Distributed Model Predictive Control for Wind-Farm Flow Control

A.A. Koelewijn

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Distributed Model Predictive Control for Wind-Farm Flow Control

Master of Science Thesis

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

A.A. Koelewijn

November 2025

Faculty of Mechanical Engineering (ME) · Delft University of Technology

Abstract

Wake interactions in wind farms create strong coupling between turbines. Coordinated, farm-level control is therefore required to meet power-tracking objectives and to reduce losses. Model predictive control (MPC) has been shown to be a promising approach to achieve coordinated control, since it can leverage a model to predict wake interactions over the prediction period. However, existing approaches either rely on low-fidelity models with steady-state simplifications that fail to capture transient wake dynamics, or on high-fidelity dynamic flow models that can be computationally prohibitive. This thesis addresses the gap between predictive control performance and real-time optimisation in MPC from a model perspective, by investigating model approximations; and from a control perspective, by designing distributed MPC approaches.

Core to the proposed approach is the use of FLORIDyn as a prediction model, a dynamic wind-farm flow model that captures nonlinear wake propagation and turbine interactions, offering a middle ground between low-fidelity steady-state models and computationally intensive high-fidelity models. Different levels of approximation within the MPC prediction model are investigated. A centralised MPC controller is developed as a baseline and, motivated by scalability of the approach, two distributed MPC approaches are designed: a full-farm distributed model predictive control (DMPC) scheme that retains farm-wide dynamics in each subproblem, and a subset DMPC architecture that restricts explicit wake modelling to locally interacting turbines, both using the nonlinear prediction model.

Results indicate that prediction-model approximations can speed up the optimisation, at the cost of introducing significant tracking error, highlighting a trade-off between computational effort and control performance. Both the centralised and distributed MPC controllers reduce tracking error significantly compared with the state-of-the-art mixed-integer quadratically-constrained quadratic program (MIQCQP) approach. The subset DMPC architecture provides the best overall trade-off compared to the other proposed approaches, combining high tracking performance with fast and predictable computation times as farm size increases. The subset-based DMPC approach is therefore a promising candidate for high-performance real-time farm-level control.

Table of Contents

Acknowledgements

First, I would like to thank my supervisor, Bart de Schutter, for his guidance and for providing helpful insights that steered the direction of this research.

I am especially grateful to my daily supervisor, Sam Mallick. Thank you for always being willing to help and for consistently making time for me. Our insightful discussions made this process much more enjoyable, and I am thankful for the support and good spirit you brought to our collaboration.

I would also like to extend my gratitude to my family and friends for their unconditional support and encouragement during this thesis period.

Finally, this thesis marks the conclusion of my time at TU Delft. Looking back, I am grateful for the experiences, knowledge, and opportunities this university has given me over the years.

Delft, University of Technology
November 2025

A.A. Koelewijn

Chapter 1

Introduction

1-1 Wind farm control

The share of wind energy in the global electricity production has increased rapidly over the past decades, driven by the need for sustainable and low-carbon energy sources [31]. As the share of wind energy increases, the wind-farm sector is evolving to match the growing demand, with developments in turbine technology and farm layout design improving potential energy output [31]. Most notably, the overall size and scale of wind farms is increasing as large-scale wind farms offer increased energy supply with efficient use of space, benefiting from reduced costs due to a common infrastructure [19].

This growth in the scale of wind farms leads to a new technical challenge, managing the wake effect [33, 34]. When a wind turbine extracts energy from the wind, it creates a region of reduced wind speed and increased turbulence downstream, known as the wake. As shown in Figure 1-1, turbines operating in the wake of upstream turbines are exposed to slower, more turbulent air, which leads to lower power output and increased mechanical stress. As a consequence, the total power production of a wind farm is often significantly lower than that which is theoretically possible based on the free wind speeds. This reduction in efficiency, together with increased mechanical stress, can lead to higher maintenance demands and reduced economic benefit of the wind farm [4, 34].

Traditionally, wind turbine control strategies have focused on optimising the performance of individual turbines [2]. However, as wind farms have expanded and become more densely packed, the turbulent wakes from one turbine increasingly interact with others downstream. Consequently, the operation of a single turbine directly affects the quality and speed of the wind reaching its neighbours, making farm-level control a critical field of research. To achieve high performance control, a wind farm must therefore be considered a single, strongly coupled dynamical system rather than a collection of independent units [2, 34]. Coordinating the operation of all turbines becomes essential to mitigate these wake-induced energy losses and to optimise the total power output of the entire farm. Indeed, wind farm flow control is regarded as one of the grand challenges in wind energy [47].

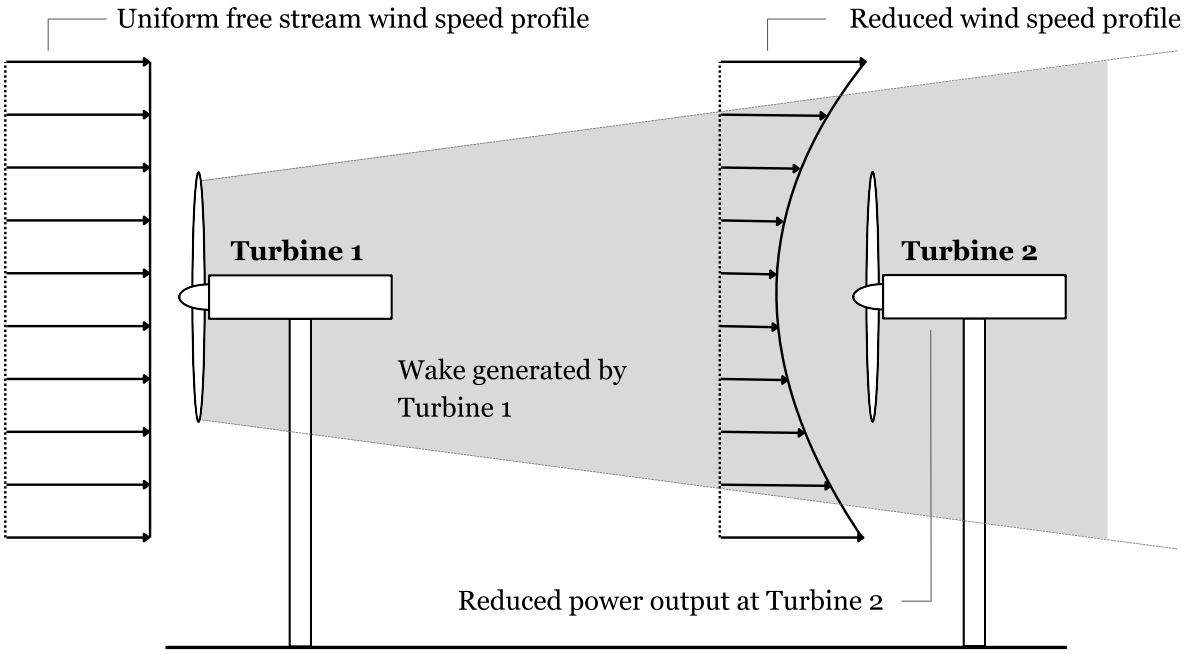

Figure 1-1: Schematic representation of the wake effect between two wind turbines, aligned with the free-stream wind. Adapted from [31].

Model predictive control (MPC) [38] has emerged as a promising strategy to address the challenges of the wake effect within wind-farm flow control. MPC is an optimisation-based approach that predicts the system's future behaviour over a finite time horizon using a prediction model, computing a sequence of optimal control inputs, of which only the first is implemented before the process repeats at the next time step. Its suitability for wind-farm flow control comes from the ability to capture wake interactions between turbines via a wind-farm flow model. These interactions are accounted for during prediction to optimise the control inputs and improve overall power output. MPC also treats operational constraints explicitly, which is another advantage in this application. From an optimisation standpoint, MPC requires solving a problem in real-time at each step. This creates a trade-off: the prediction model must be accurate enough to represent the relevant dynamics yet computationally tractable so the optimisation can be solved within the available time. Wind-farm flow dynamics are highly nonlinear and tightly coupled, featuring wake propagation, wake recovery, and cumulative wake effects that are difficult to capture with simple models [4, 31]. The wind-farm flow model used within the MPC framework must therefore be sufficiently accurate to make reliable predictions; oversimplified linear or static models are often not sufficient. Consequently, for large-scale wind farms the resulting MPC formulation becomes a high-dimensional nonlinear optimisation problem whose computational cost grows rapidly with the number of turbines.

Two possible solutions to this tension between model fidelity and computational tractability are distributed model predictive control (DMPC) and model-simplification strategies. In DMPC framework, the global optimisation problem is decomposed into smaller subproblems solved locally by individual turbines or groups of turbines, with coordination between turbines or subgroups to achieve cooperative performance. This decomposition lowers problem com-

plexity and can shorten computation time, though typically at the cost of reduced (global) performance. Model simplifications can further reduce computational demand, but they introduce approximation error that may reduce control performance. The key consideration in MPC for wind-farm flow control is the trade-off between computational tractability and control performance. This thesis therefore investigates the following research question:

For a model predictive control framework using a dynamic engineering wind-farm flow model, how can model simplifications and distributed control be used to balance computational tractability and control performance for large-scale wind farms?

1-2 Contributions

The primary scientific contributions of this thesis are summarised as follows:

1. The *first use* of FLORIDyn as a prediction model in closed-loop MPC, showing that a dynamic engineering model can explicitly capture wake effects within MPC.
2. The design of a distributed MPC scheme leveraging FLORIDyn as prediction model, yielding a favourable trade-off between tracking performance and computation time compared with a centralised controller.
3. An extension of the distributed MPC framework through a subset DMPC architecture, designed to significantly enhance its scalability for large-scale wind farms.
4. An analysis of the trade-off between computation time and tracking performance for a range of prediction-model approximations in closed-loop MPC.

1-3 Organisation of the report

The rest of this report is structured as follows. Chapter 2 consists of the background and related literature on wind-farm flow control and describes the wind-farm flow model used in this thesis. Chapter 3 presents the theoretical framework, covering prediction models, and the nonlinear centralised and distributed MPC formulations. Chapter 4 reports the case study results, examines model approximations and the performance of the centralised model predictive control (CMPC) and DMPC approaches, and benchmarks these against the state-of-the-art approach. Finally, Chapter 5 summarises the main findings and discusses limitations and recommendations for future work.

Chapter 2

Background and relevant literature

This chapter provides background on wind-farm flow control and reviews relevant literature for the approaches proposed in this work. The first section gives an introduction to the field of wind-farm flow control, followed by a section on wind-farm flow models, including the theoretical details of the model used in this research. The third section discusses relevant work on model predictive control for wind-farm flow control, followed by a brief summary of this chapter.

2-1 Introduction to wind-farm flow control

In conventional wind farm control, wind turbines are typically operated on an individual basis, each aiming to maximise its own local power output. This 'greedy' control strategy can be effective at the turbine level, but in the wind farm case, where multiple turbines operate in close proximity, the approach neglects the wake interactions that occur between turbines. As wind farms increase in size and density, the cumulative wake effects become more severe. This can lead to substantial power losses and higher structural loads. Therefore, to effectively manage these interactions, a wind farm must be considered a single, strongly coupled system rather than a collection of individual units. To address these challenges, control strategies that operate at the wind farm level have been developed [2, 34]. The primary aim of wind farm control is to improve the collective performance of the farm by explicitly accounting for wake effects when making control strategies.

There are multiple strategies to reduce wake effects within wind farms, including axial induction control, wake steering, strategic turbine positioning and wake mixing [2, 31, 34]. In this work, we only consider control of existing wind farms and refrain from discussing other strategies than axial induction control and wake steering. Axial induction control acts on the axial induction factor, which describes how much the rotor slows the wind as it passes through. A higher axial induction factor means the turbine takes more momentum from the flow and experiences greater thrust, whereas a lower axial induction factor reduces the turbine's resistance, lowers thrust, and weakens the wake it generates. Figure 2-1 illustrates this principle:

the upstream turbine is de-rated to reduce the thrust force on the flow, which lessens the wake deficit and raises the mean inflow speed at the downstream turbine. Although this reduces power at the front of the farm, the milder wakes can improve conditions for downstream turbines and can lead to a net gain in total farm production. Complementary to axial induction, which adjusts the magnitude of thrust, wake steering alters its direction. In this strategy the upstream turbine is deliberately yawed away from the wind by adjusting its yaw angle, that is, rotating the direction the turbine faces, which redirects the wake laterally and causes it to bend downstream. Also called yaw misalignment, wake redirection, or wake deflection, the aim is to steer the wake away from downstream turbines. Figure 2-2 illustrates the concept: turbine 1 operates with a yaw offset, the wake is deflected, and turbine 2 is exposed to faster, less disturbed inflow. As with axial induction, this reduces power at the front of the farm, as the turbine is no longer facing directly into the wind and thus captures less energy. However, the improved inflow for downstream turbines can raise total farm output.

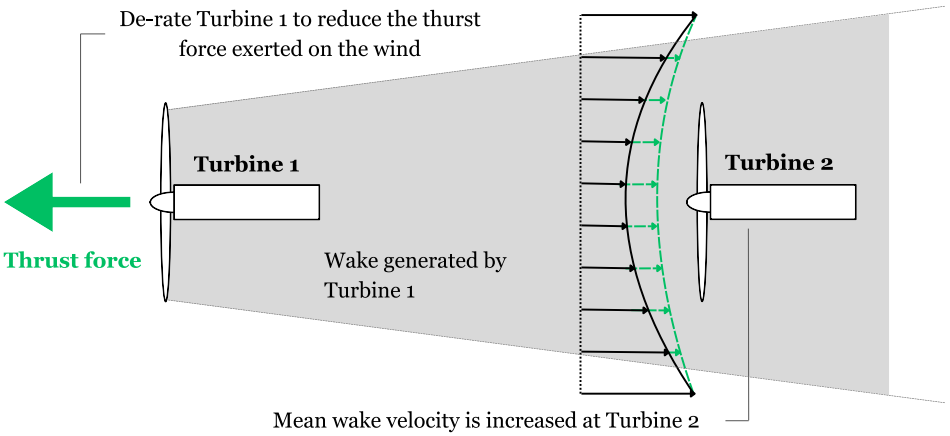

Figure 2-1: Schematic illustration of axial induction control as a wake-mitigation strategy. The upstream turbine (Turbine 1) is de-rated to reduce the thrust force it exerts on the wind, thereby lowering the energy extracted and weakening the generated wake. As a result, the mean wake velocity at the downstream turbine (Turbine 2) increases, improving its inflow conditions and potentially the total power output of the farm. Adapted from [31].

An important consideration in wind-farm flow control is the control objective. Typical control objectives are power maximisation, (power) reference tracking, and mechanical load or fatigue mitigation [2, 34]. Power maximisation is typically applied when wind conditions allow turbines to operate below their maximum capacity. In this regime, control actions such as wake steering or axial induction control are used to redistribute flow and increase the total energy extracted by the farm, often by benefiting downstream turbines at the expense of slight curtailment upstream [3, 20, 21, 23, 35, 37]. Power reference tracking becomes relevant in operational settings where the wind farm is connected to the electrical grid and must follow a schedule determined by the transmission system operator (TSO). In such cases, the farm is not tasked with producing the maximum possible power, but with delivering a specific power level that may vary over time depending on market participation, ancillary service provision, or grid stability requirements [40, 44, 46]. Achieving this requires coordination among turbines to dynamically allocate generation while maintaining overall reliability. In practice, this is often achieved by de-rating selected turbines, meaning they are operated below their maximum power capacity to create wiggle room for upward adjustments. Alter-

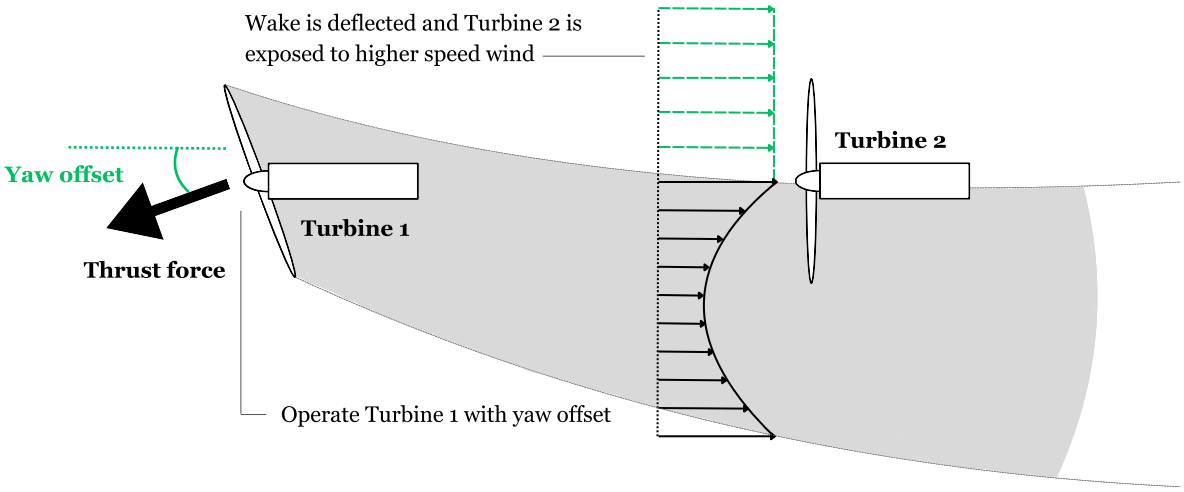

Figure 2-2: Schematic illustration of wake steering as a wake-mitigation strategy. The upstream turbine (Turbine 1) is intentionally yawed away from the wind direction, creating a yaw offset that deflects the wake laterally. This exposes the downstream turbine (Turbine 2) to higher inflow speeds and reduced wake losses. Adapted from [31].

natively, the power output can be redistributed across turbines to balance tracking accuracy and mechanical wear [40, 46]. Load and fatigue mitigation focuses on reducing mechanical stress to extend the operational lifetime of turbines. Wind turbines are subjected to fluctuating loads caused by turbulence, wake interactions, and electrical disturbances. Control strategies that reduce fatigue can increase the lifetime of individual turbines or the entire wind farm, thereby providing an economic incentive by lowering maintenance costs and improving long-term availability. In most wind farm control cases, the control strategy aims either to maximise the total power output of the wind farm, or to track a power reference. Load and fatigue mitigation is less commonly the main objective and more often appears as a secondary consideration or constraint.

Existing control strategies can be broadly grouped according to how they incorporate feedback and whether they employ optimisation. We distinguish three classes: open-loop control, non-optimisation-based feedback control and optimisation-based feedback control. Open-loop control approaches apply predetermined control inputs, which are calculated offline and do not adapt to the current state of the system [3, 20, 21]. Despite being computationally efficient, open-loop approaches tend to underperform in realistic changing flow conditions. Non-optimisation-based feedback control approaches adjust control actions using fixed rules and simple state-feedback controllers based on real-time measurements [37, 46]. These approaches offer simplicity and robustness compared to open-loop control approaches, reacting effectively to disturbances without requiring detailed prediction models. However, they lack explicit coordination and do not optimise flow interactions, leading to suboptimal performance. Optimisation-based feedback control, for example model predictive control (MPC), determines the control input by solving an optimisation problem at each time step, typically using a (prediction) model of the wind farm [23, 35, 40, 44]. This control strategy offers more advanced performance by combining feedback with (flow-aware) prediction models. However, their computational complexity often limits their real-time feasibility. This introduces a gap between high-fidelity, flow-aware MPC formulations that are too computationally de-

manding for practical implementation and simplified approaches that neglect essential wake interactions. Resolving this fundamental tension between prediction fidelity and computational tractability is therefore the primary challenge for effective, real-time wind-farm flow control.

2-2 Wind-farm flow model

In this section we review the existing literature on wind-farm flow models and provide theoretical background on the model leveraged in this work. Wind-farm flow models describe the aerodynamic interactions between wind turbines and their surrounding airflow. These models are distinct from wind-turbine models, which describe the dynamic response of individual turbines to control inputs such as blade pitch and generator torque. One use of wind-farm flow models is as a prediction model within model-based controllers, such as MPC. They can also serve as simulators that emulate the behaviour of real wind farms. Depending on the intended use, a trade-off must be made between computational complexity and accuracy; for example, for simulation purposes accuracy is preferred, whereas for control applications a balance is required to ensure real-time feasibility.

2-2-1 Overview of wind-farm flow models

Wind-farm flow models are typically categorised by their fidelity into low-, medium-, and high-fidelity models. Table 2-1 provides an overview of the specific models discussed in this section. The table categorises each model by its fidelity, time-dependency, and its primary purpose.

Low-fidelity models, often termed engineering models, are the simplest and most computationally efficient wind-farm flow models. They rely on empirical formulas and simplified wake assumptions to capture wake effects and interactions between turbines. Classic examples of engineering models include the Jensen model [28], later extended to the Park model [30]. More recent implementations, developed with a focus on controller design, include FLOw Redirection and Induction in Steady State (FLORIS) [22], used as a prediction model in [6], and toolboxes such as SimWindFarm [24], applied in [18, 40, 42]. These engineering models typically represent wakes as steady shapes and use static superposition rules. While these simplifications make them fast and convenient, they fail to capture the time-varying character of real wake fields and can overlook cumulative and evolving interactions across the farm [3, 7, 20]. To bridge this gap, FLOw Redirection and Induction Dynamics (FLORIDyn) is developed as an extension of these principles [8, 9, 22]. Although it retains the computational efficiency of engineering models, it distinguishes itself by explicitly modelling wake advection and time delays, thereby capturing the dynamic wake propagation that steady-state models neglect.

Medium-fidelity models combine reduced-order physics with empirical corrections to improve accuracy relative to engineering models, while maintaining lower computational costs compared with high-fidelity models. An example is WFSim [11], which solves a two-dimensional form of the Navier–Stokes equations along a horizontal plane. While these models offer improved realism, their practical use in control remains limited due to computational demands and the dominance of faster engineering models in early-stage control design.

High-fidelity models, such as Simulator for Wind Farm Applications (SOWFA) [17], and Sustainable Power-Wind (SP-Wind) [1], are based on computational fluid dynamics (CFD) and solve the Navier–Stokes equations to capture complex flow phenomena including turbulence, flow separation, and wake merging. Implemented in specialised simulation tools, these dynamic models provide a physically realistic environment for testing and validating control strategies [21, 46]. Their high computational cost, however, makes them unsuitable for online optimisation or real-time control [35].

In summary, high-fidelity models capture aerodynamic complexity but are too computationally intensive for real-time use, whereas engineering models are fast but miss important transient dynamics. This trade-off between accuracy and applicability motivates the development of intermediate approaches. FLORIDyn [9], while built upon low-fidelity parameterised wake equations, offers this 'middle ground' solution. It achieves this by integrating a crucial dynamic feature: the model explicitly captures transient wake advection. This combination of a parameterised wake and dynamic transport is best described as a quasi-dynamic representation. It allows the model to capture wake propagation and recovery, unlike fast, steady-state engineering models, without incurring the cost of computationally intensive CFD. This favourable compromise between predictive accuracy and computational tractability makes FLORIDyn a promising candidate for MPC.

Table 2-1: Overview of wind farm flow models used for prediction or simulation purposes, ordered from low to high fidelity. SOWFA and SP-Wind are comparable in terms of their fidelity.

Model	Fidelity	Time-dependency	Main Purpose/Use
Jensen-Park	Low	Steady	Prediction
SimWindFarm	Low	Steady	Simulation
FLORIS	Low	Steady	Prediction / Simulation
FLORIDyn	Low–Medium	Quasi-dynamic	Prediction / Simulation
WFSim	Medium	Quasi-dynamic	Simulation
SOWFA	High	Dynamic	Simulation
SP-Wind	High	Dynamic	Simulation

2-2-2 The FLORIDyn model

This section provides a brief explanation on the FLORIDyn model as described in [9], with minor adaptations as described in [44], since this model is adopted for the control framework in this thesis. FLORIDyn explicitly simulates basic flow dynamics, in contrast to steady engineering models such as FLORIS. Its key idea is to represent the wake field with discrete observation points (OPs) that are created at the rotor plane of every turbine at every time step and then advected downstream. On creation an OP stores the current turbine state (yaw angle γ and axial induction factor a) and, as it moves with the incoming flow, it samples the local wake reduction, taking into account wake interactions. Chains of OPs transport state changes through the farm with a finite delay. By aggregating OPs over the rotor plane the model obtains the rotor-effective wind speed that drives thrust and power.

Gaussian wake

FLORIDyn uses the Gaussian wake description presented in [5] as its local deficit model. This model describes the wake as a deficit, which is strongest at the center and fades towards the edges according to a Gaussian distribution. Figure 2-3 provides a schematic of this wake structure. As wind flows from left to right, the wake propagates downwind, evolving through distinct regions. Close to the rotor, the potential core (blue) and near wake (green) represent the initial formation. Further downstream, the far wake (purple) begins to broaden and mix with the ambient flow, gradually recovering its speed. To formalise this, the model operates in a wake-aligned coordinate frame, $\mathcal{K}_1 = \{x_1, y_1, z_1\}$, where the origin is at the turbine and the x_1-axis is aligned with the local inflow (downwind). In this frame, x_1 denotes the downwind distance, y_1 the lateral (crosswind) distance, and z_1 the vertical distance. The mean streamwise velocity, $v(x_1, y_1, z_1)$, at any point within this frame is then expressed as the free-stream velocity, V_∞, with a deficit;

$$v(x_1, y_1, z_1) \;=\; V_\infty \Big[\, 1 \;-\; \Delta_{x_1} \exp\Big(-\tfrac{1}{2}\tfrac{(y_1-\delta)^2}{\sigma_y^2} - \tfrac{1}{2}\tfrac{z_1^2}{\sigma_z^2} \Big) \Big] \,, \tag{2-1}$$

where V_∞ is the ambient, free-wind speed, which is constant across the farm. The exponential term represents the Gaussian deficit, whose shape is determined by several key variables visible in Figure 2-3. The term δ represents the wake deflection, which is the lateral shift of the wake's centerline caused by turbine yaw γ. The potential-core width and length are $w_{y,\mathrm{pc}}$ and x_c. The wake spreads, σ_y and σ_z, define the width and height of the wake, which broaden with increasing downwind distance x_1. These wake shape variables are not static; they are calculated individually for each OP based on its inherited parameters (e.g., $a^{\mathrm{OP}}, \gamma^{\mathrm{OP}}$), its downwind position x_1, and the ambient turbulence intensity I. The specific parametrisation follows [9], these detailed formulas are omitted from this chapter and are presented in Appendix A. While turbulence intensity is a critical input for these equations (primarily governing the wake recovery rate), it is treated as a static, known parameter. This reflects a fundamental design choice of the FLORIDyn model, which simulates the advection of the mean wake deficit and does not capture dynamic turbulent wind speed fluctuations.

The deficit amplitude, Δ_{x_1}, determines the maximum speed reduction at the center of the wake at a distance x_1. This amplitude is not constant; it diminishes as the wake recovers and is calculated based on the turbine's current yaw angle γ and thrust coefficient C_T. Following [5], a compact expression is given by

$$\Delta_{x_1} \;=\; 1 - \sqrt{1 - \frac{C_\mathrm{T} \cos^2\gamma}{8\,(\sigma_y/D)\,(\sigma_z/D)}} \,, \tag{2-2}$$

with D being the rotor diameter. This calculation links the deficit amplitude directly to thrust coefficient, C_T (later defined in Eq. 2-9), and the wake's current spread relative to the rotor size.

Finally, the model employs two coordinate frames for its calculations. While the wake physics are resolved in the local, wake-aligned frame $\mathcal{K}_1$, the overall farm layout is fixed in a global world frame, $\mathcal{K}_0 = \{x_0, y_0, z_0\}$. The model maps positions and velocities between these frames using a rotation, $R_{01}(\theta)$, which is defined by the measured wind direction θ.

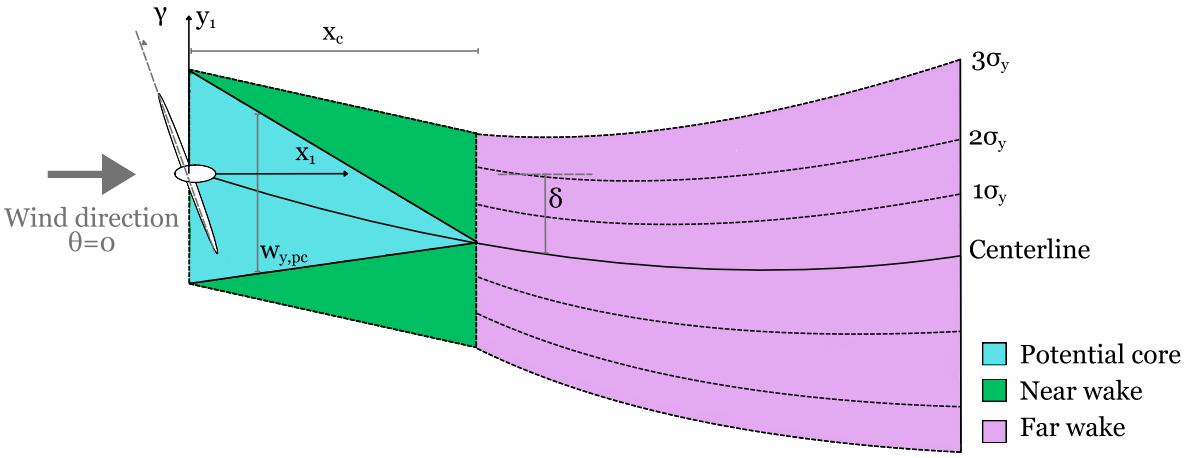

Figure 2-3: Schematic of a yawed-turbine Gaussian wake, adapted from [9]. The wake is partitioned into the potential core (blue), near wake (green) and far wake (purple). Variables x_1, y_1 and γ denote the downwind, crosswind and yaw-angle directions in the local frame. The potential-core width and length are $w_{y,\mathrm{pc}}$ and x_c; δ is the yaw-induced deflection; and σ_y, σ_z are lateral and vertical spreads.

Observation points

Figure 2-4 illustrates the core mechanism of FLORIDyn: the propagation of discrete OPs through the wind farm. At $k = 1$, a new OP (blue) is created at the rotor plane of Turbine 1 and convected downstream with the wind. As time advances ($k = 9$ and $k = 11$), this OP transports flow and turbine-state information, specifically the yaw angle γ and axial induction factor a inherited from its turbine upon creation, through the domain. It eventually reaches the rotor of Turbine 2, where its influence is accounted for in the downstream wind evaluation. In this way, chains of OPs transmit state changes through the farm with a finite delay, providing a computationally efficient representation of the dynamic memory of upstream actions. For clarity only one chain is shown in Figure 2-4.

While Figure 2-4 shows a single chain, in practice many OPs are generated on the rotor plane each time step to cover the region where wake effects are most pronounced. The number of chains per turbine N_c and the number of OPs per chain N_OP are key design parameters that define the spatial resolution of the model; a higher number provides a finer, more accurate discretisation at a greater computational cost. The OPs are positioned on the rotor plane according to a sunflower distribution [48], which yields an even, radially balanced spread over the rotor and near-wake cross-section without requiring a dense grid. Each new OP receives fixed, non-dimensional crosswind coordinates (ν_y, ν_z) from the sunflower pattern and inherits the current turbine settings (γ, a). These non-dimensional coordinates define the OP's relative lane within the wake. As new OPs are generated in these lanes at subsequent time steps, and the existing ones propagate downstream, they form distinct chains that trace the wake's evolution. The parameter N_OP explicitly defines the maximum length of each chain. When a new OP is generated, pushing the chain length beyond this limit, the oldest OP (the one furthest downstream) is simply discarded from the model dynamics. The non-dimensional crosswind coordinates (ν_y, ν_z) are mapped to the local wake frame $\mathcal{K}_1$ by scaling with the

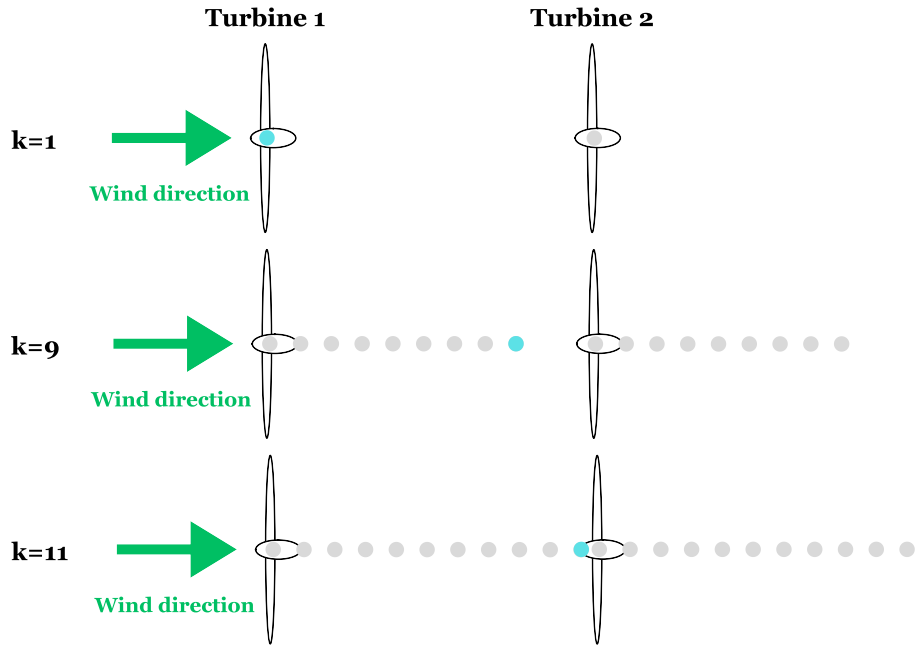

Figure 2-4: Propagation of observation points in time and space. For simplicity, only one chain of observation points is shown. At $k = 1$, a new observation point (blue) is generated at the rotor plane of Turbine 1 and is convected downstream with the wind. As time advances ($k = 9$ and $k = 11$), this observation point carries flow and turbine information through the domain and eventually reaches the rotor plane of Turbine 2, where its influence is accounted for in the downstream turbine's evaluation.

wake's current widths and including the yaw-induced deflection δ:

$$y_1 \;=\; \nu_y\,(6\sigma_y + w_{y,\mathrm{pc}}) + \delta, \qquad z_1 \;=\; \nu_z\,(6\sigma_z + w_{z,\mathrm{pc}}), \tag{2-3}$$

where $\sigma_y, \sigma_z, \delta, w_{y,\mathrm{pc}}, w_{z,\mathrm{pc}}$ correspond to the geometric quantities introduced in Figure 2-3. The factor 6σ provides a $\pm 3\sigma$ coverage of the Gaussian extent, ensuring sampling over the main velocity-deficit region. The model separates the OP's downwind motion from its crosswind position. The downwind motion in the wake frame is modelled as a simple advection step, which updates only the x_1 coordinate:

$$x_1^+ \;=\; x_1 + v_{\mathrm{adv}}\,\Delta t, \tag{2-4}$$

with v_{adv} the advection speed and Δt the model time step. The propagation in the y_1 and z_1 directions is handled differently. It is not a stepwise advection but a re-calculation using Eq. 2-3 at the new x_1^+ position. As the OP moves downwind, the wake itself expands (i.e., σ_y and σ_z increase) and deflects (i.e., δ changes). After re-evaluating these wake variables parameters (Appendix A), the OP's crosswind coordinates are updated (2-3) to their correct position within this evolved wake structure.

Finally, this full 3D propagation is mapped back to the global farm layout. The world-frame position is updated by rotating with the current wind direction θ,

$$\mathbf{r}_0^+ \;=\; \mathbf{r}_0 + R_{01}(\theta)\,(\mathbf{r}_1^+ - \mathbf{r}_1). \tag{2-5}$$

Here, $\mathbf{r}_{0,}$ and $\mathbf{r}_1$ are the OP's (x,y,z) position vectors in the world and wake-aligned frames, respectively. This equation combines the downwind step (2-4) and the new crosswind position (2-3), both contained within $\mathbf{r}_1^+$, to calculate the OP's new global (x,y,z) coordinates. Consistent with the frozen-turbulence assumption, v_{adv} is taken equal to V_∞, which avoids chain crossing and maintains numerical robustness [45]. In practice this means that OPs travel downstream with the free wind speed.

Turbine wake interactions

The final step in the FLORIDyn formulation concerns how the information carried by the OPs is used to estimate the flow conditions experienced by downstream turbines. The effective wind speed seen by an OP, v, is computed by multiplying the free-stream wind speed V_∞ with the remaining fractions after all wake reductions have been applied. Using a multiplicative superposition of deficits, the OP inflow becomes

$$v \;= V_\infty \,(1-r)\, r^f \;=\; V_\infty \,(1-r) \prod_{j \in \mathcal{U}} \left(1 - r_j\right) , \tag{2-6}$$

where r is the reduction due to the *own* (generating) turbine, and r^f is the resulting foreign reduction factor at the OP location. Each OP has a single reduction factor, r. This value is used consistently, whether it is defining its own wake deficit or acting as a foreign-wake contribution to another OP. The foreign reduction factor is the product of all contributing upstream wakes, $\mathcal{U}$. This set of contributors is determined by a proximity search; a foreign OP is typically included if its distance is less than one-quarter of the generating turbine's diameter [9].

The reduction factor r of every OP is linked to the Gaussian wake formulation. Its calculation depends on the OP's location within its own wake, as depicted in Figure 2-3. An OP is identified as being inside this region $\mathcal{C}_{core}$ by a geometric check against the potential core boundaries, which are defined in Appendix A. Inside the potential core, the reduction is treated as constant and maximal, based only on the turbine's thrust C_T (2-9). This constant reduction factor, r_{core}, is given by:

$$r_{core} = 1 - \sqrt{1 - C_\text{T}}. \tag{2-7}$$

Outside the core, in the near wake and far wake regions, the reduction decays from the center according to the Gaussian field. This reduction-factor field is defined as

$$r \;=\; 1 - \frac{v(x_1, y_1, z_1)}{V_\infty} \;=\; \Delta_{x_1} \exp\Big(- \tfrac{1}{2}\tfrac{(y_1-\delta)^2}{\sigma_y^2} - \tfrac{1}{2}\tfrac{z_1^2}{\sigma_z^2}\Big), \tag{2-8}$$

which is a recasting of Eq. 2-1. This calculation is directly driven by the deficit amplitude Δ_{x_1} (2-2), which in turn is computed from the thrust coefficient C_T, computed using the inherited $\gamma^{\text{OP}}, a^{\text{OP}}$:

$$C_\text{T} \;=\; 4a^{\text{OP}}(1 - a^{\text{OP}} \cos\gamma^{\text{OP}}). \tag{2-9}$$

While Eq. 2-6 gives the effective wind speed at the OP's location, the calculation of the wind speed at the turbine rotor is handled separately. To compute the inflow conditions at each rotor, the OPs that lie within the rotor disc (that have not yet been propagated downstream)

are projected onto the full rotor plane ($x_1 = 0$). For a given yaw angle γ, the coordinates of these OPs are

$$y_1 = \nu_y \, D \cos\gamma, \qquad z_1 = \nu_z \, D, \qquad x_1 = 0, \tag{2-10}$$

where (ν_y, ν_z) are the non-dimensional sunflower coordinates and D is the rotor diameter. The rotor-effective wind speed of turbine i, v_i, is then obtained by averaging the OP inflow velocities, weighting each by its associated area $\mathbf{w}$. This weighting is necessary because the OPs are not uniformly spaced across the rotor disc; each weight in the vector $\mathbf{w}$ corresponds to the specific portion of the total rotor area that a single OP represents. These area weights $\mathbf{w}$ are computed using a Voronoi tessellation [49] over the rotor plane, ensuring the integral over the rotor area is accurately approximated by a weighted sum. This step must only include the influence of upstream wakes, as the turbine's own wake develops downstream of the rotor and does not reduce its own inflow. Therefore the free-stream wind V_∞ is reduced only by the foreign reduction factor r^f:

$$v_i \;=\; \mathbf{w}^\top (\mathbf{r}^f \circ V_\infty). \tag{2-11}$$

Finally, this rotor-effective velocity v_i is used to determine power extraction, which is a function of the turbine's control inputs (a_i, γ_i). This coefficients are defined as

$$C_\mathrm{P} \;=\; \kappa \, 4a_i(1-a_i)^2 \cos^p\gamma_i, \tag{2-12}$$

where κ is an empirical efficiency correction and p an exponent describing yaw-related power losses [44]. Combining these relations yields the per-turbine mechanical power:

$$P_i \;=\; \eta \, \tfrac{1}{2}\rho A \, v_{T,i}^3 \, C_\mathrm{P} \;=\; \eta \, \tfrac{1}{2}\rho \tfrac{\pi D^2}{4} \, v_{T,i}^3 \, C_\mathrm{P}, \tag{2-13}$$

with air density ρ, rotor area A, and total efficiency factor η. Through this chain of relations, FLORIDyn links the Gaussian wake description to a dynamic, reduction-factor-based model that efficiently captures how turbine operating states and ambient wind conditions shape the evolving flow field across the farm. All symbols introduced are summarised in Table2-2.

FLORIDyn Summary and Limitations

The FLORIDyn model is designed for control-oriented applications, prioritising computational efficiency over high-fidelity flow resolution. Its primary advantage is that it functions as a quasi-dynamic model, capturing the essential transient effects of wake advection and propagation (as seen in Figure 2-4). This offers a practical compromise between computationally intensive CFD models and overly simplistic steady-state engineering models (e.g. FLORIS). This dynamic behaviour is achieved by executing a sequence of calculations at each discrete time step. A procedural summary of one such step, which links each operation to the defining equations introduced above, is provided in Algorithm 1. In Section 3-1, we reformulate the above model concisely into a state-space model suitable for prediction in an MPC scheme.

The model's predictive accuracy is constrained by several simplifying assumptions, each introducing specific deviations from real-world fluid dynamics. First, the wake is approximated by a symmetric Gaussian velocity-deficit profile (2-1) with a uniform potential core, a geometric simplification that neglects the complex, asymmetric vortical structures and near-wake mixing

Table 2-2: Symbols used in the FLORIDyn formulation.

Symbol	Meaning	Units
V_∞	Free-stream wind speed	$\mathrm{m\,s^{-1}}$
θ	Wind direction (world frame)	rad or deg
I	Ambient turbulence	%
$v(x_1, y_1, z_1)$	Local mean velocity (wake frame)	$\mathrm{m\,s^{-1}}$
Δ_{x_1}	Gaussian deficit amplitude	–
σ_y, σ_z	Lateral/vertical wake spreads	m
δ	Lateral wake deflection	m
D	Rotor diameter	m
r	Reduction factor	–
v	OP effective wind speed	$\mathrm{m\,s^{-1}}$
v_i	Rotor-effective wind speed (turbine i)	$\mathrm{m\,s^{-1}}$
ν_y, ν_z	Non-dimensional cross coordinates (sunflower)	–
y_1, z_1	OP cross coordinates (wake frame)	m
x_1	OP downstream coordinate (wake frame)	m
v_{adv}	OP advection speed	$\mathrm{m\,s^{-1}}$
Δt	Model time step	s
$\mathbf{r}_0$, $\mathbf{r}_1$	OP position in world/wake frames	m
$R_{01}(\theta)$	Rotation from wake to world frame	–
$\mathcal{U}$	Upstream contributors set	–
r^f	Foreign-wake combined fraction at OP	–
$\mathbf{w}$	Normalised OP area weights (rotor)	–
γ_i	Turbine yaw angle	rad or deg
a_i	Axial induction factor	–
γ^{OP}	Inherited OP yaw angle	rad or deg
a^{OP}	Inherited OP axial induction factor	–
C_{T}	Thrust coefficient	–
C_{P}	Power coefficient	–
ρ	Air density	$\mathrm{kg\,m^{-3}}$
A	Rotor area	$\mathrm{m^2}$
η	Efficiency factor	–
$w_{y,\mathrm{pc}}, w_{z,\mathrm{pc}}$	Potential-core widths	m
x_c	Potential-core length	m
P_i	Turbine power	W

present in actual flows. This approximation extends to the wake evolution, where broadening is parameterised through fixed spreads σ_y and σ_z rather than solved fluid dynamics, and aerodynamic coefficients C_{T} and C_{P} are treated as quasi-steady, thereby failing to capture transient dynamic inflow effects. A critical deviation arises from Taylor's frozen-turbulence hypothesis, which convects OPs at the uniform free-stream speed V_∞ (2-4). This decouples the information transport speed from the local flow velocity, assuming turbulent structures remain unchanged during advection. Furthermore, the interaction of multiple wakes is modelled via a simple multiplicative superposition (2-6) based on a proximity threshold, which overlooks the non-linear fluid mixing and turbulence enhancement that occur in dense arrays. Finally, discretising a continuous flow field into a finite number of OPs introduces resolution-dependent errors, where insufficient spatial density or chain length can smooth out sharp gradients and under-represent the physical extent of wake propagation.

Algorithm 1 One FLORIDyn Step

Receive current measured ambient conditions V_∞, θ, I.
Receive current measured outputs $\mathbf{y}$ (e.g., v_i, P_i).
Receive control inputs $\mathbf{u}_i = [\gamma_i \; a_i]$ for every turbine i, aiming to track P_{ref}.

Plant Update (Propagation):
Apply new control $\mathbf{u}_i$ to all turbines.
for each turbine $i = 1 \dots N_T$ **do**
    Create new OPs at rotor i, storing the turbine's input $\{\gamma_i, a_i\}$ as $\{\gamma^{op}, a^{op}\}$.
end for
Compute thrust coefficient C_{T} for all OPs using γ^{op}, a^{op} and Eq. 2-9.
Compute wake shape variables $\sigma_y, \sigma_z, \delta, w_{y,\text{pc}}, w_{z,\text{pc}}$ for all OPs using $x_1, C_{\text{T}}, \gamma^{op}, I$ and Appendix A.
Compute OP crosswind positions y_1, z_1 using $\nu_y, \nu_z, \sigma_y, \sigma_z, \delta, w_{y,\text{pc}}, w_{z,\text{pc}}$ and Eq. 2-3.
Identify potential core set $\mathcal{C}_{\text{core}}$ using $x_1, \gamma^{op}, a^{op}, I$ and Appendix A.

– *Calculate OP own Reduction r* –
For OPs in $\mathcal{C}_{\text{core}}$, compute r using C_{T} and Eq. 2-7.
For OPs *not* in $\mathcal{C}_{\text{core}}$ (i.e., in near/far wake):
    Compute deficit amplitude Δ_{x_1} using C_{T} and Eq. 2-2.
    Compute r using Δ_{x_1} and the Gaussian field in Eq. 2-8.

– *Calculate Foreign Reduction* –
Find set of foreign neighbors $\mathcal{U}$ for each OP (e.g., within $D/4$ distance).
Compute foreign reduction factor r^f using $\mathcal{U}$, r_j and Eq. 2-6.

– *Calculate Turbine Wind Speed* –
Project rotor-plane OPs and get area weights $\mathbf{w}$ using Eq. 2-10 .
Compute rotor-effective speed v_i using $\mathbf{w}, r^f, V_\infty$ and Eq. 2-11.
Compute power coefficient C_{P} using γ_i, a_i and Eq. 2-12.
Compute turbine power P_i using v_i, C_{P} and Eq. 2-13.

– *Propagate OPs for next step* –
Compute effective OP speed v using V_∞, r, r^f and Eq. 2-6.
Compute new downwind position x_1^+ using x_1, v_{adv} and Eq. 2-4.
Compute wake shape variables $\sigma_y, \sigma_z, \delta, w_{y,\text{pc}}, w_{z,\text{pc}}$ using $x_1^+, C_{\text{T}}, \gamma^{op}, I$ and Appendix A.
Compute new wake-frame positions y_1^+, z_1^+ ($\mathbf{r}_1^+$) using $\nu_y, \nu_z, \sigma_y, \sigma_z, \delta, w_{y,\text{pc}}, w_{z,\text{pc}}$ and Eq. 2-3.
Compute new world-frame position $\mathbf{r}_0^+$ using $R_{01}(\theta), \mathbf{r}_1^+, \mathbf{r}_1$ and Eq. 2-5.

2-3 Model predictive control for wind-farm flow control

This section reviews existing literature on the use of model predictive control in wind farm control. First the fundamental principles of MPC are covered, before examining both centralised and non-centralised control architectures and their limitations.

MPC is an optimisation-based feedback strategy in which a dynamic model is used to predict the system behaviour over a finite prediction horizon and an optimiser computes a sequence of future control inputs that minimises a chosen cost subject to constraints. The core of this strategy is the receding-horizon implementation, which is illustrated in Figure 2-5. At

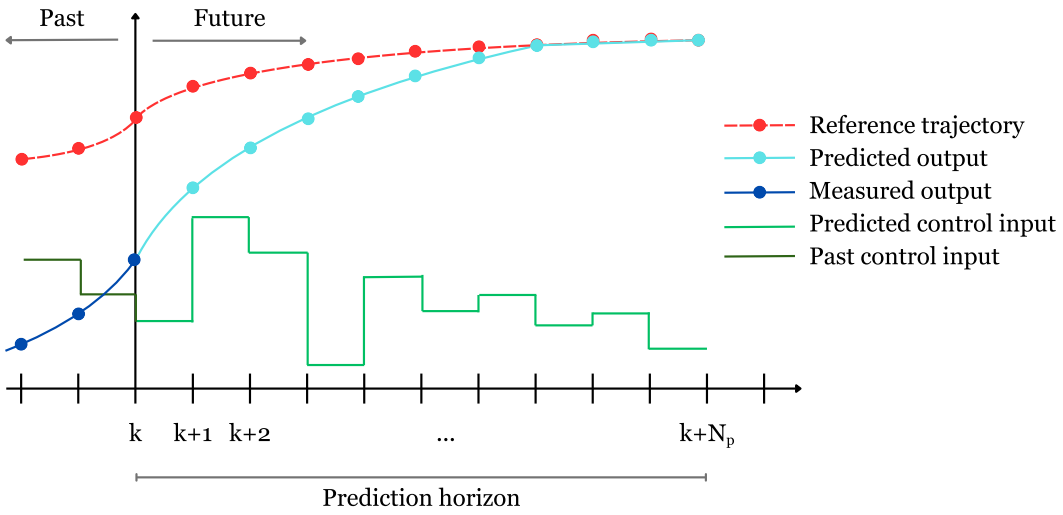

Figure 2-5: Schematic representation of the Model Predictive Control (MPC) principle. At each time step k, the controller uses a model of the system to predict its future behaviour over a finite prediction horizon N_p. The predicted output (light blue) is optimised to follow the reference trajectory (red) by computing a sequence of control inputs. Only the first control input in the sequence (green) is applied to the system before the optimisation is repeated at the next time step. The measured output (dark blue) represents the system response up to the current time, while the remaining predictions indicate the future evolution under the computed inputs.

the current time step k, the controller has access to the system's history, represented by the measured output (dark blue line) and past control inputs. Using this information, it looks ahead over a finite Prediction horizon of N_p steps. The controller uses a dynamic model of the system to forecast a predicted output (light blue line) that it aims to steer as close as possible to a desired reference trajectory (red line). To achieve this, it solves an optimisation problem to compute an entire sequence of optimal predicted control inputs (green line) for the whole horizon. Crucially, only the first element of this optimal control sequence (the control input at time k) is actually applied to the system. At the next time step, $k+1$, the horizon 'recedes' (shifts one step forward). The controller takes a new measurement from the system, and the entire optimisation process is repeated to find a new optimal control sequence. This receding-horizon implementation allows MPC to use feedback to continuously correct for model mismatch and unmeasured disturbances [38].

A general nonlinear MPC optimisation problem solved at each time step k is stated in Eq. 2-14. A predicted variable at time step n of the MPC prediction, solved at time step k, is

denoted as $\mathbf{x}(n|k)$.

$$\min_{\mathbf{U}} \quad J = \sum_{n=k}^{k+N_\mathrm{p}-1} l\big(\mathbf{x}(n|k), \mathbf{u}(n|k)\big) + F\big(\mathbf{x}(k+N_\mathrm{p}|k)\big) \tag{2-14a}$$

$$\text{s.t.} \quad \text{for } n = k, \ldots, k+N_\mathrm{p}-1,$$

$$\mathbf{x}(k|k) = \mathbf{x}(k), \tag{2-14b}$$

$$\mathbf{x}(n{+}1|k) = f\big(\mathbf{x}(n|k),\, \mathbf{u}(n|k),\, \mathbf{d}(k)\big), \tag{2-14c}$$

$$\mathbf{u}(n|k) \in \mathcal{U}, \tag{2-14d}$$

$$\mathbf{x}(n|k) \in \mathcal{X}, \tag{2-14e}$$

$$\mathbf{x}(k+N_\mathrm{p}|k) \in \mathcal{X}_f \tag{2-14f}$$

The controller's objective is to minimise the cost function J (2-14a), which is typically a sum of the stage cost, $l(\cdot)$ (e.g., penalising tracking error at each step), and a terminal cost, $F(\cdot)$ (penalising the final predicted state). This optimisation is subject to several constraints. The initial condition (2-14b) enforces that the prediction must start from the measured output $\mathbf{x}(k)$ from Figure 2-5. The nonlinear dynamic model $f(\cdot)$ (2-14c) defines how the controller predicts the future states. Constraints on the dynamics, such as safety limits, are enforced on the control inputs (2-14d) and system states (2-14e). A terminal constraint $\mathcal{X}_f$ (2-14f) may also be applied to ensure stability.

When designing an MPC controller, design choices such as the model complexity, horizon length, cost weights and constraint formulation, incur trade-offs between prediction accuracy, control performance and online computational cost. Sometimes the effects are unclear, e.g. the effects of constraints and cost terms on computation is very hard to predict [38]. In the context of wind farm control, a primary design choice that dictates these trade-offs is control architecture, which is broadly divided into centralised and non-centralised schemes. The following subsection presents centralised MPC approaches for wind-farm flow control, followed by a subsection that discusses non-centralised approaches and their main advantages and disadvantages.

2-3-1 Centralised MPC for wind-farm flow control

The standard approach is to formulate the MPC problem in a centralised way, having a single optimiser finding the optimal control sequence for all elements of the system. For wind farm control, this means a single optimiser computes the control inputs for the whole farm jointly. This centralised MPC formulation is theoretically optimal, as it is the only approach capable of finding the true global optimum. This is because the optimiser can simultaneously account for all aerodynamic coupling and exploit wake-steering and induction trade-offs to meet farm-level objectives.

However, this approach can be computationally demanding for large, networked systems. Optimisation over all turbines simultaneously, quickly becomes expensive as farm size or model fidelity grows, which limits horizon length, model complexity, and achievable sampling rates in real time. Therefore, in wind farm control literature, this control architecture is typically used in specific contexts that manage this complexity, for example by using a reduced-order prediction model [44]. In other MPC frameworks, the computational challenge is bypassed

by using a prediction model that does not take the coupled effects between turbines into account [40]. Less commonly, it is used as a high-fidelity benchmark to explore the theoretical potential of coordinated control, without the constraint of real-time computation [35].

Of particular relevance to the approach presented in this work is [44], which present a centralised MPC controller that explicitly models wake transport dynamics using observation points together with Gaussian wake parametrisations, inspired by FLORIDyn [9]. The resulting finite-horizon nonlinear problem is formulated into a mixed-integer quadratically-constrained quadratic program (MIQCQP) using approximation and reformulation techniques. The optimisation problem is solved online to obtain yaw and axial induction inputs to track a prescribed farm power reference. The case study demonstrates proper tracking performance, even power sharing among turbines, and solver times below the sampling time which underlines the real-time potential of this approach.
However, the reported solver times are already on the same order of magnitude as the sampling interval, indicating that even moderate increases in model complexity or farm size could quickly make the approach computationally infeasible for real-time use. This limitation highlights the potential need for non-centralised control strategies to reduce computational demands. Furthermore, the fundamental proposition that reformulating the original nonlinear problem into a MIQCQP reduces computational burden is not supported by an ablation study. In fact, whether such a simplification yields a net benefit is highly case-dependent and may not generalise. Additionally, mixed-integer problems are known to scale exponentially with problem size, posing a major drawback as the number of turbines increases [50].

2-3-2 Non-centralised MPC for wind-farm flow control

Given the computational drawbacks of a purely centralised approach, it can be beneficial to consider a non-centralised control architecture for large, networked systems. This category covers a range of architectures, which are often tailored to the specific characteristics of the system [41]. The main structures include decentralised, distributed, and hierarchical control.

In a decentralised control scheme, the system is divided into several parts, such as individual turbines or subsystems of multiple turbines. Each part is governed by its own local MPC regulator, which operates independently without communication or negotiation with the others [15]. A decentralised approach assumes the subsystems are decoupled, or that their interactions can be deliberately neglected. In contrast, distributed control allows for communication among the local controllers [15]. This enables them to negotiate or coordinate, sharing information about shared variables or constraints to work towards a common goal. Additionally, hierarchical control layers can be added to the architecture. These layers can operate at different time scales or ensure consistency across shared variables [41]. In practice, many control schemes combine elements from these different structures to create intermediate approaches. For the remainder of this section, we make the general distinction between decentralised and distributed MPC approaches as they apply to wind farm (flow) control.

Decentralised MPC for wind-farm flow control

In the context of wind farm control, decentralised MPC can be interpreted in two distinct ways. The simplest, and less common, approach treats turbines as truly individual entities.

In these studies, turbines are controlled independently, and the coupling through wakes is deliberately neglected or not modelled [26, 27]. This formulation does not require a farm-level view and does not treat the wind farm as a strongly coupled system.

However, most decentralised MPC schemes in the wind farm literature emerge from partitioning strategies, which do begin with a farm-level view [10, 14, 18, 32, 42]. These strategies divide a wind farm into subsystems with the explicit goal of creating groups that have strong coupling within the subsystem but weak coupling between subsystems [16]. These subsystems, which can be grouped based on aerodynamic or electrical coupling, are then controlled independently . This approach relies on the assumption that inter-subsystem coupling is negligible and that wake effects are sufficiently contained within each subsystem's boundaries. A key limitation of this method is that, in almost all mentioned cases, the partitioning is treated as fixed during operation [10, 14, 18, 32]. These static partitions can become suboptimal in dynamic environments with changing wind conditions. Only in [42], this issue is explicitly addressed, where multiple partitionings are precomputed for different wind directions and then selected from a lookup table during control.

Overall, decentralised control reduces computational load and supports modular implementation by controlling subsystems or individual turbines without communication or shared optimisation, making it suitable for large systems. Decentralisation appears mostly at the subsystem level and is typically justified by assuming limited interaction between subsystems and stable wind conditions. The main advantage of decentralised control approaches is its scalability as distinct subsystems can be solved in parallel, but this comes at the cost of coordination. Without communication across subsystems, global objectives such as load balancing or wake mitigation cannot be fully addressed. As a result, the control strategy may perform well locally but will not achieve the same overall performance as a centralised approach. This limitation becomes more critical under variable wind conditions or in tightly spaced farms, where coupling effects are stronger.

Distributed MPC for wind-farm flow control

In contrast to decentralised control, distributed control introduces a notion of communication or cooperation between subsystems or individual turbines. Distributed control approaches can be classified in multiple ways. A key distinction lies in how and when coordination occurs. Some distributed control algorithms allow for repeated rounds of communication and negotiation within each control interval or sampling time [26, 25, 27, 43, 51]. These are known as iterative algorithms. In contrast, non-iterative algorithms perform only a single round of communication before each control action is applied [6, 36]. Next to that, distributed control strategies can either be independent [6], where each controller optimises its own objective without considering others, or cooperative [26, 25, 27, 36, 43, 51], where controllers jointly pursue a shared global objective. This predominance of cooperative strategies is expected: wind farms are intrinsically coupled systems through wake interactions, so coordinating actions toward a common objective (for example power maximisation or reference tracking) naturally improves collective performance compared with local decisions.

Iterative distributed MPC schemes, such as those using alternating direction method of multipliers (ADMM) or dual decomposition with fast gradient (DDFG), have been shown to converge to the centralised optimum [25, 43]. However, this convergence is only guaranteed for

convex optimisation problems. This limits the complexity of the models used within the optimisation framework, which are often restricted to linearised (turbine) models [43]. Furthermore, full convergence is rarely achieved in practice due to real-time constraints, introducing a trade-off between optimality and computational cost. While residual-based stopping criteria [25] or fixed iteration counts [51] improve computational tractability, scalability remains an issue, as convergence rates degrade with farm size. To reduce communication overhead, most schemes avoid fully connected topologies. The choice of local information exchange improves scalability, but it may slow convergence, especially in larger wind farms where tighter coupling and longer communication paths increase the number of iterations required to reach agreement. Overall, iterative distributed control offers a promising alternative to centralised schemes. However, it introduces a trade-off between the desirable convergence properties of iterative optimisation and the computational and communication cost of performing iterations. This trade-off is addressed differently in the non-iterative methods discussed next.

Non-iterative distributed approaches reduce computational and communication burden by avoiding repeated updates, making them attractive for real-time and large-scale applications. However, this comes at the cost of limited coordination and reduced optimality, as each subsystem or turbine solves its problem independently with only partial information. These schemes can be structured with either sequential [39] or parallel [52] solving within a single time step. A sequential structure involves a pre-defined order of computation, where subsystem i must compute and pass its solution to subsystem $i+1$. This allows for faster information propagation and tighter coordination within that single step (as $i+1$ gets the most current information from i), but the total computational time increases with the number of subsystems, creating a bottleneck. A parallel structure, where all subsystems compute their solution simultaneously, is highly scalable and avoids this latency, as the total computation time is only that of the slowest subsystem. The drawback is that all subsystems solve using 'stale' information (from the previous time step k), which can limit the quality of coordination. The studies by [6] and [36] both employ a parallel non-iterative structure. In [6], all turbines solve their local MPC problem simultaneously, enabling local wake-aware control, but lacking any mechanism to ensure fair or optimal power distribution across the farm. Similarly, in [36], all turbines within a cluster update their consensus value in parallel, which improves scalability but means that this value remains approximate during transients. The communication topologies for both non-iterative studies are sparse, supporting modularity and robustness. Overall, the non-iterative distributed control schemes prioritise feasibility over optimality, making them suitable when full convergence or central oversight is impractical.

2-3-3 Limitations in model predictive control of wind farms

A central challenge in MPC literature is the fundamental trade-off between computational scalability and prediction accuracy, which becomes especially pronounced when considering wake interactions. A centralised model predictive control (CMPC) formulation, which solves for all turbines at once, can explicitly account for all aerodynamic couplings and find a global optimum. However, this approach prioritises prediction accuracy at the expense of scalability, becoming computationally prohibitive for large farms. In contrast, decentralised MPC prioritises scalability by dividing the problem, but this efficiency is typically achieved by neglecting wake interactions entirely and treating each turbine as an independent system. This

leads to a lack of coordination and suboptimal farm-level performance. Distributed model predictive control (DMPC) architectures are therefore investigated as a promising middle ground, as they have the potential to be both scalable and flow-aware. However, a central limitation of distributed approaches is that they fail to achieve both. In practice, to remain computationally tractable, wake effects are frequently not modelled explicitly. Many schemes treat the effective wind speed as an external, locally measured input, thus operating with limited awareness of farm-scale coupling. Consequently, the predictive capability of MPC is not leveraged to account for how upstream control decisions will dynamically affect the inflow, and thus the performance, of downstream turbines. This limitation appears by design choice in the model or control structure. Some studies include wakes only implicitly through short prediction horizons or static effective-speed estimates [43, 51], while others omit wake interactions entirely [25, 26, 27, 36]. Even partitioning methods that are designed to account for farm structure often leave the MPC loop free of direct wake dynamics [18, 42].

This creates a significant gap between the need for scalable, high-performance control and the lack of controllers that successfully leverage dynamic wake models in a computationally tractable way. The few exceptions that attempt to fill this gap [6, 44] typically rely on low-fidelity models or approximations. This thesis aims to address this specific gap by developing a distributed MPC framework that is both scalable and explicitly considers dynamic wake coupling through its choice of model and distributed control scheme.

2-4 Summary

This chapter provides a background on wind-farm flow control and reviews the literature on wind-farm flow models and control strategies. The review of wind-farm flow models identifies a continuum from steady, low-fidelity engineering models to high-fidelity models. FLORIDyn is positioned as a key 'middle-ground' approach, as it explicitly models wake advection with observation points while remaining compact enough for use in receding-horizon control.

Similarly, MPC is identified as the natural control paradigm for handling farm-level objectives and constraints. As established in Section 2-3-3, the central challenge is the trade-off between prediction accuracy and computational scalability, which leaves a gap in existing literature. This gap directly motivates the thesis question, which investigates how FLORIDyn-based prediction models and distributed MPC can be combined to balance computational tractability and control performance.

Chapter 3

Nonlinear model predictive control for wind-farm flow control

This chapter presents the main theory and methodology of the thesis. First the FLORIDyn model is formalised, defining it in terms of its states and inputs for model predictive control (MPC) prediction. The chapter then discusses the prediction model formulation and approximations. Next, the centralised MPC formulation is presented, leveraging this proposed prediction model. Finally, to address the computational burden, two distributed variants are proposed. The chapter ends with a short summary.

3-1 FLORIDyn in control framework

To apply MPC, we formalise the FLOw Redirection and Induction Dynamics (FLORIDyn) model described in Chapter 2 as a discrete-time nonlinear state-space system. This system takes the general form:

$$\mathbf{X}(k+1) = f(\mathbf{X}(k), \mathbf{U}(k), \mathbf{d}(k)) \tag{3-1}$$

$$\mathbf{Y}(k) = g(\mathbf{X}(k), \mathbf{U}(k), \mathbf{d}(k)) \tag{3-2}$$

where k is the time index, $\mathbf{X}$ is the full system state, $\mathbf{U}$ is the control input matrix, $\mathbf{d}$ is the measurable disturbance vector, and $\mathbf{Y}$ is the system output matrix. For clarity, the explicit time index (k) will be omitted from the equations in the remainder of this chapter. States, inputs and outputs are organised in matrices for notational simplicity. Furthermore, a notational convention is adopted where variables with a single subscript i are associated with turbine i, while variables with three subscripts i, j, l are associated with the observation point (OP) of the i-th turbine, the j-th chain, and the l-th observation point in that chain.

The control input matrix $\mathbf{U}$ contains the set of actions for all turbines at a single time step, where each column represents the input vector $\mathbf{u}_i$ of a single turbine, containing its yaw angle

γ_i and axial induction factor a_i:

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_{N_\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_{N_\mathrm{T}} \\ a_1 & a_2 & \dots & a_{N_\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{2 \times N_\mathrm{T}}. \tag{3-3}$$

The measurable disturbances $\mathbf{d}$ are external variables that affect the system. This vector includes the free-stream wind speed V_∞, the global wind direction θ and the ambient turbulence intensity I:

$$\mathbf{d} = \begin{bmatrix} V_\infty & \theta & I \end{bmatrix}^\top. \tag{3-4}$$

The disturbances are assumed to be constant over the wind farm. The system state $\mathbf{X}$ represents the complete dynamic state of the wind farm. In FLORIDyn, this state is stored in the OPs advecting downstream from each turbine. We define a fixed-size state by assigning each turbine i a set of N_c chains, and each chain j a fixed length of N_OP OPs. The total state $\mathbf{X}$ is the aggregation of the individual state matrices for each of the N_T,

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_{N_\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{(6 \times N_\mathrm{c}) \times (N_\mathrm{OP} \times N_\mathrm{T})}. \tag{3-5}$$

The state for a single turbine i, $\mathbf{X}_i$, is the collection of its N_c chains, formed by vertically stacking all its individual chain matrices:

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{X}_{i,1} \\ \mathbf{X}_{i,2} \\ \vdots \\ \mathbf{X}_{i,N_\mathrm{c}} \end{bmatrix} \in \mathbb{R}^{(6 \times N_\mathrm{c}) \times N_\mathrm{OP}}. \tag{3-6}$$

The state for a single chain j of turbine i, $\mathbf{X}_{i,j}$, is a matrix where each column l is the state vector for a single OP:

$$\mathbf{X}_{i,j} = \begin{bmatrix} \mathbf{x}_{i,j,1} & \mathbf{x}_{i,j,2} & \dots & \mathbf{x}_{i,j,N_\mathrm{OP}} \end{bmatrix} \in \mathbb{R}^{6 \times N_\mathrm{OP}}. \tag{3-7}$$

The state vector for a single OP l on chain j of turbine i is:

$$\mathbf{x}_{i,j,l} = \begin{bmatrix} x_{i,j,l} & y_{i,j,l} & z_{i,j,l} & \phi_{i,j,l} & \gamma^{\mathrm{OP}}_{i,j,l} & a^{\mathrm{OP}}_{i,j,l} \end{bmatrix}^\top \in \mathbb{R}^{6 \times 1}. \tag{3-8}$$

The coordinates $[x_{i,j,l} \quad y_{i,j,l} \quad z_{i,j,l}]^\top$ are the OP's global world frame position (in $\mathcal{K}_0$). The coordinate $\phi_{i,j,l}$ is the OP's local downwind coordinate. To avoid notational confusion, this variable is introduced to represent the position along the x_1-axis, which was previously referred to as x_1 in Section 2-2-2 and Figure 2-3. The yaw angle $\gamma^{\mathrm{OP}}_{i,j,l}$ and axial induction factor $a^{\mathrm{OP}}_{i,j,l}$ are the inherited control inputs from the turbine that each OP inherits on creation and carries for later calculations. This structure creates a total state vector $\mathbf{X}$ with a dimension of $(6 \times N_\mathrm{c}) \times (N_\mathrm{OP} \times N_\mathrm{T})$ states. Hence, the state dimension scales with the number of turbines N_T and the per turbine resolution $(N_\mathrm{c} \times N_\mathrm{OP})$.

The state update function f (3-1) describes the OP propagation, which is intuitively visualised as a matrix column shift. For each state matrix $\mathbf{X}_{i,j}$ (3-7) representing a single chain, the data in each column is advected and shifted one position to the right at every time step. The oldest OP (in the final column) is discarded, and a new OP (in the first column) is created. At the start of a simulation ($k = 0$), these matrices are initialised as zero matrices,

$\mathbf{X}_{i,j}(0) = \mathbf{0}_{6\times N_{\mathrm{OP}}}$, and subsequently fill up from the first column as the simulation proceeds. This update operation can be expressed concisely using block matrix notation:

$$\mathbf{X}^{+}_{i,j} = \begin{bmatrix}\mathbf{x}_{\mathrm{new}} & \mathrm{Advect}(\mathbf{X}_{i,j}[:, 1 : N_{\mathrm{OP}} - 1], \mathbf{d})\end{bmatrix}. \tag{3-9}$$

Here, $\mathbf{X}_{i,j}[:, 1 : N_{\mathrm{OP}} - 1]$ is the $6 \times (N_{\mathrm{OP}} - 1)$ submatrix containing all but the last (oldest) OP column. The Advect$(\cdot)$ function is applied column-wise to this submatrix: it updates the $\phi_{i,j,l}$ and position coordinate states according to their dynamics, while the $2 \times (N_{\mathrm{OP}} - 1)$ matrix of inherited parameters $[\gamma^{\mathrm{OP}}_{i,j,l} \quad a^{\mathrm{OP}}_{i,j,l}]^{\top}$ is simply copied. The final column $\mathbf{X}_{i,j}[:, N_{\mathrm{OP}}]$ is discarded as it is shifted out of the fixed-size matrix. The new 6×1 state vector, $\mathbf{x}_{\mathrm{new}}$, is created from the turbine's current position and control inputs:

$$\mathbf{x}_{\mathrm{new}}(k) = \begin{bmatrix} x_{i,j,l} & y_{i,j,l} & z_{i,j,l} & 0 & \gamma_i & a_i \end{bmatrix}^{\top}. \tag{3-10}$$

where $[x_{i,j,l} \quad y_{i,j,l} \quad z_{i,j,l}]^{\top}$ is the position of OP i, j, l upon creation at the rotor plane of turbine i. The downwind position $\phi_{i,j,l}$ is initialised to 0, and the OP inherits the current control inputs γ_i and a_i of turbine i. For $l \in [2, N_{\mathrm{OP}}]$, the new state is updated using $\phi^{+}_{i,j,l} = \phi_{i,j,l-1} + v_{\mathrm{adv}}\,\Delta t$ (Eq. 2-4). The global position states $x_{i,j,l}, y_{i,j,l}, z_{i,j,l}$ are updated according to Eq. (2-5), which depends on the current wind direction θ. This process involves computing the new wake shape variables $(\sigma_y(x_{i,j,l}), \sigma_z(x_{i,j,l}), w_{y,\mathrm{pc}}(x_{i,j,l}), w_{z,\mathrm{pc}}(x_{i,j,l}), \delta(x_{i,j,l}))$ as detailed in Appendix A to determine the new local coordinates in the wake frame $\mathcal{K}_1$, which are then rotated back to the world frame $\mathcal{K}_0$ using θ. Together, these update rules for the creation (3-10), advection (2-4), (2-5), and shifting (3-9) of all OPs in all chains define the complete state update function $f(\mathbf{X}(k), \mathbf{U}(k), \mathbf{d}(k))$ (3-1), for every OP's state $x_{i,j,l}$.

The output $\mathbf{Y}$ of the system contains the rotor-effective wind speed v_i and power P_i, with every column corresponding to a turbine i:

$$\mathbf{Y} = \begin{bmatrix} v_1 & v_2 & \dots & v_{N_{\mathrm{T}}} \\ P_1 & P_2 & \dots & P_{N_{\mathrm{T}}} \end{bmatrix} \in \mathbb{R}^{2\times N_{\mathrm{T}}}. \tag{3-11}$$

The output function g (3-2) contains all the calculations that translate the current states, inputs, and disturbances into the final output matrix. This function g translates the state $\mathbf{X}$ into measurable outputs by executing a defined sequence of algebraic equations (from Chapter 2) across every OP. First, all internal reduction factors are computed. The thrust coefficient C_{T} is found from the states $[\gamma^{\mathrm{OP}}_{i,j,l},\ a^{\mathrm{OP}}_{i,j,l}]^{\top}$ (2-9). This C_{T} is used to find the deficit amplitude Δ_ϕ (2-2):

$$\Delta_\phi(x_{i,j,l}) = 1 - \sqrt{1 - \frac{C_{\mathrm{T}}(x_{i,j,l})\cos^2\gamma^{\mathrm{OP}}_{i,j,l}}{8\,(\sigma_y(x_{i,j,l})/D)\,(\sigma_z(x_{i,j,l})/D)}}\,. \tag{3-12}$$

The OP's own reduction factor $r_{i,j,l}$ is then found using $\Delta_\phi(x_{i,j,l})$ (2-8). All foreign reduction factors $r^{f}_{i,j,l}$ are found by applying the superposition (2-6):

$$r^{f}_{i,j,l} = \prod_{(i',j',l')\in\mathcal{U}_{i,j,l}} (1 - r_{i',j',l'}). \tag{3-13}$$

Here, $\mathcal{U}_{i,j,l}$ is the set of all foreign OPs that influence the OP i, j, l. The indices i', j', l' are used to denote each neighbour OP in the set, and $r_{i',j',l'}$ is the own reduction factor of that

specific neighbour OP. Second, g calculates the rotor-effective wind speeds v_i (2-11), using the foreign reduction factors $r^f_{i,j,l}$ (3-13) of the OPs at the rotor plane:

$$v_i = \mathbf{w}^\top(\mathbf{r}^f_{i,j,l} \circ \mathbf{V}_\infty). \tag{3-14}$$

Third, g calculates the turbine powers P_i. It uses the current control input $\mathbf{u}_i$ to find the power coefficient $C_{\mathrm{P},i}$ (2-12):

$$C_{\mathrm{P},i}(\mathbf{u}_i) = \kappa\, 4a_i(1-a_i)^2 \cos^p\gamma_i. \tag{3-15}$$

The turbine power P_i (2-13) is then calculated using v_i (3-14) and $C_{\mathrm{P},i}(\mathbf{u}_i)$ (3-15):

$$P_i = \eta\, \tfrac{1}{2}\rho \tfrac{\pi D^2}{4}\, v_i^3\, C_{\mathrm{P},i}(\mathbf{u}_i). \tag{3-16}$$

Collectively, these equations constitute the complete discrete-time nonlinear state-space formulation f, g required for MPC design.

3-2 Prediction models

In this work, we propose to use the nonlinear state-space model defined by Eqs. (3-1) and (3-2) as the prediction model within an MPC framework. We propose to employ the nonlinear dynamics directly, avoiding the linearisation or reformulation strategies used in approaches such as [44]. This choice is motivated by the premise that sufficient computational speed can be achieved without these simplifications, thereby avoiding the modelling errors that degrade the prediction of strongly nonlinear wake interactions.

A foundational assumption for this control framework is that all ambient conditions are treated as static over the prediction horizon. This means that the free-stream wind speed V_∞, wind direction θ, and turbulence intensity I are assumed to remain constant at their currently measured values for all steps within the prediction window. This assumption not only simplifies the prediction problem but is also a critical prerequisite for the distributed model predictive control (DMPC) structures proposed later, as their fixed sequential solution order and static subsets are only valid for a dominant, unchanging flow direction. The validity and limitations of this assumption are discussed in Section 5-2.

The MPC controller requires the current state $\mathbf{X}(k)$ to start its prediction. While the full state $\mathbf{X}(k)$, which contains all OP data, is not directly measurable, the outputs $\mathbf{Y}(k)$, control inputs $\mathbf{U}(k)$ and disturbances $\mathbf{d}(k)$ are. The controller therefore maintains an internal estimate of the state, $\tilde{\mathbf{X}}(k)$, by running the model dynamics from the previous step, effectively using the prediction model as an open-loop observer. This provides the necessary initial state $\tilde{\mathbf{X}}(k|k) = \tilde{\mathbf{X}}(k)$ for the optimisation. Future research could investigate more advanced closed-loop estimation techniques, such as moving horizon estimation (MHE) or Luenberger observers, to better address model-plant mismatch.

A practical consideration in MPC is the tractability of solving the optimisation problem for a given prediction model. The total state dimension of $\mathbf{X}(k)$, which is $(N_\mathrm{c} \times 6) \times (N_\mathrm{T} \times N_\mathrm{OP})$, scales with farm size N_T and with the per-turbine discretisation N_c and N_OP. This large state can make the optimisation computationally prohibitive for real-time application. To enable

real-time decision-making, we therefore propose to employ a reduced-order prediction model. The state-space for the prediction model is :

$$\tilde{\mathbf{X}}(k+1) = \tilde{f}(\tilde{\mathbf{X}}(k), \mathbf{U}(k), \mathbf{d}(k)) \tag{3-17}$$

$$\tilde{\mathbf{Y}}(k) = \tilde{g}(\tilde{\mathbf{X}}(k), \mathbf{U}(k), \mathbf{d}(k)) \tag{3-18}$$

The approximation is achieved by reducing the state dimension of $\tilde{\mathbf{X}}$ compared to $\mathbf{X}$. Within the dynamics defined in Eq. (3-1), two important tunable parameters are the number of OPs per chain, N_{OP}, and the number of chains per turbine, N_{c}. The prediction model $\tilde{f}$ uses a coarser state discretisation than the true model f by reducing N_{OP}, and optionally reducing N_{c}. This reduces the total dimension of $\tilde{\mathbf{X}}$, making the optimisation problem less complex and faster to solve. The full state $\mathbf{X}$ has the dimension of N_{OP} and N_{c} that best match reality. The specific design choices for the prediction model reduction are discussed in the subsequent subsections. For the number of chains, N_{c}, a comparison between different values is made.

3-2-1 Number of observation points per chain

The number of OPs within one chain, N_{OP}, defines the length of a chain and thus, sets how far wake information is transported in the streamwise direction. In other words, chain length fixes the downstream reach of the wake signal that the model can represent. In order to reduce the state dimension and render the MPC optimisation problem less computationally intensive, we propose to deliberately limit this downstream reach by using shorter chains than in the full-complexity model. The aim is to keep enough information to predict the key interactions, while removing resolution that adds a large computational burden for limited gain. Specifically, chains in the prediction model are chosen long enough such that OPs created at an upstream turbine pass slightly beyond the rotor of the first downstream turbine. This gives the prediction model the ability to estimate the rotor-effective wind speed v_i of the first downstream turbine with the upstream influence included. This difference is illustrated in Figure 3-1: the reduced-length chain (blue) terminates shortly past each downstream rotor, whereas the full-length chain (grey) continues further downstream. In practice, this means wake information from Turbine 1 does not reach Turbine 3 in the reduced-length case, but it does in the full-length case.

The physical distance covered by a chain depends on the number of OPs a chain consists of. To determine a sufficient N_{OP}, we first calculate the minimum number of OPs required to span the physical distance between adjacent turbines.The distance Δs an OP advects in a single time step Δt is defined by the advection speed v_{adv}, which is defined as the free wind speed V_∞, the model time step Δt, and the wind angle θ:

$$\Delta s = V_\infty \ \Delta t \ \cos(\theta). \tag{3-19}$$

The minimum number of intermediate OPs needed to span the streamwise distance L_x between adjacent upstream-downstream turbines, which we denote N_{span}, is therefore

$$N_{\mathrm{span}} = \left\lceil \frac{L_x}{\Delta s} \right\rceil = \left\lceil \frac{L_x}{V_\infty \ \Delta t \ \cos(\theta)} \right\rceil. \tag{3-20}$$

The chosen chain length for the prediction model, N_{OP}, must be large enough to contain this span (i.e., $N_{\mathrm{OP}} \geq N_{\mathrm{span}}$). This choice must therefore account for the farm layout and turbine

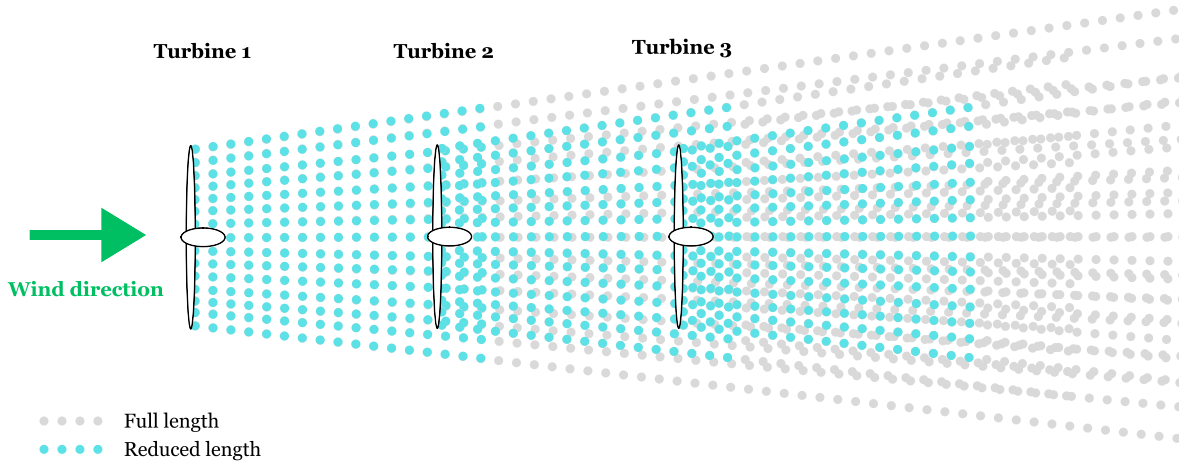

Figure 3-1: Comparison of chain lengths. Blue dots show the reduced-length chains that extend just past the next rotor, sufficient to include the upstream influence on the first downstream turbine. Grey dots show the full-length chains that continue far downstream to capture far-wake development and multi-turbine interactions. Wind flows from left to right.

spacing L_x, likely wind directions θ, expected ranges of V, and the sample time Δt, so that chains reach at least the first downstream rotor over the operating range of interest. While L_x and Δt are static design parameters, V_∞ and θ can be time-varying disturbances. The prediction model $\tilde{f}$ must therefore use a single N_{OP} that is large enough to satisfy Eq. 3-20 across the entire operational range, typically by solving for the worst-case (e.g., minimum V_∞ and maximum θ). A more advanced, adaptive formulation could adjust N_{OP} online based on measured $V_\infty(k)$ and $\theta(k)$, allowing the controller to dynamically manage the trade-off between the state dimension of $\tilde{\mathbf{X}}$ and the required predictive accuracy.

Truncating the chain length introduces an approximation error for turbines further downstream. Once a chain ends, the upstream wake influence is no longer propagated, so combined deficits behind multiple rows are under-represented; downstream conditions can appear overly favourable and power for more downstream turbines can be overestimated. This is visible in Figure 3-1: the reduced length chains (blue) terminate shortly past the next rotor, whereas the full length chains (grey) continue far downstream and laterally, capturing cumulative far-wake effects. However, this approximation is minimal as the dominant wake interactions are captured between adjacent upstream-downstream turbines, consistent with the far-wake treatment in [44]. If scenarios demand reliable prediction beyond the first downstream turbine, the chain length should be increased accordingly, with the expected impact on state size and computation time.

3-2-2 Number of chains

The number of chains per turbine, N_{c}, sets the lateral and vertical resolution of the OP discretisation across the rotor plane and its downstream wake. Each chain can be read as a thin column that carries wake information downstream. More chains give a finer sampling of the cross-section of the wake and of partial overlap between wakes. Fewer chains give a coarser sampling, so the density of OPs is smaller and each OP represents a larger area. Decreasing

the number of chains reduces the accuracy of how the model represents lateral shear, yaw-induced deflection, and partial-wake interactions. However, reducing N_c quickly decreases the dimension of the state $\tilde{X}$ of the prediction model, as every turbine N_T has the same number of chains N_c and all chains consist of multiple OPs N_{OP}. The MPC optimisation problem can again benefit from this reduction in dimension. We propose to investigate a sequence of increasingly coarse prediction models by progressively reducing the number of chains N_c. The results of this investigation are discussed in Section 4-3. Figure 3-2 shows (schematic) turbines with decreasing number of chains N_c, left to right, and illustrates how the cross-wake coverage is preserved while the sampling becomes progressively sparser: the leftmost turbine has many closely spaced chains and hence a high OP density; moving right, the number of chains is reduced, so each chain represents a larger slice of the wake and the downstream density of OPs becomes coarser. The lateral spread of the chains also visualises wake expansion, which is sampled more finely with many chains and more coarsely with few chains.

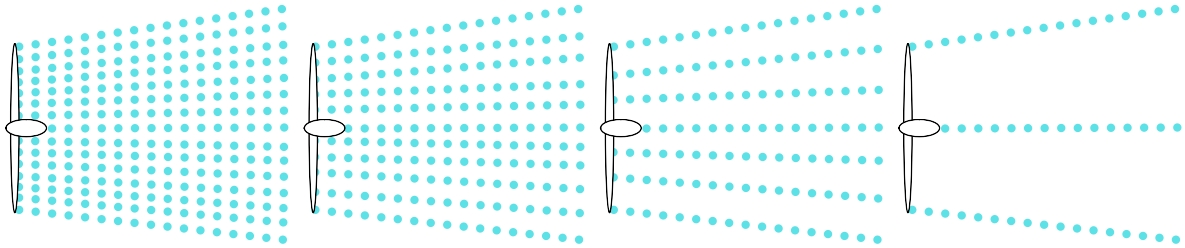

Figure 3-2: Effect of reducing the number of chains per turbine in the prediction model. Left to right, the number of chains decreases: the wake cross-section remains covered, but the OP sampling becomes coarser and each chain represents a larger portion of the flow.

The primary consequence of this coarser discretisation is on the calculation of the rotor-effective wind speed for downstream turbines. With fewer chains, the discrete OPs are sparse, forcing the model to average wake deficits over a larger area. This averaging effect is particularly problematic in partial wake scenarios, as the model loses the ability to differentiate between the high-speed and low-speed sections of a downstream rotor. This leads to inaccurate power estimations because the calculation, which is non-linear (3-16), $P \propto v_{T,i}^3$, fails to capture the disproportionate contribution of the high-speed sections versus the low-speed ones, as the "average of the cubes" is not the "cube of the average". It is important to note, however, that the underlying wake field remains an approximate Gaussian model. Increasing N_c indefinitely only provides a finer sampling of this simplified function, not a recovery of high-fidelity physics. The aim is to find a discretisation that is sufficiently fine to capture these dominant farm-level effects without incurring unnecessary computational cost.

In summary, the parameters N_{OP} and N_c serve as effective tuning knobs that govern the resolution of the wake discretisation. Adjusting these settings establishes a direct balance between the state dimension, which dictates the computational burden, and the predictive control performance. The relative performance gains and losses associated with coarser discretisations, will be explored empirically in Section 4-3.

3-3 Centralised MPC

This subsection introduces the centralised model predictive control (CMPC) formulation used to coordinate all N_T turbines using the prediction model $\tilde{f}$ (3-17), introduced in the previous section. The control objective is to track the power reference provided by the transmission system operator (TSO). At every time step k, the controller solves the following nonlinear optimisation problem over the prediction horizon N_p:

$$\min_{\mathbf{U}} \quad J(k) = \sum_{n=k}^{k+N_\mathrm{p}-1} \left(\left(\sum_{i=1}^{N_\mathrm{T}} \tilde{P}_i(n|k) \right) - P_\mathrm{ref}(k) \right)^2 \tag{3-21a}$$

$$\text{s.t.} \quad \text{for } n = k, \ldots, k+N_\mathrm{p}-1 \text{ and } i = 1, \ldots, N_\mathrm{T}:$$

$$\tilde{\mathbf{X}}(k|k) \;=\; \tilde{\mathbf{X}}(k), \tag{3-21b}$$

$$\tilde{\mathbf{X}}(n{+}1|k) \;=\; \tilde{f}\left(\tilde{\mathbf{X}}(n|k),\, \mathbf{U}(n|k),\, \mathbf{d}(n|k)\right), \tag{3-21c}$$

$$\begin{bmatrix} \tilde{v}_1(n|k) & \ldots & \tilde{v}_{N_\mathrm{T}}(n|k) \\ \tilde{P}_1(n|k) & \ldots & \tilde{P}_{N_\mathrm{T}}(n|k) \end{bmatrix} \;=\; \tilde{g}\left(\tilde{\mathbf{X}}(n|k),\, \mathbf{U}(n|k),\, \mathbf{d}(n|k)\right), \tag{3-21d}$$

$$-\gamma_\mathrm{max} \;\leq\; \gamma_i(n|k) \;\leq\; \gamma_\mathrm{max}, \tag{3-21e}$$

$$-\Delta\gamma_\mathrm{max} \;\leq\; \gamma_i(n|k) - \gamma_i(n{-}1|k) \;\leq\; \Delta\gamma_\mathrm{max}, \tag{3-21f}$$

$$a_\mathrm{min} \;\leq\; a_i(n|k) \;\leq\; a_\mathrm{max}. \tag{3-21g}$$

The cost function (3-21a) aims to minimise the deviation between total (predicted) farm power and the reference. The reference $P_\mathrm{ref}(k)$ is known at time k, but not known for future steps. Therefore, it is treated as constant across the prediction horizon. The objective couples all turbines algebraically through the total power, while the prediction model couples them dynamically through wake interactions, as upstream inputs affect downstream inflow and therefore the produced power.

The optimisation variable $\mathbf{U}$ in the optimal control problem is the system control input (3-3), stacked for all steps over the horizon:

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1(k) & \cdots & \mathbf{u}_{N_\mathrm{T}}(k) \\ \vdots & \ddots & \vdots \\ \mathbf{u}_1(k+N_\mathrm{p}-1) & \cdots & \mathbf{u}_{N_\mathrm{T}}(k+N_\mathrm{p}-1) \end{bmatrix} \in \mathbb{R}^{2N_\mathrm{p} \times N_\mathrm{T}} \tag{3-22}$$

The primary constraints of this optimisation are the system dynamics and system output, which dictate how the state and output evolves. This is enforced at every step using the nonlinear prediction model $\tilde{f}(\cdot)$ (3-21c) and output equation $\tilde{g}(\cdot)$ (3-21d)[1]. Furthermore, several constraints are imposed on the control inputs for feasible and reliable operation. First, the yaw misalignment γ_i for every turbine is bounded (3-21e). Large offsets impose asymmetric rotor loads, accelerate wear of the yaw drive and bearings, and can violate mechanical travel limits [29]. Moreover, very large yaw angles yield diminishing wake-steering benefits while sacrificing power of the yawed turbine [29]. The maximum allowed yaw angles are therefore constrained within a symmetric interval around zero. A maximum rate constraint on the yaw

[1] Note that the dynamic and output constraints (Eqs. 3-17, 3-18) could be substituted directly into the cost function (3-21a). While this would eliminate the state variables and the explicit dynamic constraints, the resulting cost function would become a highly complex, deeply nested function of the control inputs $\mathbf{U}$.

angle is imposed (3-21f), to respect physical actuator limits and to account for the physical time required for the turbine to yaw. The prediction model, for simplicity, ignores these internal actuator dynamics and assumes yaw angle changes are instantaneous. The rate constraint bridges this gap between the simplified model and physical reality. Furthermore, it is also necessary to avoid damaging changes in control actions [29]. Large step changes, which the simplified model might otherwise command, would produce high loads on the yaw bearing and gearbox and increase wear. A maximum rate constraint $\Delta\gamma_{max}$ thus keeps following yaw commands close enough that the actuators can track them smoothly within their safe operating range. A bound on the axial induction factor is introduced to keep the optimisation within physically and mechanically admissible operating points (3-21g). Constraining a_i preserves the validity of the C_T and C_P parametrisations used (Eqs. 2-9 and 2-12) and prevents unphysical or damaging set-points [44].

We propose to define the prediction horizon N_p based on the physical time required to capture the full advection of the wake from an upstream turbine to its first downstream neighbour. This is essential for the MPC controller to be able to anticipate and optimise these wake interactions. Recall, an OP is advected one step along its chain at each sampling instant Δt. Consequently, the number of prediction steps N_p directly corresponds to the downstream distance an OP travels. To ensure this coupling is fully captured, N_p must be at least as long as the physical time it takes for the wake to travel between them. This minimum time is given by N_span (3-20). A one-step safety margin is added to set the final horizon:

$$N_\mathrm{p} \;=\; N_\mathrm{span} \;+\; 1. \tag{3-23}$$

As N_span depends on the free-stream wind speed V_∞, wind direction θ, sampling time Δt, and inter-turbine spacing L_x, its maximum value must be taken if these parameters are time-varying to guarantee the interaction is always captured.

A critical component of this receding-horizon formulation is the initial state $\tilde{\mathbf{X}}(k)$ (3-21b), which is required at the start of each optimisation. As the full state is not directly measurable, we use a one-step-ahead open-loop prediction from the previous time step. The state $\tilde{\mathbf{X}}(k)$ is generated by applying the prediction model $\tilde{f}$ using the state $\tilde{\mathbf{X}}(k-1)$ and the implemented control inputs $\mathbf{U}(k-1)$ and measured disturbances $\mathbf{d}(k-1)$. While a state estimator or observer could be designed to handle model-plant mismatch, this is not within the scope of this work. After solving the optimisation, only the first optimised control inputs $\mathbf{U}[1:2,:]$ are applied to the system. The horizon then shifts and the problem is resolved at $k+1$.

The resulting optimal control problem (3-21) is a nonlinear program (NLP) which must be solved using a numerical solver. Due to the problem's inherent non-convexity, the solver is not guaranteed to find the global optimum and may instead converge to one of many local minima. Common strategies to improve the solution quality include providing an intelligent initial guess (e.g. warm start) for the optimisation variables and applying multistarting. In Chapter 4, the effect of these initialisation techniques on the controller's performance will be investigated.

3-4 Distributed MPC

The CMPC optimisation problem (3-21) can be computationally prohibitive, particularly as the number of turbines N_T increases. Therefore, we propose a DMPC approach to manage

wind-farm control at scale, which decomposes the computation into tractable subproblems while preserving a global view. It is critical, however, that this distributed controller still manages the system-wide coupling.

Having motivated a DMPC framework, the specific choice of distributed scheme is critical. As described in Section 2-3-2, DMPC approaches can be iterative or non-iterative. Both reduce local problem size by splitting the farm into per-turbine optimisations, which can improve solver times. Iterative schemes repeatedly exchange plans within each sampling interval to reach agreement on shared variables. This can enhance optimality (i.e., bring the solution closer to that of the CMPC problem), but has several drawbacks [13]. Iteration counts grow with coupling strength and problem conditioning and can therefore be problematic in this setting, which is strongly coupled through wake dynamics. Another drawback is that feasibility is often only guaranteed at convergence, so intermediate iterations may violate shared constraints. In contrast, a non-iterative scheme performs a single pass per interval, providing deterministic computation times at the cost of a suboptimality. Given the need for predictable, real-time updates, the non-iterative choice aligns better with the problem requirements. Within the non-iterative family, two update strategies are common: parallel and sequential. Parallel updates solve all subproblems simultaneously using only information from the previous interval; under strong dynamic coupling this typically yields conservative or misaligned actions and a loss in performance [52]. A sequential pass exploits causal structure: each subproblem is solved using the most recent decisions from those solved earlier in the same interval. This allows fresher information to propagate within a single non-iterative sweep, which typically reduces the suboptimality incurred by single-pass schemes while maintaining deterministic compute time [39]. For highly coupled, directionally causal flows, the sequential non-iterative strategy therefore offers a more promising accuracy-speed trade-off to the parallel alternative. In the following we propose two variants of this sequential DMPC approach, each offering a distinct trade-off between the computational complexity of the subproblems and the degree of approximation introduced.

3-4-1 Full-farm DMPC

The first approach, referred to as full-farm DMPC, splits the centralised optimisation problem (3-21) into N_T subproblems. Each subproblem i solves only for its optimal input sequence $\mathbf{u}_i$, while treating the input plans for all other turbines $\mathbf{u}_{j\neq i}$ as fixed parameters. Crucially, this formulation retains the global cost function (3-21a) and the full farm-wide dynamic model (3-21c). This allows the controller to capture the influence of upstream actions on downstream turbines, even if the optimisation problems are solved individually. The objective for each subproblem i remains the minimisation of the squared total power tracking error. However, only the constraints for turbine i are enforced on its decision variables. The resulting DMPC

subproblem for turbine i at time k is:

$$\min_{\mathbf{u_i}} \quad J_i(k) \;=\; \sum_{n=k}^{k+N_\mathrm{p}-1} \left(\left(\sum_{i=1}^{N_\mathrm{T}} \tilde{P}_i(n|k) \right) - P_\mathrm{ref}(k) \right)^2 \tag{3-24a}$$

$$\text{s.t.} \quad \text{for } n = k, \ldots, k + N_\mathrm{p} - 1 :$$

$$\tilde{\mathbf{X}}(k|k) \;=\; \tilde{\mathbf{X}}(k), \tag{3-24b}$$

$$\tilde{\mathbf{X}}(n{+}1|k) \;=\; \tilde{f}\left(\tilde{\mathbf{X}}(n|k),\, \mathbf{U}_i(n|k),\, \mathbf{d}(n|k)\right), \tag{3-24c}$$

$$\begin{bmatrix} \tilde{v}_1(n|k) & \ldots & \tilde{v}_{N_\mathrm{T}}(n|k) \\ \tilde{P}_1(n|k) & \ldots & \tilde{P}_{N_\mathrm{T}}(n|k) \end{bmatrix} \;=\; \tilde{g}\left(\tilde{\mathbf{X}}(n|k),\, \mathbf{U}_i(n|k),\, \mathbf{d}(n|k)\right), \tag{3-24d}$$

$$-\gamma_\mathrm{max} \le \gamma_i(n|k) \le \gamma_\mathrm{max}, \tag{3-24e}$$

$$-\Delta\gamma_\mathrm{max} \le \gamma_i(n|k) - \gamma_i(n{-}1|k) \le \Delta\gamma_\mathrm{max}, \tag{3-24f}$$

$$a_\mathrm{min} \le a_i(n|k) \le a_\mathrm{max}. \tag{3-24g}$$

Here, the optimisation variable $\mathbf{u}_i$ collects the yaw and axial-induction inputs for turbine i over the prediction horizon N_p (3-23). The matrix $\mathbf{U}_i(n \mid k)$ denotes the full input plan for all turbines at stage n, whose construction is dictated by the sequential sweep described next.

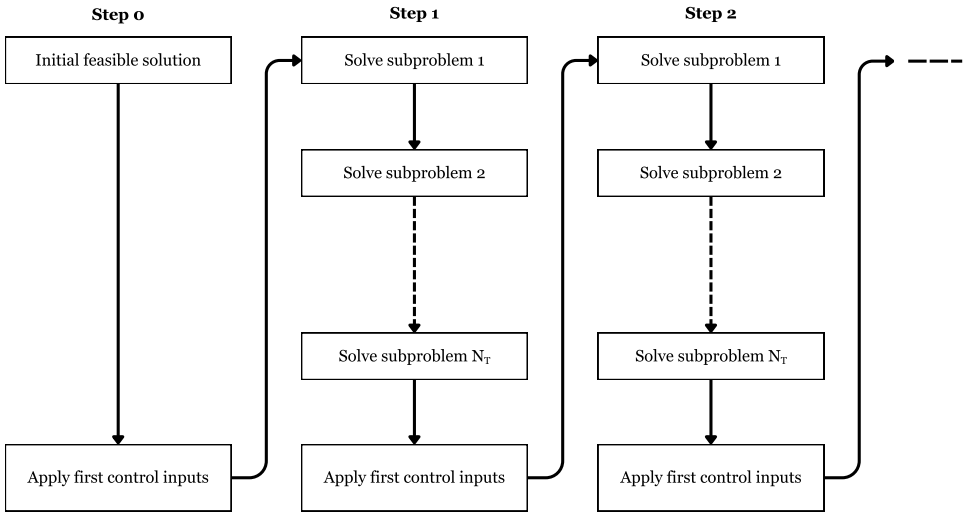

Figure 3-3: The centralised problem is decomposed into N_T per-turbine subproblems that are solved in order; after the sweep, the first control inputs are applied and the horizon shifts. Adapted from [39].

In one control interval Δt, starting at time k, N_T subproblems are executed sequentially as illustrated in Figure 3-3. Turbine 1 solves its subproblem (3-24) using the current estimated state $\tilde{\mathbf{X}}(k)$ and the shifted optimal control inputs from the previous time step $\mathbf{u}_{j\neq 1}(k-1)$ for all other turbines. These shifted plans form a prediction for the behaviour of the other turbines at time k, where the previous plan is shifted one step forward and the new final

input, $\mathbf{u}_j(k+N_\mathrm{p}-1|k)$, is set by copying the last known value, $\mathbf{u}_j(k+N_\mathrm{p}-2|k-1)$, to maintain the required horizon length. Once turbine 1 has computed its new plan, denoted as $\mathbf{u}_1^\star(k)$, that plan is passed to the next turbine in the sequence. Turbine 2 then solves its subproblem using the freshly computed plan $\mathbf{u}_1^\star(k)$ and the shifted plans for the remaining turbines $\mathbf{u}_{j>2}$. This sweep continues until the last turbine N_T has solved, using the updated plans from all $N_\mathrm{T}-1$ upstream turbines. Figure 3-4 sketches this information flow. The full input matrix $\mathbf{U}_i(k)$, while optimising turbine i, thus reads:

$$\mathbf{U}_i(k) = \begin{bmatrix} \mathbf{u}_1^\star(k|k) & \cdots & \mathbf{u}_i(k|k) & \cdots & \mathbf{u}_{N_\mathrm{T}}(k|k-1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{u}_1^\star(k+N_\mathrm{p}-2|k) & \cdots & \mathbf{u}_i(k+N_\mathrm{p}-2|k) & \cdots & \mathbf{u}_{N_\mathrm{T}}(k+N_\mathrm{p}-2|k-1) \\ \mathbf{u}_1^\star(k+N_\mathrm{p}-1|k) & \cdots & \mathbf{u}_i(k+N_\mathrm{p}-1|k) & \cdots & \mathbf{u}_{N_\mathrm{T}}(k+N_\mathrm{p}-2|k-1) \end{bmatrix}. \tag{3-25}$$

The full input matrix $\mathbf{U}_i(k)$ (3-25) serves as the input to the dynamic model constraint (3-24c). Crucially, the optimiser varies only the i-th column ($\mathbf{u}_i$) to minimise the cost, while the remaining columns ($j \neq i$) act as fixed parameters representing the neighbours' plans.

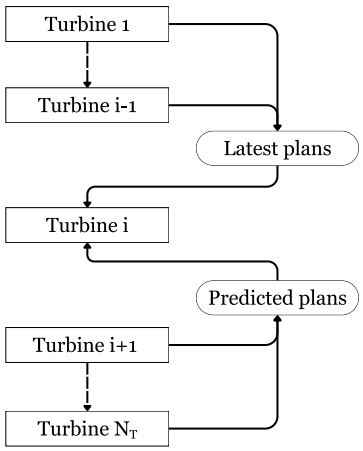

Figure 3-4: Information flow in the sequential sweep. Subproblem i uses the latest plans from already solved (upstream) subproblems and shifted plans from not-yet-solved (downstream) subproblems; once solved, its new plan is passed on to the next. Adapted from [39].

The performance of this sequential scheme is highly dependent on the solution order. The order is chosen to follow the physics: wakes travel predominantly with the inflow, so upstream actions causally influence downstream conditions. By solving from upstream to downstream, the most critical new information (the wake-generating actions) is computed first and propagated to downstream subproblems within the same sweep. A downstream-to-upstream solve, for example, would be far less effective, as downstream turbines would make decisions based on stale assumptions about the incoming wakes. In this work, the solution order is assumed to be designed by the control designer based on operating conditions (i.e., aligning the index $i = 1 \ldots N_\mathrm{T}$ with the wind direction θ). Investigating the automatic optimisation of this solution order is a potential area for future research.

This single-pass scheme introduces some suboptimality relative to the CMPC solution (3-21), since early subproblems (e.g., turbine 1) must optimise against not-yet-updated plans from downstream machines. This loss is, however, mitigated by the sequential update: information

quality improves as the sweep progresses, and the final subproblem operates with the freshest global plan available within the interval.

The approach assumes that for every turbine, at the start of each interval, all (approximated) states are available and that each subproblem has access to the full farm dynamic model $\tilde{f}(\cdot)$. This is reasonable for a homogeneous, cooperative wind farm with reliable communication and a supervisory controller that broadcasts the necessary measurements and plans at each update. In terms of implementation, the subproblems may be solved on separate devices (with plan/measurement exchange over a communication network) or on a single processor executing the sequential sweep; the communication requirement is the transmission of current measurements and updated plans between subproblems within each interval.

This decomposition lowers the decision dimension per solve from $2 \times N_{\mathrm{p}} \times N_{\mathrm{T}}$ in the CMPC problem to $2 \times N_{\mathrm{p}}$ for each DMPC subproblem. A sequential DMPC sweep, which sums the computation time of N_{T} individual solves, ideally scales linearly with the number of subproblems N_{T}. This linear scaling is typically the lowest (best) computational factor achievable for this class of distributed algorithm. However, in this full-farm formulation, each subproblem (3-24) must still evaluate the full-farm model $f(\cdot)$ and cost $J_i(k)$, which are functions of all N_{T} turbines. The computational complexity of each subproblem therefore still scales with the farm size N_{T}. Consequently, the total computational complexity of the sweep is no longer linear in N_{T}, which can become prohibitive for very large farms. The next section addresses this specific scaling challenge.

3-4-2 Subset DMPC

A drawback of the full-farm DMPC in Section 3-4-1 is that, for every turbine i, the optimiser embeds the dynamics of the entire farm, including turbines and wakes that are sufficiently far apart that their coupling is negligible within the prediction window. As the farm scales, a growing share of these dynamic calculations is redundant. To avoid this overhead, a subset-based approach is introduced in which only a locally relevant portion of the farm dynamics is enforced as an equality constraint in the MPC problem.

For each turbine i, a subset $\mathcal{S}_i \subseteq \mathcal{S}$ (with $\mathcal{S}$ the set of all turbines) is defined that contains turbine i and those neighbours with the strongest wake interactions; their states and OPs are modelled explicitly. Subsets may overlap but need not be identical. Turbines outside $\mathcal{S}_i$ are not propagated through the dynamic model during turbine i's optimisation; instead, their effects are represented by predicted rotor–effective wind speeds that act as exogenous inputs when computing their power. Similar to the sequential solution order, the specific composition of these subsets is a critical design choice; in this work, they are manually crafted to capture the dominant aerodynamic couplings.

The coordination remains sequential, ordered with the dominant flow direction (Figure 3-3 and the information flow in Figure 3-4). In the local problem for turbine i, the model $f_{\mathcal{S}_i}(\cdot)$ propagates only the states within its defined subset $\mathcal{S}_i$. To estimate the power output of turbines $j \notin \mathcal{S}_i$ while accounting for wake interactions and control inputs outside the subset, the optimiser utilises a prediction of their rotor–effective wind speed trajectories over the horizon, denoted as $\hat{\mathbf{v}}_j$. This vector $\hat{\mathbf{v}}_j(k) \in \mathbb{R}^{N_{\mathrm{p}}}$ serves as a fixed parameter profile during

the optimisation of turbine i. The resulting local optimisation for turbine i is

$$\min_{\mathbf{u_i}} \quad J_i(k) = \sum_{n=k}^{k+N_\mathrm{p}-1} \Big((\sum_{l\in\mathcal{S}_i} \tilde{P}_l(n|k)) + (\sum_{j\notin\mathcal{S}_i} \tilde{P}_j(n|k)) - P_\mathrm{ref}(k) \Big)^2 \tag{3-26a}$$

$$\text{s.t.} \quad \text{for } n = k, \dots, k+N_\mathrm{p}-1 :$$

$$\tilde{\mathbf{X}}(k|k) \;=\; \tilde{\mathbf{X}}(k), \tag{3-26b}$$

$$\tilde{\mathbf{X}}_{\mathcal{S}_i}(n{+}1|k) = \tilde{f}_{\mathcal{S}_i}\left(\tilde{\mathbf{X}}_{\mathcal{S}_i}(n|k),\, \mathbf{U}_{\mathcal{S}_i}(n|k),\, \mathbf{d}_{\mathcal{S}_i}(k)\right), \tag{3-26c}$$

$$\begin{bmatrix} \dots & \tilde{v}_l(n|k) & \dots \\ \dots & \tilde{P}_l(n|k) & \dots \end{bmatrix}_{l\in\mathcal{S}_i} \;=\; \tilde{g}_{\mathcal{S}_i}\left(\tilde{\mathbf{X}}_{\mathcal{S}_i}(n|k),\, \mathbf{U}_{\mathcal{S}_i}(n|k),\, \mathbf{d}(n|k)\right), \tag{3-26d}$$

$$\tilde{P}_{j\notin\mathcal{S}_i}(n|k) = \eta\, \tfrac{1}{2}\rho A\, \hat{v}_j(n|k)^3\, C_{P,j}(\mathbf{u}_j(n|k)), \tag{3-26e}$$

$$\gamma_\mathrm{max} \le \gamma_i(n|k) \le \gamma_\mathrm{max}, \tag{3-26f}$$

$$-\Delta\gamma_\mathrm{max} \le \gamma_i(n|k) - \gamma_i(n{-}1|k) \le \Delta\gamma_\mathrm{max}, \tag{3-26g}$$

$$a_\mathrm{min} \le a_i(n|k) \le a_\mathrm{max}. \tag{3-26h}$$

The local input matrix $\mathbf{U}_{\mathcal{S}_i}(n|k)$, required by the subset model in (3-26c) and (3-26d), contains only the inputs for turbines $l \in \mathcal{S}_i$. This matrix is formed by selecting the relevant columns from the n-th row of the full-farm matrix $\mathbf{U}_i(k)$ (3-25). The full-horizon matrix for the subset, $\mathbf{U}_{\mathcal{S}_i}(k)$, is therefore a sub-matrix of the complete farm plan, i.e., $\mathbf{U}_{\mathcal{S}_i}(k) \subseteq \mathbf{U}_i(k)$.

The external prediction trajectory $\hat{\mathbf{v}}_j(k)$ used in (3-26e) is obtained from the sequential sweep. Let $\hat{\mathbf{v}}_j^\star(\cdot)$ denote the stored version of the optimal internal wind speed sequence $\tilde{\mathbf{v}}_j$ (3-26d), computed by turbine j. Ideally, we use the fresh prediction $\hat{\mathbf{v}}_j^\star(k)$ from the current sweep. However, for downstream neighbours not yet solved, we must rely on information from the previous time step. A shifted prediction $\hat{\mathbf{v}}^\mathrm{shift}$ is constructed as:

$$\hat{v}_j^\mathrm{shift}(n|k) = \begin{cases} \hat{v}_j^\star(n|k-1) & \text{for } n = k, \dots, k+N_\mathrm{p}-2 \\ \hat{v}_j^\star(k+N_\mathrm{p}-2|k-1) & \text{for } n = k+N_\mathrm{p}-1. \end{cases} \tag{3-27}$$

Consequently, the specific input $\hat{\mathbf{v}}_j(k)$ passed to subproblem i is assembled using the most up-to-date information available:

$$\hat{\mathbf{v}}_j(k) = \begin{cases} \hat{\mathbf{v}}_j^\star(k) & \text{for } j < i \text{ (new prediction from current sweep)} \\ \hat{\mathbf{v}}_j^\mathrm{shift}(k) & \text{for } j > i \text{ (shifted prediction from } k-1). \end{cases} \tag{3-28}$$

Here, $\hat{\mathbf{v}}_j^\star(k)$ is the new rotor-effective wind speed trajectory that was generated and passed forward by the (already solved) subproblem j during the current sweep at time k.

The optimisation variable $\mathbf{u}_i$ is identical to the full-farm DMPC case (dimension $2 \times N_\mathrm{p}$). The reduction of the problem size for this approach lies in the model constraints: only the dynamics of the subset $\mathcal{S}_i$ are enforced, while turbines $j \notin \mathcal{S}_i$ contribute via the static power mapping P_j. Replacing a block of highly nonlinear state-update equations (3-24c), (3-24d), with a single algebraic equality (3-26e) per excluded turbine markedly reduces the number of propagated states and constraints per stage. The extent of this reduction depends on how small $\mathcal{S}_i$ is relative to the full set $\mathcal{S}$, and on the OP resolution chosen for the excluded machines.

This subset approach introduces an approximation because turbines outside $\mathcal{S}_i$ are not dynamically updated within subproblem i. If the subsets capture the dominant wake interactions (typically nearest upstream–downstream neighbours under the prevailing inflow), the resulting error in predicted farm power over the horizon remains limited; otherwise, deficits may be under-represented. Subset design is therefore an accuracy–speed trade-off. In practice, $\mathcal{S}_i$ should reflect farm layout, streamwise spacing, wind direction θ, and wind speed V_∞, and may be adapted over time as conditions change.

3-5 Summary

This chapter has developed the control formulation used in the remainder of this research. First, the FLORIDyn dynamics are formalised as an MPC prediction model. Two design choices structure the model: the number of chains per turbine N_c, which sets the lateral and vertical coverage, and the number of OPs per chain N_OP, which sets downstream reach. To balance model complexity and accuracy, model reductions are proposed and are further explored in the next chapter.

Next, three control architectures are formulated. The centralised MPC solves a single problem over all turbines and serves as a fully-coupled benchmark. The prediction horizon is defined such that the aerodynamic coupling between upstream and downstream turbines is captured within the prediction window. The full-farm DMPC approach decomposes the problem into per–turbine subproblems that are solved sequentially, but retains full-farm model evaluation within each step. The subset variant further reduces effort by propagating dynamics only for locally interacting machines while using passed rotor–effective wind–speed predictions for the rest. Together, these distributed formulations provide more scalable alternatives to the centralised solver by reducing the computational complexity of each subproblem, while maintaining the key wake couplings needed for control.

Chapter 4

Case study results

This chapter presents simulation results for the approach proposed in this work. First, Section 4-1 details the experimental setup. Second, the simulation model used for evaluation is introduced and validated. Third, the reduced prediction model used by the controllers is described. Section 4-4 then reports closed-loop results comparing the proposed approached with the state-of-the-art and the results for three- and six-turbine layouts. The chapter ends with a short summary.

4-1 Case study parameters

To evaluate the proposed controllers in the control of a wind farm tracking a time-varying power reference, this section describes the case study setup adopted in this work. A sampling time for the controller of $\Delta t = 13\,\mathrm{s}$ is adopted to balance temporal resolution of wake advection with computational tractability for online optimisation [44]. Using the same Δt also facilitates comparison with literature [44].

Two farm layouts are considered. The first is a compact three-turbine farm [44], shown in Figure 4-1. Let L_x and L_y denote the streamwise and spanwise spacings, respectively. Streamwise spacing is set to $L_x = 7D$. A relatively small spanwise spacing ($L_y = 1.5D$ between the two first-row turbines is used, to encourage partial-wake overlap; this increases wake interaction and makes performance differences between controllers more visible. The dominant wind direction is aligned with the positive x-axis ($\theta = 0°$), which defines the layout orientation and the reference for wind-angle perturbations. The second layout is a six-turbine farm, shown in Figure 4-2, extending the three-turbine layout by adding a third row with one additional downstream turbine and adding one extra turbine in each existing row.

To validate control approaches under a range of operating conditions, we consider different wind directions and wind speeds. For wind direction, we simulate five angles: $\theta \in \{0°, 5°, 10°, 15°, 20°\}$, defined counterclockwise from the x-axis. Only positive angles are simulated because the layout is laterally symmetric. The range is capped at 20° because larger misalignments shift wakes past downstream rotors in this farm layout. Hence, larger angles

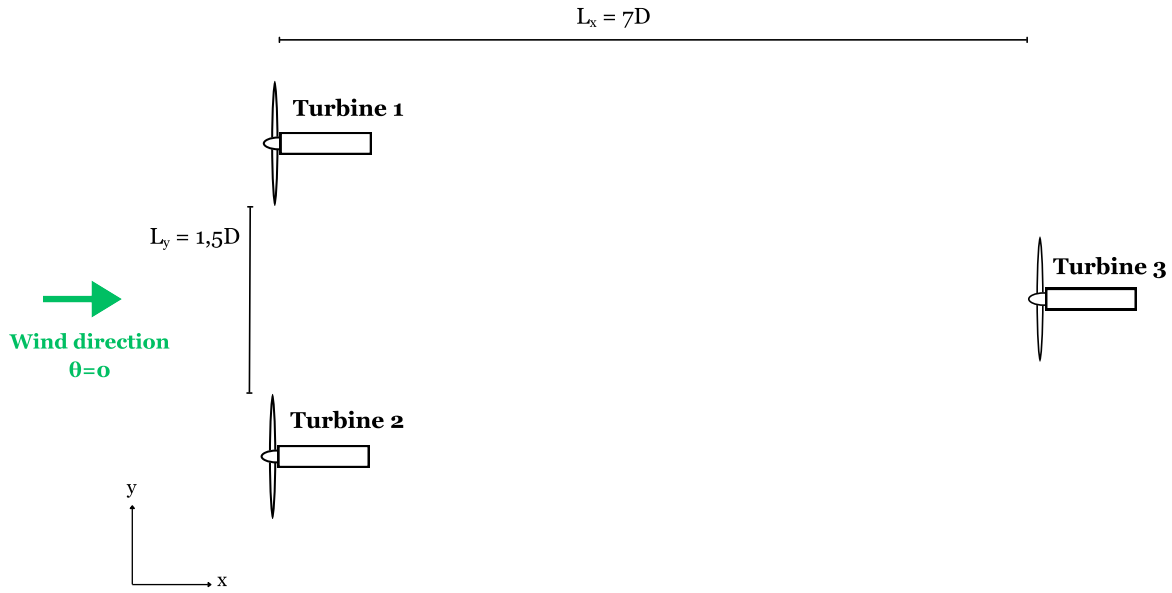

Figure 4-1: Three-turbine layout adopted from [44]. Turbines 1 and 2 are located at $x = 0$ with lateral offsets $y = \{-0.75D, +0.75D\}$. Turbine 3 is placed downstream at $x = 7D$ on the centreline. The positive x-axis aligns with the dominant wind direction ($\theta = 0°$).

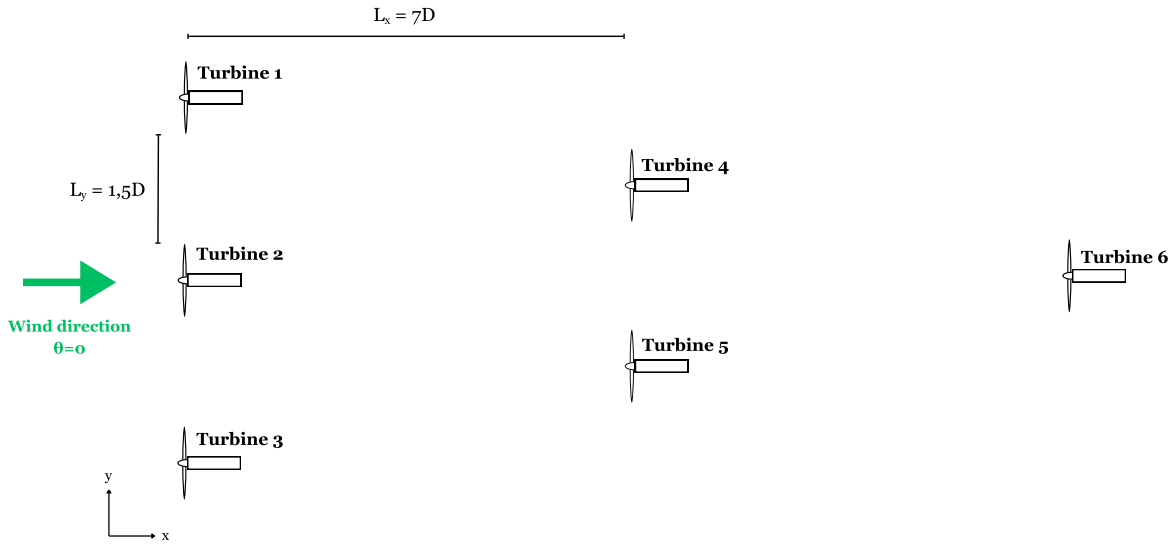

Figure 4-2: Six-turbine layout as an extension of Figure 4-1: a third row is added downstream, and one extra turbine is added to each row while retaining $L_x = 7D$ and the $\pm 0.75D$ lateral offsets in the first two rows. This extends the layout while preserving the base geometry.

are not considered; the current set already establishes whether the controller can handle inflow misalignment. Three inflow wind speeds are used, $V_\infty \in \{7, 10, 13\}\,\mathrm{m\,s^{-1}}$, with $10\,\mathrm{m\,s^{-1}}$ treated as nominal. For each simulation run, these ambient conditions are held constant for the entire duration. This aligns with the assumption made in the controller's prediction model in Section 3-2, ensuring the fixed distributed model predictive control (DMPC) structures are tested under their ideal conditions.

A simulation horizon of $T_{\text{sim}} = 480\,\text{s}$ (8 minutes) is adopted for all experiments. This duration is selected to ensure sufficient time for control actions to propagate downstream through the entire farm layout, while maintaining reasonable total computation times for the experimental campaign. To evaluate tracking performance, three distinct power reference profiles ($P_{\text{ref}}(t)$) are employed, each designed to exercise complementary aspects of the control strategy. The control objective is to ensure the total farm power accurately tracks this reference. The first profile (Reference 1) is a four-step trajectory with steps of 2 min, at 70%, 75%, 80%, and 85% of the farm's full capacity. This stepped signal creates clean transients that reveal rise time, overshoot, and settling behaviour; it is used to assess transient tracking of controllers. The second profile (Reference 2) is designed to be more representative of a real operational signal, such as a dispatch command from a transmission system operator (TSO). It consists of a high base level, set relative to the available power for the current wind speed ($b \in \{30\%, 90\%, 96\%\}$ for $V_\infty \in \{7, 10, 13\}\,\text{m s}^{-1}$, respectively), augmented with small, minute-scale adjustments. This signal tests the controllers' ability to follow a realistic, gently varying set point without overreacting to small fluctuations. The third profile (Reference 3) is a simple constant signal at 70% of rated power. All three reference signals are shown in Figure 4-3. The figure plots Reference 1, Reference 3, and Reference 2 using a base level of $b = 90\%$ (corresponding to the nominal $10\,\text{m s}^{-1}$ wind speed).

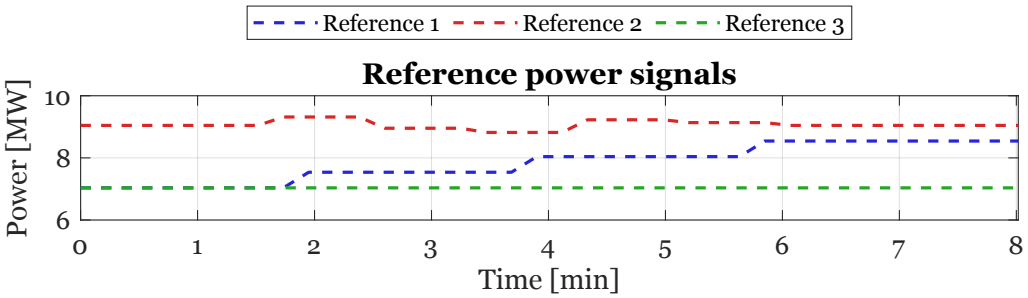

Figure 4-3: Three farm power reference profiles used for the control experiments, with a simulation time $T_{\text{sim}} = 480\,\text{s}$ (8 minutes). Profile 1 is a four-step staircase (120 s per step) at 70%, 75%, 80% and 85% of farm rated power. Profile 2 is a constant with minute-scale offsets: eight 60 s sub-intervals with percentage offsets $(0,\ 0,\ +3,\ -1,\ -2.5,\ +2,\ +1,\ 0)$ relative to a base level b set by V_∞ (shown here with $b = 90\%$). Reference 3 is a constant signal at 70% of farm rated power, used for state-of-the-art comparison.

Performance is reported using two metrics. The first metric is the relative tracking error, e_{rel} (in percent, %),

$$e_{\text{rel}}(t) = \frac{\left|P_{\text{ref}}(t) - \sum_{i=1}^{N_{\text{T}}} P_i(t)\right|}{P_{\text{ref}}(t)} \times 100 \quad [\%]. \tag{4-1}$$

This normalisation makes errors comparable across wind speeds, reference levels and farm layouts. The secondary metric is the computation time required per control step, t_{solve} (in seconds, s). To assess computational feasibility, we report the mean, standard deviation, and maximum t_{solve} observed during simulation. These performance metrics together highlight the trade-off between tracking accuracy and computational expense. To evaluate these metrics, the controllers and simulation operate under the following physical constraints and model parameters. The yaw angles are limited to $\gamma_i \in [-\pi/6, \pi/6]$ (i.e. $\pm 30°$)

because larger misalignments cause disproportionate power losses and increased structural loads, while delivering diminishing returns in wake deflection. The $\pm 30^\circ$ bounds capture the useful steering range without pushing turbines outside routine operating limits [44]. The yaw-rate limit is set to $0.25^\circ\,\mathrm{s}^{-1}$. With a sampling time Δt, the per-sample change in the yaw command is $|\Delta\gamma_{\max}| \leq 0.25^\circ\,\mathrm{s}^{-1}\,\Delta t$ ($= (0.25\pi/180)\,\Delta t$ rad). For $\Delta t = 13\,\mathrm{s}$ this gives $|\Delta\gamma_{\max}| \leq 3.25^\circ \approx 0.0567\,\mathrm{rad}$ per sample, limiting the yaw-rate per time step appropriately, to keep the yaw movements realistic and limit wear and tear [44]. The axial induction factor is constrained to $a_i \in [0.06, 0.33]$. The upper bound lies near the typical maximum of the power coefficient while remaining within the validity range of the aerodynamic parametrisations while keeping thrust loads acceptable. The small positive lower bound maintains meaningful loading (avoiding near-zero thrust), supports stable wake formation in the Gaussian model, and improves numerical conditioning [44]. Furthermore, two key controller configuration parameters are fixed for each simulation. First, the prediction horizon N_p is calculated once at the start of each run based on the scenario's specific wind speed and direction (using Eq. 3-23), ensuring the wake advection to the first downstream turbine is always captured. This horizon length is identical for all proposed controllers within a specific scenario. Second, for all DMPC simulations, the sequential solving order is fixed to follow the turbine numbering in the layout figures (e.g., Figure 4-1 and 4-2). This fixed order corresponds to the physical upstream-to-downstream flow. Initial input variables are set to $\gamma_i(0) = 0^\circ$ (no intentional steering) and $a_i(0) = 0.30$ (a common nominal axial induction). Turbine characteristics follow the IEA 3.4 MW reference machine: rotor diameter $D = 130\,\mathrm{m}$ and rated power $P_r = 3.35\,\mathrm{MW}$ [12]. Other model and flow parameters are the air density $\rho = 1.225\,\mathrm{kg\,m}^{-3}$; turbine efficiency $\eta = 0.9367$; and turbulence intensity $I = 6\%$, constant in all simulations. The power-coefficient correction is $\kappa = 0.8174$ and the power–yaw loss uses exponent $p = 2$ [44]. All symbols and values are collected in Table 4-1.

Table 4-1: Farm and controller parameters used in the control experiments.

Symbol	Description	Value	Unit
D	Rotor diameter (IEA 3.4 MW)	130	m
L_x	Streamwise spacing	$7D$	–
L_y	Spanwise spacing	$1.5D$	–
V_∞	Inflow speeds	$\{7, 10, 13\}$	$\mathrm{m\,s}^{-1}$
θ	Wind angles	$\{0, 5, 10, 15, 20\}$	deg
Δt	Sampling time	13	s
T_sim	Simulation duration	480	s
I	Turbulence intensity	6	%
ρ	Air density	1.225	$\mathrm{kg\,m}^{-3}$
η	Turbine efficiency	0.9367	–
P_r	Rated power (per turbine)	3.35	MW
κ	Power coefficient correction	0.8174	–
p	Power-yaw loss exponent	2	–
$\gamma_{\max}$	Yaw bounds	± 30	deg
$\Delta\gamma_{\max}$	Max yaw rate per step	0.0572	rad per Δt
$a_{\min}, a_{\max}$	Axial induction bounds	0.06, 0.33	–
$\gamma_i(0)$	Initial yaw	0	deg
$a_i(0)$	Initial axial induction	0.30	–

4-1-1 Optimisation implementation

For the proposed approaches, each optimal control problem is solved in MATLAB using `fmincon` with the interior point algorithm. An interior point method is adopted because it is efficient and robust for problems with a large number of inequality constraints and simple bounds. This structure is characteristic of our model predictive control (MPC) formulation, which includes per-turbine constraints (e.g., yaw, yaw rate, axial induction) at each step of the prediction horizon. As is typical in MPC, the solution from the previous time step is used to 'warm-start' the solver.

Solver settings are selected to balance speed and solution quality, and are kept identical for all controllers to ensure a fair comparison. We allow 100 interior point iterations to limit run time; too few iterations can stop before useful convergence, while many more add time with diminishing gains. Up to 5000 objective evaluations are permitted so that finite-difference derivatives are resolved; a lower cap risks early termination, a higher cap mostly increases time without clear benefit. The optimality tolerance is set to 5×10^{-1} in absolute units; a looser tolerance speeds up solves but can leave noticeable suboptimality, whereas a much tighter tolerance would cost time with negligible effect on control because the objective is measured in watts at megawatt scale. The step-size tolerance is 10^{-4}. In the solver this measures how much the optimisation variables change between two iterations. It is chosen so that, once these differences become very small, the solver terminates. If the tolerance is set too small, the solver spends time taking tiny, unproductive steps; if set too large, it may stop before the iterates have settled. This configuration gives a favourable trade-off between stable progress and computational effort. These settings were empirically validated in preliminary simulations. Table 4-2 summarises the configuration used throughout the experiments.

Table 4-2: Summary of optimisation settings (interior point). The same settings are used for all proposed nonlinear controllers.

Setting	Value
Algorithm	interior point
Interior-point terations	100
Objective evaluations	5000
Optimality tolerance	5×10^{-1}
Step-size tolerance	1×10^{-4}

To improve solution quality, a multistart scheme with $N_{\text{starts}} = 5$ is executed in parallel, for all nonlinear optimisation problems. The best feasible result is selected by the lowest cost. This value is choses (again) based on the trade-off, more multistarts potentially allow to get a better solution, but add substantial computation time. A value of $N_{\text{starts}} = 5$ is selected to strike this balance.

The solver requires an initial guess, denoted $\mathbf{u}_{i,0}$, to start the optimisation. For multistart runs, the first start is warm started from the previous found solution. Let k be the current MPC step, N_{p} the horizon length, and $\mathbf{u}^{\star}_{i,k-1}$ the previously optimal sequence for turbine i. The warm start shifts the sequence by one step and repeats the final block to maintain correct length:

$$\mathbf{u}^{(1)}_{i,0}(k) = \begin{bmatrix} \mathbf{u}^{\star}_{i,k-1}(2{:}N_{\text{p}}) \\ \mathbf{u}^{\star}_{i,k-1}(N_{\text{p}}) \end{bmatrix}. \tag{4-2}$$

This construction is used for both centralised and distributed formulations, where in the centralised case $\mathbf{u}$ contains the vectors $\mathbf{u_i}$ of all turbines i.

At the first control step $k = 1$, when no prior solution exists, a constant sequence is formed from the initial operating point. With current yaw and axial induction vectors $\gamma_i(0)$ and $a_i(0)$,

$$\mathbf{u}_{i,0}^{(s)}(1) = \mathbf{1}_{N_\mathrm{p}} \otimes \begin{bmatrix} \gamma_i(0) \\ a_i(0) \end{bmatrix} \tag{4-3}$$

For the remaining starts ($s = 2, \ldots, N_{\mathrm{starts}}$), the initial guess is formed from a single, random control input that is then held constant across the entire prediction horizon. This random input is drawn from within the admissible operating bounds and is additionally constrained to be consistent with the yaw-rate limits relative to the current yaw angle. Define the element wise clipping operator $\mathrm{clip}(x, \ell, u) = \min\{\max(x, \ell),\, u\}$, and let $r_{\gamma_i}, r_{a_i} \sim \mathcal{U}(0, 1)$:

$$\gamma_{i,0} = \mathrm{clip}(\gamma(k-1) + (2r_{\gamma_i} - 1) \circ \Delta\gamma_{\max},\ \gamma_{\min},\ \gamma_{\max}), \tag{4-4}$$

$$a_{i,0} = a_{\min} + (a_{\max} - a_{\min}) \circ r_{a_i}, \tag{4-5}$$

$$\mathbf{u}_{i,0}^{(s)} = \mathbf{1}_{N_\mathrm{p}} \otimes \begin{bmatrix} \gamma_{i,0} \\ a_{i,0} \end{bmatrix}, \qquad s = 2, \ldots, N_{\mathrm{starts}}. \tag{4-6}$$

The optimisation for the state-of-the-art mixed-integer quadratically-constrained quadratic program (MIQCQP) controller, used for comparison in Section 4-4-1, is implemented differently [44]. This controller is formulated in MATLAB using the YALMIP modelling toolbox. This formulation, being an MIQCQP, cannot be solved with an interior-point method. Instead, it requires a specialised mixed-integer solver, which in this case is the solver Gurobi. Multistart is not considered here, as its formulation as a convex mixed-integer program is guaranteed to yield a single global solution. However, the implementation does leverage a warm-start strategy, where the full solution from the previous time step is passed as an initial guess to Gurobi for the current time step. Apart from enabling this warm-start, the default Gurobi solver settings are used for the experiment.

4-2 Simulation model validation

The FLOw Redirection and Induction Dynamics (FLORIDyn) model [9] is selected as the simulation environment to represent the wind farm dynamics in this case study. Its main advantage is its computational efficiency, which permits the extensive, iterative closed-loop simulations required for controller tuning and validation, while still capturing the essential dynamic phenomena, that static models ignore. For the evaluations in this chapter, this FLORIDyn model acts as the true plant. The MPC controllers, in contrast, will use a related but more lightweight prediction model. This intentional discrepancy between the plant and the controller's internal model is valuable, as it introduces a degree of model mismatch and allows us to test the controller's robustness. Before this simulation model can be used for evaluation, its own discretisation parameters must be established. This section therefore validates the two key parameters that balance its fidelity and computational cost: the number of chains across the rotor, N_c, and the chain length expressed as the number of observation points (OPs), N_{OP}, together defining the dimension of the state $\mathbf{X}$.

To assess N_c, open-loop runs are performed on the three-turbine layout with constant inflow and constant control inputs over a five-minute horizon. This setup disables any dynamic wake-mitigation strategy so that wake advection dominates the dynamics. The open-loop experiments are carried out at $V_\infty = 10\ \mathrm{m\,s^{-1}}$ with the constant input sequence

$$\mathbf{u}_i(k) = \begin{bmatrix} \gamma_i \\ a_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} \quad \text{for all } k \text{ and all } i, \tag{4-7}$$

so that yaw angles γ_i are zero and axial induction factors a_i are constant.

Figure 4-4 shows the resulting total farm power, $P_{\mathrm{total}}(t, N_c)$, (top) and the absolute percentage difference, ϵ_{rel}, relative to a high-resolution baseline ($N_c = 250$) (bottom). Let $P_{\mathrm{baseline}}(t) = \sum_{i=1}^{N_T} P_i(N_c = 250)$ be the power from this baseline simulation. The relative difference is then given by:

$$\epsilon_{\mathrm{rel}}(t) = \frac{\left|\sum_{i=1}^{N_T} P_i(t) - P_{\mathrm{baseline}}(t)\right|}{P_{\mathrm{baseline}}(t)} \times 100 \quad [\%]. \tag{4-8}$$

A distinct power drop is observed in the top plot of Figure 4-4 around $t \approx 1.5$ min, matching the wake travel time over the streamwise spacing $L_x = 7D$ to the downstream turbine:

$$t = \frac{L_x}{V_\infty} = \frac{7D}{V_\infty} = \frac{7 \cdot 130}{10} = 91\ \mathrm{s} \approx 1.5\ \mathrm{min}. \tag{4-9}$$

This agreement indicates that wake advection is resolved at the chosen sampling and that the observed transients are physical. The figure then makes the overall behaviour clear: in the top plot, all choices of N_c show the same dip at $t \approx 1.5$ min when the wakes reach the last turbine, after which the outputs settle and match closely. In the bottom plot, the absolute percentage difference versus the $N_c = 250$ baseline peaks at wake arrival and then flattens. The $N_c = 60, 100, 200$ cases are essentially indistinguishable (sub-percent after the transient), $N_c = 50$ remains within about 2.5% percent, and $N_c = 40$ shows a small but visible bias of a few percent. Increasing N_c refines the lateral–vertical sampling of the wake, but this also increases the state dimension advected downstream and the number of OP updates per step, adding to the computational load of the simulation environment. Given the small differences in total farm power, $N_c = 50$ is adopted as a baseline, consistent with [9, 44], as a practical balance between fidelity and adding computational overhead.

Note that the wake field remains an approximate Gaussian model sampled at discrete OPs; increasing N_c indefinitely does not converge to a perfect physical model. Rather, the choice of N_c is based on capturing farm-level effects sufficiently, without unnecessary computation.

The second parameter is the chain length N_{OP}. This must be large enough for OPs from an upstream rotor to advect to the most downstream rotor, covering all layouts and operating conditions. The most demanding layout is the six-turbine farm, which spans two row spacings ($2L_x = 14D$). The minimum number of OPs required to cover this distance, N_{span} (as defined in Eq.3-20), is determined by the most conservative (i.e., slowest advection) case. This corresponds to the lowest wind speed and highest wind angle: $V_\infty = 7\ \mathrm{m\,s^{-1}}$ and $\theta = 20°$. With $D = 130$ m and $\Delta t = 13$ s, the calculation is:

$$N_{\mathrm{span}} = \left\lceil \frac{2L_x}{V_\infty\, \Delta t\, \cos\theta} \right\rceil = \left\lceil \frac{14D}{V_\infty\, \Delta t\, \cos\theta} \right\rceil = \left\lceil \frac{14 \cdot 130}{7 \cdot 13 \cdot \cos 20°} \right\rceil = 22. \tag{4-10}$$

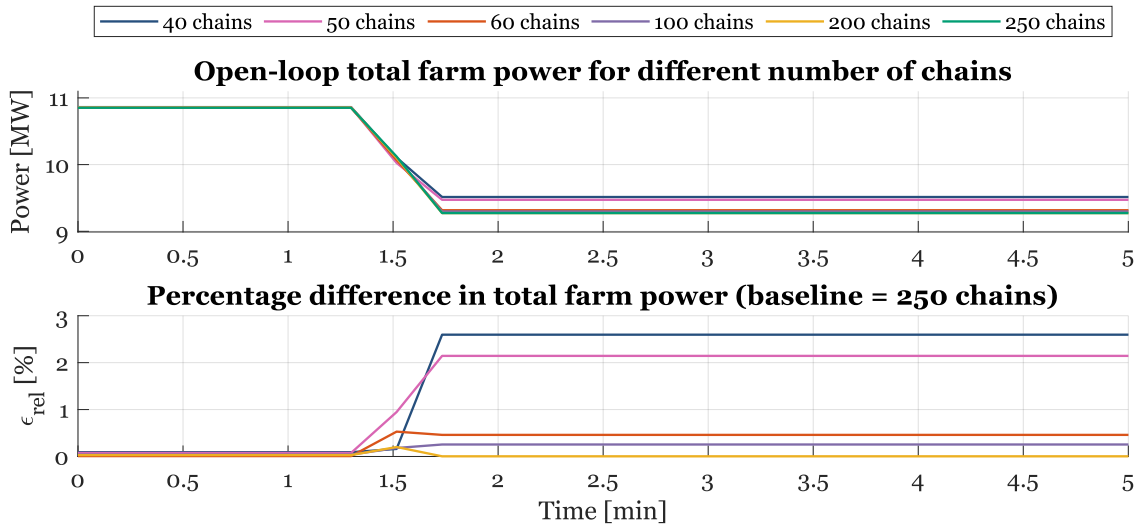

Figure 4-4: Open-loop model run for different number of chains N_c on the three-turbine layout. Top: total farm power for several N_c. Bottom: absolute percentage difference relative to the baseline of $N_c = 250$. All runs use constant inputs $(\gamma_i, a_i) = (0, 0.3)$, $\theta = 0$ and $V_\infty = 10\ \mathrm{m\,s^{-1}}$ over a five-minute horizon.

A safety margin is added, leading to the final simulation model parameters:

$$N_{OP} = 30, \qquad N_c = 50. \tag{4-11}$$

These choices keep computation reasonable for the simulation studies, are consistent with the literature [9, 44], and reflect the diminishing returns observed beyond $N_c = 50$ in Figure 4-4.

4-3 Results of prediction model approximations

This section defines and validates the prediction model used within the MPC controllers, implementing the reduced-order modelling theory proposed in Section 3-2. The central challenge is balancing the model's predictive accuracy against its computational cost. This trade-off is more critical here than in the previous section: while the simulation model's cost only affects offline simulation speed, the prediction model's complexity directly impacts the controller's solve time, t_{solve}, and thus its real-time feasibility. This involves defining the two key discretisation parameters for the prediction model, N_{OP} and N_c; we first fix N_{OP} based on our theory and then explore the impact of N_c.

Following the reduced-order modelling strategy proposed in Section 3-2, we first fix the prediction model's chain length, N_{OP}. This requires calculating the number of OPs needed to span a single row spacing ($L_x = 7D$) under the most conservative conditions ($V_\infty = 7\ \mathrm{m\,s^{-1}}$, $\theta = 20°$), using Eq. 3-20:

$$N_{span} = \left\lceil \frac{L_x}{V_\infty\, \Delta t\, \cos\theta} \right\rceil = \left\lceil \frac{7 \cdot 130}{7 \cdot 13 \cdot \cos 20°} \right\rceil = \lceil 10.64 \rceil = 11. \tag{4-12}$$

A small margin is added to this minimum to accommodate transient accelerations and model–plant mismatch. The final choice for the prediction model chain length is therefore fixed at:

$$N_{\mathrm{OP}} = 13, \tag{4-13}$$

which covers the first downstream interaction with buffer while keeping the chain short for computational efficiency. This also ensures that, for any wind angle in the case study, the number of OPs reachable within the prediction horizon N_{p} (see Eq. 3-23) does not exceed the transport capacity of the prediction model. Note that truncating the chain at $N_{\mathrm{OP}} = 13$ introduces approximation error in layouts with multiple downstream rows, since information farther downstream is not represented. In the present case studies this affects only the six-turbine layout; the three-turbine layout is unaffected.

With N_{OP} now fixed, this section explores the impact of the second parameter, the number of chains N_{c}, on the closed-loop performance and computational effort. To investigate this, experiments using the centralised model predictive control (CMPC) formulation (Eq.3-21) are conducted on the three-turbine layout with $V_\infty = 10\ \mathrm{m\,s^{-1}}$, $\theta = 0°$, and $N_{\mathrm{starts}} = 5$ multistarts. The controller tracks the step reference signal (Reference 1 from Figure 4-3). Six values for N_{c} in the prediction model are compared: $N_{\mathrm{c}} \in \{5, 10, 20, 30, 40, 50\}$.

The results are presented in Figure 4-5, Figure 4-6, and Table 4-3. The time series in Figure 4-5 shows a clear performance trend. Models with few chains ($N_{\mathrm{c}} \leq 20$) exhibit large, oscillatory tracking errors after the wake propagates to the downstream turbines. In contrast, models with $N_{\mathrm{c}} \geq 30$ demonstrate significantly improved tracking, with errors that are smaller and more stable. Visually, the performance curves for $N_{\mathrm{c}} = 30, 40$, and 50 are clustered closely together, suggesting that further increases in N_{c} beyond 30 yield diminishing returns in tracking accuracy.

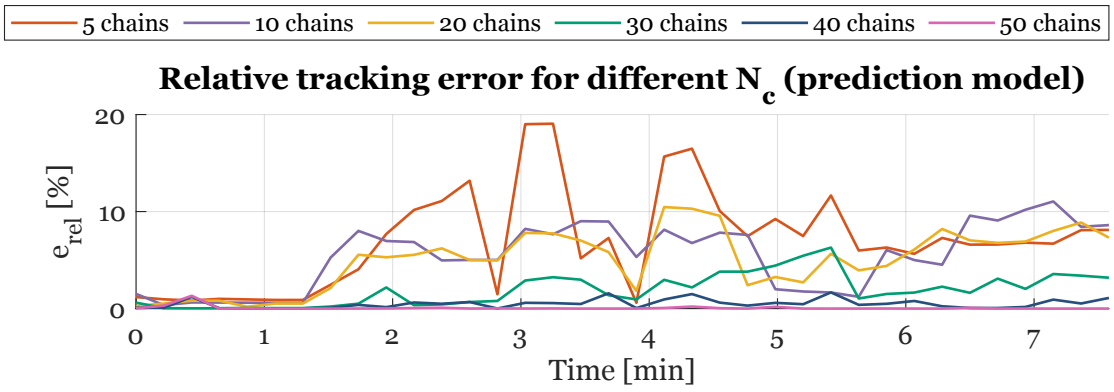

Figure 4-5: Time series of farm-level relative tracking error e_{rel} for different numbers of chains N_c in the prediction model. The test uses the CMPC controller with multistarts on the three-turbine layout, with $\theta = 0°$, $V_\infty = 10\ \mathrm{m\,s^{-1}}$, and tracking the step reference (Reference 1, Fig.4-3).

Figure 4-6 and Table 4-3 quantify this relationship. Figure 4-6 explicitly plots the trade-off, showing mean tracking error (red, dashed line) against mean solver time (blue, solid line). Two features stand out. First, $N_{\mathrm{c}} = 10$ and 20 yield very similar mean relative errors,

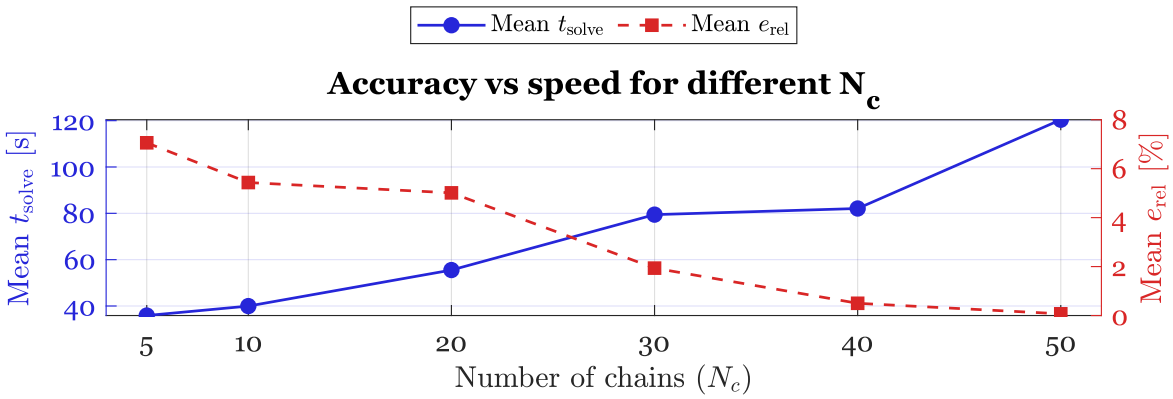

Figure 4-6: Trade-off between tracking accuracy and computational speed as a function of the number of chains N_c in the prediction model. The left y-axis (blue, solid line) plots the mean solver time per time step t_{solve}. The right y-axis (red, dashed line) plots the mean relative tracking error e_{rel}. Data is from the CMPC multistart experiment shown in Figure 4-5.

whereas moving to $N_c = 30$ reduces the mean error markedly. Second, both $N_c = 40$ and 50 achieve sub-1% mean error, yet their average solver times per step differ substantially. In short, more chains give finer wake sampling and lower tracking error, but at a steadily higher per-step solve time. Furthermore, Table 4-3 reveals that this higher mean computation time is also accompanied by a significant increase in computational volatility. Both the standard deviation and the maximum t_{solve} grow substantially with N_c, making controllers with very high N_c values less predictable. These results clearly illustrate the fundamental trade-off between model fidelity and computational cost.

Table 4-3: Mean relative tracking error e_{rel} and solver time t_{solve} statistics per time step for different N_c in the prediction model. The mean, standard deviation (Std. dev.), and maximum (Max) values for t_{solve} are reported. The most preferable result in each column is highlighted in bold.

N_c	Mean e_{rel} [%]	Mean t_{solve} [s]	Std. dev. [s]	Max [s]
5	7.058	**35.89**	**37.08**	**108.36**
10	5.434	39.96	45.16	134.74
20	5.012	55.52	58.83	180.25
30	1.937	79.43	74.85	196.63
40	0.500	82.08	71.70	240.14
50	**0.070**	120.40	113.29	406.01

4-4 MPC results

This section presents case study results for the three proposed controllers: centralised MPC (3-21), full-farm DMPC (3-24), and subset DMPC (3-26), each run with and without multi-starts. For the three-turbine layout, subsections 4-4-1 and 4-4-2 report, respectively, a com-

parison with the state-of-the-art MIQCQP controller [44] and a comprehensive study over wind speeds and directions. In this three-turbine setting the full-farm and subset distributed formulations are equivalent in practice, so only the full-farm implementation is reported and referred to as DMPC in the corresponding Figures. Subsection 4-4-3 covers the six-turbine layout, including the subset DMPC approach.

For the following case studies, the prediction model with $N_c = 50$ is selected. Since the distributed controllers inherently introduce approximations, using a more accurate prediction model ensures that performance differences are attributable to the controller design, not to a coarse model discretisation. All other simulation and prediction model settings follow Sections 4-3 and 4-2.

4-4-1 Comparison with state-of-the-art

In this section the proposed controllers are compared with a state-of-the-art formulation from [44]. This approach uses a similar Gaussian wake model with observation points, but the non-linear optimisation is reformulated through a set of approximations into a MIQCQP. The experiment uses the three-turbine layout with a constant reference at 70% of full farm capacity (Reference 3 from Figure 4-3), aligned wind direction $\theta = 0°$, free-stream speed $V_\infty = 10\ \mathrm{m\,s^{-1}}$, and a simulation time of three minutes.

Figure 4-7 shows four time series plots. The top left panel, showing the farm-level relative tracking error e_{rel}, reveals the most significant difference. The MIQCQP controller exhibits a very large error spike of almost 25% while the upstream wakes are travelling to the most downstream turbine (until $t \approx 1.5\,\mathrm{min}$). After this transient, its error reduces but remains higher than some of the other controllers. In contrast, both DMPC approaches (with and without multistart) maintain exceptionally small errors (below 0.1%) throughout. The CMPC controller with multistarts also settles to a very low error, while the single-start CMPC takes longer to settle. The poor performance of the MIQCQP can be attributed to two main factors. First, its underlying prediction model is a more simplified representation of the dynamics, analogous to using only a single chain in the FLORIDyn model, which leads to a loss of information, especially regarding wake shape and overlap. Second, the reformulation of the nonlinear program (NLP) into an MIQCQP introduces further approximations, which degrades predictive accuracy. These results indicate that the proposed nonlinear CMPC and DMPC formulations capture the wake interactions more reliably. This even leads to significant steady-state error in closed-loop, as observed during the first 1.5 minute of simulation in Figure 4-7.

The top right plot confirms the cause of the error spike, showing a drop in wind speed at turbine 3 for all controllers at $t \approx 1.5\,\mathrm{min}$ as the wake arrives. The control inputs of turbine 1, shown in the bottom panels, effect the rotor-effective wind speed at turbine 3. for the MIQCQP case, the yaw angle of turbine 1 drifts to more negative values over time, which deflects its wake towards turbine 3, resulting in a lower wind speed v_3. This suggests that, for the present reference signal, the farm can meet the target without beneficial inter-turbine cooperation.

Table 4-4 quantifies these findings. The MIQCQP controller yields the largest mean tracking error by a wide margin. Its mean solve time is higher than the single-start approaches, but stays below the multistart approaches. However, it is worth noting that while this average

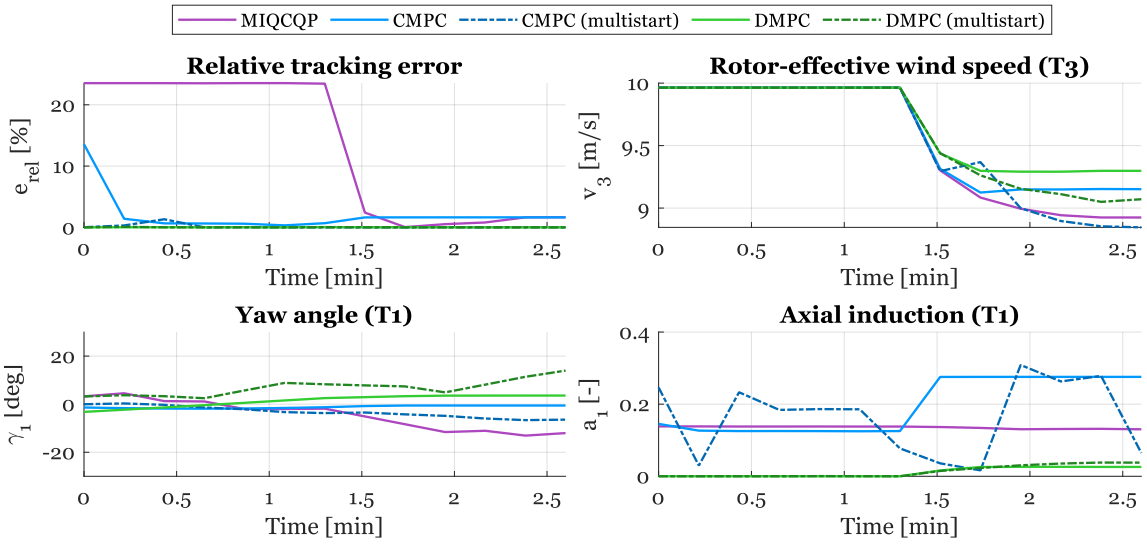

Figure 4-7: Time series comparison with the MIQCQP state-of-the-art. Top left: farm-level relative tracking error e_{rel}. Top right: rotor-effective wind speed v_3. Bottom: control inputs (yaw angle γ_1 and axial induction a_1) for Turbine 1. The test uses the three-turbine layout (Fig.4-1), with $\theta = 0°$, $V_\infty = 10\ \text{m}\,\text{s}^{-1}$, and tracking the constant reference (Reference 3, Fig. 4-3). The legend applies to all plots.

time may seem manageable, solver times for other tested simulations were highly volatile and even exceed multiple hours. This volatility, where solve times can grow unpredictably and non-linearly with small changes in problem setup, is a known drawback of mixed-integer programming. In contrast, all proposed controllers achieve significantly lower mean tracking errors. The DMPC variants are particularly effective, achieving the best tracking accuracy in this test.

Table 4-4: Mean relative tracking error e_{rel} and solver time t_{solve} statistics per time step for the state-of-the-art comparison. The mean, standard deviation (Std. dev.), and maximum (Max) values for t_{solve} are reported. The most preferable result in each column is highlighted in bold. DMPC solver times represent the total (sum) time for the sequential solve at each timestep.

Controller	Mean e_{rel} [%]	Mean t_{solve} [s]	Std. dev. [s]	Max [s]
MIQCQP	13.207	20.58	24.33	78.33
CMPC	2.126	**4.29**	**5.16**	**22.254**
CMPC (multistart)	0.135	57.37	56.22	187.18
DMPC	0.013	19.32	10.08	43.87
DMPC (multistart)	**0.012**	49.10	23.21	119.50

In summary, the proposed controllers significantly outperform the MIQCQP state-of-the-art approach in tracking accuracy. The DMPC variants are particularly effective, achieving the lowest errors in this comparison. Furthermore, this benchmark was run in a restricted experiment, distinct from the main case-study setup, because the MIQCQP controller proved sensitive to initial conditions and reference profiles. Specifically, often the mixed-integer solver falsely returned infeasible for initial conditions which are known to be feasible. Additionally,

the controller was observed to report infeasibility mid-way through simulations, indicating either false infeasibility again, or that control inputs caused the system to diverge. These observations are likely due to numerical instability or poor numerical conditioning of the MIQCQP optimisation problem, due to the extensive model reformulations. Therefore the simulation shown in this section was selected to permit a complete MIQCQP run. This sensitivity, however, constrains its practical applicability and is why the MIQCQP controller is not included in the more rigorous, varying-condition tests that follow.

4-4-2 Three-turbine layout

This subsection evaluates the CMPC and DMPC controllers, both with and without multi-starts, on the three-turbine farm under eight operating scenarios. DMPC in this case refers to the full-farm DMPC formulation (3-24). We vary wind direction $\theta \in \{0°, 5°, 10°, 15°, 20°\}$ using the step reference (Reference 1 in Figure 4-3), and free-stream speed $V_\infty \in \{7, 10, 13\}\,\mathrm{m\,s^{-1}}$ using the perturbed constant reference (Reference 2 in Figure 4-3). This perturbed reference is designed to rigorously test the controllers' tracking performance under varying wind speeds, as each reference for the specific wind speed is intentionally challenging to reach and maintain. The subset DMPC approach is not reported in this section, since it is equivalent in practice to the full-farm approach with this farm layout.

Two representative time series are shown: one for $V_\infty = 7\,\mathrm{m\,s^{-1}}$ in Figure 4-8 and one for $\theta = 15°$ in Figure 4-9. Figure 4-8 presents the farm-level relative error e_{rel}, rotor-effective wind speed of the downstream turbine v_3, and the control inputs (yaw angle γ_1 and axial-induction factor a_1) for turbine 1. Figure 4-9 also shows the farm-level e_{rel} and rotor-effective wind speed v_3, but also includes the control inputs for all three turbines. In both figures, all controllers are plotted together, with multistart variants indicated by the dotted style. A quantitative summary over all eight experiments, including mean relative error and solver-time statistics per controller, is provided in Table 4-5. A more detailed version, which also includes the per-turbine solver time statistics for the DMPC controllers, is available in Table B-2 in Appendix B.

Figure 4-8 compares the controllers for the perturbed constant reference at $V_\infty = 7\,\mathrm{m\,s^{-1}}$ and $\theta = 0°$. In the top-left panel, the farm-level relative tracking error exhibits a piecewise constant profile that mirrors the step changes in the reference signal, as can be seen in reference 2 in Figure 4-3. The multistart variants (CMPC and DMPC with dotted lines) track each segment with only minor error, whereas the single–start runs settle with noticeably larger offsets, particularly after the change around $t \approx 4\,\mathrm{min}$. This highlights the benefit of multistarting for nonlinear problems, as exploring several starting points makes it more likely to find a better quality solution.

The control inputs for turbine 1 (bottom row Figure 4-8) show why the multistart variants close the gap. Across both yaw and axial–induction inputs the dotted traces fluctuate more, indicating that the optimiser explores a wider part of the decision space and finds input sequences that better compensate for the changing operating point. By contrast, the single–start schemes tend to converge to the closest available local optimum, and from $t \gtrsim 4\,\mathrm{min}$ this suboptimal control strategy is insufficient to reduce the new error plateau.

The top–right panel reports the rotor–effective wind speed v_3 at turbine 3. It remains approximately constant initially and then drops when the upstream wakes arrive at the downstream

rotor. Here, this interaction occurs at about $t \approx 2\,\text{min}$ (rather than the earlier 1.5 min observed in faster–wind cases), because the lower free–stream speed delays the transport of wake effects downstream. The gradual increase for DMPC thereafter coincides with the steadily rising yaw of turbine 1 in the bottom–left panel, consistent with wake deflection improving the downstream inflow.

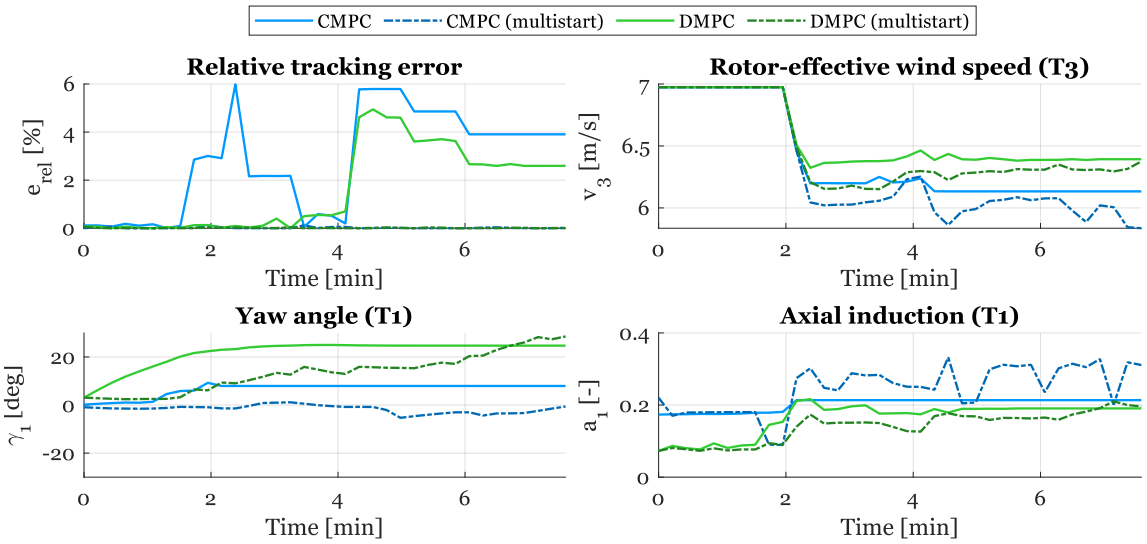

Figure 4-8: Time series comparison of all proposed controllers. Top left: farm-level relative tracking error e_{rel}. Top right: rotor-effective wind speed v_3. Bottom: control inputs (yaw angle γ_1 and axial induction a_1) for Turbine 1. The test uses the three-turbine layout (Fig.4-1), with $\theta = 0°$, $V_\infty = 7\ \text{m s}^{-1}$, and tracking the perturbed constant reference (Reference 2, Fig. 4-3). The legend applies to all plots.

Figure 4-9 reports the step–reference experiment at $\theta = 15°$ and $V_\infty = 10\,\text{m s}^{-1}$, following reference 1 in Figure 4-3. In the top–left panel the CMPC run struggles between $t \approx 2$–$4\,\text{min}$, accumulating a sustained error during one step plateau. By contrast with Figure 4-8, both distributed schemes maintain very small tracking errors, with only brief spikes at the instants when the reference changes (around $t \approx 2$ and $t \approx 6\,\text{min}$). The absence of a large error transient when the wake from the upstream machines reaches turbine 3, suggests that all controllers anticipate the wake interaction and compensate for it in advance.

The top–right panel shows the rotor–effective wind speed at turbine 3. It stays flat initially and then drops shortly after $t \approx 1.5\,\text{min}$, consistent with the wake of the upstream row arriving a little later than in the aligned case because the free-wind direction increases the path length to the downstream rotor. As in Figure 4-8, multistart runs exhibit more variability in the inputs (bottom row), however the spread is milder here. A clean step reference demands decisive but short-lived actuation at the step boundaries, whereas the perturbed constant signal (Figure 4-8) forces continuous small adjustments and thus larger input variation.

The yaw and axial-induction traces in the bottom plots clearly reveal a characteristic feature of the DMPC scheme. Because turbines are optimised sequentially, the first turbine in the sequence (T1) tends to absorb a larger share of the corrective action, particularly at the beginning of the horizon. This explains the near-zero axial-induction for T1 at start-up and its pronounced adjustments compared to the other turbines; DMPC reduces the power capture

of this upstream unit while the remaining machines are still close to their initial settings. As the power reference increases, a_1 increases in steps. Under centralised solving (CMPC), by contrast, the optimiser adjusts all turbines simultaneously, so the required effort is typically spread more evenly across the farm, and the per-turbine inputs appear less extreme. This illustrates a fundamental difference in behaviour, with sequential DMPC concentrating control action on the upstream machine, which is unfavourable from a mechanical wear standpoint, while CMPC distributes the effort more evenly across the farm.

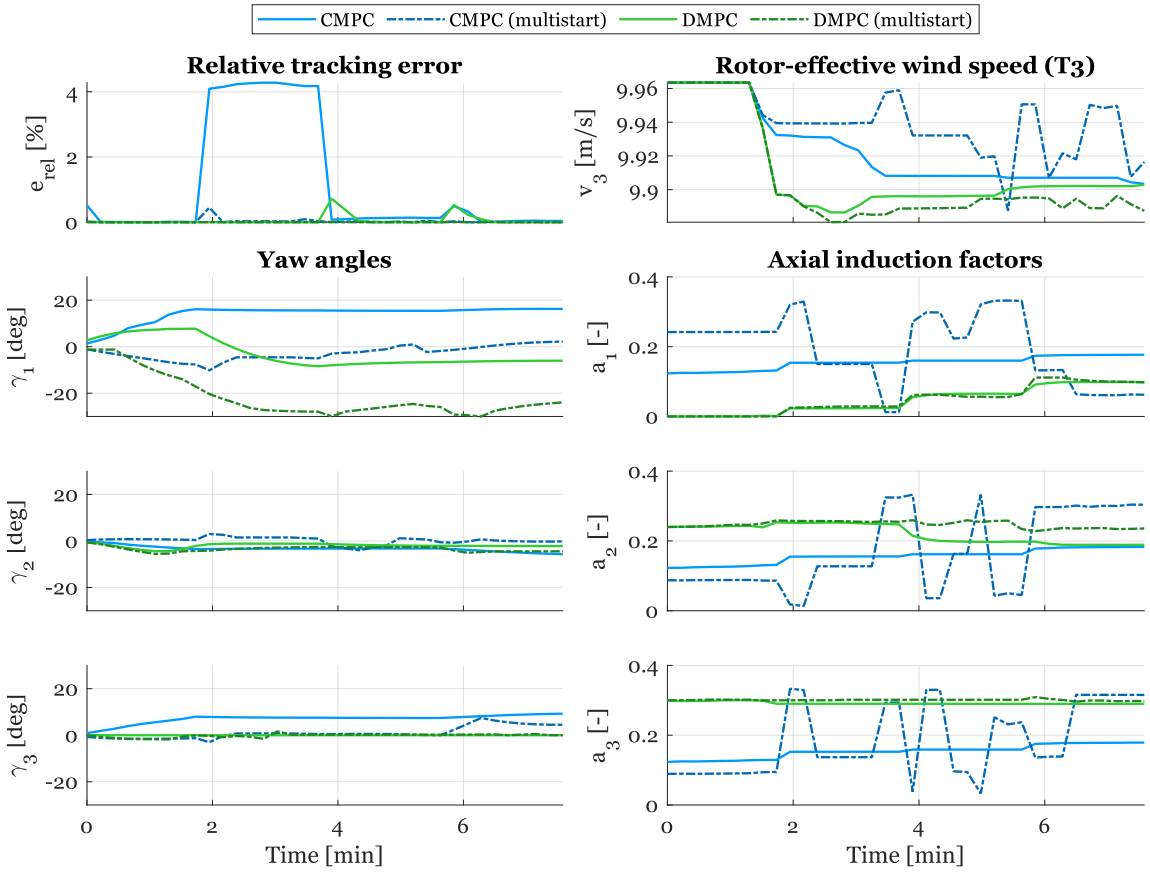

Figure 4-9: Time series comparison of all proposed controllers. Top left: farm-level relative tracking error e_{rel}. Top right: rotor-effective wind speed v_3. Bottom rows: control inputs (yaw angles γ_i and axial induction factors a_i) for all turbines. The test uses the three-turbine layout (Fig.4-1), with $\theta = 15°$, $V_\infty = 10\ \text{m s}^{-1}$, and tracking the step reference (Reference 1, Fig. 4-3). The legend applies to all plots.

A summary and trade-off visualisation for all eight experiments are presented in Table 4-5 and Figure 4-10, respectively. The table provides aggregate statistics on tracking error and solver times, with the most favourable result in each column bolded. The scatter plot in Fig.4-10 visualises the cost-versus-accuracy trade-off, where each point represents one experiment, plotting mean relative error e_{rel} against mean solver time t_{solve}.

Clear trends emerge when analysing the single-start schemes (the left-hand cluster in Figure 4-10). While the single-start CMPC appraoch achieves the lowest mean solve time, Table 4-5 shows its tracking error is the highest. The single-start DMPC approach achieves

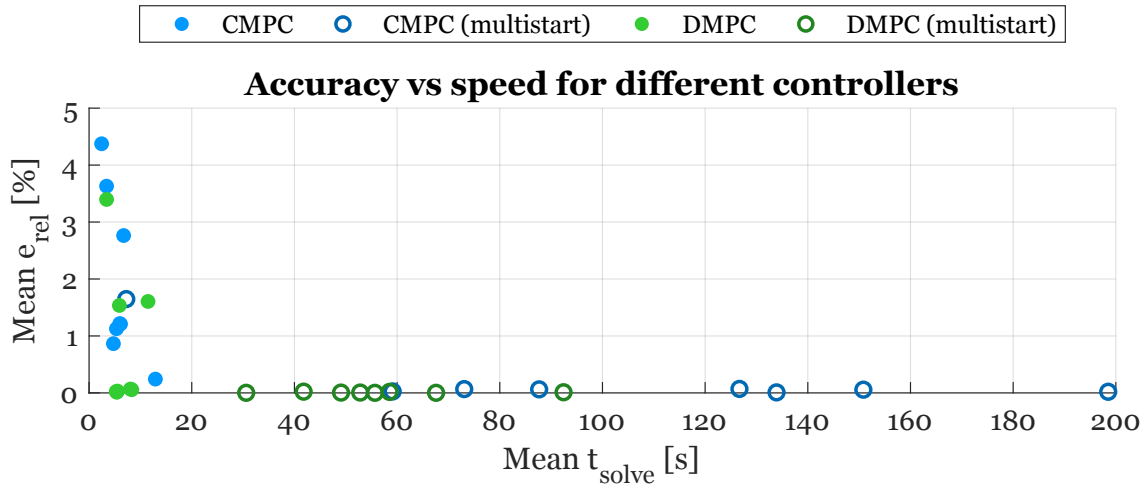

Figure 4-10: Scatter plot illustrating the accuracy-speed trade-off, showing mean relative tracking error e_{rel} versus mean solver time t_{solve} for all proposed controllers. Each point represents the mean result of one of the eight experiments conducted on the three-turbine layout (Fig. 4-1).

a significantly lower mean error for a comparable mean solve time. Crucially, DMPC also demonstrates more predictable runtimes, with the lowest standard deviation and smallest maximum solver time among all controllers. The scatter plot confirms this: the green DMPC points are consistently below (lower error) the blue CMPC points in this low-computation cluster, indicating a clear and systematic advantage for the distributed approach.

Introducing multistarts improves accuracy but at a substantial computational cost, as seen by the points shifting to the right in Figure 4-10. The CMPC with multistarts suffers from extremely high and highly variable solve times, forming a scattered cloud at the far right of the plot. In sharp contrast, the DMPC with multistarts achieves the best tracking accuracy overall (bolded in the table) at roughly half the mean computational cost and with much lower variability than its centralised counterpart. This is visualised by the tight, low-error green cluster, which is well to the left of the blue multistart points. Furthermore, the per-turbine statistics in the detailed table in Appendix B (Table B-2) reveal that the sequential solve tends to place a heavier computational burden on the upstream turbines, though this effect is less pronounced in the multistart case.

Overall, three distinct patterns are evident. First, distributing the optimisation provides superior tracking accuracy and more consistent runtimes than the centralised approach, even without multistarts. Second, while multistarts effectively eliminate residual errors, this benefit comes at a high computational price; the distributed variant offers a vastly better cost-benefit ratio for this, avoiding the extreme latencies of the centralised method. Third, a clear trade-off exists for real-time application. The single-start DMPC already delivers sub-percent mean errors with mean solve times well within the sampling-time budget (e.g., < 13 s) [1]. The multistart variants, while more accurate, both have mean solve times far exceeding

[1]Note that all absolute solver times are illustrative, as they are dependent on the specific hardware and software implementation used for this study. Actual performance on a dedicated real-time controller would likely differ.

this budget, making DMPC with multistarts a choice for systems where near-perfect tracking is critical and a larger time budget is available.

Table 4-5: Per-controller summary of mean relative tracking error e_{rel} and solver time t_{solve} statistics per time step, aggregated across all three-turbine experiments. The mean, standard deviation (Std. dev.), and maximum (Max) values for t_{solve} are reported. The most preferable result in each column is highlighted in bold. DMPC solver times represent the total (sum) time for the sequential solve at each timestep.

Controller	Mean e_{rel} [%]	Mean t_{solve} [s]	Std. dev. [s]	Max [s]
CMPC	1.928	**5.96**	8.67	77.38
CMPC (multistart)	0.245	104.63	110.78	424.73
DMPC	0.841	6.67	**5.52**	**42.94**
DMPC (multistart)	**0.009**	56.06	31.67	233.73

4-4-3 Six–turbine layout

In this section, we apply the controllers to the six-turbine farm. We consider two experiments, both with aligned inflow ($\theta = 0°$) and free-stream speed $V_\infty = 10\ \mathrm{m\,s^{-1}}$: one using the step reference (Reference 1 in Figure 4-3) and one using the perturbed constant reference (Reference 2 in Figure 4-3). The time-series behaviour is analysed for the step-reference case in Figures 4-11, 4-13 and 4-12, while the combined results from both experiments are presented in Table 4-7 and the scatter plot (Figure 4-14). In the figures that follow, the centralised controller (3-21) is labelled 'CMPC', the full-farm distributed controller (3-24) is labelled 'DMPC', and the subset controller (3-26) is labelled 'DMPC subset'.

Alongside the centralised and (full-farm) distributed formulations, we also evaluate the subset DMPC approach in which each turbine is optimised using only a carefully selected group of neighbours. For this case study, these subsets are manually selected to capture the strongest, direction-dependent wake interactions, specifically for the $\theta = 0°$ inflow condition. The selection logic for a turbine i's subset ($\mathcal{S}_i$) is as follows: turbine i itself is always included; then, the turbine(s) directly downstream of i affected by its wake are added; and finally, any upstream turbines whose wakes impact that downstream set are also included. The resulting subsets are listed in Table 4-6 (e.g., since turbine 6 has no downstream neighbours, $\mathcal{S}_6$ consists only of turbine 6). It is crucial to highlight that the performance of this subset approach is highly dependent on this specific, manually-chosen subset configuration, which is itself tied to the assumed wind direction. While effective for this fixed scenario, automating this subset selection, perhaps even dynamically based on the current inflow angle, remains a key area for future work.

Table 4-6: Subset definition for the six–turbine layout used by the subset DMPC controller. Each entry lists the turbines included in the subset $\mathcal{S}_i$ when optimising turbine i. Turbine numbers correspond to the numbers in the farm layout in Figure 4-2.

	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{S}_5$	$\mathcal{S}_6$
Turbines included	1, 2, 4	1, 2, 3, 4, 5	2, 3, 5	4, 5, 6	4, 5, 6	6

Figure 4-11 compares farm–level relative tracking errors for the six–turbine layout. Up to $t \approx 3\,\text{min}$ most controllers maintain very small errors, with only brief settling transients at the start and around $t \approx 2\,\text{min}$ when the reference steps. The exception is the centralised controller without multistarts, which exhibits larger fluctuations and a pronounced spike near $t \approx 1.5\,\text{min}$. This peak coincides with the first wake front from the upstream row reaching the downstream turbines and indicates that the centralised single–start solution struggles to anticipate the interaction; once the wake has propagated, the error drops. At $t \approx 3\,\text{min}$, the other controllers show a sharp error spike. This is not a numerical instability but a limitation of the predictive model's effective reach. Because the predictor is built on a finite chain of OPs, the wake information generated by turbines 1–3 does not fully propagate to the third row (turbine 6). As a result, the effective wind speed at the furthest downstream turbine is overestimated and the farm power is (momentarily) overpredicted. After this event the controllers resettle, but this reveals a structural model mismatch, as the model's finite horizon prevents it from fully capturing the wake propagation to the distant third row. This mismatch leads to a consistent, slight overestimation of the wind speed at turbine 6 at subsequent steps, resulting in a small, repeated tracking error. However, the distributed formulations prove less sensitive to this model–reach issue and recover faster after the event.

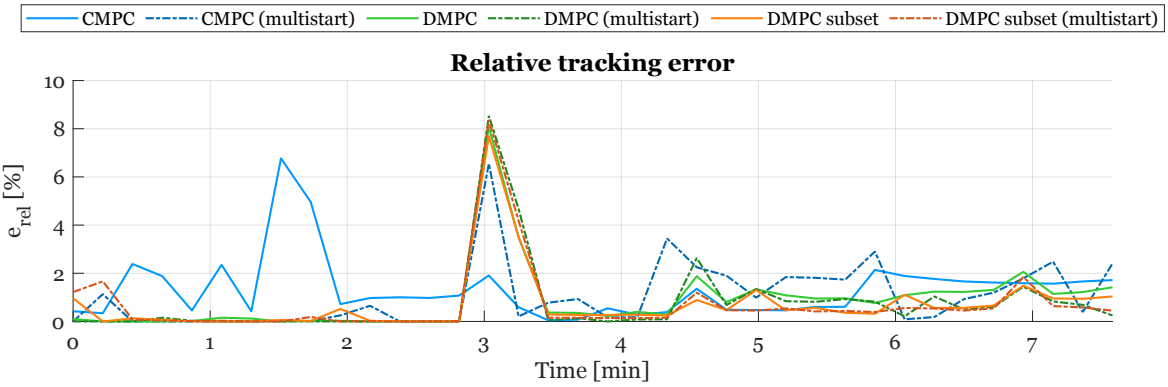

Figure 4-11: Time series of farm-level relative tracking error e_{rel} comparing all proposed controllers. The test uses the six-turbine layout (Fig.4-2), with $\theta = 0°$, $V_\infty = 10\ \text{m}\,\text{s}^{-1}$, and tracking the step reference (Reference 1, Fig. 4-3).

Figure 4-12 shows rotor–effective wind speeds for the downstream turbines (turbines 4–6). The upstream turbines (turbines 1–3) remain at their inflow values and are therefore omitted. For most controllers a first, coherent drop appears near $t \approx 1.5\,\text{min}$, consistent with the wake from the upstream row reaching the downstream row. The magnitude of this drop differs across controllers, reflecting different axial–induction and yaw actions before the wake arrives. A second feature appears between $t \approx 2.8$ and $3.2\,\text{min}$. Under CMPC the rotor–effective wind at turbine 4 and turbine 5 decreases again, whereas turbine 6 shows only a small kink rather than a sustained reduction. This is consistent with the flow–energy budget: with lower wind at turbines 4 and 5, those turbines extract less power, so the downstream velocity deficit is weaker and the perturbation that reaches turbine 6 is attenuated, allowing a quicker recovery. By contrast, the distributed controllers moderate the drop at turbines 4–5 more effectively and keep a flatter trace at turbine 6, which suggests better coordination of upstream control to shield the third row from compound wake effects.

Control inputs in Figure 4-13 show the same trends as in the three-turbine case. First,

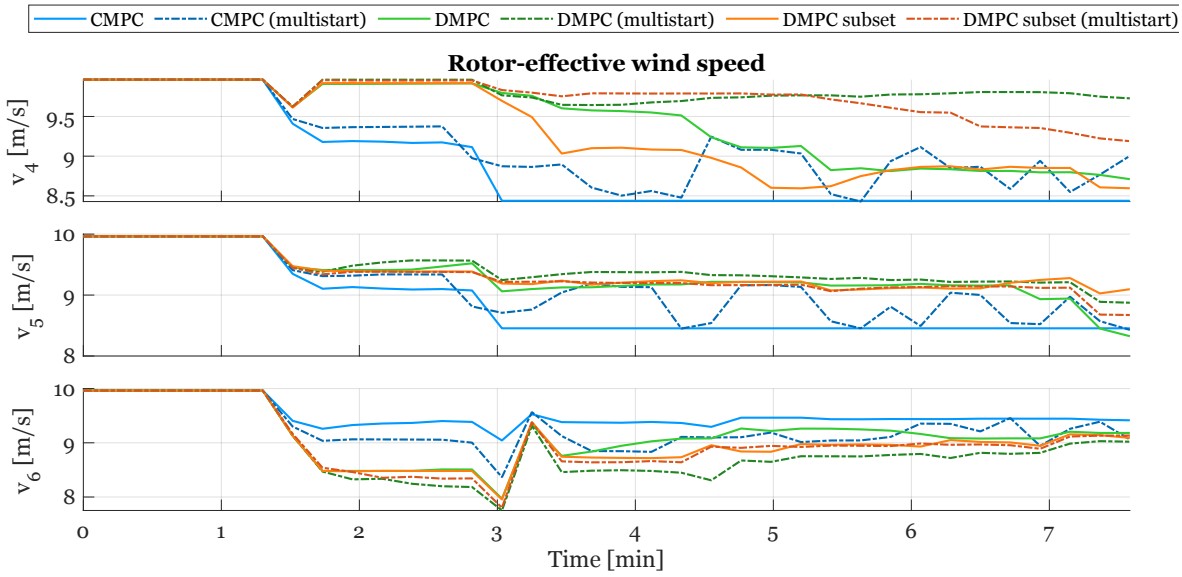

Figure 4-12: Time series of rotor-effective wind speed v_i for the downstream turbines (T4, T5, T6) comparing all proposed controllers. The test uses the six-turbine layout (Fig.4-2), with $\theta = 0°$, $V_\infty = 10\ \mathrm{m\,s^{-1}}$, and tracking the step reference (Reference 1, Fig. 4-3). The legend applies to all plots.

sequential solving concentrates most control action at the turbines solved first. In the subset DMPC case, for example, the axial induction factors of turbine 1 and turbine 2 are very low at the start, indicating an attempt to resolve much of the farm-level objective upstream at the start of the sequential sweep, while the yaw angles of downstream turbines show little variability. The centralised controllers spread control action more evenly across the turbines. Second, multistart solving introduces greater variability in the inputs. This is most visible in the axial induction factors, where the dotted traces switch more often. The effect is less pronounced for yaw angles because rate limits constrain the admissible changes, so the feasible range is more narrow and largely shaped by the current yaw state.

A comprehensive summary of controller performance across the two six-turbine experiments is presented in Table 4-7 and Figure 4-14. The table provides aggregate statistics for mean tracking error and solver times, while the scatter plot visualises the speed-versus-accuracy trade-off for each controller in each of the two experiments. On the plot, the horizontal axis represents mean computation time t_{solve}, and the vertical axis represents mean relative error e_{rel}, so points in the lower-left are ideal.

The single-start controllers form a cluster on the left 'fast' side of the plot. A key finding is visible in the single-start CMPC: the two experiments yield vastly different results. One experiment results in the highest tracking error on the entire chart, while the other has a very low error. This demonstrates that the performance of the single-start CMPC is highly dependent on the specific reference trajectory. By contrast, the other single-start controllers (DMPC and DMPC subset) show tightly clustered points, indicating they are more robust to the different scenarios. While the single-start CMPC achieves the lowest mean solve time, the DMPC subset approach achieves the lowest time variability (std. dev.) and the lowest maximum solver time, making it more predictable.

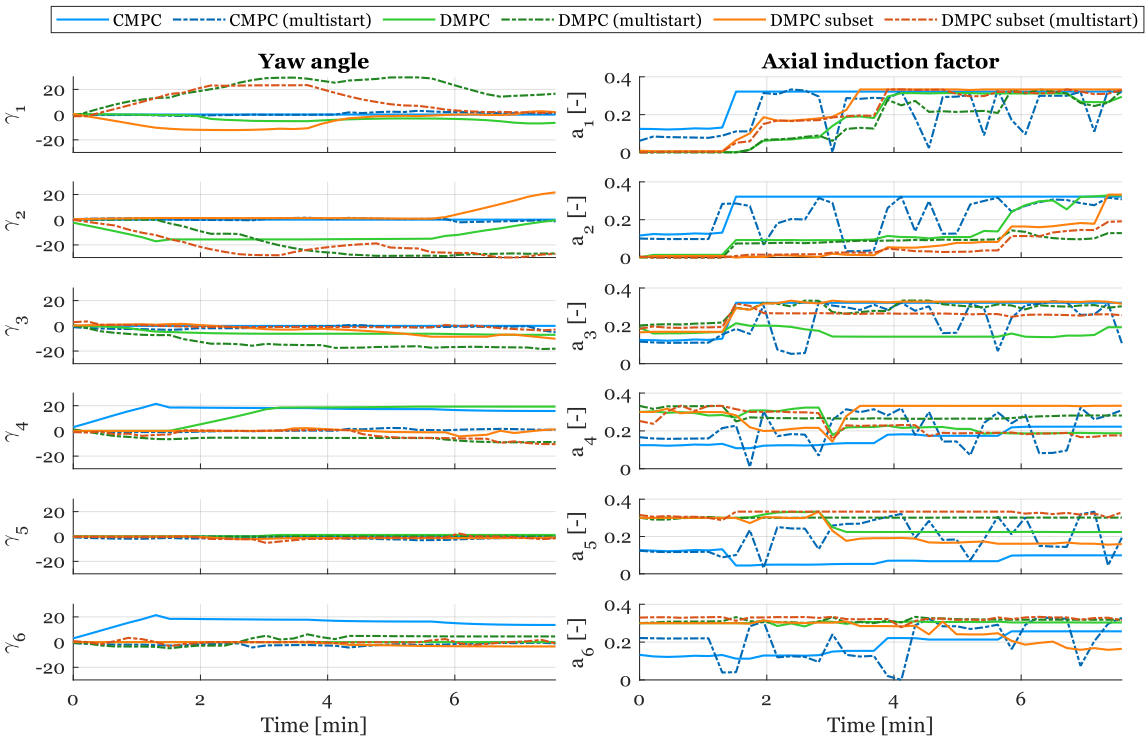

Figure 4-13: Time series of the control inputs (yaw angles γ_i and axial induction factors a_i) for all six turbines, comparing all proposed controllers. The test uses the six-turbine layout (Fig.4-2), with $\theta = 0°$, $V_\infty = 10\,\mathrm{m\,s^{-1}}$, and tracking the step reference (Reference 1, Fig. 4-3). The legend applies to all plots.

Introducing multistarts reveals a clear separation in computational cost. The full-farm multistart methods (CMPC and DMPC) are pushed to the far right of the plot, indicating extremely high mean solve times that make them computationally impractical, despite improving accuracy. This is where the subset strategy's advantage becomes most apparent. The DMPC subset with multistarts achieves the best mean relative tracking error of all controllers. More importantly, it does so at a mean solve time that is dramatically lower than the other multistart variants. This significant time improvement highlights the benefit of reducing the sub-problem complexity. The advantage is particularly pronounced in the multistart case; while single-start runs are fast for all methods due to 'warm-starting' from the previous solution, multistarting requires solving from multiple randomised initial guesses. When solving from these randomised starting points, the vast reduction in problem size and complexity offered by the subset approach translates directly into a dramatic reduction in solver time. The subset approach thus retains the high accuracy of multistarting while drastically cutting the computational burden.

The per-turbine solver times for the DMPC variants, which are presented in Table B-3 in Appendix B, reveal a final key observation. As expected from the sequential approach, the computational load is concentrated at the front of the farm. However, the specific source of the highest computational load differs: for the full-farm DMPC, Turbine 1 has the largest mean computation time. For the DMPC subset approach, this most time-consuming step shifts to Turbine 2. This is a direct consequence of the subset design; Turbine 2 has the largest

defined subset (Table 4-6), making its individual optimisation problem the most complex in the sequence.

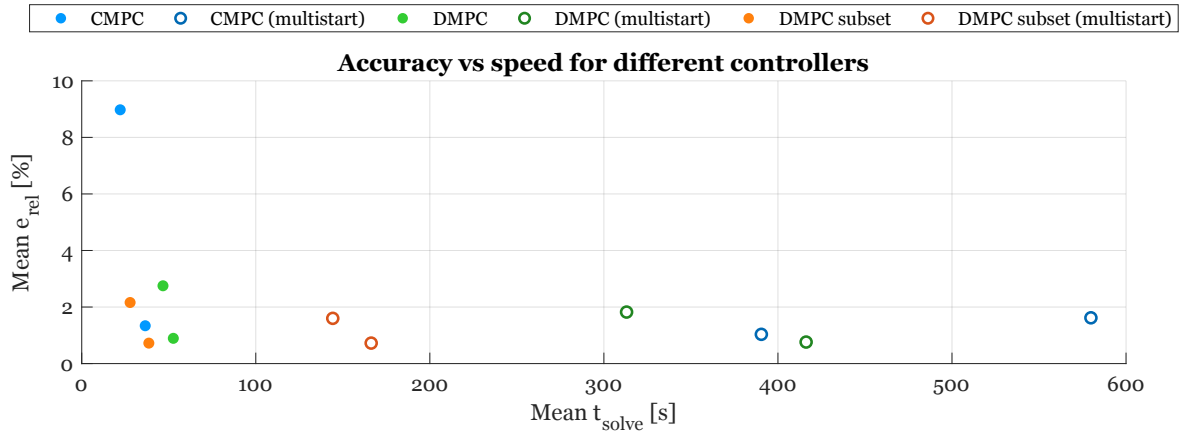

Figure 4-14: Scatter plot illustrating the accuracy-speed trade-off, showing mean relative tracking error e_{rel} versus mean solver time t_{solve} for all proposed controllers. Each point represents the mean result of one of the two experiments conducted on the six-turbine layout (Fig. 4-2).

A crucial, counterintuitive finding from these experiments is that the optimisation problem incorporating the most approximations (subset DMPC approach) yields the superior tracking performance. This unexpected result can be attributed to two possible explanations. The first is myopic control behaviour, where a controller with a reduced prediction model makes a slightly suboptimal decision early in the horizon that, by chance, leads to a superior state for the long-term performance. However, given that the prediction horizon covers a significant portion of the simulation time, truly myopic behaviour is unlikely to be the primary cause of the sustained performance benefit. The second, and more probable, explanation is the favourable trade-off between reduced computational complexity and effective approximation. The subset optimisation problem is significantly less complex (lower dimensionality and non-linearity) than the full-farm or centralised optimisation problems. With a reasonable, fixed number of multistarts, the solver is simply more effective at locating a good, near-optimal solution for the simpler subset problem. This is a practical benefit, as the highly complex centralised problems often feature numerous local minima where the solver gets trapped. This effect is amplified because the manual subset selection showed highly effective: by capturing all strong, directional wake interactions, the turbines excluded from the dynamic model are negligibly weakly coupled. However, this clear separation may become more difficult to achieve in farms with complex or irregular topologies. The simple algebraic approximation used for these turbines is highly accurate, meaning the computational reduction does not introduce a significant modelling error, thus allowing the optimisation benefits to dominate the overall performance.

In summary, three clear conclusions emerge for this larger farm. First, the single-start CMPC is fast but its performance is unreliable and trajectory-dependent. Second, the subset-based controllers offer the best trade-offs. The single-start DMPC subset provides the best balance of speed, low error, and high predictability, making it the strongest candidate for a real-time budget. Third, if near-perfect accuracy is required and more computation time is available, the DMPC subset with multistarts is the superior choice, delivering the best accuracy at a

fraction of the cost of full-farm multistart methods.

Table 4-7: Per-controller summary of mean relative tracking error e_{rel} and solver time t_{solve} statistics per time step, aggregated across the six-turbine experiments. The mean, standard deviation (Std. dev.), and maximum (Max) values for t_{solve} are reported. The most preferable result in each column is highlighted in bold. DMPC solver times represent the total (sum) time for the sequential solve at each timestep.

Controller	Mean e_{rel} [%]	Mean t_{solve} [s]	Std. dev. [s]	Max [s]
CMPC	5.157	**29.30**	64.12	558.61
CMPC (multistart)	1.327	485.15	341.05	1192.04
DMPC	1.823	49.66	29.35	150.54
DMPC (multistart)	1.292	364.62	153.83	758.76
DMPC subset	1.442	33.18	**25.33**	**138.60**
DMPC subset (multistart)	**1.162**	155.27	66.51	390.32

4-4-4 Cross-layout comparison

This subsection compares performance and computational effort between the three-turbine and six-turbine layouts, using the per-controller summary in Table 4-8. The results for both layouts are drawn from the same two operating scenarios (the step and perturbed constant references at $\theta = 0°, V_\infty = 10\,\mathrm{m\,s^{-1}}$), providing a direct comparison. To aid comparability, solver time is normalised per turbine, denoted as $\bar{t}_{\mathrm{solve}}$, by dividing the mean time per step by the number of turbines:

$$\bar{t}_{\mathrm{solve}} = t_{\mathrm{solve}}/N_{\mathrm{T}}. \tag{4-14}$$

A modest increase in tracking error is expected when moving from three to six turbines, as the prediction model's limit on downstream observation points (N_{OP}) becomes an active factor in the larger layout. This trend is generally present for most controllers. However, the single-start CMPC shows a dramatic increase in error (from 2.2% to 5.2%), suggesting its performance is highly sensitive to the layout complexity and struggles to find a good solution in the larger, more coupled six-turbine case.

The normalised solver times reveal how the approaches scale. For CMPC, the per-turbine time increases significantly (a 3-4x rise for single-start). This is because the central optimisation problem grows in size and coupling, increasing the effort for the interior-point algorithm. The full-farm DMPC scales even more poorly, with the per-turbine time increasing by a factor of 4-5. This poor scaling is twofold. First, the cost of each sub-problem is high: although solved sequentially, each turbine's optimisation must still propagate its effects through the entire farm model to evaluate its influence on all other turbines. Second, the difficulty of each sub-problem increases with farm size. In the three-turbine case, each turbine has a large individual influence (1/3), allowing the sequential solve to meet the reference with relatively easy sub-problems, which may result in somewhat optimistic performance for the smaller layout. In the six-turbine case, the influence of a single turbine is much smaller (1/6), forcing successive sub-problems to work harder and require more computational effort to reduce the remaining gap to the reference. This combination of costly full-farm evaluations and increased sub-problem difficulty explains the steep rise in per-turbine cost.

Comparing DMPC subset with full-farm DMPC on six turbines shows that restricting interactions substantially reduces mean solve time per turbine (from 8.28 s to 5.53 s) while, counterintuitively, the mean tracking error also slightly improves. This suggests that exchanging the full problem complexity for a simpler one based on good approximations can be beneficial both tracking performance and speed.

The multistart results highlight these scaling trade-offs. Relying on randomised initial conditions, the computational cost of multistarting is highly dependent on the underlying problem complexity. This cost is severe for the full-farm CMPC and DMPC, leading to massive increases in solve time. However, the effect is far less pronounced for the DMPC subset. This confirms the non-linear scaling of NLPs: by reducing the problem complexity, the subset approach achieves a non-linear improvement in computation time, making the robust multistart strategy computationally feasible.

In summary, these results show a clear trade-off. While the single-start CMPC scales best in terms of time (compared to full-farm DMPC), it does so at the cost of a significant and unacceptable increase in tracking error. The full-farm DMPC scales poorly due to the compounding cost of its full-farm rollouts, but is however able to keep the tracking error at an acceptable level. The DMPC subset approach emerges as the most scalable method, offering a performance-speed profile that remains balanced as the layout grows.

Table 4-8: Comparison of mean relative tracking error e_{rel} and mean solver time t_{solve} per time step for the three-turbine (3T) and six-turbine (6T) layouts. The solver time for both layouts is normalised per turbine $\bar{t}_{\text{solve}}$ (4-14) to aid comparability. The most preferable result in each column in highlighted in bold.

Controller	3T e_{rel} [%]	6T e_{rel} [%]	3T $\bar{t}_{\text{solve}}$ [s/turb]	6T $\bar{t}_{\text{solve}}$ [s/turb]
CMPC	2.247	5.157	**1.36**	**4.88**
CMPC (multistart)	0.066	1.327	35.73	80.86
DMPC full farm	1.710	1.823	1.45	8.28
DMPC full farm (multistart)	**0.014**	1.292	15.14	60.77
DMPC subset	–	1.442	–	5.53
DMPC subset (multistart)	–	**1.162**	–	25.88

4-5 Summary

This chapter has presented the case study setup and results used to evaluate the proposed MPC approaches. Several prediction model approximations are compared, showing that stronger approximations significantly reduce computation time but also increase tracking error, illustrating the trade-off between accuracy and efficiency.

The controller evaluations first benchmark the proposed nonlinear MPC approaches against a state-of-the-art MIQCQP method, demonstrating their superior performance in both tracking accuracy and practical applicability. Subsequent experiments reveal clear trade-offs in performance and scalability. Single-start CMPC is fast, but its tracking performance proves unreliable, degrading significantly on the larger farm. The full-farm DMPC offers more robust tracking but scales more poorly in computation due to its costly sequential, full-farm model

evaluations. The DMPC subset approach provides the best overall balance, achieving low, robust tracking error at a predictable computational cost. Introducing multistarts for higher accuracy amplifies these differences. The computational cost becomes prohibitive for the full-farm methods, but remains manageable for the DMPC subset. Overall, the DMPC subset method emerges as the most promising and scalable solution for wind-farm flow control.

Chapter 5

Conclusions, discussion and recommendations

This chapter concludes the research presented in this thesis. First, the contributions presented in the preceding chapters are summarised. Subsequently, the limitations are addressed, followed by recommendations for future work.

5-1 Conclusions

This thesis investigates how to balance computation tractability and control performance in wind-farm flow control. This is addressed from two perspectives, by integrating model approximations and by applying distributed control within an model predictive control (MPC) framework. The core of this work is the novel integration of the FLORIDyn model as the MPC prediction model to explicitly capture dynamic wake propagation and recovery.

The first contribution is the successful embedding of the dynamic FLORIDyn model within an MPC framework, moving beyond common static or linearised models. This enables the development of a centralised model predictive control (CMPC) controller and two novel distributed model predictive control (DMPC) controllers (full-farm and subset-based) that systematically decompose the global problem. These proposed controllers show to significantly outperform the state-of-the-art mixed-integer quadratically-constrained quadratic program (MIQCQP) formulation in both tracking accuracy and numerical robustness.

The second contribution is the design of a full-farm DMPC strategy, which decomposes the central problem into a sequence of per-turbine subproblems. This decomposition proves highly effective in the three-turbine case study. By breaking the large, non-convex optimisation problem into smaller, more manageable sequential solves, the DMPC approach is able to find high-quality solutions more reliably than the fully centralised controller. This leads to a preference for the DMPC controller in smaller layouts, where it offers a better accuracy-speed balance and more consistent runtimes than the centralised benchmark. This also highlights

a key difference: the CMPC controller often requires expensive multistarts to achieve good performance, whereas the single-start DMPC variant is already robust, achieving low tracking errors on its own and rendering the extra cost unnecessary.

The third contribution extends the distributed framework with a subset DMPC architecture, designed to enhance scalability for larger farms. This is validated in the six-turbine layout, where the DMPC subset variant yields the most favourable results, achieving the best tracking performance of all controllers while maintaining fast and predictable computation times. This presents an unexpected but critical finding: the most approximated approach delivers the best overall tracking. By constraining the optimisation to only locally interacting turbines in subsets, the problem's computational complexity is drastically reduced. It shows that for these experiments, the benefit of having a simpler optimisation problem is greater than the suboptimality introduced by the approximations, allowing the subset controller to find a higher quality solution with the given number of multistarts than the full-farm controllers. This performance is amplified by the high quality of the approximations; the manually tailored subsets effectively isolate the strong aerodynamic couplings, ensuring that the use of passed wind-speed predictions introduces negligible error. The cross-layout comparison further solidifies this conclusion, showing that as farm size increases, the CMPC approach suffers from unreliable tracking, while the full-farm DMPC scales poorly in computation time. The DMPC subset approach, however, provides the most balanced and scalable solution, maintaining both high performance and computational tractability.

The fourth contribution involves analysing approximations within the FLORIDyn prediction model itself, specifically truncating the observation point (OP) chain length N_{OP} and reducing the chain count N_{c}. These parameters act as tuning knobs for the trade-off between model accuracy and computation time. Analysis of these prediction model approximations confirms that while reducing the model order decreases computational demand, the trade-off is non-linear in its benefits. This non-linearity allows for a favourable balance: while significant approximation can introduce substantial tracking error, a trade-off can be made to decrease the computational effort while still preserving the essential non-linear dynamic wake features required for effective control.

In summary, this thesis confirms that the challenge of balancing computational tractability and control performance in wind-farm control can be addressed through a combination of model approximation and distributed control. The integration of the FLORIDyn model into an MPC framework (Contribution 1) proves superior to state-of-the-art methods. The analysis of model approximations (Contribution 4) reveals a non-linear trade-off, showing that computational gains are possible without big losses in accuracy. The development of distributed strategies (Contributions 2 & 3) demonstrate that a DMPC subset approach provides a scalable solution, offering a superior accuracy-speed trade-off compared to both centralised and full-farm distributed methods, a benefit that strengthens with increasing farm size. This research shows that such a structured decomposition, which focuses computation on locally interacting subsystems, can successfully balance the competing demands of control performance and computational cost, enabling cooperative closed-loop control of wind farms.

5-2 Discussion and limitations

The findings of this research should be considered in light of several limitations. These primarily concern underlying assumptions about the environment and the generalisability of the proposed control architecture.

A primary limitation is the fundamental assumption of constant wind conditions over the prediction horizon. This assumption critically shapes the control framework in three ways. First, for the sequential approach, the optimisation order must proceed from upstream to downstream. This static order, however, becomes sub-optimal or invalid under changing wind directions that alter wake propagation. Second, the constructed groups for the subset DMPC are designed to capture the most severe wake interactions for a static inflow. These manually defined subsets may no longer be appropriate if the wind direction shifts, as the dominant turbine-to-turbine couplings will change. Third, the prediction accuracy of the MPC controller itself will likely decrease in time-varying wind, as it assumes constant inflow.

Building on this, the current approach depends on several hand-crafted components tailored to the specific farm layout. This includes the prediction model's internal parameters (such as the number of OPs N_{OP} in a chain) and, most notably, the manual construction of the control subsets. This manual design process becomes more challenging and impractical as farm size and layout complexity increase and wind conditions become time-varying. While interactions in a simple, aligned layout are obvious (e.g., turbine 1 hits turbine 4), in large, irregularly staggered farms, a single turbine's wake may partially impact multiple downstream turbines. Manually identifying and defining all such complex, layout-dependent interactions for every turbine is not a scalable or generalisable approach.

Further limitations are inherent in the chosen sequential distributed control strategy. The sequential solving approach, while reducing complexity, inherently concentrates control action on the turbines that solve early in the sequence. This can lead to uneven wear and tear of upstream turbines and may increase the computational burden if significant control changes are required. Furthermore, from a scalability perspective, a sequential approach (per control step) will never be computationally faster than the sum of its sub-problems, and therefore the total computation time will continue to grow with farm size.

Additionally, the controllers optimise for a given power reference without considering structural loads. This can lead to undesirable emergent behaviours, such as a controller steering its wake directly into a downstream turbine to meet a curtailment target. While this achieves the power goal, it significantly increases loads on the downstream turbine.

Furthermore, the detailed analysis of the model approximations (such as reducing the chain count N_c) is only performed using the CMPC controller. These prediction models are not evaluated in combination with the DMPC frameworks, which possibly reveal further performance and scalability benefits.

Finally, the extensiveness of the case study is limited. The controllers are tested only on three- and six-turbine configurations, with the six-turbine case itself being restricted in the number of scenarios evaluated. This limited scale prevents a full assessment of computational scalability and the impact of cumulative, long-range wake interactions that would be present in larger farms. Furthermore, the farm layout used in the case study features close spanwise spacing (in the L_y direction). This specific configuration creates strong wake couplings that the subset

approach exploits effectively. However, this performance benefit may not fully generalise to other operating environments. In layouts with wider spacing or irregular staggering, wake interactions become more diffuse and transient. In such scenarios, the effectiveness of the subset strategy is likely to diminish; the approximations used for non-local turbines may introduce greater suboptimality than observed in this work.

5-3 Recommendations and future work

Building on the conclusions and the limitations identified, this section outlines recommendations to enhance the proposed control framework and presents a path for future work towards real-world implementation. The suggestions focus on improving robustness, computational performance, and generalisability, thereby addressing the key challenges that currently constrain the controller's practical deployment.

To improve the controller's adaptability to changing environmental conditions and its transferability to new or larger farms, future work must address the framework's current reliance on static, manually-tuned, and layout-specific parameters. The use of a fixed upstream-to-downstream optimisation order and manually defined subsets constitutes a significant limitation, as this structure becomes sub-optimal under varying wind directions and prevents easy application without case-by-case redesigns. Therefore, it is recommended to adopt a time-varying approach where parameters are formulated in an abstract, automated manner based on real-time wind data. Specifically, a robust algorithm should use wind direction measurements and a wake model to automatically determine the DMPC solving order and dynamically identify the most strongly interacting subsets. This adaptive framework can be further enhanced by integrating short-term wind predictions and by making the prediction model's chain length, N_{OP}, an adaptive parameter that automatically scales based on wind speed, wind direction, and turbine spacing to ensure wake interactions are captured without needless computation.

Further recommendations concern the distributed strategy and its consequences. The sequential solving approach is noted to concentrate control action at the front of the farm, which can lead to uneven turbine wear. This can be countered by embedding a more even distribution of control action within the control objective or constraints. Furthermore, to improve the scalability of the distributed controller, which is currently limited by the sequential solve over all turbines, a partitioning strategy can be explored. A promising approach is to divide the farm into a few larger, weakly interacting subsystems. These subsystems can then be optimised in parallel, or sequentially with fewer steps, potentially offering a significant reduction in computation time without a substantial loss in performance.

While this work shows that tuning FLORIDyn's approximations is a viable strategy, its high baseline computational cost remains a bottleneck as farm size increases. Future work should therefore investigate alternative prediction models that offer a more favourable balance between fidelity and speed. The methodology developed in this thesis can be applied to such scalable middle-ground models to further reduce computational demand, though the resulting trade-off regarding model approximation error would require careful management.

For future work aimed at practical implementation, the limited scope of the case study must first be expanded. More extensive testing across various layouts, turbine counts, and wind con-

ditions is necessary to validate the controller's performance and scalability robustly. Following this, the controller must be tested more rigorously in high-fidelity simulation environments. This will allow for a thorough evaluation of its interaction with more realistic environment. A crucial step will be to integrate the farm-level controller with the individual wind turbine controllers, creating a hierarchical system where the controller provides optimal setpoints to the local controllers.

Finally, a critical step for real-world deployment is to address the inevitable mismatch between the prediction model and the physical world. This requires the development of a robust observer to reconstruct the necessary model states from available sensor data. Furthermore, the controller's robustness to this model-plant mismatch, as well as to sensor noise and unmodelled disturbances, must be a central focus. This may involve incorporating robust control techniques to ensure safe and reliable performance under real-world operating conditions. By systematically addressing these recommendations, the control framework presented in this thesis can be developed into a more robust and scalable solution for large-scale wind farm flow control.

Appendix A

Wake shape parameter calculation

This appendix details the specific formulas used within the FLORIDyn model [9, 44] to calculate the key wake shape variables for each observation point (OP): the lateral and vertical wake spreads (σ_y, σ_z), the lateral wake deflection (δ), and the potential core length and widths ($x_c, w_{y,\mathrm{pc}}, w_{z,\mathrm{pc}}$), defining the potential core ($\mathcal{C}_{\mathrm{core}}$) boundaries. These calculations are performed individually for each OP based on its current downwind position x_1, its inherited internal state $\gamma^{\mathrm{OP}}, a^{\mathrm{OP}}$, and the ambient conditions, specifically the turbulence intensity I.

First the effective turbulence intensity, I^{OP}, is calculated at the OP's position by combining the ambient turbulence I and the foreign-wake-induced turbulence I_f:

$$I^{\mathrm{OP}} = \sqrt{I^2 + I_f^2} \tag{A-1}$$

The foreign-wake-induced turbulence I_f represents the cumulative added turbulence intensity generated by upstream turbines. It is computed using the empirical Crespo-Hernández model [9]. For an OP located at a downstream distance Δx_j from an upstream turbine j operating with an axial induction factor a_j, the added turbulence intensity $I_{+,j}$ is given by:

$$I_{+,j} = 0.73 a_j^{0.8325} I^{-0.0325} \left(\frac{\Delta x_j}{D}\right)^{-0.32}. \tag{A-2}$$

If the OP is influenced by a set of multiple upstream turbines $\mathcal{U}$, the total foreign turbulence I_f is obtained using a sum of squares superposition:

$$I_f = \sqrt{\sum_{j \in \mathcal{U}} (I_{+,j})^2} \tag{A-3}$$

The length of the potential core, x_c, is calculated based on the OP's stored state, C_{T} (2-9) and this effective turbulence intensity I^{OP}:

$$x_c = \frac{\cos(\gamma^{\mathrm{OP}})(1 + \sqrt{1 - C_{\mathrm{T}}})}{\sqrt{2}(\alpha_c I^{\mathrm{OP}} + \beta_c(1 - \sqrt{1 - C_{\mathrm{T}}}))} D \tag{A-4}$$

where $\alpha_c = 2.32$ and $\beta_c = 0.154$ are empirical constants [5]. The wake growth rates, k_y and k_z, determine the expansion of the wake and are also a function of the effective turbulence intensity I^{OP}:

$$k_y = k_a I^{\mathrm{OP}} + k_b \tag{A-5}$$

$$k_z = k_y \tag{A-6}$$

where $k_a = 0.38371$ and $k_b = 0.003678$ are empirical parameters [9]. The wake spreads σ_y, σ_z are calculated using a piecewise function that handles the linear growth of the initial wake width inside the potential core ($x_1 \leq x_c$) and the turbulence-driven expansion in the far wake ($x_1 > x_c$):

$$\sigma_y = k_y \max(x_1 - x_c, 0) + \min\left(\frac{x_1}{x_c}, 1.0\right) \frac{\cos(\gamma^{\mathrm{OP}})D}{\sqrt{8}} \tag{A-7}$$

$$\sigma_z = k_z \max(x_1 - x_c, 0) + \min\left(\frac{x_1}{x_c}, 1.0\right) \frac{D}{\sqrt{8}} \tag{A-8}$$

where the $\max(\cdot)$ function ensures the turbulence-driven term only applies after the potential core, and the $\min(\cdot)$ function provides a linear scaling of the initial wake width up to the end of the core. The lateral wake deflection δ is also a piecewise function. First, the initial deflection angle at the rotor, θ_0, is calculated:

$$\theta_0 = \frac{0.3\gamma^{\mathrm{OP}}}{\cos(\gamma^{\mathrm{OP}})}(1 - \sqrt{1 - C_{\mathrm{T}}\cos(\gamma^{\mathrm{OP}})}) \tag{A-9}$$

The total deflection δ is the sum of a near-wake component δ_{nfw} and a far-wake component δ_{fw}, which is activated only for $x_1 > x_c$ using a Heaviside step function $H(x_1 - x_c)$:

$$\delta(x_1) = \delta_{nfw} + H(x_1 - x_c) \cdot \delta_{fw} \tag{A-10}$$

The near-wake component provides linear deflection within the core:

$$\delta_{nfw} = \theta_0 \min(x_1, x_c) \tag{A-11}$$

The far-wake component is more complex, composed of two terms $\delta_{fw,1}$ and $\delta_{fw,2}$:

$$\delta_{fw} = \delta_{fw,1} \cdot \delta_{fw,2} \cdot D \tag{A-12}$$

$$\delta_{fw,1} = \frac{\theta_0}{14.7}\sqrt{\frac{\cos(\gamma^{\mathrm{OP}})}{k_y k_z C_{\mathrm{T}}}}(2.9 + 1.3\sqrt{1 - C_{\mathrm{T}}} - C_{\mathrm{T}}) \tag{A-13}$$

$$\delta_{fw,2} = \log\left(\frac{(1.6 + \sqrt{C_{\mathrm{T}}})(1.6\sqrt{\frac{8\sigma_y\sigma_z}{D^2\cos(\gamma^{\mathrm{OP}})}} - \sqrt{C_{\mathrm{T}}})}{(1.6 - \sqrt{C_{\mathrm{T}}})(1.6\sqrt{\frac{8\sigma_y\sigma_z}{D^2\cos(\gamma^{\mathrm{OP}})}} + \sqrt{C_{\mathrm{T}}})}\right) \tag{A-14}$$

Finally, the boundaries of the potential core, $w_{y,\mathrm{pc}}$ and $w_{z,\mathrm{pc}}$, are calculated. These are used to determine the reduction factor for OPs inside the core (2-7). The calculation starts with the velocity ratio at the rotor, $v_{r,0}$, representing the ratio of the wind speed immediately behind the rotor to the free-stream wind speed:

$$v_{r,0} = \frac{C_{\mathrm{T}}\cos(\gamma^{\mathrm{OP}})}{2(1 - \sqrt{1 - C_{\mathrm{T}}\cos(\gamma^{\mathrm{OP}})})\sqrt{1 - C_{\mathrm{T}}}} \tag{A-15}$$

The boundaries then linearly decrease from the rotor plane to zero at the end of the potential core:

$$w_{y,\mathrm{pc}} = D\cos(\gamma^{\mathrm{OP}})\sqrt{v_{r,0}}\max\left(1-\frac{x_1}{x_c},0\right) \tag{A-16}$$

$$w_{z,\mathrm{pc}} = D\sqrt{v_{r,0}}\max\left(1-\frac{x_1}{x_c},0\right) \tag{A-17}$$

A special case is made for OPs exactly at the rotor plane ($x_1 = 0$), where the boundaries are set to match the rotor's physical dimensions to ensure correct initialisation:

$$w_{y,\mathrm{pc}}(x_1 = 0) = D\cos(\gamma^{\mathrm{OP}})$$
$$w_{z,\mathrm{pc}}(x_1 = 0) = D$$

In summary, this appendix has detailed the parametrisation used to compute the key OOP-specific wake shape variables: the potential core length (x_c), wake growth rates (k_y, k_z), wake spreads (σ_y, σ_z), and lateral deflection (δ). Furthermore, it provided the formulas for the potential core boundaries ($w_{y,\mathrm{pc}}, w_{z,\mathrm{pc}}$), which are used to geometrically identify the set of OPs within the potential core, $\mathcal{C}_{core}$.

Appendix B

Detailed controller performance statistics

This appendix presents the controller performance statistics tables, which are summarised in Chapter 4. These tables provide the full, detailed results for the experiments. Table B-1 corresponds to the state-of-the-art comparison on the three-turbine layout (Section 4-4-1). Table B-2 contains the aggregated results from all eight experiments on the three-turbine layout (Section 4-4-2). Table B-3 presents the aggregated results from the experiments on the six-turbine layout (Section 4-4-3). The primary addition in these detailed tables is the inclusion of the per-turbine solver time statistics for all distributed model predictive control (DMPC) controller variants, which were omitted from the main text for brevity.

Table B-1: Detailed performance statistics for the state-of-the-art comparison (Section 4-4-1). The table shows mean relative tracking error e_{rel} and solver time t_{solve} statistics, including per-turbine data for DMPC. The mean, standard deviation (Std. dev.), and maximum (Max) values for t_{solve} are reported. The most preferable result in each column is highlighted in bold. For DMPC rows, 'total' denotes the sum across turbines at each time step.

Controller	Mean e_{rel} [%]	Mean t_{solve} [s]	Std. dev. [s]	Max [s]
MIQCQP	13.207	20.58	24.33	78.33
CMPC	2.126	**4.29**	**5.16**	**22.254**
CMPC (multistart)	0.135	57.37	56.22	187.18
DMPC (total)	0.013	19.32	10.08	43.87
Turbine 1		8.15	5.96	21.23
Turbine 2		8.42	8.43	35.34
Turbine 3		2.74	0.10	2.95
DMPC (multistart, total)	**0.012**	49.10	23.21	119.50
Turbine 1		14.92	9.80	41.82
Turbine 2		15.97	8.59	34.84
Turbine 3		18.20	18.59	78.42

Table B-2: Detailed performance statistics aggregated across all three-turbine experiments (Section 4-4-2). The table shows mean relative tracking error e_{rel} and solver time t_{solve} statistics, including per-turbine data for DMPC. The mean, standard deviation (Std. dev.), and maximum (Max) values for t_{solve} are reported. The most preferable result in each column is highlighted in bold, comparing the centralised model predictive control (CMPC) controllers and the DMPC 'total' rows.

Controller	Mean e_{rel} [%]	Mean t_{solve} [s]	Std. dev. [s]	Max [s]
CMPC	1.928	**5.96**	8.67	77.38
CMPC (multistart)	0.245	104.63	110.78	424.73
DMPC (total)	0.841	6.67	**5.52**	**42.94**
Turbine 1		2.39	3.45	38.17
Turbine 2		2.52	3.29	35.18
Turbine 3		1.76	1.64	18.30
DMPC (multistart, total)	**0.009**	56.06	31.67	233.73
Turbine 1		21.00	22.04	189.06
Turbine 2		16.99	12.38	118.21
Turbine 3		18.07	15.47	147.49

Table B-3: Detailed performance statistics aggregated across all six-turbine experiments (Section 4-4-3). The table shows mean relative tracking error e_{rel} and solver time t_{solve} statistics, including per-turbine data for all DMPC variants. The mean, standard deviation (Std. dev.), and maximum (Max) values for t_{solve} are reported. The most preferable result in each column is highlighted in bold, comparing the top-level controller rows (CMPC and DMPC 'total'). Mean relative tracking errors are farm-level values. For DMPC rows, 'total' denotes the sum across turbines at each time step; indented rows report the per-turbine solver times.

Controller	Mean e_{rel} [%]	Mean t_{solve} [s]	Std. dev. [s]	Max [s]
CMPC	5.157	**29.30**	64.12	558.61
CMPC (multistart)	1.327	485.15	341.05	1192.04
DMPC (total)	1.823	49.66	29.35	150.54
Turbine 1		12.02	16.69	113.26
Turbine 2		10.28	10.92	57.62
Turbine 3		8.50	11.23	75.85
Turbine 4		7.77	8.22	64.14
Turbine 5		6.15	10.93	93.62
Turbine 6		4.94	3.31	23.97
DMPC (multistart, total)	1.292	364.62	153.83	758.76
Turbine 1		74.82	68.86	373.55
Turbine 2		58.22	48.69	234.81
Turbine 3		69.63	75.04	373.60
Turbine 4		61.06	39.89	171.78
Turbine 5		48.85	40.52	236.05
Turbine 6		52.05	52.62	274.87
DMPC subset (total)	1.442	33.18	**25.33**	**138.60**
Turbine 1		10.72	16.13	86.50
Turbine 2		12.25	14.48	92.00
Turbine 3		3.74	5.07	33.79
Turbine 4		3.00	3.34	20.03
Turbine 5		2.86	2.82	14.73
Turbine 6		0.62	0.31	2.48
DMPC subset (multistart, total)	**1.162**	155.27	66.51	390.32
Turbine 1		29.12	29.06	142.80
Turbine 2		49.53	46.26	323.46
Turbine 3		22.99	19.58	137.51
Turbine 4		27.81	23.31	138.30
Turbine 5		21.04	15.38	71.47
Turbine 6		4.78	3.85	16.65

Bibliography

[1] D. Allaerts and J. Meyers. Large eddy simulation of a large wind-turbine array in a conventionally neutral atmospheric boundary layer. *Physics of Fluids*, 27(6):065–108, 2015.

[2] L. E. Andersson, O. Anaya-Lara, J. O. Tande, K. O. Merz, and L. Imsland. Wind farm control - Part I: A review on control system concepts and structures. *IET Renewable Power Generation*, 15(10):2085–2108, 2021.

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

[4] R. J. Barthelmie, S. C. Pryor, S. T. Frandsen, K. S. Hansen, J. G. Schepers, K. Rados, W. Schlez, A. Neubert, L. E. Jensen, and S. Neckelmann. Quantifying the impact of wind turbine wakes on power output at offshore wind farms. *Journal of Atmospheric and Oceanic Technology*, 2010.

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

[6] C. J. Bay, J. Annoni, T. Taylor, L. Pao, and K. Johnson. Active power control for wind farms using distributed model predictive control and nearest neighbor communication. pages 682–687, 2018.

[7] C. J. Bay, P. Fleming, B. Doekemeijer, J. King, M. Churchfield, and R. Mudafort. Addressing deep array effects and impacts to wake steering with the cumulative-curl wake model. *Wind Energy Science*, 8(3):401–419, 2023.

[8] M. Becker, D. Allaerts, and J.-W. van Wingerden. FLORIDyn - A dynamic and flexible framework for real-time wind farm control. *Journal of Physics: Conference Series*, 2265(3):032103, 2022.

[9] M. Becker, B. Ritter, B. Doekemeijer, D. van der Hoek, U. Konigorski, D. Allaerts, and J.-W. van Wingerden. The revised FLORIDyn model: implementation of heterogeneous flow and the Gaussian wake. *Wind Energy Science*, 7(6):2163–2179, 2022.

[10] F. Bernardoni, U. Ciri, M. A. Rotea, and S. Leonardi. Identification of wind turbine clusters for effective real time yaw control optimization. *Journal of Renewable and Sustainable Energy*, 13(4):043301, 2021.

[11] S. Boersma, B. Doekemeijer, M. Vali, J. Meyers, and J.-W. van Wingerden. A control-oriented dynamic wind farm model: WFSim. *Wind Energy Science*, 3(1):75–95, 2018.

[12] P. Bortolotti, H. Tarres, K. Dykes, K. Merz, L. Sethuraman, D. Verelst, and F. Zahle. IEA Wind TCP Task 37: Systems engineering in wind energy - WP2.1 reference wind turbines. Technical Report NREL/TP–5000-73492, 1529216, 2019.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. *Distributed optimization and statistical learning via the alternating direction method of multipliers*, volume 3. 2011.

[14] W. Cai, Y. Hu, F. Fang, L. Yao, and J. Liu. Wind farm power production and fatigue load optimization based on dynamic partitioning and wake redirection of wind turbines. *Applied Energy*, 339:121000, 2023.

[15] E. Camponogara, D. Jia, B.H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 22(1):44–52, 2002.

[16] P. Chanfreut, J. M. Maestre, and E. F. Camacho. A survey on clustering methods for distributed and networked control systems. *Annual Reviews in Control*, 52:75–90, 2021.

[17] M. J. Churchfield, S. Lee, J. Michalakes, and P. J. Moriarty. A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics. *Journal of Turbulence*, 13:N14, 2012.

[18] H. Del Pozo González and J. L. Domínguez-García. Non-centralized hierarchical model predictive control strategy of floating offshore wind farms for fatigue load reduction. *Renewable Energy*, 187:248–256, 2022.

[19] D. E. Dismukes and G. B. Upton. Economies of scale, learning effects and offshore wind development costs. *Renewable Energy*, 83:61–66, 2015.

[20] P. Fleming, J. King, E. Simley, J. Roadman, A. Scholbrock, P. Murphy, J. K. Lundquist, P. Moriarty, K. Fleming, J. van Dam, C. Bay, R. Mudafort, D. Jager, J. Skopek, M. Scott, B. Ryan, C. Guernsey, and D. Brake. Continued results from a field campaign of wake steering applied at a commercial wind farm – Part 2. *Wind Energy Science*, 5(3):945–958, 2020.

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

[22] P. M. O. Gebraad and J.-W. van Wingerden. A Control-Oriented Dynamic Model for Wakes in Wind Plants. *Journal of Physics: Conference Series*, 524(1):012186, 2014.

[23] P. M. O. Gebraad and J.-W. van Wingerden. Maximum power-point tracking control for wind farms. *Wind Energy*, 18(3):429–447, 2015.

[24] J. D. Grunnet, M. Soltani, T. Knudsen, M. Kragelund, and T. Bak. Aeolus toolbox for dynamics wind Farm model, simulation and control. *European Wind Energy Conference and Exhibition, EWEC 2010 : Conference Proceedings*, 2010.

[25] Y. Guo, H. Gao, H. Xing, Q. Wu, and Z. Lin. Decentralized Coordinated voltage control for VSC-HVDC connected wind farms based on ADMM. *IEEE Transactions on Sustainable Energy*, 10(2):800–810, 2019.

[26] Yifei Guo, Houlei Gao, Qiuwei Wu, Jacob Østergaard, Dachuan Yu, and Mohammad Shahidehpour. Distributed coordinated active and reactive power control of wind farms based on model predictive control. *International Journal of Electrical Power & Energy Systems*, 104:78–88, 2019.

[27] S. Huang, Q. Wu, Yifei Guo, and Z. Lin. Bi-level decentralized active and reactive power control for large-scale wind farm cluster. *International Journal of Electrical Power & Energy Systems*, 111:201–215, 2019.

[28] N.O. Jensen. A note on wind generator interaction. Report 87-550-0971-9, Risø National Laboratory, Roskilde, 1983.

[29] 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*, 13(6):559–572, 2010.

[30] I. Katic, J. Højstrup, and N. O. Jensen. A simple model for cluster efficiency. 1986.

[31] A. C. Kheirabadi and R. Nagamune. A quantitative review of wind farm control with the objective of wind farm power maximization. *Journal of Wind Engineering and Industrial Aerodynamics*, 192:45–73, 2019.

[32] Z. Lin, Z. Chen, C. Qu, Yifei Guo, J. Liu, and Q. Wu. A hierarchical clustering-based optimization strategy for active power dispatch of large-scale wind farm. *International Journal of Electrical Power & Energy Systems*, 121:106155, 2020.

[33] O. Maas and S. Raasch. Wake properties and power output of very large wind farms for different meteorological conditions and turbine spacings: a large-eddy simulation case study for the German Bight. *Wind Energy Science*, 7(2):715–739, 2022.

[34] J. Meyers, C. Bottasso, K. Dykes, P. Fleming, P. Gebraad, G. Giebel, T. Göçmen, and J.-W. van Wingerden. Wind farm flow control: prospects and challenges. *Wind Energy Science*, 7(6):2271–2306, 2022.

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

[36] T.T. Nguyen and H.M. Kim. Cluster-based predictive PCC voltage control of large-scale offshore wind farm. *IEEE Access*, 9:4630–4641, 2021.

[37] S. Raach, D. Schlipf, F. Borisade, and P. W. Cheng. Wake redirecting using feedback control to improve the power output of wind farms. pages 1387–1392, 2016.

[38] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model predictive control: theory, computation, and design*. Nob Hill Publishing, Madison, Wisconsin, 2nd edition edition, 2017.

[39] A. Richards and J. P. How. Robust distributed model predictive control. *International Journal of Control*, 80(9):1517–1531, 2007.

[40] S. Riverso, S. Mancini, F. Sarzo, and G. Ferrari-Trecate. Model predictive controllers for reduction of mechanical fatigue in wind farms. *IEEE Transactions on Control Systems Technology*, 25(2):535–549, 2017.

[41] R. Scattolini. Architectures for distributed and hierarchical Model Predictive Control – A review. *Journal of Process Control*, 19(5):723–731, 2009.

[42] S. Siniscalchi-Minna, Fernando D. Bianchi, C. Ocampo-Martinez, J. L. Domínguez-García, and B. De Schutter. A non-centralized predictive control strategy for wind farm active power control: A wake-based partitioning approach. *Renewable Energy*, 150:656–669, 2020.

[43] V. Spudić, C. Conte, M. Baotić, and M. Morari. Cooperative distributed model predictive control for wind farms. *Optimal Control Applications and Methods*, 36(3):333–352, 2015.

[44] A. Sterle, C. A. Hans, and J. Raisch. Model predictive control of wakes for wind farm power tracking. *Journal of Physics: Conference Series*, 2767(3):032005, 2024.

[45] G. I. Taylor. The Spectrum of Turbulence. *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, 164(919):476–490, 1997.

[46] J.-W. van Wingerden, L. Pao, J. Aho, and P. Fleming. Active power control of waked wind farms. *IFAC-PapersOnLine*, 50(1):4484–4491, 2017.

[47] P. Veers, K. Dykes, E. Lantz, S. Barth, C. L. Bottasso, O. Carlson, A. Clifton, J. Green, P. Green, H. Holttinen, D. Laird, V. Lehtomäki, J. K. Lundquist, J. Manwell, M. Marquis, C. Meneveau, P. Moriarty, X. Munduate, M. Muskulus, J. Naughton, L. Pao, J. Paquette, J. Peinke, A. Robertson, J. Sanz Rodrigo, A. M. Sempreviva, J. C. Smith, A. Tuohy, and R. Wiser. Grand challenges in the science of wind energy. *Science*, 366(6464), 2019.

[48] H. Vogel. A better way to construct the sunflower head. *Mathematical Biosciences*, 44(3):179–189, 1979.

[49] G. Voronoi. Nouvelles applications des parametres continus à la theorie des formes quadratiques. Deuxiäm e Memoire . *Journal für die reine und angewandte Mathematik (Crelles Journal)*, pages 198–287, 1908.

[50] L. A. Wolsey. *Integer programming*. John Wiley & Sons, 2020.

[51] H. Zhao, Q. Wu, Q. Guo, H. Sun, and Y. Xue. Distributed model predictive control of a wind farm for optimal active power control part II: Implementation with clustering-based piece-wise affine wind turbine model. *IEEE Transactions on Sustainable Energy*, 6(3):840–849, 2015.

[52] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick. Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies. *IEEE Transactions on Control Systems Technology*, 25(3):899–910, 2017.

Glossary

List of Acronyms

ADMM	alternating direction method of multipliers
CFD	computational fluid dynamics
DDFG	dual decomposition with fast gradient
DMPC	distributed model predictive control
FLORIDyn	FLOw Redirection and Induction Dynamics
FLORIS	FLOw Redirection and Induction in Steady State
MPC	model predictive control
SOWFA	Simulator for Wind Farm Applications
SP-Wind	Sustainable Power-Wind
TSO	transmission system operator
MIQCQP	mixed-integer quadratically-constrained quadratic program
OP	observation point
CMPC	centralised model predictive control
NLP	nonlinear program
MHE	moving horizon estimation