-12 compares

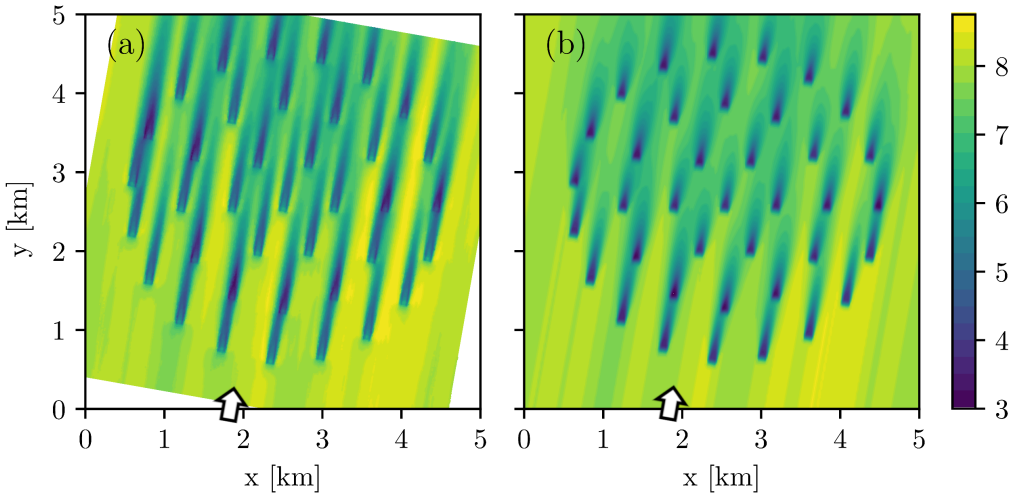

Figure 2-11. Comparison of (a) the time-averaged velocity field from the LES data to (b) the velocity field calculated from the ALC model for a wind direction of 190°. The arrows indicate the incoming flow direction.

the planar averaged velocities in each Voronoi cell over the farm in Fig. 2-4 for (a) the LES, (b) the resulting ALC wake model and (c) the resulting ALC top-down model velocity fields for the 190° wind inlet direction. We can see that the wake model, which provides the output of the ALC model, and the LES average velocity fields compare well with each other. The freestream turbines agree well, which is a result of the use of a nonuniform inflow velocity profile. Further back in the wind farm, the LES average velocity plot has a pattern of cells where some of the cells have a slower average velocity and thus more wake effects. The wake model is able to capture the overall patterns of cell velocities well, although there are some cells with lower velocity in the LES. Note that the wake model and top-down model average cell velocities do not agree exactly since we minimize the difference over the entire farm to find a global α value rather than computing individual values of α on a cell-by-cell basis. Using only one variable in the minimization rather than cell-specific α values enables the models to inform each other with significantly reduced the computation time over a multiple α approach, but may reduce the accuracy of the model.

The differences between the LES and the wake and top-down portion of the models can be quantified by computing the root mean squared (RMS) error between the cell velocity values from the LES and from the model. The RMS error was calculated according to

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(u_{h,n}^{LES} - \overline{u}_{h,n}^{m})^2}, \tag{2.38}$$

where $u_{h,n}^{LES}$ denotes the average cell velocity from LES and $\overline{u}_{h,n}^{m}$ represents the average cell velocity from the ALC model (in the figure, (b) wake model and (c) top-down model). These values are reported in the caption of Fig. 2-12

Next, we compare the ALC approach to some prior wind farm analytical models. In the second row of Fig. 2-12 we compare scatter plots of the power produced by each turbine in the farm computed with the Jensen model, the model in [22] and the ALC model. Here, the LES power is on the x-axis and the model power is on the y-axis, so points along the 45° line (marked as a black dashed line) represent $1:1$ agreement. The red triangles denote the freestream (unwaked) turbines, while the blue circles are the waked turbines. The left plot (d) shows the scatter plot using the Jensen model with a wake expansion coefficient derived from the average friction velocity of the inflow, which was $k = 0.0653$. The middle plot (e) uses the model in [22] with Voronoi cells and a uniform inflow. In this case, we can see that the uniform inflow prevents the freestream turbines from capturing the behavior of the LES, causing all of the freestream turbines to give almost the same value. In the right plot (f), we show results from the ALC model, where the freestream turbines are much closer to the 45° line as a result of the nonuniform inflow profile used. The ALC model predictions for the waked turbines are also closer to the LES results. The improvement in this direction is especially prominent in turbines with higher power, whose prediction is aided by the inclusion of the nonuniform inflow velocity profile. We also calculated the RMS error between each model and the individual power from LES, evaluated according the Eq. 2.38 for power rather than velocity, given in the caption of Fig. 2-12.

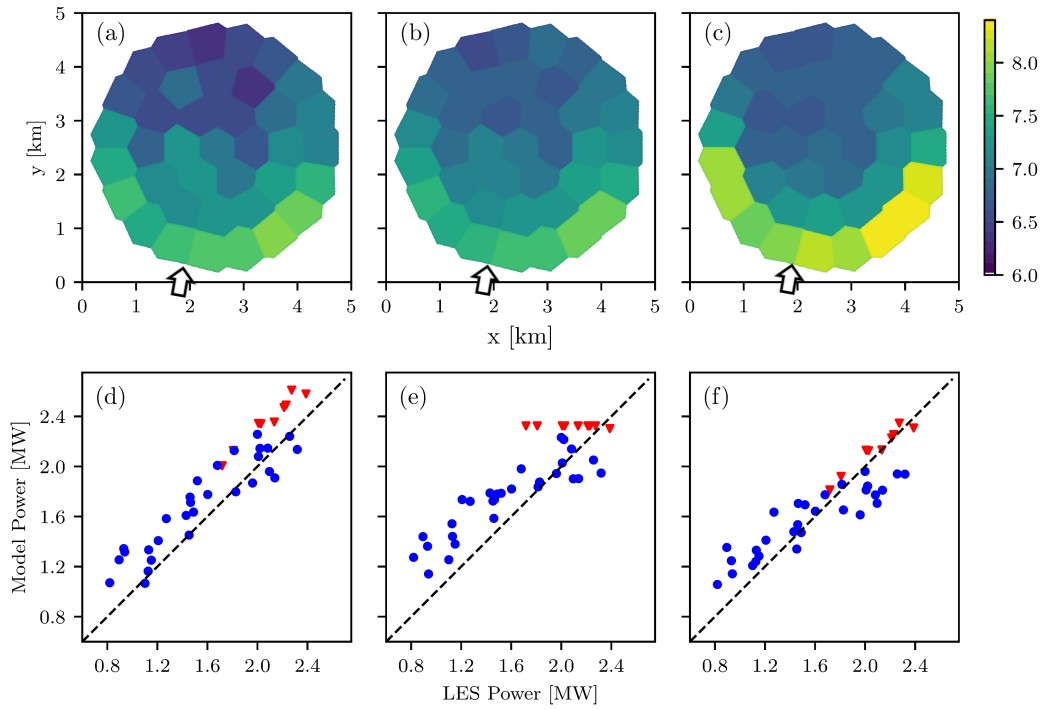

Figure 2-12. For a wind direction of 190°, in (a)-(c) the average planar velocities is each cell are compared from the following sources: (a) the LES data ($\overline{u}_n = 7.05$ m/s), (b) the ALC wake model ($\overline{u}_{n,wm} = 7.08$ m/s), and (c) the ALC top-down model ($\overline{u}_{n,td} = 7.21$ m/s). The root mean square (RMS) error between the wake model and the LES cell velocities is 0.135 m/s, and between the top-down model and the LES cell velocities is 0.282 m/s. The arrows indicate the incoming flow direction. The scatter plots show the power for the unwaked (▼) and waked (•) turbines compared to the LES power from (d) the Jensen model ($RMSE = 0.241$ MW), (e) the model in Shapiro et al. 2019 ($RMSE = 0.299$ MW), and (f) the ALC model ($RMSE = 0.209$ MW), where a 1:1 relationship is represented by a 45° line (- -). The average indivual turbine power from the LES for the 190° direction is 1.684 MW,

This is the only direction where a detailed comparison is completed with multiple analytical wake models, but a comparison for the other LES directions should be investigated in future work.

Figure 2-13 provides a detailed comparison of the LES field to the wake and top-down models comprising the ALC model for four different incoming wind directions: 70°, 130°, 190°, and 250°. Each direction corresponds to a row in the figure, while each column depicts a different quantity for that direction. The first column shows

the time averaged LES velocity field with the Voronoi cells drawn in to show the area over which the planar velocity values are calculated. The first column also has arrows to indicate the direction of the incoming wind. The second column shows the cell averaged LES planar velocities calculated from the time averaged LES velocity fields. The third column shows the average planar velocities from the wake model of the ALC model, and the last column shows the same quantity from the top-down model. In all the cases, the wake model is able to predict the pattern of faster and slower velocity in the cells in the waked region, the back half of the wind farm, of the LES farm quite well. The pattern of the cells is important in seeing how the turbines and wakes interact with each other. The accurate prediction is consistent with the power output comparison in Fig. 2-8, where ALC model matches the LES for the 130° and 250° degree directions quite well. However, in the 190° direction, the model over predicts the power, which is clear from the slightly higher velocities predicted from both the top-down and wake models for that wind direction.

We can further examine wake interactions between turbines by examining the wake expansion coefficient determined as part of the ALC model. Even though only one parameter is matched across the whole farm (α), the model gives varying wake expansion coefficients across the farm because each cell has a different freestream velocity and friction velocity $u_{*,hi,n}$, with the freestream friction velocity $u_{*,n}$ used for the freestream turbines. Figure 2-14 shows how the wake expansion coefficient differs across the farm for the same four wind directions shown in Fig. 2-13, this time with the wind direction changing across the row. A lighter value indicates a higher wake expansion coefficient, and thus a wider wake, more expansion and velocity recovery, and more turbulence in the flow. A darker value indicates a lower value and a stronger wake effect and velocity deficit on the following turbines. Here, we can see that the wake expansion coefficient is higher overall for the first direction (a) 70°, but is lower for the later three directions:(b) 130°, (c) 190° and (d) 250°. The

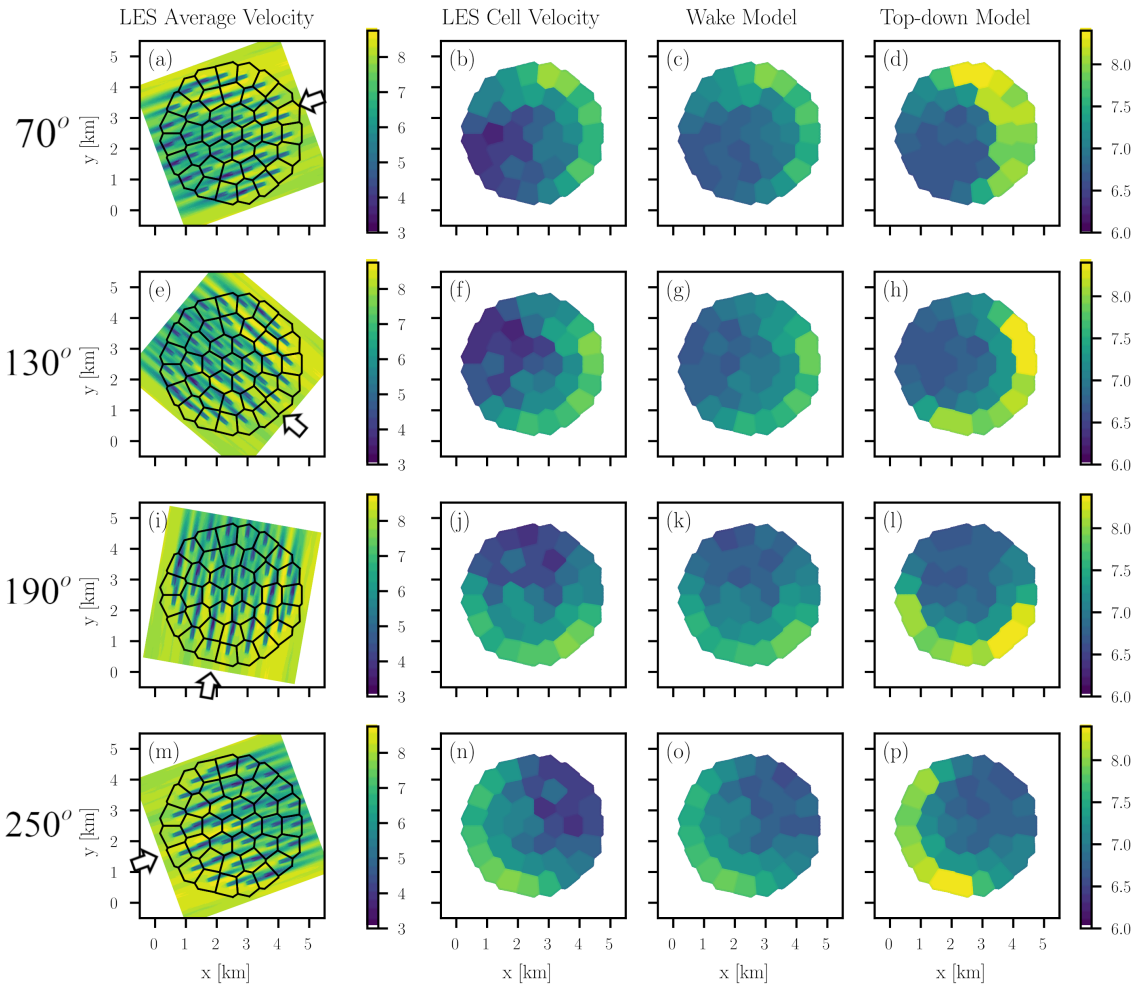

Figure 2-13. Comparison of LES data and the planar velocities given by the ALC model. For 70°, (a) shows the average velocity data from the LES with the Voronoi cells drawn, (b) shows the average planar velocity calculated from (a), (c) shows the wake model average planar velocity, and (d) shows the same quantity from the top-down model. The subfigures (e)-(h) show these same quantities for 130°, subfigures (i)-(l) show the velocites for 190°, and subfigures (m)-(p) show the velocites for 250°.

significantly higher wake expansion coefficient for the case of 70° for turbines inside the wind farm can be attributed to the fact that for this wind direction, several inside turbines are freestream since they do not fall in wakes. The freestream turbines have smaller expansion coefficients in their wakes, i.e., the wakes dissipate more slowly initially. However, once these relatively stronger wakes interact with downstream turbines inside the farm, the large friction velocity and low mean velocity associated

with the cells with strong wakes translates into large wake expansion coefficients for the turbines further downstream. These interactions explain the suddenly large k_w values for the 70° degree case.

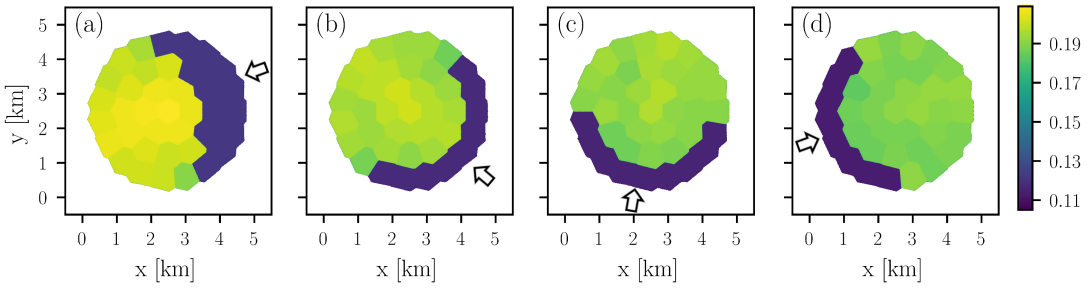

Figure 2-14. Contour plots shows the value of the wake expansion coefficient calculated by the model for (a) 70°, (b) 130°, (c) 190° and (d) 250°.

In all cases, the freestream turbines have a significantly lower wake expansion coefficient than the waked turbines. To capture the larger velocity deficit that occurs behind freestream turbines, we use the freestream friction velocity for the freestream wake expansion coefficient since they are the first to interact with the incoming flow. The waked turbines wake expansion coefficients are determined based on Eq. 2.20, which depends on the local friction velocity and inflow in each cell. The variations in these values are significantly smaller than the difference between the unwaked and waked wake expansion coefficients. Another visible trend is that the wake expansion coefficient is typically the highest in the back center of the farm, as referenced from the freestream wind direction. This trend reproduces the realistic effect that an increase of turbulence further back in the farm is due to cumulative wake interactions.

In addition to the planar velocity comparisons presented above, we can also examine a more quantitative view of the model performance using scatter plots of the individual turbine velocities for each wind direction. Figure 2-15 shows the scatter plots for all 12 directions provided by the LES data. As in Fig. 2-12 the freestream turbines are represented by the red triangles, the waked turbines are represented by blue circles and

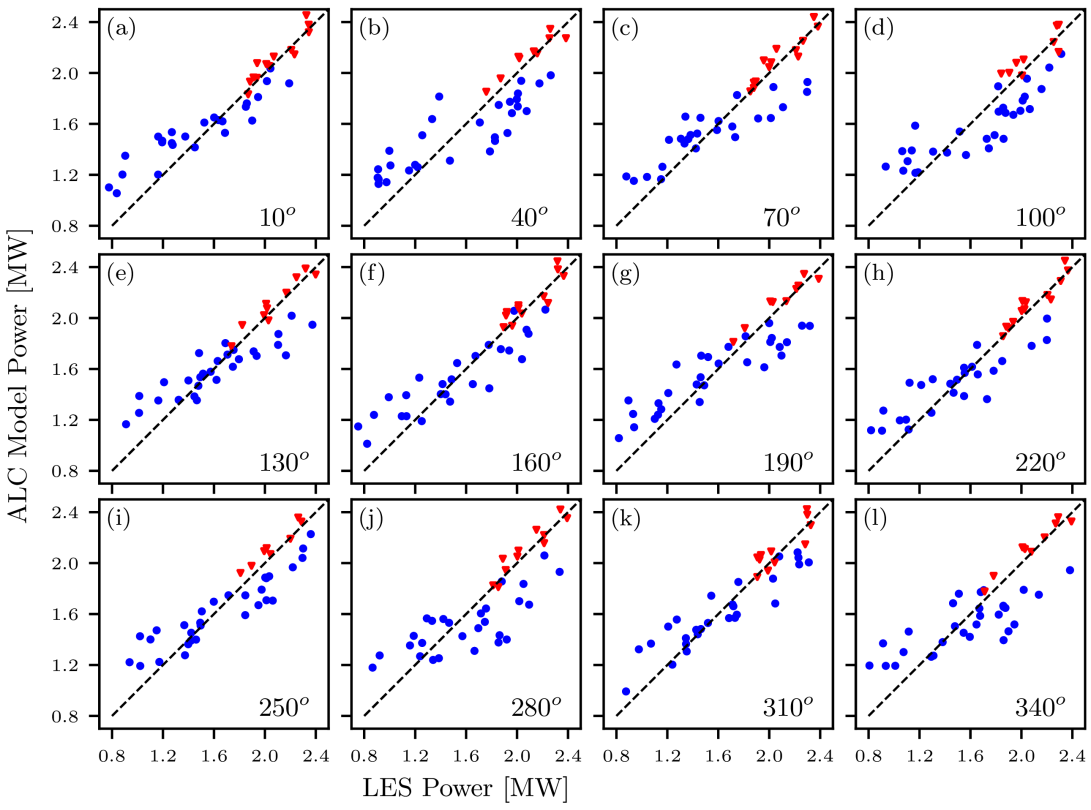

Figure 2-15. Scatter plots comparing the power predicted by the ALC model with the LES data for all the LES wind directions, where (▼) represents the unwaked turbines and (•) represents the waked turbines and a 1:1 relationship is represented by a 45° line (- -). The RMS error across all of the directions is $RMSE_{ALC} = 0.197$ MW.

the 1 : 1 correspondence line with LES data is represented by the black dashed lines. As can be expected, the ALC model provides relatively accurate power predictions for the freestream turbines compared to the LES results, while the results for the waked turbines have more spread overall. Some of the discrepancies can be explained by the spread in the local power coefficients calculated from the waked turbines, which is evident in Fig. 2-9, shown earlier. Since we chose the power coefficient based on the freestream turbines, the waked turbines could deviate from this constant value in the LES somewhat, possibly due to wake interactions, causing more spread in the results. Overall, however, the ALC model reproduces the LES trends quite well for any of the

angles considered, with an overall RMS error of 0.197 MW.

2.2.3 Hybrid Regular/Random Wind Farm

In this section, we compare the model to LES of a wind farm layout that mixes a regular array region with a random region. More specifically, if the flow moves from left to right, the wind farm starts with an array consisting of four staggered rows, and then has a total of fourteen additional turbines placed in a random fashion behind the staggered-array turbines, shown in Fig 2-16. The LES is performed with the JHU LESGO code [91] with an actuator disk model using a correction factor accounting for finite grid resolution [95]. The code has been validated in simulations of several wind energy applications [96–98]. As the inflow, the simulation uses a field generated by the concurrent-precursor approach [99]. The turbines have a diameter and hub height of $D = z_h = 100$ meters and a ground surface roughness height of $z_{0,lo} = 0.1$ meters is used to prescribe the bottom boundary condition of the LES. The main simulation domain uses $N_x \times N_y \times N_Z = 384 \times 256 \times 128$ grid-points. The Lagrangian scale dependent dynamic subgrid-scale model [100] is used to determine the eddy-viscosity without adjustable parameters. The turbine (local) coefficient of thrust is kept constant at $C_T' = 1.33$ throughout the simulation. The same parameters are used in the ALC model. The LES is run over a range of 90°, from 270° to 345° in 15° increments for a time period of approximately ten flow-through times for the farm, which translates to roughly 1.75 hours in real time. The data also provide a well-converged averaged inflow condition for the ALC model application.

First we examine the inflow direction of 270°. Figure 2-17 shows the comparison of the time averaged streamwise velocity field at hub-height given by (a) the LES and (b) the ALC model. The inflow used in the ALC model was taken from the time-average LES velocity field at $x = 0$ m in front of the turbines. The dimensionless wake expansion parameter found through the error minimization was $\alpha = 1.68$, which

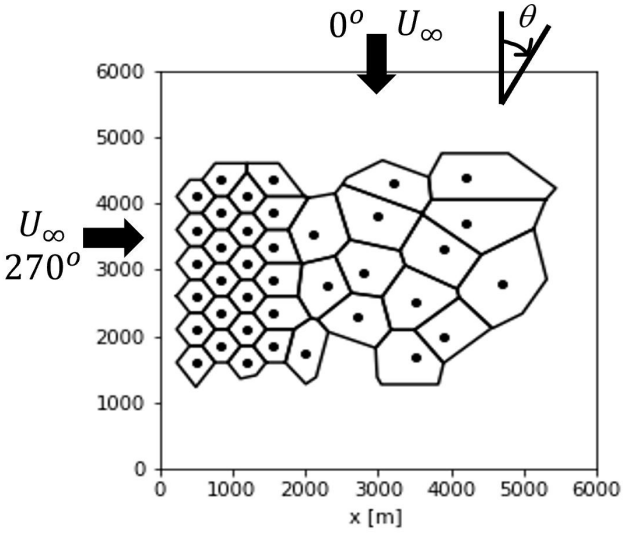

Figure 2-16. Orientation of the wind direction for the hybrid regular-random wind farm. This figure shows the locations of the turbines along with the Voronoi cells for each turbine.

is slightly lower than the values found in the circular wind farm study, but still of order unity. We can see that the ALC model captures the flow well, and represents the wakes, particularly in the rear of the farm, quite accurately. The wakes in the ALC model appear sharper and the velocity is slightly lower in the wake as compared to the LES results, but their relative speed of decay is similar in both cases. The comparison also shows the importance of the nonuniform inflow condition in accurately modeling the overall flow especially near the inlet. The lower panels of Fig. 2-17 show the average velocities in each of the Voronoi cells obtained from (c) the time and cell-averaged field from the LES the (d) the mean velocity obtained from the ALC model (the wake model part). We can see that the cell velocities match well, showing the same general trends in both cases.

We can also analyze the wake expansion coefficients calculated for the hybrid wind farm, shown in Fig. 2-18. In this farm, the wake expansion coefficients start out the lowest at the front of the farm, then increase to their highest values in the

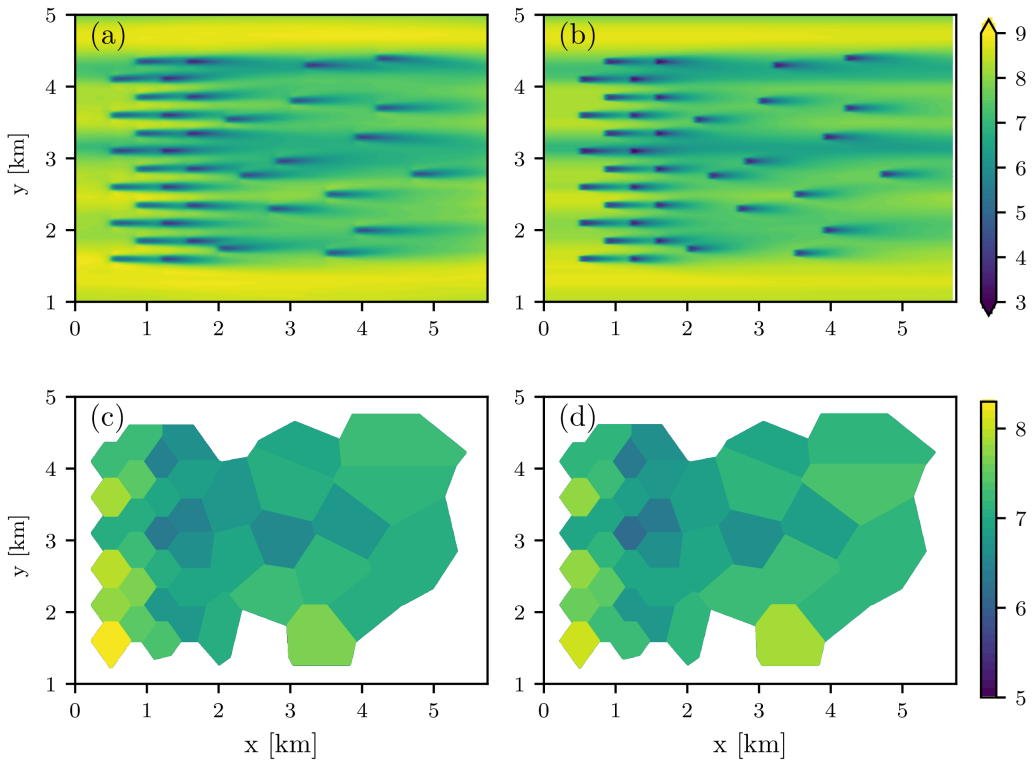

Figure 2-17. Comparison of the velocity fields (a) of the LES with (b) the ALC wake model, and the comparison of the hub height average cell velocity from (c) the LES and (d) the ALC model.

rows immediately after the first rows. The values then gradually decrease from this point as a function of downstream distance, which follows the expected trend for the friction velocity development in a smooth-to-rough transition. The boundary layer flow downstream of the transition, in the third and fourth rows at around 1-1.5 km in the streamwise direction, displays larger mean shear near the surface (here at the turbine height) and thus larger turbulence levels and k_w. Further downstream, as the boundary layer approaches equilibrium conditions where the turbines are randomly placed, the shear decreases and so do the turbulence levels causing a slight decrease in the wake expansion coefficient

Now we consider the other incoming wind directions over which the LES were performed in 15° increments: $285° - 345°$. Figure 2-19 shows a comparison between

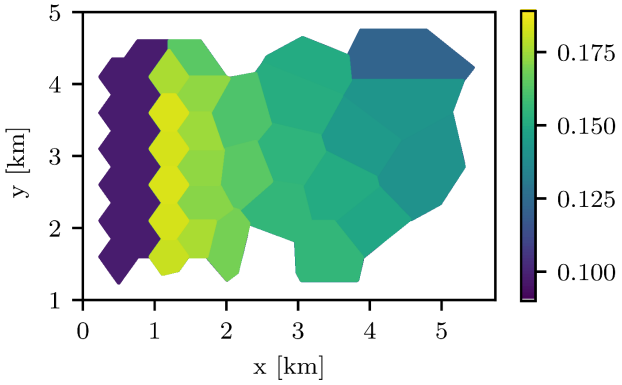

Figure 2-18. Contour plot of the k_w values found in the 270° case for the hybrid wind farm.

the average velocity fields from the LES with the velocity fields predicted by the ALC Model. While the ALC Model wakes have a slightly different shape when compared to the LES wakes, the wake expansion coefficients are predicted well. The ALC Model does well when predicting the strength of the wake of a forward turbine when it reaches a turbine positioned in its wake, which is essential in predicting the power output of the farm. This can be seen particularly when considering the comparison between the fields for 300°, where the ALC model predicts the cluster of six turbines in the lower left corner of the farm well. This is particularly important since these turbines lie in the area of the farm that is most affected by wake effects.

This study is also informative when considering how the changing inflow angle affects the composition of the wake interactions. For example, the randomly placed turbines are all waked when the inflow is at 270°, but many are in the freestream when the inflow angle reaches 345°. This change affects the wake expansion coefficient of the turbines as well as their power outputs. Since the ALC Model enables a separate calculation of the wake expansion coefficient for each turbine, it enables us to sudy how the wake expansion coefficients of the turbines react to the change. Figure 2-20 shows how the 29^{th} turbine in the farm, which is denoted by the black circle in Fig.

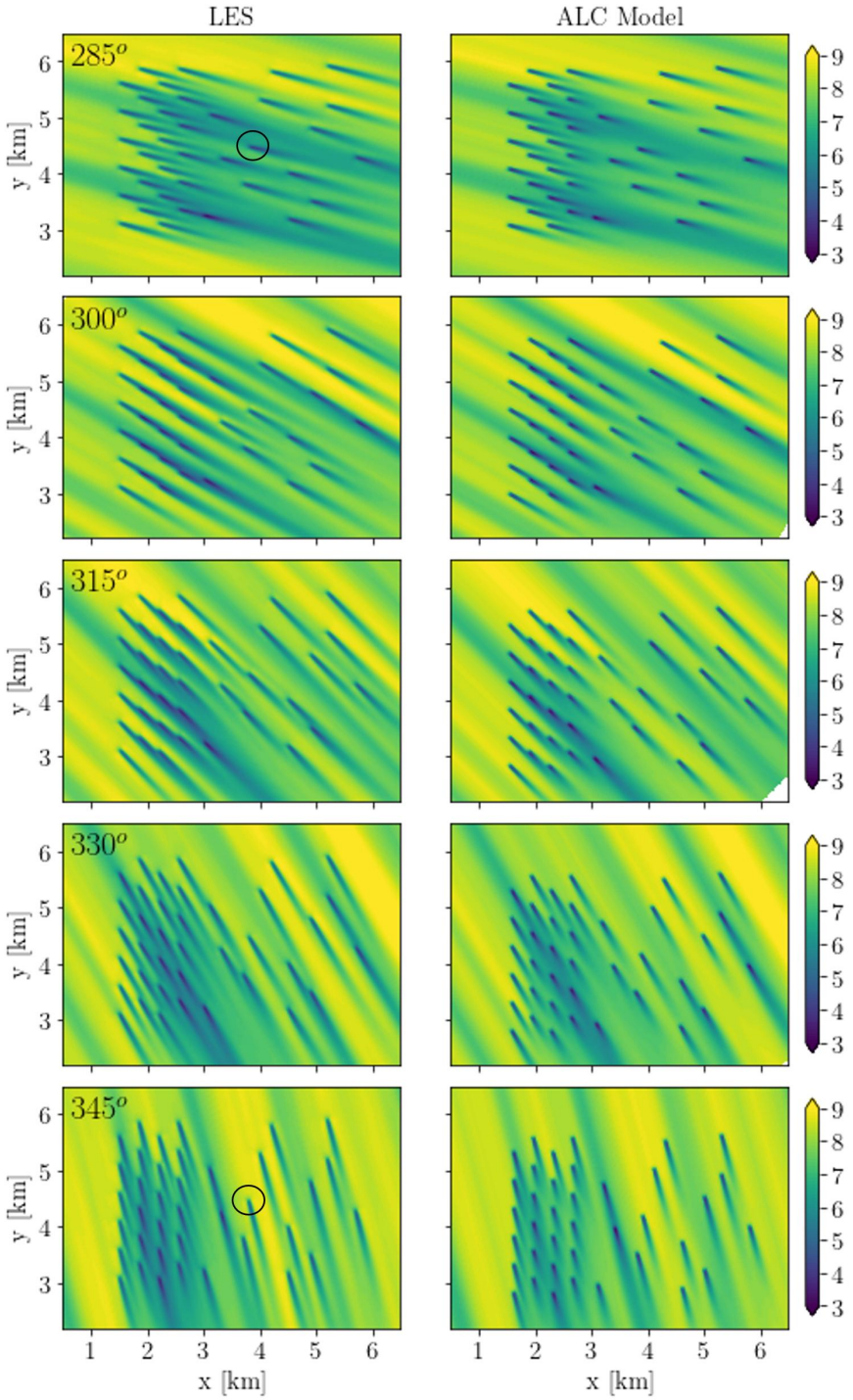

Figure 2-19. Comparison of the velocity fields of the LES (left) with the ALC wake model (right) for the range of inflow directions. The black circles indicate the 29^{th} turbine, which is referenced in Fig. 2-20.

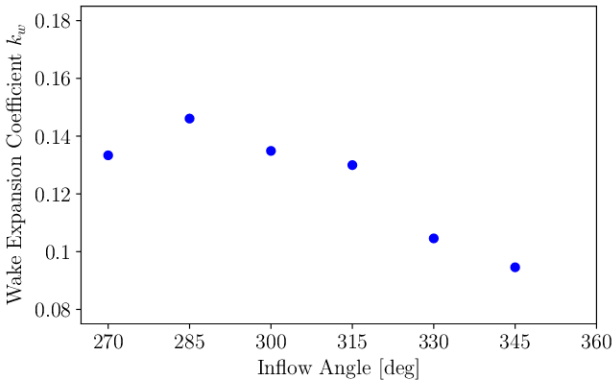

Figure 2-20. The value of the wake expansion coefficient for Turbine 29 as a function of the freestream inflow angle as an example of how the wake expansion coefficient throughout the farm responds to changes in incoming wind direction.

2-19, changes with the inflow angle. We can see that the wake expansion coefficient peaks when this turbine is fully in the wake of multiple turbines, shown in Fig. 2-19 in the 285° case. Then, as the inflow angle shifts the wake expansion coefficient decreases until it is fully in the freestream flow, as in the 345° case. This is expected since freestream turbines typically have lower wake expansion coefficients since they do not encounter the extra turbulence added to the flow by the turbine wakes. This model enables us to study the behavior of the wake expansion coefficient more closely and connect this behavior with the physics of wake interactions in wind farms.

Finally, as a more quantitative comparison between the ALC model results and the LES results, Fig. 2-21 shows a scatter plot comparing the power from each turbine over all the wind directions. As before, the freestream turbines in each direction are denoted by the red triangles while the waked turbines further downstream are denoted by the blue circles. We can see that the points follow the $1:1$ correspondence line rather closely, implying very good agreement of the ALC predictions with the LES results. There is substantially less scatter compared to the circular wind farm case considered in the last section, with this study having an average RMS error over all directions of 0.081 MW. This could be due to the longer time period used for averaging,

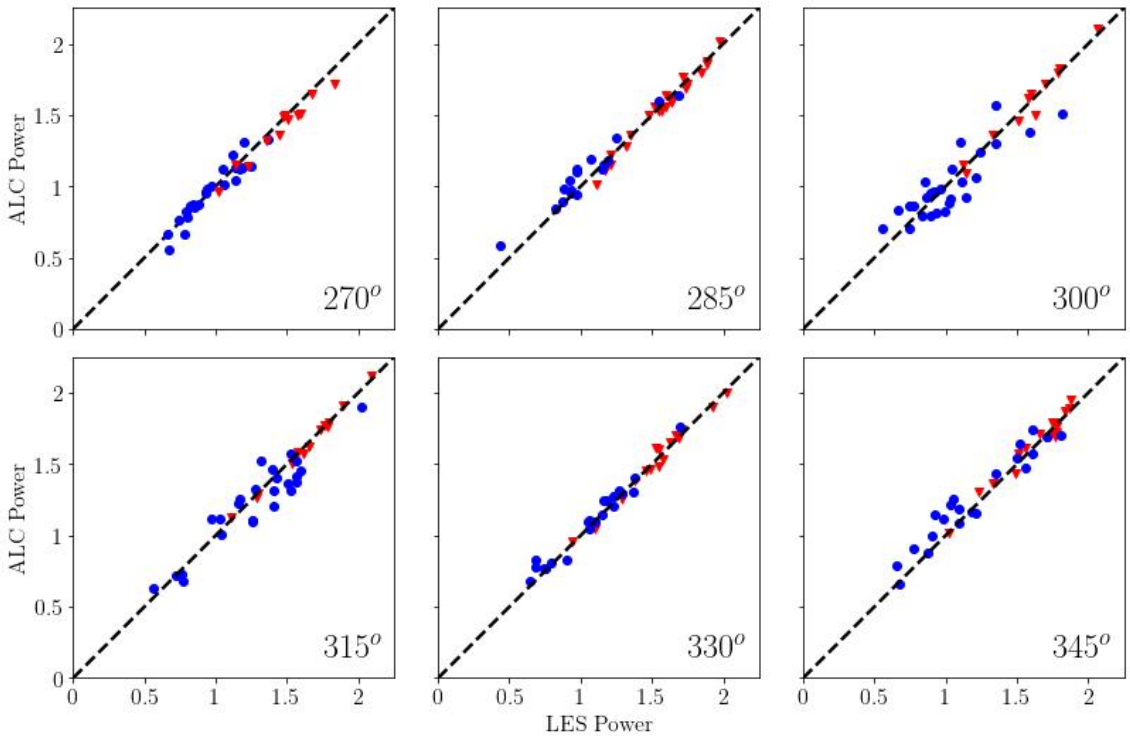

Figure 2-21. Scatter plot comparing the power predicted by the ALC model with LES data from the mixed regular-nonuniform wind farm, where (▼) represents the unwaked turbines and (•) represents the waked turbines and a 1:1 relationship is represented by a 45° line (- -). The RMS error for this configuration over all directions is $RMSE_{ALC} = 0.081$ MW.

which for the mixed regular-nonuniform wind farm was 1.75 hours (approximately 11 wind farm flow through times), while it was only 45 mins (approximately 5 wind farm flow through times) for the circular wind farm. We recall that the ALC model is designed to predict the mean flow, which requires significant time averaging in LES. In summary, we can conclude that the ALC model is able to capture the mean flow and power production of the mixed regular-nonuniform wind farm quite well.

2.3 Conclusions

The Area Localized Coupled (ALC) model combines a wake model and a top-down model in a localized fashion to create a more generally applicable approach that

predicts several useful quantities about wind farms. The model uses Voronoi cells to divide the wind farm up into areas that belong to each turbine. Since the calculations can be applied to each cell, the model can be applied to regular as well as irregularly arranged wind farms. The model includes an important parameter, the wake expansion coefficient, that is obtained as function of the local turbulence properties as described by the ratio of the friction velocity to the mean advection velocity. The dependence of the wake expansion coefficient on the local friction velocity determined from the growth of a local internal boundary layer enables a localized result involving different position-dependent wake expansion coefficients across the wind farm. A global parameter α is selected to minimize the difference in the cell-averaged mean velocity predictions between the top-down and wake superposition models. In addition, the model can be implemented using a nonuniform inflow for both the wake and top-down model constitutive parts. This feature improves the performance of the model for all of the turbines but especially for turbines near the inflow that are directly exposed to the position-dependent inflow velocity.

The model was validated using the LES data for two different wind farms. The first was a circular wind farm. In this case, the model was able to accurately predict the cells of higher and lower velocity across the wind farm when compared to the time average LES data. The ALC model also reproduced the velocity patterns in the cells further back in the farm, where the wake interactions are most important. The ALC model also captured general trends in the variation of total wind farm power with wind direction as obtained from LES data, and it predicted the power of individual turbines well.

The ALC model was also applied to a mixed regular-nonuniform wind farm for which LES data were also generated. The wind farm started with regular staggered rows and then continued with a set of randomly placed turbines downstream. The ALC model-predicted velocity field matched the time-average velocity from the LES

very well. The model was able to predict the power of individual turbines quite accurately, as well.

The newly proposed ALC model provides more information than typical wake models since physically relevant information about position-dependent friction velocity, internal boundary layer and roughness height becomes available. Since the Voronoi tessellation is generally applicable to any turbine arrangement, there are no limitations to applying the ALC to any wind farm geometry, regular, random or mixed. It can account for position-dependent inflow velocity distributions. Additionally, the model has the potential to be extended to dynamic, time dependent situations if, instead of using the steady state solution of the partial differential equation for the velocity deficit, one uses the full time-dependent solution. It should be noted that the top-down model assumes a quasi-steady description of the boundary layer and so further improvements to include time-dependence in the top-down model may still be required. At any rate, such possible extensions make the ALC model an attractive and versatile option to be used in wind farm design and control. Future work will include the validation of this model on operational wind farms and possible extensions of this framework to a dynamic model.

Chapter 3

A Graph Model Representation of Wind Farms

In addition to physics-based modeling to improve our understanding of wake interactions, we also need a way to represent the dynamic response of wind farms to changing conditions. This chapter outlines a control-oriented modeling approach that is developed based on an estimation algorithm that leverages a network framework to compute the time evolution of the wind speed at the turbine hub-height and corresponding power output throughout the wind farm under dynamically changing conditions, which is published in [101]. Our approach exploits prior work showing the promise of representing a wind farm as a network of turbines described by a directed graph with the turbines as nodes and the interactions between turbines (i.e., the inter-turbine wake propagation) as edges, see e.g., [57, 58]. Those authors showed that the combination of such a model with a consensus algorithm led to improved wind farm speed estimates given noisy field data. The performance of optimal control algorithms has also been improved by adding constraints based on a corresponding wind farm graph [102]. A related network model in [103] used correlations between power output data from turbines to form real-time graphs describing the interactions between turbines in the wind farm. Graph representations have also been used to study the arbitrage potential of kinetic energy stored in aligned wind farm configurations

[104].

These previous studies relied on static graphs that are not designed to account for the dynamic changes in the turbine interconnections resulting from real-time changes. Accounting for these dynamics is complicated by the fact that changes in the wind direction and yaw propagate downstream at different rates depending on the inlet velocity and the geometry of the wind farm. Therefore changes at the inlet may not be seen at the outlet until various time delays have passed. We account for these behaviors by introducing a time-varying graph framework with edge switching [105]. This type of graph model has been widely used to capture similar types of information propagation structures, such as those in transportation networks [106–108]. We adopt this model to our problem setting to define the Dynamic Graph model by defining edges in terms of the time delays associated with the turbine to turbine propagation of the effects of changes in wind inlet direction or yaw through the farm. This behavior is captured using a time-dependent adjacency matrix that depends on the individual time delays associated with each edge. We impose dynamic changes in the graph structure due to changing conditions through an event-based framework. Our algorithm is then constructed as an input-output map that outputs predictions of the transient response of the system to the disturbances. Each step in the algorithm is computationally efficient enough to be implemented in real-time, horizon-based, control strategies, which is an advantage over current methods that are able to account for dynamic wind direction or yaw changes.

In Section 3.1, we present a version of the Dynamic Graph model that focuses on representing inlet wind direction changes in a wind farm. In Section 3.2, we adapt the Dynamic Graph model to also incorporate dynamic yaw changes.

3.1 Model Framework

We represent the wind farm as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where the turbines define the nodes $\mathcal{N}$ and the edges $\mathcal{E}$ describe the directed interconnections between turbines. In particular, when the wake of turbine j influences turbine i, there is a directed edge connecting node j to node i with non-negative edge weight whose magnitude is defined by the strength of the interaction. In general, a wake model is used to determine the interconnection structure and the edge weights of the edges since the scope of the wake needs to be determined in order to define the wake interactions. The adjacency matrix of $\mathcal{G}$, denoted by Λ, has elements $\lambda_{i,j} = 1$ if Turbine i is in the wake of Turbine j and $\lambda_{i,j} = 0$ if Turbine i is not in the wake of Turbine j. We designate turbines not affected by the wakes of any of the other turbines as freestream turbines, and these turbines act as the head nodes of the graph. Clearly, the structure of the wind farm means that the graph will not always, or perhaps ever, be connected. However, we can separate the graph into connected subgraphs, composed of leader-follower trees. The lead turbines are the freestream turbines, i.e., the turbines not affected by wakes of other turbines, and the follower turbines for each subgraph are all the turbines that are weakly connected to each leader turbine, i.e., there exists a directed path from the lead turbine to each follower. In this way, we can sum multiple connected subgraphs to obtain a representation of the entire system.

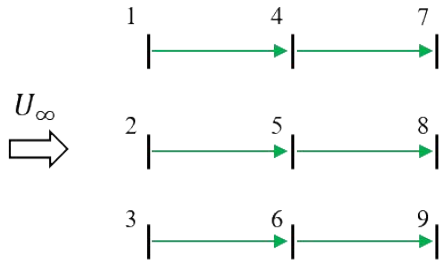

Figure 3-1. Example of a graph for an aligned wind farm where turbines 1, 2, and 3 are the lead turbines for the three subgraphs attained for the given wind direction.

Figure 3-1 shows an example of a graph for an aligned wind farm where, based on

the freestream velocity U_∞, nodes 1, 2, and 3 correspond to freestream turbines. For the given wind direction the graph structure is comprised of three line graphs with the directed edges shown, where the lead turbine of each graph only affects the turbines in its column. For example, in Figure 3-1, for the lead turbine, Turbine 1, the follower turbines would be Turbine 4 and Turbine 7. The three graphs obtained in this example are a product of the wind direction, the wind farm geometry, and the atmospheric conditions. For example, different atmospheric conditions could cause interconnections between the three graphs if, e.g., increased prevailing turbulence levels cause larger wake expansion coefficients leading to lateral wake overlaps. Arbitrary wind farm geometries and changing atmospheric conditions may also result in leader-follower trees that have shared nodes which must be taken into account in the implementation of the model. The next three subsections describe the building blocks of the input-output estimation algorithm.

3.1.1 Wind Farm System Graph Identification

We now describe how to identify the steady state graph representation of an arbitrary geometry wind farm corresponding to a given wind direction. This graph depends on the geometry and the current atmospheric conditions. Voronoi tessellation is used to partition the domain into cells obtained by placing each vertex equidistant from three nodes (turbines). We determine the freestream turbines by drawing a vector from each turbine location to the front of the farm along the current wind inlet direction. If the line only crosses one cell, its own, to reach the front of the farm, then it is determined to be a freestream turbine. Figure 2-2(a) shows an example of the line drawn to the front of the farm from the ninth turbine.

The turbines that are not identified as freestream for a given wind direction are then tested to determine if they are in the wake of another turbine. The wake of each turbine is defined through linear wake expansion with expansion coefficient k_w

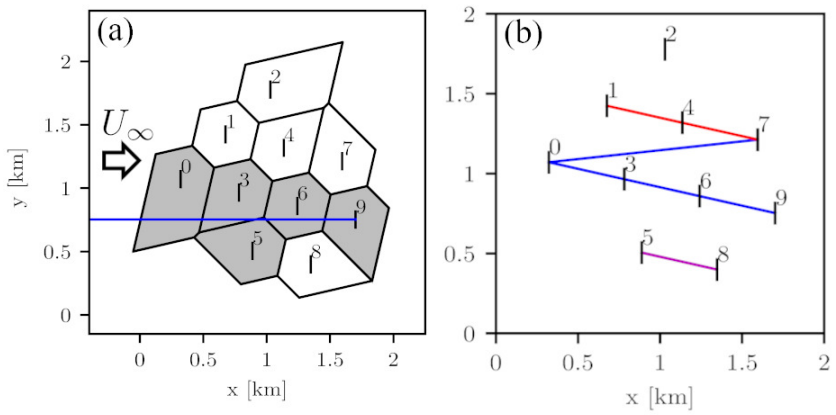

Figure 3-2. A simple 10 turbine wind farm example illustrating the process of defining the wind farm graph. (a) Voronoi cells drawn for a left-to-right wind direction. The grey shaded cells are the cells on the line from the ninth node (turbine) to the front of the farm. (b) The leader-follower graphs resulting from applying the algorithm to this wind farm with left-to-right wind direction. The different colors represent subgraphs with independent lead turbines.

determined from the atmospheric conditions as

$$k_w = \alpha^* \frac{u^*}{U_\infty}. \tag{3.1}$$

Here u^* is the friction velocity, U_∞ is the freestream velocity, and α^* is a model flow parameter of order one. A turbine is said to be in the wake of an upstream turbine if a line with slope k_w (in the coordinate frame of the incoming wind direction) extending from either edge of the upstream turbine hits the given turbine. To streamline the process, for each non-freestream turbine, we only test the cells that the line drawn to identify freestream turbines crossed to reach the front of the farm for that particular turbine. Taking the example in Figure 2-2(a), the cells of the turbines that would be tested for Turbine (node) 9 are shaded gray. If the upstream turbine's wake affects the current turbine, there is an edge defined between these turbines (nodes). The graph formed using the configuration shown in Figure 2-2(a) and $k_w = 0.0625$ is shown in Figure 2-2(b), where there is a subgraph associated with each of the freestream (lead) turbines (nodes 0, 1, 2, and 5) and the different subgraphs are represented by different colors. In this case Turbine 9 is not in the wake of Turbine 5, even though

the cell surrounding that turbine is shaded in Fig. 3-2 (i.e., the line crosses through it). However, a larger value of k_w may lead to a connection between these turbines. Further discussion of the definition of the wake coefficients for each turbine is provided in the next subsection.

3.1.2 System Dynamics

Having described the process for identifying the system graph, we next specify the system dynamics. The states of the system are the velocity deficits resulting from the interactions between each turbine pair (i.e., the edges of the graph). To define this quantity we use the concept of 'deficit coefficients', which represent the normalized velocity deficits between each turbine pair, see e.g., [109]. Our description of the deficit coefficient at Turbine i caused by Turbine j, represented by ϕ_i^j, is adapted from an existing turbine deficit model [22] that uses a super Gaussian wake profile. In particular, we apply this model in the far wake limit, where the model assumes a fully Gaussian wake profile and is employed in the deficit coefficient

$$\phi_i^j = \frac{2a}{\left(1 + 2k_w \Delta x_i^j\right)^2} W_i^j, \tag{3.2}$$

where a is the induction factor of the turbine, which is related to the local coefficient of thrust (C_T') by $a = C_T'/(4 + C_T')$. The direct downstream distance between turbines j and i normalized by the diameter of the turbines is Δx_i^j. The variable W_i^j is the average of the linear superposition wake function (Eq. 10 in Ref. [22]) for the wake of Turbine j over the disk area of Turbine i, which represents the extent of the effect of the wake of Turbine j. This coefficient can be computed as

$$W_i^j = \frac{1}{2\pi R_i^2} \int_0^{R_i} \int_0^{2\pi} W(D\Delta x_i^j, r_{i,j})\ r_i\ d\theta dr_i, \tag{3.3}$$

where the variable $r_{i,j}$, which denotes the distance from the center of Turbine j's wake to an infinitesimal point on the disk area Turbine i, is found using $r_{i,j} =$

$\sqrt{(R_j + r_i\cos\theta)^2 + (r_i\sin\theta)^2}$. Here, R_j is the distance between the center of the wake from Turbine j and the center of the disk of Turbine i.

The wake expansion coefficient varies throughout the farm depending on the properties of the farm. The local wake coefficient values for Turbine i, denoted by k_{w_i}, lie within an interval bounded by a freestream value (k_0), and a fully developed, or waked, value (k_{waked}). As in the ALC model described in Chapter 2, the 'waked' region of the wind farm using the height of the internal boundary layer resulting from the presence of the wind farm in the atmospheric boundary layer [11, 27, 110]. The fully developed 'waked' region begins when the height of the internal boundary layer reaches the maximum boundary layer height, represented by H. Defining the start of the boundary layer as the location of the freestream turbines, we vary the wake expansion coefficient linearly between the two values according to the weighting function

$$k_{w_i} = k_0 + \left(\frac{\delta_i - \delta_0}{H - \delta_0}\right)\left[k_{waked} - k_0\right], \tag{3.4}$$

where δ_0 represents the initial boundary layer height, and δ_i is the height of the internal boundary layer at Turbine i, modeled according to Eq. 34 of Ref. [22].

When the model is implemented, the values of k_0 and k_{waked} must be determined. For k_0, the definition from Eq. 3.1 is used, with $\alpha^* = 1$. The value of k_{waked} is more difficult to determine. In this work, we obtain this value from examining several runs of the ALC model, from Ch. 2, applied to the study's wind farm geometry and using the average value $\alpha-$value from the most 'waked' turbines, or the turbines located at the rear of the farm.

To form the state vector Φ_k, the individual deficit coefficients between each turbine pair are stacked in the form

$$\Phi_k = \begin{bmatrix}\phi_1^1 & \phi_1^2 & \phi_1^3 & \dots & \phi_1^N & \phi_2^1 & \dots & \phi_N^{N-1} & \phi_N^N\end{bmatrix}^T. \tag{3.5}$$

Since we model the interactions between each turbine pair individually, the system

has N^2 states, where N is the number of turbines. We use the following update model for the dynamics of the state vector

$$\Phi_{k+1} = A\ \Phi_k + E_k, \tag{3.6}$$

where in this case $A = I$. The input E_k represents the changes in system state due to the wind direction changes (events) in our framework. The precise form of E_k is detailed in 3.1.3.

The output of the system is the total wake deficit coefficient α, which represents the superposition of the individual deficits that affect a given turbine. The current output of the system is a function of the states at earlier times, which depends on edge-specific time delays that arise because of the finite time it takes information to travel between turbines. This information, which comprises quantities such as changes in wind direction or in the wake of a forward turbine, is contained in the flow field, and thus travels at the speed of the flow. In other words, this information is a function of the velocity of the flow field and the distance between two turbines. The time delay associated with information travel from Turbine j to i can be approximated as

$$\tau^i_{k,(i)} = \frac{D\ \Delta x^j_i}{u_j}, \tag{3.7}$$

where u_j is the local velocity at Turbine j, representing the speed that information will propagate to Turbine i. The diagonal of the matrix τ_k will be zeros since there is no delay of information when moving from a turbine to itself. The corresponding output equation is given by

$$\alpha_{k+1} = \Lambda(\tau_k)\Phi_k(\tau_k), \tag{3.8}$$

where $\Lambda(\tau_k)$ is a time-dependent adjacency matrix and is found from the subgraphs present at the delayed time. The quantity $\Phi_k(\tau_k)$ represents the states of the system as a function of the time delays, and describes the delay of information propagation (here the change in wind direction) through the system. The states are updated at

every time step, but the effect of that update does not reach the output until after the time delay. The architecture of this formulation results in a linear superposition of the wake velocity deficit coefficients. These coefficients are related to the velocity deficit through $\delta u_i = \alpha_i U_\infty$, which enables the disk velocity of the turbines to be found using

$$U_{d,k+1} = U_\infty \left(1 - \alpha_{k+1}\right) \left(1 - \frac{C'_T}{4 + C'_T}\right) \tag{3.9}$$

where C'_T is the local coefficient of thrust for the turbine.

The power from the wind farm is computed using the disk velocities calculated in Equation (3.9) as

$$P_k = \frac{1}{2}\rho \left(\frac{1}{4}\pi D^2\right) U^3_{d,k+1} C'_P \tag{3.10}$$

where ρ is the air density, $\pi D^2/4$ is the rotor disk area, and C'_P is the local coefficient of power. C'_P is sometimes assumed to be the same as the local coefficient of thrust C'_T (Betz limit), but in applications it is usually less due to losses.

3.1.3 Network Changes

Our time-varying graph is based on a log-file approach [105], where the static directions are the snapshots, and the changes between two snapshots - or two wind directions - are cataloged in time. The changes that occur in the graph are then implemented in an event-based framework, wherein an event is defined when the wind direction changes. The events are then associated with their application times t_k.

For a given event, the model has a current end-goal graph that represents what the final form of the graph will look like after the changes have been applied. When a subsequent event occurs, the final graph for the new wind direction is calculated and compared with the current end-goal graph. Changes between these two graphs are noted and sorted as a function of the time at which the event state reaches a given turbine. This time is computed as the streamwise distance between the turbine and the front of the farm divided by the freestream velocity. Here the front of the farm

is defined by the turbine at the front (i.e., the one that first experiences the inlet velocity associated with the new wind direction). Once the changes are defined and sorted, they are integrated into the existing event framework to be applied at the appropriate time in the simulation. The input for the system is then a function of the event parameters that are relevant to the current application time.

$$E_k = f(\Phi_k, \tau_k, \Delta\mathcal{E}_k) \tag{3.11}$$

Each input is a function of the new state values $\Phi_{e,i}$, the new time delay values $\tau_{e,i}$, and the list of the edge changes $\Delta\mathcal{E}_{e,i}$.

3.1.4 Results

The model is tested using the circular wind, shown in Figure 3-3, farm comprised of 38 5MW NREL reference turbines [93]. The wind direction reference is the inflow direction from the north at 0°. Measuring angles in a clockwise direction places the standard left-to-right inflow to the farm at 270°, as shown in Figure 3-3. Each turbine has a height $z_h = 90$ meters and a turbine diameter $D = 126$ meters.

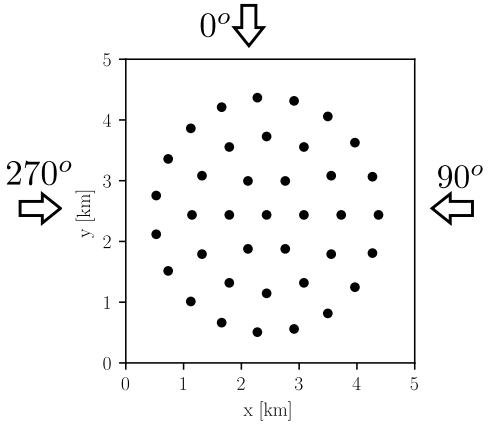

Figure 3-3. The circular wind farm configuration used to validate the graph model. The arrows and angles show the orientation of the wind direction.

The model is validated for both the steady state power prediction and the transient response of the wind farm during a wind direction change. The dynamic results are

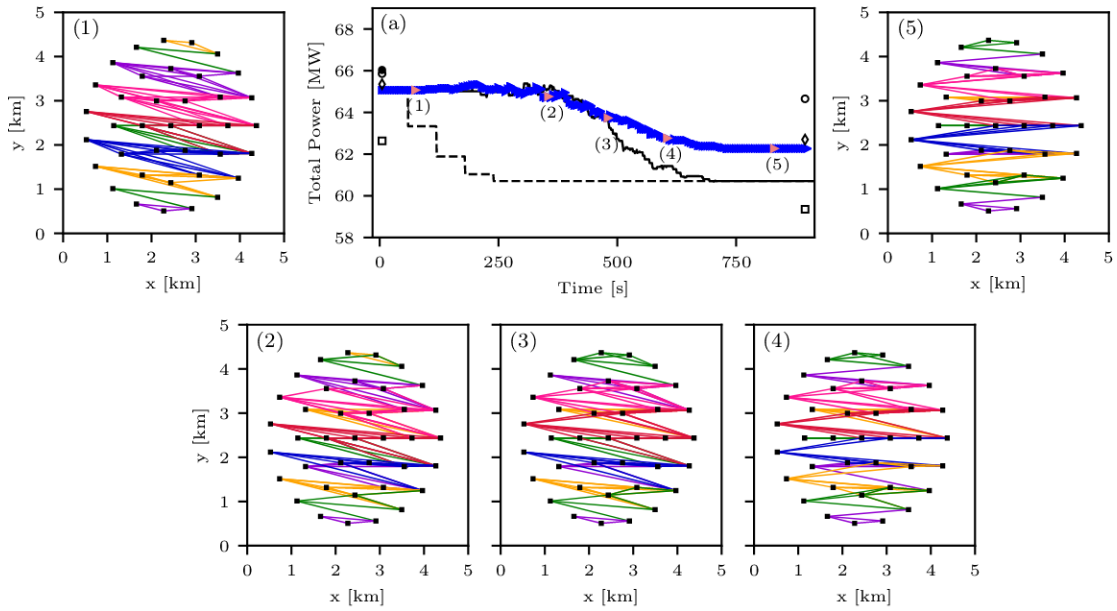

Figure 3-4. The central figure (a) shows the power of the graph model (▶) as a function of time as the wind direction changes from 280° to 270°, compared with the dynamic FLORIS simulation (-) and the static power from FLORIS (- -) over the same wind direction change. The model is compared to LES data at 280° (•), the Jensen/Park model (□), the coupled model from Shapiro et al. [22] (○), and the ALC model from Starke et al (♢) for both directions. The numbered figures correspond to the connections of the graph of the wind farm at different points in the wind direction change, which are shown by (▶) in (a). Each different color line in Figures (1)-(5) represents a separate subgraph and its connections, each of which have an independent lead turbine.

compared to modified dynamic FLOw Redirection and Induction in Steady State (FLORIS) simulations under changing inflow directions [111]. FLORIS is a dynamic model where direction changes are also incorporated using time delays. The outputs of FLORIS are tuned using heuristics, and therefore provide good agreement with LES but may not the the optimal choice for real-time control applications. The model was first tested on a direction change from 280° to 270°. The change was implemented at a rate of 2° per minute, for a total time of five minutes.

Figure 3-4(a) shows the power output of the farm as the wind direction change from 280° to 280°. The plot also includes steady-state power output predictions at the initial point from an LES simulation, using Simulator for Offshore Wind Farm

Applications (SOWFA) [88], and at both end points for the Jensen model [12, 24], the coupled model in Shapiro et al. [22], and the ALC model [110]. The dynamic response of the graph model in Figure 3-4(a) is compared to the dynamic FLORIS model implemented with the same direction change, which is represented by the solid line. The LES have a roughness height of $z_0 = 0.15$ meters and average inflow wind speed of $U_\infty = 8.07\ m/s$. The simulations use an actuator disk model for the turbines. The wake models use the same turbine parameters and U_∞ value. The friction velocity for the models is calculated using $u^* = U_\infty\ \ln(z_h/z_{0,lo})$. The values for coefficients of thrust and power are taken from the analysis in [110].

In the graph model, the values of $k_0 = 0.12$ and $k_{waked} = 0.2$ were computed from the average values from runs of the ALC model spanning 90° at 15° increments. The initial boundary layer height δ_0 is computed using a modified top-down model in the atmospheric boundary layer [27]. The maximum boundary layer height was set to 750 m, due to a temperature inversion at that height in the LES.

All the models and LES agree well at 280° except the Jensen model, but the coupled model from Shapiro et al. diverges at 270°. The Jensen model, while lower than the other models, perhaps because the way we define the wake expansion coefficient formulation, exhibits a drop in power from 280° to 270° that is similar in magnitude to that of the graph model, the ALC model, and FLORIS. Since the Jensen model is the only model without a data or physics-informed way to determine the wake expansion coefficient, the power output numbers may be less due to the the assumption of uniform wake behavior through the farm.

Considering the dynamic response, both of the models exhibit similar trends, with a ramp down to the 270° value. Though the steady-state FLORIS power for 270° is lower than that of the graph model, both models reach a steady state at very similar times, and the difference in the slope is a result of the variation in the final values. Modeling differences in the wake deficits likely account for the range of values for the

power at 270°. The black dashed line in the figure shows what the predicted farm power would be if the instantaneous wind direction change was assumed to be valid through the whole wind farm, (i.e., at each wind direction the steady state output power is computed from a static model). In this case, the steady-state prediction would show that all the effects of the change in wind direction are accounted for around the same time that the dynamic direction models start to show the effects of the wind direction change. This comparison illustrates that neglecting the dynamic response to the wind direction change can lead to inaccurate power output predictions.

Figure 3-4 also shows the graph connections at different points during the wind direction change. The wind direction change is evident in the graph when the initial graph in Figure 3-4(1) is compared to the final graph in Figure 3-4(5). In the initial graph, the connections between turbines are predominantly diagonal, while the connections in the final graph reflect the left-to-right inflow direction. The intermediate graphs, in Figure 3-4(2)-(4), show how the change of direction propagates through the farm.

The connections (or adjacency) matrix of the graph does not provide complete information for the graph model. We must also consider the intensity of the connections, represented by the weights of the edges. In this context, the weights indicate how much the wake of the upstream turbine in the edge affects the power of the following turbine. Figure 3-5(1)-(5), shows the weighted connections of the graph, where darker colors indicate a stronger connection. We can see the importance of the weights of the connections by noting that even though the graph for the farm with a 280° wind direction appears to have more connections, the connections are relatively weak across the farm. The connections in the graph with a 270° wind direction are much stronger, particularly for the aligned turbines in the center, resulting in a lower overall power.

We next apply the graph model to a larger change in wind direction, from 280° to 250°. The rate of change for the wind direction was kept at 2° per minute, resulting

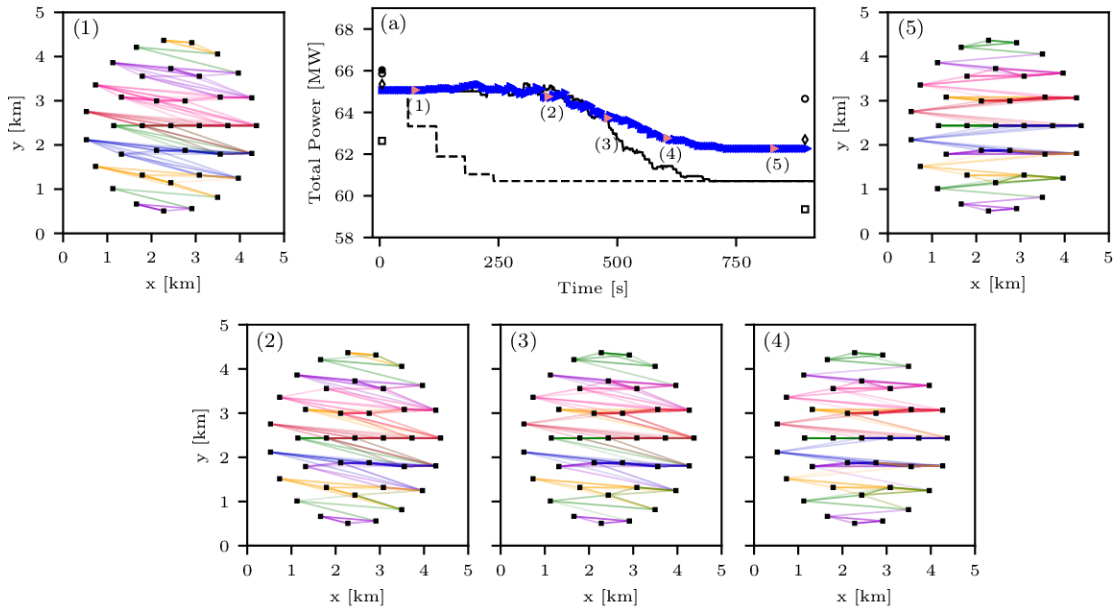

Figure 3-5. The central figure (a) shows the power of the graph model (▶) as a function of time as the wind direction changes from 280° to 270°, compared with the dynamic FLORIS simulation (-) and the static power from FLORIS (- -) over the same wind direction change. The model is compared to LES data at 280° (•), the Jensen/Park model (□), the coupled model from Shapiro et al. [22] (○), and the ALC model from Starke et al (♢) for both directions. The numbered figures correspond to the weighted connections of the graph of the wind farm, where darker colors indicate stronger connections, at different points in the wind direction change, which are shown by (▶) in (a). Each different color line in Figures (1)-(5) represents a separate subgraph and its connections, each of which have an independent lead turbine.

in a total change time of 15 minutes. Figure 3-6 shows the power versus time from the graph model during this change. Again, steady state results from the Jensen model [12, 24], the coupled model in Shapiro et al. [22], and the ALC model [110], as well as LES from SOWFA are plotted at the beginning and end points of the wind direction change. The steady results agree well with the LES data for the steady state power in both directions, and also with the model from Shapiro et al. [22], the ALC model, and the FLORIS model. The Jensen model is lower than all other points for the 280° case, but is much closer in the 250° case.

In the dynamic response, the graph model is compared with the results from

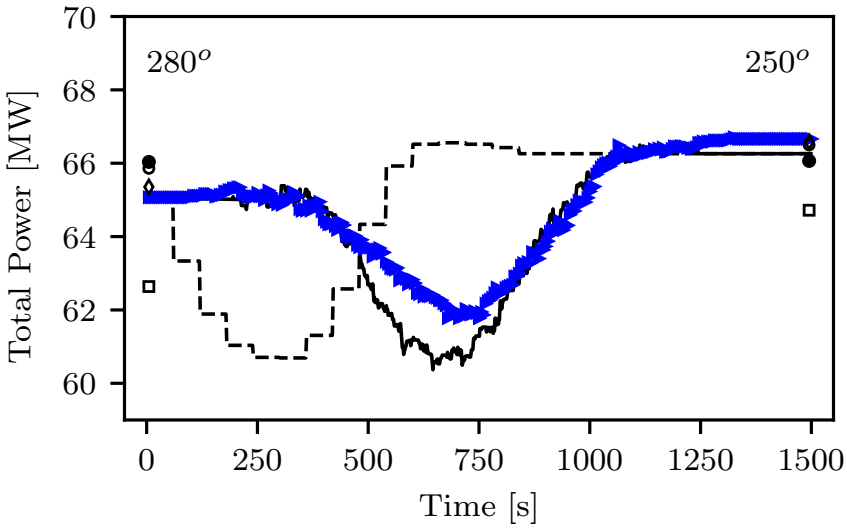

Figure 3-6. The power prediction of the graph model (▶) as the wind direction changes from 280° to 250° at a rate of 2° per minute, compared with the FLORIS simulation (-) and the static power from FLORIS (- -) for the same wind directions. The endpoints are compared with LES data (•), the Jensen model (□), the model in Shapiro et al.[22] (○), and the ALC Model from Starke et al. [110] (♢)

dynamic FLORIS, represented by the solid black line. We can see that the trends of the two models agree well, both exhibiting a dip in power before coming back to a similar value at the end of the simulation. This is a result of the more aligned nature of the farm at 270°, as seen in the previous example. The dynamic responses were again compared with the static step response, depicting the result if the wind direction is changed through the whole farm simultaneously, shown by the black dashed line. While the magnitude and the general trend of these values are similar, the static response reaches the trough and the final wind direction steady state much sooner than the dynamic response. This illustrates that using a static model for wind farm control under a wind direction change would give an inaccurate power prediction, rendering control of the system less effective. For implementation purposes, the graph model is significantly faster than the dynamic FLORIS model, with the entire simulation in Figure 3-6 taking between two and three minutes. This makes the graph model approximately 20 times faster than the dynamic FLORIS model, which is an

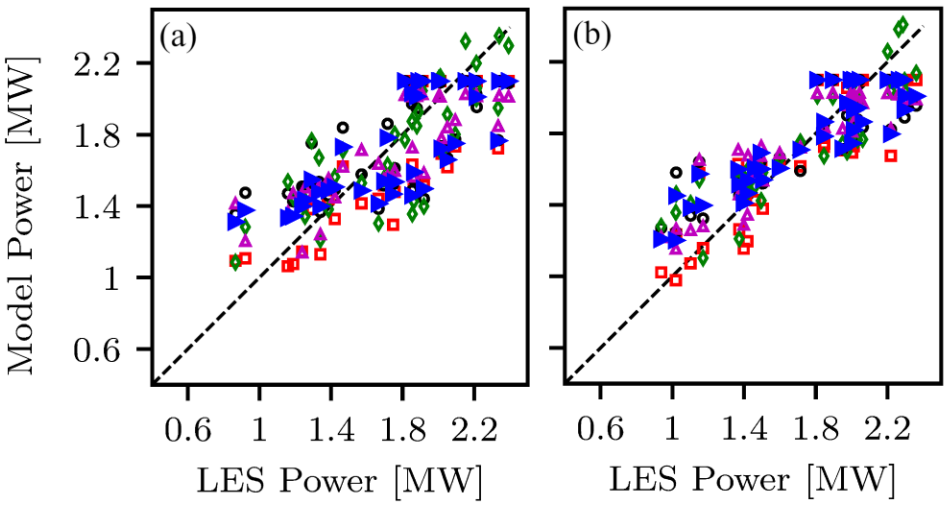

Figure 3-7. Scatter plots comparing the model-predicted power with the LES data for the LES wind directions (a) 280° and (b) 250°, for the Jensen model (□), the model in Shapiro et al.[22] (○), the ALC model in Starke et al.[110] (◇), the FLORIS model (△), and the graph model (▶). A 1:1 relationship is represented by a 45° line (- -).

important factor for applicability in control applications. This is partially the result of the Dynamic Graph model only resolving the connections between the turbines rather than the entire velocity field. This simplification comes with a significant increase in speed, but with a loss of some of the detail and resolution of the physics in the wind farm. In this control-oriented model, we have prioritized speed, and the spread of the power predictions is comparable with other wake model representations.

In addition to the average power of the farm, we can compare the individual power for each turbine. Figure 3-7(a) shows a scatter plot for 280° and Fig. 3-7(b) shows the scatter plot for 250°, where the individual power of the turbines predicted by the Jensen/Park model, the coupled model in [22], the ALC model, the dynamic FLORIS model, and the graph model, are plotted against the average individual turbine power from the LES. The scatter in the individual power prediction is similar for all of the models compared in Figure 3-7. The graph model thus provides a reasonable prediction of the power for each turbine, and gives an accurate estimation of the total power. The results in this section demonstrate that the Dynamic Graph model is

similar in accuracy to the more detailed wake models compared here.

3.2 Wind Turbine Yaw

The model outlined above accounts for changes in wind direction, but this framework can also be applied to dynamic yaw changes in a wind farm. When a turbine is yawed, it affects both the wake shape behind the turbine and also the power produced by the turbine. The direction of the wake deflection is also modified by the addition of yaw and this changes the connections between turbines in the graph model representation.

We now describe how the graph model in the previous subsection can be adapted to account for changes in turbine yaw. As established previously, the states of the graph model system are the normalized velocity deficit coefficients between each turbine pair. In order to determine these coefficients for the yaw case, we need a wake model that can represent the effect of yaw on the wind turbine wake. In this work we use the wake model for a yawing turbine developed by [49], which describes the deformation in the wake shape using an analytical function. We now present a short overview of this model which is explained more fully in [49].

The model is centered on an altered wake expansion function, which manifests in the wake deficit equation in the following way

$$\frac{\delta u}{U_\infty} = C(x) \exp\left[-\frac{(y-y_c)^2 + (z-z_h)^2}{2\sigma(x,\theta)^2}\right]. \tag{3.12}$$

Here, δu is the wake deficit, U_∞ is the freestream velocity, $C(x)$ is the initial velocity deficit, which is described later, y_c is the y-location of the center of the wake, z_h is the turbine height, and $\sigma(x,\theta)$ is the adjusted wake function, which is a function of both streamwise distance x and polar angle θ. The polar angle is defined as

$$\theta = \arctan\left(\frac{z-z_h}{y-y_c}\right). \tag{3.13}$$

The polar angle and the velocity deficit both depend on the streamwise position of

the wake center, which is defined as

$$y_c = \frac{(\pi-1)|\hat{t}|^3 + 2\sqrt{3}\pi^2\hat{t}^2 + 48(\pi-1)^2|\hat{t}|}{2\pi(\pi-1)\hat{t}^2 + 4\sqrt{3}\pi^2|\hat{t}| + 96(\pi-1)^2} sgn(\hat{t}) - \frac{2}{\pi}\frac{\hat{t}}{\left[(z+zh)/\tilde{\xi}_0\right]^2 - 1}, \quad (3.14)$$

where $\hat{t}$ is a dimensionless time quantity defined by the approximate analytical solution

$$\hat{t}(x,z) \approx -1.3634\frac{U_h}{u_*}\frac{R}{\tilde{\xi}_0}C_T\cos^2\gamma\sin\gamma\left[1-\exp\left(-0.4\frac{u_*}{U_{in}(z)}\frac{x}{R}\right)\right], \quad (3.15)$$

where R is the turbine radius, u_* is the friction velocity, and $U_in(z)$ and U_h are the z-dependent inflow profile and the inflow value at hub height ($z = z_h$), respectively. The expression

$$\tilde{\xi}_0 \approx R\sqrt{A_*} \quad (3.16)$$

is the initial wake shape based on the projected area of the yawed rotor with

$$A_* = \frac{1-a}{1-2a} = \frac{1+\sqrt{1-C_T\cos^2\gamma}}{2\sqrt{1-C_T\cos^2\gamma}}, \quad (3.17)$$

where a is the turbine induction factor and is related to the disk velocity u_d by $u_d = U_\infty\cos\gamma(1-a)$. The induction factor is affected by the yaw of the turbine and the relationship between the induction factor and the local thrust coefficient can be approximated [44] as

$$a = \frac{C_T'\cos^2(\gamma)}{4 + C_T'\cos^2(\gamma)}. \quad (3.18)$$

From this equation, the thrust coefficient can also be calculated based on the local thrust coefficient as

$$C_T = \frac{16C_T'}{(4 + C_T'\cos^2(\gamma))^2} \quad (3.19)$$

This is the coefficient of thrust used in the Eq. 3.17 above.

Once the wake streamwise centerline and the polar angle are defined, we can calculate the wake function as

$$\sigma(x,\theta) = k_w x + 0.4\xi(x,\theta), \quad (3.20)$$

where the equation is comprised of the standard linear wake expansion $k_w x$, where k_w is the wake expansion coefficient, and a second term to determine the evolution of the wake shape. The function ξ is a complicated function that depends on the streamwise distance, the polar angle, the current yaw angle, the dimensionless time, and the rotation rate of the turbine blades. The function exhibits different behaviors for different regions of the value of the dimensionless time, so our model uses the empirically fitted solution for this function that is derived in Appendix B of [49]. The equation consists of two parts:

$$\xi(\theta, x) = \xi_0 \hat{\xi}. \tag{3.21}$$

The first part consists of a an initial wake area defined by

$$\xi_0(\theta) = R\sqrt{A_*}\frac{|\cos\gamma|}{\sqrt{1-\sin^2\gamma\sin^2\theta}}, \tag{3.22}$$

for which a small γ approximation was given earlier in Eq. 3.16. The second part is the dimensionless shape of the vortex sheet

$$\begin{aligned}\hat{\xi}(\theta,\hat{t}) = 1 - \alpha_z \Bigg[&\frac{\hat{t}^2}{8\alpha_z}\cos(2\theta) + \left(\frac{-\hat{t}^3}{24\alpha_z}\chi\sin(2\theta) + \frac{-\hat{t}^3}{32\alpha_z}\cos(3\theta)\right) \\ &+ \left(\frac{-1}{6}\frac{\hat{t}^4}{16\alpha_z}\chi^2\cos(2\theta) + \frac{5}{16}\frac{\hat{t}^4}{16\alpha_z}\chi\sin(3\theta) + \frac{-5}{48}\frac{\hat{t}^4}{16\alpha_z}\cos(2\theta) + \frac{7}{48}\frac{\hat{t}^4}{16\alpha_z}\cos(4\theta)\right)\Bigg]\end{aligned} \tag{3.23}$$

where $\alpha_z = 1.263\cos(0.33\chi)$, and $\chi = 1/(\lambda\sin\gamma)$ is the rotation rate, which defined as the ratio of the strength of the vortex generation due to the rotation blades to the one generated due to the yaw offset and depends on the tip speed ratio λ.

Finally, to fully define this model, the initial wake deficit $C(x)$ is calculated using the following equation

$$C(x) = 1 - \sqrt{1 - \frac{C_T\cos^3(\gamma)}{2\tilde{\sigma}^2(x)/R^2}} \tag{3.24}$$

where $\tilde{\sigma}^2$ uses a simplified version of Eq. 3.20. The simplified version avoids a dependence on θ and is defined as

$$\tilde{\sigma}^2(x) = (k_w x + 0.4\tilde{\xi}_0)(k_w x + 0.4\tilde{\xi}_0\cos\gamma), \tag{3.25}$$

which uses $\tilde{\xi}_0$ from Eq. 3.16 which is defined earlier, and whose asymmetry is used as a way to approximate the effect of yaw on the velocity deficit.

Using the wake model formulation from [49] above, we can calculate the deficit coefficient for the graph model. This process applies this wake deficit model to each turbine pair, and then takes the average of the wake deficit from the j^{th} turbine over the disk area of the i^{th} turbine, represented by the expression

$$\phi_i^j = \frac{1}{A_{i^{th}disk}} \int_{i^{th}disk} C(\Delta x_{i,j}) \exp\left[-\frac{(y - y_{c,j})^2 + (z - z_{h,j})^2}{2\sigma(\Delta x_{i,j}, \theta_j)^2} \right] dy\ dz \tag{3.26}$$

The states of the model are thus the normalized velocity deficit coefficients calculated between each turbine pair in the wind farm. To form the state vector Φ_k, these individual deficit coefficients are stacked in the form

$$\Phi_k = \begin{bmatrix} \phi_1^1 & \phi_1^2 & \phi_1^3 & \dots & \phi_1^N & \phi_2^1 & \dots & \phi_N^{N-1} & \phi_N^N \end{bmatrix}^T, \tag{3.27}$$

resulting in a system with a total of N^2 states.

After the individual velocity deficits have been calculated, they are then linearly superposed using the time-dependent adjacency matrix from the graph to determine the appropriate wake connections. The disk velocities at each turbine are then found by multiplying the superposed velocity deficit coefficients by the the streamwise component of the velocity, since this is the component that produces power,

$$u_{d,k} = (1 - \alpha_k) U_\infty cos(\gamma_k)(1 - a). \tag{3.28}$$

In the disk velocity equation, the induction factor used is that defined in Eq. 3.18. The disk velocity can now be used to calculate the power for the wind turbine using

$$P_k = \frac{1}{2} \pi R^2 \rho (u_{d,k})^3 C_P cos(\gamma)^p, \tag{3.29}$$

where for our case we set $C_P = C_T$ and the exponent p determines the effect of yaw on the power and changes for different turbines. The range for p is generally $0.5 \leq p \leq 2$.

3.2.1 Time-Delay Formulation

The time-delay between turbines is characterized by $\tau = \Delta x/u$, where $\Delta x = s_{nx,j} - s_{nx,i}$ is the distance between the two turbines and u is the velocity of the wind. However, the velocity of the wind varies between the turbines as a result of the wake behind the first turbine. In the previous formulation of the model, the freestream velocity was used to find the time-delay. Our studies using this model showed that this underestimates the time delay (since the velocity is slower than the freestream velocity due to the wake). To improve upon the initial formulation, we derive an analytical result for the average velocity based on the velocity deficit. We know from 3.12 the expression for the velocity deficit. For the wake velocity, we use the velocity on the centerline of the wake, where $y = y_c$ and $z = z_h$. This simplifies the velocity deficit equation to

$$\frac{\delta u}{U_\infty} = C(x). \tag{3.30}$$

where $C(x)$ is defined by 3.24. We know that the velocity of the flow is found by subtracting the velocity deficit from the freestream velocity: $u = U_\infty - \delta u$. We can then find a more accurate time-delay by integrating the velocity over the distance between the turbines

$$\tilde{\tau} = \int_{s_{nx,i}}^{s_{nx,j}} \frac{1}{U_\infty(1 - C(x))} dx \tag{3.31}$$

Since most wake models are more accurate in the far field rather than immediately after the turbine, we adjust the integral to start one diameter downstream of the forward turbine. Also substituting the previous relations in for $C(x)$ gives the following expression:

$$\tilde{\tau} = \int_{s_{nx,i}+D}^{s_{nx,j}} \frac{1}{U_\infty \left[\sqrt{1 - \frac{C_T \cos^3(\gamma)}{(2/R^2)(k_w x + 0.4\tilde{\xi_0})(k_w x + 0.4\tilde{\xi_0}\cos\gamma)}}\right]} dx \tag{3.32}$$

We now apply a Taylor series expansion to the integrand to simplify the expression. If we set

$$y = \frac{C_T \cos^3(\gamma)}{(2/R^2)(k_w x + 0.4\tilde{\xi_0})(k_w x + 0.4\tilde{\xi_0}\cos\gamma)}, \tag{3.33}$$

we can find the Taylor series expansion for $1/\sqrt{1-y}$, which is

$$\frac{1}{\sqrt{1-y}} \approx 1 + \frac{y}{2} + \frac{3y^2}{8} + \frac{5y^3}{16} + \mathcal{O}(y^4) \tag{3.34}$$

Using the first order approximation, we an write the integral as

$$\begin{aligned} \tilde{\tau} &= \int_{s_{nx,i}+D}^{s_{nx,j}} \frac{1}{U_\infty} \left(1 + \frac{y}{2}\right) dx \\ &= \int_{s_{nx,i}+D}^{s_{nx,j}} \frac{1}{U_\infty} \left(1 + \frac{1}{2} \frac{C_T \cos^3(\gamma)}{(2/R^2)(k_w x + 0.4\tilde{\xi_0})(k_w x + 0.4\tilde{\xi_0}\cos\gamma)}\right) dx \end{aligned} \tag{3.35}$$

Performing the integration gives the following expression for the time delay

$$\begin{aligned} \tilde{\tau} = \frac{1}{U_\infty} \Bigg[&\Delta x - D + \frac{C_T R^2 \cos^3\gamma}{1.6\tilde{\xi_0} k_w (1-\cos\gamma)} \\ &\left(\ln \frac{k_w s_{nx,j} + 0.4\tilde{\xi_0}}{k_w s_{nx,j} + 0.4\tilde{\xi_0}\cos\gamma} \ln \frac{k_w(s_{nx,i}+D) + 0.4\tilde{\xi_0}}{k_w(s_{nx,i}+D) + 0.4\tilde{\xi_0}\cos\gamma}\right) \Bigg]. \end{aligned} \tag{3.36}$$

As can be seen from this expression, in the third term of the integral there is a singularity when $\gamma = 0$. This term will tend to infinity due to the $(1 - \cos(\gamma))$ term in the denominator. However, the natural logarithm term will tend to zero as this term becomes $\ln(1)$ when $\gamma = 0$. Numerically, this was dealt with by adding a very small value of the same sign as the cosine term to keep the expression $(1 - \cos(\gamma))$ from equalling zero exactly. The logarithmic term seems to dominate the expression as the yaw goes to zero. Figure 3-8(a) shows the integrated quantity of the time delay over the range of yaw angles from $\gamma = -20°$ to $\gamma = 20°$. Figure 3-8(b) shows the average time-delay values found for this range. The average time-delay found for $\gamma = 0$ is a finite value, is not an outlier when compared to the rest of the values and gives the time delay we would expect for zero yaw.

3.2.2 Validation of the model

The model was first tested in a static capacity by comparison to LES. The setup has two turbines that are aligned with $7D$ between them in the streamwise direction. The first turbine is then set to various yaw positions between $-30°$ and $30°$ yaw and the averaged power results are compared to the model results.

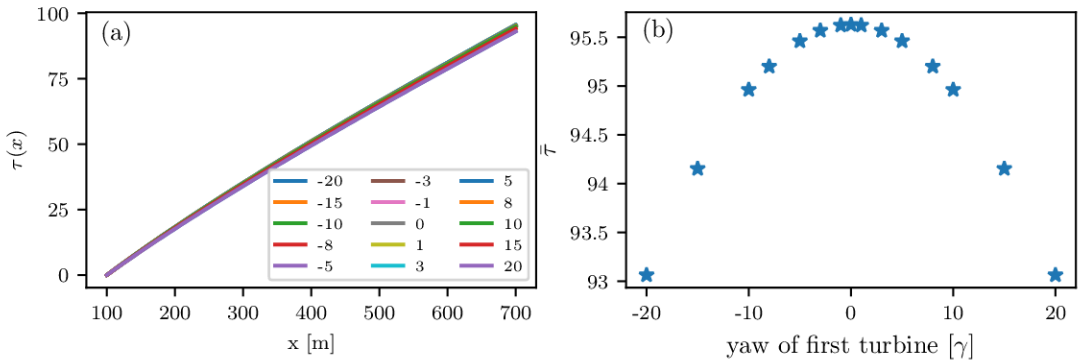

Figure 3-8. (a) The integration of the time delay quantity as a function of streamwise distance x and (b) the calculated time delays between two turbines as a function of the yaw of the first turbine

The LES is performed with the JHU LESGO code [91] with an actuator disk model. As the inflow, the simulation used a field generated by the concurrent-precursor approach [99]. The turbines have a diameter and hub height of $D = z_h =$ 100 meters and a ground surface roughness height of $z_{0,lo} = 0.1$ meters is used to prescribe the bottom boundary condition of the LES. The main simulation domain uses $(N_x \times N_y \times N_Z = 256 \times 256 \times 192)$ grid-points for a domain size of 3 $km \times 3$ $km \times 1$ km. The turbine (local) coefficient of thrust is kept constant at $C_T' = 1.33$ throughout the simulation.

Figure 3-9 shows the comparison of the LES study with the graph model results, where the results are normalized by the power of the first turbine at zero yaw. The graph model result using the wake model for a yawed wind turbine from [49], called the curled wake formulation, is represented by the red symbols and the graph model result using the Gaussian wake formulation from Section 3.1 is represented by the blue symbols. We can see that, even with the matching of the deficit when $\gamma = 0$, the Gaussian formulation leads to a power output curve at the second turbine that is too shallow compared to the LES. The curled wake does a better job of capturing the behavior of the wake of the first turbine which determines the behavior of the second turbine.

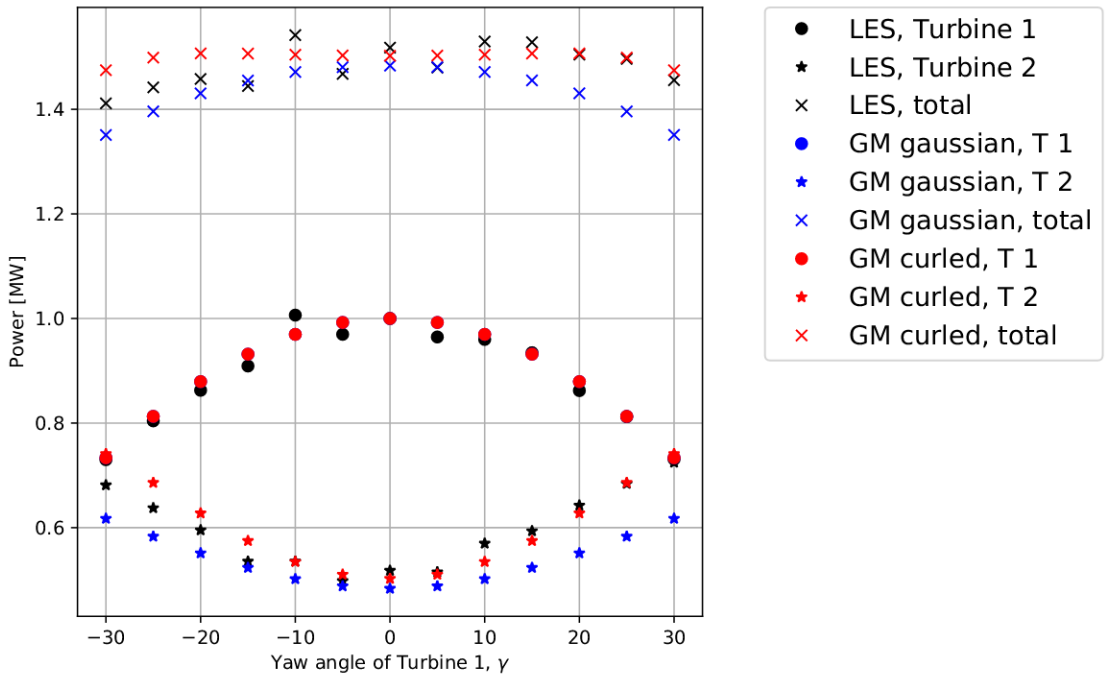

Figure 3-9. This plot compares the results from the LES study with the graph model using a Gaussian formulation for the velocity deficit and the graph model using the curled wake shape formulation for the velocity deficit.

The yaw formulation for the Graph Model was also tested in a dynamic setting using the results from an LES performed using the JHU LESGO code of a dynamically yawing wind turbine pair. In the simulation, the two turbines are placed in one column and the first turbine is dynamically yawed. The results are averaged over 120 realizations to smooth out the instantaneous turbulence fluctuations. The LES results are portrayed in Fig. 3-10, which shows the power from the first turbine and the second turbine as a function of time during the dynamic yaw. The model result, which is portrayed in blue, applies the change in discrete steps, which can be seen in the power of the first turbine. In this test, the model is able to match both the timing of the yaw change, as in how long it takes the change from the first turbine to affect the second turbine, and the magnitude, which is how much the power of the second turbine increases as a result of the yawing of the first turbine.

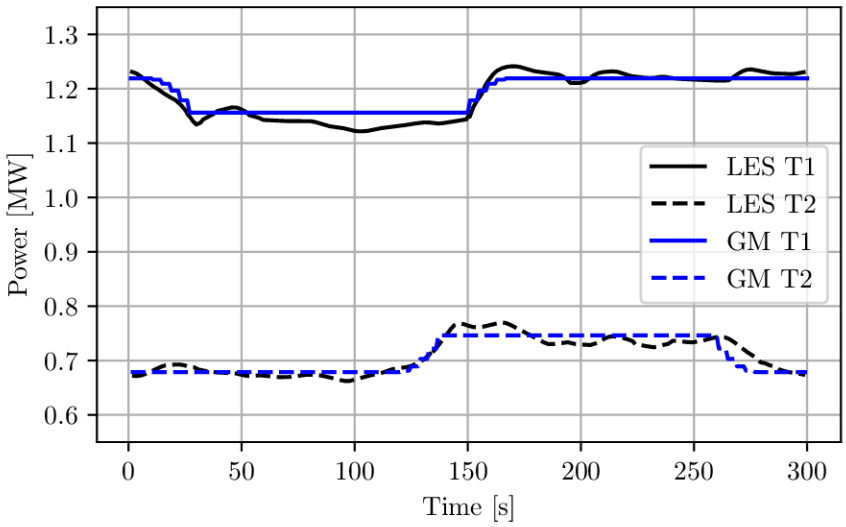

Figure 3-10. Comparison of the Graph Model to a dynamic yaw case from LES

3.3 Conclusions

Operational wind farms are routinely subjected to changing wind inlet conditions that need to be taken into account in algorithms that aim to regulate wind farm power output. These changing conditions are difficult to represent using most analytical wake models. Additionally, as yaw control becomes more common, wind farms must also account for transient responses to dynamic yaw actions. This chapter introduced an efficient analytical wake model based on a graph theory approach which is able to represent a wind farm under dynamically changing conditions such as wind direction changes and dynamic yaw changes. The model is an estimation algorithm that represents the interactions between turbines as directed edges in a graph structure, which enables one to capture the propagation of the changes in these interactions and conditions through the wind farm. The Dynamic Graph model is the first graph-based analytical wind farm model, as well as one of the first dynamic yaw models, that has been implemented and validated. It also implemented the yaw using a new analytical formulation of a yawed wind turbine wake, which is able to capture the curling of a yawed wake as it moves downstream [49] The model was applied in two applications:

dynamic wind direction changes and dynamic yaw changes. First, it was applied to a nonuniform wind farm to demonstrate tracking a transient response of the farm to a change in wind direction. The model results are comparable with LES and wake models for a steady wind direction. More importantly, the model performed more realistically than static models over a wind direction sweep, when validated against an unsteady RANS model. The yaw formulation of the model was also validated with LES, both statically and dynamically. The static test showed good agreement with LES for varying yaw angles. The dynamic test was performed over 120 realizations of an LES of a dynamically yawing turbine pair. The model was able to capture both the magnitude and timing of the change in power due to the dynamic yaw. The structure and speed of the model lends itself to integration into real-time, horizon-based, control strategies, which is explored in the remaining chapters.

Chapter 4

Error Correction and State Estimation in Wake Modeling

The wake models described in the previous chapters provide good representations of some of the key dynamics in the wind farm that affect the power prediction, such as the wake deficit and advection and dissipation of the wake. However, they must necessarily neglect and simplify many aspects of the physical processes affecting the wind farm in order to reduce the order of the model and the computation time. Notably, the model has no knowledge of the variations in the power output due to turbulence in the flow, which is related to the conditions of the incoming velocity, and changing conditions that can affect the wake expansion coefficient, k_w, which determines how the wake expands as it advects downstream.

Wake models can be improved by incorporating real-time measurements taken from the wind farm. In this chapter, we outline various ways of incorporating these measurements and their effects on the wind farm power output. Here, we focus on using only the power measurements available from operating turbines in a wind farm. In Section 4.1, we show how a nonuniform inflow for a wake model can be derived from these measurements. Then, in Section 4.2, we use these measurements to update the states of a wake model to reduce modeling error using an ensemble Kalman filter.

4.1 Dynamic Inflow in Wake Modeling

One of the most straightforward ways to improve the prediction of the wake model is to enable a nonuniform inflow as an inlet condition. Since wind farms can span a distance of kilometers, the inflow conditions can change significantly across the inflow area. This is illustrated in Fig. 4-1, which shows an average velocity field from an LES of a wind farm. The wind farm depicted in Fig. 4-1 is 6 kilometers in the spanwise direction, which is not an unusual wind farm size, and the black box in the figure shows the inlet condition of the farm. Clearly, the inlet condition varies quite a bit over the wind farm span, and this can be exacerbated in operating wind farms by geological features upstream of the farm. By including these variations in the model, we can better capture local conditions in the flow.

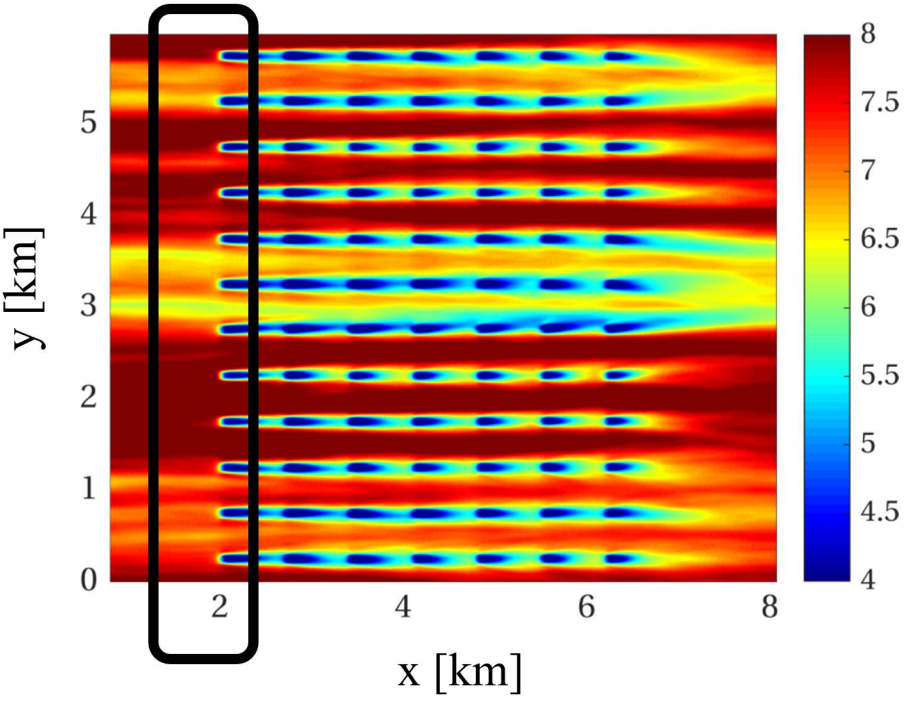

Figure 4-1. The average velocity field of an eighty four-turbine farm showing the variation of the average velocity seen at the first row of the farm (denoted by the black box).

Instantaneous power measurements at each turbine can be filtered using the

equations for the disk velocity and power from a turbine to find the upstream velocity

$$U_{\infty,n} = \left(\frac{2P_n}{\rho A C'_{P,n}}\right)^{1/3} \left(\frac{4 + C'_{T,n}}{C'_{T,n}}\right). \tag{4.1}$$

This expression is only valid for the freestream turbines in the farm. The turbine wakes of upstream turbines distort the velocity measurements of the waked turbines which prevents the waked turbine power data from providing an accurate measure of the freestream velocity. The freestream turbines are either determined a priori by the wind direction, or computed dynamically if the wind direction is unsteady. An example method for identifying these turbines was shown in Chapter 3.

4.1.1 Incorporation with LES

Since this method uses the instantaneous power measurements from the turbines, it can be applied in real-time to update the inflow conditions for a given model. In our implementation, the model is run simultaneously with the wind farm plant (LES in our case) and the inflow is updated dynamically using the instantaneous power measurements from the turbines. This procedure is illustrated using a time-dependent version of the Jensen/Park model [12] to represent the 84 turbine farm, whose geometry is shown in Fig. 4-1.

The evolution of the velocity field for this case is shown in Fig. 4-2, where the velocity profile in the spanwise direction is plotted against time. We can see that the velocity profile is initialized with a constant value and then updates according to the power measurements. In this case, the change in the freestream velocity takes the form

$$\frac{dU_{\infty,n}}{dt} = \frac{1}{\tau_{fil}} \left(\frac{4 + C'_{T,n}(t)}{C'_{T,n}(t)} \left(\frac{8P_n(t)}{\rho \pi D^2 C'_{T,n}(t)}\right)^{1/3} - U_{\infty,n}(t)\right), \tag{4.2}$$

for the freestream turbines (shown in the black box in Fig. 4-1) where τ_{fil} is the time constant that determines how quickly the inflow velocity profile is allowed to change, and the local thrust coefficient is assumed to be the same as the local power coefficient.

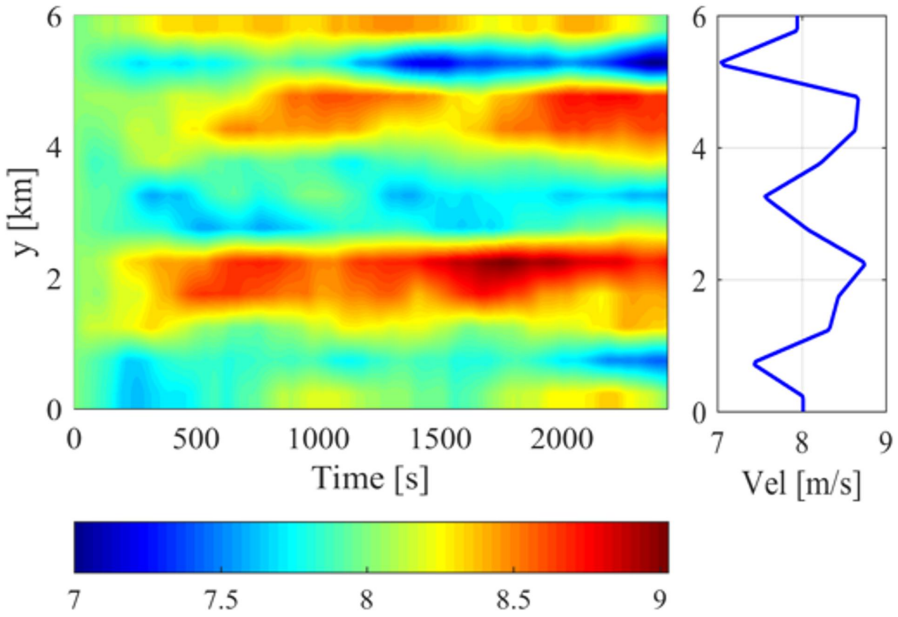

Figure 4-2. The evolution of the inflow velocity field filtered from the freestream turbines.

The ending profile in Fig. 4-2, shown on the right, is able to reflect the large scale trends of the inflow profile shown in the LES in Fig. 4-1, which means that it is able to incorporate the time and space dependent changes in the inflow to the wind farm.

Figure 4-3 shows a comparison between an instantaneous snapshot of the LES velocity field compared with the wake model of the flow. Since the wake model is aimed at capturing the time averaged behavior of the farm along with key flow aspects, it does not match an instantaneous snapshot exactly. However, it is able to reproduce the faster and slow streaks in the inflow that can be seen in the LES inflow (though in a more turbulent sense) and also reflects the general placement of wakes and wake interactions in the farm.

While the wake model is able to dynamically update its inflow based on the power measurements from the LES, there remain a few challenges with this method. The main challenge is the fact that while the spanwise velocity is heterogeneous in this case, the streamwise velocity is constant along each spanwise location. Applying spacial

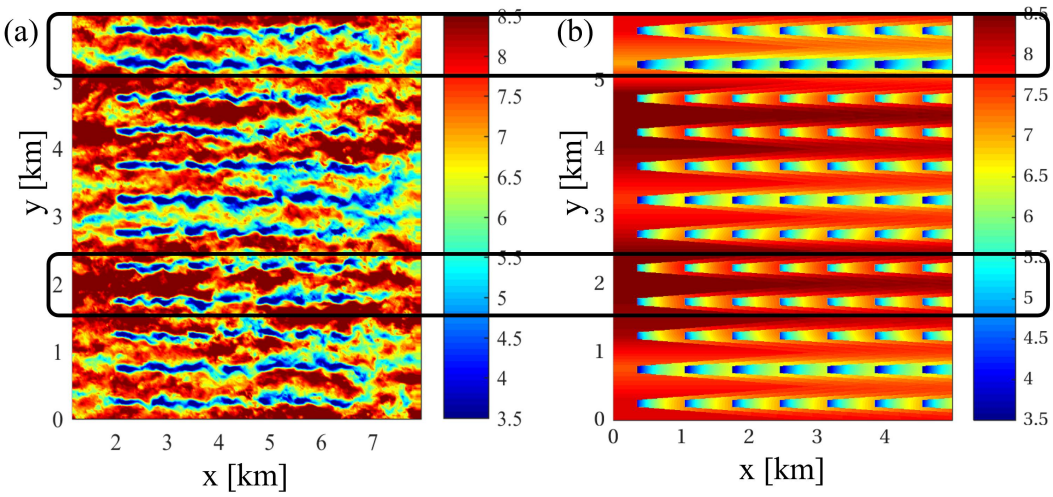

Figure 4-3. A comparison between (a) an instantaneous velocity snapshot of the LES simulation and (b) the wake model with the span-dependent inflow velocity based on the LES.

variance in both the spanwise and the streamwise directions is an area of current research in flow field-resolving wake models [112], but the majority of wake models currently are only equipped for spanwise heterogeneity. This presents the challenge that the inflow profile changes that occur at the front of the farm will be applied instantaneously to the whole farm. This is problematic because the previous inlet conditions are still progressing through the farm with a finite travel time. The fact that the wake model is trying capture the average effect of the inflow over the whole farm is why the time constant in Eq. 4.2 is important. It determines how quickly the inflow should change, and how that change will effect the total power output of the entire farm.

The Dynamic Graph model, due to its structure, is better equipped to represent these variations throughout the farm. Since the Dynamic Graph model relies on the pairwise interconnections between the turbines and does not resolve a complete velocity field, a changing inflow can be advanced through the farm in the same way as the varying wake based connections between the turbines. The Dynamic Graph model can then be implemented alongside LES and use instantaneous power measurements,

from Eq. 4.1, to determine an inflow that is updated with each measurement and then propagated through the farm.

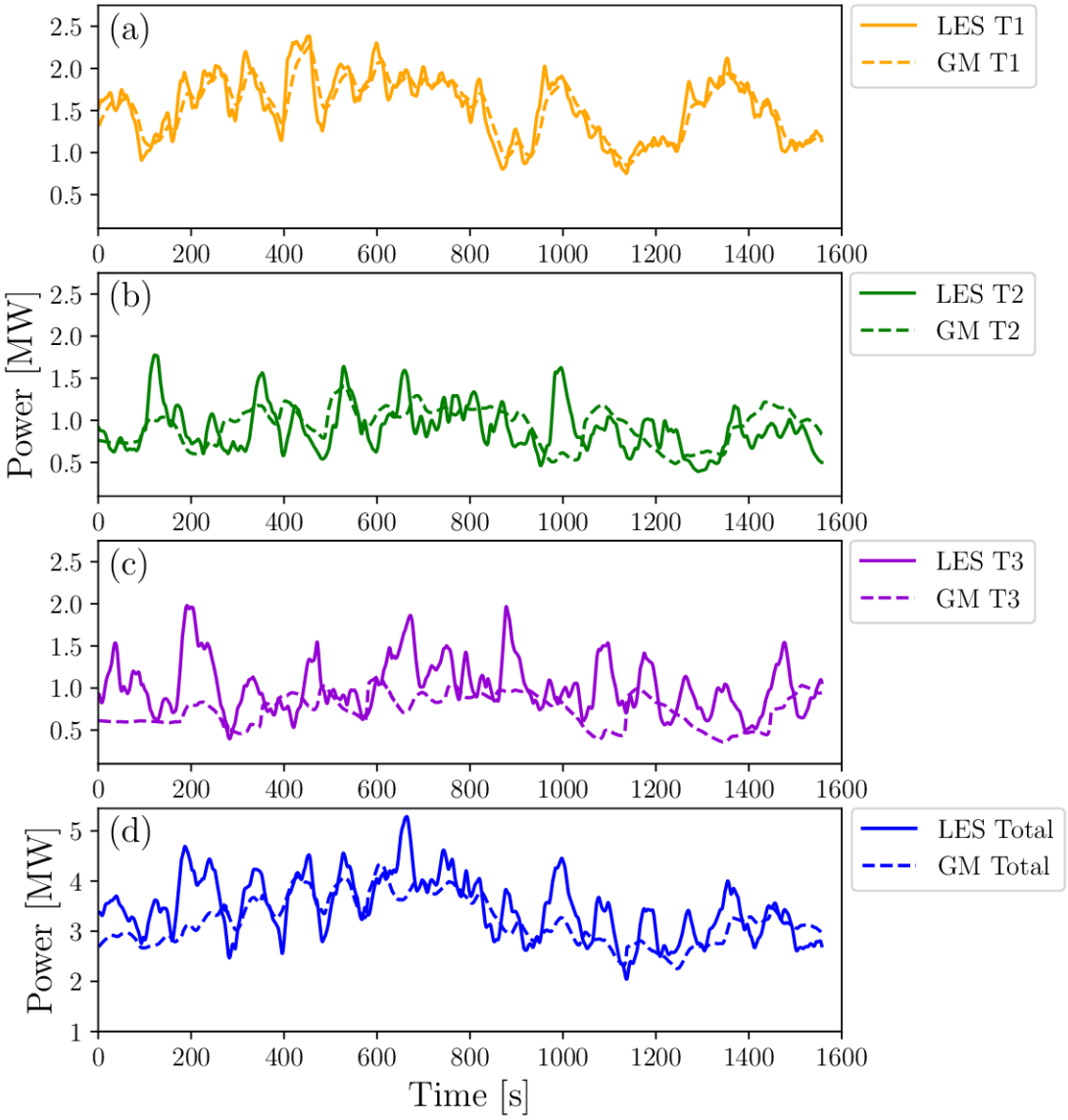

Figure 4-4. A comparison between power from an LES compared with a Dynamic Graph model power output using the same inflow for three turbines which are in an aligned configuration, where (a) is power from the freestream turbine in the column, (b) is the power from the second turbine and (c) is the power from the third turbine. The total power for the column from the LES and the Dynamic Graph model is shown in panel (d).

Figure 4-4 shows a comparison between a timeseries of the power from an LES and the corresponding power prediction from the Dynamic Graph model for a configuration

where three turbines are aligned in a single column according to the incoming wind direction. The figure shows the power from the first turbine in the column, the freestream turbine, in Fig. 4-4(a), where we can see that the Dynamic Graph model follows the LES power very well. This makes sense as the Dynamic Graph model is using the filtered velocity for the first turbine. Then Fig. 4-4(b) and (c) show the time series for the second turbine and the third turbine, respectively. In these cases, the model does not match as well, especially on the peaks and turbulent fluctuations, where more energy is added to the flow from wake meandering, for example. Here, the model does not have enough information from the inflow to be able to reproduce these features of the flow. Even so, the model is able to follow the general trend of the second and third turbines. One of the artifacts of this model is that it takes the inter-turbine travel time from the first turbine for the second and third turbines to pick up on the changes in the inflow after they are initialized (which is most obvious in the inital constant value of the third turbine). The second point of note is that the variations of the inflow, such as the large troughs and peaks around $800 - 1200$ seconds in Fig. 4-4(a) clearly propagated to the second and third turbines, see the trough at 1000 seconds in Fig. 4-4(b), which shows that this model is able to represent heterogeneous flow in both the spanwise and streamwise directions due to its inherent structure.

Figure 4-4(d) shows a comparison between the total power from the column of turbines from the LES and the Dynamic Graph model. As mentioned above, the Dynamic Graph model misses the larger turbulent fluctuations in the power. However, it is able to track the average power output of the wind farm, and looks like a smoothed-out version of the LES signal. The Dynamic Graph model here exhibits a lower variability over time when compared with the LES. Since this model is intended for control-oriented applications, this artifact could affect the control actions selected using this model. The strength of this model is its ability to represent changing

conditions in the wind farm. These conditions, such as a wind direction change, are also typically on a slower timescale, as they can take possibly 10s of minutes to propagate fully through the farm. Since the events that this model is designed to represent are also changes that are typically slower and affect the bulk power more than the turbulent variations, this model is a good candidate to inform the control decisions in these cases. However, since the model does exhibit less variability than the LES, it may not be a good candidate for determining pitch control, in its current form. However, applying a more advanced estimation technique could also improve the performance and responsiveness of the graph model.

4.2 The Ensemble Kalman Filter

In addition to using a nonuniform inflow, we can also adjust the parameters of a wake model dynamically to improve the power output estimate. In order to find the update for the parameters based on the measurements, we employ the Ensemble Kalman filter (EnKF). The EnKF is an extension of a traditional Kalman filter and is used in situations with large parameter spaces. Instead of calculating the covariance matrix, the EnKF extracts the error information from an ensemble of perturbed models, whose average serves as the 'true' state of the system [62].

The following outlines a brief description of the EnKF and it's application to a wake model. This estimation method is applied to the types of dynamic wake models developed in previous work [22], and not to the wake and wind farm models presented in detail in this thesis. More details of the EnKF and its application to this problem, which is published in [22], can be found in Appendix A. In this application, the EnKF is used to reduce the error in the velocity deficits δu_n (the states of the model) and the wake expansion coefficients $k_{w,n}$ (the parameters of the model) based on power measurements at each turbine. In particular, we extend the approach of Shapiro *et al.* [65] to incorporate the modified dynamic model presented in [22]. This extension

allows the estimation method to be applied to arbitrary turbine configurations, and include spanwise variations in inflow velocities. The previous approach of Shapiro *et al.* [65] was limited to regular wind farm configurations in which all rows were averaged and the farm could be treated as a single column of representative turbines.

The EnKF assumes that the resulting modeled wind farm system is governed by discrete update equations of the form

$$\boldsymbol{\psi}_{k+1} = \mathbf{f}(\boldsymbol{\psi}_k, \mathbf{C}'_{Tk}) + \mathbf{B}\boldsymbol{\chi}_k, \tag{4.3}$$

$$\boldsymbol{\xi}_k = \mathbf{h}(\boldsymbol{\psi}_k, \mathbf{C}'_{Tk}) + \boldsymbol{\epsilon}_k \tag{4.4}$$

where $\boldsymbol{\psi}_{k+1} = \mathbf{f}(\boldsymbol{\psi}_k, \mathbf{C}'_{Tk})$ and $\boldsymbol{\xi}_k = \mathbf{h}(\boldsymbol{\psi}_k, \mathbf{C}'_{Tk})$ are temporal and spatial discretizations of the wake model, $\boldsymbol{\psi}_k$ is the vector of the model states, which are the velocity deficits and the wake expansion coefficients of the turbines, $\mathbf{C}'_{Tk}$ is a vector of the local thrust coefficients at each turbine, and the function $\mathbf{h}(\boldsymbol{\psi}_k, \mathbf{C}'_{Tk})$ maps the states to the measured quantities (the power at each turbine). Measurement and modeling errors are represented by the zero-mean white noise processes $\boldsymbol{\epsilon}$ and $\boldsymbol{\chi}$, respectively. The variance of the measurement, state, and parameter noise must be prescribed. In the examples shown, they are specified as $\sigma_{\delta u} = 0.1$ m/s, $\sigma_k = 0.00025$, and $\sigma_P = 4.83$ kW.

An ensemble of 256 wake model implementations, governed by (4.3) and (4.4), is used to estimate the error covariance of the wake model based on the measurements. The state and parameter estimates are computed in two steps. First, the wake model is used to perform an intermediate forecast of the wake model states. Then, power measurements are used in conjunction with the ensemble estimates in the measurement analysis step to produce states and parameter estimates. Further details of this method can be found in [65].

The estimation algorithm is tested using an LES of a 38-turbine wind farm composed of four staggered rows of six turbines followed by 14 irregularly arranged

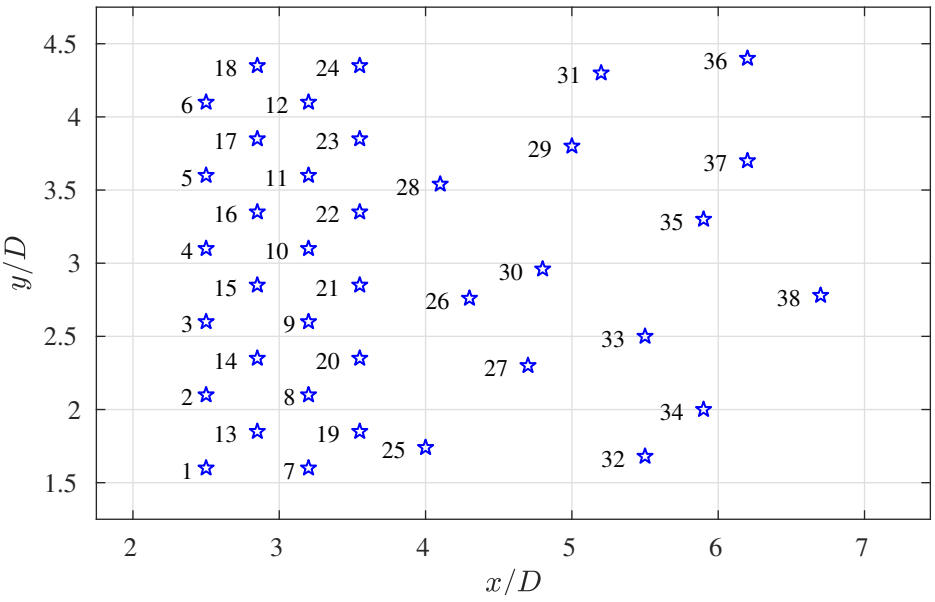

Figure 4-5. Locations of 38-turbine wind farm simulated in LES.

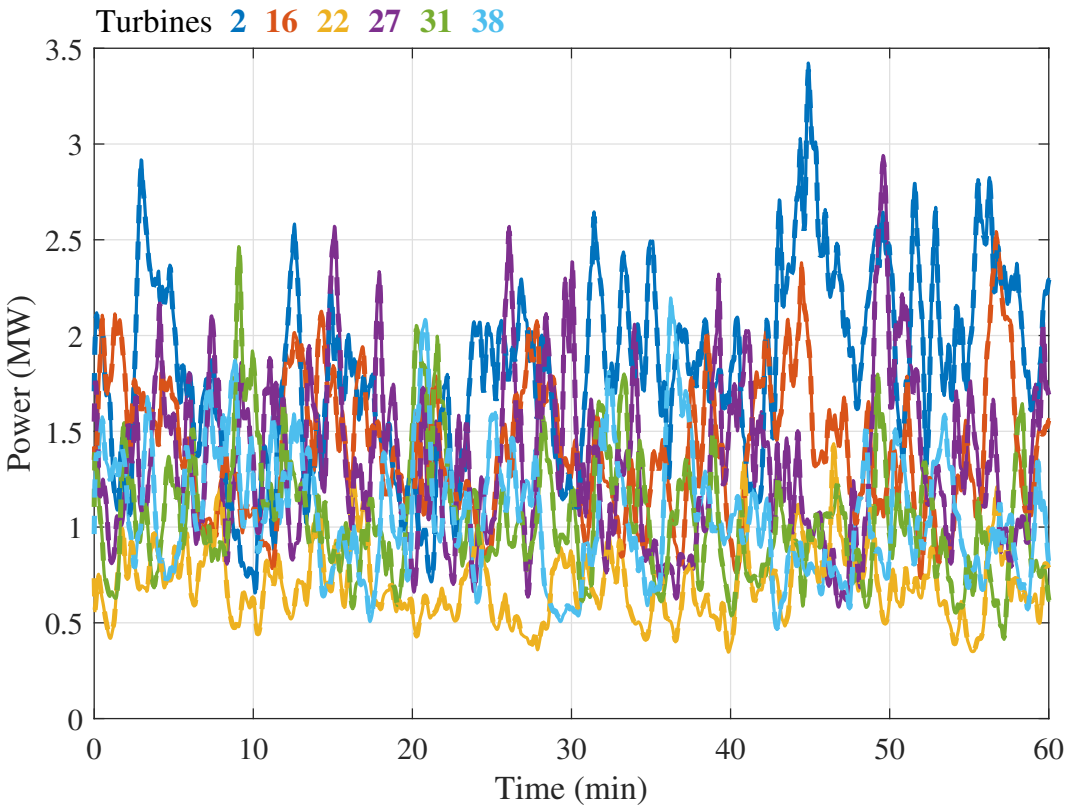

Figure 4-6. Measured (——) and estimated power production (– – –) for 38-turbine wind farm. The turbine locations are shown in Figure 4-5.

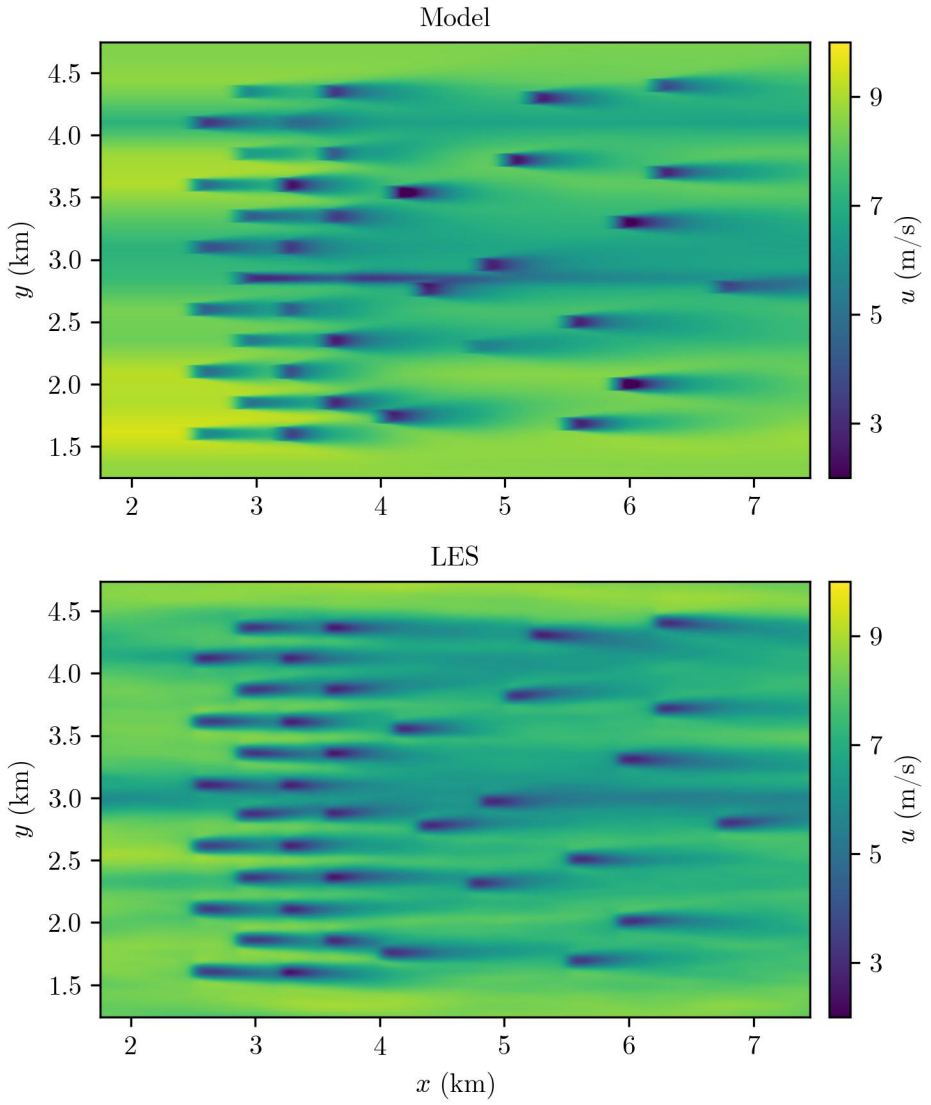

Figure 4-7. Comparison of state-estimated dynamic wake model velocity fields at $t = 60$ min and LES averaged between $t \approx 57.5$ min and $t = 62.5$ min. (Figure reproduced from [22])

turbines at the back of the farm, as shown in Figure 4-5. This is also the same geometry that was used in the hybrid regular-random wind farm validation of the ALC model in Section 2.2.3. The simulations are performed using the LESGO code [97], using the concurrent precursor inflow method [99], the Lagrangian-averaged scale dependent subgrid scale model [100], and actuator disk wind turbine models [11, 113].

The friction velocity is set to be $u_* = 0.45$ m/s and the surface roughness height is specified as $z_0 = 0.1$ m. The turbines have a rotor diameter of $D = 100$ m, a hub height of $z_h = 100$ m, and a local thrust coefficient of $C_T' = 4/3$.

The measured and estimated power production for six turbines which are representative of the farm are shown in Figure 4-6. The estimated power production shows very good agreement with measurements. The RMS error has a value of 10.4 kW, which corresponds to a percent error of 0.8%. The modeled velocity field at $t = 60$ min is compared to a time-averaged LES velocity field averaged between $t = 57.5$ min and $t = 62.5$ min in Figure 4-7. As demonstrated by this plot, the velocity field generated from the state-estimated dynamic wake model gives realistic velocity fields at a computational cost that is significantly lower than higher-fidelity approaches like LES or RANS. Specifically, for this simulation comprised of an ensemble of 256 perturbed wake models, the ensemble calculations are spread over 32 processors, which means that each processor handles eight of the ensemble members. The average time to advance these models is 0.9 seconds for a simulation time step of 2.1 seconds, which means that these calculations can be computed in real time based on typical wind farm control loops.

4.3 Conclusions

The incorporation of measurements from a wind farm is essential for calibrating wind farm models and dynamically updating the parameters and input values as conditions change within the wind farm. In this chapter we outlined two methods of incorporating the power measurements that are typically available in an operating wind farm. The first method was filtering the freestream turbines to find a time-dependent velocity inflow profile that can change in the spanwise direction. Incorporating a spanwise hetergeneous inflow into wake models improves the large scale matching with LES, however the entire wind farm experiences the same velocity profile in the streamwise

direction, which gives rise the the question of how quickly the inflow profile should react to changes in the LES. On the other hand, The Dynamic Graph model is able to represent the streamwise changes in time and space as well as the spanwise variations. This is a novel capability that is starting to be explored for wake models [112]. This enables the Dynamic Graph model to be able to follow the average trends in the power produced by the wind farm in time. However, these dynamic models still miss some of the turbulence-influenced events in the waked turbines, especially when the waked turbines receive energy from the meandering of turbulent eddies. To adjust for events like wake meandering and changing conditions throughout the farm, we design an EnKF that updates the states of the wind farm, the velocity deficit and the wake expansion coefficient of each turbine, to enable dynamic wake models such as that in [22] to better reproduce the power output of the LES. This method is able to represent the LES results in both a quantitative sense, Fig. 4-6, and a qualitative sense, Fig. 4-7. This approach provides a computationally efficient way to correct for modeling errors in control applications. This chapter outlines various estimation techniques that use data from LES to improve the prediction of wind farm models. These techniques are used to provide error correction when wake models are used in wind farm control applications.

Chapter 5

Wind Farm Power Tracking Control Using Yaw and Pitch

In this chapter we investigate the use of dynamic yaw actuation to augment pitch actuation in active power output regulation of wind farms. Power regulation (tracking) is a component of the grid services that it would be beneficial for wind farms to provide to aid in maintaining the power supply-demand balance of the electrical grid. Here we focus on secondary frequency regulation, which occurs over a period of tens of minutes [53]. In this situation, the grid provides a power demand curve and the wind farm must follow the demand signal. In this chapter, we present a simple pitch proportional integral (PI) controller framework for power tracking. However, pitch control can sometimes fall short in power tracking applications, particularly when the demand signal increases, or ramps up, and possibly exceeds the amount of power the is available in the farm. There have been many studies that indicate that yaw can increase the maximum power of a wind farm, both in static [47, 77] and dynamic applications[52, 85]. In this study, we want to incorporate dynamic yaw into the power tracking control framework to examine if augmenting pitch control with active yaw control improves the power tracking capability of a wind farm. Including dynamic yaw in this process is challenging as yaw changes on a similar timescale as the demand signal. In order to incorporate yaw, we use a model-constrained optimal control

strategy, employing the Dynamic Graph model outlined in Chapter 3 as the control model. We use the method from Chapter 4 to incorporate measurements from the turbines to estimate a spanwise heterogeneous time-varying inflow to the modeled farm. The controller is tested on an LES wind farm test plant with actuator disk turbine models executed using the JHU LESGO code described in Section 2.2.1 [91].

In Section 5.1, we look at a simple proportional integral (PI) controller framework for power tracking. Then in Section 5.2, we outline a model-based optimal control for dynamic yaw actuation to augment the pitch control. Finally, in Section 5.4.2, we tune the controller parameters and compare the performance of the pitch only and yaw augmented pitch controllers.

5.1 Pitch Controller Formulation

Figure 5-1 shows a block diagram of the controller, where the pitch PI loop is shown in teal. The PI control receives information from the power feedback from the wind farm plant. It then calculates a pitch update to adjust the thrust coefficients of the wind turbines to track the power reference signal.

The PI controller for the pitch control has the form

$$\Delta C_T' = \mathrm{k}_p e_P(t) + \mathrm{k}_i \int_{T_{\mathrm{k}_i}} e_P(\tau) d\tau, \tag{5.1}$$

where k_p is the proportional control gain, k_i is the integral control gain, e_P is the error between the power reference signal and the current LES power, and T_{k_i} is the timescale of the integral control.

PI control is a control strategy that has been utilized in multiple wind farm control applications, such as for pitch control in power tracking [76]. However, this pitch control strategy requires turbine-specific power set points. Since we are only given a total power reference signal for the farm we need to determine how to distribute this signal to the individual turbines. The process of determining how the power reference

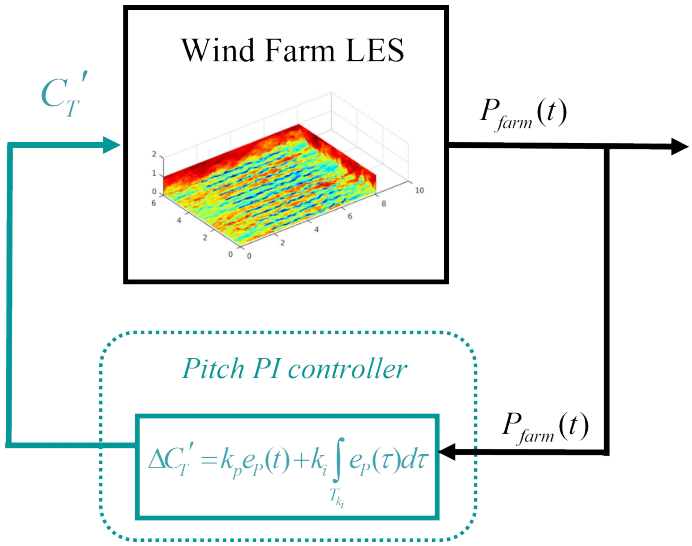

Figure 5-1. A block diagram that shows the structure of the pitch PI controller of the wind farm. The pitch PI controller (shown in teal) update is based on the power feedback from the LES. The new pitch control is then implemented in the wind farm LES, represented in the top block with an instantaneous snapshot from an LES wind farm simulation.

signal should be distributed to each turbine is a nontrivial problem. The fraction of the power provided to the total power from each turbine varies according to the incoming velocity and the relative position of the turbine in the farm. In this work, we present a model-free method that depends on past wind turbine power measurements. We determine the distribution for each turbine according to

$$P_{Ref,i} = \frac{1/T_{C_T'} \int_{T_{C_T'}} P_{LES,i}(\tau) d\tau}{1/T_{C_T'} \int_{T_{C_T'}} P_{LES}(\tau) d\tau} P_{Ref}, \tag{5.2}$$

where $P_{Ref,i}$ is the individual power reference signal for the i^{the} turbine, $T_{C_T'}$ is the time between pitch controller updates, $P_{LES,i}$ is the LES power for the i^{the} turbine, P_{LES} is the total LES power, and P_{Ref} is the total power reference signal. This equation illustrates that we take the average of the LES power for the i^{the} turbine over the pitch update time and divide by the average of the total LES power, giving the average percentage the i^{the} turbine has contributed to the power since the last pitch controller update. The power reference signal for the i^{the} turbine is then found by multiplying the average percentage with the overall power reference signal. Since

we take the average over the update time period, this could skew the sum of the distributed turbine-specific power reference signals, however the error between the total power reference signal and the sum of the individual power reference signals was found to be less than 1%.

In order to facilitate a fair comparison, we use the same proportional and integral gains for the pitch control and the yaw augmented pitch control. The tuning of the PI gains is discussed later in Section 5.4.1.

5.2 Yaw Controller Formulation

We now discuss the model-constrained optimal control for dynamic yaw actuation. In this section, we first examine the yaw controller formulation on its own and its application to a wind farm plant. We also look at a way to quantify the possibilities of dynamic yaw control.

5.2.1 Controller Formulation

The structure of the yaw model-constrained optimal control is shown in Fig. 5-2, where the dashed box outlines the controller. The controller is composed of an optimization solver which uses information from the cost function and the Dynamic Graph model to find an optimal yaw solution $\vec{\gamma}$ for each turbine. This solution is then given to the LES wind farm plant.

The optimal control power tracking problem is composed of solving a finite-time optimal control problem for each controller update. At each iteration, the optimization solver finds one optimal yaw for each turbine over the finite time horizon T. The resulting yaw is then applied to the wind farm for a time $T_a < T$. The problem is then solved again for a new time frame starting at T_a and ending at $T_a + T$, representing one controller update. The power from the farm provides updated initial conditions

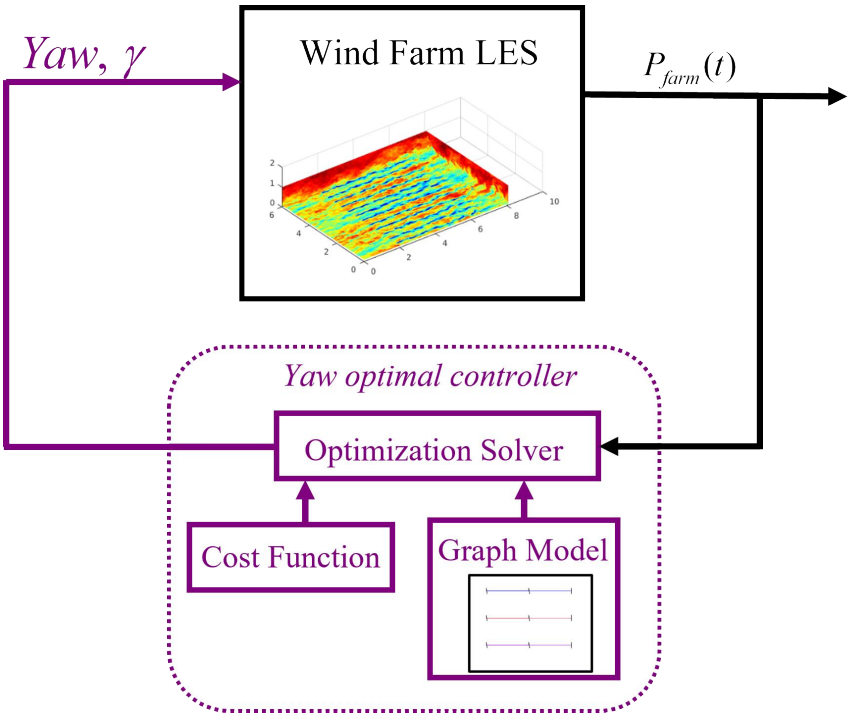

Figure 5-2. A block diagram that shows the structure of the model-constrained optimal control of the wind farm. The model-constrained optimal controller (shown in purple) is composed of an optimization solver which minimizes a cost function constrained by the Dynamic Graph model. The controller then feeds the optimal yaw to the wind farm LES, represented in the right block with an instantaneous snapshot from a wind farm simulation.

to Dynamic Graph model by supplying a velocity inflow condition based on current measurements, as outlined in Section 4.1.

Note that in this application, we are not use MPC, as that would require the calculation of a full yaw trajectory for each turbine. Due to the structure of the graph model, which prevents the use of gradient-based optimization at this time, we only calculate the best yaw value overall for each turbine over the finite time horizon. For this reason, the time horizon does not encompass the whole control signal, but rather encompasses a subsection of the control signal at each evaluation. This formulation is a first step in developing a full MPC framework for dynamic yaw.

The yaw of each turbine, γ, is used as the control variable in the model-constrained optimal control. For each iteration, the optimization problem finds the best yaw angles

over the horizon time according to a cost function that minimizes the difference between the power reference signal and the total LES power output. The cost optimization has the form

$$\min_{\gamma} \left[\int_0^{T_a} (P_{GM} - P_{ref})^2 dt \right], \tag{5.3}$$

where P_{GM} is the power from the Dynamic Graph model which is integrated into the future to predict the behavior of the wind farm. Yaw is incorporated into this problem through the Dynamic Graph model formulation. The optimization is subject to the limits

$$-30^\circ \leq \gamma \leq 30^\circ, \tag{5.4}$$

since the models for wind turbine yaw are less accurate beyond these points. The minimization of the objective function is accomplished using Powell's Algorithm, see A.2, in Python. Since this version of Python does not support bounds on the minimization, the bounds on the yaw values were enforced using the minimization algorithm by not allowing values that fall outside of the allowable yaw range, where if a value falls outside the viable range, the optimization function exits that evaluation and the optimization continues.

We first demonstrate a simple implementation in which the optimization process assumes that the optimal yaw can be applied instantaneously, and is applied at the start of the time horizon. Figure 5-3 shows the result of one of the optimizations. The wind farm configuration used is a six-turbine aligned wind farm arranged in two columns of three turbines each, shown in Fig. 5-3(a). In this case, the optimization is attempting to reach a final steady power output of 12 MW, shown in Fig. 5-3(b) by the black dashed line. The un-yawed power of the farm is 11.61 MW, which is the initial condition for the optimization. The function then attempts to minimize the difference between the power reference and the wind farm power over the entire time horizon. Figure 5-3(a) shows the final yaw values of the turbines, where only

the first two turbines are yawed a significant amount. Figure 5-3(b) shows the time evolution of the power of the wind farm (in blue) when the final yaw is applied to all the turbines at the start time. What is obvious from this plot is the initial dip when the turbines are yawed, and then the two increases in the power, which is the result of the wake effects from the first row decreasing on the second and the third rows. The effect of the wake on the second row is larger than on the third row, which makes sense as the wake effects are inversely proportional to distance due to the dissipation of the wake. The final value that the optimization achieves is 11.98 MW.

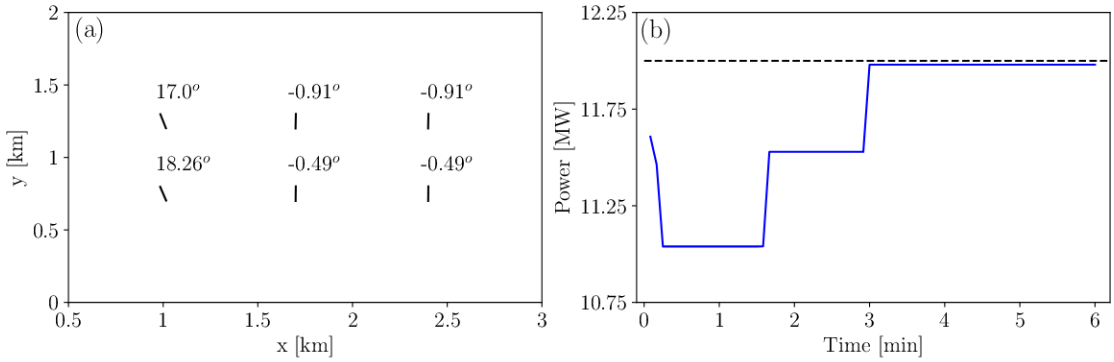

Figure 5-3. An example of a six-turbine wind farm optimized using a dynamic yaw response, where (a) shows the final yaw of turbines, where we are looking down at the farm from above, and (b) shows the evolution of the power of the farm in time (—), as compared to the target power reference signal (——)

This method takes into account the dynamic response of the farm to yaw, including the initial drop in power. Therefore, in order for the yaw to be effective, the time horizon needs to be long enough for the improvement in the total error measured by the cost function to outweigh the initial degradation. For example, if the time horizon in the Figure 5-3(b) was only four minutes, the decrease in the error would not have been enough to outweigh the initial drop in the integral of the total error over the time horizon, and the optimal solution would be to take no action. This behavior reflects the inter-turbine travel distance that needs to be reflected in modeling the behavior of dynamic yaw. This travel time plays an important role in determining the optimal control action. In the next section we discuss modifications of the setting of

Figure 5-3 that takes this and other timescales important in the dynamic yaw setting into account.

5.2.2 Yaw Control Results

We now examine the yaw controller performance when applied to wind farm plants for power tracking. First, we detail the application of the control framework to control the power output of a wind farm plant represented by the Dynamic Graph model. Then, the yaw controller is applied to an LES wind farm plant.

The application of the control framework to the Dynamic Graph model wind farm plant is shown in Fig. 5-4. This is nontrivial due to a few differences between the application of yaw in the optimization framework and the application of the yaw control to the wind farm plant. In the case shown in Fig. 5-3, the optimization assumes that the control is applied instantaneously at the beginning of the time horizon, and searches for a single set of yaw angles such that the error between the power output and the power reference signal will be minimized. However, this is an unrealistic model for the application of the control to the wind farm plant since yaw does not happen instantaneously. In practice, there is a finite time that a turbine needs to yaw. In the case shown in Fig. 5-4, we set the yaw rate to 4° per minute. The controller is then run as follows: First, the optimal yaw over the time horizon is found. Then, the turbines are yawed, according to the set yaw rate, until either they reach the desired yaw, or it is time for another controller update. The current yaw of the turbines is then used as the initial conditions for the next update, and the process is repeated. From this process, one can infer that the turbines will not always reach the target yaw within one controller update period. Additionally, since the optimization is nonlinear, it can be dependent on the initial conditions. To lessen this dependence, an ensemble of optimizations are run, each with differing initial conditions that are perturbed randomly within $\pm 2°$ from the current yaw. This ensemble of optimizations can be

parallelized since each realization is independent of the others. Thus, the addition of an ensemble of optimisations will not extend the time it takes to reach the final control since they can be calculated in parallel. The final yaw applied as the controller output is the average of the ensemble results.

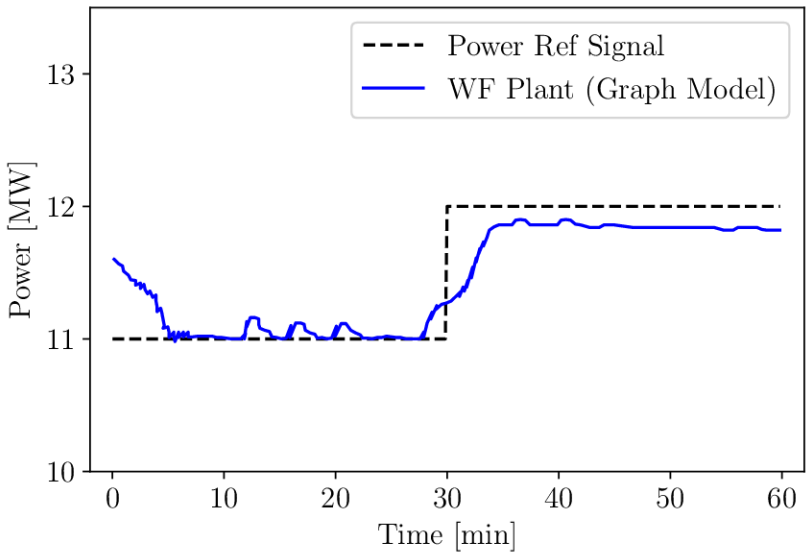

Figure 5-4. An example where the yaw model-constrained optimal control was used on a six-turbine wind farm with Dynamic Graph model as the wind farm plant. The figure shows the evolution of the power of the farm in time (—), as compared to the target power reference signal (– –). The controlled signal has an RMSE of 0.24 MW from the reference signal.

Figure 5-4 shows the results from using this framework with Dynamic Graph model as the wind farm plant, as well as the model-constrained optimal model. This farm has the same number of turbines and configuration as that shown in Fig. 5-3(a). In this case, the time horizon was 300 seconds, and the controller updated every 240 seconds. This was done as a proof of concept, letting the optimal yaw propagate fully through the farm before the next update. Finally, the controller used an ensemble of 10 optimizations for the control signal. There are a few aspects of the control worth remarking upon at this juncture. The first is that the controller is able to curtail the power down to 11 MW from the initial power of 11.61 MW. Then, the yaw control is asked to navigate a step change in the middle of the control period. A step change is

a somewhat unrealistic reference signal for solo yaw control due to the slower time scale of the control, however, we can see here that the yaw control is able to anticipate the step change before it happens, due to the model-constrained optimal framework. Finally, we can see that even though the power output increases with the step change, it is unable to fully reach the final power reference value, and seems to hover with a constant offset under the power reference signal. This could be a result of using the average of an ensemble of optimizations.

Following the proof of concept of the proposed dynamic yaw control approach, we now move on to the application of the controller to an LES plant model. In this work, we show the case where the yaw control is implemented in LES with a realistic finite yaw rate time. In this case, the horizon time is 5 minutes, and the controller is updated every 2 minutes. Additionally, an ensemble of five optimizations is used.

Figure 5-5(a) shows the total power of the farm from the LES, compared with the uncontrolled LES case, the power reference signal, and the model prediction of the power for the case where we assume instantaneous yaw implementation. Similarly to the previous case, we use a step function as the power reference signal. The higher value of the step function is 6.85 MW, which is based on the average of the uncontrolled case before the control was applied. The final value of the lower step is 6 MW. Figure 5-5(b) shows the yaw control actions applied to the simulation for each individual turbine.

In the initial stage of the control, which is the higher part of the step signal, the LES power is oscillating around the power reference signal already, so the optimal control action is to change nothing. Then, the control starts to actuate around 10 minutes, which can be seen in how the LES total power starts to differ from the uncontrolled case at this point. As a result of the finite yaw rate, this case starts to react to the step change at around 16 minutes, but only reaches a 30° yaw 2-3 minutes later. This slower yawing can be seen reflected in the LES power in Fig. 5-5(a), where

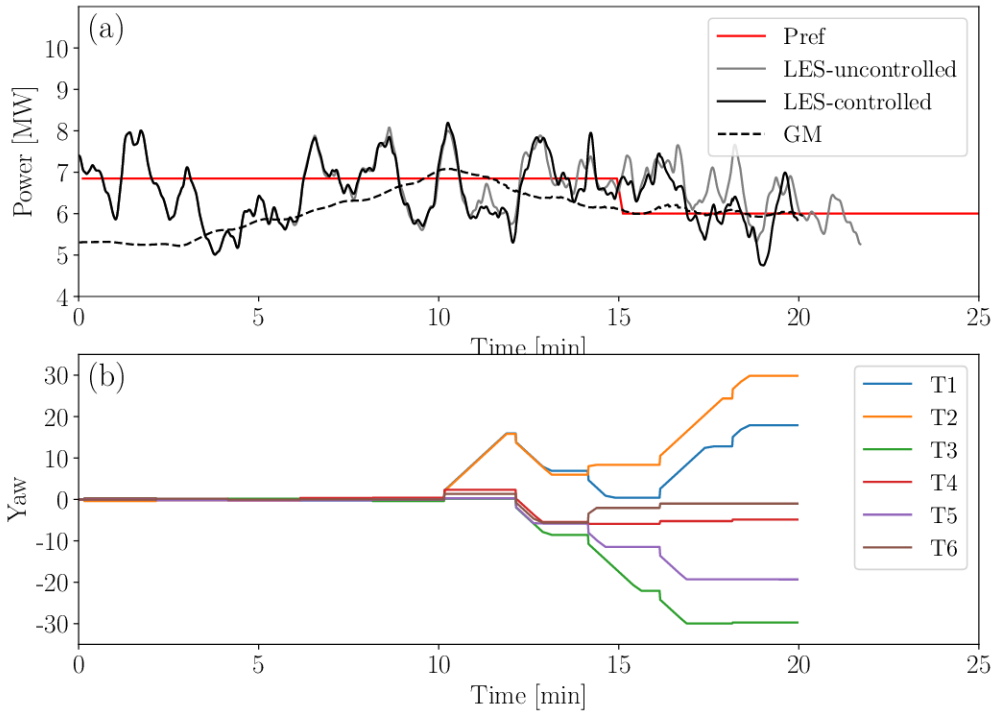

Figure 5-5. This figure shows the case with yaw control of an LES using Dynamic Graph model in model-constrained optimal, updating every two minutes using more realistic yaw actuation. The figure shows (a) the total power (—) comparison to the power reference signal (—), the uncontrolled LES case (—), and Dynamic Graph model prediction (——), and (b) the yaw control actions the of the turbines as a function of time.

the LES power reaches the lower power reference level only around 17 minutes. While it reacts on a slower timescale, we can still see a difference between the controlled case and the uncontrolled case.

While the LES was unable to follow the fine details of the power reference signal, due to the slower actuation rate and inter-turbine travel distances involved, as opposed to pitch control, the yaw control was able to encourage the LES power to fluctuate around the power reference signal and follow the step down more quickly than the uncontrolled case. This is also an important example of the time delay between the application of the optimal yaw, and when it impacts the system as a whole, which complicates the problem.

5.2.3 Quantifying the Potential of Yaw Control

Looking at the results in Fig. 5-5, we can see that the effect of the yaw control is limited. When applying this control, it would be useful to be able to quantify the extent of the control authority available. Previously, the affect of wake steering has been studied using LES in static yaw scenarios [77] or in power maximization in [52, 85]. However, now we can use the Dynamic Graph model to capture the dynamic response of the farm to a change in yaw to determine the potential of the yaw control action. In this section we demonstrate the application of the Dynamic Graph model on an 18-turbine wind farm to estimate the effect of yaw control actions. There are a large number of yaw combinations available in an 18-turbine wind farm, so we limit the scope of the study to an example of this idea where we assume that each of the six rows are yawed together and that the yaw is limited to be either 0° or 30°. In this context, we aim to show that the model can be used as a faster alternative to LES to examine the effect and scope of dynamic yaw actions on wind farms. The 18-turbine wind farm, arrayed in three columns of six turbines each, is shown in Fig. 5-6. This figure shows the locations of the turbines, and also depicts the graph of the wind farm if the inflow direction is 270° (flowing left to right). Due to the graph structure, the farm is partitioned into three separate subgraphs, where the lead turbines are the three turbines of the first row.

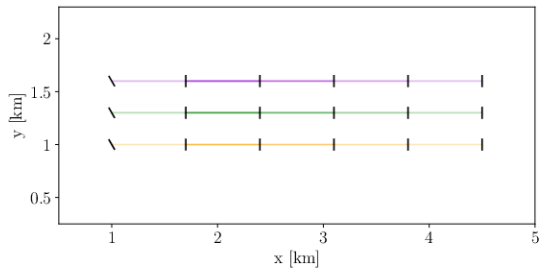

Figure 5-6. An example of the 18-turbine wind farm, with a graph of the interactions between turbines when the inflow is flowing from left to right at 270°, where the first row is yawed to 30°

In the study, we yaw all turbines in a given row by 30 degrees and then look at how the power is affected by the number and combination of rows that are yawed. Figure 5-6 shows an example of the farm where the first row is yawed at 30°. Figure 5-7 shows the dynamic response of the farm to the different yawing configurations. The initial dip in the power is in response to the yawing of the turbines, and varies according to how many rows are being yawed. The effect of the yaw then propagates through the farm before the farm settles into equilibrium by the final time.

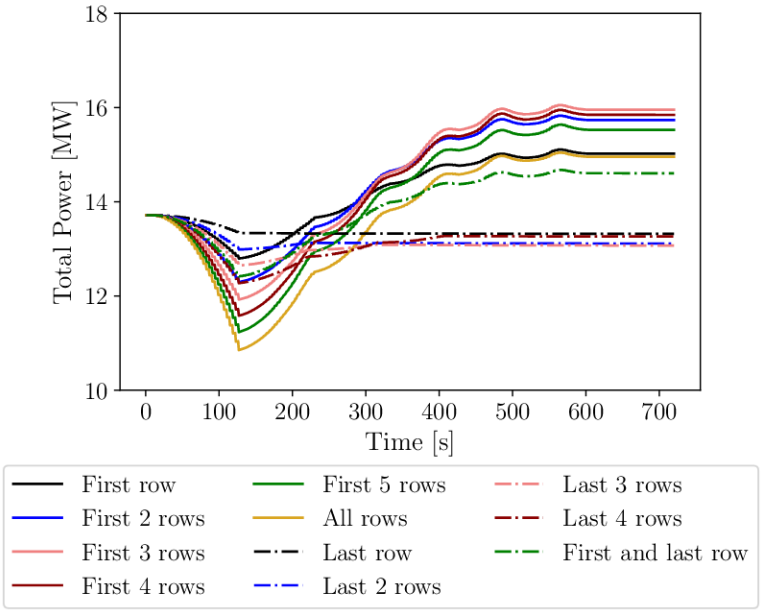

Figure 5-7. The dynamic response of the farm to a 30° yaw applied to different combinations of rows of the 18-turbine farm.

Figure 5-8 shows the initial power of the wind farm, with all turbines at zero yaw, compared to the final steady-state power of the wind farm. On the right side of the figure, the first rows of the wind farm are yawed, showing that there is a maximum when the first three rows are yawed to 30°. The right side of the figure explores the minimum power achievable with this particular yaw strategy by yawing the back rows together. In this configuration, the minimum power is reached by yawing the back two rows of the wind farm. However, the magnitude of the power increase is approximately three times that achievable for the power decrease. This suggests that yaw should be

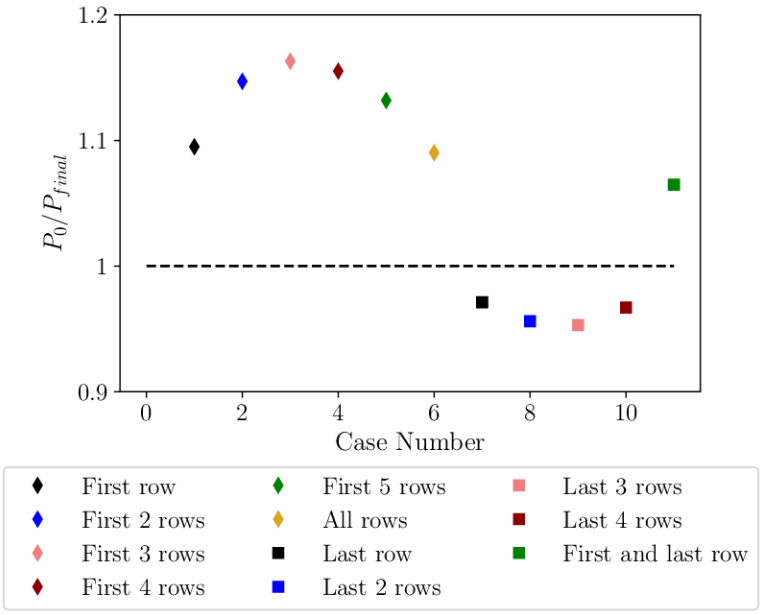

Figure 5-8. The normalized difference in the final power of the farm compared to the initial power (with no yaw) when a 30° yaw applied to different combinations of rows of the 18-turbine farm.

used predominantly to increase power output and less so to derate the power (which is better achieved using pitch actuation.)

5.3 Yaw and Pitch Controller Formulation

In this section, we present the incorporation of the yaw controller into the pitch control. We combine these controllers by separating the control framework into an inner loop for pitch control and an outer loop for yaw control. Figure 5-9 shows a block diagram of the controller, where the model-based optimal control yaw loop is shown in purple, and the pitch PI loop is shown in teal.

In our framework, the pitch is accounting for not only the difference between the power reference signal and the current LES power subject to the turbulent inflow, but also the effect of the changing yaw on the LES power output. However, since the yaw controller provides a trajectory over the yaw update time period, we can use our prior knowledge of the yaw actions and account for these changes in the pitch. This leads

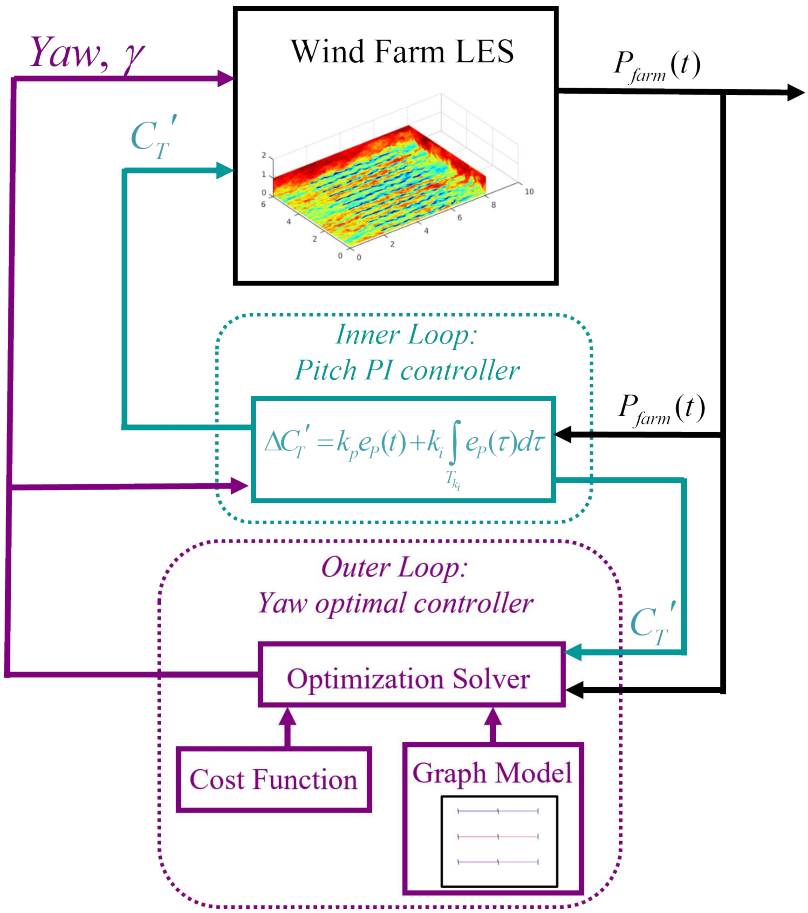

Figure 5-9. A block diagram that shows the structure of the model-based optimal control of the wind farm. The outer loop portraying the model-based optimal yaw controller is shown in purple and the inner loop portraying the pitch PI controller is shown in teal.

to a feed-forward pitch correction whose purpose is to account for the change in yaw as it happens, which is included in the PI control as

$$\Delta C_T' = \mathrm{k}_p e_P(t) + \mathrm{k}_i \int_{T_{\mathrm{k}_i}} e_P(\tau) d\tau + \Delta C'_{T,\gamma}, \tag{5.5}$$

where $\Delta C'_{T,\gamma}$ represents the change in the thrust coefficient required to keep the power constant if the yaw changes. We use a linear approximation to find this change in pitch following the equation

$$\Delta C'_{T,\gamma} = \frac{\delta C'_T}{\delta \cos(\gamma)} \left[\cos(\gamma_2) - \cos(\gamma_1)\right], \tag{5.6}$$

where $\frac{\delta C'_T}{\delta \cos(\gamma)}$ is the derivative of the coefficient of thrust according to the cosine of

the yaw, and γ_1 and γ_2 are the original yaw and the final yaw, respectively. The original and final yaw values are known from the yaw controller output, which if fed into the pitch controller, as shown in Fig. 5-9. In Eq. 5.6, we linearized the change in pitch according to $\cos(\gamma)$ to simplify the expression due to the thrust coefficient's dependence on this quantity.

We now need to find a way to define the derivative term $\frac{\delta C'_T}{\delta \cos(\gamma)}$. We use the power measurements from the farm, starting with the expression shown in Eq. 4.1. Rearranging Eq. 4.1 to separate the coefficient of thrust and the yaw terms, and using C'_T for C'_P, we find the following equation

$$\frac{2P_n}{\rho A U_{\infty,n}^3} = C'_{T,n} \cos^3(\gamma_n) \left(\frac{C'_{T,n}}{4 + C'_{T,n}} \right). \tag{5.7}$$

We can now take the derivative of the equation with respect to the quantity $\cos(\gamma)$, and solve for the quantity $\frac{\delta C'_T}{\delta \cos(\gamma)}$, resulting in the expression

$$\frac{\delta C'_{T,n}}{\delta \cos(\gamma_n)} = \frac{-3C'_{T,n}(C'_{T,n} + 4)}{\cos(\gamma_n)(C'_{T,n} + 16)}, \tag{5.8}$$

which represents the derivative of the coefficient of thrust with respect to the cosine of the yaw angle for the n^{th} turbine.

This linear approximation is determined to be reasonably accurate near the turbine, with an error of within 2.5% within a change of $\pm 1^\circ$ of the initial yaw angle, which is the range over which this correction will be applied.

5.4 Power Tracking Control Results

The model-constrained optimal yaw controller combined with the PI pitch controller was tested on an eighteen-turbine wind farm, whose geometry is the same as that shown in Fig. 5-6. The average velocity from this LES wind farm plant is shown in Fig. 5-10. The simulations are performed using the JHU LESGO code [97], using the concurrent precursor inflow method [99], the Lagrangian-averaged scale dependent

subgrid scale model [100], and actuator disk wind turbine models [11, 113]. The domain extends $6 \times 3 \times 1$ km in the streamwise, spanwise, and vertical directions with $512 \times 256 \times 192$ grid points in these directions, respectively. The friction velocity is set to be $u_* = 0.45$ m/s and the surface roughness height is specified as $z_0 = 0.1$ m. The turbines have a rotor diameter of $D = 100$ m, a hub height of $z_h = 100$ m, and a local thrust coefficient of $C_T' = 4/3$. The farm is composed of three aligned columns of six turbines each. This configuration was chosen because it is common for wind farms to be positioned in regular arrays.

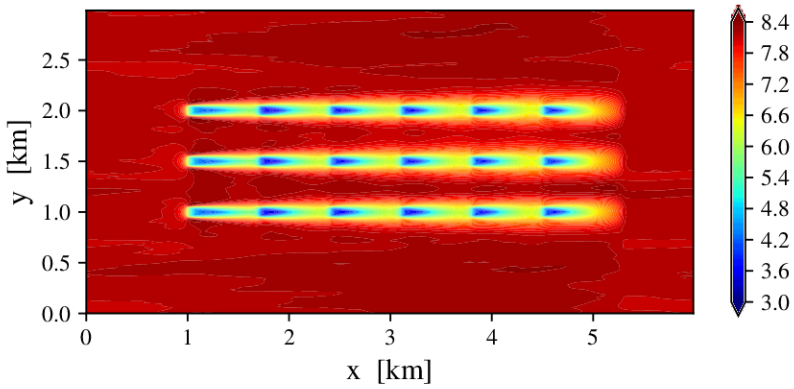

Figure 5-10. The average velocity, represented by the colorbar in m/s, from an LES of the eighteen-turbine wind farm used in the control study.

Three power reference signals are used to evaluate the control. The first is a sample power reference signal used in previous active power control studies [74] and is designed to require power above and below the baseline power, and to also integrate to zero. The other two signals are "RegA" and "RegD" signals and are types of secondary frequency regulation signals used by PJM, which is an ISO in the United States Eastern Interconnection. These signals are based on the area control error (ACE) signal. the ACE is a combined measure of the power imbalance and deviation of the frequency from its nominal operation value. The RegA signal is a low pass filter of the ACE that is usually followed using conventional regulating resources. The RegD signal is a high-pass filter of the ACE and is typically followed by more responsive energy

sources such as energy storage devices [114]. The test signal we selected integrates to zero. The power reference signal has the form $P_{ref}(t) = [1 - \alpha_d + r(t)]P_{base}$, where P_{base} is the average power of the wind farm in the five minutes preceding the start of the control simulation. The time dependent part of the power reference signal $r(t)$ represents the demands of the grid in an active power control situation, and varies $\pm 8\%$. The parameter α_d is the power setpoint reduction implemented to allow power tracking of increases in the reference signal. This parameter is referred to as the derate of the case and also represents a loss in revenue for the wind farm, since the power is being reduced. The goal of this work is to maximize the control authority while also minimizing the derate necessary to track the signal.

5.4.1 PI Tuning Parameter Study

We begin by examining the tuning of the parameters for the PI control. The PI controller gains were tuned using the full yaw and pitch controller. Figure 5-11 shows some examples of results from the controller using different PI gains. We performed both a proportional and integral gain study. The proportional gain was tested at increasing values until the trajectory exhibited worsening oscillations rather than an improved performance. The success of the power tracking is measured using the root mean square error (RMSE) of the signal, calculated as

$$RMSE = \sqrt{\frac{1}{N_{ST}} \sum_{i=1}^{N_{ST}} (P_{LES,i} - P_{Ref,i})^2}, \tag{5.9}$$

where N_{ST} is the number of measurements taken over the timeframe the control is applied, $P_{LES,i}$ is the total power from the LES at time i and $P_{Ref,i}$ is the power reference signal value at time i.

Figure 5-11 shows some results from the integral gain study, using the same eighteen-turbine wind farm and LES conditions used in the previous section. In this study, $\alpha_d = 0.06$ was used, resulting in a 6% derate to the base power of the wind

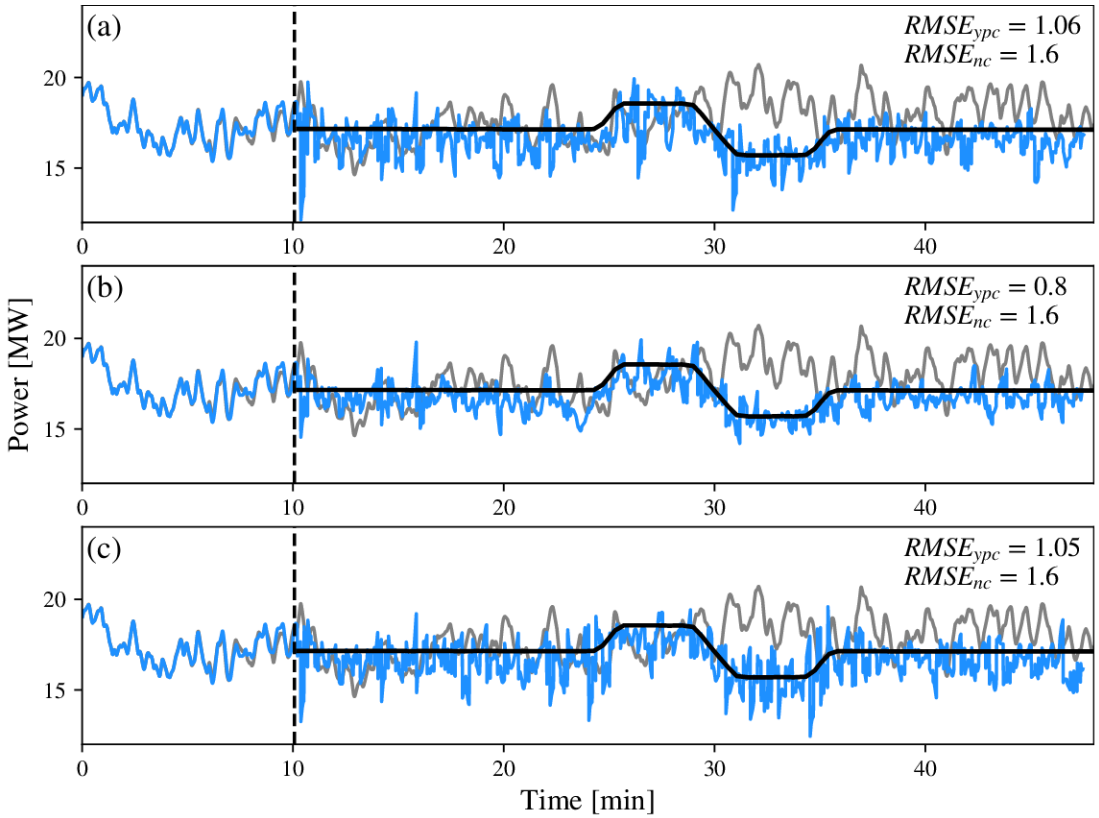

Figure 5-11. A sampling of the results from the gain parameter study for the PI controller where the power reference signal (—) is compared with an uncontrolled LES case (—) with the yaw and pitch controlled LES case (—). This figures shows the cases where (a) $k_p = 1$ and $k_i = 0.2$, (b) $k_p = 1.25$ and $k_i = 0.1$, and (c) $k_p = 1.5$ and $k_i = 0.15$.

farm. Figure 5-11(a) shows the total power results using $k_p = 1$ and $k_i = 0.2$, Fig 5-11(b) shows the total power results results using $k_p = 1.25$ and $k_i = 0.1$, and Fig. 5-11(c) shows the total power results using $k_p = 1.5$ and $k_i = 0.15$. The sucess of these studies was determined using RMSE as a quantitative measure. The cases shown in Fig. 5-11(a) and Fig. 5-11(c) clearly perform worse than the middle case, both from an RMSE perspective, and from a qualitative perspective when comparing the amplitude of the oscillations in the total controlled power. The case in Fig. 5-11(b) performs better than the other two, with smaller oscillations and also a lower RMSE. This reflects the outcome of the study that found that lower k_i values provided better results, since the case in Fig. 5-11(b) also has the lowest integral gain of the cases.

This suggests that putting too much weight on the history of the error negatively impacts the PI control in this application. While the case in Fig. 5-11(b) performs better than the others shown, we determined that a smaller integral gain was more successful. The proportional gain was chosen to be $k_p = 1$ to reduce the excessive oscillations that appeared at higher gain values. Thus, the final values decided upon from this study to be used in the rest of the results for this section is $k_p = 1$ and $k_i = 0.05$.

5.4.2 Controller Results

Figure 5-12 shows an example of the control applied to the LES wind farm plant. Figure 5-12(a) shows the total power of the wind farm, compared with the given power reference signal and the power output of the wind farm if it was run without any control at a constant coefficient of thrust and zero yaw. In the total power plots, the control is turned on after the vertical black dashed line (located at 10 minutes). In the case shown in Fig. 5-12, $\alpha_d = 0.04$, resulting in a 4% power derate from the regular operating conditions.

In this case in Fig. 5-12(a), we display the RMSE of both the controlled and the uncontrolled signal. Here, the controlled signal has an error that is less than half that of the uncontrolled case. We can also see that the controlled case is able to track the signal both when it requires more power than the uncontrolled case (minutes $25 - 30$), and also when it requires less power (minutes $30 - 35$).

In addition to the total power, we can also look at the control signals. Figure 5-12(b) shows the time series of the yaw of the wind turbines throughout the controlled time period. The turbines are grouped according to row with each row designated by a different color. In this application of the controller, the average of an ensemble of five optimizations was used to determine the yaw control. The number of ensemble members was reduced to five for this application in order to save computational time

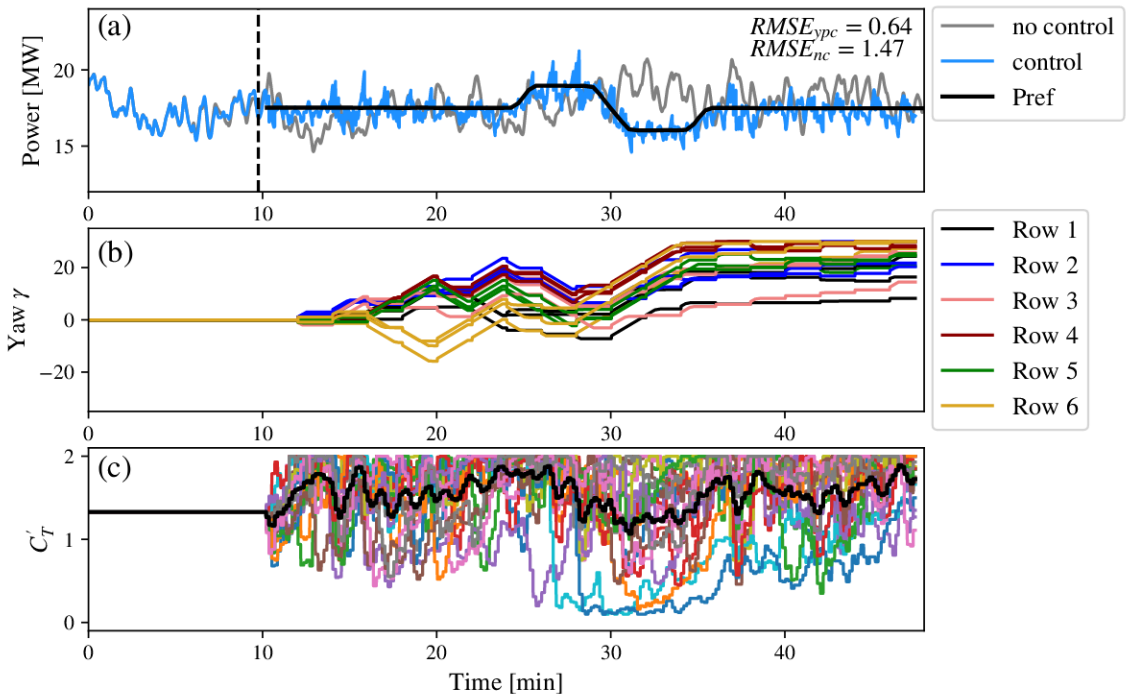

Figure 5-12. An example of the yaw and pitch controller applied to active power control using an eighteen-turbine wind farm LES plant. The figure shows (a) a comparison between the power reference signal, the LES if no control was applied, and the controlled LES results, (b) the time history of the yaw control, where each row is represented by the same color, and (c) the time history of the coefficient of thrust for each turbine, where the average pitch is shown by the heavy black line.

as the controller optimizations were calculated in serial in this implementation. All the turbines start at zero yaw in the beginning of the simulation, and then become active after the control is initiated, whose start time is indicated by the black dashed line. It can also be seen from this plot that the yaw is limited by the finite yaw rate, which necessitates the ramping behavior that starts around the 30^{th} minute, where the turbines are yawed over several yaw updates. Looking at the behavior of the yaw over the control period can help us determine the dynamic effects of yaw control and achieve a better understanding of wind farm behavior.

Finally, Fig. 5-12(c) shows a time history of the coefficient of thrust over the control period. In this application of the controller, the gains $k_p = 1$ and $k_i = 0.05$ were used for the PI control, as discussed in Sec. 5.4.1. The coefficients of thrust also

have an initial value of $C_T' = 4/3$, which is meant to represent standard operating conditions of the farm , though it is not the maximum allowable value that would be used in greedy control. Greedy control in wind farms occurs when all of the turbines are attempting to extract maximum power by setting the thrust coefficient to the maximum value $C_T' = 2$. A discussion of how the gains were chosen follows. The different turbine signals are denoted by different colors. The black line denotes the average behavior of the thrust coefficient, which can be used to determine how much pitch actuation is available in the farm at a given time. The three quantities shown in Fig. 5-12 represent the measures of the yaw and pitch controller.

In Section 5.3, we outlined how a pitch feed-forward control was implemented to mitigate the effect of the yaw changes on the pitch control. Figure 5-13 shows the behavior of the pitch correction term compared to the yaw actions that are being accounted for by the pitch feed-forward method. Figure 5-13(a) shows the coefficient of thrust yaw compensation term that accounts for the yaw term in the PI control, and Fig. 5-13(b) shows the yaw actions that the coefficient of thrust term is associated with. This figure focuses on the second turbine and the sixth turbine (which are in the the second row and sixth row, respectively) to illustrate the behavior of this term.

One thing to note about this term is that regardless of the sign of the current yaw of the turbine, the compensation term is positive if the yaw is moving away from zero and negative if the yaw is moving towards zero. This makes sense, as the power will reduce if the turbine is yawed further from the zero yaw position, so the coefficient of thrust should be increased to keep the power steady. This effect is reversed if the turbine is yawed closer to zero yaw. This behavior is reflected in the Fig. 5-13, where the positive sections of the compensation term in Fig. 5-13(a) correspond to the yaw of the turbine moving further from the zero yaw position, and the negative section in Fig. 5-13(a) correspond to the yaw of the turbine moving closer to the zero yaw position in Fig. 5-13(b).

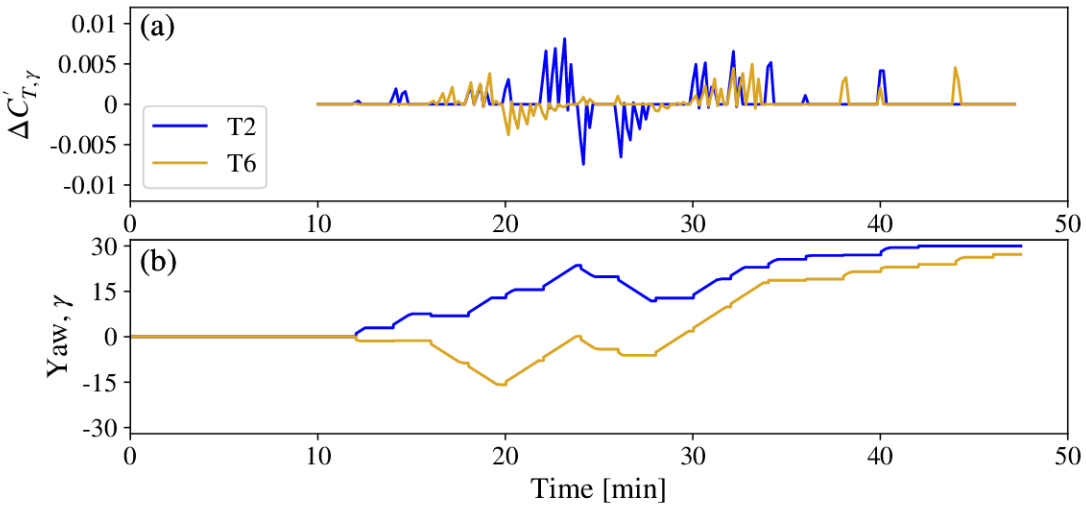

Figure 5-13. An example of the pitch compensation for the yaw changes prescribed by the controller in the simulation shown in Fig. 5-12, where (a) shows the changes in the coefficient of thrust in reaction to the yaw changes, and (b) shows the yaw of a turbine in the second row and a turbine in the sixth row.

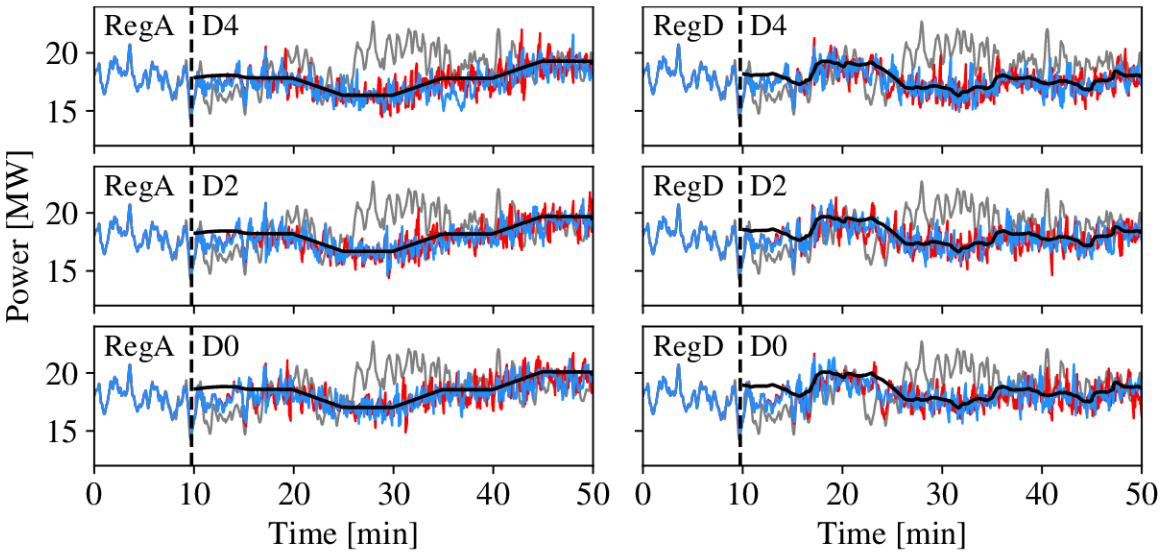

Figure 5-14. A comparison of the power reference signal (—), an uncontrolled LES case (—), the yaw and pitch controlled LES case (—), and the only pitch control LES case (—). This figure shows comparisons for a representative RegA signal (left column), which has a slower rate of change, and a representative RegD signal (right column), which has a slower rate of change, for varying pitch derate values. The derate values are arranged by row, where the first row has a 4% derate, the second row has a 2% derate, and the third row is has no derate.

5.4.3 Yaw and Pitch Control Compared to Only Pitch Control

In this section we investigate the performance of the controller when applied to a RegA and a RegD signal, and compare the performance of the yaw plus pitch controller with the performance of a controller only using pitch control. The controller was applied to two different one RegA and one RegD test signal, and the control was implemented with three different derates: 0%, 2%, and 4% derates for each signal. Further details regarding these signals are provided in [114]. The performance of the yaw plus pitch controller was compared to the uncontrolled LES case, but also to the LES case where only the PI pitch controller was applied and all turbines remained unyawed. The initial flow and simulation conditions are the same as those outlined in Section 5.4.2, with an initial thrust coefficient value of $C_T' = 4/3$. Both test signals are 40 minutes long. Figure 5-14 shows the total power from these simulations, where grey denotes the uncontrolled LES case, blue denotes the yaw and pitch controlled LES case, red denotes the only pitch controlled LES case, and black represents the power reference signal. The black dashed line shows where the control is initiated in the simulations. The type of signal is denoted in the upper left corner, followed by the derate value, where "D0" denotes a 0% derate, and so on. Qualitatively, the yaw plus pitch control performs similarly to the only pitch control, with perhaps smaller oscillations in some portions. Notably, in the inital portion of the control (from 10-20 minutes), both of the controllers perform very similarly. In this region for the yaw and pitch controller, the pitch control is dominant due to the delay that is inherent in the yaw control. Since the pitch control in both controllers, the yaw and pitch controller and the only pitch control, have the same formulation, the resulting total power from the wind farm is very similar. Once the yaw control becomes more present in the control, which takes approximately 5-10 minutes, the controlled signals from the two cases start to differ.

Quantitatively, we can use RMSE to compare the difference between the yaw and

Signal Type	4% derate		2% derate		0% derate	
	Yaw + Pitch	Pitch	Yaw + Pitch	Pitch	Yaw + Pitch	Pitch
RegA	0.90	0.82	0.84	0.85	0.90	0.93
RegD	0.82	0.84	0.85	0.92	0.97	1.01

Table 5-I. The RMSE for the RegA and RegD Test Signal Comparison

pitch controller and the only pitch controller. The RMSE for each of these cases is shown in Table 5-I. The average RMSE for the yaw and pitch controller for these six cases is 0.88, and the average RMSE for just the pitch control is 0.90. We can see that the yaw and pitch control performs better than just the pitch control for all cases except the RegA with 4% derate. This is a result of the dip in the total power for the yaw and pitch control visible around minute 35 in Fig. 5-14 for this case. This dip is a combination of yaw being applied to the wind farm and a sudden dip in the overall power caused by a turbulent fluctuation, such that the pitch was unable to compensate for a short amount of time. Since the yaw control reacts on a slower timescale, it is unable to react to sudden changes due to turbulent fluctuations. In this case, the pitch was also unable to compensate for this changing condition since it did not have enough control actuation available. In these simulations, the yaw and pitch control offers a slight advantage over just the pitch control, and is able to actively leverage yaw to improve the control.

Figure 5-15 gives a more detailed view of the total power for the RegA 0% derate signal in Fig. 5-14, along with the time-dependent yaw and pitch signals applied by the controller. Figure 5-15(a) shows the total power of the yaw and pitch controlled LES case compared with the power reference signal (black), the only pitch control (red) and the uncontrolled LES case (grey). Figure 5-15(b) shows the time-dependent yaw actions from the simulation. We can understand the yaw actions better by comparing them to the behavior of the total power and the reference signal. The yaw begins to activate larger changes just after 20 minutes in the simulation, particularly the

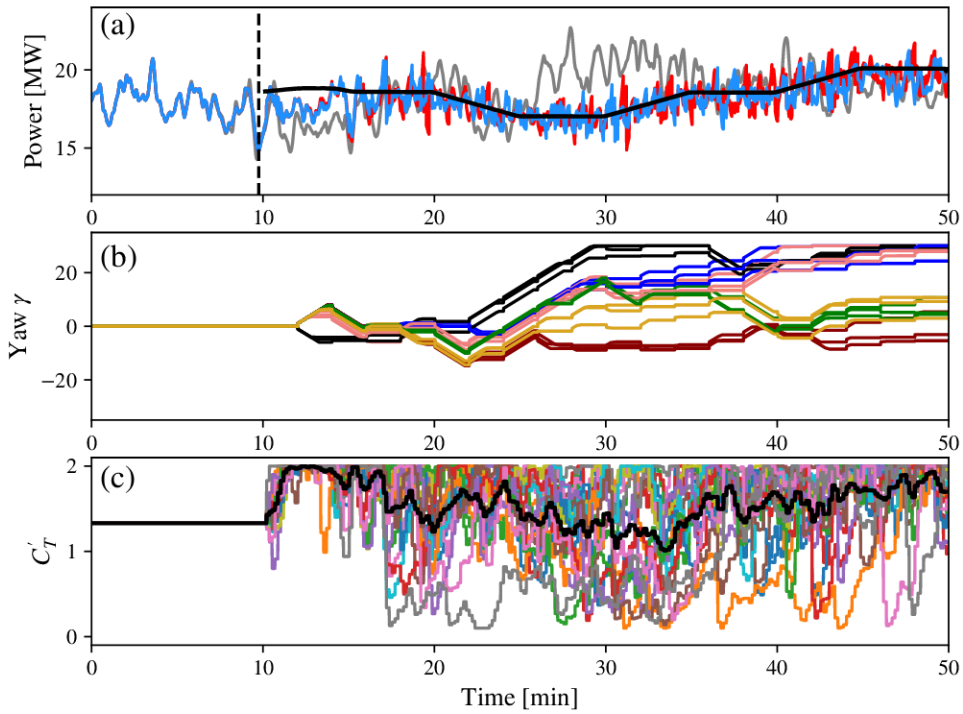

Figure 5-15. A closer look at the results from the RegA 0% derate case from IC2, showing (a) a comparison of the total power from the wind farm with the reference signal (—), an uncontrolled LES case (—) [$RMSE_{nc} = 1.97$], the yaw and pitch controlled LES case (—) [$RMSE_{ypc} = 0.9$], and the only pitch controlled LES case (—) [$RMSE_{ct} = 0.93$], (b) the time history of the yaw control, again separated into rows according to color in the same way as Fig. 5-12, and (c) the time history of the coefficient of thrust for each turbine, where the average pitch is shown by the heavy black line.

first row (denoted in black). This occurs at the same time that the uncontrolled LES dips below the power reference signal, which could prompt a yaw response to increase power. Most of the other rows follow the first row in yawing, between minutes 25 and 35, where the uncontrolled power has a peak. Finally, just after 35 minutes, the power references signal again increases beyond the uncontrolled power and the yaw of the last three rows of the wind farm tend towards zero. This enables the wind farm to follow the power reference signal as it continues to increase until the end of the simulation. Notably, this configuration also happens to mirror the results in Section 5.2.3, where it was found that the highest power resulted from the case where the first three rows were yawed and the final three rows were unyawed, though the cases on

either side were close in magnitude to the optimum case found. Figure 5-15(c) shows the time-dependent thrust coefficient, with the thicker black line showing the average of all the turbines. From this, we can conclude that we have a reasonable degree of thrust control authority, with the thrust coefficient only approaching the maximum value of 2 at the beginning of the simulation where the power reference signal started above the uncontrolled power. This also shows that the pitch controller performs the majority of the power tracking effort in this study.

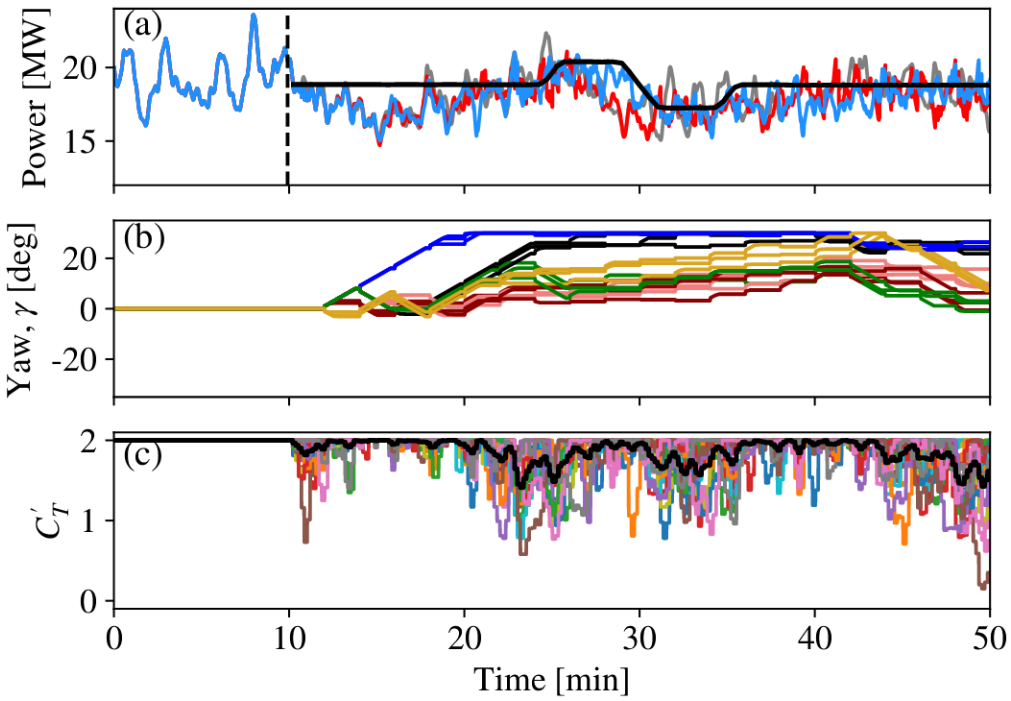

Figure 5-16. The initial condition for this case is the greedy power of the wind farm with zero yaw derated by 2%. The figure shows (a) a comparison of the total power from the wind farm with the reference signal (—), an uncontrolled LES case (—) [$RMSE_{nc} = 1.49$], the yaw and pitch controlled LES case (—) [$RMSE_{ypc} = 1.29$], and the only pitch controlled LES case (—) [$RMSE_{ct} = 1.48$], (b) the time history of the yaw control, again separated into rows according to color in the same way as Fig. 5-12, and (c) the time history of the coefficient of thrust for each turbine, where the average pitch is shown by the heavy black line.

Something to note about previous studies presented here is that the coefficient of thrust was initialized at 4/3 to simulate standard operating conditions. However, as can be seen in Fig. 5-15(c), this enables the thrust coefficient to have initial

actuation authority, since the thrust coefficient can increase to its maximum value of 2. This means that the derates shown above are derated from the average power when $C_T' = 4/3$ and not from the greedy power where $C_T' = 2$, building in an addition derate. We also wanted to investigate the effect of yaw on a wind farm operating in greedy control conditions where the thrust coefficient actuation is severely limited. Figure 5-16 shows the result of applying control to a wind farm initialized with $C_T' = 2$ (greedy control) and a 2% derate. Figure 5-16(a) shows the total power of the yaw and pitch controlled LES case compared with the power reference signal (black), the only pitch control (red) and the uncontrolled LES case (grey). Figure 5-15(b) shows the time-dependent yaw actions from the simulation and Fig. 5-15(c) shows the time-dependent thrust coefficient, with the thicker black line showing the average of all the turbines. Initially, both control schemes struggle to meet the the power reference signal since there is not enough energy in the uncontrolled power to meet the demand. However, the yaw and pitch controller is able to leverage yaw to increase the power available, and is able to better meet the demand during the increase in the power reference signal, between 25 and 30 minutes. This is notable because both the uncontrolled LES case and the only pitch case are unable to meet this demand, suggesting that the yaw control was able to increase the power that could be extracted from the farm through wake steering. The yaw signal shows that both the first (black) and second (blue) rows are yawed to nearly 30° before this elevated section, which contributes to the increase in power. After the decrease in the power reference signal, the yaw and pitch control struggles for a few minutes, but eventually adjusts the yaw to be able to meet the final steady value. The yaw is adjusted here by reducing the yaw of the last four rows, while leaving the first two rows yawed. The effect of this can also be seen in Fig. 5-15(c), where the thrust coefficient lowers during the final five minutes, showing that using yaw control can help increase the control actuation in cases where the power reference signal is close to the maximum power the farm can

provide. These results show that the yaw control can be beneficial in power tracking but this implementation struggles at fast timescales.

5.5 Conclusions

This chapter proposed a comparison between a yaw and pitch outer and inner loop control framework and a pitch only control framework to see if adding dynamic yaw can improved power tracking. The pitch controller in this framework is a simple PI controller. The yaw outer loop controller proposed takes into account the dynamic response of the wind farm to determine the control actions. The yaw controller is based on a model-constrained optimal control design, using the Dynamic Graph model as the model in order to be able to incorporate the dynamic effect of yaw on a wind farm. The yaw and pitch controller was applied to an LES wind farm plant and tested over multiple conditions and signals. First, as a proof of concept, the controller was shown to be able to follow a signal, improving upon the uncontrolled case by leveraging both pitch and yaw. Then, the controller was applied to a RegA and a RegD test signal, which are two different types of secondary frequency signals used by PJM, an ISO in the United States Eastern Interconnection. The study compared the performance of the controller when applied to the two test signals for three different derate values with control that only uses pitch actuation. This study found a slight improvement using the yaw and pitch controller as compared to using only the pitch control. Finally, the controller was applied to a case initialized with greedy control, where the thrust coefficient starts out at its maximum, which is a plausible operating condition for wind farm. The controller was then applied to the farm to track a power reference signal that was derated 2% from the greedy power. This case showed that the yaw control was able to increase the total power in the farm using wake steering and was able to meet demand when the power reference signal increased above the uncontrolled power. However, the yaw control lagged behind the power reference

signal in places due to the timescale that our yaw control operates. This was an initial study to examine the question of whether augmenting pitch control with dynamic yaw control is beneficial in power tracking. The next step to answering this question is developing more computationally efficient methods to apply this control in an MPC framework. This would enable the controller to have more complete information about the future of the demand signal and the turbine interactions in the farm. In addition to active power control, the dynamic response of the wind farm to yaw could be used advantageously to adjust the pitch to account for turbine yaw that is applied from a static yaw optimization. Additionally, the Dynamic Graph model is promising for controlling wind farms through changing conditions such as varying wind inlet directions and in applying optimal yaw conditions during these changes.

Chapter 6

Conclusions and Future Work

As wind energy becomes a larger component of the electrical supply, we must fully understand its capabilities and its new role in the electrical grid. In addition, we can leverage this new understanding to better design and operate existing and future wind farms. This thesis seeks to expand upon this current understanding through physics-based modeling, examining and modeling the transient response of wind farms to changing conditions and integrating this model into an active power controller.

One of the greatest challenges in wind farm modeling is representing the interactions between turbines via their wakes. Representing these relationships accurately aids in more accurate power predictions to improve the design of wind farms. This thesis provides advancement in the area of coupled models by enabling coupled models to represent a nonuniform inflow and to become more generally applicable to modeling the wake interactions in wind farms with arbitrary wind turbine geometries. It also provides more in-depth analysis of the results of the model which can provide more insight into wind farm physics. We present the Area Localized Coupled (ALC) model in Chapter 2 which combines two physics-based models of different scales to provide local information for each turbine in the farm. More specifically, the ALC model consists of a wake model, which focuses on the scale of individual turbines, and a top-down model, which focuses on the scale of the atmospheric boundary layer. The combination of these scales enables the ALC model to provide more information than

typical wake models and the localized framework enables its application to any wind farm geometry. The ALC model was validated against two nonuniform wind farms in LES for multiple wind inlet directions, and was able to capture both the general trends in the variation of total wind farm power with wind direction, and the pattern of the power of individual turbines as well. Additionally, since the wake model used in the ALC model has a dynamic implementation, the ALC model could be extended to a time-dependent framework in future work.

While our initial implementation of the ALC model is more suited for wind farm design and analysis applications, the dynamic reaction of a wind farm to changing conditions is also necessary to understanding the full capability to wind farms. In Chapter 3, we present an estimation algorithm, the Dyanmic Graph model, that uses a graph to represent the relationships between turbines that computes the wind speed and power of each turbine in the farm under dynamically changing conditions. In Chapter 1, we mention that representing dynamic changes in wind farm conditions is challenging and computationally expensive. This model represents the first analytical dynamic graph-based wind farm model that is capable of representing these changing conditions. The Dynamic Graph model denotes the turbines as the nodes of the graph and the turbine interactions as the edges. Under this framework, we only compute the turbine velocities at hub height, based on their interactions with upstream turbines captured by the graph edges, rather than estimating a velocity field around each turbine as in the ALC model. This increases the efficiency of the model at the cost of the detail provided by a fully-resolved velocity field, however this was judged an acceptable trade-off for a control-oriented model. The dynamic graph setting enables the model to represent time-varying changes in the farm behavior due to changing conditions, such as an incoming wind direction shift or a yaw change. This model is also the first analytical dynamic yaw model, for which it employs a novel analytic expression for wake shape deformation under yaw [49]. The Dynamic Graph model

was validated under a changing wind direction with an unsteady RANS simulation, and showed good agreement over the transient response. The Dynamic Graph model was also approximately 20 times faster than the RANS simulation it was compared with. In addition to a wind direction change, the Dynamic Graph model also showed good agreement with LES for a dynamic yaw case.

While the wind farm models described above do well representing key factors of the wind farm, a reduced order model must leave out some of the physics of the full system to decrease computation time. In Chapter 4 we use measurements from the wind farm plant for error correction in data estimation due to unmodeled effects. In this work, we only use the power measurements from the turbines, which is information that would be available in an operating wind farm. Since wind farms can cover multiple kilometers of land and the local wind conditions can vary over this area, we first apply these measurements to implement a nonuniform inflow in a dynamic Jensen wake model and the Dynamic Graph model. This addition improves the power prediction of the individual turbines in the farm. Finally, the measurements were also used in an Ensemble Kalman filter to improve the estimation of the wake deficits and wake expansion coefficients using the model presented in [22].

In Chapter 5, after the Dynamic Graph model is coupled with the dynamic nonuniform inflow estimation, it could be applied in an optimal control framework for power tracking. While dynamic yaw has been applied in power maximization applications, it has not, to date, been applied to the power tracking problem. This work performs an initial study to examine if the addition of dynamic yaw control to pitch control could be beneficial for power tracking. The controller consisted of an outer loop yaw controller, to account for slower bulk power changes, and an inner pitch loop controller, to perform faster power adjustments. The outer yaw loop used model-constrained optimal control, using the Dynamic Graph model as the controller model. This yaw and pitch controller was applied to an LES wind farm

plant containing eighteen turbines. The addition of yaw control slightly improved the controller performance over using only pitch control in this case, and was able to increase the power output in a greedy control situation. This initial study provided a proof of concept that dynamic yaw control could improve pitch control in power tracking, but the next step is to work towards implementing a full model predictive control framework to take full advantage of the dynamic yaw model. We used Dynamic Graph model to examine both the limitations of yaw control and the dynamic response of a wind farm to yaw actions. The capability to predict the dynamic response of the wind farm to yaw a priori has potential to be useful in adjusting the pitch control to account for yaw changes as they happen. Additionally, the Dynamic Graph model could be used for controlling wind farms through changing conditions such as varying wind inlet directions and in applying optimal yaw conditions during these changes.

While this dissertation has presented advances in wind farm modeling and power control of wind farms, there is still additional work needed to implement the approaches outlined previously.

Vast literature exists on representing wind farms on an individual turbine basis with wake models and on an atmospheric scale with top-down models. We presented the ALC model that combines these two views to provide more information about the wind farm than each piece offers. The ALC model compared well with LES and also was able to offer a rich array of information for additional analysis on the physics of wind farms. However, it would also benefit from comparison with field wind farm data. Additionally, open questions still remain around analytically representing wind farms in stable and convective atmospheric conditions as well as the neutral boundary layer description used in this work. It would be beneficial if the ALC model could be extended to be applicable in any atmospheric conditions. The ALC model is currently a static model, which represents the average power from the wind farm. It would also be advantageous if this coupled framework could be extended to dynamic settings.

In addition to static representations, dynamic representations of wind farms are also needed to inform control action of wind farms to enable optimal operation. Wind farms are subjected to various changing conditions during operation, such as varying incoming wind direction, that need to be taken into account in control schemes. The open question in this area is how to represent these dynamic events in an analytical framework. The Dynamic Graph model is the first analytical graph-based wind farm model, and is capable to modeling these dynamic events. This graph-based model is also one of the first wind farm models that is validated in modeling dynamic yaw events. However, the model could also benefit from validation with operational wind farm data and data from multiple atmospheric conditions. Another open question is this area is how to represent the transition between different atmospheric conditions. Additionally, the structure of the Dynamic Graph model does not enable gradient-based optimization currently, which would be a very advantageous adjustment.

Active power control is an important and active area of wind farm research, both for power maximization and power tracking. Since yaw has been shown to be able to increase the bulk power of the wind farm, an important question is how to leverage yaw control in both of these applications. Initial studies have been performed in power maximization, such as the LES proof of concept study done in [85], which showed the potential gains of control that possesses perfect information, and in [52] which was applied to operational turbines. However, in power tracking, to date, only static yaw control has been applied, such as in [66]. The study presented here represents a first step towards investigating the benefits of dynamic yaw control in power tracking applications. In order to more fully understand the impact of yaw, we need a way to compute a trajectory of yaw signals over a longer time horizon so that we can take advantage of knowledge of the temporal and spatial scales of wake interactions. This is a challenge and requires further research into dynamic yaw models and optimization techniques. Ideally, a full MPC approach could be applied in the outer loop of the yaw

and pitch controller, to examine the full potential of dynamic yaw control. Finally, accounting for the dynamic response of a wind farm to yaw changes could also be applied in static yaw control and in active power control over slower timescales and conditions with a changing inlet wind direction.

The study of wind farms and their capabilities is a vast and varied subject undertaken by many dedicated researchers to increase our understanding of wind farms and to enable wind to continue to grow into a major electrical provider. Through this research, we can contribute to steps towards a renewable future.

References

[1] U. E. I. Agency, "International energy statistics," Tech. Rep., 2018.

[2] BNEF, "New energy outlook 2019," Tech. Rep., 2019.

[3] I. E. A. (IEA), "World energy outlook 2018," Tech. Rep., 2018.

[4] P. Veers, K. Dykes, E. Lantz, S. Barth, C. L. Bottasso, O. Carlson, A. Clifton, J. Green, P. Green, H. Holttinen, D. Laird, V. Lehtomäki, J. K. Lundquist, J. Manwell, M. Marquis, C. Meneveau, P. Moriarty, X. Munduate, M. Muskulus, J. Naughton, L. Pao, J. Paquette, J. Peinke, A. Robertson, J. S. Rodrigo, A. M. Sempreviva, J. C. Smith, A. Tuohy, and R. Wiser, "Grand challenges in the science of wind energy," *Science*, vol. 366, no. 6464, p. eaau2027, 2019.

[5] D. GL, "Energy transition outlook 2018: A global and regional forecast of the energy transition to 2050," Tech. Rep., 2018.

[6] G. W. E. Council, "51.3 gw of global wind capacity installed in 2018," Tech. Rep., 2019.

[7] B. Kroposki, B. Johnson, Y. Zhang, V. Gevorgian, P. Denholm, B.-M. Hodge, and B. Hannegan, "Achieving a 100% renewable grid: Operating electric power systems with extremely high levels of variable renewable energy," *IEEE Power and Energy Magazine*, vol. 15, no. 2, pp. 61–73, 2017.

[8] M. Ahlstrom, E. Ela, J. Riesz, J. O'Sullivan, B. F. Hobbs, M. O'Malley, M. Milligan, P. Sotkiewicz, and J. Caldwell, "The evolution of the market: Designing a market for high levels of variable generation," *IEEE Power and Energy Magazine*, vol. 13, no. 6, pp. 60–66, 2015.

[9] T. Ackermann, T. Prevost, V. Vittal, A. J. Roscoe, J. Matevosyan, and N. Miller, "Paving the way: A future without inertia is closer than you think," *IEEE Power and Energy Magazine*, vol. 15, no. 6, pp. 61–69, 2017.

[10] R. B. Stull, *An introduction to boundary layer meteorology*. Springer, 1988.

[11] M. Calaf, C. Meneveau, and J. Meyers, "Large eddy simulation study of fully developed wind-turbine array boundary layers," *Physics of Fluids*, vol. 22, no. 1, p. 015110, 2010.

[12] N. O. Jensen, "A note on wind generator interaction," Risø National Laboratory, Tech. Rep. Risø-M-2411, 1983.

[13] M. Bastankhah and F. Porté-Agel, "A new analytical model for wind-turbine wakes," *Renewable Energy*, vol. 70, pp. 116–123, 2014.

[14] L. Tian, W. Zhu, W. Shen, N. Zhao, and Z. Shen, "Development and validation of a new two-dimensional wake model for wind turbine wakes," *Journal of Wind*

Engineering and Industrial Aerodynamics, vol. 137, pp. 90–99, 2015.

[15] J. Annoni, P. Fleming, A. Scholbrock, J. Roadman, S. Dana, C. Adcock, F. Porte-Agel, S. Raach, F. Haizmann, and D. Schlipf, "Analysis of control-oriented wake modeling tools using lidar field results," *Wind Energy Science*, vol. 3, pp. 819–831, 11 2018.

[16] M. Ge, Y. Wu, Y. Liu, and X. I. Yang, "A two-dimensional jensen model with a gaussian-shaped velocity deficit," *Renewable Energy*, vol. 141, pp. 46–56, 2019.

[17] R. J. Barthelmie, L. Folkerts, F. T. Ormel, P. Sanderhoff, P. J. Eecen, O. Stobbe, and N. M. Nielsen, "Offshore wind turbine wakes measured by sodar," *J. Atmos. Oceanic Technol.*, vol. 20, p. 466–477, 2003.

[18] G. V. Iungo and F. Porté-Agel, "Volumetric lidar scanning of wind turbine wakes under convective and neutral atmospheric stability regimes," *J. Atmos. Oceanic Technol.*, vol. 31, pp. 2035–2048, 2014.

[19] H. Y. Zilong Ti, Xiao Wei Deng, "Wake modeling of wind turbines using machine learning," *Applied Energy*, vol. 257, 2020.

[20] A. Niayifar and F. Porté-Agel, "A new analytical model for wind farm power prediction," *Journal of Physics: Conference Series*, vol. 625, p. 012039, 2015.

[21] ——, "Analytical modeling of wind farms: A new approach for power prediction," *Energies*, vol. 9, no. 9, p. 741, 2016.

[22] C. R. Shapiro, G. M. Starke, D. F. Gayme, and C. Meneveau, "A wake modeling paradigm for wind farm design and control," *Energies*, vol. 12, no. 15, p. 2956, Aug 2019.

[23] H. Zong and F. Porté-Agel, "A momentum-conserving wake superposition method for wind farm power prediction," *Journal of Fluid Mechanics*, vol. 889, p. A8, 2020.

[24] I. Katic, J. Højstrup, and N. O. Jensen, "A simple model for cluster efficiency," in *European Wind Energy Association Conference and Exhibition*, 1986, pp. 407–410.

[25] R. J. Templin, "An estimation of the interaction of windmills in widespread arrays," National Aeronautical Establishment, Tech. Rep. LTR-LA-171, 1974.

[26] S. Frandsen, "On the wind speed reduction in the center of large clusters of wind turbines," *Journal of Wind Engineering and Industrial Aerodynamics,*, vol. 39, pp. 251–265, 1992.

[27] C. Meneveau, "The top-down model of wind farm boundary layers and its applications," *Journal of Turbulence*, vol. 13, p. N7, 2012.

[28] A. Peña and O. Rathmann, "Atmospheric stability-dependent infinite wind-farm models and the wake-decay coefficient," *Wind Energy*, vol. 17, no. 8, pp. 1269–1285, 2013.

[29] A. Sescu and C. Meneveau, "Large-eddy simulation and single-column modeling of thermally stratified wind turbine arrays for fully developed, stationary atmospheric conditions," *Journal of Atmospheric and Oceanic Technology*, vol. 32, no. 6, pp. 1144–1162, 2015.

[30] M. Abkar and F. Porté-Agel, "A new wind-farm parameterization for large-scale atmospheric models," *Journal of Renewable and Sustainable Energy*, vol. 7, 2015.

[31] S. Frandsen, R. Barthelmie, S. Pryor, O. Rathmann, S. Larsen, J. Højstrup, and

M. Thøgersen, "Analytical modelling of wind speed deficit in large offshore wind farms," *Wind Energy*, vol. 9, no. 1-2, pp. 39–53, 2006.

[32] R. J. A. M. Stevens, D. F. Gayme, and C. Meneveau, "Coupled wake boundary layer model of wind-farms," *Journal of Renewable and Sustainable Energy*, vol. 7, no. 2, p. 023115, 2015.

[33] ——, "Generalized coupled wake boundary layer model: applications and comparisons with field and LES data for two wind farms," *Wind Energy*, vol. 19, no. 11, pp. 2023–2040, 2016.

[34] F. Blondel and M. Cathelain, "An alternative form of the super-gaussian wind turbine wake model," *Wind Energ. Sci. Discuss.*, in review, 2020.

[35] F. Porté-Agel, Y.-T. Wu, and C.-H. Chen, "A numerical study of the effects of wind direction on turbine wakes and power losses in a large wind farm," *Energies*, vol. 6, no. 10, pp. 5297–5313, 2013.

[36] W. Munters, C. Meneveau, and J. Meyers, "Turbulent inflow precursor method with time-varying direction for large-eddy simulations and applications to wind farms," *Boundary-Layer Meteorology*, vol. 159, pp. 305–328, 2016.

[37] A. Rott, B. Doekemeijer, J. Seifert, J. W. Wingerden, and M. Kühn, "Robust active wake control in consideration of wind direction variability and uncertainty," *Wind Energy*, vol. 3, pp. 869–882, Sept 2018.

[38] E. Simley, P. Fleming, and J. King, "Design and analysis of a wake steering controller with wind direction variability," *Wind Energy Sci.*, vol. 5, no. 2, pp. 451–468, 2020.

[39] E. G. Antonini, D. A. Romero, and C. H. Amon, "Improving CFD wind farm simulations incorporating wind direction uncertainty," *Renewable Energy*, vol. 133, pp. 1011 – 1023, 2019.

[40] M. Gaumond, P.-E. Réthoré, S. Ott, A. Peña, A. Bechmann, and K. S. Hansen, "Evaluation of the wind direction uncertainty and its impact on wake modeling at the Horns Rev offshore wind farm," *Wind Energy*, vol. 17, no. 8, pp. 1169–1178, 2014.

[41] T. Chatterjee, N. W. Cherukuru, Y. T. Peet, and R. J. Calhoun, "Large eddy simulation with realistic geophysical inflow of alpha ventus wind farm: a comparison with LIDAR field experiments," *Journal of Physics: Conference Series*, vol. 1037, p. 072056, jun 2018.

[42] A. Stieren, S. Gadde, and R. J. A. M. Stevens, "Modeling dynamic wind direction changes in large eddy simulations of wind farms," *Renewable Energy*, 2021.

[43] R. S. Arthur, J. D. Mirocha, N. Marjanovic, B. D. Hirth, J. L. Schroeder, S. Wharton, and F. K. Chow, "Multi-scale simulation of wind farm performance during a frontal passage," *Atmosphere*, vol. 11, no. 3, 2020.

[44] C. R. Shapiro, D. F. Gayme, and C. Meneveau, "Modelling yawed wind turbine wakes: a lifting line approach," *Journal of Fluid Mechanics*, vol. 841, 2018.

[45] Á. Jiménez, A. Crespo, and E. Migoya, "Application of a LES technique to characterize the wake deflection of a wind turbine in yaw," *Wind Energy*, vol. 13, no. 6, pp. 559–572, 2010.

[46] M. Bastankhah and F. Porté-Agel, "Experimental and theoretical study of wind turbine wakes in yawed conditions," *Journal of Fluid Mechanics*, vol. 806, pp. 506–541, 2016.

[47] M. F. Howland, S. K. Lele, and J. O. Dabiri, "Wind farm power optimization through wake steering," *Proceedings of the National Academy of Sciences*, vol. 116, no. 29, pp. 14 495–14 500, 2019. [Online]. Available: https://www.pnas.org/content/116/29/14495

[48] B. Dou, T. Qu, L. Lei, and P. Zeng, "Optimization of wind turbine yaw angles in a wind farm using a three-dimensional yawed wake model," *Energy*, vol. 209, p. 118415, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S036054422031522X

[49] M. Bastankhah, C. R. Shapiro, S. Shamsoddin, D. F. Gayme, and C. Meneveau, "A vortex sheet based analytical model of the curled wake behind yawed wind turbines," p. A2, 2022.

[50] D. Wei, W. Zhao, D. Wan, and Q. Xiao, "A new method for simulating multiple wind turbine wakes under yawed conditions," *Ocean Engineering*, vol. 239, p. 109832, 2021.

[51] C. J. Bay, J. Annoni, T. Taylor, L. Pao, and K. Johnson, "Active power control for wind farms using distributed model predictive control and nearest neighbor communication," in *American Control Conference*, 2018.

[52] M. F. Howland, A. S. Ghate, S. K. Lele, and J. O. Dabiri, "Optimal closed-loop wake steering – part 1: Conventionally neutral atmospheric boundary layer conditions," *Wind Energy Science*, vol. 5, no. 4, pp. 1315–1338, 2020. [Online]. Available: https://wes.copernicus.org/articles/5/1315/2020/

[53] L. Y. Pao and K. E. Johnson, "Control of wind turbines," *IEEE Control Systems*, vol. 31, no. 2, pp. 44–62, apr 2011.

[54] C. R. Shapiro, G. M. Starke, and D. F. Gayme, "Turbulence and control of wind farms," *Annual Review of Control, Robotics, and Autonomous Systems*, 2021.

[55] S. Raach, D. Schlipf, F. Haizmann, and P. W. Cheng, "Three dimensional dynamic model based wind field reconstruction from lidar data," *J. Phys.: Conference Series*, vol. 524, p. 012005, jun 2014.

[56] B. M. Doekemeijer, S. Boersma, L. Y. Pao, T. Knudsen, and J.-W. van Wingerden, "Online model calibration for a simplified LES model in pursuit of real-time closed-loop wind farm control," *Wind Energy Science*, vol. 3, no. 2, pp. 749–765, 2018.

[57] J. R. Annoni, C. Bay, K. E. Johnson, E. Dall'Anese, E. W. Quon, T. W. Kemper, and P. A. Fleming, "Wind direction estimation using SCADA data with consensus-based optimization," *Wind Energy Sci. (Online)*, vol. 4, no. 2, June 2019.

[58] J. Annoni, C. Bay, K. Johnson, and P. Fleming, "Short-term forecasting across a network for the autonomous wind farm," in *Proc. of the American Control Conf.*, 2019, pp. 2837–2842.

[59] M. Sinner, L. Y. Pao, and J. King, "Estimation of large-scale wind field characteristics using supervisory control and data acquisition measurements," in *2020 American Control Conference (ACC)*, 2020, pp. 2357–2362.

[60] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960.

[61] N. Moustakis, S. P. Mulders, J. Kober, and J.-W. van Wingerden, "A practical bayesian optimization approach for the optimal estimation of the rotor effective wind speed," in *2019 American Control Conference (ACC)*. Piscataway, NJ: IEEE, jul 2019, pp.

4179–4185.

[62] G. Evensen, "The ensemble kalman filter for combined state and parameter estimation," *IEEE Control Systems Magazine*, vol. 29, no. 3, pp. 83–104, 2009.

[63] B. M. Doekemeijer, S. Boersma, L. Y. Pao, and J. W. van Wingerden, "Ensemble kalman filtering for wind field estimation in wind farms," in *American Control Conference*, 2017, pp. 19–24.

[64] B. Doekemeijer, S. Boersma, L. Pao, and J. van Wingerden, "Joint state-parameter estimation for a control-oriented LES wind farm model," *J. Phys.: Conference Series*, vol. 1037, p. 032013, jun 2018.

[65] C. R. Shapiro, J. Meyers, C. Meneveau, and D. F. Gayme, "Dynamic wake modeling and state estimation for improved model-based receding horizon control of wind farms," in *American Controls Conference*, 2017, pp. 709–716.

[66] S. Boersma, B. Doekemeijer, S. Siniscalchi-Minna, and J. van Wingerden, "A constrained wind farm controller providing secondary frequency regulation: An LES study," *Renewable Energy*, vol. 134, pp. 639–652, 2019.

[67] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Wind Energy Handbook*. John Wiley & Sons, 2011.

[68] P. A. Fleming, P. M. Gebraad, S. Lee, J.-W. van Wingerden, K. Johnson, M. Churchfield, J. Michalakes, P. Spalart, and P. Moriarty, "Evaluating techniques for redirecting turbine wakes using SOWFA," *Renewable Energy*, vol. 70, pp. 211–218, 2014.

[69] J. Annoni, A. Scholbrock, M. Churchfield, and P. Fleming, "Evaluating tilt for wind plants," in *American Control Conference*, 2017, pp. 717–722.

[70] G. P. Corten, K. Lindenburg, and P. Schaak, "Assembly of energy flow collectors, such as windpark, and method of operation," Nov 2007.

[71] B. Biegel, D. Madjidian, V. Spudić, A. Rantzer, and J. Stoustrup, "Distributed low-complexity controller for wind power plant in derated operation," in *2013 IEEE International Conference on Control Applications (CCA)*, 2013, pp. 146–151.

[72] H. Zhao, Q. Wu, Q. Guo, H. Sun, and Y. Xue, "Distributed model predictive control of a wind farm for optimal active power controlpart i: Clustering-based wind turbine model linearization," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 3, pp. 831–839, 2015.

[73] S. Siniscalchi-Minna, F. D. Bianchi, and C. Ocampo-Martinez, "Predictive control of wind farms based on lexicographic minimizers for power reserve maximization," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 701–706.

[74] C. R. Shapiro, P. Bauweraerts, J. Meyers, C. Meneveau, and D. F. Gayme, "Model-based receding horizon control of wind farms for secondary frequency regulation," *Wind Energy*, vol. 20, no. 7, pp. 1261–1275, 2017.

[75] W. Munters and J. Meyers, "An optimal control framework for dynamic induction control of wind farms and their interaction with the atmospheric boundary layer," *Phil. Trans. R. Soc.*, vol. 375, 2017.

[76] J.-W. van Wingerden, L. Pao, J. Aho, and P. Fleming, "Active power control of waked wind farms," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4484–4491, 2017, 20th IFAC World Congress.

[77] P. M. O. Gebraad, F. W. Teeuwisse, J. W. van Wingerden, P. 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*, vol. 19, no. 1, pp. 95–114, 2016.

[78] P. A. Fleming, A. Ning, P. M. O. Gebraad, and K. Dykes, "Wind plant system engineering through optimization of layout and yaw control," *Wind Energy*, vol. 19, no. 22, 3 2015.

[79] F. Mühle, J. Schottler, J. Bartl, R. Futrzynski, S. Evans, L. Bernini, P. Schito, M. Draper, A. Guggeri, E. Kleusberg, D. S. Henningson, M. Hölling, J. Peinke, M. S. Adaramola, and L. Sætran, "Blind test comparison on the wake behind a yawed wind turbine," *Wind Energy Science*, vol. 3, no. 2, pp. 883–903, 2018.

[80] M. Bastankhah and F. Porté-Agel, "Wind farm power optimization via yaw angle control: A wind tunnel study," *J. Renew. Sustain. Ener.*, vol. 11, 2019.

[81] P. Fleming, J. Annoni, J. J. Shah, L. Wang, S. Ananthan, Z. Zhang, K. Hutchings, P. Wang, W. Chen, and L. Chen, "Field test of wake steering at an offshore wind farm," *Wind Energy Science*, vol. 2, no. 1, pp. 229–239, 2017.

[82] P. Fleming, J. King, K. Dykes, E. Simley, J. Roadman, A. Scholbrock, P. Murphy, J. K. Lundquist, P. Moriarty, K. Fleming, J. van Dam, C. Bay, R. Mudafort, H. Lopez, J. Skopek, M. Scott, B. Ryan, C. Guernsey, and D. Brake, "Initial results from a field campaign of wake steering applied at a commercial wind farm – part 1," *Wind Energy Science*, vol. 4, no. 2, pp. 273–285, 2019.

[83] F. Campagnolo, V. Petrović, J. Schreiber, E. M. Nanos, A. Croce, and C. L. Bottasso, "Wind tunnel testing of a closed-loop wake deflection controller for wind farm power maximization," *Journal of Physics: Conference Series*, vol. 753, p. 032006, sep 2016.

[84] N. Cassamo and J.-W. van Wingerden, "Model predictive control for wake redirection in wind farms: A koopman dynamic mode decomposition approach," in *2021 American Control Conference (ACC)*. Piscataway, NJ: IEEE, May 2021, pp. 1772–1778.

[85] W. Munters and J. Meyers, "Dynamic strategies for yaw and induction control of wind farms based on large-eddy simulation and optimization," *Energies*, vol. 11, no. 1, pp. 1–32, 2018.

[86] S. Boersma, B. Doekemeijer, M. Vali, J. Meyers, and J.-W. van Wingerden, "A control-oriented dynamic wind farm model: WFSim," *Wind Energy Science*, vol. 3, no. 1, pp. 75–95, 2018.

[87] R. J. A. M. Stevens, "Dependence of optimal wind turbine spacing on wind farm length," *Wind energy*, vol. 19, no. 4, pp. 651–663, 2016.

[88] M. Churchfield and S. Lee, "Nwtc design codes (simulator for offshore wind farm applications)," 2013.

[89] P. Fleming, S. Lee, M. Churchfield, A. Scholbrock, J. Michalakes, K. Johnson, P. Moriarty, P. Gebraad, and J. van Wingerden, "The sowfa super-controller: A high-fidelity tool for evaluating wind plant control approaches," in *EWEA Annual Meeting*, 2013.

[90] J. D. Albertson and M. B. Parlange, "Surface length scales and shear stress: Implications for land-atmosphere interaction over complex terrain," *Water Resource Research*, vol. 35, no. 7, pp. 212–2132, 1999.

[91] Johns Hopkins University, "Lesgo: A parallel pseudo-spectral large-eddy simulation code," https://lesgo.me.jhu.edu, 2019.

[92] L. Martinez, S. Leonardi, M. Churchfield, and P. Moriarty, *A Comparison of Actuator Disk and Actuator Line Wind Turbine Models and Best Practices for Their Use*, 2012.

[93] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-mw reference wind turbine for offshore system development," National Renewable Energy Laboratory, Tech. Rep. NREL/TP-500-38060, 2009.

[94] J. Thomas, J. Annoni, P. Fleming, and A. Ning, "Comparison of wind farm layout optimization results using a simple wake model and gradient-based optimization to large eddy simulations," in *AIAA Scitech 2019 Forum*, 01 2019.

[95] C. R. Shapiro, D. F. Gayme, and C. Meneveau, "Filtered actuator disks: Theory and application to wind turbine models in large eddy simulation," *Wind Energy*, vol. 22, July 2019.

[96] L. E. M. Lignarolo, D. Mehta, R. J. A. M. Stevens, A. E. Yilmaz, G. van Kuik, S. J. Andersen, C. Meneveau, C. J. Ferreira, D. Ragni, J. Meyers, G. J. van Bussel, and J. Holierhoek, "Validation of four les and a vortex model against stereo-piv measurements in the near wake of an actuator disc and a wind turbine," *Renewable energy*, vol. 94, pp. 510–523, 2016.

[97] R. J. A. M. Stevens, L. A. Martínez, and C. Meneveau, "Comparison of wind farm large eddy simulations using actuator disk and actuator line models with wind tunnel experiments," *Renew. Energy*, vol. 116, no. Part A, pp. 470 – 478, 2018.

[98] L. A. Martínez-Tossas, M. J. Churchfield, A. E. Yilmaz, H. Sarlak, P. L. Johnson, J. N. Sørensen, J. Meyers, and C. Meneveau, "Comparison of four large-eddy simulation research codes and effects of model coefficient and inflow turbulence in actuator-line-based wind turbine modeling," *Journal of Renewable and Sustainable Energy*, vol. 10, no. 3, p. 033301, 2018.

[99] R. J. Stevens, J. Graham, and C. Meneveau, "A concurrent precursor inflow method for large eddy simulations and applications to finite length wind farms," *Renewable Energy*, vol. 68, pp. 46–50, 2014.

[100] E. Bou-Zeid, C. Meneveau, and M. Parlange, "A scale-dependent lagrangian dynamic model for large eddy simulation of complex turbulent flows," *Physics of Fluids*, vol. 17, no. 2, p. 025105, 2005.

[101] G. M. Starke, P. Stanfel, C. Meneveau, D. F. Gayme, and J. King, "Network based estimation of wind farm power and velocity data under changing wind direction," in *2021 American Control Conference (ACC)*, 2021, pp. 1803–1810.

[102] J. Annoni, E. Dall'Anese, M. Hong, and C. J. Bay, "Efficient distributed optimization of wind farms using proximal primal-dual algorithms," in *Proc. of the American Control Conf.*, 2019, pp. 4173–4178.

[103] F. Bernardoni, U. Ciri, M. Rotea, and S. Leonardi, "Real-time identification of clusters of turbines," *J. Phys.: Conf. Series*, vol. 1618, p. 022032, Sep 2020.

[104] C. R. Shapiro, C. Ji, and D. F. Gayme, "Real-time energy market arbitrage via aerodynamic energy storage in wind farms," in *Proc. of the American Control Conf.*, 2020, pp. 4830–4835.

[105] A. Zaki, M. Attia, D. Hegazy, and S. Amin, "Comprehensive survey on dynamic graph models," *International J. of Advanced Computer Sci. and Applications*, vol. 7, Feb 2016.

[106] I. Maduako and M. Wachowicz, "A space-time varying graph for modelling places and events in a network," *International J. of Geographical Information Sci.*, vol. 33, no. 10, pp. 1915–1935, 2019.

[107] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Trans. on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.

[108] W. Zhu and D. Cheng, "Leader-following consensus of second-order agents with multiple time-varying delays," *Automatica*, vol. 46, no. 12, pp. 1994 – 1999, 2010.

[109] E. Bitar and P. Seiler, "Coordinated control of a wind turbine array for power maximization," in *Proc. of the American Control Conf.*, June 2013, pp. 2898–2904.

[110] G. M. Starke, C. Meneveau, J. R. King, and D. F. Gayme, "The Area Localized Coupled model for analytical mean flow prediction in arbitrary wind farm geometries," *Journal of Renewable and Sustainable Energy*, vol. 13, 6 2020.

[111] NREL, "FLORIS. Version 2.2.3," *GitHub repository*, 2020. [Online]. Available: https://github.com/NREL/floris

[112] A. Farrell, J. King, C. Draxl, R. Mudafort, N. Hamilton, C. J. Bay, P. Fleming, and E. Simley, "Design and analysis of a wake model for spatially heterogeneous flow," *Wind Energy Science*, vol. 6, no. 3, pp. 737–758, 2021. [Online]. Available: https://wes.copernicus.org/articles/6/737/2021/

[113] J. Meyers and C. Meneveau, "Large eddy simulations of large wind-turbine arrays in the atmospheric boundary layer," in *48th AIAA Aerospace Sciences Meeting*, 2010, p. 827.

[114] C. R. Shapiro, J. Meyers, C. Meneveau, and D. F. Gayme, "Wind farms providing secondary frequency regulation: Evaluating the performance of model-based receding horizon control," *Wind Energy Science*, vol. 3, pp. 11–24, 2018.

[115] R. P. Brent, *Algorithms for minimization without derivatives*, ser. Prentice-Hall series in automatic computation. Prentice-Hall, 1972.

[116] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes: the art of scientific computing*, 3rd ed. Cambridge University Press, 2007.

[117] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

Appendix A

Algorithms Used

A.1 The Ensemble Kalman Filter

The following is a brief overview of the Ensemble Kalman filter (EnKF) applied in a wake model framework, as discussed in [22] and extended from [65]. First, in order to apply the Ensemble Kalman filter, we must start with an array of states. In this case, the states consist of the wake velocity deficit along the center line of the wake for each turbine (an array of N_x points running through the center of each turbine in the streamwise direction) and the wake espansion coefficients for each turbine. These states are written as

$$\psi = \left[\delta \mathbf{u_1}^T, ..., \delta \mathbf{u_N}^T, k_1, ..., k_N\right]^T \quad \in \mathbb{R}^{N_s}, \tag{A.1}$$

where $\delta \mathbf{u_i}$ is a column vector representing the discretization of the wake deficit for the i^{th} turbine, k_i is the wake expansion coefficient for the i^{th} turbine and $N_s = (N_x + 1)N$, where N is the number of turbines. The column vector of the measured power output of each turbine is denoted by $\xi \in \mathbb{R}^N$.

The wind farm system is then governed by the discrete update equations

$$\begin{aligned} \psi_{k+1} &= \mathbf{f}(\psi_k, \mathbf{C}'_{T,k}) + \mathbf{B}\chi_k \\ \xi_\mathbf{k} &= \mathbf{h}(\psi_k, \mathbf{C}'_{T,k}) + \epsilon_k, \end{aligned} \tag{A.2}$$

where ψ_{k+1} and ξ_k are the updated states and measurements at times $k+1$ and k, respectively, ψ_k is the state vector at time k, $\mathbf{C}'_{T,k}$ is a vector of the local thrust

coefficients at each turbine and $\mathbf{f}(\psi_k, \mathbf{C}'_{T,k})$ and $\mathbf{h}(\psi_k, \mathbf{C}'_{T,k})$ are the temporal and spatial discretizations of the wake model outlined fully in [22]. The measurement and modeling errors are represented by $\epsilon \in \mathbb{R}^N$ and $\chi \in \mathbb{R}^N$, respectively, which are zero-mean white noise processes. The process noise is divided into tow vectors, reflecting the two state categories, $\chi = \left[\chi_{\delta u}^T, \chi_k^T\right]^T \in \mathbb{R}^{2N}$, where $\chi_{\delta u} \in \mathbb{R}^N$ has the variance $\sigma_{\delta u}^2$, and $\chi_k \in \mathbb{R}^N$ has the variance σ_k^2 both with zero mean.

While the identity matrix is often chosen for the matrix $\mathbf{B}$, to distribute the error to all the states, here we only want to apply one error correction to each wake deficit equation. This necessitates the error term having a lower dimensions enabling it to be distributed to each wake deficit field independently. This is accomplished by defining $\mathbf{B}$ as

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{\delta u} \\ \mathbf{I}_{N\times N} \end{bmatrix} \in \mathbb{R}^{N_s \times N}, \tag{A.3}$$

where $\mathbf{I}_{N\times N}$ is the identity matrix of size N and $\mathbf{B}_{\delta u}$ distributes $\chi_{\delta u}$ to the wake deficits. We assume that the wake deficits are uncoupled and the resulting noise is therefore only distributed to each turbine and uncoupled between turbines.

Since the true state is not available in this application, the EnKF estimates the error statistics of the true state using an ensemble of N_e perturbed wake models. The wake models are perturbed with statistically independent noise, represented by χ and ϵ. The ensemble is represented as

$$\Psi = \left[\psi^{(1)}, \quad \psi^{(2)}, \quad ..., \quad \psi^{(N_e)}\right] \in \mathbb{R}^{N_s \times N_e}, \tag{A.4}$$

where each $\psi^{(i)}$ represents one of the perturbed wake models. An ensemble of perturbed measurements is also generated from the measurements of the true system (the wind farm plant). Each perturbed measurement has the form

$$\xi^{(i)} = \mathbf{P} + \epsilon^{(i)}, \tag{A.5}$$

where $\mathbf{P}$ is the power measurements from the wind farm. This leads to an ensemble

of the form

$$\boldsymbol{\Xi} = \left[\xi^{(1)}, \quad ..., \quad \xi^{(N_e)}\right] \in \mathbb{R}^{N \times N_e}, \tag{A.6}$$

along with an ensemble of measurement perturbations

$$\mathbf{E} = \left[\epsilon^{(1)}, \quad ..., \quad \epsilon^{(N_e)}\right] \in \mathbb{R}^{N \times N_e}. \tag{A.7}$$

The nonlinear outputs of the ensemble of wake models is then represented as

$$\hat{\boldsymbol{\Psi}} = \left[\mathbf{h}(\psi^1), \quad ..., \quad \mathbf{h}(\psi^{N_e})\right] \in \mathbb{R}^{N \times N_e}. \tag{A.8}$$

The mean of the ensemble and output states, which are used in the error statistics calculation, are found using

$$\begin{aligned} \overline{\boldsymbol{\Psi}} &= \boldsymbol{\Psi}\mathbf{1}_{N_e} \in \mathbb{R}^{N_s \times N_e} \\ \overline{\hat{\boldsymbol{\Psi}}} &= \hat{\boldsymbol{\Psi}}\mathbf{1}_{N_e} \in \mathbb{R}^{N \times N_e}, \end{aligned} \tag{A.9}$$

where $\mathbf{1}_{N_e} \in \mathbb{R}^{N_e \times N_e}$ is a matrix where all of the elements are equal to $1/N_e$. The ensemble state perturbation matrix $\boldsymbol{\Psi}'$ has the form

$$\boldsymbol{\Psi}' = \boldsymbol{\Psi} - \bar{\boldsymbol{\Psi}} \in \mathbb{R}^{N_s \times N_e}, \tag{A.10}$$

and the ensemble output perturbation matrix $\hat{\boldsymbol{\Psi}}'$ has the

$$\hat{\boldsymbol{\Psi}}' = \hat{\boldsymbol{\Psi}} - \overline{\hat{\boldsymbol{\Psi}}} \in \mathbb{R}^{N_s \times N_e}. \tag{A.11}$$

Now the two previous update equations A.2 is replaced by the EnKF update equations. This process has two parts: first, an intermediate forecast step, shown as k_+, is completed using the equation

$$\boldsymbol{\Psi}_{k_+} = \left[f\left(\psi_k^{(1)}, C_{T,k}^{\prime(1)}\right) + \xi_k^{(1)}, ..., f\left(\psi_k^{(N_e)}, C_{T,k}^{\prime(N_e)}\right) + \xi_k^{(N_e)}\right], \tag{A.12}$$

and second, an analysis step that has the form

$$\boldsymbol{\Psi}_{k+1} = \boldsymbol{\Psi}_{k_+} + \boldsymbol{\Psi}'_{k_+}\hat{\boldsymbol{\Psi}}'^{T}_{k_+}\left(\hat{\boldsymbol{\Psi}}'_{k_+}\hat{\boldsymbol{\Psi}}'^{T}_{k_+} + \mathbf{E}_{k+1}\mathbf{E}_{k+1}^T\right)^{-1}\left(\boldsymbol{\Xi}_{k+1} - \hat{\boldsymbol{\Psi}}_{k_+}\right). \tag{A.13}$$

The estimated wake model states are then stored in the columns of the matrix $\overline{\boldsymbol{\Psi}}$, which represents the ensemble mean matrix.

A.2 Powell's Algorithm

Powell's Algorithm is a conjugate direction method which performs a minimization without calculating the derivatives. The algorithm is initialized using an initial point and an initial set of search vectors, represented by $s_1, ..., s_N$ for an N-dimensional problem. The method minimizes the function then iterates over the search directions, finding the minimum of the function in a line search over each search direction. The minima found in each bi-directional line search can be expresses as

$$x_0 + \alpha_1 s_1, x_0 + \sum_{i=1}^{2} \alpha_i s_i, ..., x_0 + \sum_{i=1}^{N} \alpha_i s_i, \tag{A.14}$$

where x_0 is the initial point and α_i is the scalar that determines the minimum in the s_i line search direction. The next point is then expressed as

$$x_1 = x_0 + \sum_{i=1}^{N} \alpha_i s_i. \tag{A.15}$$

A new direction is then added to the search directions, defined by $x_1 - x_0$, or $\sum_{i=1}^{N} \alpha_i s_i$, which can be thought of as the average direction moved during this iteration. A direction is also dropped from the search directions, to keep the search directions of size N. In Powell's modified algorithm, which is what is used in Python, the direction that achieved the larges decrease in the bi-directional minimization is dropped, if it satisfies specific conditions. This is to decrease the chances of linear dependence in the search directions, since the search direction that caused the larges change will also be a significant part of the new search direction. The conditions for the search directions to be kept the same is if (1) the function shows no great improvement along the direction of greatest increase in the iteration, or (2) the direction of the greatest improvement does not account for a sufficiently large fraction of the new search direction, which would indicate that several directions are contributing to the minimization.

The algorithm is iterated until there is no significant improvement to the objective function. [115–117]

Vita

Genevieve Starke was born in Reading, Pennsylvania in 1995 and graduated from high school in 2013. She received her B.S. in Aerospace Engineering from Syracuse University in 2017. She began her doctoral studies immediately after the completion of her undergraduate degree in 2017. She received a M.S.E. from Johns Hopkins University in 2020.