Power Reference Tracking in Wind Farms Through Distributive Model Predictive Control

V. van de Scheur

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Power Reference Tracking in Wind Farms Through Distributive Model Predictive Control

Master of Science Thesis

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

V. van de Scheur

March 6, 2020

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

Abstract

Because of the increasing influence of greenhouse gases on the environment, the world is transitioning from fossil fuels to renewable energy sources. Among these renewable energy sources, wind energy is one of the biggest contributors to the current power network. With the increasing use of wind energy as a power source, new challenges arise. One of these challenges is maintaining a stable power grid. On the power grid, the amount of energy generated and consumed should be in balance. As wind power has a unpredictable and fluctuating nature, it is presumable that its increasing use makes it problematic to maintain this balance. As most wind energy is generated at sites consisting of multiple wind turbines, called wind farms, this thesis focuses on these sites to overcome this problem. In this thesis model predictive control (MPC) strategies are introduced that not only stabilize the power produced by wind farms, but also create the possibility to perform power reference tracking with wind farms. With power reference tracking, it is possible for grid operators to adapt the power production to a change in the power demand and to counteract fluctuations introduced by other power generators. The model used within these controllers, is a control-oriented low-fidelity wind farm model, that is developed in this thesis. In this control model, the wake dynamics are taken into account. Wakes are areas downwind from turbines with decreased wind speed and increased turbulence. These wakes cause the wind turbines within a wind farm to influence each other. It is envisioned that taking these effects into account, will benefit the tracking quality of the controller. Because with centralized MPC it becomes problematic to provide real-time control for large wind farms due to the large order of the controller model required for such wind farms, also distributed controllers are proposed within this thesis. With distributed control, the central control problem is divided into smaller local control problems that will be solved on local controllers that communicate with each other. This makes it possible to also solve large complex control problems in real-time. An additional benefit of distributed control, is that if one of the local controllers fails, the rest of the wind farm can still be controlled. This makes the system more fail-safe. The control model and controllers introduced in this thesis are validated in a medium-fidelity wind farm model called WindFarmSimulator (WFSim) [1].

Table of Contents

List of Figures

List of Tables

Acknowledgements

I'm most grateful to ir. Sjoerd Boersma, Ph.D for his always sharp comments and support during this thesis. Even after finishing his Ph.D program at the TU Delft and moving to France, he continued to support me through Skype meetings. I'd also like to express my sincere gratitude to prof. dr. ir. J.W. van Wingerden for his support and providing the means necessary to finish this thesis. Furthermore, I'm grateful to my friends and family that always supported me through this thesis. And, lastly, I would also like to thank the committee members dr.ir. W.A.A.M. Bierbooms and ir. M.J. van den Broek for taking the time to scrutinize my work.

Delft, University of Technology
March 6, 2020

V. van de Scheur

Chapter 1

Introduction

With the increasing influence of greenhouse gases on the environment around the world, it is important to make the switch from fossil fuels to renewable energy sources. According to data from Eurostat [3], some countries, such as the Netherlands, still have a lot of work to do to make this switch. In 2016 only 6% of the Dutch gross final production of energy was from renewable sources. Fortunately, a lot of countries are on the way to become less dependent on fossil fuels. Among the renewable energy sources, wind energy is one of the biggest contributors to the current power network. In the EU-28 in 2016, 9.6 % of the electricity was produced from wind power [3]. Only hydro power had a bigger share with 10.8% [3].

The Need for Active Power Control (APC)

With wind power becoming a bigger source of energy, new challenges arise. As explained by the North American Electric Reliability Corporation (NERC) [4], the power demand on the electricity grid is constantly fluctuating. Power sources and loads should be able to counteract such fluctuations in order to keep the power line frequency constant. This is called frequency control. According to the NERC, for most sources it is easy to provide frequency control services. However, as stated by Ela et al. [5], for wind turbines it is more difficult. The amount of power they supply is dependent on the wind speed. They explain that, because of the fluctuating nature of the wind speed, the available power at wind turbines is constantly fluctuating. Because of this, it is not only difficult for wind turbines to provide frequency control services, but they even increase the instability on the grid [5]. Ela et al. state that, with the current share of electricity from wind turbines, this is not yet a problem. However, with the increasing amount of wind turbines, this can become a problem in the future [5]. A solution is not only stabilizing the power output from wind turbines, but even letting wind farms provide frequency control services via (advanced) control methods. Methods that strive to do this by controlling the power output, are called active power control (APC) methods. According to Ela et al., APC can cause wind energy to become a more attractive source of energy. However, such APC architectures can be difficult to design. This is not only caused

by the fluctuating nature of the available power at the turbines, but also because most wind turbines are placed together in so called wind farms. In Europe these wind farms nowadays consist of up to 175 turbines [6]. In these wind farms the wind turbines influence each other via their wakes as can be seen in figure 1-1. Wakes are regions, which can be found downwind the turbines, having increased turbulence and decreased wind speed, also called wind speed deficit. Modeling these wakes is an active field of research since not all wake dynamics are precisely known yet. These wakes make the dynamics within a wind farm complex and make it challenging to design an APC controller for wind farms. More research is needed in order to design a proper functioning APC controller for wind farms.

Figure 1-1: Photography of Horn Rev 1 offshore wind farm [2].

Because of these reasons, this thesis will focus on APC or more specifically wind farm power reference tracking.

Literature on APC

Multiple papers have been written in which APC controllers are designed. Hierarchical control is among the popular choices. In these controllers a central controller will generate a control signal that is equal to the total desired power output of the wind farm. This control signal is divided over the individual wind turbines. At these turbines local controllers will make sure that the control signals are tracked by the wind turbines.

One of such controllers is designed by A.D. Hansen et al. [7]. In this controller the control signal, coming from a central proportional-integral (PI) controller, is distributed over the wind turbines proportional to the available power at each wind turbine. The available power is the maximum amount of energy that can be generated at a given wind speed. Similar controllers have been designed by H. Badihi et al. [8]. H. Badihi et al. proposed both a fuzzy gain scheduled PI and an adaptive pole placement controller as central controller to account for the non-linear behavior of wind farms. Another hierarchical controller has been designed by J.W. van Wingerden et al. [9]. Here the control signal is divided equally over the wind turbines. The central controller is a gain scheduled PI controller of which the gain is increased when turbines reach their maximum available power. In this way the other turbines will compensate for the lower power production of these turbines. All of these controllers have been tested successfully in simulation models.

Another popular control architecture for power reference tracking in wind farms is model predictive control (MPC). MPC relies on models of the to be controlled subject, in order to find optimal control actions. An advantage of such a controller is that it is possible to apply constraints on control signals and states. Furthermore, it is possible to put a weight on different variables such that one controller can focus on multiple goals. Additionally, as it takes time for wake effects to travel from wind turbine to wind turbines, the dynamics between the wind turbines are delayed. As MPC can cope well with such delayed dynamics, it is possible to take the interaction between the wind turbines into account by including a wake model in the controller model. It is envisioned that taking these interactions into account will be beneficial for the tracking quality of the controller. Because of these reasons, MPC is seen as a very promising control technique for APC in wind farms.

One of such controllers has been designed by S. Boersma et al. [10]. This controller consists of two control loops. One focuses on maximizing the available energy in the wind farm by yawing the rotors, whilst using the wind farm model called FLOw Redirection and Induction in Steady-state (FLORIS). This model takes the steady-state wake effects into account [11]. The other loop focuses on power reference tracking, whilst keeping the change in trust force to a minimum to reduce the fatigue damage. This second loop does not utilize a wake model, but models the farm with uncoupled parameter varying wind turbine models with the wind as scheduling variable. This controller has been validated in a Large Eddy Simulation (LES) study. Also S. Siniscalchi-Minna et al. [12] designed a control scheme utilizing MPC. This MPC scheme focuses on both maximizing the available power in the wind farm and minimizing the tracking error. This controller has been applied to a simplified wind farm model. However, both controllers do not utilize a wake model to obtain the optimal control sequence that will minimize the reference tracking error. This means that the interaction between the wind turbines is not taken into account.

Because it is interesting to research if taking the wake dynamics into account will improve the tracking quality of the controller, C.R. Shapiro et al. designed a MPC scheme in which a wake model is used for reference tracking [13]. This controller was tested successfully in a high fidelity model. With this controller the wind farm was able to temporally produce more power than it would have produced when all the turbines would operate at their individual theoretical optimum, i.e. the so called Betz limit. This amount of power will be denoted by P^{greedy}. Before a reference exceeds this P_{greedy}, the controller decreases the power extraction of the upwind turbines and increases the power extraction of the downwind turbines. In this way, more wind power will flow into the farm. This power will then be available to the downwind turbines at the time at which P_{greedy} is exceeded. Because of this, less de-rating is needed with this controller. However, also controllers without wake models, for example the controllers by S. Boersma, were able to exceed this limit for a limited amount of time. More research is needed to prove that including the wake effects in the controller will lead to a controller that requires less de-rating and is better able to follow reference signals that exceed P^{greedy}. Interesting to point out is that C.R. Shapiro et al. also designed a MPC in which the rotational kinetic energy of the rotors is modeled [14]. This makes it possible for wind turbines to store energy in their rotors. It is expected that doing so will cause the wind farm will be able to exceed P^{greedy} even more, such that even less de-rating is needed. Unfortunately, this controller has only been tested in the same model as the controller model [14].

In the controllers mentioned above centralized control is used. According to T. Knudsen et

al. [15], it is interesting to use distributed control in wind farms, as for large wind farms centralized control can become too computationally complex for real-time applications. It is stated that distributed control can reduce the complexity of the global control problem. The problem is separated over multiple controllers, thus requiring less computational resources per controller. Furthermore, distributed control is more fail safe [15]. With centralized control, if the central controller fails, the entire system fails. If one of the distributed controllers fails, the system can still be controlled. Also distributed controlled wind farms are easier to expand [15]. However, the performance of distributed control can be worse than that of centralized control [16]. Therefore, it is important that, if distributed control is used, this loss in performance is kept to an acceptable limit. Because of the reasons mentioned above, also multiple controllers using distributed control can be found in literature.

One such controller is a distributed H_2 controller designed by M. Soleimanzadeh et al. The controller has been tested in the same model that is used to synthesize the controller. Stronger conclusions can be made on the performance of the control architecture, if the controller is tested in a different model. However, the model did perform well in the model in which it was tested.

In multiple papers on APC, distributed model predictive control (DMPC) is used. H. Zhao et al. proposed a DMPC control scheme for a wind farm equipped with an energy storage system. A disadvantage of such a system, is that an energy storage system can be a costly and environmentally unfriendly factor. V. Spudić et al. [17] has also proposed a DMPC scheme. In this scheme no energy storage system was used. In both the controller by H.Zhao et al. and V. Spudić et al. no wake model has been included, which means that the interaction between the turbines is not taken into account. In a paper by C.J. Bay et al., a distributed controller that does take these effects into account has been designed. This controller has been tested in a low fidelity wind farm model. In these simulations, tracking errors were noticed. The validity of the wake model used in the controller may be compromised by the approximations and assumptions made to derive it. As the controller has only been tested in a low fidelity wind farm model, it is hard to make conclusions on the performance in real life conditions.

Goal of This Thesis

As stated above, previously a distributed model predictive control (DMPC) architecture that incorporates a dynamic wake model has been designed. However, to the best knowledge of the author of this thesis, no such controller has been designed that is successfully tested in a medium or high fidelity model. Therefore and because of the reasons stated above, this thesis will focus on designing such a controller and validating it in a medium or high fidelity model. Furthermore, it is also desired by the author of this thesis that the controller is able to provide real-time control. In the context of this thesis, real-time means that the time to update the control signals is smaller than the sample time of the controller.

To achieve this goal, in this thesis, not only the controller is designed, but also the controller model used in this controller. This model is a low-fidelity control-oriented wind farm model. This model is tested against data from the medium fidelity model called WindFarmSimulator (WFSim) [1]. This model will be used as a controller model in a centralized and distributed model predictive controller. These controllers are also validated in WFSim.

In chapter 2, the wind farm model is derived and validated, in chapter 3, the controllers are designed, in chapter 4 the controllers are tested in WFSim and compared to a MPC controller designed by S. Boersma et al. [10], in chapter 5 the conclusions that can be drawn from this the results are stated and, finally, in chapter 6 the proposed controller model and controller are discussed and recommendations are made.

Chapter 2

Wind Farm Model

As model predictive control (MPC) utilizes a controller model to estimate future system outputs in order to obtain optimal control sequences, a wind farm model is needed to create a MPC architecture for active power control (APC) in wind farms. In this chapter such a model is sought. First of all, requirements are set on the controller model. After this, it is researched if a controller model that conforms to these requirements can be found in literature or that a new control-oriented model needs to be developed. Through this research, it is decided that a new control-oriented model will be developed. The rest of this chapter will focus on the development and validation of this model.

Requirements

Certain requirements are set on the controller model. First of all, the model is preferred to be linear, as this would create the opportunity to use linear MPC techniques. These are generally computationally faster than non-linear MPC techniques [18]. Because the goal of the thesis is to achieve real-time control, the computational speed of the algorithm is important. Also, linear MPC techniques are able to guarantee global optimality, whereas non-linear MPC techniques are generally not able to do so.

To achieve real-time control, the controller should be computationally fast. Since the controller contains a controller model, consequently also this model should be fast. Wind farm models can be classified into low, medium and high fidelity models, low fidelity meaning least precise and high fidelity meaning most precise. As high fidelity models have simulation times in the order of hours, medium fidelity in the order of seconds to minutes and low fidelity in the order of milliseconds to seconds, a low fidelity model is required for real-time control. Low computational complexity implies that the amount of states in the model should be limited. Also sparsity of used state-space matrices can contribute to low computational complexity.

Also, within a wind farm the air flow is mostly dynamic, i.e the wakes are under constant influence of wake meandering and turbulence. Furthermore, it takes time for the wake effects to travel through the wind farm, i.e. there is a time delay in the interaction between the wind

turbines. It is envisaged that the MPC controller can take advantage of dynamic these effects as shown in a paper by Shapiro et al. [13]. In this paper a controller is developed that takes the wake dynamics into account. This controller down-regulates the upwind turbines in the farm before an increase in the reference signal. In this way the wind speed in the wakes of the wind turbines will be higher, and thus more power will flow to the downwind turbines. As it takes time for the wakes to travel through the wind farm, this power will be available to the downwind turbines when the increase in the reference happens. Most low fidelity wind farms models, however, assume steady state conditions and, thus, include none of these dynamic effects. To take advantage of the wake dynamics in the same way as is done in the paper by Shapiro et al., it is deemed important that at least the time delay in these interaction is taken into account in the controller model.

In addition, as the goal of this thesis is to accurately track a power reference, it is necessary to have the power as the output of the model. Furthermore, were in reality mostly the blade pitch angle and the generator torque are used to control wind turbines, in this thesis, it is required that the input to the model is either the thrust coefficient C_T or the power coefficient C_P, as defined in the Actuator Disk Model (ADM) [19]. This will reduce the complexity of the problem at hand. Also, as shown by L.A. Martínez et al. in a Large Eddy Simulation (LES) study, the ADM model performs well on simulating far wake dynamics [20]. As the influence that the wind turbines within a wind-farm exert on each other is governed by the far wake dynamics, the ADM will be sufficient for this project. It is possible to see the controllers developed in this thesis as wind farm level controllers that work on top of local controllers that steer the turbines to the demanded C_T or C_P. A disadvantage to this, is that using C_T or C_P as input implies that the turbine dynamics can not be used to work in favor of the power tracking performance of the wind farm. An example of how the turbine dynamics could be used in such a way is given in a paper by Shapiro et al. [14]. In the controller developed in this paper, kinetic energy is stored in the rotors of the wind turbines to be able to temporally increase the power production of the wind farm when needed. Nevertheless, as the focus of this project is the effect of a wake model on the controller performance, it is chosen to use the ADM model. Future research could focus on also incorporating turbine dynamics in the wind farm controller. Lastly, the model should be able to give estimates of the power outputs that are accurate enough such that a MPC scheme, in which the model is used, can provide proper reference tracking.

Models Found in Literature

A low fidelity wind farm model is FLOw Redirection and Induction Dynamics (FLORIDyn) designed by P.M.O. Gebraad and J.W. van Wingerden based on the Jensen's model [21] and a wake deflection model introduced by Jiménez et al. [22, 23]. This model is able to capture the dynamical behavior of the air flow with the use of operational points. This model was validated against LES data for a 2 by 3 turbine wind farm. A dynamical wind farm model developed by C.R. Shapiro et al. is also based on the Jensen's model [13]. Also this model has been validated in against LES data for a wind farm consisting of 14 by 10 turbines. Another model developed by Soleimanzadeh et al. [24] is based on the spatial discretization of the linearized Navier-stokes equations. This model is able to approximate measurements taken from a real 1x5 wind farm. Unfortunately, all of these models contain non-linearities. Therefore, other models were sought However, models that complied with all

of the set requirements were not found. Because of this reason it has been decided to develop a new low fidelity linear dynamical wind farm model that can be employed in an MPC that can provide APC.

In section 2-1, this linear control-oriented model is developed. In section 2-2 the model is compared to WindFarmSimulator (WFSim), which is a medium-fidelity, control-oriented wind farm model based on Navier-Stokes equations [1]. WFSim is easily obtainable via GitHub, requires little computational power and it is validated successfully against LES data. WFSim itself is unfortunately too computational complex to use in the MPC scheme, as a farm of 9 turbines already uses up to 1000 states, which is high for a control-oriented model. The model will be used in a MPC algorithm for APC to achieve secondary frequency control on the electricity grid as will be detailed in chapter 3.

2-1 Model development

In this section, the control-oriented model will be introduced. The model will consist of a flow model and a turbine model. The flow model models the flow of air through the wind farm and the turbine model models the behavior of the wind turbine and the interaction with the air flow. The flow model is based on the Frandsen's flow model [25]. For the turbine model, the ADM [19] is used. The model is linearized and made dynamic using Taylor's frozen turbulence hypothesis [26]. The flow model is developed in subsection 2-1-1 and the turbine model in subsection 2-1-2. In subsection 2-1-3 the flow model and turbine model are combined. Finally, in 2-1-4 the model is written in a state-space notation that can be decomposed into smaller state-space systems to be used in the distributed controllers designed in chapter 3.

2-1-1 Flow model

First the flow model will be defined. The flow model is based on the Frandsen's model [25].

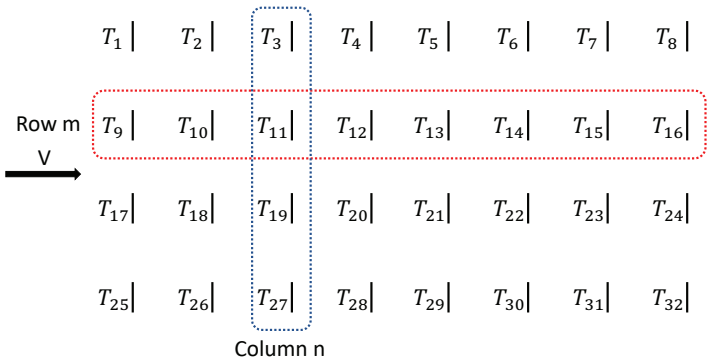

Figure 2-1: A wind farm can be divided into rows m and columns n. The wind turbines are depicted by the vertical black lines.

The model is defined for rectangular wind farms consisting of M parallel and equal rows and N parallel and equal columns of wind turbines as can be seen in figure 2-1. The wind turbines

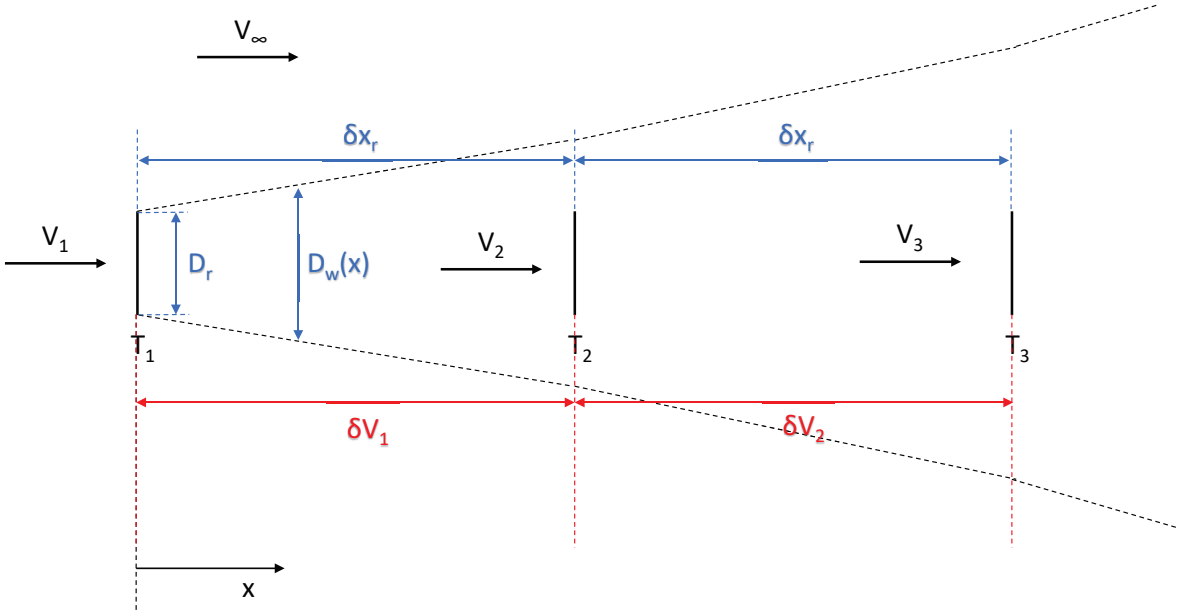

Figure 2-2: Close up on row of wind turbines in high waking conditions.

are depicted by the vertical black lines. The total number of wind turbines in the farm is denoted by $G = N \cdot M$. The rows are indexed with m, where $\{m \in \mathbb{Z} | 1 \leq m \leq M\}$, and the columns with n, where $\{n \in \mathbb{Z} | 1 \leq n \leq N\}$. The wind turbines T_i, $\{i \in \mathbb{Z} | 1 \leq i \leq G\}$, are numbered consecutively starting at the top row as can be seen in figure 2-1.

Within the proposed model, the interaction between the rows is ignored. Because of this, a separate model can be defined for each row m. These separate models can then be stacked together to form a model for the complete wind farm. Below the model will be defined for a single row m.

In figure 2-2, a close up of a part of one row is given. The wake zone is depicted as the area within the dotted black lines. The inflow wind speed into each wind turbine is given by V_i. This wind speed is defined as the wind speed just in front of the rotors, i.e. $V_i = V_{Ri}/(1-a_i)$, where V_{Ri} is the inflow wind speed defined at the rotor of turbine T_i and a_i is the induction factor at turbine T_i. V_∞ is the free-stream wind speed, i.e. the wind speed far in front of the wind farm. δV_i is the wind speed deficit (the amount by which the wind speed is decreased in the wake) between turbine T_i and T_{i+1}, i.e. $\delta V_i = V_i - V_{i+1}$. D_r is the diameter of the rotors and δx_r is the distance between the turbines. As can be seen the wake zone diameter $D_w(x)$ gets wider over distance x with a certain slope. This slope increases after each wind turbine depending on the local turbine setting, i.e. the thrust coefficients C_{Ti}.

The wind speed V_i at each downwind turbine in the row will be calculated as follows. The wind speed at the most upwind turbine, V_1, will be taken equal to V_∞ and is assumed to be known. From this wind speed the sum of the wind speed deficits created by all the turbines upwind form turbine T_i in the row is subtracted:

$$V_i = V_\infty - \sum_{j=1}^{i-1} \delta V_j. \tag{2-1}$$

To make this equation dependent on time, the Taylor frozen turbulence hypothesis [26] will be used:

$$\delta t_{j,i} = \frac{x_{j,i}}{V_\infty}, \tag{2-2}$$

in which $\delta t_{j,i}$ is the time it takes for the wake effects to travel from turbine T_j to turbine T_i and $x_{j,i}$ is the distance between turbine T_j and T_i. Using this assumption, equation 2-1 can be made dependent on time:

$$V_i(t) = V_\infty(t - \delta t_{1,i}) - \sum_{j=1}^{i-1} \delta V_j(t - \delta t_{j,i}). \tag{2-3}$$

It can be observed that also V_∞ became dependent on time. This can be problematic. Within MPC future output trajectories are approximated using the controller model. This means that, it would be necessary to know future values of V_∞. It is, however, difficult to make predictions on V_∞. Therefore, it is chosen to assume laminar inflow conditions, which is equal to making V_∞ constant over time. This means that equation 2-3 will transform into

$$V_i(t) = V_\infty - \sum_{j=1}^{i-1} \delta V_j(t - \delta t_{j,i}). \tag{2-4}$$

With this simplification, the model can be seen as an averaged model, i.e., it provides the averaged wind speeds. A possible way to also account for turbulent inflow conditions, is to model V_∞ as a stochastic variable. By doing so stochastic MPC strategies can be used as done in a paper by S. Boersma et al. [27]. This is, however, not considered in this thesis.

In discrete time, equation 2-4 can be written as

$$V_i[k] = V_\infty - \sum_{j=1}^{i-1} \delta V_j[k - d_{j,i}], \tag{2-5}$$

where the difference between time instance k and $k+1$ is equal to sample time h, $\delta V_j[k] = V_j[k] - V_{j+1}[k]$ and $d_{j,i}$ is the number of samples it takes for the wake effects to travel from turbine T_j to T_i, i.e.

$$d_{j,i} = \text{round}\left(\frac{\delta t_{j,i}}{h}\right). \tag{2-6}$$

The wind speed deficit $\delta V_j[k]$ will be calculated with the Frandsen's model [25]. This model will be linearized to get the linearized wind speed deficit $\delta \hat{V}_j[k]$. In the Frandsen's model, the wind speeds are determined by

$$\frac{V_{j+1}}{V_\infty} = 1 - \left(\frac{A_j}{A_{j+1}}\left(1 - \frac{V_j}{V_\infty}\right) + \frac{1}{2}\frac{A_R}{A_{j+1}} C_{Tj} \frac{V_j}{V_\infty}\right), \tag{2-7}$$

where A_R is the rotor area, given by $A_R = \frac{\pi}{4} D_R^2$ and A_j is the cross sectional area of the wake just in front of turbine T_j, which is calculated with $A_j = \frac{\pi}{4} D_w(x_j)^2$, where x_j is chosen just in front of turbine T_j. The formula can be rewritten as

$$V_{j+1} = V_\infty - \left(\frac{A_j}{A_{j+1}}(V_\infty - V_j) + \frac{1}{2}\frac{A_R}{A_{j+1}} C_{Tj} V_j\right). \tag{2-8}$$

Then, when adding time indices and using the definition for the wind speed deficit $\delta V_j[k] = V_j[k] - V_{j+1}[k]$, it can be derived that

$$\begin{aligned}\delta V_j[k] &= V_j[k] - V_{j+1}[k] \\ &= V_j[k] - V_\infty[k] + \frac{A_j[k]}{A_{j+1}[k]}\left(V_\infty[k] - V_j[k]\right) + \frac{1}{2}\frac{A_R}{A_{j+1}[k]} C_{Tj}[k] V_j[k].\end{aligned} \tag{2-9}$$

In the Frandsen's model, the difference in A_j between two wind turbines is given by

$$\delta A_j = A_{j+1} - A_j = \frac{1}{2} A_R \frac{c_w}{1 - c_w} C_{Tj}, \tag{2-10}$$

where $c_w = \frac{V_i}{V_\infty}\Big|_{i\to\infty}$, which is the wind speed at the most downwind turbine in an infinitely large row of wind turbines relative to the free stream wind speed V_∞. This value tends to converge asymptotically to a constant value and is only marginally dependent on the free stream wind speed. c_w can be determined for a specific wind farm by comparing the model against test data from that wind farm.

Equation 2-10 can be rewritten as

$$\begin{aligned}1 - \frac{A_j}{A_{j+1}} &= \frac{1}{2}\frac{A_R}{A_{j+1}}\frac{c_w}{1 - c_w} C_{Tj} \rightarrow \\ \frac{A_j}{A_{j+1}} &= 1 - \frac{1}{2}\frac{A_R}{A_{j+1}}\frac{c_w}{1 - c_w} C_{Tj},\end{aligned} \tag{2-11}$$

Substituting this into equation 2-9, gives

$$\begin{aligned}\delta V_j[k] &= V_j[k] - V_\infty[k] + \left(1 - \frac{1}{2}\frac{A_R}{A_{j+1}[k]}\frac{c_w}{1 - c_w} C_{Tj}\right)\left(V_\infty[k] - V_j[k]\right) + \frac{1}{2}\frac{A_R}{A_{j+1}[k]} C_{Tj}[k] V_j[k] \\ &= V_j[k] - V_\infty[k] + V_\infty[k] - V_j[k] + \frac{1}{2}\frac{A_R}{A_{j+1}[k]} C_{Tj}[k]\left(V_j[k] - \frac{c_w}{1 - c_w}\left(V_\infty[k] - V_j[k]\right)\right) \\ &= \frac{1}{2}\frac{A_R}{A_{j+1}[k]} C_{Tj}[k]\left(V_j[k] - \frac{c_w}{1 - c_w}\left(V_\infty[k] - V_j[k]\right)\right).\end{aligned} \tag{2-12}$$

$A_{j+1}[k]$ in this equation can be calculated using the relation

$$\begin{aligned}A_{j+1}[k] &= A_R + \sum_{l=1}^{j} \delta A_l(k - d_{l,j}) \\ &= A_R + \frac{1}{2} A_R \frac{c_w}{1 - c_w} \sum_{l=1}^{j} C_{Tl}(k - d_{l,j}).\end{aligned} \tag{2-13}$$

It is chosen not to substitute this equation into equation 2-12, as this would create the necessity to differentiate with respect to the delayed thrust coefficients of all turbines upwind from turbine T_{j+1} in order to linearize $\delta V_j[k]$. This would result in a more complicated equation. Instead, it is chosen differentiate equation 2-12 with respect to $V_j[k]$, $C_{Tj}[k]$ and A_{j+1}, i.e.

$$\begin{aligned}\delta\hat{V}_j[k] &= \delta V_{j,0} + \frac{\partial \delta V_j}{\partial V_j}\Big|_{x_0}\Delta V_j[k] + \frac{\partial \delta V_j}{\partial C_{Tj}}\Big|_{x_0}\Delta C_{Tj}[k] + \frac{\partial \delta V_j}{\partial A_{j+1}}\Big|_{x_0}\Delta A_{j+1}[k] \\ &= \delta V_{j,0} + \frac{1}{2}\frac{A_R}{A_{j+1,0}}C_{Tj,0}\left(1+\frac{c_w}{1-c_w}\right)\Delta V_j[k] \\ &\quad + \frac{1}{2}\frac{A_R}{A_{j+1,0}}\left(V_{j,0} - \frac{c_w}{1-c_w}\left(V_\infty - V_{j,0}\right)\right)\Delta C_{Tj}[k] \\ &\quad - \frac{1}{2}\frac{A_R}{A_{j+1,0}^2}C_{Tj,0}\left(V_{j,0} - \frac{c_w}{1-c_w}\left(V_\infty - V_{j,0}\right)\right)\Delta A_{j+1}[k],\end{aligned} \tag{2-14}$$

where x_0 is the linearization point, $\delta\hat{V}_j[k]$ is the linearized version of δV_j, ΔV_j is the deviation from the linearization point $V_{j,0}$, i.e. $\Delta V_j[k] = V_j[k] - V_{j,0}$, $\Delta C_{Tj}[k]$ is the deviation from the linearization point $C_{Tj,0}$, i.e. $\Delta C_{Tj}[k] = C_{Tj}[k] - C_{Tj,0}$ and ΔA_{j+1} is the deviation from the linearization point $A_{j+1,0}$, i.e. using equation 2-13

$$\begin{aligned}\Delta A_{j+1}[k] &= A_{j+1} - A_{j+1,0} \\ &= A_R + \frac{1}{2}A_R\frac{c_w}{1-c_w}\sum_{l=1}^{j}C_{Tl}(k-d_{l,j}) - A_{j+1,0},\end{aligned}$$

in which A_R is equal to $A_{1,0}$

Equation 2-14 and the linear version of equation 2-5, i.e.

$$\hat{V}_i[k] = V_\infty - \sum_{j=1}^{i-1}\delta\hat{V}_j[k-d_{j,i}], \tag{2-15}$$

together form the flow model for a single row of wind turbines.

2-1-2 Turbine model

Now that the flow model is defined and linearized, a turbine model is needed to model the interaction between the turbines and the air flow. It is chosen to use the ADM. This model approximates the power output of a wind turbine with:

$$P_i[k] = \frac{1}{2}\rho V_i^3[k]A_R C_{Pi}[k], \tag{2-16}$$

where ρ is the air density and $C_{Pi}[k]$ is the power coefficient of turbine T_i at time instance k. However, since in the flow model $C_{Ti}[k]$ and not $C_{Pi}[k]$ is used, it is practical to adapt equation 2-16. In the ADM it is stated that $C_P = C_T(1-a)$, where a is the axial induction factor. Using this relation it can be concluded that

$$P_i[k] = \frac{1}{2}\rho V_i^3[k]A_R C_{Ti}[k](1-a_i[k]), \tag{2-17}$$

where $a_i[k]$ is the induction factor at turbine T_i at time instant k.

$a_i[k]$ is taken as a constant equal to the value it has at the linearization point. Then, by differentiating equation 2-17 with respect to $V_i[k]$ and $C_{Ti}[k]$, the next equation is found:

$$\begin{aligned}\hat{P}_i[k] =& P_{i,0} + \frac{\partial \delta P_i}{\partial V_i}\Big|_{x_0} \Delta V_i[k] + \frac{\partial \delta P_i}{\partial C_{Ti}}\Big|_{x_0} \Delta C_{Ti} \\ =& P_{i,0} + \frac{3}{2}\rho V_{i,0}^2 A_R C_{Ti,0}(1-a_{i,0})\Delta V_i[k] \\ &+ \frac{1}{2}\rho V_{i,0}^3 A_R (1-a_{i,0})\Delta C_{Ti}[k],\end{aligned} \tag{2-18}$$

in which $\hat{P}_i$ is the linear version of P_i and $P_{i,0}$ and $a_{i,0}$ are the linearization points of P_i and a_i.

As can be seen, the input $C_{Ti}[k]$ influences the output $\hat{P}_i[k]$ directly, i.e. no turbine dynamics are taken into account. Following W. Munters and J. Meyers [28], in order to add an approximation of the turbine dynamics and to smooth the input signal a first order filter is added to the system:

$$\tau \frac{d\tilde{C}_{Ti}[t]}{dt} + \tilde{C}_{Ti}[t] = C_{Ti}[t], \tag{2-19}$$

in which τ is a time constant and $\tilde{C}_{Ti}[t]$ is the filtered version of $C_{Ti}[t]$. When discretized using a zero order hold, the filter is equal to

$$\tilde{C}_{Ti}[k+1] = e^{-\frac{1}{\tau}h}\tilde{C}_{Ti}[k] + \left(1 - e^{-\frac{1}{\tau}h}\right) C_{Ti}[k]. \tag{2-20}$$

This filter, the turbine model and the flow model will be combined in the next subsection.

2-1-3 Combining the flow and turbine model

In this subsection, the flow model and turbine model will be put together. First, to account for mismatches between the model and reality, the tuning variables $c_{VV}, c_{VC_T}, c_{VA}, c_{PV}$ and c_{PC_T} are added to the partial fractions from equations 2-14 and 2-18:

$$\begin{aligned}\frac{\partial \delta V_j}{\partial V_j}\Big|_{x_0} &= \frac{1}{2} c_{VV} \frac{A_R}{A_{j+1,0}} C_{Tj,0}\left(1 + \frac{c_w}{1-c_w}\right) \\ \frac{\partial \delta V_j}{\partial C_{T_j}}\Big|_{x_0} &= \frac{1}{2} c_{VC_T} \frac{A_R}{A_{j+1,0}}\left(V_{j,0} - \frac{c_w}{1-c_w}\left(V_\infty - V_{j,0}\right)\right) \\ \frac{\partial \delta V_j}{\partial A_{j+1}}\Big|_{x_0} &= -\frac{1}{2} c_{VA} \frac{A_R}{A_{j+1,0}^2} C_{Tj,0}\left(V_{j,0} - \frac{c_w}{1-c_w}\left(V_\infty - V_{j,0}\right)\right) \\ \frac{\partial \delta P_i}{\partial V_i}\Big|_{x_0} &= \frac{3}{2} c_{PV} \rho V_{i,0}^2 A_R C_{Ti,0}\left(1 - a_{i,0}\right) \\ \frac{\partial \delta P_i}{\partial C_{Ti}}\Big|_{x_0} &= \frac{1}{2} c_{PC_T} \rho V_{i,0}^3 A_R \left(1 - a_{i,0}\right).\end{aligned} \tag{2-21}$$

Now, by combining the definitions $\Delta V_i[k] = V_i[k] - V_{i,0}$, $\Delta C_{Ti}[k] = C_{Ti}[k] - C_{Ti,0}$ and

$$\Delta A_{i+1}[k] = A_R + \frac{1}{2} A_R \frac{c_w}{1-c_w} \sum_{l=1}^{i} C_{Tl}(k - d_{l,i}) - A_{i+1,0}$$

with equations 2-14, 2-15, 2-18 and 2-20, the total model becomes:

$$\begin{aligned}
\hat{V}_i[k] &= V_{\infty,0} - \sum_{j=1}^{i-1} \delta\hat{V}_j[k - d_{j,i}] \\
\delta\hat{V}_j[k] &= \frac{\partial\delta V_j}{\partial V_j}\Big|_{x_0}\hat{V}_j[k] + \frac{\partial\delta V_j}{\partial C_{Tj}}\Big|_{x_0}\tilde{C}_{Tj}[k] + \frac{\partial\delta V_j}{\partial A_{j+1}}\Big|_{x_0}\frac{1}{2}A_R\frac{c_w}{1-c_w}\sum_{l=1}^{j}\tilde{C}_{Tl}(k - d_{l,j}) \\
&\quad + c_{\delta V j} \\
\hat{P}_i[k] &= \frac{\partial\delta P_i}{\partial V_i}\Big|_{x_0}\hat{V}_i[k] + \frac{\partial\delta P_i}{\partial C_{Ti}}\Big|_{x_0}\tilde{C}_{Ti}[k] + c_{Pi} \\
\tilde{C}_{Ti}[k+1] &= e^{-\frac{1}{\tau}h}\tilde{C}_{Ti}[k] + (1 - e^{-\frac{1}{\tau}h})C_{Ti}[k],
\end{aligned} \tag{2-22}$$

where the partial fractions are given by equation 2-21 and the constant biases $c_{\delta V j}$ and c_{Pi} are given by:

$$c_{\delta V j} = \delta V_{j,0} - \frac{\partial\delta V_j}{\partial V_j}\Big|_{x_0}V_{j,0} - \frac{\partial\delta V_j}{\partial C_{Tj}}\Big|_{x_0}C_{Tj,0} + \frac{\partial\delta V_j}{\partial A_{j+1}}\Big|_{x_0}A_{1,0} - \frac{\partial\delta V_j}{\partial A_{j+1}}\Big|_{x_0}A_{j+1,0}$$

and

$$c_{Pi} = P_{i,0} - \frac{\partial\delta P_i}{\partial V_i}\Big|_{x_0}V_{i,0} - \frac{\partial\delta P_i}{\partial C_{Ti}}\Big|_{x_0}C_{Ti,0}.$$

Equation 2-22 gives the model for a single row m of wind turbines within a wind farm. To make the system valid for wind farms with an arbitrary number of rows, it is necessary to create such a system for each row m within the wind farm. As the dynamics between the rows of turbines is not considered, these systems can just be stacked together to form a model for the complete wind farm.

2-1-4 State-Space Model

To use the model within the MPC, it should be transformed into state-space notation. Because the model will also be used in a distributed MPC, it is important that it is possible to decompose it into smaller local subsystems. To achieve this goal, the notation as proposed in the book written by S. Li and Y. Zheng will be used [16]. First, separate state-space notations for the local subsystems, consisting of single turbines, will be derived. These state-space notations, together, form the so called decomposed state-space notation. These state-space notations can then be stacked together to form the centralized state-space notation, i.e. the state-space notation of the model of the complete system.

Lets call the complete system developed in the previous section S. If each wind turbine, T_i, is taken as a subsystem, it can be said that the system S consists of $G = N \cdot M$ subsystems S_i. Here, as explained before, M is the total number of rows and N is the total number of columns of wind turbines in the wind farm (see figure 2-1). The rows are indexed by m and the columns by n. As explained in subsection 2-1-1, the turbines and thus the subsystems are numbered consecutively starting at the top row. The most upwind subsystem in each

row m will be denoted S_{u_m} and the most downwind S_{dw_m}. This means that in a wind farm with 3 rows and 8 columns of turbines, the subsystems in the first row will be S_1 till S_8, in the second row S_9 till S_{16} and in the third row S_{17} till S_{24}, where $S_{u_1} = S_1$, $S_{u_2} = S_9$ and $S_{u_3} = S_{17}$ are the most upwind subsystems and $S_{dw_1} = S_8$, $S_{dw_2} = S_{16}$ and $S_{dw_3} = S_{24}$ are the most downwind subsystems.

The subsystems interact with each other via their states. To explain these interactions, two new terms are introduced: downstream and upstream. It is important to note that these terms are different from downwind and upwind. If the states of subsystem S_j are directly affected by subsystem S_i, it is said that subsystem S_j is downstream of subsystem S_i and subsystem S_i is upstream of subsystem S_j. Note the difference between the terms downstream and upstream and the terms downwind and upwind. All the turbines in a row m that are located behind a turbine T_i in the direction of the air flow are located downwind from T_i. For example, in figure 2-1, turbines T_{12} until T_{16} are located downwind from T_{11}. This does, however, not mean that the states of all the subsystems downwind from S_i are directly affected by S_i. This depends on how the states in the state-space system are designed. Thus, not all the subsystems downwind from S_i are necessarily downstream from S_i. It can be said that all the turbines behind a turbine are downwind and only the ones in that set that are affected are downstream. θ_{-i} will denote the set of subsystems downstream of S_i, θ_{+i} will denote the set of subsystems upstream of S_i and $\theta_i = \theta_{+i} \cup \theta_{-i}$. Using these sets it is possible to describe the subsystem S_i in state space notation as

$$\begin{aligned} x_i[k+1] &= A_{i,i}x_i[k] + B_i C_{Ti}[k] + \sum_{j \in \theta_{+i}} A_{i,j}x_j[k] + c_{xi} \\ \hat{P}_i[k] &= C_i x_i[k] + c_{Pi}, \end{aligned} \tag{2-23}$$

where x_i are the states of subsystem S_i, $A_{i,i} \in \mathbb{R}^{nx_i \times nx_i}$, $B_i \in \mathbb{R}^{nx_i \times 1}$ and $C_i \in \mathbb{R}^{1 \times nx_i}$ are the system matrices of subsystem S_i that connect the states $x_i[k]$ and inputs $C_{Ti}[k]$ to the states $x_i[k+1]$ and outputs $\hat{P}_i[k]$. nx_i is the amount of states defined for subsystem S_i. $A_{i,j} \in \mathbb{R}^{nx_i \times nx_j}$ connects the states of subsystems $S_j \in \theta_{+i}$ to S_i. No $B_{i,j}$ and $C_{i,j}$ are defined for $j \neq i$, because, in the control model defined in this chapter, the inputs and outputs of the subsystems do not interact directly with the states of other subsystems. c_{xi} and c_{Pi} are constant biases for S_i. The states, system matrices and constant biases are defined in detail in appendix A.

Now the sets θ_{+i} will be defined for each subsystem S_i. None of the most upwind subsystems, S_{u_m} $\forall\{m \in \mathbb{Z} | 1 \leq m \leq M\}$, are affected by any other subsystem. This means that it is possible to state that $\theta_{+u_m} = \varnothing$ $\forall\{m \in \mathbb{Z} | 1 \leq m \leq M\}$. Because of the way that the states, x_i, of the other subsystems, S_i $\forall\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \forall m\}$, are defined in appendix A, each of these subsystems are only directly affected by the subsystems directly upwind from it. This means that $\theta_{+i} = S_{i-1}$ $\forall\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \forall m\}$. An visual representation of an example of such sets is also given in figure 2-3 (Θ_{+i} and Θ_{-i} in this figure are defined later in this chapter). Using these sets, equation 2-23 can be transformed into

$$\begin{aligned} x_{u_m}[k+1] &= A_{u_m,u_m}x_{u_m}[k] + B_{u_m}C_{Tu_m}[k] + c_{xu_m} \\ \hat{P}_{u_m}[k] &= C_{u_m}x_{u_m}[k] + c_{Pu_m} \end{aligned} \tag{2-24}$$

for the most upwind subsystem, S_{u_m}, in each row m, where $\{m \in \mathbb{Z} | 1 \leq m \leq M\}$. For all the

other subsystems, S_i, equation 2-23 can be transformed into

$$\begin{aligned} x_i[k+1] =& A_{i,i}x_i[k] + A_{i,i-1}x_{i-1}[k] + B_i C_{Ti}[k] + c_{xi} \\ \hat{P}_i[k] =& C_i x_i[k] + c_{Pi}, \end{aligned} \tag{2-25}$$

where $\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \forall m\}$. This finalizes the derivation of the decomposed state-space notation.

These state-space notation can be stacked together to form the centralized state-space notation, i.e. the state space notation of the complete system S:

$$\begin{aligned} x[k+1] &= Ax[k] + BC_T[k] + c_x \\ \hat{P}[k] &= Cx[k] + c_y, \end{aligned} \tag{2-26}$$

where

$$x[k] = \begin{bmatrix} x_1[k] \\ x_2[k] \\ \vdots \\ x_G[k] \end{bmatrix}, \quad C_T[k] = \begin{bmatrix} C_{T1}[k] \\ C_{T2}[k] \\ \vdots \\ C_{TG}[k] \end{bmatrix}, \quad \hat{P}[k] = \begin{bmatrix} \hat{P}_1[k] \\ \hat{P}_2[k] \\ \vdots \\ \hat{P}_G[k] \end{bmatrix}, \quad B = \begin{bmatrix} B_{1,1} & 0 & \cdots & 0 \\ 0 & B_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{G,G} \end{bmatrix},$$

$$C = \begin{bmatrix} C_{1,1} & 0 & \cdots & 0 \\ 0 & C_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_{G,G} \end{bmatrix}, \quad c_x = \begin{bmatrix} c_{x1} & 0 & \cdots & 0 \\ 0 & c_{x2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{xG} \end{bmatrix}, \quad c_P = \begin{bmatrix} c_{P1} & 0 & \cdots & 0 \\ 0 & c_{P2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{PG} \end{bmatrix}$$

and

$$A = \begin{bmatrix} \mathbf{A}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{A}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_M \end{bmatrix},$$

in which $\mathbf{A}_m$ is a submatrix containing the submatrices $A_{i,j}$ for all the subsystems S_i within the row of turbines m. This matrix is defined as:

$$\mathbf{A}_m = \begin{bmatrix} A_{u_m,u_m} & 0 & \cdots & 0 & 0 & 0 \\ A_{u_m+1,u_m} & A_{u_m+1,u_m+1} & \cdots & 0 & 0 & 0 \\ 0 & A_{u_m+2,u_m+1} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & A_{dw_m-2,dw_m-2} & 0 & 0 \\ 0 & 0 & \cdots & A_{dw_m-1,dw_m-2} & A_{dw_m-1,dw_m-1} & 0 \\ 0 & 0 & \cdots & 0 & A_{dw_m,dw_m-1} & A_{dw_m,dw_m} \end{bmatrix}.$$

Note that $A_{i,j} = 0$ for all $A_{i,j}$ for which $S_j \notin \theta_{+i}$ and $j \neq i$. B and C are diagonal matrices, because, as explained above, within the controller model the outputs and inputs of subsystems do not interact directly with the states of other subsystems. This finalizes the derivation of the centralized state-space notation.

To design the controllers, it is necessary to explore the interaction between the subsystems more thoroughly. It is possible to create a so called adjacency matrix A_{adj} as described by S. Li and Y. Zheng [16]:

$$A_{adj} = \begin{bmatrix} \mathbf{A}_{adj\,1} & 0 & \cdots & 0 \\ 0 & \mathbf{A}_{adj\,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_{adj\,M} \end{bmatrix}, \tag{2-27}$$

where $\mathbf{A}_{adj\,m}$ is a $N \times N$ matrix and is defined for each row m as

$$\mathbf{A}_{adj\,m} = \begin{bmatrix} * & 0 & 0 & \cdots & 0 & 0 & 0 \\ * & * & 0 & \cdots & 0 & 0 & 0 \\ 0 & * & * & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & * & 0 & 0 \\ 0 & 0 & 0 & \cdots & * & * & 0 \\ 0 & 0 & 0 & \cdots & 0 & * & * \end{bmatrix}. \tag{2-28}$$

An $*$ in the ith row and j column expresses that S_i is directly impacted by S_j and, thus, that S_i is downstream from S_j and $S_i \in \theta_{-j}$.

So far, it has only been investigated if subsystems impact each other directly. It is also possible that subsystems impact each other indirectly via the states of other subsystems. If S_j can impact S_i directly and/or indirectly, it is said that S_j is accessible to S_i. The systems that are accessible to S_i will be denoted by Θ_{+i}, the systems to which S_i is accessible will be denoted by Θ_{-i} and $\Theta_i = \Theta_{+i} \cup \Theta_{-i}$. Note that $\theta_{+i} \in \Theta_{+i}$, $\theta_{-i} \in \Theta_{-i}$ and $\theta_i \in \Theta_i$. As described by S. Li and Y. Zheng [16], it is possible to determine these sets by looking at the powers of A_{adj}, i.e. the accessible relations are described by the accessibility matrix:

$$A_{acc} = A_{adj} \cup \cdots \cup A_{adj}^{G-1}. \tag{2-29}$$

This results in the accessibility matrix

$$A_{acc} = \begin{bmatrix} \mathbf{A}_{a_1} & 0 & \cdots & 0 \\ 0 & \mathbf{A}_{a_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_{a_M} \end{bmatrix}, \tag{2-30}$$

where $\mathbf{A}_{acc_m}$ is a $N \times N$ matrix and is defined for each row m within the windfarm as

$$\mathbf{A}_{acc_m} = \begin{bmatrix} * & 0 & \cdots & 0 & 0 \\ * & * & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & \cdots & * & 0 \\ * & * & \cdots & * & * \end{bmatrix}. \tag{2-31}$$

It can be deduced from this accessibility matrix that a subsystem S_i is directly and/or indirectly affected by all subsystems in the same row of turbines m that are located upwind from

S_i. Thus Θ_{+i} contains all subsystems upwind from S_i that are located in the same row of turbines as S_i. In figure 2-3 a visual representation of an example of the sets θ_{+i}, θ_{-i}, Θ_{+i} and Θ_{-i} is given for a wind farm consisting of 8 turbines in a single row.

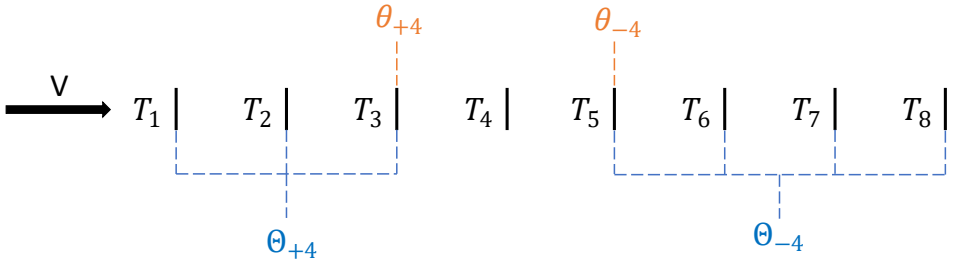

Figure 2-3: Visual representation of sets θ_{+4}, θ_{-4}, Θ_{+4} and Θ_{-4} in a 1x8 wind farm.

2-2 Model validation

The model has been implemented and tested in MATLAB and has been compared with data from WFSim. WFSim is a medium-fidelity, control-oriented wind farm model based on the two-dimensional Navier-Stokes equations. [1]

The controller model will be tested in a setup with $G = 10$ turbines. The 10 turbine (10T) setup can be seen in figure 2-4. The wind farm consists of $M = 2$ rows and $N = 5$ columns of turbines. The turbines have a diameter, D_r, of 90 meters. The distance between the turbines in the direction of the wind, δx_r, is 630 meters. This is equal to 7 turbine diameters, which is a common choice in wind farms. The distance between the turbines perpendicular to the wind, δy_r, is 378 meters. The free-stream wind speed, V_∞, is 7.5 m/s. Taking already into account the signals that will be tracked later, a sample period of $h = 1$ seconds is chosen. It is chosen to take $\tau = 5$, following W. Munters et al. [28]. The tuning variables, $c_{VV}, c_{VC_T}, c_{VA}, c_{PV}$ and c_{PC_T}, are tuned by putting a step change on the turbines and comparing the simulation results between the proposed model and WFSim. This resulted in the tuning variables that can be found in table 2-1. In this same table also the settings stated above and the settings used in WFSim can be found.

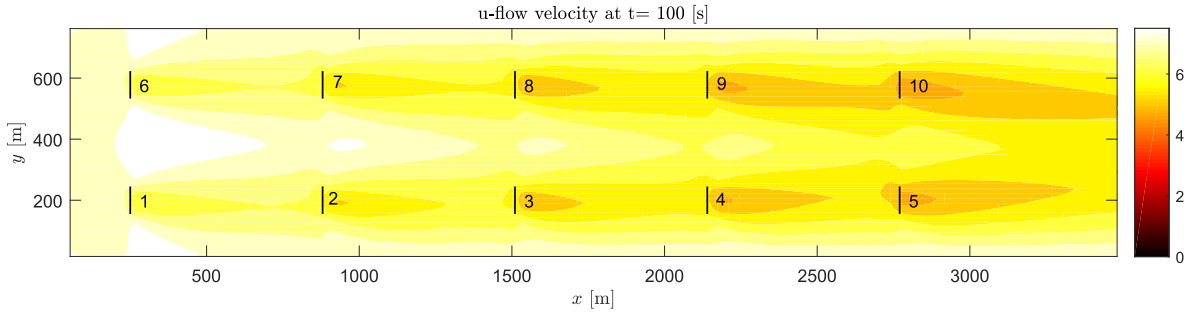

Figure 2-4: Visual representation of wind speed in 5x2 wind farm in WFSim.

For the validation of the model, different data sets and test signals are used than for the tuning. The proposed model will be compared with WFSim using the root mean square

Table 2-1: Simulation case details and settings for 10 turbine (10T).

	Variable	Value
Windfarm Setup	G	10
	M	5
	N	2
	δx_r	630 m
	δy_r	378 m
	D_r	90 m
	V_∞	7.5 m/s
Control Model Settings	c_w	0.68
	c_{VV}	1.0
	c_{VC_T}	1.0
	c_{VA}	0.9
	c_{PV}	1.0
	c_{PC_T}	1.1
	τ	5
	h	1

	Variable	Value
WFSim Settings	type	'lin'
	Lx	3500
	Ly	778
	Nx	100
	Ny	50
	powerscale	1
	forcescale	1.25
	mu	0
	Rho	1.2
	u_Inf	7.5
	v_Inf	0
	p_init	0
	lmu	2
	turbul	true
	n	2
	m	6

error (RMSE), i.e.

$$\text{RMSE} = \frac{\sum_{k=1}^{N_s}(P[k] - \hat{P}[k])^2}{Ns}, \tag{2-32}$$

in which N_s is the total number of samples in the simulation, $P[k]$ is the total power output predicted by WFSim and $\hat{P}[k]$ is the total power output predicted by the model introduced in this chapter at sample time k.

In the simulation case, the thrust coefficients, C_{Ti}, of the turbines, T_i, are initially kept constant. At 100 seconds there is a step change in C_{T1} and C_{T6}. Then, at 300 seconds there is a step change in C_{T2} and C_{T7}. Finally, at 500 seconds there is a step change in C_{T4} and C_{T8}.

In figure 2-5, 2-6 and 2-7 the results of this simulation can be witnessed. The RMSE is 0.15 MW. It can be concluded that although the predicted power output from both models is not identical, the proposed model is able to give an estimation of the power output of a wind farm simulated in WFSim. Propagating the system one time step takes $3.1 \cdot 10^{-04}$ seconds on a single core of the 2.3 GHz Intel(R) Core(QM) i7-3610QM. In chapter 4, the controller, in which the proposed controller model is used, is test. From these results it can be concluded that the controller is accurate and fast enough to be used within a MPC architecture.

In appendix B the model has also been compared to data from Parallelized Large-Eddy Simulation Model (PALM), which is a high-fidelity three-dimensional LES model. [29]

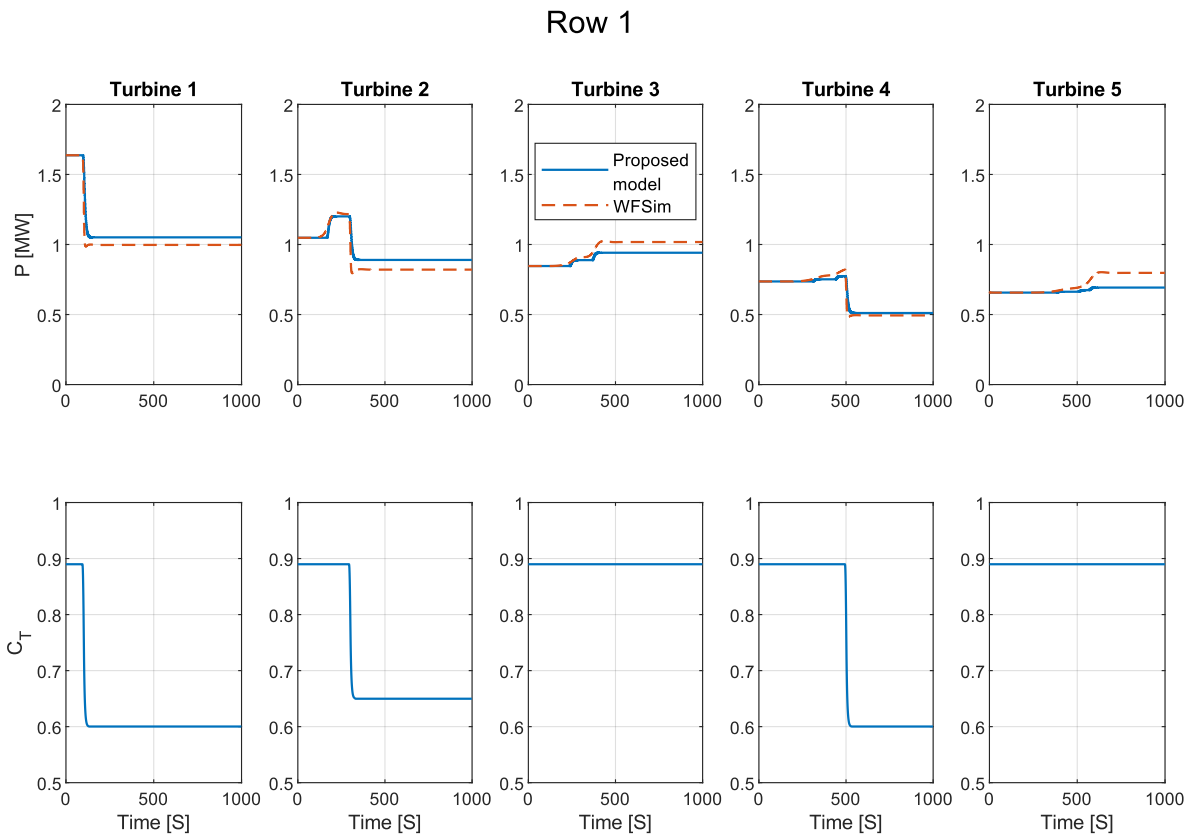

Figure 2-5: Comparison of power outputs of turbines in row 1 predicted by the proposed model (blue) compared to WFSim (red dashed) with validation data. The used input signals can be found in the bottom figures.

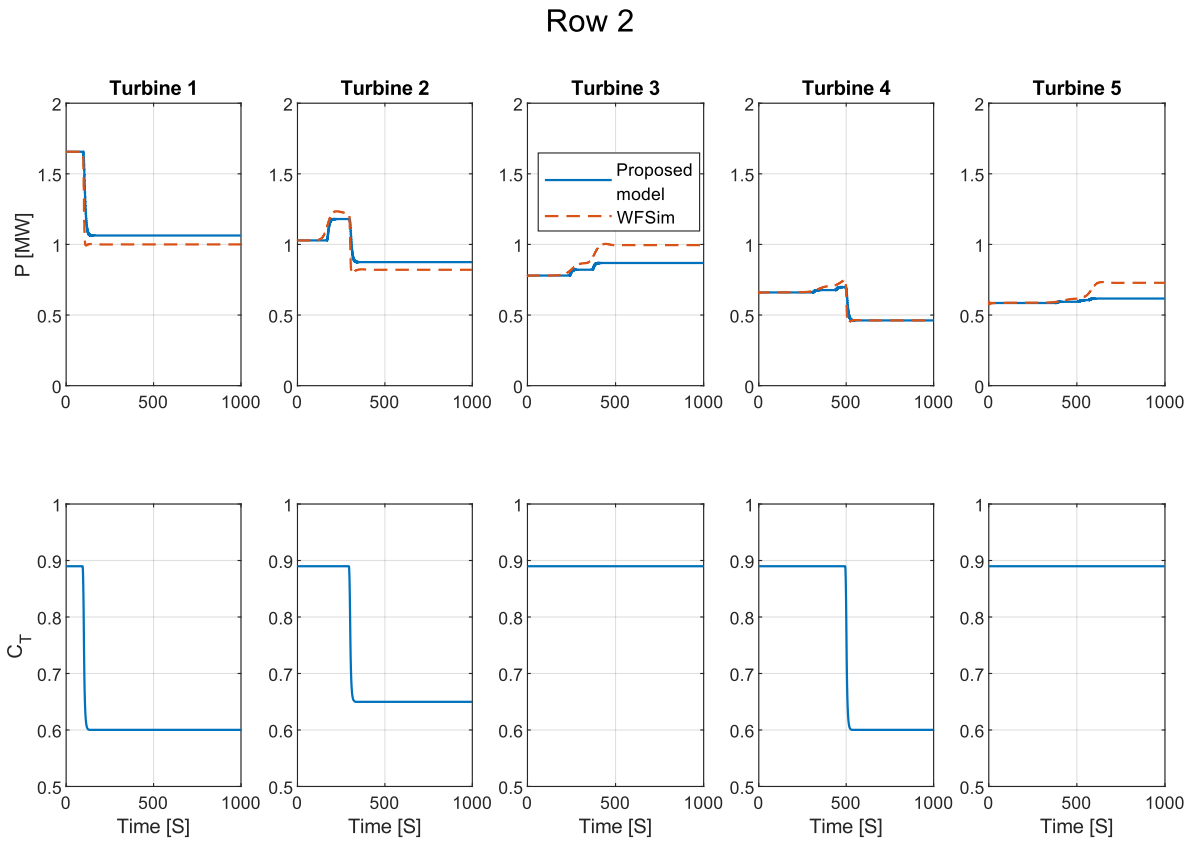

Figure 2-6: Comparison of power outputs of turbines in row 2 predicted by the proposed model (blue) compared to WFSim (red dashed) with validation data. The used input signals can be found in the bottom figures.

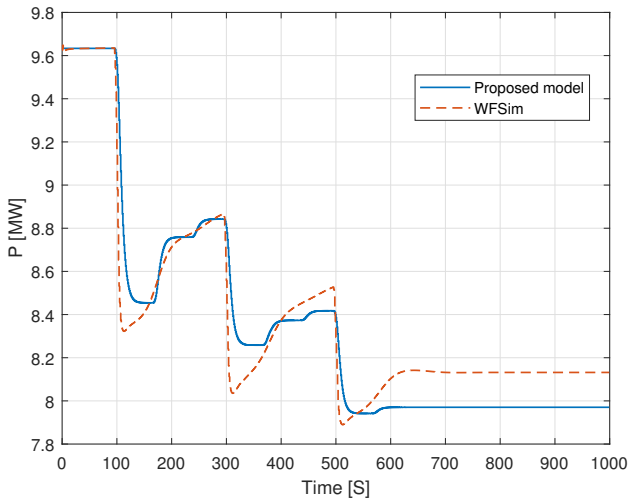

Figure 2-7: Comparison of total power output predicted by the proposed model (blue) compared to WFSim (red dashed) with validation data. The used input signals can be found in the bottom figures.

2-3 Summary

In this chapter, a linear, dynamic and fast controller model has been proposed. The model uses the power of the turbines as output and the thrust coefficients as inputs and it consists of a flow model and a turbine model. The flow model is based on the Frandsen's model that is linearized and made dynamic using the Taylor frozen turbulence hypothesis. The turbine model is formed by linearizing the ADM. State-space notations of the model, given in equations 2-24 and 2-25, are formed for local subsystems that consist of a single wind turbine. These notations are used in distributed model predictive control (DMPC) architectures in chapter 3. The local state-space notations can be stacked together to form a state-space system for the entire wind farm, given in equation 2-26. This notation is used within a centralized MPC in chapter 3. The model has been validated against the medium fidelity wind farm model WFSim.

Chapter 3

Control Architecture

In this chapter, model predictive control (MPC) algorithms for active power control (APC) will be introduced. In the context of this thesis, MPC is an algorithm that will search for the most optimal control sequence that steers the output of a system to a given reference signal by minimizing a cost function, whilst keeping constraints into account. To do so, it will use a control model (here the wind farm model introduced in chapter 2) and the current state of the system. At time instance k, measurements are taken from the real system (here represented by WindFarmSimulator (WFSim)), such that the control model can be initialized. After this, the cost function is minimized over the prediction horizon H, thus until time instance $k+H$. Then, only the first instance of the found control sequence is applied. This sequence repeats itself every sample time.

Both a centralized and distributed MPC algorithms will be designed. In the centralized control architecture, measurements from all wind turbines in the wind farms are sent to a centralized controller. In this controller, all optimal control actions are determined and send to the wind turbines. A problem with this way of controlling, is that calculating the optimal control actions can be very computationally heavy if the wind farm is large. This means that it is a major challenge to provide real-time control for large wind farms with centralized MPC. Distributed controllers can be a solution to this problem. In the distributed controller, the centralized control problem will be divided into smaller sub-problems. These sub-problems will be solved in parallel on separate controllers, that share information with each other. With this way of controlling, it will be possible to provide real-time control for large wind farms. Another advantage of distributed control is that it is more fail safe. With centralized control, if the centralized controller fails, the entire farm fails. With distributed control, if one of the controllers fails, the rest of the farm can still be controlled. In the controllers, the wind farm model developed in chapter 2 will be used as controller model. To provide offset free tracking despite model mismatches and (unknown) disturbances, integral action will be added to the controllers. This will be done by rewriting the wind farm model in the velocity form as described by L. Wang and M.A. Stephens et al. [30, 31].

In the coming sections both the centralized and the distributed controllers will be developed. The centralized controller will be developed in section 3-1 and in section 3-2 the distributed

controller will be developed. In the next chapter, chapter 4, the proposed controllers will be tested in WFSim, a medium fidelity wind farm model introduced in chapter 2.

3-1 Centralized Model Predictive Controller

As explained above, in the centralized control architecture, a centralized controller will determine the optimal control actions using a controller model and the measured power outputs, thrust coefficients and wind speeds. These optimal control actions are send to all the wind turbines. In figure 3-1 a schematic representation of the proposed control loop is given. $P_{ref}[k]$ is the reference for the total power output of the wind farm. $C_T[k]$ is a vector that contains the desired thrust coefficients, i.e. $C_T[k] = \begin{bmatrix} C_{T1} & C_{T2} & \cdots & C_{TG} \end{bmatrix}^T [k]$, where, G is the total amount of wind turbines in the farm. $y_m[k]$ are measurements taken from WFSim containing the power outputs, the actual thrust coefficients and the wind speeds just in front of the wind turbines. This is enough information to update and initialize the model given in chapter 2. As it is relatively easy to measure the power outputs of the turbines, it is valid to assume they are known. For the wind speed and the actual thrust coefficients, this is more difficult. Research suggests using LIght Detection And Ranging of Laser Imaging Detection And Ranging (LIDAR) to measure the flow field in front of wind turbines [32]. From this data it is possible to estimate the wind speeds and thrust coefficients using the Actuator Disk Model (ADM) [19]. Another possibility would be to use observers as proposed in literature [33]. This is however not considered in this thesis.

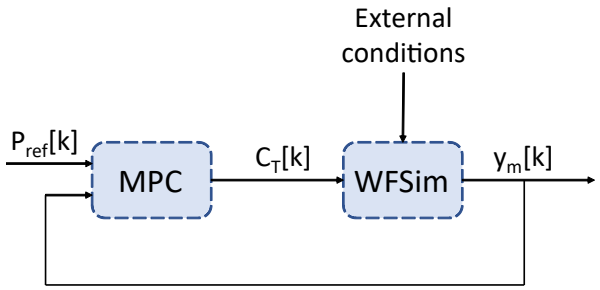

Figure 3-1: Schematic representation of centralized MPC loop.

In this centralized controller, the state-space notation for the complete system S as given in equation 2-26 will be used. Due to model mismatches and unknown disturbances, offset free tracking can not be ensured. A way to overcome this problem is by adding integral action to the controller by reformulating the controller model in the velocity form [30, 31]. This will be done in the first subsection 3-1-1. In the second subsection 3-1-2, the centralized control problem will be defined

3-1-1 Reformulating Centralized Model in Velocity Form

In this subsection, the model given in equation 2-26, will be rewritten in velocity form. This means that the model will be reformulated such that instead of the absolute input

$C_T[k]$, the incremental input $\Delta C_T[k] = C_T[k] - C_T[k-1]$ is used [30, 31, 34]. By defining $\Delta x[k] = x[k] - x[k-1]$, the state update defined in equation 2-26 can be transformed into

$$\begin{aligned}\Delta x[k+1] &= A(x[k] - x[k-1]) + B(C_T[k] - C_T[k-1]) + c_x - c_x \\ &= A\Delta x[k] + B\Delta C_T[k].\end{aligned} \tag{3-1}$$

Note that the constant vector c_x is eliminated from the equation. To relate the output $\hat{P}[k] = \begin{bmatrix}\hat{P}_1 & \hat{P}_2 & \cdots & \hat{P}_G\end{bmatrix}^T[k]$ to $\Delta x[k]$ the next definition is derived

$$\begin{aligned}\Delta\hat{P}[k+1] &= C\Delta x[k+1] + c_y - c_y \\ &= CA\Delta x[k] + CB\Delta C_T[k],\end{aligned} \tag{3-2}$$

where $\Delta\hat{P}[k] = \hat{P}[k] - \hat{P}[k-1]$. By defining the new state $x_I[k] = \begin{bmatrix}\hat{P}^T[k] & \Delta x^T[k]\end{bmatrix}^T$, a new state space system can be derived:

$$\begin{aligned}\begin{bmatrix}\hat{P}[k+1] \\ \Delta x[k+1]\end{bmatrix} &= \underbrace{\begin{bmatrix}I & CA \\ 0 & A\end{bmatrix}}_{A_I}\begin{bmatrix}\hat{P}[k] \\ \Delta x[k]\end{bmatrix} + \underbrace{\begin{bmatrix}CB \\ B\end{bmatrix}}_{B_I}\Delta C_T[k] \\ \hat{P}[k] &= \underbrace{\begin{bmatrix}I & 0\end{bmatrix}}_{C_I}\begin{bmatrix}\hat{P}[k] \\ \Delta x[k]\end{bmatrix},\end{aligned} \tag{3-3}$$

with the identity matrix $I \in \mathbb{R}^{G\times G}$. This state space system is the velocity form of the controller model introduced in chapter 2. This system will from now on be denoted by

$$\begin{aligned}x_I[k+1] &= A_I x_I[k] + B_I\Delta C_T[k] \\ \hat{P}[k] &= C_I x_I[k].\end{aligned} \tag{3-4}$$

3-1-2 Centralized Control Problem

With velocity form of the controller model defined, the control problem can be formulated. The total power output of the wind farm will be steered to the reference signal by solving the optimization objective

$$\begin{aligned}\min_{\Delta U[k]} \Bigg\{ & q^2 \sum_{i=1}^{H}\left(\sum_{j=1}^{G}\hat{P}_j[k+i] - P_{ref}[k+i]\right)^2 \\ & + \sum_{i=0}^{H-1}\Delta C_T^T[k+i]R^T R\Delta C_T[k+i]\Bigg\},\end{aligned} \tag{3-5}$$

where $\Delta U[k] = \begin{bmatrix}\Delta C_T[k] & \Delta C_T[k+1] & \cdots & \Delta C_T[k+H-1]\end{bmatrix}^T$ is the optimal control sequence and $\hat{P}_j[k+i]$ are the predicted power outputs of the wind turbines, contained in the output vector $\hat{P}[k+i]$, i.e. $\hat{P}[k+i] = \begin{bmatrix}\hat{P}_1[k+i] & \hat{P}_2[k+i] & \cdots & \hat{P}_N[k+i]\end{bmatrix}^T$. $P_{ref}[k+i]$ is the reference for the total power output of the wind farm, q is the weight that is put on the

tracking error and $R = r \cdot I \in \mathbb{R}^{G \times G}$ is the weighting matrix on $\Delta C_T[k+i]$. Choices have to be made regarding the weights q and r and the prediction horizon H.

It is important to note that the behavior of the controller is influenced by the ratio between q and r. Therefore the easiest way to tune both values, is to make q equal to 1 and try different values for r. Increasing the weight, r, on $\Delta C_T[k+i]$ will lead to a more conservative controller that will react less to disturbances and will provide smoother and less fluctuating input signals. Smooth control actions can be desirable, as it is possible that real life wind turbines will not be able to follow large and fast changes in C_T. Furthermore, abrupt changes in the input signal can create large stresses in the wind turbines resulting in more fatigue damage. Lastly, heavy varying input signals will also result in an increase of turbulence in the wakes, which will in turn increase the fatigue damage to downwind turbines. However, a too conservative controller will also result in poor tracking. The weights, q and r, must be tuned in order to achieve a stable controller with a balanced trade-off between smooth input signals and proper tracking. It is possible that different conditions require different settings.

For choosing the prediction horizon H, a trade-off has to be made as well. H should be large enough for the controller to be stable and, preferable, to include the dominant plant dynamics [30]. This last requirement can be achieved by taking the maximum amount of time samples it takes for the wake effects to travel from the most upwind turbine T_{u_m} to the most downwind turbine T_{dw_m} in any of the rows m. This is equal to $\max_m d_{u_m,dw_m}$. To this $\lceil \tau/h \rceil$ should be added. d_{u_m,dw_m} and the time constant τ were introduced in chapter 2. The time constant, τ, is used to filter the control actions C_T. d_{u_m,dw_m} is equal to the number of time samples it takes for the wakes to get from the most upwind turbine T_{u_m} to the most downwind turbine T_{dw_m} in row m. This variable is dependent on the distance between the wind turbines and the free stream wind speed $V_{\infty,0}$. This means that the chosen prediction horizon is also dependent on $V_{\infty,0}$. Increasing H will, unfortunately, require more computational effort, resulting in a slower controller from a computational point of view. To provide real-time control, the controller should be able to solve the optimization objective within one sample time of length h.

Besides the optimization objective, also constraints have to be defined for the MPC. Theoretically, the maximum amount of energy is subtracted from a flow of air by a single wind turbine when the thrust coefficient is at the Betz's limit of $C_{Ti} = 8/9$ [35]. If the wind turbine exceeds this Betz's limit, the rotor will rotate faster, but less energy is subtracted from the flow of air. In chapter 2 it was decided not to model this behavior above the Betz's limit. Therefore, if any C_{Ti} were to exceed the Betz's limit of $C_{Ti} = 8/9$, the developed model will not be valid. For this reason, it is decided to set the constraint that C_{Ti} should not exceed $C_{Tmax} = 8/9$ for all $\{i \in \mathbb{Z} | 1 \leq i \leq G\}$. Also, a common requirement is that the wind turbines should not shut down completely. Therefore, it is decided that $C_{Ti}[k]$ should not become smaller than $C_{Tmin} = 0.01$ for all $\{i \in \mathbb{Z} | 1 \leq i \leq G\}$.

In equation 3-3, the increments of the thrust coefficients, ΔC_{Ti}, and not of the absolute values of the thrust coefficients, C_{Ti}, are being used as input. Hence, it is not possible to put direct constraints on C_{Ti}. This can be solved by realising that $C_{Ti}[k+j] = C_{Ti}[k-1] + \sum_{l=0}^{j} \Delta C_{Ti}[k+l]$. Because at sample time k, $C_{Ti}[k-1]$ is known, it is possible to write the

constraints for the complete prediction horizon as

$$\left.\begin{array}{rcccl} C_{T\,min} & \leq & C_{Ti}[k-1] + \Delta C_{Ti}[k] & \leq & C_{T\,max} \\ C_{T\,min} & \leq & C_{Ti}[k-1] + \Delta C_{Ti}[k] + \Delta C_{Ti}[k+1] & \leq & C_{T\,max} \\ & & \vdots & & \\ C_{T\,min} & \leq & C_{Ti}[k-1] + \sum_{j=0}^{H-1} \Delta C_{Ti}[k+j] & \leq & C_{T\,max} \end{array}\right\} \forall\{i \in \mathbb{Z} | 1 \leq i \leq G\}, \tag{3-6}$$

which can be rewritten as

$$S_1 \left(C_{T\,min} - C_T^T[k-1]\right) \leq S_2 \Delta U[k] \leq S_1 \left(C_{T\,max} - C_T^T[k-1]\right), \tag{3-7}$$

where $\leq$ is defined as an element-wise inequality and S_1 and S_2 are given by

$$\begin{aligned} S_1 &= \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^T \in \mathbb{R}^{H \times 1} \\ S_2 &= \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \in \mathbb{R}^{H \times H}. \end{aligned} \tag{3-8}$$

This leads to the complete centralized optimization problem that is solved in for each time step:

$$\begin{aligned} \min_{\Delta U[k]} \Bigg\{ & q^2 \sum_{i=1}^{H} \left(\sum_{j=1}^{G} \hat{P}_j[k+i] - P_{ref}[k+i] \right)^2 \\ & + \sum_{i=0}^{H-1} \Delta C_T^T[k+i] R^T R \Delta C_T[k+i] \Bigg\} \\ \text{s.t.} \quad & x_I[k+1] = A_I x_I[k] + B_I \Delta C_T[k], \quad \hat{P}[k] = C_I x_I[k] \quad \text{and} \\ & S_1 \left(C_{T\,min} - C_T^T[k-1]\right) \leq S_2 \Delta U[k] \leq S_1 \left(C_{T\,max} - C_T^T[k-1]\right). \end{aligned} \tag{3-9}$$

This finalizes the design of the centralized controller. This controller is implemented in Gurobi 8.1.0 in MATLAB 2019a.

3-2 Distributed Model Predictive Controller

In the distributed controllers, the centralized control problem is divided into smaller local control problems. By solving these local control problems and by sharing information, the local controllers will, together, find the most optimal control actions for the entire wind farm. To achieve this goal, two different distributed controllers based on the Jacobi algorithm as defined by A.N. Venkat et al. [36] are developed. It is chosen to use this algorithm, because it is possible to solve a large part of the control algorithm in parallel, making it computationally fast.

In this section these distributed controllers are developed and explained in more detail. First, in subsection 3-2-1, the decomposed state-space notation given in equations 2-24 and 2-25 is transformed into the velocity form to add integral action. Then, in subsection 3-2-2, the control problem formed with this decomposed state-space notation, is transformed into a coupled cost decoupled constraint (CCDC) problem, as is necessary for the Jacobi algorithm. Lastly, in subsection 3-2-3, the two distributed control algorithms are defined.

3-2-1 Reformulating Distributed Model in Velocity Form

In this subsection, to ensure offset free tracking, integral action is added to the controller by transforming the decomposed controller model given in equations 2-24 and 2-25 into the velocity form following the same method as for the centralized controller. By defining $\Delta x_i[k] = x_i[k] - x_i[k-1]$ and $\Delta C_{Ti}[k] = C_{Ti}[k] - C_{Ti}[k-1]$ it is possible to state that

$$\begin{aligned}\Delta x_{u_m}[k+1] =& A_{u_m,u_m}(x_{u_m}[k] - x_{u_m}[k-1]) + B_{u_m}(C_{Tu_m}[k] - C_{Tu_m}[k-1]) \\ &+ c_{x,u_m} - c_{x,u_m} \\ =& A_{u_m,u_m}\Delta x_{u_m}[k] + B_{u_m}\Delta C_{Tu_m}[k],\end{aligned} \tag{3-10}$$

for the most upwind subsystem, S_{u_m}, in each row m, where $\{m \in \mathbb{Z} | 1 \leq m \leq M\}$. Here, u_m is the index of the most upwind subsystem in row m as defined in chapter2. For all the other subsystems, S_i, it is possible to state, using the same definitions for $\Delta x_i[k]$ and $C_{Ti}[k]$, that

$$\begin{aligned}\Delta x_i[k+1] =& A_{i,i}(x_i[k] - x_i[k-1]) + A_{i,i-1}(x_{i-1}[k] - x_{i-1}[k-1]) \\ &+ B_i(C_{Ti}[k] - C_{Ti}[k-1]) + c_{x,i} - c_{x,i} \\ =& A_{i,i}\Delta x_i[k] + A_{i,i-1}\Delta x_{i-1}[k] + B_i\Delta C_{Ti}[k],\end{aligned} \tag{3-11}$$

where $\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \ \forall m\}$. Using this result and by defining $\Delta \hat{P}_i[k] = \hat{P}_i[k] - \hat{P}_i[k-1]$ it follows that

$$\begin{aligned}\Delta \hat{P}_{u_m}[k+1] &= C_{u_m}\Delta x_{u_m}[k+1] + c_{y,u_m} - c_{y,u_m} \\ &= C_{u_m}A_{u_m,u_m}\Delta x_{u_m}[k] + C_{u_m}B_{u_m}\Delta C_{Tu_m}[k]\end{aligned} \tag{3-12}$$

for all $\{m \in \mathbb{Z} | 1 \leq m \leq M\}$ and

$$\begin{aligned}\Delta \hat{P}_i[k+1] =& C_i\Delta x_i[k+1] + c_{y,i} - c_{y,i} \\ =& C_iA_{i,i}\Delta x_i[k] + C_iA_{i,i-1}\Delta x_{i-1}[k] + C_iB_i\Delta C_{Ti}[k]\end{aligned} \tag{3-13}$$

for all $\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \ \forall m\}$. Notice that the terms $A_{i,i-1}\Delta x_{i-1}[k]$ in equation 3-11 and $C_iA_{i,i-1}\Delta x_{i-1}[k]$ in equation 3-13 define the interactions between the subsystems S_i and S_{i-1}.

By defining the new state $x_{Ii}[k] = \begin{bmatrix}\hat{P}_i^T[k] & \Delta x_i^T[k]\end{bmatrix}^T$, the previous equations can be rewritten to form the decomposed state space system in velocity form:

$$\begin{aligned}\begin{bmatrix}\hat{P}_{u_m}[k+1] \\ \Delta x_{u_m}[k+1]\end{bmatrix} &= \underbrace{\begin{bmatrix}I & C_{u_m}A_{u_m,u_m} \\ 0 & A_{u_m,u_m}\end{bmatrix}}_{A_{Iu_m,u_m}}\begin{bmatrix}\hat{P}_{u_m}[k] \\ \Delta x_{u_m}[k]\end{bmatrix} + \underbrace{\begin{bmatrix}C_{u_m}B_{u_m} \\ B_{u_m}\end{bmatrix}}_{B_{Iu_m}}\Delta C_{Tu_m}[k] \\ \hat{P}_{u_m}[k] &= \underbrace{\begin{bmatrix}I & 0\end{bmatrix}}_{C_{Iu_m}}\begin{bmatrix}\hat{P}_{u_m}[k] \\ \Delta x_{u_m}[k]\end{bmatrix}\end{aligned} \tag{3-14}$$

for all $\{m \in \mathbb{Z} | 1 \leq m \leq M\}$ with the identity matrix $I \in \mathbb{R}^{1\times 1}$ and

$$\begin{aligned}\begin{bmatrix}\hat{P}_i[k+1]\\ \Delta x_i[k+1]\end{bmatrix} &= \underbrace{\begin{bmatrix}I & C_iA_{i,i}\\ 0 & A_{i,i}\end{bmatrix}}_{A_{Ii,i}}\begin{bmatrix}\hat{P}_i[k]\\ \Delta x_i[k]\end{bmatrix} + \underbrace{\begin{bmatrix}0 & C_iA_{i,i-1}\\ 0 & A_{i,i-1}\end{bmatrix}}_{B_{Ii,i-1}}\begin{bmatrix}\hat{P}_{i-1}[k]\\ \Delta x_{i-1}[k]\end{bmatrix}\\ &\quad + \underbrace{\begin{bmatrix}C_iB_i\\ B_i\end{bmatrix}}_{B_{Ii}}\Delta C_{Ti}[k]\\ \hat{P}_i[k] &= \underbrace{\begin{bmatrix}I & 0\end{bmatrix}}_{C_{Ii}}\begin{bmatrix}\hat{P}_i[k]\\ \Delta x_i[k]\end{bmatrix}\end{aligned} \tag{3-15}$$

for all $\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \forall m\}$ with the identity matrix $I \in \mathbb{R}^{1\times 1}$. This can be written more compact as

$$\begin{aligned}x_{Iu_m}[k+1] &= A_{Iu_m,u_m}x_{Iu_m}[k] + B_{Iu_m}\Delta C_{Tu_m}[k]\\ \hat{P}_{u_m}[k] &= C_{Iu_m}x_{Iu_m}[k]\end{aligned} \tag{3-16}$$

for all $\{m \in \mathbb{Z} | 1 \leq m \leq M\}$ and

$$\begin{aligned}x_{Ii}[k+1] &= A_{Ii,i}x_{Ii}[k] + A_{Ii,i-1}x_{Ii-1}[k] + B_{Ii}\Delta C_{Ti}[k]\\ \hat{P}_i[k] &= C_{Ii}x_{Ii}[k]\end{aligned} \tag{3-17}$$

for all $\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \forall m\}$. These two sets of equations together form the decomposed state-space system in velocity form.

Using this new state-space system, the centralized control problem given in equation 3-9 can be rewritten in the distributed form as

$$\begin{aligned}&\min_{\Delta U_1,\cdots,\Delta U_G} \left\{ q^2 \sum_{l=1}^{H}\left(\sum_{i=1}^{G}\hat{P}_i[k+l] - P_{ref}[k+l]\right)^2 \right.\\ &\qquad \left. + r^2\sum_{i=1}^{G}\sum_{l=0}^{H-1}\Delta C_{Ti}[k+l]^2 \right\} \quad \forall\{i \in \mathbb{Z} | 1 \leq i \leq G\}\\ &\text{s.t.} \quad x_{Iu_m}[k+1] = A_{Iu_m,u_m}x_{Iu_m}[k] + B_{Iu_m}\Delta C_{Tu_m}[k] \text{ and}\\ &\qquad \hat{P}_{u_m}[k] = C_{Iu_m}x_{Iu_m}[k] \quad \forall m,\\ &\qquad x_{Ii}[k+1] = A_{Ii,i}x_{Ii}[k] + A_{Ii,i-1}x_{Ii-1}[k] + B_{Ii}\Delta C_{Ti}[k] \text{ and}\\ &\qquad \hat{P}_i[k] = C_{Ii}x_{Ii}[k] \quad \forall\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \forall m\} \text{ and}\\ &\qquad S_1\left(C_{Tmin} - C_{Ti}[k-1]\right) \leq S_2\Delta U_i \leq S_1\left(C_{Tmax} - C_{Ti}[k-1]\right)\\ &\qquad \forall\{i \in \mathbb{Z} | 1 \leq i \leq G\},\end{aligned} \tag{3-18}$$

where $\Delta U_i = \begin{bmatrix}C_{Ti}[k] & C_{Ti}[k+1] & \cdots & C_{Ti}[k+H-1]\end{bmatrix}^T$ and $\{m \in \mathbb{Z} | 1 \leq m \leq M\}$

3-2-2 Coupled Cost Decoupled Constraints Problem

To solve the control problem via the Jacobi algorithm, it is necessary that the control problem is only coupled via the cost function and not via the constraints [36]. That means that it should

be a so called coupled cost decoupled constraint (CCDC) problem. Decoupled, here, means that it is possible to split the cost function or the constraints into cost functions or constrains that are independent from each other. For example, the decoupled cost function $f(y_1, y_2, y_3) = y_1 + y_2 + y_3$ can be split into the cost functions $f(y_1) = y_1$, $f(y_2) = y_2$ and $f(y_3) = y_3$, such that $f(y_1, y_2, y_3) = f(y_1) + f(y_2) + f(y_3)$. For coupled cost functions or constraints this is not possible. Note that in the control problem given in equation 3-18, the subsystems are coupled via the cost function because of the part $q^2 \sum_{t=1}^{H} \left(\sum_{i=1}^{G} \hat{P}_i[k+t] - P_{ref}[k+t] \right)^2$ and via the constraints because of the part $x_{Ii}[k+1] = A_{Ii,i} x_{Ii}[k] + A_{Ii,i-1} x_{Ii-1}[k] + B_{Ii} \Delta u_i[k]$. This means that, to be able to solve the control problem via the Jacobi algorithm, it is necessary to rewrite the control problem. In the coming part, the control problem will be transformed into a CCDC problem by incorporating the coupled constraints, that represent the control model, into the cost function.

First, for simplicity, a wind farm consisting of 1 row with a minimum of 3 three wind turbines is considered. It is possible to calculate the predicted future state variables for the most upwind subsystem with

$$\begin{aligned}
x_1[k+1] =& A_{I1,1} x_1[k] + B_{I1} \Delta u_1[k] \\
x_1[k+2] =& A_{I1,1} x_1[k+1] + B_{I1} \Delta u_1[k+1] \\
=& A_{I1,1}^2 x_1[k] + A_{I1,1} B_{I1} \Delta u_1[k] + B_{I1} \Delta u_1[k+1] \\
& \vdots \\
x_1[k+H] =& A_{I1,1}^H x_1[k] + A_{I1,1}^{H-1} B_{I1} \Delta u_1[k] + A_{I1,1}^{H-2} B_{I1} \Delta u_1[k+1] \\
& + \cdots + B_{I1} \Delta u_1[k+H-1].
\end{aligned}$$

This set of equations can be written in matrix form as

$$X_1 = F_{x1} x_1[k] + \Phi_{x1} \Delta U_1,$$

where

$$X_i = \begin{bmatrix} x_i[k+1] \\ x_i[k+2] \\ \vdots \\ x_i[k+H] \end{bmatrix}, \; F_{xi} = \begin{bmatrix} A_{Ii,i} \\ A_{Ii,i}^2 \\ A_{Ii,i}^3 \\ \vdots \\ A_{Ii,i}^H \end{bmatrix}, \; \Phi_{xi} = \begin{bmatrix} B_{Ii} & 0 & 0 & \cdots & 0 \\ A_{Ii,i} B_{Ii} & B_{Ii} & 0 & \cdots & 0 \\ A_{Ii,i}^2 B_{Ii} & A_{Ii,i} B_{Ii} & B_{Ii} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{Ii,i}^{H-1} B_{Ii} & A_{Ii,i}^{H-2} B_{Ii} & A_{Ii,i}^{H-3} B_{Ii} & \cdots & B_{Ii} \end{bmatrix}.$$

For the second subsystem in the row it is possible to state that

$$\begin{aligned}
x_2[k+1] =& A_{I2,2}x_2[k] + B_{I2}\Delta u_2[k] + \underbrace{A_{I2,1}x_1[k]}_{\text{effects } T_1}\\
x_2[k+2] =& A_{I2,2}x_2[k+1] + B_{I2}\Delta u_2[k+1] + A_{I2,1}x_1[k+1]\\
=& A_{I2,2}^2x_2[k] + A_{I2,2}B_{I2}\Delta u_2[k] + B_{I2}\Delta u_2[k+1] + \underbrace{A_{I2,2}A_{I2,1}x_1[k] + A_{I2,1}x_1[k+1]}_{\text{effects } T_1}\\
&\vdots\\
x_2[k+H] =& A_{I2,2}^H x_2[k]\\
&+ A_{I2,2}^{H-1}B_{I2}\Delta u_2[k] + A_{I2,2}^{H-2}B_{I2}\Delta u_2[k+1] + \cdots + B_{I2}\Delta u_2[k+H-1]\\
&+ \underbrace{A_{I2,2}^{H-1}A_{I2,1}x_1[k] + A_{I2,2}^{H-2}A_{I2,1}x_1[k+1] + \cdots + A_{I2,1}x_1[k+H-1]}_{\text{effects } T_1}.
\end{aligned}$$

This set of equations can be written in matrix form as

$$\begin{aligned}
X_2 =& F_{x2}x_2[k] + \Phi_{x2}\Delta U_2 + \underbrace{\Xi_{2,1}\left(L_1X_1 + W_1x_1[k]\right)}_{\text{effects } T_1}\\
=& F_{x2}x_2[k] + \Phi_{x2}\Delta U_2 + \underbrace{\Xi_{2,1}\Big(\left(L_1F_{x1} + W_1\right)x_1[k] + L_1\Phi_{x1}\Delta U_1\Big)}_{\text{effects } T_1},
\end{aligned}$$

where

$$\Xi_{i,i-1} = \begin{bmatrix}
A_{Ii,i-1} & 0 & 0 & \cdots & 0\\
A_{Ii,i}A_{Ii,i-1} & A_{Ii,i-1} & 0 & \cdots & 0\\
A_{I_{i,i}}^2A_{Ii,i-1} & A_{Ii,i}A_{Ii,i-1} & A_{Ii,i-1} & \cdots & 0\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
A_{I_{i,i}}^{H-1}A_{Ii,i-1} & A_{I_{i,i}}^{H-2}A_{Ii,i-1} & A_{I_{i,i}}^{H-3}A_{Ii,i-1} & \cdots & A_{Ii,i-1}
\end{bmatrix},$$

$$L_i = \begin{bmatrix} 0 & 0\\ I & 0 \end{bmatrix} \in \mathbb{R}^{(nx_iH)\times(nx_iH)},$$

with identity matrix $I \in \mathbb{R}^{(nx_i(H-i))\times(nx_i(H-i))}$ and nx_i is the number of states in x_i and

$$W_i = \begin{bmatrix} I\\ 0 \end{bmatrix} \in \mathbb{R}^{(nx_iH)\times nx_i},$$

with identity matrix $I \in \mathbb{R}^{nx_i\times nx_i}$. Then, for the third subsystem in the row it holds that

$$\begin{aligned}
X_3 =& F_{x3}x_3[k] + \Phi_{x3}\Delta U_3 + \underbrace{\Xi_{3,2}\Bigg(\left(L_2F_{x2} + W_2\right)x_2[k] + L_2\Phi_{x2}\Delta U_2}_{\text{effects } T_2}\\
&+ \underbrace{L_2\Xi_{2,1}\Big(\left(L_1F_{x1} + W_1\right)x_1[k] + L_1\Phi_{x1}\Delta U_1\Big)}_{\text{effects } T_1}\Bigg).
\end{aligned}$$

This can be extended for wind farms with an arbitrary number of rows and columns for any arbitrary subsystem $S_i, \{i \in \mathbb{Z} | 1 \leq i \leq G\}$ as

$$X_{u_m} = F_{xu_m}x_{u_m}[k] + \Phi_{xu_m}\Delta U_{u_m} \ \forall m \tag{3-19}$$

and

$$
\begin{aligned}
X_i =& F_{xi}x_i[k] + \Phi_{xi}\Delta U_i + \underbrace{\Xi_{i,i-1}\Big(\left(L_{i-1}F_{xi-1} + W_{i-1}\right) x_{i-1}[k] + L_{i-1}\Phi_{xi-1}\Delta U_{i-1}}_{\text{effects } T_{i-1}} \\
&+ \underbrace{L_{i-1}\Xi_{i-1,i-2}\Big(\left(L_{i-2}F_{xi-2} + W_{i-2}\right) x_{i-2}[k] + L_{i-2}\Phi_{xi-2}\Delta U_{i-2}}_{\text{effects } T_{i-2}} + \dots \\
&+ \underbrace{L_{u_m+1}\Xi_{u_m-1,u_m}\Big(\left(L_{u_m}F_{xu_m} + W_{u_m}\right) x_{u_m}[k] + L_{u_m}\Phi_{xu_m}\Delta U_{u_m}\Big)}_{\text{effects } T_{u_m}} \dots \Big)\Big) \\
&\forall\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \ \forall m\}.
\end{aligned}
\tag{3-20}
$$

Notice that in this last equation all the subsystems that are accessible to S_i, i.e., all subsystems within Θ_{+i} as defined in subsection 2-1-4, are taken into account. This is, however, not necessary. It takes time for the wake effects to travel from a wind turbine T_j to a downwind turbine T_i. This amount of time was defined as $d_{j,i}$ in chapter 2. If the wind farm is large and/or if the chosen prediction horizon H is low, there will be wind turbines upwind from T_i of which the wake effects will not reach T_i within the prediction horizon H, i.e. turbines T_j for which $d_{j,i} > H$. A visual representation of such a case is given in figure 3-2. For subsystem S_i it is not necessary to take the dynamics of these wind turbines into account. Following the same line of thought, it is also not necessary to take the effects into account that subsystem S_i has on downwind subsystems that are not reached by the wake of T_i within the time horizon H, i.e. turbines T_j for which $d_{i,j} > H$. To include these insights into the controller, new sets are introduced. From now on, the set Θ^H_{+i} will contain all the subsystems within Θ_{+i} that are accessible to S_i within the prediction horizon H, i.e. the set Θ^H_{+i} contains all subsystems $\{S_j \in \Theta_{+i} | d_{j,i} \leq H\}$. The set Θ^H_{-i} will contain all the subsystems within Θ_{-i} to which S_i is accessible within the prediction horizon H, i.e. the set Θ^H_{-i} contains all subsystems $\{S_j \in \Theta_{-i} | d_{i,j} \leq H\}$. Again, see figure 3-2 for a visual representation of an example of these sets. By incorporating only the interactions between the subsystem S_i and the subsystems within Θ^H_{+i} and Θ^H_{-i} in the local controller of subsystem S_i, the computational load on this controller becomes lower, without losing performance. Even more, no matter how big the wind farm, a local subsystem S_i will never have to take more interactions into account than it has with the subsystems within Θ^H_{+i} and Θ^H_{-i}. This means that, the local controllers have a maximum computational complexity that is independent of the size of the wind farm. Thus, if the local control problems are solved in parallel on separate processors and if the prediction horizon H is kept equal, the total control problem will be solved in the same amount of time, no matter how large the wind farm. That is, if the communicational delays between the

subsystems are disregarded. For these reasons, equation 3-20 is changed into

$$\begin{aligned} X_i =& F_{xi}x_i[k] + \Phi_{xi}\Delta U_i + \Xi_{i,i-1}\Big(\left(L_{i-1}F_{xi-1} + W_{i-1}\right) x_{i-1}[k] + L_{i-1}\Phi_{xi-1}\Delta U_{i-1} \\ &+ L_{i-1}\Xi_{i-1,i-2}\Big(\left(L_{i-2}F_{xi-2} + W_{i-2}\right) x_{i-2}[k] + L_{i-2}\Phi_{xi-2}\Delta U_{i-2} + \dots \\ &+ L_{j+1}\Xi_{j+1,j}\Big(\left(L_jF_{xj} + W_j\right) x_j[k] + L_j\Phi_{xj}\Delta U_j\Big) \dots \Big)\Big) \\ &\forall\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \ \forall m\}, \end{aligned} \tag{3-21}$$

where j is the index of the most upwind turbine within Θ^H_{+i}.

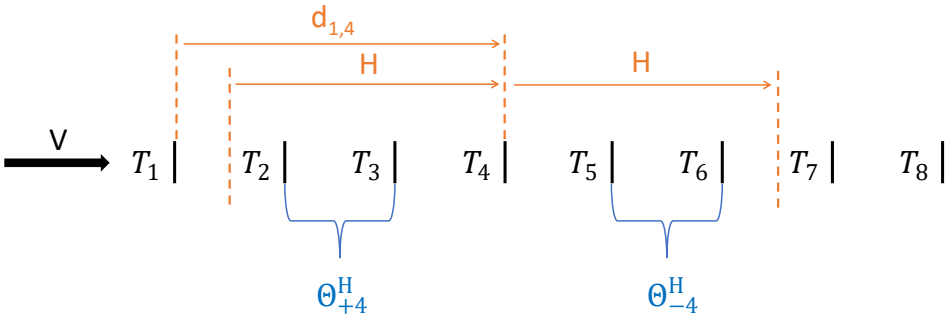

Figure 3-2: Visual representation of the sets $\Theta^H_{+4} = \{S_i | d_{i,4} \leq H\}$ and $\Theta^H_{-4} = \{S_i | d_{4,i} \leq H\}$ in an 1x8 wind farm in which $d_{1,4} > H$ and $H < d_{4,7} < d_{4,8}$.

Using equations 3-21 and 3-19 it is possible to state for the output of any subsystems S_i that

$$Y_{u_m} = F_{u_m}x_{u_m}[k] + \Phi_{u_m}\Delta U_{u_m} \ \forall m \tag{3-22}$$

and

$$\begin{aligned} Y_i =& F_i x_i[k] + \Phi_i\Delta U_i + \mathbf{C_{I}}_i\Xi_{i,i-1}\Big(\left(L_{i-1}F_{xi-1} + W_{i-1}\right) x_{i-1}[k] + L_{i-1}\Phi_{xi-1}\Delta U_{i-1} \\ &+ L_{i-1}\Xi_{i-1,i-2}\Big(\left(L_{i-2}F_{xi-2} + W_{i-2}\right) x_{i-2}[k] + L_{i-2}\Phi_{xi-2}\Delta U_{i-2} + \dots \\ &+ L_{j+1}\Xi_{j+1,j}\Big(\left(L_jF_{xj} + W_j\right) x_j[k] + L_j\Phi_{xj}\Delta U_j\Big) \dots \Big)\Big) \\ &\forall\{i \in \mathbb{Z} | 1 \leq i \leq G, i \neq u_m \ \forall m\}, \end{aligned} \tag{3-23}$$

where $Y_i = \begin{bmatrix}\hat{P}_i[k+1] & \hat{P}_i[k+2] & \cdots & \hat{P}_i[k+H]\end{bmatrix}^T$, $F_i = \mathbf{C_{I}}_i F_{xi}$, $\Phi_i = \mathbf{C_{I}}_i\Phi_{xi}$, $F^{\Theta^H_{+i}} = \mathbf{C_{I}}_i {F_x}^{\Theta^H_{+i}}$, $\Phi^{\Theta^H_{+i}} = \mathbf{C_{I}}_i {\Phi_x}^{\Theta^H_{+i}}$ and

$$\mathbf{C_{I}}_i = \begin{bmatrix} C_{Ii} & 0 & \cdots & 0 \\ 0 & C_{Ii} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_{Ii} \end{bmatrix}.$$

Using this result, the objective can be rewritten as

$$\begin{aligned}\min_{\Delta U_1,\cdots,\Delta U_G} &\left\{\left(\sum_{i=1}^{G} Y_i - Y_{ref}\right)^T Q^T Q \left(\sum_{i=1}^{G} Y_i - Y_{ref}\right) + \sum_{i=1}^{G} \Delta U_i^T R^T R \Delta U_i\right\} \\ \text{s.t.} \quad & S_1\left(C_{Tmin} - C_{Ti}[k-1]\right) \leq S_2 \Delta U_i \leq S_1\left(C_{Tmax} - C_{Ti}[k-1]\right) \\ & \forall\{i \in \mathbb{Z} | 1 \leq i \leq G\},\end{aligned} \tag{3-24}$$

where $Y_{ref} = \begin{bmatrix} P_{ref}[k+1] & P_{ref}[k+2] & \cdots & P_{ref}[k+H] \end{bmatrix}^T$, $Q = q \cdot I \in \mathbb{R}^{H \times H}$ and $R = r \cdot I \in \mathbb{R}^{H \times H}$ with $q > 0$ and $r > 0$. Note that this control problem is now no longer coupled via the constraints, but only via the cost. This means that it is now a CCDC problem that can be solved by the Jacobi algorithm as explained before. For simplicity and future use, a shorter notation for the same objective is given:

$$\min_{\Delta U_1 \in \bar{\Delta U}_1, \cdots, \Delta U_G \in \bar{\Delta U}_G} f(\Delta U_1, \cdots, \Delta U_G), \tag{3-25}$$

where the set $\bar{\Delta U}_i$ represents the constraint $S_1\left(C_{Tmin} - C_{T_i}^T[k-1]\right) \leq S_2 \Delta U_i[k] \leq S_1\left(C_{Tmax} - C_{T_i}^T[k-1]\right)$ and

$$f(\Delta U_1, \cdots, \Delta U_G) = q^2 \left(\sum_{i=1}^{G} Y_i - Y_{ref}\right)^2 + \sum_{i=1}^{G} \Delta U_i^T R^T R \Delta U_i.$$

3-2-3 Distributed Control Algorithms

Because the control problem is now transformed into a CCDC problem, it can be solved using the Jacobi algorithm. As has been stated before, two different distributed controllers will be developed using this algorithm. One controller will use the Jacobi algorithm as described by A.N. Venkat et al. [36]. This controller will be called the normal Jacobian distributed controller (NJDC). The second controller is based on a modification of this algorithm. This controller will be called the modified Jacobian distributed controller (MJDC).

In the NJDC, at each iteration, local control problems are solved in which the most optimal control actions are sought for a single subsystem S_i, whilst the control actions of all other subsystem are kept equal to the ones calculated at the previous iteration. The solutions of these control problems are then combined with the solution of the previous iteration using a weight. This update procedure is described in the next equation:

$$\begin{aligned}\Delta U_i^{*p+1} &= \underset{\Delta U_i \in \bar{\Delta U}_i}{\arg\min}\ f(\Delta U_1^p, \cdots, \Delta U_{i-1}^p, \Delta U_i, \Delta U_{i+1}^p, \cdots, \Delta U_G^p), \\ \Delta U_i^{p+1} &= w \Delta U_i^{*(p+1)} + (1-w)\Delta U_i^{(p)}, \\ & \forall\{i \in \mathbb{Z} | 1 \leq i \leq G\},\end{aligned} \tag{3-26}$$

where ΔU_i^{*p+1} is the solution to the local control problem of subsystem S_i at iteration $p+1$ and ΔU_i^p are the control actions for subsystem S_i calculated at iteration p. As can be seen, the control actions ΔU_i^{p+1} are updated by combining ΔU_i^{*p+1} with ΔU_i^p using weight $w = 1/G$.

The MJDC works similar as the NJDC, but with a modification. In the MJDC, at each iteration, local control problems are solved in which the most optimal control actions are sought not only for a single subsystem, but for multiple subsystems at the same time. The control actions of all other subsystems are kept equal to the ones calculated at the previous iteration. The solutions of all these control problems are combined using a weight. The idea behind this is, that, in this way, the effect that a subsystem has on downwind subsystems will be taken better into account by the controller. Notice that this, on it's own, is not a modification to the Jacobi algorithm yet. The subsystems used in the Jacobi algorithm can be chosen freely. Thus, if the subsystems were to consist of multiple wind turbines, the NJDC would be similar to the MJDC algorithm described here. However, in the MJDC, the sets of subsystems for which the control actions are solved in the same local control problem, will be overlapping. Best results are achieved if in the local control problems the control actions of subsystem S_i and all subsystems within Θ^H_{-i} are updated simultaneously. These control problems are formulated for each subsystem S_i, except for the subsystems more downwind in the farm. More precise, local control problems are formulated for each subsystem S_i, except for the subsystems S_i for which the time it takes for the wake to travel to the most downwind turbine is lower than the prediction horizon H. Thus, except for subsystems S_i for which $d_{i,dw_m} < H$. dw_m is, as defined in chapter 2, the index of the most downwind turbine within row m, where, here, m is the row in which turbine S_i is located. The idea behind this, is that these subsystems will already be included in the control problems of the subsystems upwind from these subsystems. The local control problems in the MJDC are defined as follows

$$\begin{bmatrix} \Delta U_i^{p+1} \\ \vdots \\ \Delta U_j^{p+1} \end{bmatrix} = \underset{\Delta U_i \in \underline{\Delta}\bar{U}_i, \ldots, \Delta U_j \in \underline{\Delta}\bar{U}_j}{\arg\min} f(\Delta U_1^p, \ldots, \Delta U_{i-1}^p, \Delta U_i, \ldots, \Delta U_j, \Delta U_{j+1}^p, \ldots, \Delta U_G^p) \quad \forall\{i \in \mathbb{Z} | (dw_m | S_{dw_m} \in \Theta_{-i})[d_{i,dw_m} \geq H]\}, \tag{3-27}$$

where j is the index of the most downwind subsystem within Θ^H_{-i}. Note that the sets of subsystems used within this notation are overlapping. To clarify, an example is given. For a wind farm with one row of G turbines, where $\Theta^H_{-i} = 1$ for all i except for the most downwind turbines, the next control problems would be formulated:

$$\begin{aligned}
\begin{bmatrix} \Delta U_1^{*p+1} \\ \Delta U_2^{*p+1} \end{bmatrix} &= \underset{\Delta U_1 \in \underline{\Delta}\bar{U}_1, \Delta U_2 \in \underline{\Delta}\bar{U}_2}{\arg\min} f(\Delta U_1, \Delta U_2, \Delta U_3^p, \cdots, \Delta U_G^p) \\
\begin{bmatrix} \Delta U_2^{*p+1} \\ \Delta U_3^{*p+1} \end{bmatrix} &= \underset{\Delta U_2 \in \underline{\Delta}\bar{U}_2, \Delta U_3 \in \underline{\Delta}\bar{U}_3}{\arg\min} f(\Delta U_1^p, \Delta U_2, \Delta U_3, \Delta U_4^p, \cdots, \Delta U_G^p) \\
\begin{bmatrix} \Delta U_3^{*p+1} \\ \Delta U_4^{*p+1} \end{bmatrix} &= \underset{\Delta U_3 \in \underline{\Delta}\bar{U}_3, \Delta U_4 \in \underline{\Delta}\bar{U}_4}{\arg\min} f(\Delta U_1^p, \Delta U_2^p, \Delta U_3, \Delta U_4, \Delta U_5^p, \cdots, \Delta U_G^p) \\
&\vdots \\
\begin{bmatrix} \Delta U_{G-1}^{*p+1} \\ \Delta U_G^{*p+1} \end{bmatrix} &= \underset{\Delta U_{G-1} \in \underline{\Delta}\bar{U}_{G-1}, \Delta U_G \in \underline{\Delta}\bar{U}_G}{\arg\min} f(\Delta U_1^p, \cdots, \Delta U_{G-2}^p, \Delta U_{G-1}, \Delta U_G)
\end{aligned}$$

As can be seen, the sets of subsystems for which the local control problems are solved, are overlapping. To combine the results from these overlapping control problems, it is necessary

to make a modification to the update rule described in equation 3-26. The weight, w, will now be defined as $w = 1/N_s$, where N_s is the number of subproblems defined by equation 3-27, which is equal to the number of subsystems in $\{S_i|(dw_m|S_{dw_m} \in \Theta_{-i})[d_{i,dw_m} \geq H]\}$. The results of the local control problems can then be combined in a linear way using the next rule:

$$\begin{bmatrix} \Delta U_1^{p+1} \\ \vdots \\ \Delta U_G^{p+1} \end{bmatrix} = \sum_{\{i \in \mathbb{Z}|(dw_m|S_{dw_m} \in \Theta_{-i})[d_{i,dw_m} \geq H]\}} w \begin{bmatrix} \Delta U_1^{p} \\ \vdots \\ \Delta U_{i-1}^{p} \\ \Delta U_i^{*p+1} \\ \vdots \\ \Delta U_j^{*p+1} \\ \Delta U_{j+1}^{p} \\ \vdots \\ \Delta U_G^{p} \end{bmatrix}, \tag{3-28}$$

where j again is defined as the index of the most downwind turbine within Θ_{-i}^H. To clarify the update rule given above, an example is given. For a wind farm with one row of G turbines, where $\Theta_{-i}^H = 1$ for all i except for the most downwind turbines, this rule translates to

$$\begin{bmatrix} \Delta U_1^{p+1} \\ \Delta U_2^{p+1} \\ \Delta U_3^{p+1} \\ \vdots \\ \Delta U_G^{p+1} \end{bmatrix} = w \begin{bmatrix} \Delta U_1^{*p+1} \\ \Delta U_2^{*p+1} \\ \Delta U_3^{p} \\ \vdots \\ \Delta U_G^{p} \end{bmatrix} + w \begin{bmatrix} \Delta U_1^{p} \\ \Delta U_2^{*p+1} \\ \Delta U_3^{*p+2} \\ \Delta U_4^{p} \\ \vdots \\ \Delta U_G^{p} \end{bmatrix} + \cdots + w \begin{bmatrix} \Delta U_1^{p} \\ \Delta U_2^{p} \\ \Delta U_3^{p} \\ \vdots \\ \Delta U_{G-2}^{p} \\ \Delta U_{G-1}^{*p+1} \\ \Delta U_G^{*p+1} \end{bmatrix}. \tag{3-29}$$

The complete update procedure for the MJDC is given by equations 3-27 and 3-28. These equations are combined in the next equation set:

$$\begin{aligned} \begin{bmatrix} \Delta U_j^{p+1} \\ \vdots \\ \Delta U_i^{p+1} \end{bmatrix} &= \underset{\Delta U_j \in \underline{\Delta} U_j, \ldots, \Delta U_i \in \underline{\Delta} U_i}{\arg\min} f(\Delta U_1^p, \ldots, \Delta U_{j-1}^p, \Delta U_j, \ldots, \Delta U_i, \Delta U_{i+1}^p, \ldots, \Delta U_G^p) \\ &\quad \forall \{i \in \mathbb{Z}|(dw_m|S_{dw_m} \in \Theta_{-i})[d_{i,dw_m} \geq H]\}, \\ \begin{bmatrix} \Delta U_1^{p+1} \\ \vdots \\ \Delta U_G^{p+1} \end{bmatrix} &= \sum_{\{i \in \mathbb{Z}|(dw_m|S_{dw_m} \in \Theta_{-i})[d_{i,dw_m} \geq H]\}} w \begin{bmatrix} \Delta U_1^{p} \\ \vdots \\ \Delta U_{j-1}^{p} \\ \Delta U_j^{*p+1} \\ \vdots \\ \Delta U_i^{*p+1} \\ \Delta U_{i+1}^{p} \\ \vdots \\ \Delta U_G^{p} \end{bmatrix} \end{aligned} \tag{3-30}$$

The update procedures for the NJDC and the MJDC are repeated until the difference between the control actions calculated at $p+1$ and p is smaller than a certain threshold $\{\epsilon \in \mathbb{R} | \epsilon > 0\}$ for all subsystems S_i. The precision of the algorithm is determined by this threshold. The full NJDC algorithm is given in algorithm 3.1 and the full MJDC algorithm is given in algorithm 3.2. It is possible to do part of these algorithm in parallel for each subproblem. By doing so, assuming that a processor core is available for each subproblem, it is envisioned that the algorithm will be equally fast regardless the size of the wind farm. In both the algorithms, the local optimization problems will be solved using Gurobi 8.1.0 in MATLAB 2019a.

Algorithm 3.1 Normal Jacobian Algorithm

Given $i \in \mathbb{Z} | 1 \leq i \leq G,\ u_i[k-1]\ \forall i,\ Y_{ref},\ H,\ q > 0,\ R = r \cdot I \in \mathbb{R}^{H \times H},\ r > 0,\ S_1,\ S_2,\ C_{Tmin},\ C_{Tmax},\ \Delta U_i^0 = 0\ \forall i,\ w_i > 0\ \forall i$, while $\sum w_i = 1,\ p_{max} > 0$ and $\epsilon > 0$
$p \leftarrow 0,\ e_i \leftarrow \Gamma\epsilon,\ \Gamma > 1$
while $e_i > \epsilon\ \exists i$ *and* $p \leq p_{max}$ **do**
  for all $\{i \in \mathbb{Z} | 1 \leq i \leq G\}$ **do** ▷ Do this in parallel
    $\Delta U_i^{*p+1} = \arg\min_{\Delta U_i \in \bar{\Delta U}_i} f(\Delta U_1^p, \cdots, \Delta U_{i-1}^p, \Delta U_i, \Delta U_{i+1}^p, \cdots, \Delta U_G^p)$
    $\Delta U_i^{p+1} = w_i \Delta U_i^{*(p+1)} + (1 - w_i)\Delta U_i^{(p)}$
    $e_i = \left\| \Delta U_i^{p+1} - \Delta U_i^p \right\|$
    Calculate Y_i using equation 3-22 or 3-23.
    Share Y_i with all other subsystems.
    Share ΔU_i^{p+1} with all subsystems within Θ_i^H
  end for
  $p \leftarrow p + 1$
end while
ΔU_i^p is the optimal control sequence $\forall S_i\ \{i \in \mathbb{Z} | 1 \leq i \leq G\}$

The local control problems will be solved with these algorithms every time step k, after which the first control actions of the optimal control sequences $\Delta U_i^{p-1} \forall \{i \in \mathbb{Z} | 1 \leq i \leq G\}$ are applied, the states $x_i[k]$ are measured and the procedure is repeated.

Note that in both algorithms Y_i is shared with all the other subsystems, but ΔU_i^{p+1} is only shared with the subsystems within Θ_i^H. For the MJDC, a central agent is needed to combine the results from the local control problems. For the NJDC no such central agent is needed.

3-3 Summary

In this chapter both a centralized and distributed MPC's have been designed using the model developed in chapter 2. To account for model mismatch and disturbances, integral action has been added to the controller by writing the model in velocity form. The objective function of the centrallized MPC, given in equation 3-9, is solved in Gurobi 8.1.0. The distributed MPC is solved using two different versions of the Jacobi algorithm, in which the local control objectives, given in equation 3-24, are also solved using Gurobi 8.1.0. The controllers are tested in chapter 4.

Algorithm 3.2 Modified Jacobian Algorithm

Given $i \in \mathbb{Z} | 1 \leq i \leq G,\ u_i[k-1]\ \forall i,\ Y_{ref},\ H,\ q > 0,\ R = r \cdot I \in \mathbb{R}^{H \times H},\ r > 0,\ S_1,\ S_2,\ C_{Tmin},\ C_{Tmax},\ \Delta U_i^0 = 0\ \forall i,\ w_i > 0\ \forall i$, while $\sum w_i = 1,\ p_{max} > 0$ and $\epsilon > 0$
$p \leftarrow 0,\ e_i \leftarrow \Gamma\epsilon,\ \Gamma > 1$
while $e_i > \epsilon\ \exists i\ and\ p \leq p_{max}$ **do**
    for all $\{i \in \mathbb{Z} | (dw_m | S_{dw_m} \in \Theta_{-i})[d_{i,dw_m} \geq H]\}$ **do** ▷ Do this in parallel
        Solve local control problem given in equation 3-27
    end for
    Combine the results with the rule given in equation 3-28
    for all $\{i \in \mathbb{Z} | 1 \leq i \leq G\}$ **do** ▷ Do this in parallel
        Share ΔU_i^{p+1} with S_i and all subsystems within Θ_i^H
        $e_i = \left\| \Delta U_i^{p+1} - \Delta U_i^p \right\|$
        Calculate Y_i using equation 3-22 or 3-23.
        Share Y_i with all other subsystems.
    end for
    $p \leftarrow p + 1$
end while
ΔU_i^p is the optimal control sequence $\forall S_i\ \{i \in \mathbb{Z} | 1 \leq i \leq G\}$

Chapter 4

Simulation Results

In this chapter the controllers introduced in chapter 3 will be tested in the medium fidelity model WindFarmSimulator (WFSim) [1] on a single core of a 2.3 GHz Intel(R) Core(QM) i7-3610QM processor. The controllers will be compared with the model predictive control (MPC) algorithm proposed by S. Boersma that was mentioned earlier in chapter 1 [10]. This controller will, from now on, be denoted as the baseline controller. The control model within this controller does not take the wake effects into account. By comparing the proposed controllers with this controller, it is possible to make conclusions about the advantages of including wake effect in the controller model. First, in section 4-1, the settings for the simulation cases will be defined. Next, in section 4-2, the results for the centralized controller will be given. After this, in section 4-3, the same will be done for the the distributed controllers. Lastly, in section 4-4, a summary of the results will be given. Conclusions drawn from the results will be given in chapter 5.

4-1 Simulation Settings

The controllers will be tested in setups with $G = 10$ and $G = 64$ turbines. The 10 turbine (10T) case is the same as used in chapter 2 for the validation for the controller model and can be seen in figure 2-4. The 64 turbine (64T) case can be seen in figure 4-1. The 10T case consists of $M = 2$ rows and $N = 5$ columns of turbines. The 64T case consists of $M = 8$ rows and $N = 8$ columns of turbines. The turbines have a diameter, D_r, of 90 meters. The distance between the turbines in the direction of the wind, δx_r, is 630 meters. This is equal to 7 turbine diameters, which is a common choice in wind farms. The distance between the turbines perpendicular to the wind, δy_r, is 378 meters. The free-stream wind speed, V_∞, is 7.5 m/s. All these settings can also be found in table 4-1. The tuning variables and settings for the controller model, WFSim and the controller can also be found in table 4-1. The tuning variables for the control model for the 10T case are the same as for the 10T case in section 2-2. The tuning variables for the 64 turbine case are found in the same way as for the 10 turbine case. The chosen sample time, $h = 1$, is chosen according to the relatively

fast changing reference signal. The weights q and r are found by taking q equal to 1 and trying different values for r, whilst taking the trade-off described in subsection 3-1-2 into account. The reasons for the choices made for the values of $C_{T\,max}$ and $C_{T\,min}$ are also given in subsection 3-1-2. The settings p_{max} and ϵ are only used for the distributed controller and defined in subsection 3-2-3 and algorithms 3.1 and 3.2. For the prediction horizon, H, a variety of choices will be made to customize the controller to the needs of different simulation cases.

In the simulation, both laminar and turbulent inflow conditions are considered. This turbulence is modelled by adding Gaussian noise on the boundaries within WFSim. It is important to note that this will differ from real life turbulence. It is, however, interesting for testing purposes and it will give an indication as to how the controllers will react to real-life turbulent inflow conditions.

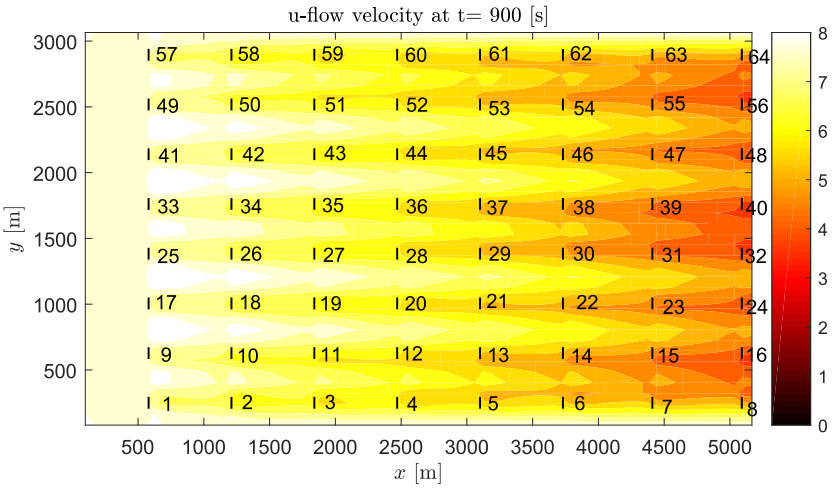

Figure 4-1: Visual representation of layout of 8x8 wind farm in WFSim.

Two different reference signals will be used. One reference signal will start at $0.8P_{greedy}$ and will have a filtered step change at 400 seconds to $1.1P_{greedy}$. P_{greedy} is the power that the wind farm would produce if all its turbines would operate at the so called Betz limit. A turbine, T_i, reaches the Betz limit if its thrust coefficient, $C_{T\,i}$ is equal to 8/9. At this thrust coefficient, the turbine will extract the theoretical maximum amount of energy from the airflow. This kind of control, where each turbine operates at its individual optimal settings, is called greedy control. P_{greedy} can be seen as the maximum steady-state power output of the wind farm. The other reference signal will be in the form

$$P_{ref}[k] = 0.8P_{greedy} + \gamma P_{greedy}\delta P[k], \tag{4-1}$$

where $\delta P[k]$ is a normalized RegD type AGC signal as defined by PJM [37] and γ is a variable for which different values will be used to obtain different simulation cases. If γ is chosen smaller than 0.2, P_{ref} will not exceed P_{greedy} and if γ is chosen bigger than 0.2, P_{ref} will exceed P_{greedy}.

Table 4-1: Simulation case details and settings for 10 turbine (10T) and 64 turbine (64T) case.

	Variable	Value	
		10T Case	64T Case
Windfarm Setup	G	10	64
	M	5	8
	N	2	8
	δx_r	630 m	630 m
	δy_r	378 m	378 m
	D_r	90 m	90 m
	V_∞	7.5 m/s	7.5 m/s
Control Model Settings	c_w	0.68	0.31
	c_{VV}	1.0	0.1
	c_{VC_T}	1.0	0.6
	c_{VA}	0.9	0.8
	c_{PV}	1.0	0.9
	c_{PC_T}	1.1	1.1
	τ	5	5
	h	1	1
Control Settings	q	1	1
	r	0.4	0.4
	$C_{T_{max}}$	8/9	8/9
	$C_{T_{min}}$	0.01	0.01
	p_{max}	200	200
	ϵ	$1 \cdot 10^{-2}$	$1 \cdot 10^{-2}$

	Variable	Value	
		10T Case	64T Case
WFSim Settings	type	'lin'	'lin'
	L_x	3500	5200
	L_y	778	3146
	N_x	100	80
	N_y	50	40
	powerscale	1	1
	forcescale	1.25	1.25
	μ	0	0
	ρ	1.2	1.2
	u_∞	7.5	7.5
	v_∞	0	0
	p_{init}	0	0
	lmu	2	2
	turbul	true	true
	n	2	2
	m	6	6

4-2 Centralized Controller

In this section, the centralized controller will be tested for the 10 turbine case and the 64 turbine case. First, in subsection 4-2-1, the controller will first be tested for the 10T case under laminar inflow conditions, then, in subsection 4-2-2, for the 10T case under turbulent inflow conditions and lastly, in subsection 4-2-3, for the 64T case under laminar inflow conditions.

4-2-1 Laminar 10 Turbine Case

Before the controller can be tested, a choice has to be made regarding the prediction horizon H found in objective function 3-9. The choice for the prediction horizon is influenced by the layout of the wind farm. The time it takes for the wake effects to travel from the most upwind turbine to the most downwind turbine in 10T case is equal to $\max_m d_{u_m,dw_m} = 304$ seconds and the sample time is $h = 1$ second. Therefore, H should be bigger than 304 to include all the dynamics within the wind farm. Because of this reason, as an initial choice, $H = 310$ is used. It will, however, be shown that this makes the controller too computationally slow for real-time control. Therefore, the controller will also be tested with a lower prediction horizon, i.e. $H = 160$.

Normalized RegD AGC signal with γ equal to 0.2

First, the centralized controller will be tested with the reference signal

$$P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k].$$

Note that P_{ref} will not exceed P_{greedy} as $\gamma = 0.2$. The results of this simulation for $H = 310$ can be seen in the left two plots in figure 4-2. As can be seen, the controller is able to properly track the reference signal. The root mean square error (RMSE) as defined in equation 2-32 and the time to update the control actions for one time step are given in table 4-2. The RMSE is equal to 0.010 MW and it takes averagely 8.13 seconds to update the control actions for one time step. This is longer than the sample time of 1 second, meaning that the controller is not fast enough for real-time control with these settings.

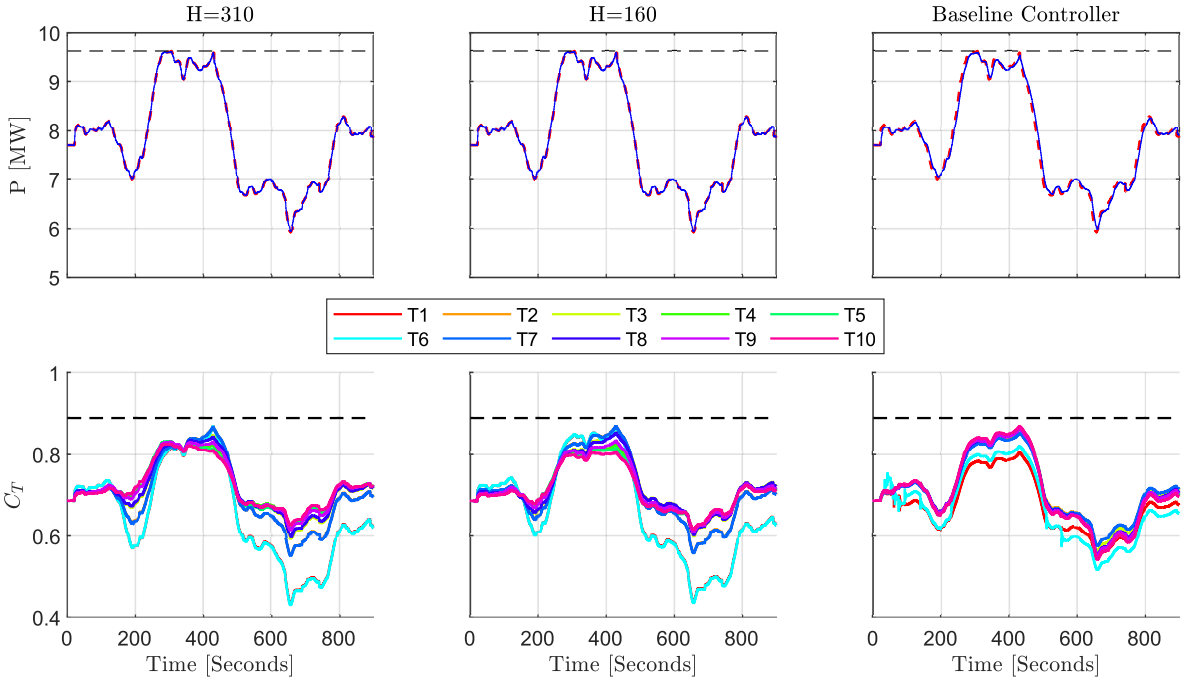

Figure 4-2: Tracking results with 10 turbine (10T) case with $P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k]$ for centralized controller with $H = 310$ (left), $H = 160$ (middle) and the baseline controller (right). In the top figures P^{ref} (red dashed), $\sum_{i=1}^{G} P_i$ (blue) and P_{greedy} (black dashed) can be seen. In the bottom figures $C_{Tmax} = 8/9$ (dashed black) and $C_{Ti}\forall i$ (see legend) can be seen.

A way to make the controller faster, is to lower the prediction horizon. The prediction horizon is lowered until the controller is fast enough for real-time control. This results in $H = 160$. Since it takes 152 seconds for the wake effects to travel down two columns of wind turbines, this prediction horizon is big enough include the interactions between three consecutive wind turbines. With these settings, as shown in table 4-2, it takes 0.76 seconds to calculate the optimal control actions for one time step. This is smaller than the sample time of $h = 1$ second. Thus, with $H = 160$ the controller is fast enough to provide real-time control. The results with $H = 160$ can be seen in the middle two plots in figure 4-2. As can be seen, there is barely a difference in the tracking performance compared to when $H = 310$ is used. The

Table 4-2: RMSE and calculation time for control update achieved with centralized controllers in WFSim for 10 turbine (10T) case for different prediction horizons H and reference $P_{ref}[k] = 0.8P_{greedy} + 0.2\delta P[k]$ under laminar inflow conditions. Lower values are better.

		RMSE [MW]	Time [s]
Centralized Controller	$H = 310$	0.010	8.13
	$H = 160$	0.010	0.76
Baseline Controller	N/A	0.077	N/A

RMSE is 0.010 MW, which is equal to the RMSE that was achieved when $H = 310$ was used. Later in this chapter, it will be shown that the prediction horizon becomes more important if the reference exceeds P_{greedy}.

In the right two plots, the results with the baseline controller can be seen. There seems to be only a slight difference in tracking behaviour. Looking at the RMSE makes the difference more evident. The baseline controller results in a RMSE of 0.077 MW, as shown in table 4-2. This is still marginal compared to the order of magnitude of the reference signal, but it is more than 7 times higher then with the controller proposed in this thesis.

Note that in table 4-2 the calculation time for the baseline controller is omitted. This is because this controller is written in YALMIP [38] in MatLab, which is a program that translates a given optimization problem to notation that can be used by a solver such as Gurobi. This gives a lot of overhead, compared to if the program was written directly in the used solver. Because of this, no fair estimate can be given about the speed of the baseline controller compared to the controllers proposed in the thesis.

Normalized RegD AGC signal with γ equal to 0.3

It is interesting to see how the controller will function if P_{ref} will exceed P_{greedy}. Therefore, in figure 4-3 the tracking results are shown for the reference

$$P_{ref}[k] = 0.8P_{greedy} + 0.3P_{greedy}\delta P[k]. \tag{4-2}$$

It can be seen that the centralized controller with $H = 160$ and $H = 310$ and the baseline controller are able to temporally exceed P_{greedy}. However, after a certain amount of time, the available energy in the wind farm becomes too low and the controllers cease to be able to follow the reference. Looking closely at the figure, it possible to see a difference in tracking behavior between $t = 400$ and $t = 450$. With $H = 310$, the central controller can follow the reference signal slightly longer than with $H = 160$. The baseline controller is able to follow the reference signal even shorter. Looking at the RMSE's given in table 4-3, the difference is evident. With $H = 160$ a RMSE of 0.067MW is achieved, compared to a RMSE of 0.055MW with $H = 310$ and a RMSE of 0.137MW. Interesting is the difference in the control signals. The proposed centralized controller decreases C_T for the upwind turbines and increases C_T for the downwind turbines before the surge in the reference signal that exceeds P_{greedy}. With $H = 310$ this behavior is stronger and starts earlier than with $H = 160$. In the baseline controller no such behavior is observed due to the lack of a wake model in the controller.

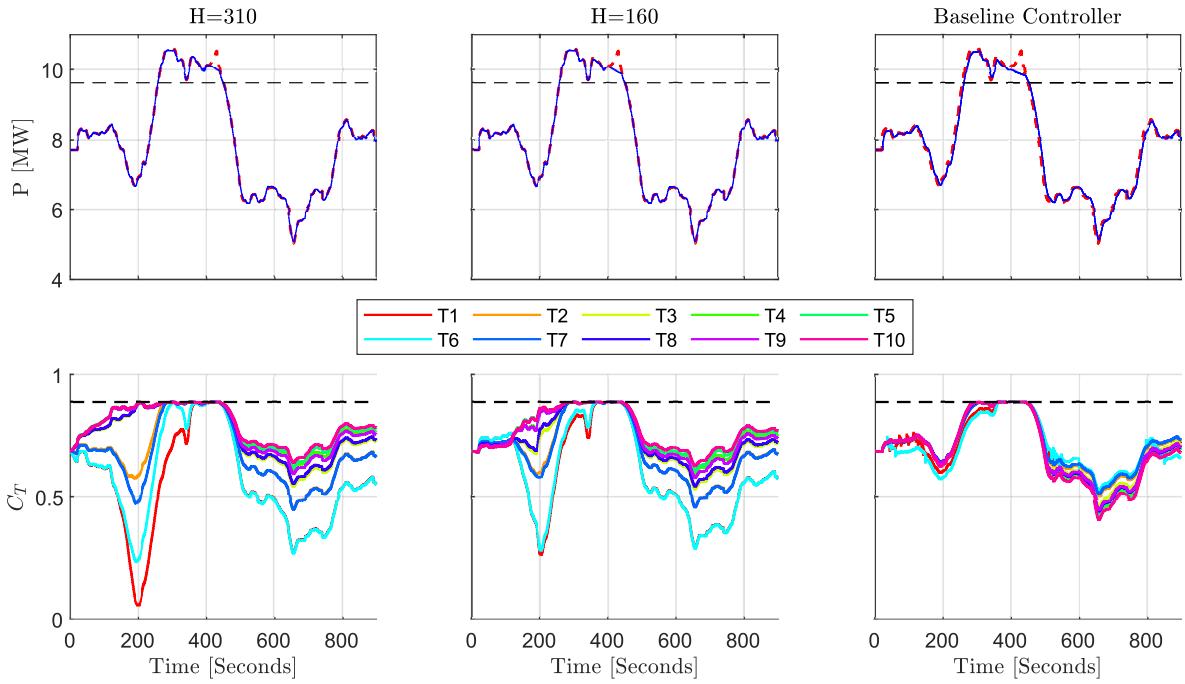

Figure 4-3: Tracking results in 10 turbine (10T) case with $P_{ref}[k] = 0.8P_{greedy} + 0.3P_{greedy}\delta P[k]$ for centralized controller with $H = 310$ (left), $H = 160$ (middle) and the baseline controller (right). In the top figures P^{ref} (red dashed), $\sum_{i=1}^{G} P_i$ (blue) and P_{greedy} (black dashed) can be seen. In the bottom figures $C_{Tmax} = 8/9$ (dashed black) and $C_{Ti}\forall i$ (see legend) can be seen.

Table 4-3: RMSE achieved with centralized controllers in WFSim for 10 turbine (10T) case for different prediction horizons H and reference $P_{ref}[k] = 0.8P_{greedy} + 0.3\delta P[k]$ under laminar inflow conditions. Lower values are better.

		RMSE [MW]
Centralized Controller	$H = 310$	0.055
	$H = 160$	0.067
Baseline Controller	N/A	0.137

Normalized RegD AGC signal with variety of values for γ

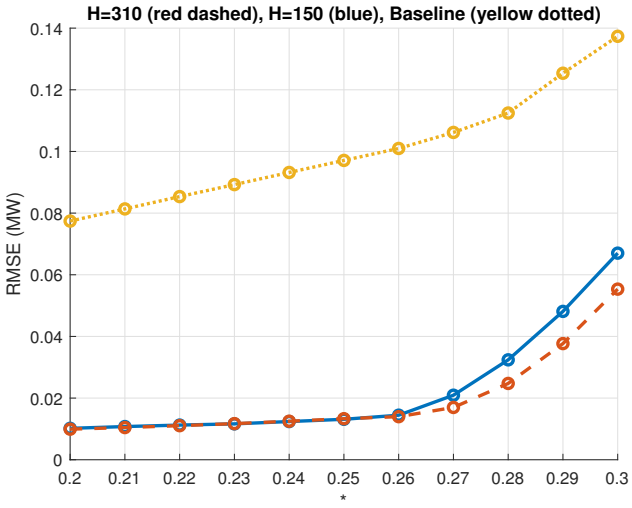

Figure 4-4: RMSE 10 turbine (10T) case with $P_{ref}[k] = 0.8P_{greedy} + \gamma P_{greedy}\delta P[k]$ for different values of γ for centralized controller with $H = 160$ and $H = 310$.

In figure 4-4 the RMSE for different values of γ in the reference signal

$$P_{ref}[k] = 0.8P_{greedy} + \gamma P_{greedy}\delta P[k]$$

can be seen for both $H = 160$ and $H = 310$. Until $\gamma = 0.26$ there is almost no difference between the results for $H = 160$ and $H = 310$. For $\gamma > 0.26$, the controller achieves better tracking results with $H = 310$ than with $H = 160$. In this figure also the RMSE for the baseline controller for different values of γ is shown. It can be seen that the RMSE of that controller is higher for all the values of γ and that for $\gamma \leq 0.26$, the RMSE increases slightly more with higher values of γ than the proposed controller.

Reference with step change

To make the way in which the proposed controller anticipates surges in the reference signal that exceed P_{greedy} more clear, this controller is also tested with a reference signal that starts at $0.8P_{greedy}$ and increases to $1.1P_{greedy}$ after 400 seconds. The results for the controller with $H = 310$, $H = 160$ and the baseline controller can be seen in figure 4-5. In this figure, the difference between the output signals is not clear. However, the difference between the input signals is. It can clearly be seen that the controller proposed in this thesis decreases C_T of the upwind turbines and increases C_T of the downwind turbines before the surge happens. It can also be seen that the controller starts doing this 310 seconds in advance to the surge when $H = 310$ and 160 seconds in advance when $H = 160$. In the baseline controller no such behavior is visible, due to the lack of a wake model in the controller.

Figure 4-6 zooms in on the part where the outputs exceed P_{greedy}. In this figure, the difference in the output signals is more evident. As can be seen, with both $H = 310$ and $H = 160$, the power outputs reach $1.1P_{greedy}$. With $H = 160$, the power output declines faster after

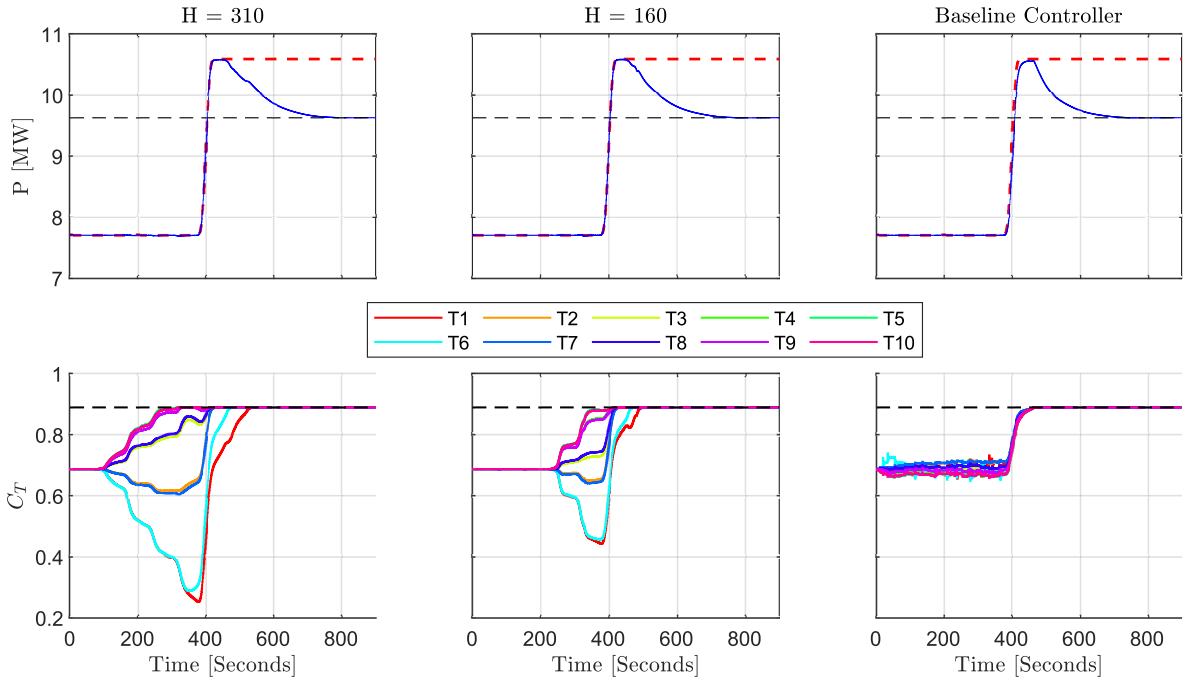

Figure 4-5: Tracking results for 10 turbine (10T) case with reference signal that starts at $0.8P_{greedy}$ and goes to $1.1P_{greedy}$ at 400 seconds for centralized controller with $H = 310$ (left), $H = 160$ (middle) and the baseline controller (right). In the top figures P^{ref} (red dashed), $\sum_{i=1}^{G} P_i$ (blue) and P_{greedy} (black dashed) can be seen. In the bottom figures $C_{T\,max} = 8/9$ (dashed black) and $C_{T_i} \forall i$ (see legend) can be seen.

reaching its top power output, than with $H = 310$. The power output of the baseline controller does not reach $1.1P_{greedy}$ and also declines faster after reaching its top power output. To get a better understanding of the performance of the controllers, the sum of the power outputs minus P_{greedy} from 400 seconds till 900 seconds is taken, i.e. $\sum_{k=400}^{900} h(P[k] - P_{greedy})$. This is equal to the area between the power outputs and P_{greedy} from $t = 400$ till the end of the simulation. Note that the result of this summation is an amount of energy. More precise, it results in the amount of energy that the wind farm will produce more than when the power output is equal to P_{greedy} during this time window. The values of this sum for the different controllers can be found in table 4-4. These values make the difference between the controllers even more evident. With $H = 310$ this value is 33% higher than with the baseline controller and with $H = 160$ this value is 22% higher.

Table 4-4: $\sum_{k=400}^{900} h(P[k] - P_{greedy})$ achieved with centralized controller in WFSim for 10 turbine (10T) case for different values of prediction horizons H and a reference signal that starts at $0.8P_{greedy}$ and goes to $1.1P_{greedy}$ at 400 seconds. Higher values are better.

		$\sum_{k=400}^{900} h(P[k] - P_{greedy})$ [MJ]
Centralized Controller	$H = 160$	128
	$H = 310$	140
Baseline Controller	N/A	105

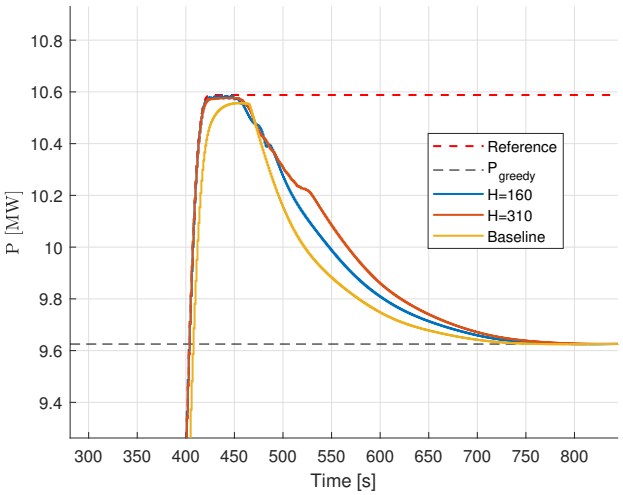

Figure 4-6: Tracking results for 10 turbine (10T) case with reference signal (red dashed) that starts at $0.8P_{greedy}$ and goes to $1.1P_{greedy}$ at 400 seconds for centralized controller with $H = 310$ (red), $H = 160$ (blue) and the baseline controller (yellow). This figure zooms in on the part where the power outputs exceed P_{greedy} (black dashed).

4-2-2 Turbulent 10 Turbine Case

It is also studied how the controller behaves under turbulent inflow conditions. As explained in the introduction of this chapter, the turbulent inflow behavior is approximated by adding Gaussian noise to the boundaries in WFSim.

For this simulation the prediction horizon will be taken equal to $H = 160$ to achieve real-time control. The results with these settings and the reference signal

$$P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k]$$

can be seen in the left two plots of figure 4-7. As stated in table 4-5, the $RMSE$ is 0.049 MW. It can be seen that the power output oscillates a lot around the power reference, but that the mean of the power output is still correct. Also the input signals oscillate; the controller is amplifying the disturbance. It is researched how this behavior is influenced by the weight on the change in the control signal, r. To this end, r is increased from $r = 0.4$ to $r = 1.0$. The results with these new settings can be seen in the right two figures in figure 4-7. It can be seen that both the output signals and the input signals oscillate less. As shown in table 4-5, the $RMSE$ is 0.040, which is less then with $r = 0.4$.

Table 4-5: RMSE achieved with centralized controller in WFSim for 10 turbine (10T) case for different values of weight r and reference $P_{ref}[k] = 0.8P_{greedy} + 2\delta P[k]$ under turbulent inflow conditions. Lower values are better.

	RMSE [MW]
$r = 0.4$	0.049
$r = 1.0$	0.040

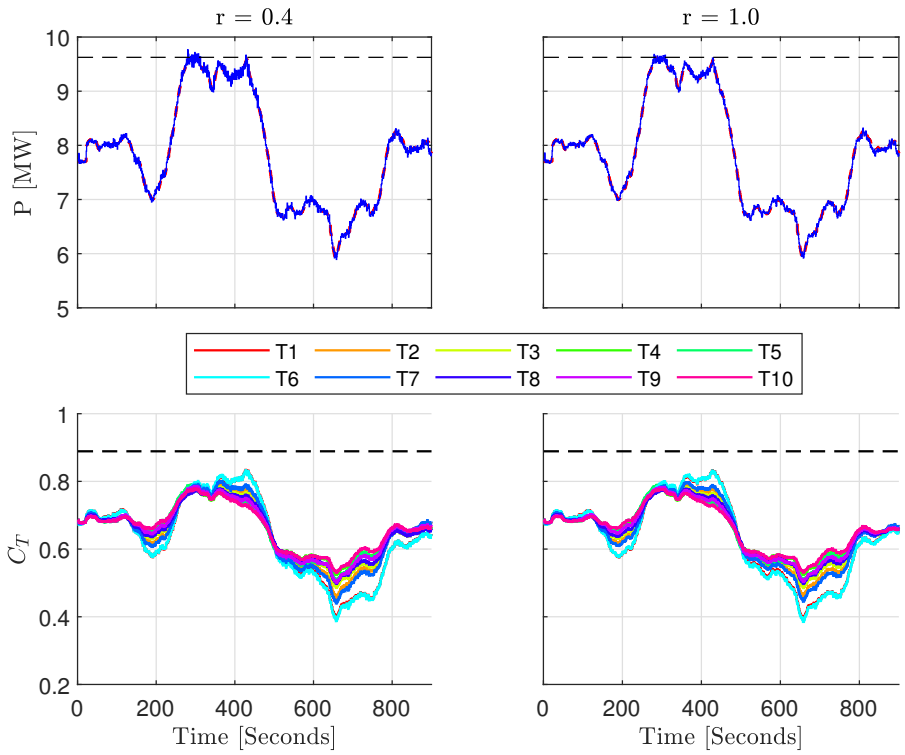

Figure 4-7: Tracking results with 'turbulence' for 10 turbine (10T) case with centralized controller with $P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k]$ and $H = 160$. In the left two plots $r = 0.4$ is used and in the right two plots $r = 1.0$. In the top figures the power reference, the total power output and P_{greedy} can be seen. In the bottom figures the inputs used to achieve this tracking can be seen.

4-2-3 Laminar 64 Turbine Case

Lastly, it is interesting to see how the centralized controller performs in a bigger wind farm. The controller has been tested on the 64 turbine case with

$$P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k].$$

To include the interaction between all the turbines in the wind farm. the prediction horizon would need to be bigger than $\max_m d_{u_m,dw_m} = 543$ seconds. This will, however, result in a very high computational costs. Therefore, instead, it is chosen to keep $H = 160$. Laminar inflow conditions are considered. The tracking results are shown in the left two plots of figure 4-8. As shown in table 4-6, the $RMSE$ is 0.020 MW. Thus, also for big wind farms, the centralized controller can achieve proper tracking. The time to calculate the optimal control actions for one time step is 6.65 seconds. This means that it is not possible to do real-time control in a large wind farm with the centralized controller with these settings. Therefore, the distributed controller, which will be tested in the following section, was introduced.

In the right two plots of figure 4-8, the results for a $64T$ wind farm with the baseline controller is shown. When looking at the figure, the tracking results look equal as to the proposed

controller. However, the RMSE of 0.22 MW is 11 times higher than with the proposed controller.

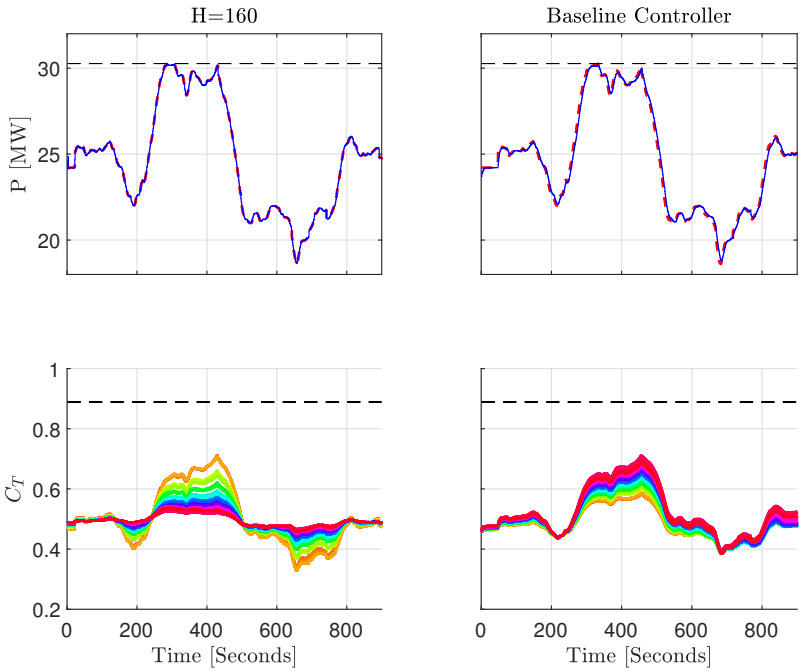

Figure 4-8: Tracking results for 64 turbine (64T) case with $P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k]$ for centralized controller with $H = 160$ (left) and the baseline controller (right). In the top figures P^{ref} (red dashed), $\sum_{i=1}^{G} P_i$ (blue) and P_{greedy} (black dashed) can be seen. In the bottom figures $C_{T\,max} = 8/9$ (dashed black) and $C_{T_i} \forall i$ can be seen.

Table 4-6: RMSE and calculation time for control update achieved with centralized controller and the baseline controller in WFSim for 64 turbine (10T) case for prediction horizon $H = 160$ and reference $P_{ref}[k] = 0.8P_{greedy} + 0.2\delta P[k]$ under laminar inflow conditions. Lower values are better.

		RMSE [MW]	Time [s]
Centralized Controller	$H = 160$	0.020	6.65
Baseline Controller	N/A	0.22	

4-3 Distributed Controller

In this section, the results of testing the two developed distributed controllers will be given. Also these controllers will be compared against the centralized controller and the active power control (APC) designed by S. Boersma et al. [10] that is taken as baseline controller. Both the 10 turbine and 64 turbine cases under laminar inflow conditions will be considered. The controllers are tested with both a prediction horizon of $H = 160$ and $H = 310$. The parameters for both controllers defined in algorithm 3.1 and 3.2 can be found in table 4-7. The rest of the parameters can be found in table 4-1.

Variable	Value
$p_{\max}$	200
ϵ	$1 \cdot 10^{-2}$

Table 4-7: Parameter choices for algorithms 3.1 and 3.2

As no computer with 64 cores is available to the author of this thesis, lines 4 until 11 in algorithm 3.1 and lines 4 until 6 and 8 until 13 in algorithm 3.2 will not be solved in parallel. However, it is still possible to give a indication for the time it would take to solve the controller problem if these lines were solved in parallel. This can be done by taking the maximum time that the algorithm spends on these lines for any of the sub-problems and adding this to the time it takes to run the rest of the controller. It is important to take the maximum time and not the mean time, because the algorithm has to wait until the parallel lines are performed for all of the subsystems before it can continue. This means that if this part of the algorithm is solved in parallel, it will take as long as the slowest controller.

4-3-1 Laminar 10 Turbine Case

First, the 10 turbine case under laminar inflow conditions is discussed for different reference signals.

Reference with step change

The results for the reference that starts at $0.8P_{greedy}$ and goes to $1.1P_{greedy}$ for both controllers with $H = 160$ can be seen in figure 4-9. As can be seen, just as with the centralized controller, both controllers lower the C_T of the upwind turbines and increase the C_T of the downwind turbines before the surge in the reference signal. This behavior is stronger in the modified Jacobian distributed controller (MJDC) than in the normal Jacobian distributed controller (NJDC). With the method explained before, it is approximated that, if the algorithms would be solved in parallel, the NJDC would take around 0.09 seconds to calculate the optimal control actions for a single time step and the MJDC around 0.92 seconds. The NJDC is faster than the MJDC and both controllers are fast enough to provide real-time control, as the calculation times are lower than the sample time $h = 1$ second.

Figure 4-10 zooms in on the part where the power outputs exceed P_{greedy}. To get a better comparison, in this figure also the tracking results for the centralized controller with $H = 160$ and the baseline controller is shown. As can be seen, both the centralized controller and

the MJDC perform equally well. The NJDC performs a little less, but still outperforms the baseline controller. Also note that both the NJDC and MJDC oscilate a little bit and that this behavior is stronger for the NJDC. As shown in table 4-8, for the MJDC $\sum_{k=400}^{900} h(P[k] - P_{greedy}) = 129MJ$ and for the NJDC $\sum_{k=400}^{900} h(P[k] - P_{greedy}) = 121MJ$. This value is 23% higher for the MJDC compared to the baseline controller and 15% higher for the NJDC.

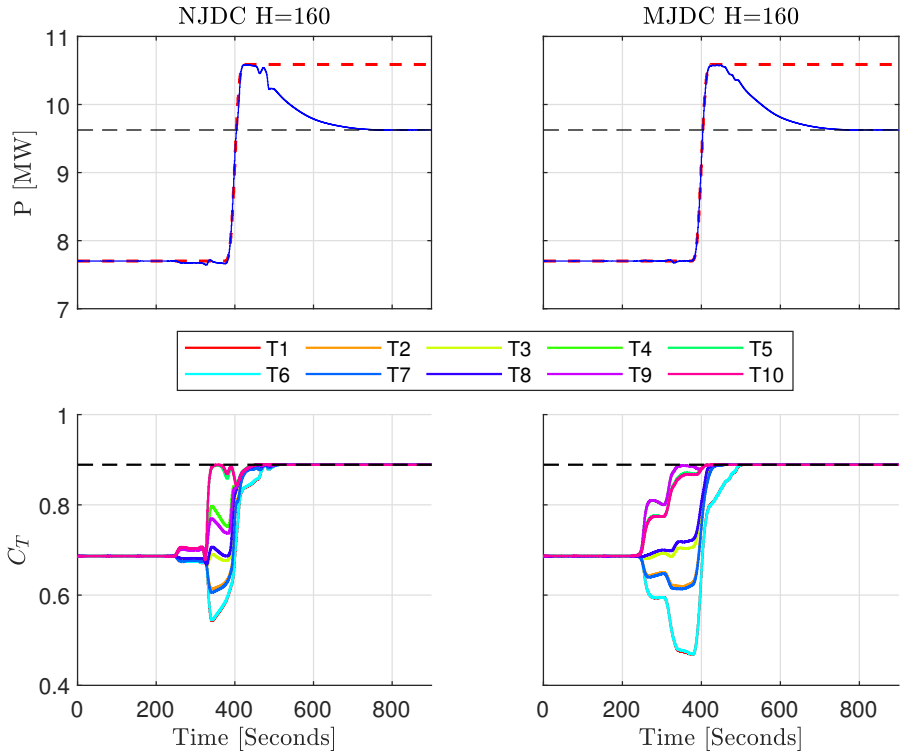

Figure 4-9: Tracking results for 10 turbine (10T) case with NJDC (left) and MJDC (right) with $H = 160$ and a reference that start at $0.8P_{greedy}$ and goes to $1.1P_{greedy}$ after 400 seconds. In the top figures P^{ref} (red dashed), $\sum_{i=1}^{G} P_i$ (blue) and P_{greedy} (black dashed) can be seen. In the bottom figures $C_{T\,max} = 8/9$ (dashed black) and $C_{T_i} \forall i$ can be seen.

Both controllers are also tested for the same test signal whilst using $H = 310$. The results can be seen in figure 4-11. As can be seen both controller now start to change the C_T of the turbines 310 seconds ahead of the surge in the reference. If the algorithms would be solved in parallel, the NJDC would take approximately 0.52 seconds to calculate the optimal control actions for the next time step and the MJDC 18 seconds. This means that the NJDC is still able to provide real time control with $H = 310$ and $h = 1$, but the MJDC is not. Figure 4-12 zooms in on the part where the power outputs exceed P_{greedy}. As can be seen, again the MJDC performs equally as well as the centralized controller with $H = 310$. The NJDC still performs a little less, but the difference seems to be smaller. Also note that both the controllers oscillate a little and that this behavior is still stronger for the NJDC. As stated in table 4-8, for the MJDC $\sum_{k=400}^{900} h(P[k] - P_{greedy}) = 140MJ$, which is 33% higher than with the baseline controller and for the NJDC $\sum_{k=400}^{900} h(P[k] - P_{greedy}) = 137MJ$, which is 30% higher than with the baseline controller.

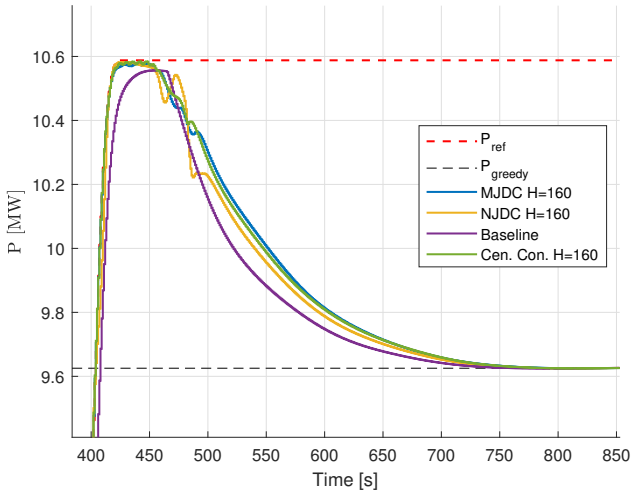

Figure 4-10: Tracking results for 10 turbine (10T) case with reference signal (red dashed) that starts at $0.8P_{greedy}$ and goes to $1.1P_{greedy}$ at 400 seconds for NJDC (yellow), MJDC (blue) and centralized controller (green) with $H = 160$ and the baseline controller (purple). This figure zooms in on the part where the power outputs exceed P_{greedy} (black dashed).

Table 4-8: $\sum_{k=400}^{900} h(P[k] - P_{greedy})$ and time to update control actions for single time step with distributed controllers in WFSim for 10 turbine (10T) case for different values of prediction horizons H and a reference signal that starts at $0.8P_{greedy}$ and goes to $1.1P_{greedy}$ at 400 seconds. Higher values of the sum are better.

		$\sum_{k=400}^{900} h(P[k] - P_{greedy})$ [MJ]	Time [s]
NJDC	$H = 160$	121	0.09
	$H = 310$	137	0.52
MJDC	$H = 160$	129	0.92
	$H = 310$	140	18
Baseline Controller	N/A	105	

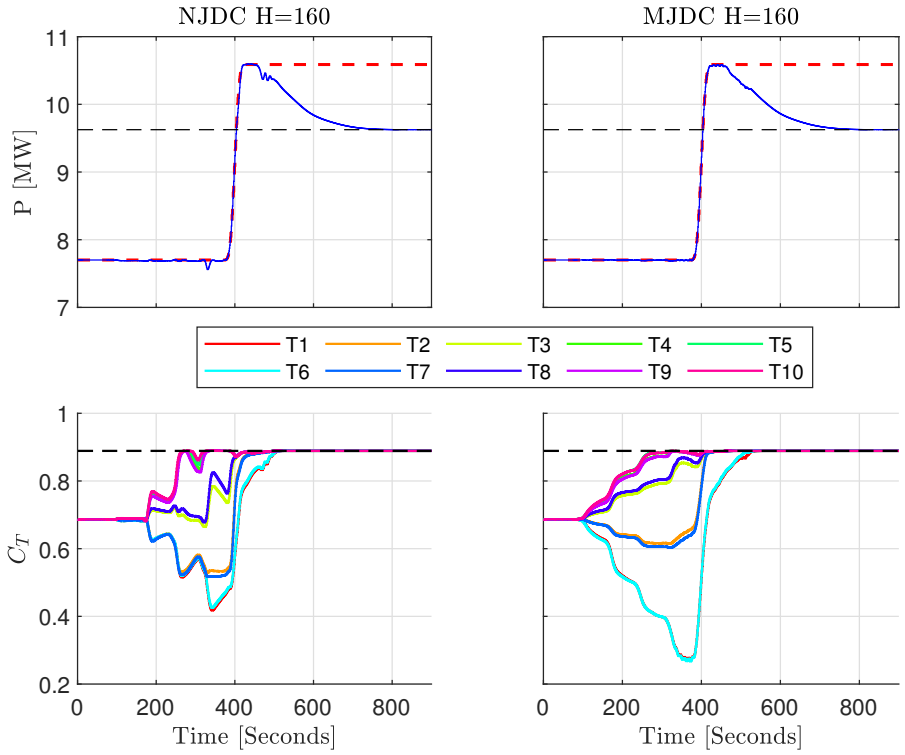

Figure 4-11: Tracking results for 10 turbine (10T) case with NJDC (left) and MJDC (right) with $H = 310$ and a reference that start at $0.8P_{greedy}$ and goes to $1.1P_{greedy}$ after 400 seconds. In the top figures P^{ref} (red dashed), $\sum_{i=1}^{G} P_i$ (blue) and P_{greedy} (black dashed) can be seen. In the bottom figures $C_{T\,max} = 8/9$ (dashed black) and $C_{Ti} \forall i$ can be seen.

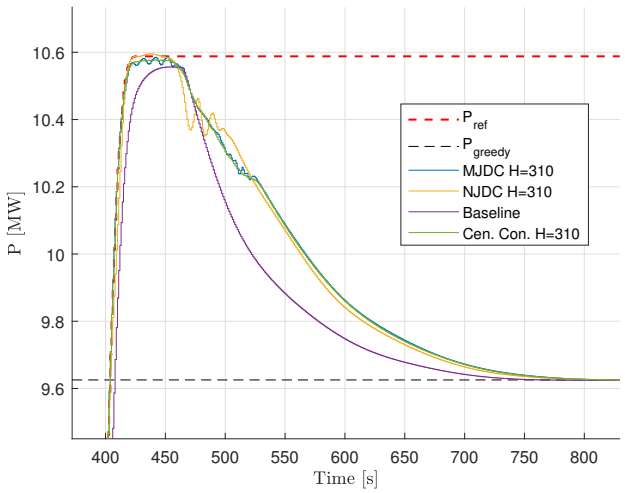

Figure 4-12: Tracking results for 10 turbine (10T) case with reference signal (red dashed) that starts at $0.8P_{greedy}$ and goes to $1.1P_{greedy}$ at 400 seconds for NJDC (yellow), MJDC (blue) and centralized controller (green) with $H = 310$ and the baseline controller (purple). This figure zooms in on the part where the power outputs exceed P_{greedy} (black dashed).

Normalized RegD AGC signal with γ equal to 0.2

The controllers are also tested with the RegD AGC signal:

$$P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k].$$

The MJDC is only tested with $H = 160$, as the version with $H = 310$ is not able to achieve real-time control. The results can be found in figure 4-13 and table 4-9. As can be seen, all the distributed controllers are able to track the reference properly. For all three versions of the distributed controller the RMSE is 0.017 MW. The difference between the controllers is more apparent when P_{ref} exceeds P_{greedy}. The RMSE is higher than for the centralized controller, but still marginal compared to the order of magnitude of the reference signal.

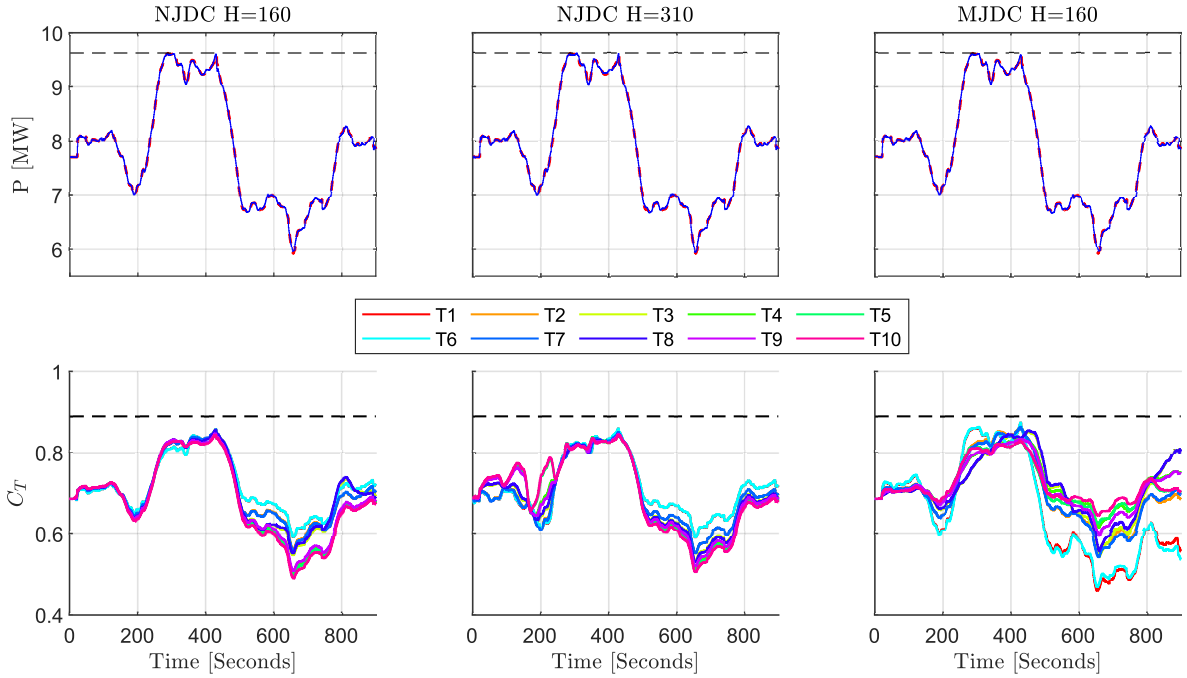

Figure 4-13: Tracking results for 10 turbine (10T) case with NJDCwith $H = 160$ (left) and $H = 310$ (middle) and with MJDC with $H = 160$ (right) with $P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k]$. In the top figures P^{ref} (red dashed), $\sum_{i=1}^{G} P_i$ (blue) and P_{greedy} (black dashed) can be seen. In the bottom figures $C_{T\,max} = 8/9$ (dashed black) and $C_{T_i} \forall i$ can be seen.

Table 4-9: RMSE achieved with distributed controllers in WFSim for 10 turbine (10T) case for different values of prediction horizons H and the reference signal $P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k]$. Lower values are better.

		RMSE [MW]
NJDC	$H = 160$	0.017
	$H = 310$	0.017
MJDC	$H = 160$	0.017
Baseline Controller	N/A	0.077

Normalized RegD AGC signal with γ equal to 0.3

To explore how the distributed controller behaves when the reference signal exceeds P_{greedy}, the next reference signal is used:

$$P_{ref}[k] = 0.8P_{greedy} + 0.3P_{greedy}\delta P[k].$$

The results are shown in figure 4-14. As can be seen, the distributed controllers are able to track a reference signal that exceeds P_{greedy} for a certain amount of time. Form the plot it is hard to tell the difference between the tracking quality of the three controllers. For the NJDC with $H = 160$ the RMSE is 0.079MW, for the same controller with $H = 310$ it is 0.078MW and for the MJDC with $H = 160$ it is 0.065MW. Here the difference between the RMSE of the two versions of the NJDC really small. Even more interesting is that the MJDC with $H = 160$ here outperforms the NJDC with $H = 310$. This might be due to the fact that the output signals of the NJDC sometimes oscillates. This increases the RMSE. Both RMSE are, however, still marginal and really close to the RMSE of the centralized controller and smaller than the RMSE of 0.137MW achieved with the baseline controller.

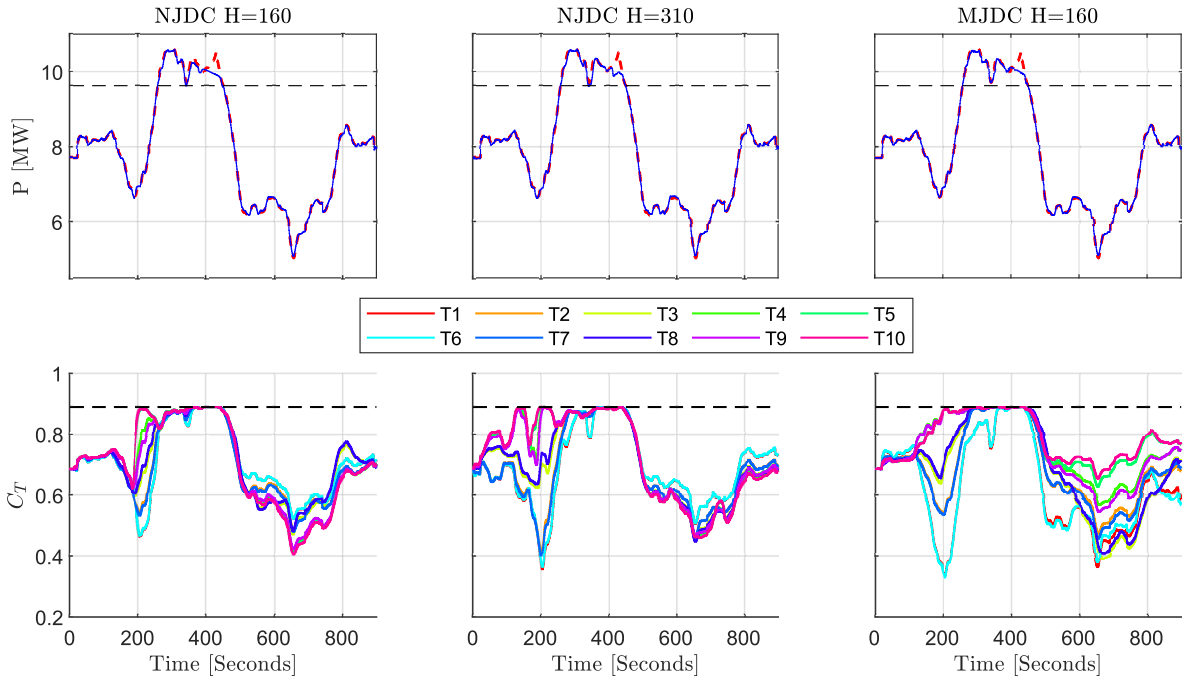

Figure 4-14: Tracking results for 10 turbine (10T) case with NJDC with $H = 160$ (left) and $H = 310$ (middle) and with MJDC with $H = 160$ (right) with $P_{ref}[k] = 0.8P_{greedy} + 0.3P_{greedy}\delta P[k]$. In the top figures P^{ref} (red dashed), $\sum_{i=1}^{G} P_i$ (blue) and P_{greedy} (black dashed) can be seen. In the bottom figures $C_{T_{max}} = 8/9$ (dashed black) and $C_{T_i} \forall i$ can be seen.

Table 4-10: RMSE achieved with distributed controllers in WFSim for 10 turbine (10T) case for different values of prediction horizons H and the reference signal $P_{ref}[k] = 0.8P_{greedy} + 0.3P_{greedy}\delta P[k]$. Lower values are better.

		RMSE [MW]
NJDC	$H = 160$	0.079
	$H = 310$	0.078
MJDC	$H = 160$	0.065
Baseline Controller	N/A	0.137

4-3-2 Laminar 64 Turbine Case

To test how the distributed controllers behave in larger wind farms, they have also been tested in the 64 turbine wind farm. The controllers have been tested with the reference signal

$$P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k].$$

The results can be seen in figure 4-15 and table 4-11. As can be seen all the controllers are able to track the reference signal properly. The RMSE with both versions of the NJDC is 0.042 MW and thus 2 times as big as with the centralized controller for the 64 turbine case. It is, however, still marginal in contrast to the reference signal. The MJDC scores even closer to the centralized controller with a RMSE of 0.033. The small difference between the two distributed controller is probably, again, caused by the fact that the output of the NJDC oscillates a bit more. The time it approximately takes to calculate the optimal control actions for a single time step if the algorithm is run in parallel within the NJDC with $H = 160$ is 0.11 seconds for the 64T case, compared to 0.09 seconds in the 10T case. For the same controller with $H = 310$, this is 0.58 seconds for the 64T case, compared to 0.52 seconds in the 10T case. For the MJDC with $H = 160$ this is 1.06 seconds for the 64T case, compared to 0.92 seconds for the 10T case. Notice that these times are slightly higher for the 64T case than for the 10T case and that this goes against the idea that the controllers would be equally fast, regardless of the size of the wind farm. The difference is, however, small and it is primarily caused by data handling in the controller. When just looking at the time that the controllers would approximately spend on the lines 4 until 11 in algorithm 3.1 and lines 4 until 6 and 8 until 13 in algorithm 3.2 if they would be calculated in parallel, the NJDC with $H = 160$ takes 0.08 seconds in both the 10T and 64T case. For the same controller with $H = 310$ this is 0.51 seconds for the 10T case and 0.53 seconds for the 64T case and for the MJDC with $H = 160$ this is 0.91 seconds for the 10T case and 0.94 seconds for the 64T case. The differences between these times are substantially smaller.

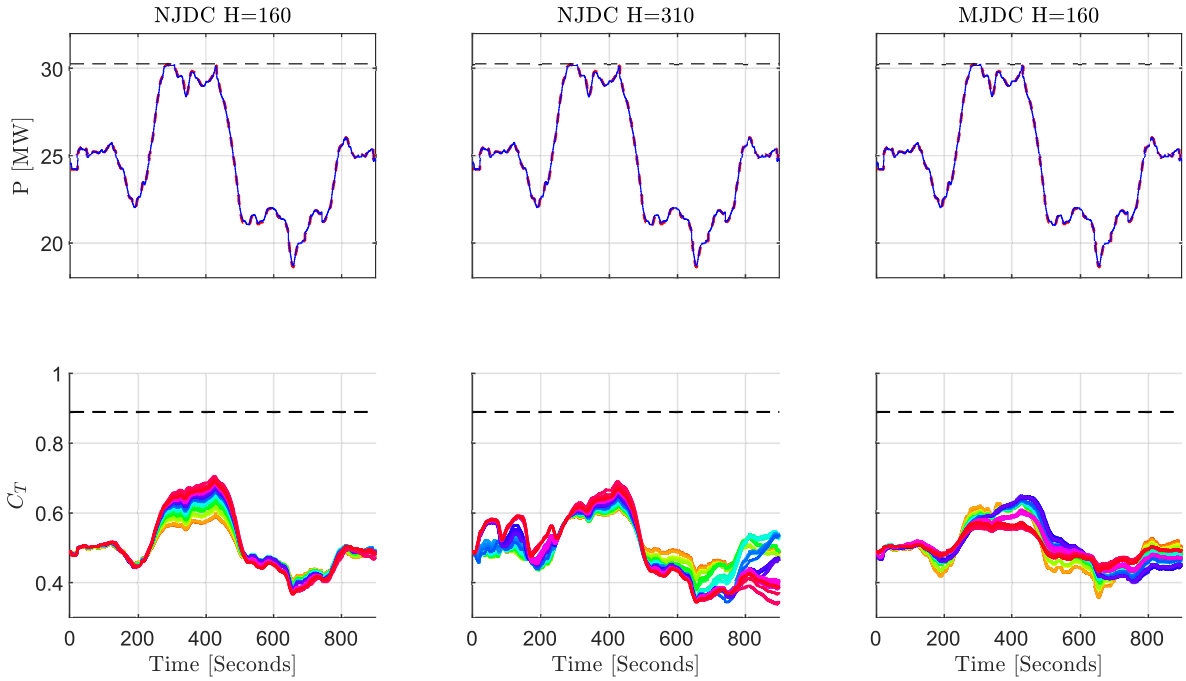

Figure 4-15: Tracking results for 64 turbine (64T) case with NJDC with $H = 160$ (left) and $H = 310$ (middle) and with MJDC with $H = 160$ (right) with $P_{ref}[k] = 0.8P_{greedy} + 0.3P_{greedy}\delta P[k]$. In the top figures P^{ref} (red dashed), $\sum_{i=1}^{G} P_i$ (blue) and P_{greedy} (black dashed) can be seen. In the bottom figures $C_{T\,max} = 8/9$ (dashed black) and $C_{T\,i} \forall i$ can be seen.

Table 4-11: RMSE and time to update control actions with distributed controllers in WFSim for 64 turbine (64T) case for different values of prediction horizons H and the reference signal $P_{ref}[k] = 0.8P_{greedy} + 0.2P_{greedy}\delta P[k]$. Lower values are better.

		RMSE [MW]	Time [s]
NJDC	$H = 160$	0.043	0.11
	$H = 310$	0.043	0.58
MJDC	$H = 160$	0.033	1.06
Baseline Controller	N/A	0.137	

4-4 Summary

In this chapter the centralized and distributed controllers have been tested in the medium-fidelity wind farm model WFSim with a variety of simulation cases. The results are summarized in tables 4-12, 4-13 and 4-14. The most important results are:

All proposed controller are able to properly track the given reference signals. All controllers are able to track the reference signals with RMSE values smaller than the baseline controller with which the results are compared.

The proposed controllers decrease the input signals of the upwind turbines and increase the input signals of the downwind turbines before a reference exceeds P_{greedy}. How far ahead this happens, depends on the prediction horizon. For the distributed controllers, this behavior is more apparent in the MJDC than in the NJDC

The proposed controllers are able to exceed P_{greedy} more and longer than the baseline controller. This is especially evident when looking at the RMSE's and the sums $\sum_{k=400}^{900} h(P[k] - P_{greedy})$.

The centralized controller becomes computationally slower for bigger wind farms. For the 10 turbine case, the centralized controller was able to compute the control actions for a single time step faster than the sample time of $h = 1$ second. For the 64 turbine case, the centralized controller was substantially slower.

The distributed controllers stay approximately equally computationally fast, regardless of the size of the wind farm. That is, if it is assumed that a separate processor core is available for each control problem. The NJDC was able to update the control signals for a single time step in a time smaller than the sample time of $h = 1$ second. The MJDC was slower. For this controller it takes around 1 second to update the control actions for a single time step.

The proposed controllers also work under turbulent inflow conditions. By increasing the weight on the change in the control signals, r, the controller reacts less to the turbulent inflow conditions. Within WFSim turbulent inflow conditions are approximated by adding Gaussian noise to the boundaries.

Table 4-12: RMSE achieved and calculation time [s] to update control actions with centralized and distributed controllers in WFSim for 10 turbine (10T) and 64 turbine (64T) cases for different prediction horizons H and different values of γ in reference $P_{ref}[k] = 0.8P_{greedy} + \gamma\delta P[k]$ under laminar inflow conditions. Lower values are better.

		RMSE [MW]			Time [s]	
		10T $\gamma = 0.2$	10T γ=0.3	64T $\gamma = 0.2$	10T	64T
Centralized Controller	$H = 160$	0.010	0.067	0.020	0.76	6.65
	$H = 310$	0.010	0.055	N/A	8.13	N/A
NJDC	$H = 160$	0.017	0.079	0.043	0.09	0.11
	$H = 310$	0.017	0.078	0.043	0.52	0.58
MJDC	$H = 160$	0.017	0.065	0.033	0.92	1.06
Baseline Controller	N/A	0.077	0.137	0.22	N/A	N/A

Table 4-13: RMSE achieved with centralized controller in WFSim for 10 turbine (10T) case for different values of weight r and reference $P_{ref}[k] = 0.8P_{greedy} + 2\delta P[k]$ under turbulent inflow conditions. Lower values are better.

	RMSE [MW]
$r = 0.4$	0.049
$r = 1.0$	0.040

Table 4-14: $\sum_{k=400}^{900} h(P[k] - P_{greedy})$ achieved with centralized and distributed controllers in WFSim for 10 turbine (10T) case for different values of prediction horizons H and a reference signal that starts at $0.8P_{greedy}$ and goes to $1.1P_{greedy}$ at 400 seconds. Higher values are better.

		$\sum_{k=400}^{900} h(P[k] - P_{greedy})$ [MJ]
Centralized Controller	$H = 160$	128
	$H = 310$	140
NJDC	$H = 160$	121
	$H = 310$	137
MJDC	$H = 160$	129
	$H = 310$	140
Baseline Controller	N/A	105

Chapter 5

Conclusions

As the share of electrical power produced by wind farms increases, it is presumable that it becomes problematic to maintain a stable power grid. On the power grid, the amount of energy generated and consumed should be in balance. As wind power has a unpredictable and fluctuating nature, its increasing use can make it problematic to maintain this balance. To overcome this problem, this thesis focused on designing a control architecture that not only stabilizes the power output of wind farms, but is also able to provide real-time power reference tracking. With power reference tracking, it is possible for grid operators to adapt the power production to a change in the power demand and to counteract fluctuations introduced by other power generators. It is chosen to use model predictive control (MPC), such that the (delayed) wake dynamics within the wind farm can be taken into account. Wakes are areas downwind from turbines with decreased wind speed and increased turbulence. These wakes cause the wind turbines within a wind farm to influence each other. It is envisioned that taking these effects into account will create more freedom to temporally operate above the theoretical steady-state maximum power output, P_{greedy}.

MPC requires a controller model that is able to simulate the behavior of the wind farm. Therefore, in this thesis, a new control-oriented low-fidelity wind farm model that takes the wake dynamics into account, is developed. The model is based on the Frandsen's model and the Actuator Disk Model (ADM). The model is made dynamic by using the Taylor frozen turbulence hypothesis. The resulting model is validated against data from the medium fidelity wind farm model called WindFarmSimulator (WFSim) [1].

The developed model is used within a centralized MPC architecture. In this controller, the model is rewritten in the so called 'velocity form' to incorporate integral action. This is necessary to ensure tracking despite model mismatches and (unknown) disturbances. Although this controller was able to provide real-time control for small (2x5 turbine) wind farms, for big wind farms the order of the model becomes to large to provide real-time control. Therefore, in this thesis, two different distributed MPC architectures are proposed. With distributed control, the central control problem is divided into smaller local control problems that will be solved on local controllers that communicate with each other. This makes it possible to also solve large complex control problems in real time. An additional benefit of distributed

control, is that if one of the local controllers fails, the rest of the wind farm can still be controlled. This makes the system more fail-safe. Also in these distributed controllers, the model is rewritten in the 'velocity form' to incorporate integral action. The distributed controllers are based on the Jacobi algorithm as defined by A.N. Venkat et al. [36]. One controller (the normal Jacobian distributed controller (NJDC)) uses this algorithm exactly as defined in [36], the other (the modified Jacobian distributed controller (MJDC) uses the same algorithm but with a modification as explained in chapter 3.

The developed controllers are tested in chapter 4 in a medium fidelity wind farm model called WFSim [1]. From these simulations, the next conclusions about the developed wind farm model and controllers can be made:

Taking wake effects into account in a MPC for APC in a wind farm gives more freedom to temporally exceed P_{greedy}. The proposed controllers were compared to a MPC developed by Boersma et al. [10] in which the wake effects are not taken into account. This controller was taken as the baseline controller. It was shown in chapter 4 that the proposed controllers were able to exceed P_{greedy} more and longer than the baseline controller. This is especially evident when looking at figures 4-6, 4-10 and 4-12 and table 4-14. The controllers anticipate the surge in the reference signal by decreasing C_T for the upwind turbines and increasing C_T for the downwind turbines. By doing so, the upwind turbines will let more wind power flow into the farm. As it takes time for air to travel through the wind farm, this extra power will be available to the downwind turbines at a later time, such that P_{greedy} can be exceeded longer. This effect can be seen more clear in the MJDC than in the NJDC. In the baseline controller no such behavior was witnessed. Also interesting to note, is that the prediction horizon H has influence on how much and long P_{greedy} can be exceeded. If the prediction horizon is larger, the controller will start to anticipate the surge in the reference signal earlier.

The proposed distributed controllers are able to provide real-time control in large wind farms. That is, if it is assumed that a processor core is available for each sub-problem. Here, each sub-problem consists of a single turbine in the NJDC and a cluster of turbines in the MJDC. It was shown that, under this assumption, the time it takes to calculate the control actions for a single time step stays approximately equal regardless of the size of the wind farm. For the NJDC this time was lower than the sample time of $h = 1$ second with the used settings, meaning that this controller is fast enough for real-time control. The MJDC took around 1 second to calculate the control inputs for a single time step. This means that the MJDC is too slow to provide real-time control. It is however really close to providing real-time control. A solution would be to update the control actions every two seconds. Other solutions would be increasing the sample time and/or decreasing the prediction horizon.

The centralized controller is not able to provide real-time control in large wind farms. For the centralized controller, the time to calculate the control settings for a single time step increased with the size of the wind farm. For large wind farms, the centralized controller will take longer to update the control actions than the sample time of $h = 1$ seconds. This means that it will not be fast enough to provide real-time control in these wind farms. For a small 2x5 turbine wind farm, the calculation time was 0.76 seconds. For a large 8x8 turbine wind farm, this time was 6.65 seconds.

Adding integral action by rewriting the model in the velocity form causes the controller to compensate for model mismatches. In chapter 2 it was shown that there were model mismatches between the proposed controller model and the simulation model WFSim. In chapter 4 it was shown that, despite these mismatches, the controllers were able to properly follow a reference signal.

Of the two developed distributed controllers, the MJDC is able to exceed P_{greedy} longer and more than the NJDC. This can be concluded from the results shown in table 4-8. Also, the way in which the controllers anticipate a surge in the reference that exceeds P_{greedy} by decreasing the C_T at the upwind turbines is more noticeable in the MJDC than in the NJDC.

The controllers are able to properly track under turbulent inflow conditions, although laminar inflow conditions are assumed in the controller model. This can be concluded from the results presented in figure 4-7. The controller amplifies the effects of the turbulence. Increasing the weight, r, on the change of the input signals reduced this effect. Turbulent inflow conditions were approximated by adding Gaussian noise to the boundaries.

The proposed controller model is fast and accurate enough to function in a MPC algorithm for active power control (APC) in wind farms. That is, if integral action is added to the controller to account for model mismatches. In chapter 2, it was already shown that the model is able to give an estimate for the power output of a wind farm. In chapter 4 it was shown that the controllers in which the model was used, were able to properly track reference signals. The model was also able to predict whether the available power in the wind farm will be too low to follow a given reference signal. Because of this, the controller is able to anticipate this by decreasing C_T for the upwind turbines and increasing C_T for the downwind turbines.

Summarizing, it can be concluded that the goal of creating a distributed MPC that can provide real-time power reference tracking in wind farms whilst taking the wake dynamics into account was successfully reached in this thesis.

Chapter 6

Discussion and Recommendations

In this chapter the proposed controller model and controllers are discussed and recommendations are made.

First of all, it is interesting to note that, although within the controller model laminar inflow conditions are assumed, the controllers are still able to properly track reference signals in approximated turbulent inflow conditions. Nonetheless, tracking results and/or robustness for turbulent inflow conditions might improve if turbulent inflow conditions were assumed in the controller model. It is, however, debatable whether it would be possible to simulate turbulent inflow conditions in a control oriented low-fidelity wind farm model. Simulating turbulence is generally computationally heavy. Though, it should be possible to address turbulent inflow conditions using stochastic control such as proposed in a paper by Boersma et al. [27]. This is interesting for future research.

Moreover, it is interesting to note that besides creating the possibility to exceed P_{greedy} more and longer, taking the wake dynamics into account also makes it possible to provide additional information to the grid operators. With the control architectures proposed in this thesis, it is possible to predict whether certain reference signals can or can not be tracked by a wind farm depending on the maximum available energy. This is useful information for grid operators.

Furthermore, the simulations presented in this thesis show that the developed concepts work. But, to further validate controllers, the controllers should also be tested in high-fidelity and, eventually, in a wind tunnel scale model and/or a real wind farm. It is possible to do this in future work.

To add, future work could also focus on extending the controller model to work for wind farms with non-rectangular layouts and arbitrary wind directions.

One thing that is debatable, is whether it is possible for grid operators to accurately predict the energy demand. This was assumed to be possible to provide the reference signal to the wind farm that extends into the future as far as the used prediction horizons in the controllers. It should, however, be possible for grid operators to give an estimate for the future energy demand. Short term load forecasting is an existing and developing field of research and should be able to provide such estimates [39].

Furthermore, in the controller model and the controllers, C_T is used as the input signal, as is a common choice within wind farm control. This choice is made to simplify the problem at hand. It is, however, not possible to control C_T directly. As proposed before, a possibility is to use the controllers proposed in this thesis as wind farm level controllers and let local controllers steer the wind turbines to the demanded C_T. Another possibility is to use a turbine model that uses the yaw, pitch and generator torque as inputs. A advantage of this solution is that it creates the opportunity to use the wind turbine dynamics to further exceed P_{greedy} as shown in [14]. A drawback from this way of working is that it makes the control problem more complex. These two solutions could be further explored in future work.

Also, it is interesting to use true parallel processing to solve the used distributed control algorithms. Because not enough processor cores were available on the computer used for the simulations, these algorithms were run sequentially. In the introduction of section 4-3 it is explained how the time is calculated that it would take to update the control actions if the algorithms were solved in parallel. With this method, the communicational delay between the processor cores and the overhead are not taken into account. Also the communicational delay between the controller and the turbines is not taken into account. This means that, in reality, each update cycle will take a certain amount of time ϵ longer.

Lastly, it is interesting to including dynamic axial induction control and/or wake redirection via yawing into the proposed controllers, such that P_{greedy} can be exceeded further. This could also be done in future work. It is shown that dynamic axial induction control can increase the maximum available energy in a wind farm in multiple papers by Munters et al. and Frederik et al. [28, 40, 41, 42]. In a paper by Boersma et al. wake redirection is used in combination with a active power control (APC) scheme to increase the available power in a wind farm [10].

Appendix A

Vectors and Sub-Matrices of State-Space Notation of Proposed Wind Farm Model

In this appendix the vectors and sub-matrices of the state-space notation of the proposed wind farm model are given. They are given for a wind farm with a single row m of turbines. For wind farms with multiple rows of wind turbines, these vectors and matrices can be defined for each row in the same way as described in this appendix. They can than be stacked together as explained in chapter 2. Notation used within this appendix is defined in chapter 2.

The states $x_n, \{n \in \mathbb{Z} | 1 \leq n \leq N\}$ for each subsystem $S_n, \{n \in \mathbb{Z} | 1 \leq n \leq N\}$ are given by

$$x_1[k] = \begin{bmatrix} \delta\hat{V}_{d,1} \\ \tilde{C}_{Td,1} \\ \tilde{C}_{T1} \end{bmatrix}, \quad x_{N-1} = \begin{bmatrix} \hat{V}_{N-1} \\ \delta\hat{V}_{d,N-1} \\ \tilde{C}_{TN-1} \end{bmatrix}, \quad x_N = \begin{bmatrix} \hat{V}_N \\ \tilde{C}_{TN} \end{bmatrix}$$

and

$$x_n = \begin{bmatrix} \hat{V}_n \\ \delta\hat{V}_{d,n} \\ \tilde{C}_{Td,n} \\ \tilde{C}_{Tn} \end{bmatrix} \forall \{n \in \mathbb{Z} | 2 \leq n \leq N-1\},$$

where

$$\delta\hat{V}_{d,1} = \begin{bmatrix} \delta\hat{V}_1[k-1] \\ \delta\hat{V}_1[k-2] \\ \vdots \\ \delta\hat{V}_1[k-d_{1,2}+1] \end{bmatrix},$$

$$\delta\hat{V}_{d,n} = \begin{bmatrix} \sum_{i=1}^{n-1}\left(\delta\hat{V}_i[k-d_{i,n}] + \frac{\partial\delta V_n}{\partial A_{n+1}}\Big|_{x_0}\frac{1}{2}A_R\frac{c_w}{1-c_w}\tilde{C}_{Ti}[k-d_{i,n}]\right) \\ \sum_{i=1}^{n}\delta\hat{V}_i[k-d_{i,n}-1] \\ \sum_{i=1}^{n}\delta\hat{V}_i[k-d_{i,n}-2] \\ \vdots \\ \sum_{i=1}^{n}\delta\hat{V}_i[k-d_{i,n+1}+1] \end{bmatrix} \forall\{n\in\mathbb{Z}|2\leq n\leq N-1\},$$

$$\tilde{C}_{Td,1} = \begin{bmatrix} \tilde{C}_{T1}[k-1] \\ \tilde{C}_{T1}[k-2] \\ \vdots \\ \tilde{C}_{T1}[k-d_{1,2}+1] \end{bmatrix}$$

and

$$\tilde{C}_{Td,n} = \begin{bmatrix} \sum_{i=1}^{n-1}\tilde{C}_{Ti}[k-d_{i,n}] \\ \sum_{i=1}^{n-1}\tilde{C}_{Ti}[k-d_{i,n}-1] \\ \sum_{i=1}^{n-1}\tilde{C}_{Ti}[k-d_{i,n}-2] \\ \vdots \\ \sum_{i=1}^{n-1}\tilde{C}_{Ti}[k-d_{i,n+1}+1] \end{bmatrix} \forall\{n\in\mathbb{Z}|2\leq n\leq N-2\}.$$

Such that the subsystems $S_n, \{n\in\mathbb{Z}|1\leq n\leq N\}$ can be expressed by

$$\begin{aligned} x_1[k+1] = &\underbrace{\begin{bmatrix} A_{\delta\hat{V}_{d,1}\delta\hat{V}_{d,1}} & 0 & A_{\delta\hat{V}_{d,1}\tilde{C}_{T1}} \\ 0 & A_{\tilde{C}_{Td,1}\tilde{C}_{Td,1}} & A_{\tilde{C}_{Td,1}\tilde{C}_{T1}} \\ 0 & 0 & A_{\tilde{C}_{T1}\tilde{C}_{T1}} \end{bmatrix}}_{A_{1,1}} x_1[k] + \underbrace{\begin{bmatrix} 0 \\ 0 \\ B_{\tilde{C}_{T1}C_{T1}} \end{bmatrix}}_{B_1} C_{T1}[k] \\ &+ \underbrace{\begin{bmatrix} C_{x\delta\hat{V}_{d,1}C_{T0,1}} & C_{x\delta\hat{V}_{d,1}\delta V_{0,1}} & C_{x\delta\hat{V}_{d,1}A_{0,1}} & C_{x\delta\hat{V}_{d,1}A_{0,2}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} C_{T0,1} \\ \delta V_{0,1} \\ A_{0,1} \\ A_{0,2} \end{bmatrix}}_{c_{x1}}, \end{aligned} \tag{A-1}$$

$$\hat{P}_1[k] = \underbrace{\begin{bmatrix} 0 & 0 & C_{\hat{P}_1\tilde{C}_{T1}} \end{bmatrix}}_{C_1} x_1[k] + \underbrace{\begin{bmatrix} C_{y\hat{P}_1C_{T0,1}} & C_{y\hat{P}_1P_{0,1}} \end{bmatrix}\begin{bmatrix} C_{T0,1} \\ P_{0,1} \end{bmatrix}}_{c_{y1}},$$

$$
\begin{aligned}
x_{N-1}[k+1] = &\underbrace{\begin{bmatrix} 0 & 0 & 0 \\ A_{\delta\hat{V}_{d,N-1}\hat{V}_{N-1}} & A_{\delta\hat{V}_{d,N-1}\delta\hat{V}_{d,N-1}} & A_{\delta\hat{V}_{d,N-1}\tilde{C_T}_{N-1}} \\ 0 & 0 & A_{\tilde{C_T}_{N-1}\tilde{C_T}_{N-1}} \end{bmatrix}}_{A_{N-1,N-1}} x_{N-1}[k] \\
&+ \underbrace{\begin{bmatrix} 0 & A_{\hat{V}_{N-1}\delta\hat{V}_{d,N-2}} & 0 & 0 \\ 0 & A_{\delta\hat{V}_{d,N-1}\delta\hat{V}_{d,N-2}} & A_{\delta\hat{V}_{d,N-1}\tilde{C_T}_{d,N-2}} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{N-1,N-2}} x_{N-2}[k] \\
&+ \underbrace{\begin{bmatrix} 0 \\ 0 \\ B_{\tilde{C_T}_{N-1}C_{T\,N-1}} \end{bmatrix}}_{B_{N-1}} C_{T\,N-1}[k] \\
&+ \underbrace{\begin{bmatrix} 0 & C_{x\delta\hat{\hat{V}}_{d,N-1}C_{T\,0,N-1}} & 0 & 0 \\ C_{x\hat{V}_{N-1}V_{0,1}} & 0 & 0 & 0 \\ 0 & C_{x\delta\hat{V}_{d,N-1}V_{0,N-1}} & 0 & 0 \\ 0 & C_{x\delta\hat{V}_{d,N-1}\delta V_{0,N-1}} & 0 & 0 \\ 0 & C_{x\delta\hat{V}_{d,N-1}A_{0,1}} & 0 & 0 \\ 0 & C_{x\delta\hat{V}_{d,N-1}A_{0,N}} & 0 & 0 \end{bmatrix}^T \begin{bmatrix} C_{T\,0,N-1} \\ V_{0,1} \\ V_{0,N-1} \\ \delta V_{0,N-1} \\ A_{0,1} \\ A_{0,N} \end{bmatrix}}_{c_{xN-1}}, \\
\hat{P}_{N-1}[k] = &\underbrace{\begin{bmatrix} C_{\hat{P}_{N-1}\hat{V}_{N-1}} & 0 & C_{\hat{P}_{N-1}\tilde{C_T}_{N-1}} \end{bmatrix}}_{C_{N-1}} x_{N-1}[k] \\
&+ \underbrace{\begin{bmatrix} C_{y\hat{P}_{N-1}C_{T\,0,N-1}} & C_{y\hat{P}_{N-1}P_{0,N-1}} & C_{y\hat{P}_{N-1}V_{0,N-1}} \end{bmatrix} \begin{bmatrix} C_{T\,0,N-1} \\ P_{0,N-1} \\ V_{0,N-1} \end{bmatrix}}_{c_{yN-1}},
\end{aligned}
\tag{A-2}
$$

$$
\begin{aligned}
x_N[k+1] = &\underbrace{\begin{bmatrix} 0 & 0 \\ 0 & A_{\tilde{C_T}_N\tilde{C_T}_N} \end{bmatrix}}_{A_{N,N}} x_N[k] + \underbrace{\begin{bmatrix} 0 & A_{\hat{V}_N\delta\hat{V}_{d,N-1}} & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A_{N,N-1}} x_{N-1}[k] \\
&+ \underbrace{\begin{bmatrix} 0 \\ B_{\tilde{C_T}_N C_{T\,N}} \end{bmatrix}}_{B_N} C_{T\,N}[k] + \underbrace{\begin{bmatrix} C_{x\hat{V}_N V_{0,1}} \\ 0 \end{bmatrix} \begin{bmatrix} V_{0,1} \end{bmatrix}}_{c_{xN}}, \\
\hat{P}_N[k] = &\underbrace{\begin{bmatrix} C_{\hat{P}_N\hat{V}_N} & C_{\hat{P}_N\tilde{C_T}_N} \end{bmatrix}}_{C_N} x_N[k] + \underbrace{\begin{bmatrix} C_{y\hat{P}_N C_{T\,0,N}} & C_{y\hat{P}_N P_{0,N}} & C_{y\hat{P}_N V_{0,N}} \end{bmatrix} \begin{bmatrix} C_{T\,0,N} \\ P_{0,N} \\ V_{0,N} \end{bmatrix}}_{c_{yN}}
\end{aligned}
\tag{A-3}
$$

and $\forall\{n \in \mathbb{Z} | 2 \leq n \leq N-2\}$

$$
\begin{aligned}
x_n[k+1] = & \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ A_{\delta\hat{V}_{d,n}\hat{V}_n} & A_{\delta\hat{V}_{d,n}\delta\hat{V}_{d,n}} & 0 & A_{\delta\hat{V}_{d,n}\tilde{C}_{Tn}} \\ 0 & 0 & A_{\tilde{C}_{Td,n}\tilde{C}_{Td,n}} & A_{\tilde{C}_{Td,n}\tilde{C}_{Tn}} \\ 0 & 0 & 0 & A_{\tilde{C}_{Tn}\tilde{C}_{Tn}} \end{bmatrix}}_{A_{n,n}} x_n[k] \\
& + \underbrace{\begin{bmatrix} 0 & A_{\hat{V}_n\delta\hat{V}_{d,n-1}} & 0 & 0 \\ 0 & A_{\delta\hat{V}_{d,n}\delta\hat{V}_{d,n-1}} & A_{\delta\hat{V}_{d,n}\tilde{C}_{Td,n-1}} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A_{n,n-1}} x_{n-1}[k] \\
& + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ B_{\tilde{C}_{Tn}C_{Tn}} \end{bmatrix}}_{B_n} C_{Tn}[k] \\
& + \underbrace{\begin{bmatrix} 0 & C_{x\delta\hat{V}_{d,n}C_{T0,n}} & 0 & 0 \\ C_{x\hat{V}_nV_{0,1}} & 0 & 0 & 0 \\ 0 & C_{x\delta\hat{V}_{d,n}V_{0,n}} & 0 & 0 \\ 0 & C_{x\delta\hat{V}_{d,n}\delta V_{0,n}} & 0 & 0 \\ 0 & C_{x\delta\hat{V}_{d,n}A_{0,1}} & 0 & 0 \\ 0 & C_{x\delta\hat{V}_{d,n}A_{0,n+1}} & 0 & 0 \end{bmatrix}^T \begin{bmatrix} C_{T0,n} \\ V_{0,1} \\ V_{0,n} \\ \delta V_{0,n} \\ A_{0,1} \\ A_{0,n+1} \end{bmatrix}}_{c_{xn}}, \\
\hat{P}_n[k] = & \underbrace{\begin{bmatrix} C_{\hat{P}_n\hat{V}_n} & 0 & 0 & C_{\hat{P}_n\tilde{C}_{Tn}} \end{bmatrix}}_{C_n} x_n[k] \\
& + \underbrace{\begin{bmatrix} C_{y\hat{P}_nC_{T0,n}} & C_{y\hat{P}_nP_{0,n}} & C_{y\hat{P}_nV_{0,n}} \end{bmatrix} \begin{bmatrix} C_{T0,n} \\ P_{0,n} \\ V_{0,n} \end{bmatrix}}_{c_{yn}},
\end{aligned}
\tag{A-4}
$$

where

$$
A_{\delta\hat{V}_{d,n}\hat{V}_n} = \begin{bmatrix} 0 \\ \frac{\partial\delta V_n}{\partial V_n}\Big|_{x_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{d_{n,n+1}\times 1} \quad \{n \in \mathbb{Z} | 2 \leq n \leq N-1\},
$$

$$
A_{\delta\hat{V}_{d,1}\delta\hat{V}_{d,1}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(d_{1,2}-1)\times(d_{1,2}-1)},
$$

$$A_{\delta\hat{V}_{d,n}\delta\hat{V}_{d,n}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(d_{n,n+1})\times(d_{n,n+1})} \quad \forall\{n \in \mathbb{Z} | 2 \leq n \leq N-1\},$$

$$A_{\delta\hat{V}_{d,1}\tilde{C_T}_1} = \begin{bmatrix} \frac{\partial \delta V_1}{\partial \tilde{C_T}_1}\Big|_{x_0} + \frac{\partial \delta V_1}{\partial A_2}\Big|_{x_0} \frac{1}{2} A_R \frac{c_w}{1-c_w} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{(d_{1,2}-1)\times 1},$$

$$A_{\delta\hat{V}_{d,n}\tilde{C_T}_n} = \begin{bmatrix} 0 \\ \frac{\partial \delta V_n}{\partial \tilde{C_T}_n}\Big|_{x_0} + \frac{\partial \delta V_n}{\partial A_{n+1}}\Big|_{x_0} \frac{1}{2} A_R \frac{c_w}{1-c_w} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{(d_{n,n+1})\times 1} \quad \forall\{n \in \mathbb{Z} | 2 \leq n \leq N-1\},$$

$$A_{\tilde{C_T}_{d,1}\tilde{C_T}_{d,1}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(d_{1,2}-1)\times(d_{1,2}-1)},$$

$$A_{\tilde{C_T}_{d,n}\tilde{C_T}_{d,n}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(d_{n,n+1})\times(d_{n,n+1})} \quad \forall\{n \in \mathbb{Z} | 2 \leq n \leq N-2\},$$

$$A_{\tilde{C_T}_{d,1}\tilde{C_T}_1} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{(d_{1,2}-1)\times 1},$$

$$A_{\tilde{C_T}_{d,n}\tilde{C_T}_n} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{d_{n,n+1}\times 1} \quad \forall\{n \in \mathbb{Z} | 2 \leq n \leq N-2\},$$

$$A_{\tilde{C_T}_n\tilde{C_T}_n} = e^{-\frac{1}{\tau}h} \quad \forall\{n \in \mathbb{Z} | 1 \leq n \leq N\},$$

$$A_{\hat{V}_2\delta\hat{V}_{d,1}} = \begin{bmatrix} 0 & \cdots & 0 & -1 \end{bmatrix} \in \mathbb{R}^{1\times(d_{1,2}-1)},$$

$$A_{\hat{V}_n\delta\hat{V}_{d,n-1}} = \begin{bmatrix} 0 & \cdots & 0 & -1 \end{bmatrix} \in \mathbb{R}^{1\times d_{n-1,n}} \quad \forall\{n \in \mathbb{Z} | 3 \leq n \leq N\},$$

$$A_{\delta\hat{V}_2\delta\hat{V}_{d,1}} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{d_{2,3}\times(d_{1,2}-1)},$$

$$A_{\delta\hat{V}_n\delta\hat{V}_{d,n-1}} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{d_{n,n+1}\times d_{n-1,n}} \quad \forall\{n \in \mathbb{Z} | 3 \leq n \leq N-1\},$$

$$A_{\delta\hat{V}_{d,2}\tilde{C_T}_{d,1}} = \begin{bmatrix} 0 & \cdots & 0 & \frac{\partial\delta V_2}{\partial A_3}\Big|_{x_0}\frac{1}{2}A_R\frac{c_w}{1-c_w} \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{d_{2,3}\times(d_{1,2}-1)},$$

$$A_{\delta\hat{V}_{d,n}\tilde{C_T}_{d,n-1}} = \begin{bmatrix} 0 & \cdots & 0 & \frac{\partial\delta V_n}{\partial A_{n+1}}\Big|_{x_0}\frac{1}{2}A_R\frac{c_w}{1-c_w} \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{d_{n,n+1}\times d_{n-1,n}} \quad \forall\{n \in \mathbb{Z} | 3 \leq n \leq N-1\},$$

$$B_{\tilde{C_T}_n C_{T_n}} = 1 - e^{-\frac{1}{\tau}h} \quad \forall\{n \in \mathbb{Z} | 1 \leq n \leq N\}$$

$$C_{x\delta\hat{V}_{d,1}C_{T0,1}} = \begin{bmatrix} -\frac{\partial\delta V_1}{\partial C_{T1}}\Big|_{x_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{(d_{1,2}-1)\times 1},$$

$$C_{x\delta\hat{V}_{d,n}C_{T0,n}} = \begin{bmatrix} 0 \\ -\frac{\partial\delta V_n}{\partial C_{Tn}}\Big|_{x_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{d_{n,n+1}\times 1} \quad \forall\{n \in Z | 2 \leq n \leq N-1\},$$

$$C_{x\hat{V}_n V_{0,1}} = 1 \quad \forall\{n \in \mathbb{Z} | 2 \leq n \leq N\},$$

$$C_{x\delta\hat{V}_{d,n}V_{0,n}} = \begin{bmatrix} 0 \\ -\frac{\partial\delta V_n}{\partial V_n}\Big|_{x_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{d_{n,n+1}\times 1} \quad \forall\{n \in Z | 2 \leq n \leq N-1\},$$

$$C_{x\delta\hat{V}_{d,1}\delta V_{0,1}} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{(d_{1,2}-1)\times 1},$$

$$C_{x\delta\hat{V}_{d,n}\delta V_{0,n}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{d_{n,n+1}\times 1} \quad \forall\{n \in Z | 2 \leq n \leq N-1\},$$

$$C_{x\delta\hat{V}_{d,1}A_{0,1}} = \begin{bmatrix} \frac{\partial \delta V_1}{\partial A_2}\Big|_{x_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{(d_{1,2}-1)\times 1},$$

$$C_{x\delta\hat{V}_{d,n}A_{0,1}} = \begin{bmatrix} 0 \\ \frac{\partial \delta V_n}{\partial A_{n+1}}\Big|_{x_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{d_{n,n+1}\times 1} \quad \forall\{n \in Z | 2 \leq n \leq N-1\},$$

$$C_{x\delta\hat{V}_{d,1}A_{0,1}} = \begin{bmatrix} -\frac{\partial \delta V_1}{\partial A_2}\Big|_{x_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{(d_{1,2}-1)\times 1},$$

$$C_{x\delta\hat{V}_{d,n}A_{0,1}} = \begin{bmatrix} 0 \\ -\frac{\partial \delta V_n}{\partial A_{n+1}}\Big|_{x_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{d_{n,n+1}\times 1} \quad \forall\{n \in Z | 2 \leq n \leq N-1\},$$

$$C_{\hat{P}_n\hat{V}_n} = \frac{\partial \delta P_n}{\partial V_n}\Big|_{x_0} \quad \forall\{n \in \mathbb{Z} | 2 \leq n \leq N\},$$

$$C_{\hat{P}_n\tilde{C_{T}}_n} = \frac{\partial \delta P_n}{\partial C_{T}{}_n}\Big|_{x_0} \quad \forall\{n \in \mathbb{Z} | 1 \leq n \leq N\},$$

$$C_{y\hat{P}_nC_{T0,n}} = -\frac{\partial \delta P_n}{\partial C_{T}{}_n}\Big|_{x_0} \quad \forall\{n \in \mathbb{Z} | 1 \leq n \leq N\},$$

$$C_{y\hat{P}_nP_{0,n}} = 1 \quad \forall\{n \in \mathbb{Z} | 1 \leq n \leq N\}$$

and

$$C_{y\hat{P}_n V_{0,n}} = -\frac{\partial \delta P_n}{\partial V_n}\Big|_{x_0} \quad \forall\{n \in \mathbb{Z} | 2 \leq n \leq N\}.$$

Appendix B

Validation of model with PALM

In this appendix the proposed controller model is compared with data from Parallelized Large-Eddy Simulation Model (PALM). PALM is a high-fidelity three-dimensional Large Eddy Simulation (LES) model. For the comparison a wind farm with three turbines in a row parallel to the wind speed is used. A visual representation of this farm can be found in figure B-1. The diameter of the rotors is 90 meters and the spacing between the wind turbines is 540 meters. Laminar inflow conditions with a free stream wind speed of 8m/s are considered. The sample time $h = 1$ seconds. The chosen decision variables for the proposed model can be found in table B-1.

In figures B-2, B-3 and B-4 power outputs of the turbines predicted by the proposed model compared to PALM for three different reference signals can be seen. From these figures it can be concluded that the proposed model is able to also properly track the mean behaviour of the power output compared to PALM. In figure B-5 the total power output of the wind farm for simulation cases 1, 2 and 3 is shown. Also from this figure it can be concluded that the mean behaviour of the power output is tracked properly. Table B-2 gives the root mean square error (RMSE) for the different simulation cases.

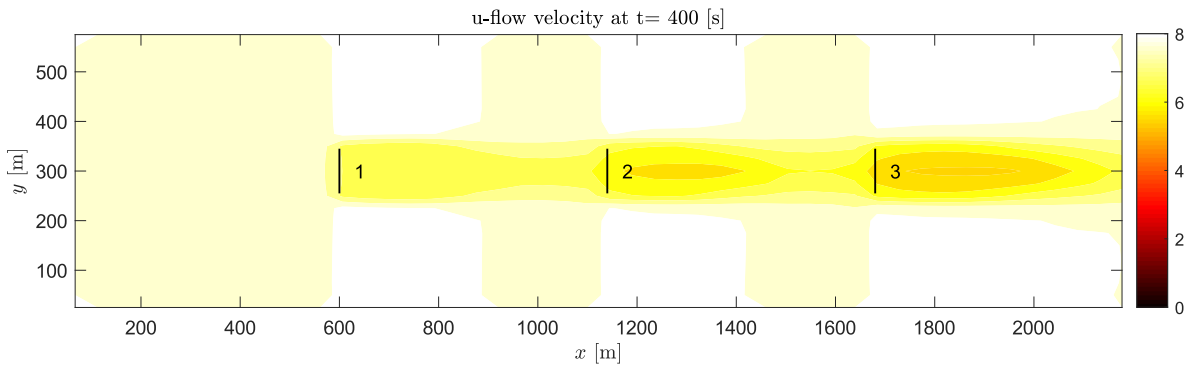

Figure B-1: Layout of wind farm used for comparison with PALM data.

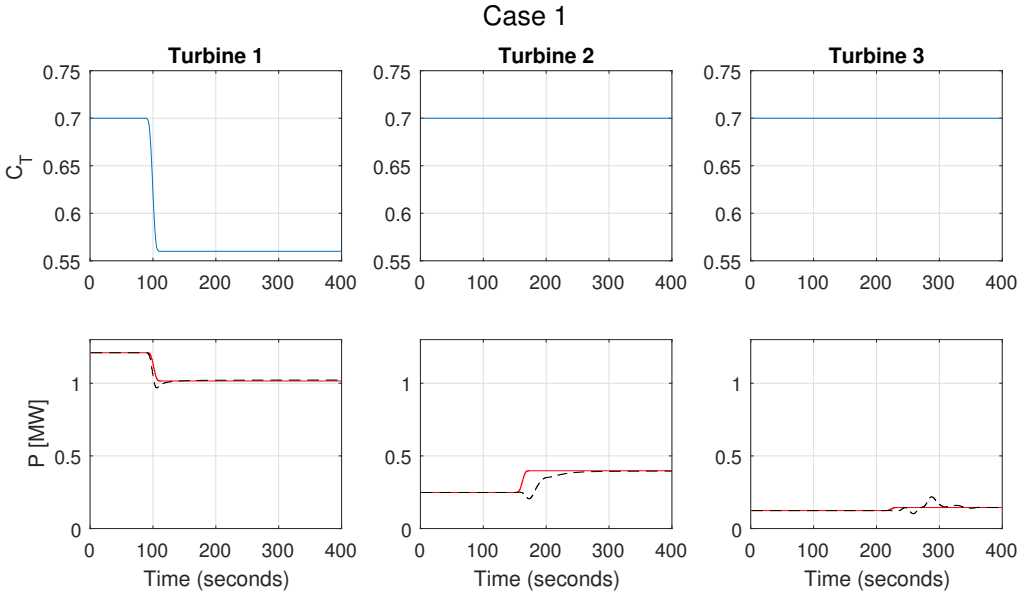

Figure B-2: Comparison of power outputs predicted by the proposed model (red) compared to PALM (black dotted) with a step change in the thrust coefficient of turbine number 1.

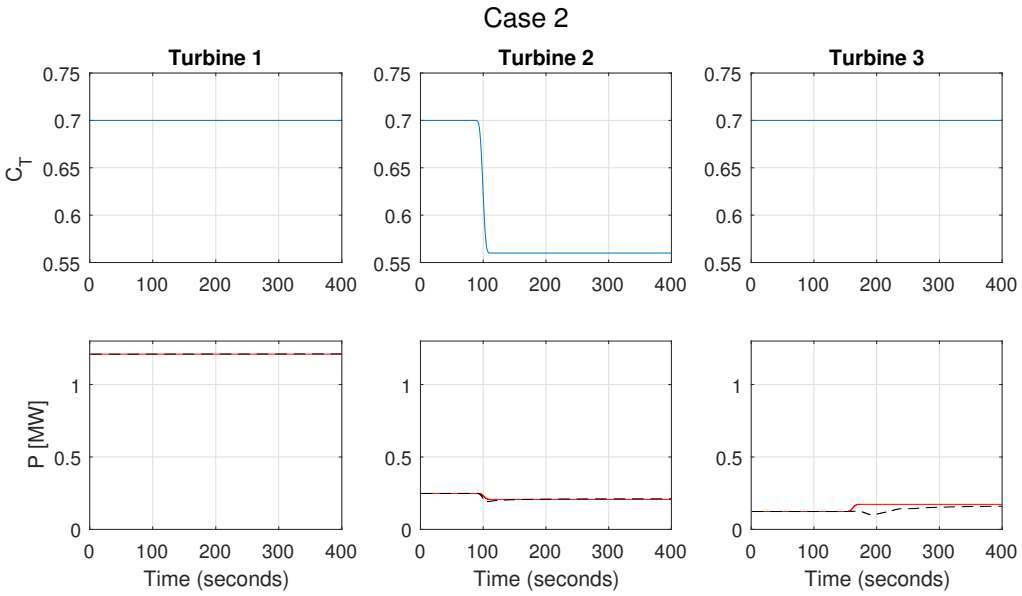

Figure B-3: Comparison of power outputs predicted by the proposed model (red) compared to PALM (black dotted) with a step change in the thrust coefficient of turbine number 2.

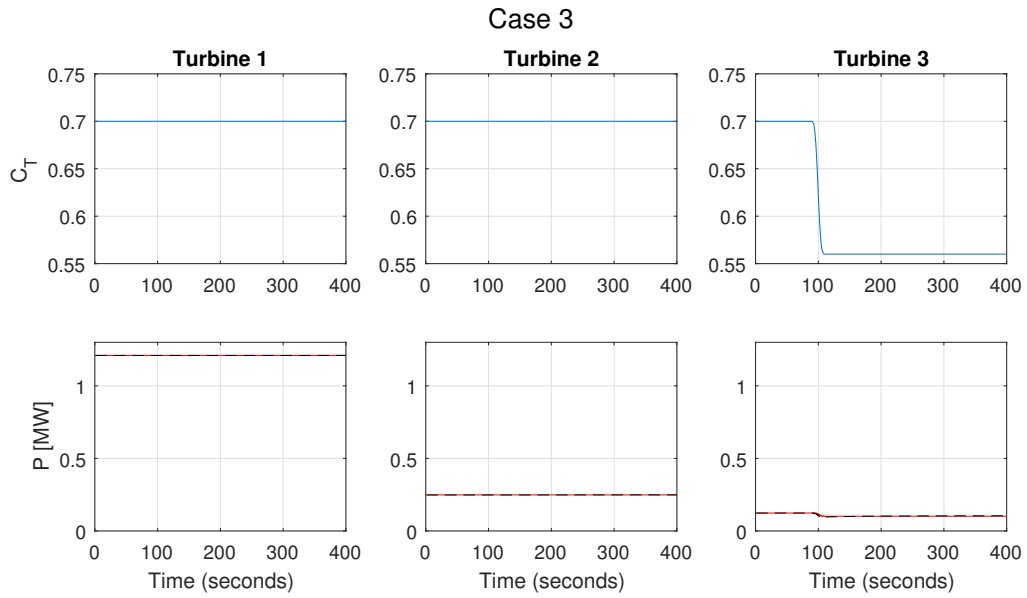

Figure B-4: Comparison of power outputs predicted by the proposed model (red) compared to PALM (black dotted) with a step change in the thrust coefficient of turbine number 3.

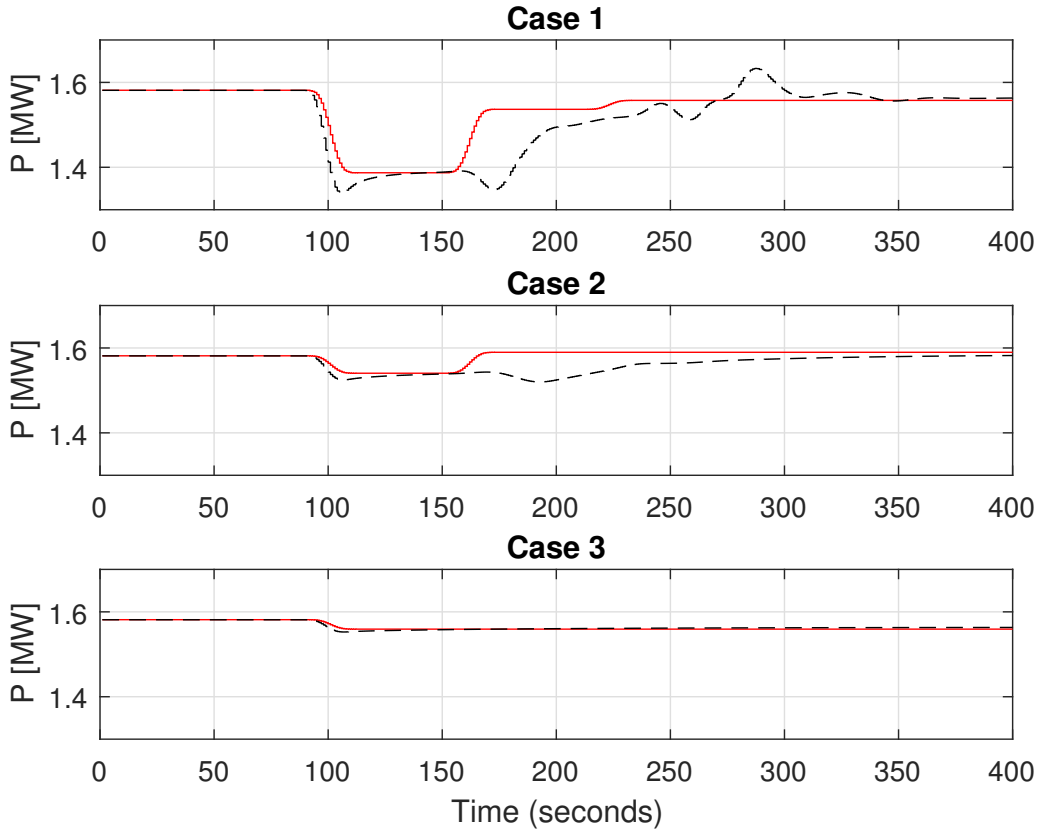

Figure B-5: Comparison of total power output predicted by the proposed model (red) compared to PALM (black dotted) for cases 1, 2 and 3.

Table B-1: Chosen decision variables for proposed model

Variable	Value
c_w	0.15
c_{VV}	2.1
c_{VC_T}	1.52
c_{VA}	1
c_{PV}	1.2
c_{PC_T}	0.8
τ	5

Table B-2: RMSE for different simulation cases.

Case	RMSE [MW]
1	0.046
2	0.025
3	0.003

Bibliography

[1] 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, pp. 75–95, 03 2018.

[2] C. Steiness, *Horns Rev wind farm, DK*. Vattenfall, Feb 2008.

[3] Eurostat, "Shares 2016 results," 2016. Data retrieved from Eurostat, http://ec.europa.eu/eurostat/documents/38154/4956088/SHARES-2016-SUMMARY-RESULTS.xlsx.

[4] NERC Resources Subcommittee, "Balancing and frequency control," tech. rep., North American Electric Reliability Corporation, 2011.

[5] E. Ela, V. Gevorgian, P. Fleming, Y. C. Zhang, M. Singh, E. Muljadi, A. Scholbrook, J. Aho, A. Buckspan, L. Pao, V. Singhvi, A. Tuohy, P. Pourbeik, D. Brooks, and N. Bhatt, "Active power controls from wind power: Bridging the gaps," tech. rep., National Renewable Energy Laboratory (NREL), 2014.

[6] 4C Offshore Ltd, "Global offshore wind farms database," 2017. Data retrieved from Global Offshore Wind Farms Database, http://www.4coffshore.com/windfarms/.

[7] A. D. Hansen, P. Sørensen, F. Iov, and F. Blaabjerg, "Centralised power control of wind farm with doubly fed induction generators," *Renewable Energy*, vol. 31, no. 7, pp. 935–951, 2006.

[8] H. Badihi, Y. Zhang, and H. Hong, "Active power control design for supporting grid frequency regulation in wind farms," *Annual Reviews in Control*, vol. 40, pp. 70–81, 2015.

[9] J.-W. van Wingerden, L. Pao, J. Aho, and P. Fleming, "Active power control of waked wind farms," *IFAC-PapersOnLine*, vol. 50, pp. 4484–4491, 2017.

[10] S. Boersma, V. Rostampour, B. Doekemeijer, W. van Geest, and J.-W. van Wingerden, "A constrained model predictive wind farm controller providing active power control: an les study," *Journal of Physics: Conference Series*, vol. 1037, no. 3, 2018.

[11] P. M. O. Gebraad, F. W. Teeuwisse, J.-W. van Wingerden, P. A. Fleming, S. Ruben, J. R. Marden, and L. Y. Pao, "A data-driven model for wind plant power optimization by yaw control," in *American Control Conference (ACC)*, (Portland, Oregon, USA), June 2014.

[12] S. Siniscalchi-Minna, F. D. Bianchi, and C. Ocampo-Martinez, "Predictive control of wind farms based on lexicographic minimizers for power reserve maximization," in *American Control Conference (ACC)*, (Wisconsin Center, Milwaukee, USA), June 2018.

[13] 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, pp. 1261–1275, 2017.

[14] C. R. Shapiro, J. Meyers, C. Meneveau, and D. F. Gayme, "Coordinated pitch and torque control of wind farms for power tracking," in *Annual American Control Conference (ACC)*, pp. 688–694, June 2018.

[15] T. Knudsen, T. Bak, and M. Svenstrup, "Survey of wind farm control-power and fatigue optimization," *Wind Energy*, vol. 18, pp. 1333–1351, 2015.

[16] S. Li and Y. Zheng, *Distributed Model Predictive Control for Plant-Wide Systems*. John Wiley & Sons Singapore Pte. Ltd., 2015.

[17] V. Spudić, C. Conte, M.Baotić, and M. Morari, "Cooperative distributed model predictive control for wind farms," *Optimal Control Algorithms and Methods*, vol. 36, pp. 333–352, 2015.

[18] T. Johansen, "Introduction to nonlinear model predictive control and moving horizon estimation," in *Selected Topics on Contrained and Nonlinear Control* (M. Huba, S. Skogestad, M. Fikar, M. Hovd, T. Johansen, and B. Róhaĺ-Ilkiv́, eds.), ch. 5, pp. 187–239, STU Bratislave - NTNU Trondheim, 2011.

[19] R. Mikkelsen, *Actuator Disc Methods Applied to Wind Turbines*. PhD thesis, Technical University of Denmark, 2003.

[20] L. A. Martínez, S. Leonardi, M. J. Churchfield, and P. L. Moriarty, "A comparison of actuator disk and actuator line wind turbine models and best practices for their use," in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, January 2012.

[21] N. O. Jensen, "A note on wind generator interaction," tech. rep., 1983.

[22] 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, 2010.

[23] P. M. O. Gebraad and J.-W. van Wingerden, "A control-oriented dynamic model for wakes in wind plants," *Journal of Physics: Conference Series*, vol. 524, 2014.

[24] M. Soleimanzadeh, R. Wisniewski, and A. Brand, "State-space representation of the wind flow model in wind farms," *Wind Energy*, vol. 17, 2014.

[25] S. Frandsen, R. Barthelmie, S. Pryor, O. Rathmann, S. Larsen, J. Højstrup, and M. Thøgersen, "Analytical modelling of wind speed deficit in large wind farms," vol. 9, pp. 39–53, 04 2006.

[26] G. Taylor, "The Spectrum of Turbulence," *Proceedings of the Royal Society of London Series A*, vol. 164, pp. 476–490, 02 1938.

[27] S. Boersma, B. M. Doekemeijer, T. Keviczky, and J. van Wingerden, "Stochastic model predictive control: uncertainty impact on wind farm power tracking," in *2019 American Control Conference*, pp. 4167–4172, 2019.

[28] W. Munters and J. Meyers, "An optimal control framework for dynamic induction control of wind farms and their interaction with the atmospheric boundary layer," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 375, no. 2091, 2017.

[29] B. Maronga, M. Gryschka, R. Heinze, F. Hoffmann, F. Kanani-Sühring, M. Keck, K. Ketelsen, M. O. Letzel, M. Sühring, and S. Raasch, "The parallelized large-eddy simulation model (palm) version 4.0 for atmospheric and oceanic flows: model formulation, recent developments, and future perspectives," *Geoscientific Model Development*, vol. 8, 2015.

[30] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. Springer-Verlag London Limited, 2009.

[31] M. A. Stephens, C. Manzie, and M. C. Good, "Model predictive control for reference tracking on an industrial machine tool servo drive," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 808–816, May 2013.

[32] D. A. Smith and M. Harris, "Wind turbine control having a lidar wind speed measurement apparatus," Oct. 16 2007. US Patent 7,281,891.

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

[34] L. Wang, "A tutorial on model predictive control: Using a linear velocity-form model," *Developments in Chemical Engineering and Mineral Processing*, vol. 12, pp. 573–614, 05 2008.

[35] A. Betz, *Introduction to the theory of flow machines*. Pergamon Press, 1966.

[36] A. Venkat, J. Rawlings, and S. Wright, "Distributed model predictive control of large-scale systems," in *Assessment and Future Directions of Nonlinear Model Predictive Control* (R. Findeisen, F. Allgöwer, and L. Biegler, eds.), pp. 591–605, Berlin, Heidelberg: Springer, 2007.

[37] PJM, "PJM manual 12: Balancing operations," tech. rep., PJM, 2018.

[38] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, (Taipei, Taiwan), 2004.

[39] V. G. D. Jacob M., Neves C., "Short term load forecasting," in *Forecasting and Assessing Risk of Individual Electricity Peaks*, ch. 2, pp. 15–37, Springer, Cham, 2020.

[40] 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, 2018.

[41] W. Munters and J. Meyers, "Towards practical dynamic induction control of wind farms: analysis of optimally controlled wind-farm boundary layers and sinusoidal induction control of first-row turbines," *Wind Energy Science*, vol. 3, no. 1, pp. 409–425, 2018.

[42] J. Frederik, R. Weber, S. Cacciola, F. Campagnolo, A. Croce, C. Bottasso, and J.-W. van Wingerden, "Periodic dynamic induction control of wind farms: proving the potential in simulations and wind tunnel experiments," *Wind Energy Science Discussions*, vol. 2019, pp. 1–18, 2019.

Glossary

List of Acronyms

ADM	Actuator Disk Model
MPC	model predictive control
DMPC	distributed model predictive control
WFSim	WindFarmSimulator
PALM	Parallelized Large-Eddy Simulation Model
LES	Large Eddy Simulation
RMSE	root mean square error
MPC	model predictive control
APC	active power control
LIDAR	LIght Detection And Ranging of Laser Imaging Detection And Ranging
NERC	North American Electric Reliability Corporation
APC	active power control
PI	proportional-integral
FLORIS	FLOw Redirection and Induction in Steady-state
FLORIDyn	FLOw Redirection and Induction Dynamics
CCDC	coupled cost decoupled constraint
NJDC	normal Jacobian distributed controller
MJDC	modified Jacobian distributed controller