Imperial College
London

Department of
AERONAUTICS

Wind farm optimisation via efficient wake modelling

Final Year Project

Author: Phakkhaphol (Peem) Limsupanark
CID: 01854920
Academic Supervisor: Dr. Andrew Wynn
Second Marker: Prof. Joaquim Piero
Department: Department of Aeronautics
Course: MEng Aeronautical Engineering
Date: June 3, 2024

Department of Aeronautics
South Kensington Campus
Imperial College London
London SW7 2AZ
U.K.

Abstract

This project formulates a wind farm wake model by incorporating effects of wake interactions into an existing single turbine wake model and performs a qualitative validation. In addition, this project also extends the existing Moving Horizon Estimation algorithm from only reconstructing the wake of a single turbine to being able to reconstruct the wake of a wind farm. Results of the extended Moving Horizon Estimation algorithm illustrates an accurate reconstruction of the wind farm wake flow field. Lastly, this project performs a preliminary design of a wake steering controller using a Model Predictive Control approach to maximise the power output of a wind farm given unsteady disturbances. The performance of the controller was evaluated with different disturbances and has displayed success in enhancing wind farm power in some scenarios. The weaknesses of the controller were discussed and improvements that could be made are also suggested. Overall, this project has made advancements in developing a robust, responsive and practical wake steering controller to improve the power output of a wind farm.

Acknowledgements

My heartfelt appreciation goes to the many individuals who have supported me in the completion of this project. I would like to express my sincere gratitude to my supervisor, Dr. Andrew Wynn, whose knowledge and expertise on the subject has greatly contributed to the completion of my dissertation. Also, I extend my heartfelt thanks to all my friends for their support throughout this project, both in challenging moments and joyful times. I would also like to express my gratitude to Ethos sport centre for providing a space for me to alleviate my stress during the project. Lastly, a special thanks to my family for their support throughout my time at Imperial.

Contents

List of Figures

List of Tables

List of Symbols

A	Turbine rotor area
a	Turbine axial induction factor
C_P	Power coefficient
C_T	Thrust coefficient
D	Turbine diameter
d_w	Normalised wake width
E_T	Total wind farm energy over horizon
g_r	Streamwise velocity deficit estimate
h_r	Wake centerline estimate
k_w	Wake expansion coefficient
l	Model Predictive Control horizon length
M	Moving Horizon Estimate horizon length
P	Wind turbine power
P_T	Wind farm power
p	Number of segments in the MPC horizon
$s_{i,k}$	Forcing function at time t_k
S_i	Turbine forcing function
t_k	Timestep k
U	Streamwise velocity
U_∞	Freestream wind speed
u_i	Streamwise velocity at turbine i
u_{ij}	Streamwise velocity at turbine i due to wake of turbine j
χ	Initial wake skew angle
W_t	Browning motion
$w_{1,k}$	Streamwise exogenous disturbance at time t_k
$w_{2,k}$	Transverse exogenous disturbance at time t_k
x	Downstream position
x_d	Turbine spacing
x_p	x coordinate of sensor
y_c	Wake centerline
y_p	y coordinate of sensor
z_h	Hub height
α_1, α_2	Wake center estimation optimisation weights
β_1, β_2	Streamwise velocity deficit estimation optimisation weights
γ	Turbine yaw angle
δu_1	Streamwise velocity deficit
δu_2	Transverse velocity deficit
$\delta u_1^{(0)}$	Initial streamwise velocity deficit
$\delta u_2^{(0)}$	Initial transverse velocity deficit
μ	Mean inflow velocity
ν	Diffusion parameter
ω	Mean adversion rate
ρ	Density of air
σ_0	Wake proportionality constant
σ_y	Lateral characteristic wake width
σ_z	Vertical characteristic wake width
θ	Wake skew angle

List of Acronyms

GS	Geometric Sum
LES	Large-Eddy Simulations
LSVD	Linear Superposition of Velocity Deficits
MHE	Moving Horizon Estimation
MPC	Model Predictive Control
PDE	Partial Differential Equation
SED	Sum of Energy Deficits
SMPC	Stochastic Model Predictive Control
SS	Sum of Squares

1 Introduction

1.1 Context & Motivation

The demand for wind energy has been increasing as the global community pursues a cleaner and more sustainable energy source to meet the Net Zero Emissions target by 2050; expecting approximately 7400 TWh of wind electricity generation by 2030. To achieve this goal, an average annual growth rate of around 17% is necessary, underscoring the need for larger scale wind farms [2]. However, these wind farms encounter significant challenges such as wake losses, which account for a 10% to 20% loss in annual energy production [3]. Turbulent flows from wind passing through the rotors of upstream wind turbines are transported downstream, decreasing both power output and the lifespan of wind turbines [3]. Addressing this issue is complex, and cannot merely be solved by increasing the space between wind turbines due to land and transmission line limitations [4].

With the goal of reducing wake losses to enhancing wind farm performance, two approaches have been extensively researched. The first approach is turbine layout optimisation, that involves placing wind turbines in an optimised manner to maximise annual power generation. The second approach involves wind farm control, which entails manipulating the wind field within the wind farm to maximise farm level power generation [5]. While static wake models are adequate for layout optimisation, a dynamic model is paramount to the design of a robust and effective wind farm controller [5].

Static wake models like FLORIS and 3DEG have been used to optimise static quantities, including mean power generation — but falls short in providing unsteady values like power fluctuations [5]. Because of this, leveraging dynamic models to utilise unsteady values presents an opportunity to further increase power generation through active yaw control — hence, the need for dynamic wake models. With the high demand for wind farms and limited availability of High-Performance Computing (HPC) clusters, there is a necessity for computationally efficient dynamic models in order to provide more responsive control.

With a dynamic model, a wind farm controller can be designed. One wind farm control technique to deal with wake loss is wake steering. Wake steering involves purposely yawing upstream wind turbines with the goal of redirecting their wakes from downstream turbines, as illustrated in Figure 1 [6]. This results in the downstream turbine operating in better flow conditions but reduces power output of the upstream turbine, creating a solvable optimisation problem to obtain the yaw angle at which total farm power is maximised. An experiment on wake steering conducted at a commercial wind farm led to a decrease in wake losses by 9.3%, this demonstrates the effectiveness of wake steering in increasing wind farm efficiency [6]. This ultimately motivates the need for an effective, practical and robust wake steering controller.

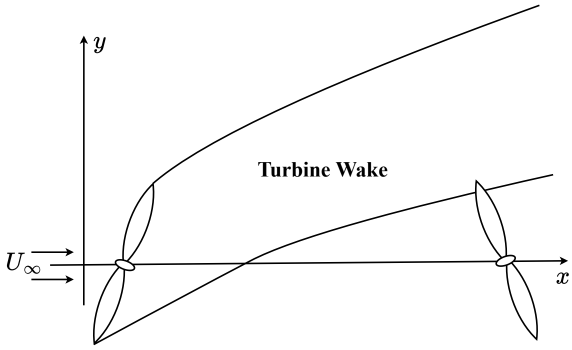

Figure 1: Wake steering mechanism

1.2 Aims & Objectives

The ultimate aim is to create a robust, responsive and practical wake steering controller that maximises the power output of a wind farm. To achieve this aim, various objectives must be fulfilled with relatively low computational costs to ensure that the controller can be used commercially. Due to time constraints, the scope of this project focuses on three main objectives listed below:

1. The development of a multiple turbine model from an existing single turbine wake model to enable the construction of a wind farm flow field.
2. The extension of an existing single turbine estimation algorithm to be able to estimate the wakes of multiple turbines to enable the reconstruction of a wind farm flow field with only sensor information.
3. The creation of an initial practical wake steering controller for a two-turbine configuration and suggest future improvements for the controller.

While the development of a multiple turbine model is sufficient for building a theoretical controller, it falls short of creating a practical controller due to limited information in realistic conditions. Hence, the need for the completion of the second objective that ensures the wind farm flow field can be reconstructed from sensor information. Therefore, the objectives mentioned above must be completed sequentially to ensure the construction of a practical controller. Ultimately, achieving these objectives will propel the advancement of creating a wake steering controller which maximises the power output of a commercial wind farm.

2 Literature Review

As wake steering strategies have been proven to increase wind power production [6], there have been many attempts to generate wake steering strategies for wind farms using various wind turbine wake models and control methods. Section 2 investigates different wind turbine models, estimation and control approaches that could be used to build a practical wake steering controller, discussing the benefits and drawbacks of each.

2.1 Wind turbine wake models

There are various wake modelling approaches that can be utilised to generate a wake steering controller, each with its own strength and weaknesses. These wake deficit models are usually combined with wake deflection and wake superposition models to represent the dynamics of a yawing wind turbine scenario [7].

2.1.1 Top-hat model

The Jensen wake velocity deficit model is a classic top-hat model that has been used in numerous literatures due to its simplicity ensuring low computational cost. The Jensen model which is derived from momentum balance, assumes a constant wake growth rate and a uniform streamwise velocity deficit [8]. The streamwise velocity deficit, δu_1, according to the Jensen model is given from Equation 2.1 [8].

$$\delta u_1 = \frac{1 - \sqrt{1 - C_T}}{\left(1 + 2k_w \frac{x}{D}\right)^2} \tag{2.1}$$

where x is the downstream position, D is the wind turbine diameter, C_T is the coefficient of thrust and k_w is the wake expansion coefficient. To include wind turbine yawing dynamics, the

Jensen wake velocity deficit model is used with the Jiménez wake deflection model [7]. The wake skew angle, θ, according to Jiménez wake deflection model is given from Equation 2.2 [9].

$$\theta = \frac{\cos^2 \gamma \sin \gamma \frac{C_T}{2}}{\left(1 + \beta \frac{x}{D}\right)^2} \tag{2.2}$$

where β is the wake growth rate with values typically ranging from 0.09 to 0.125 and γ is the turbine yaw angle [9].

While the Jensen model is capable of obtaining the correct order of magnitude of the velocity deficit, it cannot accurately capture the velocity profile [7]. In addition, due to the simple nature of top-hat models like the Jensen model, the effects of turbulence intensity levels are not considered [7]. Since the turbulence intensity level highly affect yaw control strategies, the use of the Jensen model for wind farm optimisation via wake steering is not recommended [7].

2.1.2 Gaussian-shaped wake deficit model

The Gaussian-shaped velocity deficit model developed by Bastankhah et al. assumes that the velocity profile in the far-wake region follows a Gaussian distribution. This assumption is valid for the far-wake region as flow achieves a self-similar profile [1]. The far-wake velocity for a yawed turbine derived by Bastankhah et al. is given as:

$$\frac{\delta \overline{u}}{U_\infty} = \left(1 - \sqrt{1 - \frac{C_T \cos(\gamma)}{8(\sigma_y \sigma_z / D^2)}}\right) e^{-0.5((y-\delta)/\sigma_y)^2} e^{-0.5((z-z_h)/\sigma_z)^2} \tag{2.3}$$

where $\delta \overline{u}$ is the wake velocity deficit, U_∞ is the freestream velocity, σ_y and σ_z are the characteristic wake widths in the lateral (y) and vertical (z) directions as, δ is the wake deflection, z_h is the hub height and γ is the yaw angle.

The results of Equation 2.3 agree well with experimental results in the far-wake region, with the velocity profile produced by the Gaussian model being far more accurate than the Jensen model [1]. This results in a large improvement in terms of wind turbine flow field predictions, showing that the Gaussian profile is a much more suitable model for developing a wake steering controller compared to the Jensen model [7].

2.2 Estimation algorithms

In realistic conditions, it is not feasible to place sensors everywhere in the wake with the goal of obtaining a full flow field in order to implement yaw steering strategies. Furthermore, the non-linear nature of wind turbine wakes prohibits a reconstruction of the full flow from sensor information alone, therefore a state estimation algorithm is required in combination with sensor information.

There are many state estimation algorithms available to be utilised, each offering specific advantages and disadvantages. One frequently used method is Kalman filtering, designed to estimate states of linear systems [10]. However, the nonlinear nature of the system requires a more generalised technique to be employed. Another method that can be used is the full information estimation method which solves a constrained optimisation problem using all past measurements to obtain state estimates. Despite being able to generate accurate state estimates, this method is unfeasible as computational burden increases as simulation time increases due to the need to store and process more past measurement data [11]. The Moving Horizon Estimation (MHE) method overcomes this computational burden associated with the full information estimation method. This method utilises a finite sliding estimating window (horizon) of past measurements instead of all past measurements, exponentially reducing the computational complexity while providing relatively accurate state estimates within the horizon. Figure 2 illustrates the finite sliding estimating window that is utilised in the MHE approach. Therefore, given its estimation performance and the computational costs involved, the MHE algorithm is the most suitable method for reconstructing wind turbine wakes with limited sensor information. A detailed discussion of the MHE algorithm formulation to reconstruct wind turbine wakes can be found in Section 3.2.

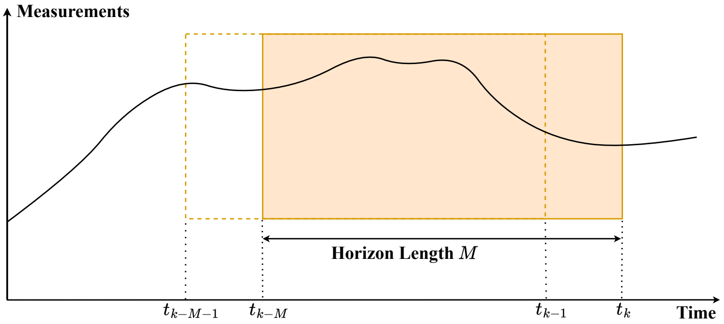

Figure 2: Finite sliding estimating window for MHE method

2.3 Control techniques

Wake steering involves controlling the yaw of each turbine with the goal of maximising the overall power output of the wind farm. This is essentially a control problem which can be solved with the use of a suitable control technique. Therefore, to find a suitable control technique for the purposes of wake steering, it is essential to explore typical control approaches that have the potential to be the foundation in constructing an effective wake steering controller.

One of the most widely used control techniques is Proportional-Integral-Derivative (PID) control. This technique employs a feedback control method where the controller adjusts its corrective inputs based on the error, defined as the discrepancy between the desired and the sensed output of the system [12]. The controller consists of three core elements: the proportional, the integral and the derivative term. The proportional term of the controller examines the current error, where a large error results in a large response and vice versa [13]. While having the proportional term may provide some basic control over the system, it is unable to provide further control over the system when the error is small, typically denoted as the steady-state error. The integral term is used to remove the steady-state error, as it examines the error over time and increases its contribution to the input over time [13]. Lastly, the derivative term examines the derivative of the error. This term thus contributes more to the input when there is a high rate of change, with the attempt to reduce future error [13]. Therefore, with the correct tuning values for these three terms, an effective controller can be built to facilitate the desired control over a system with external disturbances.

While being relatively simple, the PID control is not a suitable approach to be used to build a controller for wake steering due to many reasons. Firstly, PID controllers are typically single-input-single-output controllers thus are ineffective in controlling multi-variable control problems such as the wake steering control problem which require the simultaneous yaw control of multiple turbines. Additionally, PID controllers do not respond well to delayed systems and are usually applied to systems with instantaneous responses. In this situation, where the effect of yawing an upstream turbine on downstream turbines is delayed, PID controllers may provide poor control. Most importantly, as the desired goal of yaw control is to maximise power, PID controllers are not suited for this task. This is because PID controllers are designed to form inputs such that the difference between the desired output and actual output is minimised rather than performing optimisation. Therefore, lack the capability to determine the best control actions to maximise power. Instead, PID control only provides control inputs to achieve a predefined target, which may not correspond to the optimal configuration for power maximisation, thus is inadequate for the purposes of this project. Due to all the reasons stated above, PID control was not utilised to build a wake steering controller.

A more advanced control approach is the Model Predictive Control (MPC) technique. The MPC is an optimal-control based technique that selects control inputs to provide outputs to match a set point [14]. At each timestep, the MPC controller receives the current state of the system and predicts the future states using an accurate dynamic model of the system. It then calculates the optimal sequence of control actions required for the output to reach the set points over a prediction horizon by solving a constrained optimisation problem [14]. The controller then applies the first control input from this sequence and disregards the rest [14]. This process is then repeated at each timestep, continuously updating the control inputs to follow the best trajectory get to the set points over time. Figure 3 illustrates the basic concept for MPC. The set points for the MPC controller can be derived based on maximising a quantity of the system (e.g. power) using a suitable objective function [15]. Thus, using this derived set point, the MPC can deliver a sequence of control inputs to drive the system to perform optimally based on the objective function.

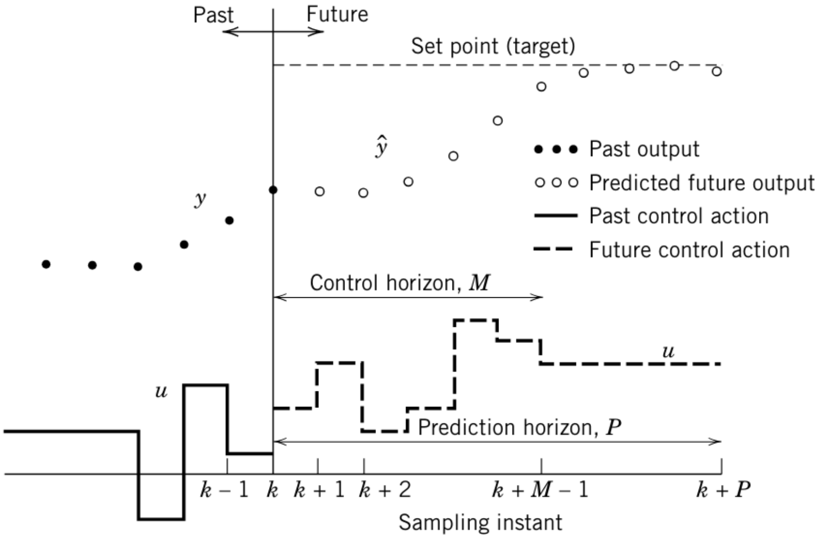

Figure 3: Basic concept for MPC [15]

Despite being harder to implement and requiring more computational power, the MPC technique is a appropriate approach to be used to construct a wake steering controller. This is because the MPC approach is designed to deal with multi-variable control problems, which is essential for wind farms where there is interaction with multiple turbines [16]. This allows the controller to simultaneously obtain the best collective control strategy for the wind farm. Additionally, the MPC technique involves predicting the future states over the horizon. This allows the MPC controller to be able to anticipate the effect of its current control inputs on future power production, ensuring maximum power production is achieved at all times. Most importantly, since the primary goal of wake steering is to maximise power output, the MPC technique is well-suited for this purpose. This is because the MPC approach allows set points to be derived based on the maximisation of an objective function, which can be the maximisation of power in this case [16]. This ensures that the control actions provided by the MPC controller can be directed towards the maximisation of power of the wind farm.

3 Background Theory

This Section details the derivations and concepts on the single turbine wake model and the estimation algorithm that was done in a previous Imperial report but is reiterated here to provide context for the rest of the project.

3.1 Wind turbine wake model

There are many models that have the ability to provide insight on the flow field of a wind turbine wake. However, for the purposes of this project, the model used must be: simple to ensure fast computation, able to respond to unsteady inflow conditions to model real world scenarios and take into account time-varying control inputs to be used for wake steering. Section 3.1 covers the basics of the model used and the adaptations made to satisfy the requirements of this project.

3.1.1 Model overview

As the Gaussian-shape wake deficit model is a suitable model to be used to build a wake steering controller as mentioned in Section 2.1.2, it will be used as the wind turbine wake model for this project. From Equation 2.3, the wake deflection and characteristic wake widths in the model given by Bastankhah et al. are highly non-linear and can lead to high computational times, therefore simplications must be made to this model.

Shapiro et al. proposed a more simplified wind turbine wake model that corresponds well with the aforementioned model and experimental results [17]. The model is broken down into two key regions: a near-wake region where flow is primarily inviscid and far-wake region in which flow becomes turbulent - thus an inviscid model is used for the near-wake region [17]. The streamwise velocity deficit in the near-wake region, δu_{01}, was derived by applying Bernoulli's equation and the momentum conservation principle in the near rotor disk region, while the transverse velocity deficit in the near-wake region, δu_{02}, was derived using Prandtl's lifting line theory.

$$\delta u_{01} = U_\infty \left(1 - \sqrt{1 - C_T \cos^2 \gamma}\right) \tag{3.1}$$

$$\delta u_{02} = \frac{1}{4} C_T U_\infty \cos^2 \gamma \sin \gamma \tag{3.2}$$

The far-wake model as proposed by Shapiro et al. was derived from Reynold's Average Navier-Stokes (RANS) equations and simplified to specifically capture the dynamics of the wake using velocity deficits. The simplifications that were made involved neglecting the viscous stress terms as well as linearising the advective term i.e. $u_j \partial / \partial x_j = U_\infty \partial / \partial x$ [17]. The velocity deficits can be found by solving Equation 3.3 [17].

$$\frac{\partial \delta u_i}{\partial t} + U_\infty \frac{\partial \delta u_i}{\partial x} = -2U_\infty \frac{d_w'(x)}{d_w(x)} \delta u_i(x,t) + S_i \delta(x) \tag{3.3}$$

where δu_i is the velocity deficit with subscript i representing the direction, $S_i = U_\infty \delta u_i$ represents the velocity deficit strengths determined from the inviscid model in the near-wake region and $\delta(x)$ is the Dirac delta function and d_w is the normalised wake width of a turbine while $d_w'(x)$ is its derivative [17]. The normalised wake width can be found using Equation 3.4.

$$d_w = 1 + k_w \ln \left(1 + e^{2x/D}\right) \tag{3.4}$$

where $0 < k_w < 1$ is the wake expansion coefficient, given as 0.0834 for an unyawed turbine wake [17]. The use of S_i with the Dirac delta function $\delta(x)$ sets the initial conditions for the development of the wake, localising the turbine impact on the flow to be precisely at the location of the turbine. Thus, the flow behind the turbine is merely the evolution of these initial conditions moving downstream. The wake centreline, $y_c(x,t)$, can be obtained by solving the Equation 3.5 [17].

$$\frac{\partial y_c}{\partial t} + U_\infty \frac{\partial y_c}{\partial x} = -\delta u_2(x,t) \tag{3.5}$$

Additionally, Shapiro et al. proposed steady state analytical solutions for velocity deficit and wake centre as seen in Equation 3.6 and Equation 3.7 [17].

$$\delta u(x) = \frac{\delta u_i}{d_w^2(x)} \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\Delta_w \sqrt{2}}\right)\right] \tag{3.6}$$

$$y_c(x) = \int_{-\infty}^{x} \frac{-\delta u_2(x')}{U_\infty} dx' \tag{3.7}$$

The steady state solution shows an excellent agreement to the experimental data on yawed turbine wakes [17]. However, a weakness of steady state solutions are their inability to capture the dynamics of perturbations. Despite its simple form, these steady state solution are unable to model time-dependent forcing terms as well as exogenous disturbances and hence are inadequate to be used for the purposes of this project. Therefore, a modified model is proposed to tackle these limitations. The modified solution to the velocity deficit and wake centre are assumed to be a superposition of the steady state solution and a time-dependent perturbation as seen in Equation 3.8 and Equation 3.9.

$$\delta u_1(x,t) = \delta\bar{u}_1(x) + \delta u_1(x,t) \tag{3.8}$$

$$y_c(x,t) = \bar{y}_c(x) + \delta y_c(x,t) \tag{3.9}$$

As the partial differential equations (PDEs) in Equation 3.3 and Equation 3.5 are linear, the perturbation states must also satisfy the same PDEs. Therefore, the solution to the states can be viewed as a perturbation from the steady solution. In the steady state case, the PDEs in Equation 3.3 and Equation 3.5 reduce to ordinary differential equations as seen below.

$$U_\infty \frac{d\delta\bar{u}_i}{\partial x} = -2U_\infty \frac{d'_w(x)}{d_w(x)} \delta\bar{u}_j + \bar{S}_j \delta(x) \tag{3.10}$$

$$U_\infty \frac{d\bar{y}_c}{dx} = -\delta u_2 \tag{3.11}$$

Where the overbar represents the time-average component. The steady state solutions are given as follows:

$$\delta\bar{u}_j(x) = \begin{cases} \frac{S_i}{U_\infty} \frac{d_w(0)^2}{d_w(x)^2}, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{3.12}$$

$$\bar{y}_c(x) = -\frac{S_2}{U_\infty^2} \int_0^x \left(\frac{d_x(0)}{d_x(\xi)}\right)^2 d\xi \tag{3.13}$$

Thus, the solution to the velocity deficit with perturbation is given as:

$$\delta u_j(x,y) = \frac{\delta u_j^{(0)}(x - U_\infty t) d_w(x - U_\infty t)^2}{d_w(x)^2} + \frac{S_j\left(t - \frac{x}{U_\infty}\right)}{U_\infty} \left(\frac{d_w(0)}{d_w(x)}\right)^2 \mathcal{X}_{[0,U_\infty t]}(x) \tag{3.14}$$

where $\delta u_j^{(0)}$ represents the initial velocity deficit and $\mathcal{X}_{[0,U_\infty t]}$ is the characteristic function defined as:

$$\mathcal{X}_{[0,U_\infty t]}(x) = \begin{cases} 1, & 0 \leq x \leq U_{\mathrm{ot}} \\ 0, & \text{otherwise} \end{cases} \tag{3.15}$$

This physically implies that the forcing history, $S_j(\tau)_{0<\tau<t}$ only affects the wake up to the distance $U_\infty t$ from the turbine. Using the solution of $u_2(x,t)$, the analytical solution of the wake centre with perturbation, $y_c(x,t)$, can be found to be:

$$y_c(x,t) = -\frac{1}{U_\infty}\frac{V_w(x)}{d_w(0)^2}\left[\delta u_2^{(0)}(x-U_\infty t)d_w(x-U_\infty t)^2 + \frac{1}{U_\infty}S_2\left(t-\frac{x}{U_\infty}\right)\right]\mathcal{X}_{[0,U_\infty t]}(x) \quad (3.16)$$

where $V_w(x) = \int_0^x (d_x(0)/d_x(\xi))^2 d\xi$. The boundary conditions and initial conditions imposed when solving the PDE are stated below:

$$y_c(0,t) = 0 \quad (3.17)$$

$$y_c(x,0) = 0 \quad (3.18)$$

With the ability to calculate the velocity deficit and the wake centre, the streamwise velocity of the wake behind a wind turbine can be modelled using Equation 3.19 [17].

$$U(x,y,t) = U_\infty - \delta u_1(x,t)\frac{D^2}{8\sigma_0^2}\exp\left(-\frac{(y-y_c)^2}{2\sigma^2(x)}\right) \quad (3.19)$$

where $\sigma(x) = \sigma_0 d_w(x)$ and σ_0 represents a wake proportionality constant $\sigma_0/D = 0.235$ for unyawed turbines [17]. The solutions in Equation 3.14 and Equation 3.16 are in continuous time, however to proceed with both the estimation problem and the construction of a wake steering controller, a discrete time model is required.

3.1.2 Discretisation of the model

By exploiting the simple mathematical form of the continuous model, a discrete time model can be constructed. To discretise the model, two state vectors are introduced for the streamwise velocity deficit, $\boldsymbol{z}_k$, and wake centre, $\boldsymbol{y}_k$. These vectors consist of the streamwise velocity deficit, δu_1, and wake centre, y_c, at N uniformly discretised points along the x axis at time t_k. These vectors can be written as:

$$\boldsymbol{z}_k = \begin{pmatrix} \delta u_1(x_1,t_k) \\ \delta u_1(x_2,t_k) \\ \vdots \\ \delta u_1(x_N,t_k) \end{pmatrix}, \quad \boldsymbol{y}_k = \begin{pmatrix} y_c(x_1,t_k) \\ y_c(x_2,t_k) \\ \vdots \\ y_c(x_N,t_k) \end{pmatrix} \quad (3.20)$$

As there is a coupling between time and distance through U_∞, the timestep in the model is related to the spatial grid as seen in Equation 3.21.

$$\Delta t = \frac{\Delta x}{U_\infty} \quad (3.21)$$

The discrete time model equations for the streamwise velocity deficit and wake centre are given as:

$$\boldsymbol{z}_{k+1} = A_1\boldsymbol{z}_k + E_1 w_{1,k} + B_1 s_{1,k} \quad (3.22)$$

$$\boldsymbol{y}_{k+1} = A_c\boldsymbol{y}_k + E_c w_{2,k} + B_c s_{2,k} \quad (3.23)$$

where $s_{i,k}$ represents the known forcing function at time t_k i.e. $s_{i,k} = S_i(t_k)$ and $w_{i,k}$ represents the exogenous disturbance at time t_k. The exogenous disturbance, $w_{i,k}$, can be found using Equation 3.24.

$$w_{i,k} = \delta u_i^{(0)}(-t_k U_\infty)\frac{d_w(-t_k U_\infty)^2}{d_w(0)^2} \quad (3.24)$$

The matrices within the discrete time model equation for streamwise velocity deficit are defined as:

$$A_1 = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ d_1 & 0 & \cdots & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{N-1} & 0 \end{pmatrix} \in \mathbb{R}^{N\times N}, \quad E_1 = \begin{pmatrix} d_0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{N}, \quad B_1 = \begin{pmatrix} \frac{d_0}{U_\infty} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{N} \tag{3.25}$$

where d_k can be calculated using Equation 3.26.

$$d_k = \left(\frac{d_w(x_k)}{d_w(x_{k+1})} \right)^2 \tag{3.26}$$

Additionally, the matrices within the discrete time model equation for wake centre are expressed as:

$$A_c = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ \nu_1 & 0 & \cdots & \cdots & 0 \\ 0 & \nu_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \nu_{N-1} & 0 \end{pmatrix}, \quad E_c = \begin{pmatrix} \frac{-V_w(x_1)}{U_\infty} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}, \quad B_c = \begin{pmatrix} \frac{-V_w(x_1)}{U_\infty^2} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \tag{3.27}$$

where ν_k is defined as:

$$\nu_k = \frac{V_w(x_{k+1})}{V_w(x_k)} \tag{3.28}$$

The purpose of the A matrix is to transport the states from grid point x_n to x_{n+1} after each timestep. Matrices E and B introduce the exogenous disturbances and control inputs into the system at the first grid point behind the turbine, x_1. Therefore, by implementing this discrete time model equations iteratively, the future states of the wake can be calculated using the present states, the known control inputs and external disturbances.

As the discrete time model is relatively simple and takes into account unsteady inflow conditions and responds to time-varying control inputs; it is suitable to be used for the purposes of this project.

3.2 Moving Horizon Estimation (MHE) Algorithm

As aforementioned, state estimation algorithm such as the MHE algorithm will be used in combination with sensor information to reconstruct the wake. From Equation 3.19, $U(x, y, t)$ comprises of a nonlinear combination of the velocity deficit, $\delta u_1(x, t)$, and wake centre, $y_c(x, t)$. Due to this nonlinear relationship, it is impossible to differentiate whether an increase in $U(x, y, t)$ is an effect of a decrease in velocity deficit, $\delta u_1(x, t)$, or a change in wake centre, $y_c(x, t)$, using only sensor information at $U(x_p, \pm y_p, t)$. To supplement this lack of information, the past history of the sensor measurements are considered and used in combination with current sensor measurements. With this extra information, the wake can be accurately reconstructed. This Section details the formulation of the MHE algorithm for the wind turbine wake system.

3.2.1 MHE algorithm objective

Given the horizon length of $M + 1$, the past measurements available at time t_k are the measurements from times, $t_{k-M}, t_{k-M+1}, \ldots, t_k$. The objective of the MHE algorithm is to provide estimates of the wake states at time t_k by using the historical measurements available, the control inputs at time t_k and the estimated wake states at time t_{k-1}. Therefore, by iteratively running the MHE algorithm, the dynamic estimate of the wake can be obtained.

3.2.2 Modifications from the discrete model

From the linear recursive nature of Equation 3.23, it is possible to define the states at any arbitrary time in the future using the value of the initial states, future exogenous disturbances and future control inputs. Equation 3.29 represent the expressions for the states at time t_k.

$$
\begin{aligned}
\boldsymbol{z}_k &= A_1^k \boldsymbol{z}_0 + \sum_{j=1}^{k} \left[A_1^{j-1} E_1 w_{1,k-j} + A_1^{j-1} B_1 S_1(t_{k-j}) \right] \\
\boldsymbol{y}_k &= A_c^k \boldsymbol{y}_0 + \sum_{j=1}^{k} \left[A_c^{j-1} E_c w_{2,k-j} + A_c^{j-1} B_c S_2(t_{k-j}) \right]
\end{aligned}
\tag{3.29}
$$

Thus, every state in the horizon can be found using the earliest states in the window; $\boldsymbol{z}_{k-M}$ and $\boldsymbol{y}_{k-M}$, the exogenous disturbances and control inputs within the horizon. For ease of coding, the vectors, $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$, are defined which contain the earliest state in the window and the exogenous disturbance within the horizon. Additionally, the vector, $\boldsymbol{s}_i$, which contains the control inputs within the horizon are also defined.

$$
\boldsymbol{\xi} = \begin{pmatrix} \boldsymbol{z}_{k-M} \\ w_{1,k-M} \\ w_{1,k-M+1} \\ \vdots \\ w_{1,k-1} \end{pmatrix}, \quad
\boldsymbol{\zeta} = \begin{pmatrix} \boldsymbol{y}_{k-M} \\ w_{2,k-M} \\ w_{2,k-M+1} \\ \vdots \\ w_{2,k-1} \end{pmatrix}, \quad
\boldsymbol{s}_i = \begin{pmatrix} s_{i,k-M} \\ s_{i,k-M+1} \\ s_{i,k-M+2} \\ \vdots \\ s_{i,k-1} \end{pmatrix}
\tag{3.30}
$$

Thus, using vectors, $\boldsymbol{\xi}$, $\boldsymbol{\zeta}$ and $\boldsymbol{s}_i$, Equation 3.29 can be modified to express all the states in the horizon as seen below:

$$
\begin{pmatrix} \boldsymbol{z}_{k-M} \\ \boldsymbol{z}_{k-M+1} \\ \vdots \\ \boldsymbol{z}_{k-1} \end{pmatrix} = \mathcal{A}_{1,M} \boldsymbol{\xi} + \mathcal{B}_{1,M} \boldsymbol{s}_1, \quad
\begin{pmatrix} \boldsymbol{y}_{k-M} \\ \boldsymbol{y}_{k-M+1} \\ \vdots \\ \boldsymbol{y}_{k-1} \end{pmatrix} = \mathcal{A}_{c,M} \boldsymbol{\zeta} + \mathcal{B}_{c,M} \boldsymbol{s}_2
\tag{3.31}
$$

where $\mathcal{A}_{1,M}$ and $\mathcal{B}_{1,M}$ are defined as:

$$
\mathcal{A}_{1,M} = \begin{pmatrix}
\boldsymbol{I}_N & \mathbf{0}_{N\times M} & & & & \\
A_1 & E_1 & \mathbf{0}_{N\times(M-1)} & & & \\
A_1^2 & A_1 E_1 & E_1 & \mathbf{0}_{N\times M-2} & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \\
A_1^{M-1} & A_1^{M-2} E_1 & A_1^{M-3} E_1 & \cdots & E_1 & \mathbf{0}_{N\times 1} \\
A_1^{M-1} & A_1^{M-2} E_1 & A_1^{M-3} E_1 & \cdots & A_1 E_1 & E_1
\end{pmatrix} \in \mathbb{R}^{N(M+1)\times(N+M)}
\tag{3.32}
$$

$$
\mathcal{B}_{1,M} = \begin{pmatrix}
\mathbf{0}_{N\times M} & & & & \\
B_1 & \mathbf{0}_{N\times(M-1)} & & & \\
A_1 B_1 & B_1 & \mathbf{0}_{N\times M-2} & & \\
\vdots & \vdots & \vdots & \ddots & \\
A_1^{M-2} B_1 & A_1^{M-3} B_1 & \cdots & B_1 & \mathbf{0}_{N\times 1} \\
A_1^{M-1} B_1 & A_1^{M-2} B_1 & \cdots & A_1 B_1 & B_1
\end{pmatrix} \in \mathbb{R}^{N(M+1)\times M}
\tag{3.33}
$$

and likewise, $\mathcal{A}_{c,M}$ and $\mathcal{B}_{c,M}$ are given as:

$$\mathcal{A}_{c,M} = \begin{pmatrix} \boldsymbol{I}_N & \mathbf{0}_{N\times M} & & & & \\ A_c & E_c & \mathbf{0}_{N\times(M-1)} & & & \\ A_c^2 & A_cE_c & E_c & \mathbf{0}_{N\times M-2} & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ A_c^{M-1} & A_c^{M-2}E_c & A_c^{M-3}E_c & \cdots & E_c & \mathbf{0}_{N\times 1} \\ A_c^{M-1} & A_c^{M-2}E_c & A_c^{M-3}E_c & \cdots & A_cE_c & E_c \end{pmatrix} \in \mathbb{R}^{N(M+1)\times(N+M)} \tag{3.34}$$

$$\mathcal{B}_{c,M} = \begin{pmatrix} \mathbf{0}_{N\times M} & & & & \\ B_c & \mathbf{0}_{N\times(M-1)} & & & \\ A_cB_c & B_c & \mathbf{0}_{N\times M-2} & & \\ \vdots & \vdots & \vdots & \ddots & \\ A_c^{M-2}B_c & A_c^{M-3}B_c & \cdots & B_c & \mathbf{0}_{N\times 1} \\ A_c^{M-1}B_c & A_c^{M-2}B_c & \cdots & A_cB_c & B_c \end{pmatrix} \in \mathbb{R}^{N(M+1)\times M} \tag{3.35}$$

3.2.3 Initialisation

The first timestep where $M+1$ past measurements are available is at time t_{M+1} which will be the starting time for the MHE algorithm. Therefore, for the MHE algorithm to compute state estimates for time t_{M+1}, it is necessary to initialise state estimates at time t_M. A reasonable guess for the state estimates, $\hat{\boldsymbol{z}}_{M+1}$ and $\hat{\boldsymbol{y}}_{M+1}$, are the steady states solutions in Equation 3.12 and Equation 3.13. These state estimations can be computed using the known control inputs at time t_M. After finding the first state estimates, the MHE algorithm is able to compute the states estimates of the next timestep using the states estimate from the previous timestep.

3.2.4 Sensor measurements and locations

The sensors in this case would provide measurements of the streamwise velocity behind the wake, $U(x_p, y_p, t)$, where (x_p, y_p) are the coordinates of the sensors. Due to the mathematical structure of the system, placing two sensors at locations (x_p, y_p) and $(x_p, -y_p)$ where x_p is an arbitrary coordinate behind the wind turbine and y_p is a coordinate that lie within the wake is most convenient. This is because using measurement data at these locations allows symmetry to be exploited resulting in an optimisation problem which has a known analytical optimal solution, eliminating the need for an optimiser algorithm which is a computational burden. Figure 4 demonstrates the setup of sensors relative to the wind turbine.

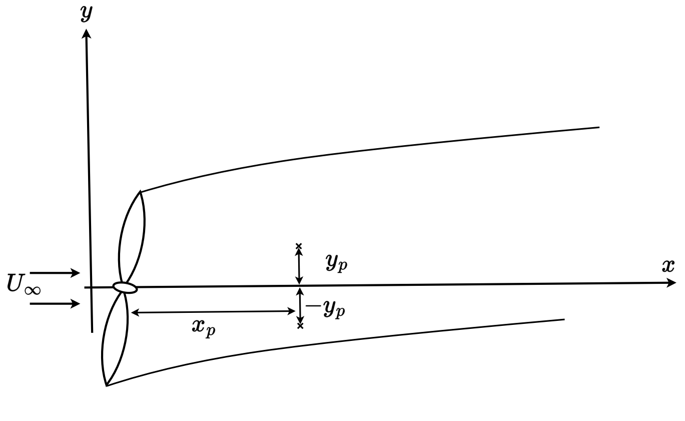

Figure 4: Sensor setup where sensors are represented as x

It is evident that this sensor configuration may not result in the most accurate estimates due to the limited number of sensors utilised. Further discussion and work on utilising more sensors will be done in [18].

As mentioned, the sensors measure the velocity, $U(x_p, \pm y_p, t_k)$, given as:

$$U(x_p, \pm y_p, t_k) = U_\infty - \delta u_1(x_p, t)\frac{D^2}{8\sigma_0^2}\exp\left(-\frac{(\pm y_p - y_c(x_p, t))^2}{2\sigma_0^2 d_w^2(x_p)}\right) \tag{3.36}$$

For convenience, the sensor locations are treated as points on the spatial grid. Therefore, a measurement matrix, $C_p \in \mathbb{R}^{1\times N}$, is established to extract the states at the sensor location. The measurement matrix is defined as zero everywhere, except at the location of the sensor where it is set to unity. Therefore, the states at the location of the sensor are given as:

$$\begin{aligned} \delta u_1(x_p, t_k) &= C_p \boldsymbol{z}_k \\ y_c(x_p, t) &= C_p \boldsymbol{y}_k \end{aligned} \tag{3.37}$$

Therefore, Equation 3.36 can be rewritten in terms of the states, $\boldsymbol{z}_k$ and $\boldsymbol{y}_k$:

$$U(x_p, \pm y_p, t_k) = U_\infty - C_p \boldsymbol{z}_k \frac{D^2}{8\sigma_0^2}\exp\left(-\frac{(\pm y_p - C_p \boldsymbol{y}_k)^2}{2\sigma_0^2 d_w^2(x_p)}\right) \tag{3.38}$$

3.2.5 State Estimation

As the available measurements at time t_k are:

- Measurements of streamwise velocity, $U(x_p, \pm y_p, t_r)$ for $k - M \leq r \leq k$
- The known inputs, $S_1(t_r)$ and $S_2(t_r)$ for $k - M \leq r \leq k$
- State estimates, $\hat{\boldsymbol{z}}(r; k-1)$ and $\hat{\boldsymbol{y}}(r; k-1)$ for $k - M - 1 \leq r \leq k - 1$ which are calculated from the previous timestep t_{k-1}

It is important to note the notation used to describe the states, where $\hat{\boldsymbol{z}}(r; k)$ is the estimated state $\hat{\boldsymbol{z}}$ at time t_r computed using the information available at time t_k.

Exploiting the fact that the sensors are located the same distance downstream, a wake centre estimate can be found. Using all the measurements of the streamwise velocity, the initial wake centre estimate, h_r, is given as:

$$h_r = \frac{\hat{\sigma}_0^2 \hat{d}_w(x_p)^2}{2y_p}\ln\left(\frac{\hat{U}_\infty - U(x_p, y_p, t_r)}{\hat{U}_\infty - U(x_p, -y_p, t_r)}\right), \quad k - M \leq r \leq k \tag{3.39}$$

where $\hat{}$ denotes the estimates of these parameters. From the initial estimate of the wake centreline, the wake centreline optimisation problem can be formulated. The optimisation problem is given as:

$$\begin{aligned} \underset{(\boldsymbol{\xi}_r), (w_{2,r})}{\text{minimize}} \quad & \alpha_1 ||\boldsymbol{\xi}_{k-M} - \overline{\boldsymbol{\xi}}_{k-M}||_2^2 + \alpha_2 \sum_{r=k-M}^{k} w_{2,r} + \sum_{r=k-M}^{k} |h_r - C_p \boldsymbol{\xi}_r|^2 \\ \text{subject to} \quad & \boldsymbol{\xi}_{r+1} = A_c \boldsymbol{\xi}_r + E_c w_{2,r} + B_c S_2(t_r), \qquad r = k - M, \ldots, k-1, \\ & \overline{\boldsymbol{\xi}}_{k-M} = \hat{\boldsymbol{y}}(k - M; k - 1), \\ & \boldsymbol{\xi}_r \in \mathbb{R}^N \qquad r = k - M, \ldots, k, \\ & w_{2,r} \in \mathbb{R} \qquad r = k - M, \ldots, k - 1 \end{aligned} \tag{3.40}$$

This optimisation problem can be broken down as follows:

1. The first term, $\alpha_1||\boldsymbol{\xi}_{k-M} - \overline{\boldsymbol{\xi}}_{k-M}||_2^2$, penalises large deviations between the estimated wake centre states from the previous timestep. The weight term, α_1, determines the how much the current states estimates can differ from the previous state estimates. A relatively small value of α_1 will permit the current state estimates to differ from the previous estimated states by a large amount and vice versa. Therefore, α_1 can be used to tune the variability of the estimates.
2. The second term, $\alpha_2 \sum_{r=k-M}^{k} w_{2,r}$, penalises the magnitude of the estimated transverse disturbance, such that the estimation of disturbances are not too large. The weight α_2 can be used to tune the magnitude of estimated disturbance.
3. The third term, $\sum_{r=k-M}^{k} |h_r - C_p \boldsymbol{\xi}_r|^2$, estimates the wake centreline at the sensor location. This is done by penalising the difference between the value of the wake centreline obtained from the sensor and the estimated wake centreline value at the sensor location.

Conveniently, Equation 3.40 can be expressed in a least squares form, which has an analytical optimal solution [19]. The Least Squares form of Equation 3.40 is given as:

$$\begin{aligned} \underset{\boldsymbol{\xi}}{\text{minimize}} \quad & ||Y\boldsymbol{\xi} - b||_2^2 \\ \text{subject to} \quad & \boldsymbol{\xi} \in \mathbb{R}^{N+M} \end{aligned} \tag{3.41}$$

where the matrix Y can be expressed as:

$$Y = \begin{pmatrix} \alpha_1 \boldsymbol{I}_N & : & \boldsymbol{0}_{N\times M} \\ \boldsymbol{0}_{M\times N} & : & \alpha_2 \boldsymbol{I}_M \\ & C_M \mathcal{A}_{c,M} & \end{pmatrix} \in \mathbb{R}^{(N+M+1)\times(N+M)} \tag{3.42}$$

with C_M given as:

$$C_M = \begin{pmatrix} C_p & 0 & \dots & \dots & 0 \\ 0 & C_p & 0 & \dots & 0 \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ 0 & 0 & \dots & 0 & C_p \end{pmatrix} \in \mathbb{R}^{(M+1)\times N(M+1)} \tag{3.43}$$

and the vector b defined as:

$$b = \begin{bmatrix} \alpha_1 \overline{\boldsymbol{\xi}}_{k-M} \\ h_{k-M} \\ \vdots \\ h_k \end{bmatrix} - \begin{bmatrix} \boldsymbol{0}_{N\times 1} \\ \boldsymbol{0}_{M\times 1} \\ C_M \mathcal{B}_{c,M} \boldsymbol{s}_2 \end{bmatrix} \in \mathbb{R}^{N+2M+1} \tag{3.44}$$

The analytical solution, $\boldsymbol{\xi}^*$, to Equation 3.41 is given as:

$$\boldsymbol{\xi}^* = (Y^\intercal Y)^{-1} Y^\intercal b \tag{3.45}$$

Thus, the final solution of the estimated wake centreline can be expressed by substituting the optimal solution into Equation 3.31.

$$\begin{pmatrix} \hat{\boldsymbol{y}}(k-M;k) \\ \hat{\boldsymbol{y}}(k-M+1;k) \\ \vdots \\ \hat{\boldsymbol{y}}(k;k) \end{pmatrix} = \mathcal{A}_{c,M} \left[(Y^\intercal Y)^{-1} Y^\intercal b\right] + \mathcal{B}_{c,M} s_2 \tag{3.46}$$

After estimating the wake centreline state, the streamwise velocity deficit state can be estimated in a similar fashion. By rearranging Equation 3.19, the estimated streamwise velocity deficit at the sensor locations can be found.

$$g_r = \frac{8\sigma_0^2}{D^2}\left(U_\infty - U(x_p, y_p, t_r)\right)\exp\left(\frac{1}{2}\left[\frac{y_p - C_p\hat{\boldsymbol{y}}(r;k)}{\sigma_0 d_w(x_p)}\right]^2\right) \tag{3.47}$$

Given that estimates of $\hat{\boldsymbol{y}}(r;k)$ are accurate, it follows that $g_r \approx C_p\hat{\boldsymbol{z}}_r$. Therefore, the estimated streamwise velocity deficit can be found by solving the optimisation problem in Equation 3.48

$$\begin{aligned}
\underset{(\boldsymbol{\zeta}_r),\,(w_{2,r})}{\text{minimize}} \quad & \beta_1||\boldsymbol{\zeta}_{k-M} - \overline{\boldsymbol{\zeta}}_{k-M}||_2^2 + \beta_2\sum_{r=k-M}^{k} w_{1,r} + \sum_{r=k-M}^{k} |g_r - C_p\boldsymbol{\zeta}_r|^2 \\
\text{subject to} \quad & \boldsymbol{\zeta}_{r+1} = A_1\boldsymbol{\zeta}_r + E_1 w_{1,r} + B_1 S_1(t_r), \qquad r = k-M, \dots, k-1, \\
& \overline{\boldsymbol{\zeta}}_{k-M} = \hat{\boldsymbol{z}}(k-M; k-1), \\
& \boldsymbol{\zeta}_r \in \mathbb{R}^N \qquad r = k-M, \dots, k, \\
& w_{1,r} \in \mathbb{R} \qquad r = k-M, \dots, k-1
\end{aligned} \tag{3.48}$$

Similarly, this optimisation problem can be expressed in a least squares form as seen in Equation 3.49.

$$\begin{aligned}
\underset{\boldsymbol{\zeta}}{\text{minimize}} \quad & ||Z\boldsymbol{\zeta} - c||_2^2 \\
\text{subject to} \quad & \boldsymbol{\zeta} \in \mathbb{R}^{N+M}
\end{aligned} \tag{3.49}$$

where the matrix Z can be expressed as:

$$Z = \begin{pmatrix} \beta_1\boldsymbol{I}_N & : & \boldsymbol{0}_{N\times M} \\ \boldsymbol{0}_{M\times N} & : & \beta_2\boldsymbol{I}_M \\ & C_M\mathcal{A}_{1,M} & \end{pmatrix} \in \mathbb{R}^{(N+M+1)\times(N+M)} \tag{3.50}$$

and the vector c given as:

$$c = \begin{bmatrix} \beta_1\overline{\boldsymbol{\zeta}}_{k-M} \\ g_{k-M} \\ \vdots \\ g_k \end{bmatrix} - \begin{bmatrix} \boldsymbol{0}_{N\times 1} \\ \boldsymbol{0}_{M\times 1} \\ C_M\mathcal{B}_{1,M}\boldsymbol{s}_1 \end{bmatrix} \in \mathbb{R}^{N+2M+1} \tag{3.51}$$

Likewise to Equation 3.45, the analytical solution, $\boldsymbol{\zeta}^*$, to Equation 3.49 is given as:

$$\zeta^* = (Z^\intercal Z)^{-1}Z^\intercal c \tag{3.52}$$

Equation 3.53 gives the solution to the estimated streamwise deficit.

$$\begin{pmatrix} \hat{\boldsymbol{z}}(k-M;k) \\ \hat{\boldsymbol{z}}(k-M+1;k) \\ \vdots \\ \hat{\boldsymbol{z}}(k;k) \end{pmatrix} = \mathcal{A}_{1,M}\left[(Z^\intercal Z)^{-1}Z^\intercal c\right] + \mathcal{B}_{1,M}s_1 \tag{3.53}$$

With knowledge of the estimated wake centreline and streamwise velocity deficit states at time t_k from the MHE algorithm, the full wake estimate can be computed using Equation 3.54.

$$\hat{U}(x, y, t_k) = \hat{U}_\infty - \hat{\boldsymbol{z}}(k;k)\frac{D^2}{8\hat{\sigma}_0^2}\exp\left(-\frac{(y - \hat{\boldsymbol{y}}(k;k))^2}{2\hat{\sigma}_0^2\hat{d}_w^2(x)}\right) \tag{3.54}$$

3.3 Results and Validation

Before extending the discretised model for the purposes of wake steering, it is crucial to compare and validate the flow field against experimental data or other validated models to confirm the correctness of the model. Additionally, the results from the MHE algorithm can be obtained by inputting streamwise velocity at the location of the sensors. As real measurement data with the corresponding full flow field from Large-Eddy Simulations (LES) are not available, the flow field generated by the discretised model was used as a replacement given that flow field produced corresponds to the experimental results from Bastankhah et al.

3.3.1 Discrete model

The validation of the discretised model as mentioned in Section 3.1.2 will be split into two parts. Firstly, the validation of the model behaviour with different yaw angle inputs against the experimental results from Bastankhah et al. and the model proposed by Shapiro et al., by using steady inflow conditions to clearly observe the wake deflection due to turbine yaw. Secondly, the model will also be validated using unsteady inflow conditions. Due to the absence of data on unsteady inflow conditions with the corresponding flow field obtained from other validated models. The validation will only involve investigating whether the disturbances are properly transported downstream at the speed of the incoming freestream velocity.

For the purposes of a fair comparison, the wake proportionality constant σ_0, and wake expansion coefficient, k_w, where set to coincide with the values used by Shapiro et al. as seen in Table 1 [17].

Parameter	**Value**
σ_0	0.235
k_w	0.0834

Table 1: Parameters used to validate the model [17]

Figure 5 illustrates the results from the discretised model as discussed in Section 3.1.2 with different yaw angles and steady inflow conditions. Additionally, Figure 6 displays the experimental results obtained by Bastankhah et al. and the proposed model introduced by Shapiro et al. as discussed in Section 3.1.1. Figure 6 also presents the wake centreline obtained using various methods. The dotted line represents the experimentally measured centrelines of the wake, the solid line are the centrelines obtained from the model proposed by Jiménez et al. and the dashed line are wake centrelines from the model from Shapiro et al..

As seen from Figure 5 and Figure 6, it is clear that the wakes from the discretised model respond similarly to turbine yaw when compared with experimental results from Bastakhah et al. and the proposed model by Shapiro et al.. Therefore it can be deduced that the discretisation adaptations made from the model proposed by Shapiro et al. are correct and thus can be extended for the purposes of wake steering.

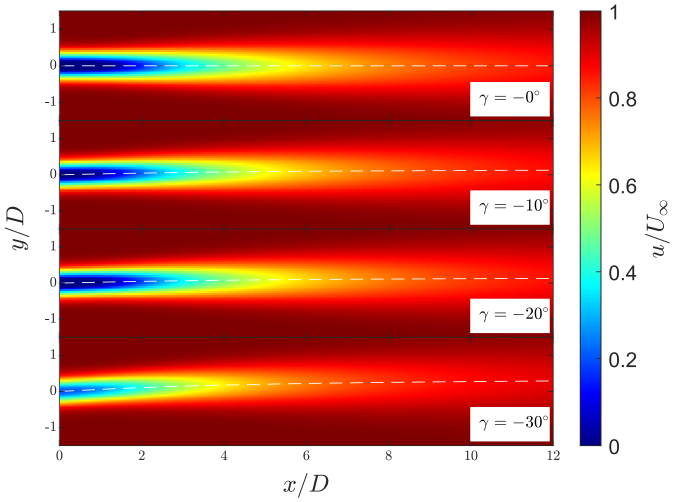

Figure 5: Contour plots of normalised streamwise velocity at hub height for different yaw angles with steady inflow conditions generated using discretised model.

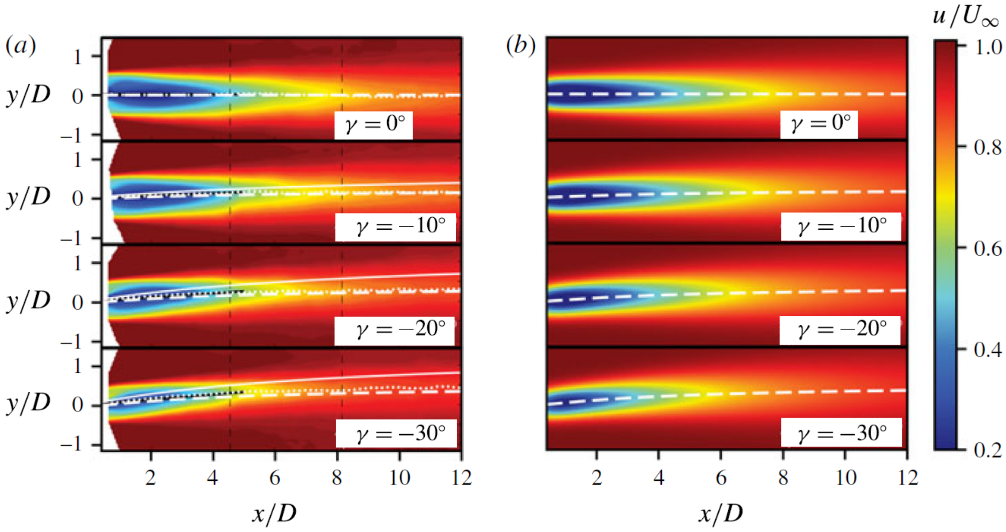

Figure 6: Contour plots of normalised streamwise velocity for different yaw angles (a) using experimental results from Bastakhah et al. [1] (b) calculated using model proposed by Shapiro et al. [17].

In addition to the model correctly responding to turbine yaw, it is also crucial that it correctly transports disturbances downstream to ensure an effective wake steering controller to be built. To investigate the propagation of disturbances, unsteady inflow velocities were generated and fed into the discretised model. The synthetic inflow disturbances were generated using the Ornstein-Uhlenbeck process which is a type of stochastic differential equation that is known to be effective in modelling random wind disturbances [20]. Equation 3.55 is utilised to obtain random disturbances based on the Ornstein-Uhlenbeck process [20],

$$\delta u_i^0(x-1) = \delta u_i^0(x) + \omega \left(\mu - u_i^0(x)\right) dt + \nu W_t \tag{3.55}$$

where δu_i^0 is the initial disturbance, ω is the mean adversion rate, μ is the mean inflow velocity, dt is the time increment, ν is the diffusion parameter and W_t is the Browning motion. The mean adversion rate and diffusion parameter were tuned such that inflow disturbances generated are realistic for the timescale used and that the disturbances do not grow unbounded. Figure 7 illustrates the streamwise and transverse inflow velocities used to generate an unsteady wake flow field with the discretised model.

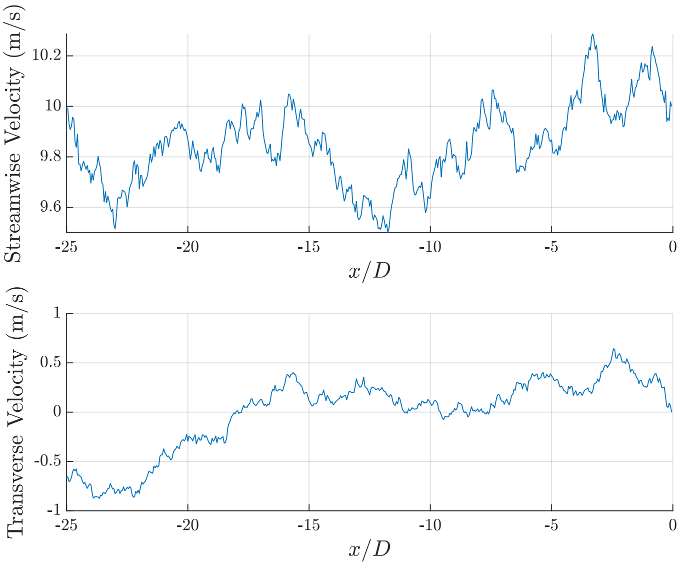

Figure 7: Streamwise and transverse inflow velocities

The negative x/D in the x-axis illustrates that these inflow velocities lie in front of the turbine at a distance x/D and will move at a speed of U_∞ towards the turbine when the simulation is started. From observing Figure 7, it is noticeable that the streamwise inflow velocity hover around the freestream wind speed of $U_\infty = 10$ m/s while the transverse inflow velocity fluctuates around zero. The random disturbances generated using the Ornstein-Uhlenbeck process exhibits realistic traits of real wind speed data and thus are a sensible synthetic inflow conditions to be fed into the discretised model.

To clearly see the transport of disturbances downstream, the disturbances were only set to have values within a certain distance in front of the turbine. This allowed a clearer view of the transport of disturbances. Figure 8 shows the inflow conditions fed into the model to validate the transport of disturbances. Figure 9 displays the wake produced by the discretised model given inflow conditions from Figure 8 at different times.

With freestream velocity set to $U_\infty = 10$ m/s, every 40 seconds, it is expected that the disturbance would have moved downstream a distance of 400 m. Therefore, with turbine diameter $D = 100$ m, it is anticipated that the disturbance would move downstream by $x/D = 4$. From Figure 9, at $t = 250$ s, a kink in the wake is observed at $x/D = 2$ indicating that the disturbances have propagated up to $x/D = 2$. At $t = 290$ s, the kink is now located at $x/D = 6$ implying that the disturbances have moved a distance of $4D$ downstream, agreeing with the statement mentioned beforehand. A similar argument can be made for $t = 330$ seconds, where the kink is now located at $x/D = 10$. Therefore, it can be said that the model transports the disturbance downstream at the correct advection speed.

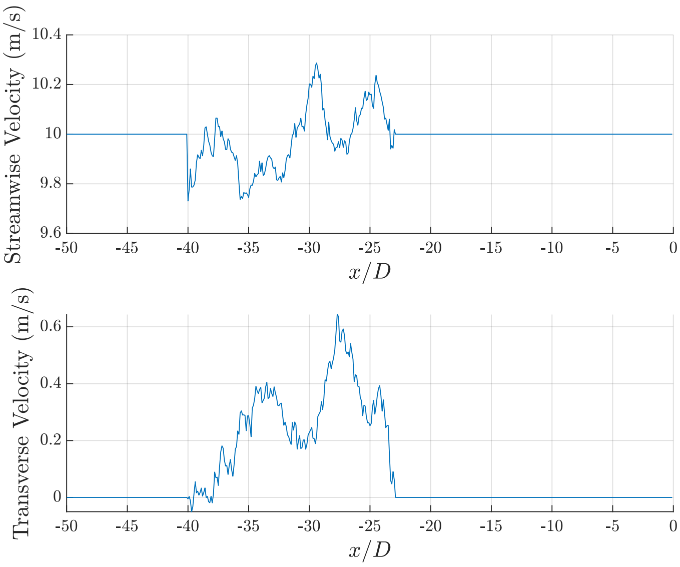

Figure 8: Streamwise and transverse inflow velocities used for validation of model

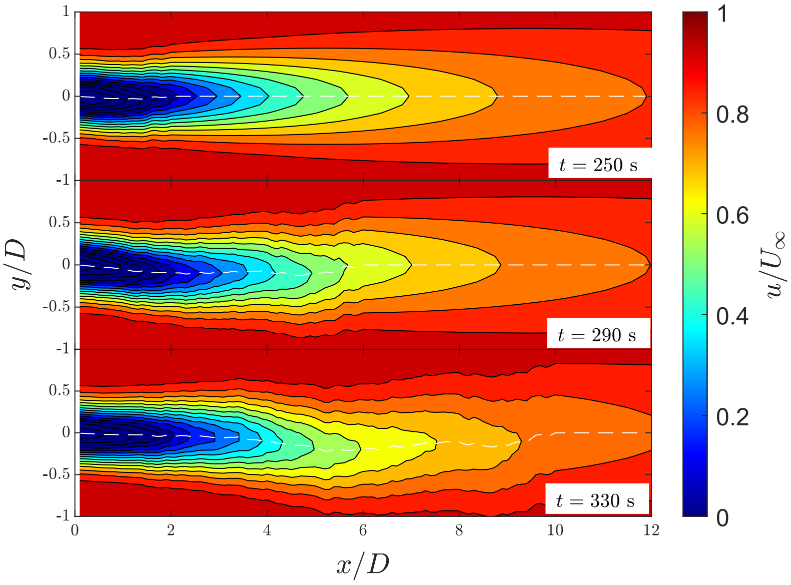

Figure 9: Contour plots of normalised streamwise velocity due to inflow conditions seen in Figure 8 at different times with yaw angle set to zero.

By validating the discretised model's response to change in yaw and unsteady inflow conditions, it can be deduced that the model responds correctly to turbine yaw and disturbances.

3.3.2 MHE algorithm

As the discrete model validated above shows close correspondence to experimental results given by Bastankhah et al. as seen in Figure 5 and Figure 6, the flow field generated by the model will be used to validate the accuracy of the MHE algorithm. The steps in investigating the accuracy of the reconstructed wake will be as follows:

1. Generate a flow field with a given turbine yaw angle and a set of randomly generated inflow velocities using the discrete model.

2. Obtain the streamwise velocity produced by the discrete model at specific points in the flow field over the whole simulation time to emulate having sensors that pick up the streamwise velocities.
3. Use these velocities as input to the MHE algorithm to reconstruct the turbine wake.
4. Compare the flow field produced by the discrete model and the reconstructed flow field to validate the accuracy of the MHE algorithm.

To fully test the accuracy of the MHE algorithm, the flow field used as input to the MHE algorithm is generated by the discrete model with turbine yaw angle set to -10° and with unsteady disturbances as seen in Figure 7. This examines the prediction performance of the MHE algorithm in the most general case where the turbine is yawed and subjected to unsteady inflow conditions.

Regarding some parameters required for the MHE algorithm, the sensors are selected to be at locations $(2D, \pm 0.1D)$. The y-coordinates of the sensors, $\pm 0.1D$, ensure that both the sensors lie within the wake. Additionally, the horizon length used was $M = 50$. The tuning parameters that were used are $\alpha_1 = \beta_1 = 50$ and $\alpha_2 = \beta_2 = 0.01$. Figure 10 illustrates the wakes generated by the discrete model and MHE algorithm.

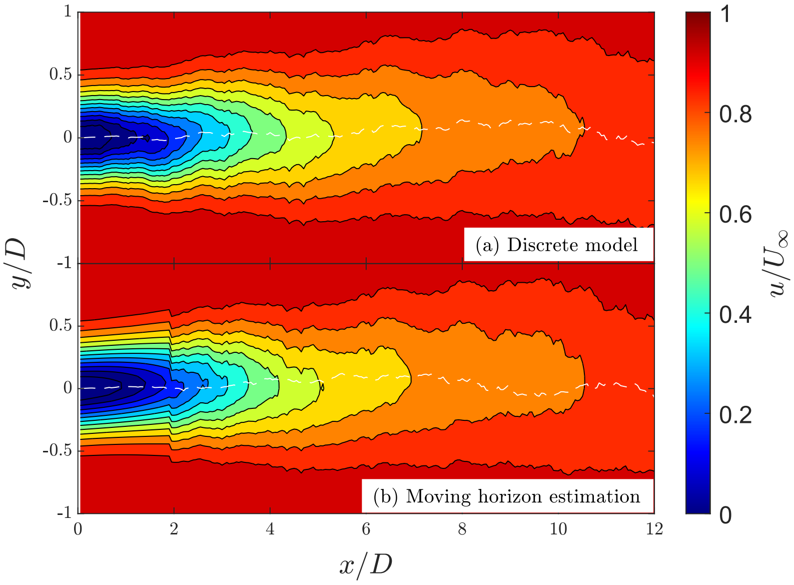

Figure 10: Comparison of contour plots of normalised streamwise velocity (a) generated using the discrete model (b) estimated using MHE

To further quantify the error, Figure 11 shows a contour plot of the difference between the normalised streamwise velocity from the actual and estimated wake in Figure 10. As seen from both Figure 10 and Figure 11, the estimation of streamwise velocity is highly accurate behind the sensors. However, the estimation prior to the sensor is observed to be inaccurate. This is mainly due to the size of the horizon used being too small, thus having insufficient data to accurately estimate the streamwise velocity upstream of the sensor. Utilising a larger window size will improve estimation performance prior to the sensor but comes with an increased computational cost. Using $M = 50$ strikes a balance between the level of accuracy acquired and the computational cost.

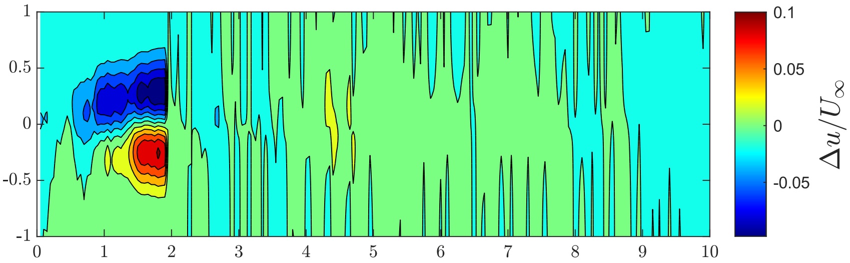

Figure 11: Difference between the normalised streamwise velocity generated by the discrete model and estimated by MHE algorithm

Additionally, as the MHE algorithm provide the state estimates, $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{y}}$, it is also important the investigate the accuracy of these estimations. Figure 12 show the comparison plots between the actual and estimated states.

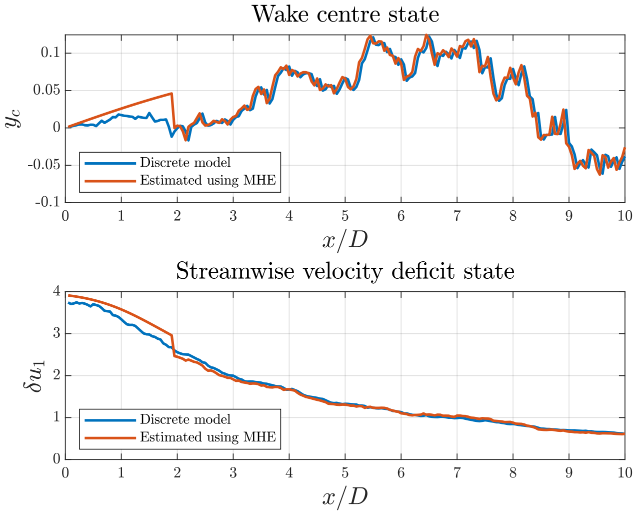

Figure 12: Plots of actual and estimated states of the wake in Figure 10

As seen from Figure 12, the estimated states are highly accurate after the sensor and are incorrect prior to the sensor as expected due to reasons discussed above. Thus, a limitation of the MHE algorithm developed is the prediction of states and streamwise velocity in the near-wake region (regions prior to the sensor). However, for the purposes of wake steering, only accurate estimates in the far-wake region are required. As the MHE algorithm is able to provide highly accurate estimates after the sensor, it is adequate to be used for wake steering given that sensors are placed prior the subsequent turbines.

4 Extension to multiple wind turbines

The models developed so far are only capable of providing the flow field corresponding to the wake of a single turbine. As a result, these models are inadequate for modelling wind turbine wakes within a wind farm where wake interactions occur between multiple wind turbines. Due to the complex turbulent phenomena involved in wake interactions, it is not adequate to simply sum the wakes from multiple wind turbines to obtain the wind farm wake [21]. This Section explores the existing wake superposition models and the process of extending the single turbine discrete model to multiple turbines. In addition to building a discrete model for multiple turbines, this

Section also discusses the extension of the MHE algorithm to estimate and reconstruct the wakes of multiple turbines.

4.1 Wake superposition models

To extend the models developed to account for the wake interaction of multiple turbines, it is important to explore different wake superposition models that can be implemented. A few approaches exist in literature to determine the streamwise velocity downstream of multiple turbine wakes. These approaches include: Geometric Sum (GS), Linear Superposition of Velocity Deficits (LSVD), Sum of Energy Deficits (SED) and Sum of Squares (SS) [21]. Table 2 summarises the wake superposition models that currently exist in literature [21].

Wake Superposition Model	**Equation**
Geometric Sum (GS)	$\frac{u_i}{U_\infty} = \prod_{j=1}^{N} \frac{u_{ij}}{u_j}$
Linear Superposition of Velocity Deficits (LSVD)	$\left(1 - \frac{u_i}{U_\infty}\right) = \sum_{j=1}^{N} \left(1 - \frac{u_{ij}}{U_\infty}\right)$
Sum of Energy Deficits (SED)	$\left(U_\infty^2 - u_i^2\right) = \sum_{j=1}^{N} \left(u_j^2 - u_{ij}^2\right)$
Sum of Squares (SS)	$\left(1 - \frac{u_i}{U_\infty}\right)^2 = \sum_{j=1}^{N} \left(1 - \frac{u_{ij}}{U_\infty}\right)^2$

Table 2: Summary of wake superposition models

In the equations mentioned in Table 2, u_i denotes the streamwise velocity at turbine i, u_{ij} denotes the streamwise velocity at turbine i as a result of the wake of turbine j and the summations/products are taken over the N turbines upstream of turbine i [21].

Garrad Hassan conducted a comparison of these four models using wind tunnel data. It was found that the SS approach was the best performing method for most scenarios [22]. Additionally, the GS and LSVD approaches are known to overestimate the velocity deficit [22]. While it may be an obvious choice to opt for SS approach to extend the model to handle multiple turbines as it gives the most accurate results, its non-linearity may pose problems. Since the ultimate goal of this project is to maximise power through wake steering which will be done by maximising an objective function, the use of the SS approach would increase the non-linearity of an already non-linear objective function, leading to increased computational times and reduced solvability [21]. For this situation, the SS approach introduces more drawbacks than benefits.

Due to the reasons mentioned above, the LSVD wake superposition approach was used to introduce wake interactions to the single turbine discrete model. Despite providing less accurate results, the LSVD approach is linear and thus will eliminate the aforementioned complications. Equation 4.1 gives the expression for the streamwise velocity at turbine i using the LSVD approach.

$$u_i = U_\infty \left[1 - \sum_{j=1}^{N} \left(1 - \frac{u_{ij}}{U_\infty}\right)\right] \tag{4.1}$$

4.2 Implementation of wake superposition into the discrete model

To implement Equation 4.1 into the single turbine discrete model, it is important to note its dependence of the conditions upstream. Thus, there is a specific order in which the wakes of turbines have to be computed - starting from the most upstream turbine and moving downstream.

The wake of the most upstream turbine is computed as mentioned in Section 3.1.2. However, some alterations have to be made to obtain the correct wakes for the subsequent turbines. Firstly, as the subsequent wind turbines are not located at the origin, it is necessary to perform a coordinate translation to situate the subsequent turbines at the correct positions downstream. Thus, the states at time t_k of the subsequent turbine will be given as:

$$\boldsymbol{z}_k = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ u_1(x_\tau, t_k) \\ u_1(x_{\tau+1}, t_k) \\ \vdots \\ u_1(x_N, t_k) \end{pmatrix} \in \mathbb{R}^N, \quad \boldsymbol{y}_k = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ y_c(x_\tau, t_k) \\ y_c(x_{\tau+1}, t_k) \\ \vdots \\ y_c(x_N, t_k) \end{pmatrix} \in \mathbb{R}^N \tag{4.2}$$

where x_τ is the discretised downstream location of the subsequent turbine. As seen from Equation 4.2, the streamwise velocity deficit and wake centre of the discretised points prior to the subsequent turbine are set to zero. This ensures that the wake of the subsequent turbine only manifests downstream of its location. It will be convenient to define local state vectors for subsequent turbines. This reduces the sizes of the vectors and matrices thus increasing computational efficiency. The mathematical definition of the local states for subsequent turbines are presented in Equation 4.3.

$$\boldsymbol{z}_{k,loc} = \begin{pmatrix} u_1(x_\tau, t_k) \\ u_1(x_{\tau+1}, t_k) \\ \vdots \\ u_1(x_N, t_k) \end{pmatrix} \in \mathbb{R}^{N-N_\tau}, \quad \boldsymbol{y}_{k,loc} = \begin{pmatrix} y_c(x_\tau, t_k) \\ y_c(x_{\tau+1}, t_k) \\ \vdots \\ y_c(x_N, t_k) \end{pmatrix} \in \mathbb{R}^{N-N_\tau} \tag{4.3}$$

Where N_τ is the number of uniformly discretised points along the x axis from 0 to x_t. In addition to alterations made to the state vectors, changes are also made to the A, E and B matrices as shown in Equation 4.4.

$$A_1' = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ d_1 & 0 & \cdots & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{N_\mathcal{T}-1} & 0 \end{pmatrix} \in \mathbb{R}^{N_\mathcal{T} \times N_\mathcal{T}}, \quad E_1' = \begin{pmatrix} d_0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{N_\mathcal{T}}, \quad B_1' = \begin{pmatrix} \frac{d_0}{U_\infty} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{N_\mathcal{T}} \tag{4.4}$$

Where $N_\mathcal{T} = N - N_\tau$. As seen in Equation 4.4, the structure of the matrices remain the same and only the alterations in sizes were made to accommodate the new dimensions of the local state vectors. The same alterations were also made to matrices, A_c, E_c and B_c. Similarly to the single turbine model, to advance the local states by one timestep, Equation 3.23 is used however with local state vectors and the altered matrices as stated above. Likewise, the $s_{i,k}$ terms in this case are the known forcing function at time t_k for the subsequent turbine and will be denoted as $s_{i,k}'$. However, for subsequent wind turbines adjustments must be made to the input of disturbances at time t_k, $w_{i,k}$, to reflect the altered conditions from the wakes of upstream wind turbines.

In this multiple turbine model, the effects of wake interactions from upstream turbines are modelled as disturbances that impact subsequent turbines. This assumption is considered valid as wake interactions from upstream turbines effectively alter the incoming freestream airspeed, analogous to how disturbances influence the incoming airflow. The adoption of this assumption allows the wake superposition model to be seamlessly integrated into the existing single turbine wake model. The streamwise velocity disturbance for subsequent turbines at time t_k can be expressed as the difference between the freestream velocity, U_∞, and the streamwise velocity at the subsequent turbine at time t_k, $u_{i,k}$. Therefore, the streamwise velocity disturbance for subsequent turbines at time t_k, $w'_{1,k}$, can be expressed as:

$$w'_{1,k} = U_\infty - u_{i,k} \tag{4.5}$$

Thus, substituting the expression for u_i as given in Equation 4.1 to get:

$$\begin{aligned} w'_{1,k} &= U_\infty - U_\infty \left[1 - \sum_{j=1}^{N} \left(1 - \frac{u_{ij,k}}{U_\infty}\right)\right] \\ &= U_\infty \left[\sum_{j=1}^{N} \left(1 - \frac{u_{ij,k}}{U_\infty}\right)\right] \end{aligned} \tag{4.6}$$

Where $u_{ij,k}$ is the streamwise velocity at turbine i due to the wake of turbine j at time t_k and the summation is taken over the N turbines upstream of turbine i. As the streamwise velocity of the wake due to upstream turbines varies spatially, the average value of $u_{ij,k}$ over the downstream turbine's rotor swept area was used. This ensures a more accurate representation of the upstream turbine's wake effects across the entire rotor swept area, enhancing the accuracy of modelling wake interaction effects as a disturbance input. Since the wakes of the upstream turbines already incorporate the initial disturbances that lie in front of the most upstream turbine (Figure 7), thus there is no need to introduce additional streamwise velocity disturbances beyond the wake effects from upstream turbines.

On the other hand transverse velocity disturbances, $w'_{2,k}$, are modelled differently. As disturbances are transported downstream at the freestream speed, U_∞, the transverse velocity disturbances experienced by subsequent wind turbines are assumed to be equivalent to those that affect the most upstream turbine, though with a time delay. This delay accounts for the time required for these disturbances to travel from one turbine to the next downstream. Thus, with assumptions of disturbances, modified state vectors and matrices, the discrete model for multiple wind turbines can be built as seen in Equation 4.7 and Equation 4.8.

$$\boldsymbol{z}_{k+1,loc} = A'_1 \boldsymbol{z}_{k,loc} + E'_1 w'_{1,k} + B'_1 s'_{1,k} \tag{4.7}$$

$$\boldsymbol{y}_{k+1,loc} = A'_c \boldsymbol{y}_{k,loc} + E'_c w'_{2,k} + B'_c s'_{2,k} \tag{4.8}$$

To obtain the streamwise flow field of the entire wind farm, a precise compilation process is necessary. The non-linear relationship between the wake states and streamwise velocity as seen in Equation 3.19, means that simply superposing the wake states of each turbine will not yield the correct results. Instead the streamwise velocity must be individually calculated for each turbine using Equation 3.19. After the After the velocities for each turbine are obtained, they can be carefully superposed to result in the flow field of the wind farm.

4.3 Discrete model for multiple wind turbines

The single turbine discrete model is now extended to model wakes of multiple turbines placed arbitrarily using the assumptions and concepts introduced above. Figure 13 provides three examples of the streamwise velocity contour plots for multiple turbines placed in different positions.

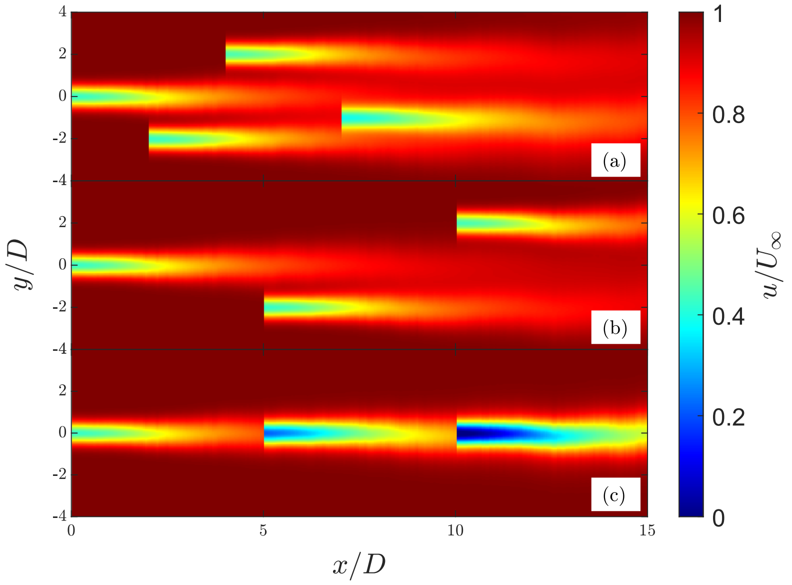

Figure 13: Contour plot of normalised streamwise velocity from multiple turbine discrete model for three different turbine configurations

From case (a) and (b), the wakes from the three turbines look similar with relatively small velocity deficits. This is because, in this case, there are no wake interactions since the downstream turbines are not placed in the vicinity of the upstream turbine's wakes. On the other hand, looking at the case (c), the wake of the most downstream turbine shows significantly larger velocity deficit due to the large amounts of wake interaction from the two upstream turbines. This qualitatively illustrates that the multiple turbine discrete model is able to simulate wake interaction from multiple turbines. Additionally, a qualitative comparison was made with a multiple turbine wake model introduced by Wei et al. as seen in Figure 14.

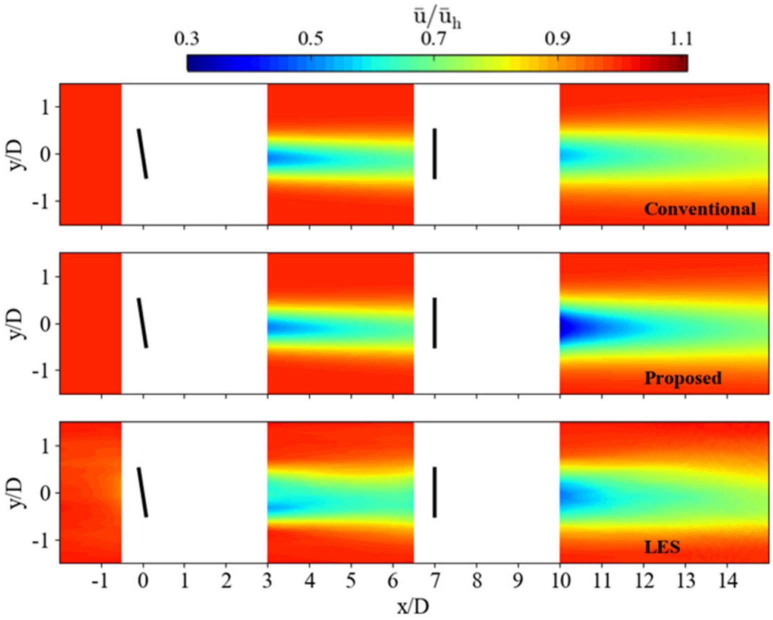

Figure 14: Contours of normalised streamwise velocity for two turbines with upstream turbine yawed by 10°: A conventional analytical model (top), model proposed by Wei et al (middle) and LES simulations (bottom) [23]

To effectively compare the developed model with Figure 14, the wake of two turbines with $7D$ spacing and upstream turbine yawed by $10°$ was generated using the same wake parameters as seen in Figure 15.

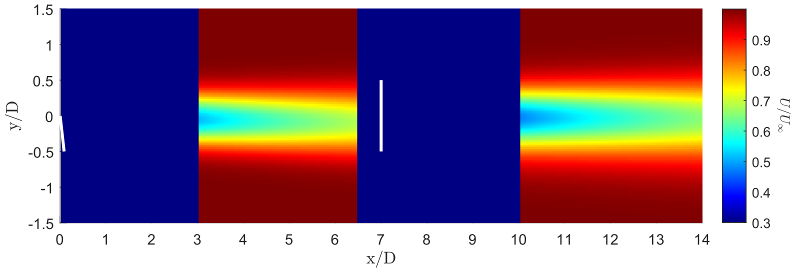

Figure 15: Contour of normalised streamwise velocity for two turbines

The wakes from Figure 14 and Figure 15 show resemblance in terms of the streamwise velocity deficit. Thus, the multiple turbine discrete model qualitatively correct. While the model has been validated qualitatively it is crucial that is also quantitatively validated. Due to time constraints, a quantitative validation was not conducted and is part of the future work that can be done.

4.4 MHE algorithm for multiple wind turbines

Extending the MHE algorithm to accommodate multiple turbines within a wind farm is a simpler process, provided that sensors are placed behind each turbine. This setup ensures that the MHE algorithm takes into account the wake interactions that occur in the form of the streamwise velocity measured from the sensors. This allows the MHE algorithm to adjust its estimates of the flow field behind every turbine.

Specifically, extending the MHE algorithm to multiple turbines involves running the algorithm individually for each turbine, with the streamwise velocity measurements collected behind each turbine. When the estimated states for each turbine are determined, the estimated streamwise velocity flow field for each turbine wake is obtained using Equation 3.19. The velocities obtained for each turbine are then superposed using the same compilation process as the multiple turbine discrete model. This results in the reconstructed flow field of the wind farm.

The wakes of multiple turbines in Figure 13 were reconstructed using the MHE algorithm where sensors are placed at locations $(0.5D, \pm 0.2D)$ relative to each turbine. The horizon length used was $M = 50$ and the tuning parameters were $\alpha_1 = \beta_1 = 50$ and $\alpha_2 = \beta_2 = 0.01$. Figure 16 presents the difference between the normalised streamwise velocity generated by the multiple turbine discrete model and that estimated by the MHE algorithm for the turbine configurations in Figure 13.

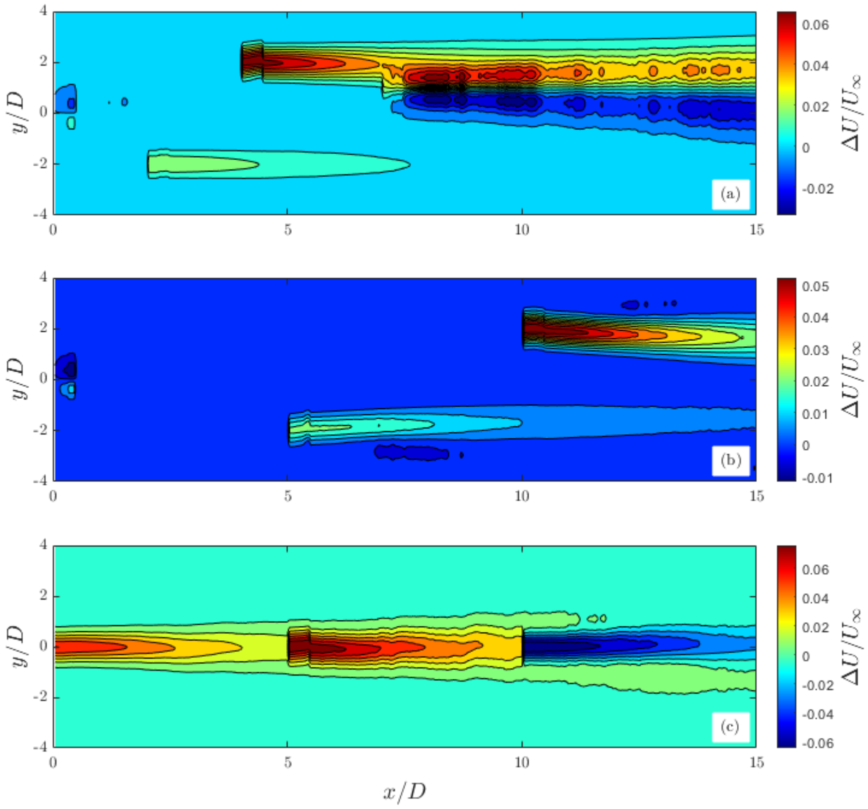

Figure 16: Difference between the normalised streamwise velocity generated by the multiple turbine discrete model and that estimated by the MHE algorithm

As seen from the plot in Figure 16, the estimated streamwise velocity are in tolerable limits and are more accurate in the far-wake region. Thus, given that the downstream turbines are positioned in the far-wake region and with sensors situated near the upstream turbine, the MHE algorithm is able to accurately estimate the streamwise velocity at the downstream turbines. Therefore, the MHE algorithm developed for multiple turbines is adequate to be used to build a practical wake steering controller.

5 Wind farm optimisation via wake steering

After establishing a model to simulate and estimate the flow field of a wind farm, a wake steering controller can be built with the sole purpose of maximising the power output of a wind farm. As the main objective is to maximise power, this Section details a study on power evaluation of yawed wind turbines. Additionally, this Section also introduces the optimisation problem in more detail and the selection of suitable parameters to be used to ensure sensible values for optimal yaw angles. Lastly, the design of the wake steering controller is also presented.

5.1 Power evaluation of yawed wind turbines

To evaluate the power, P for an individual wind turbine, Equation 5.1 is utilised [24].

$$P = \frac{1}{2}\rho A C_p U^3 \tag{5.1}$$

The power of the wind farm, P_T, is given as the sum of the powers of each individual turbine within the wind farm as seen in Equation 5.2.

$$P_T = \sum_{i=1}^{N} \frac{1}{2} \rho A_i C_{p_i} U_i^3 \tag{5.2}$$

where ρ is the density of the air, A is the rotor swept area, C_p is the power coefficient, U is the inflow wind speed of the turbine and N is the number of turbines within the wind farm. For convenience, the density of air will be fixed at 1.225 kg/m^3, which is the commonly used value for density of air at sea level [25]. The power coefficient, C_p, represents the proportion of wind power that a turbine can convert into mechanical energy [24]. The power coefficient is expected to decrease when the wind turbine is in a yawed state, resulting in a reduction in power output; henceforth the need to apply a correction to the power coefficient [26]. An illustration of this relationship can be shown in Equation 5.3 [24].

$$C_{p_i} = 4a_i \left(\cos\gamma_i - a_i\right) \left(\cos\gamma_i + \tan\frac{\chi_i}{2} \sin\gamma_i - a_i \sec^2\frac{\chi_i}{2}\right) \tag{5.3}$$

with the initial wake skew angle, χ_i, defined as:

$$\chi_i = \left(0.6a_i + 1\right)\gamma_i \tag{5.4}$$

and a_i is the axial induction factor of the turbine. Typically, a value of 1/3 is used for axial induction factor to achieve maximum power coefficient for a given yaw angle [24]. Figure 17 illustrates the relationship between turbine yaw angle and the power coefficient.

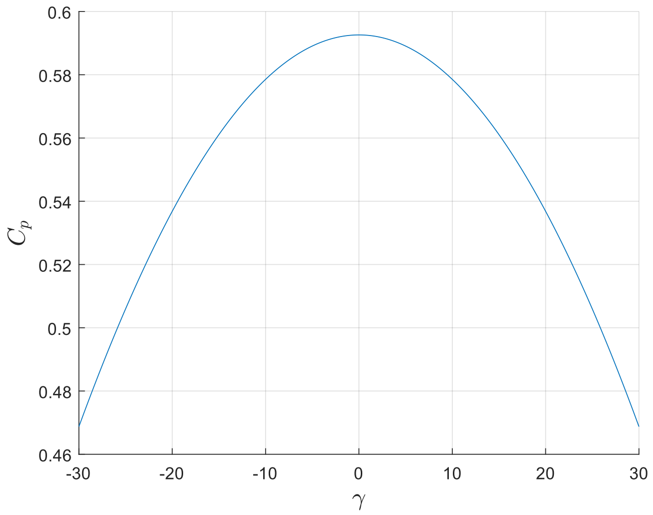

Figure 17: Plot of power coefficient evaluated with different turbine yaw angles and axial induction factor of 1/3

Figure 17 clearly illustrates that the power coefficient decreases as the yaw angles increases, with maximum power coefficient being achieved when the turbine is in an unyawed state. Since the objective function is to maximise power output, using the corrected C_p as seen from Equation 5.3, introduces a penalty on turbine yaw as it will consequently decrease power output. However, another variable aspect of the power function in Equation 5.1, is the inflow wind speed, defined as the wind speed immediately in front of the turbine. Given from Equation 5.1, the power production is directly proportional to the cube of the inflow wind speed. Thus it is also paramount to obtain the highest inflow wind speeds possible for each individual turbine to maximise power generation.

5.2 Optimisation problem formulation and parameter selection

When evaluating the entire wind farm, where the power output from each turbine is crucial for maximising overall farm efficiency, the yaw angles and incoming wind speeds at each turbine emerge as critical factors to consider. For instance, consider two turbines aligned in a row as depicted in Figure 18. Yawing the upstream turbine redirects its wake away from the downstream turbine, increasing the incoming wind speed reaching the downstream turbine boosting its power production. However, as a consequence of yawing the upstream turbine, its power output is reduced due to a decrease in power coefficient. Therefore, there exists an optimal yaw angle for the upstream turbine that maximises the downstream turbine power production while minimising the reduction in power production for the upstream turbine. This creates a highly complex optimisation problem due to the inherent nonlinear characteristics involved in wind farm wake systems.

Due to the complexity of the optimisation problem, numerous parameters can influence the optimal yaw angle. Therefore, it is crucial to investigate each parameter individually to ensure that a sensible optimal yaw angle exists before building a wake steering controller. For this purpose, a simple scenario with two turbines placed in a line was used as seen in Figure 18. In this setup, the optimal solution for maximising overall farm power involves yawing the upstream turbine while keeping the downstream turbine unyawed. The optimal yaw angle is expected to steer the wake away from the downstream turbine such that its power production is maximised from the increased incoming wind speed, while minimising the power loss from the effects of yawing the upstream turbine. This simplified two-turbine configuration allows for a clearer understanding of the dynamics involved in yaw optimisation and helps aid the selection of suitable parameters such that a sensible optimal yaw angle exists.

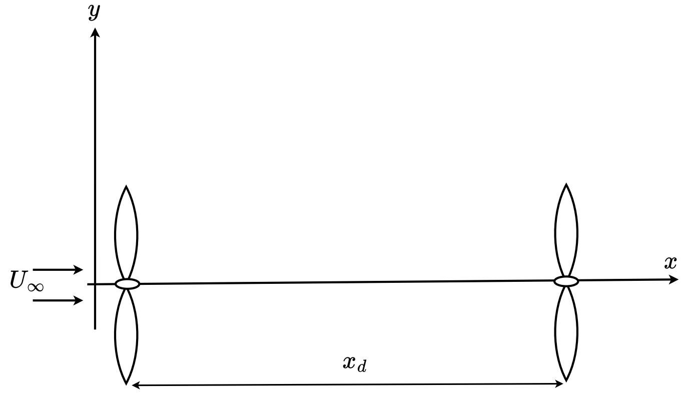

Figure 18: Turbine configuration used to select parameters

5.2.1 Thrust coefficient

The thrust coefficient, C_T, is the non-dimensional quantity that describes the force exerted by the turbine onto the incoming flow [27]. The wind turbine thrust coefficient can significantly influence the wind turbine wake characteristic. A high thrust coefficient promotes the formulation of turbulent flow near the turbine, this enhanced mixing leads to a quicker recovery of the wind's initial velocity [27]. As the thrust coefficient value affects the wake shape, it will consequently affect the optimal yaw angle for maximum overall farm power.

Conveniently, [24] provides an expression coefficient of thrust as a function of the axial induction factor and yaw angle as seen in Equation 5.5

$$C_{T_i} = 4a_i \left(\cos\gamma_i + \tan\frac{\chi_i}{2}\sin\gamma_i - a_i\sec^2\frac{\chi_i}{2}\right) \tag{5.5}$$

Therefore, using $a = 1/3$, the thrust coefficient can be evaluated explicitly for yawed turbines using Equation 5.5.

5.2.2 Wake proportionality constant, σ_0

From observation of Equation 3.19, the wake proportionality constant σ_0 determines the spread of the streamwise velocity deficit across the direction perpendicular to the flow direction immediately behind turbine. A larger wake proportionality constant indicates a wider spread of the wake across the perpendicular axis, implying that the impact of the velocity deficit is distributed over a larger area. While a smaller wake proportionality constant suggests a narrower wake, concentrating the velocity deficit over a smaller area. Ultimately, this means that the wake proportionality constant, is a key parameter that affects the wake characteristic and thus influences the optimal yaw angle.

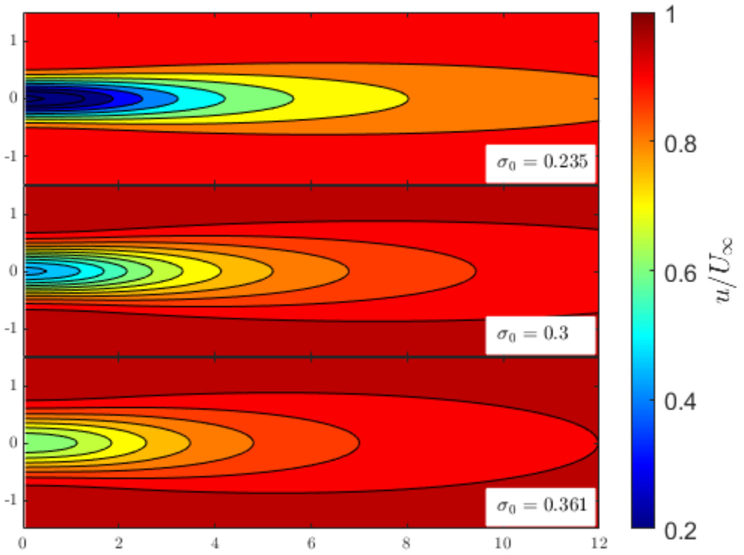

Figure 19: Wakes with three different values of wake proportionality constant with $k_w = 0.0834$

From Figure 19, it can be deduced that the wake proportionality constant highly affects the characteristic of the wake and the streamwise velocity deficit behind the turbine. Therefore, the optimal yaw angle is highly dependent on the value of the wake proportionality constant. In [26], the wake proportionality constant used was 0.361, for wake steering using another form of the Gaussian-shape wake deficit model and has shown sensible results. Therefore, to build a controller for wake steering in this project, it is assumed that the wake proportionality constant takes the same value.

5.2.3 Wake expansion coefficient, k_w

The wake expansion coefficient, k_w, is also another parameter that can affect the wake characteristic and defines the expansion rate of the wake downstream of the turbine. Figure 20 illustrate the turbine wakes for different values of wake expansion coefficient with the wake proportionality constant set to 0.361.

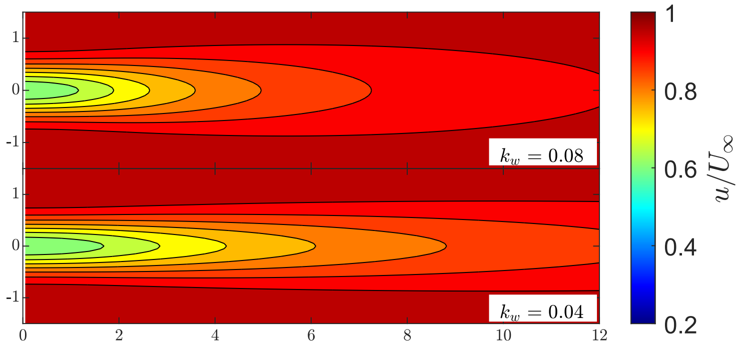

Figure 20: Wakes with two different values of wake expansion coefficient with $\sigma_0 = 0.361$

Thus, from observation of Figure 20, the wake expansion coefficient affects both the length and width of the wind turbine wake and thus will influence the optimal yaw angle. The wake coefficient is semi-empirically based on the ambient turbulence intensity, where a value of 0.075 typically is used for onshore wind farms, whereas offshore wind farms will use values between $0.04 - 0.05$ [28]. Additionally, in [26], a value of 0.08 was used for the wake expansion coefficient and has produced sensible optimal yaw angles. Therefore, for the initial design stage of the controller, a value of 0.08 will be used.

5.2.4 Turbine spacing

The selection of spacing between turbines, x_d, is another important parameter to investigate. This is because the spacing between turbines effectively determines the magnitude of the yaw angle. As the model is only validated up to yaw angles of 30°, the yaw angles limits are placed between $-30°$ and $30°$. A tight spacing between turbines can cause the optimal yaw angle to exceed the modelling limits, leading to an inaccurate simulation. This larger optimal yaw angle stems from the need to divert the wake away from the downstream turbine in a short distance. Conversely, if the spacing between turbines is sufficiently large, the optimal yaw angle may be zero. In this case, the power gained from the increased incoming wind speed at the downstream turbine does not compensate for the power loss from yawing the upstream turbine. This leads to a uninteresting scenario where the optimal solution is to not perform any further control.

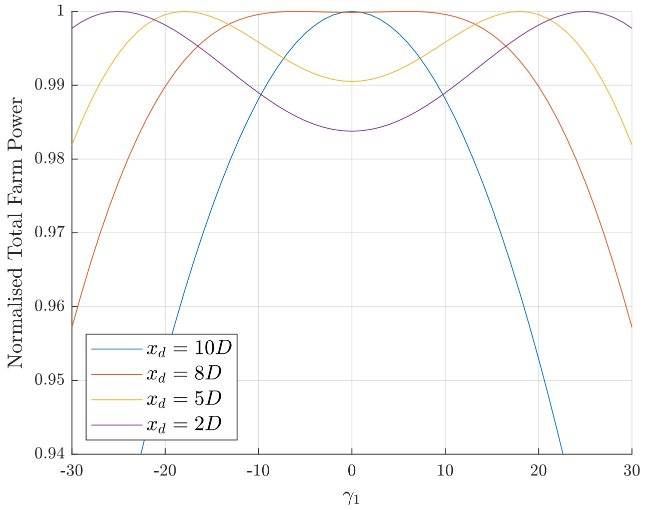

Figure 21: Normalised total farm power against yaw angles of the upstream turbine, γ_1, for different values of x_d, with $\sigma_0 = 0.3607$ and $k_w = 0.08$

To determine an appropriate distance between turbines, the total farm power in steady freestream conditions was plotted against the yaw angles of the upstream turbine while keeping the downstream turbine in an unyawed state for different turbine distances. Figure 21 illustrates the relationship between total farm power and yaw angle of the upstream turbine for different turbine distances.

As seen from the plot, it can be deduced that $x_d > 8D$ is not a suitable turbine distance to engage in wake steering. This is due to the turbines being positioned at an adequate distance from each other, allowing the wind to regain its initial speed. Therefore, yawing the turbine to steer the wake away from the downstream turbine has minimal positive effect on its performance as the wind has recovered to near its initial speed, while having more detrimental effect on the upstream turbine's performance through yawing. This results in the optimal yaw angle being zero as seen on Figure 21. On the other hand, for $x_d < 5D$, yawing the upstream turbine does indeed boost the overall farm power as seen from non-zero optimal yaw angles in Figure 21. This is because the power gained due to increased incoming wind speed at the downstream turbine is larger than the power loss from yawing the upstream turbine. Therefore $x_d < 5D$ formulate the desired optimisation problems to be solved to increase overall farm power through wake steering. However, as it is observed that the optimal yaw angle increases as the turbine spacing decreases; therefore, the choice of turbine spacing must be not be too small such that the optimal yaw angle exceed modelling limits.

As a reference, the Horns Rev offshore wind farm in Denmark positions turbines with a spacing of $7D$ apart [29]. Figure 28 (Appendix) illustrates the Horns Rev wind farm turbine layout. However, a wide spacing of 7D can lead to ambiguous optimal solutions, similar to $x_d = 8D$ in Figure 21. Therefore, for the purposes of this project, a turbine spacing of $5D$ will be used. This spacing ensures a clear and definite optimal solution that are within the limits of the model.

5.2.5 Final parameters

The final parameters used to construct a wake steering controller are as follows:

Parameters	Symbol	Value
Freestream wind speed	U_∞	10 ms^{-1}
Turbine diameter	D	100 m
Axial induction factor	a	1/3
Wake proportionality constant	σ_0	0.361
Wake expansion coefficient	k_w	0.08
Turbine spacing	x_d	$5D$

Table 3: Final parameters used to construct a wake steering controller

5.3 MPC controller design for wake steering

As stated in Section 2.3, an MPC approach will be used to design a wake steering controller to maximise total power output of a wind farm. The MPC controller is initially designed and tested for a two-turbine configuration as seen in Figure 18 with parameters in Table 3. This conveniently allows the results to be validated with the known optimal yaw angles as seen in Figure 21.

5.3.1 Cost function

Prior to creating the MPC controller, it is important to define a set point (objective function) in which the controller will try adjust its control inputs to achieve. The objective function for the MPC controller aims to maximise the total wind farm energy, E_T, over the horizon length, l, at time t. Equation 5.6 details the objective function.

$$\begin{aligned} \underset{\gamma_{i,j}}{\text{maximize}} \quad & E_T = \sum_{i=t}^{t+l} \sum_{j=1}^{N} P_{i,j}(U_{i,j}, a_j, \gamma_{i,j}) \\ \text{subject to} \quad & -30^\circ < \gamma_{i,j} < 30^\circ \end{aligned} \tag{5.6}$$

Where $P_{i,j}$ is the power of turbine j at time i which is a function of the inflow wind speed of the turbine U, the turbine axial induction factor a and the turbine yaw angle γ. While N is the number of turbines in the wind farm. Since the model is only validated up to $\gamma = 30^\circ$, constraints are imposed on the yaw angle. Without these constraints, the controller may pick optimal yaw angles outside the modelling limits, leading to an inaccurate simulation. This objective function ensures that the MPC controller produces control inputs directed towards maximising the power generated by the wind farm over the horizon.

5.3.2 Construction of the MPC controller

To construct an effective MPC controller for wake steering, the underlying nature of wind farm wakes must be considered. There is a delay between the time the control inputs are applied by the upstream turbine and when the downstream turbine experience the effects. Therefore, the horizon length of the MPC controller must be long enough to account for this delay, ensuring that the effects of the current control inputs from the upstream turbine have been propagated sufficiently. As the turbine spacing is $5D$, the horizon length, l, must be:

$$l = \frac{5D}{U_\infty} \tag{5.7}$$

This ensures that the MPC controller projects the states sufficiently into the future to observe the effects of the upstream turbine's control inputs on the downstream turbine. As the value of l is significant, projecting the states from the upstream turbine to the downstream turbine becomes a computational burden. This is due to two main reasons; firstly, for sufficient projection of states, the controller must iteratively calculate the states l times requiring very high computational resources. Additionally, for every projection by one timestep, an additional control input is required, thus l optimal control inputs would be required to propagate the states sufficiently. This is a huge computational burden as the MPC controller would have to optimise l control inputs per turbine simultaneously.

To address the issue regarding the need to calculate the states l times, a multi-step model is utilised instead of the traditional single-step model. The multi-step model used is adapted from Equation 3.29 for multiple turbines, which allows the prediction of states at any arbitrary time in the future given that the future disturbances and control inputs are known. Therefore, with the horizon length split into p segments and the multi-step model propagating the states by l/p timesteps per iteration, only p iterations required to propagate the states to the downstream turbine, thereby reducing the computational cost. Additionally, to solve the issue regarding the need for l optimal inputs per turbine, a sparse control approach is adopted. With the horizon split into p segments, the optimal control inputs are found for every l/p timesteps instead of every timestep. This certainly reduces computational burden but comes with the cost of a lower control effectiveness. The value of p can be chosen arbitrarily to balance between the control accuracy required and the available computational power. Figure 22 illustrates the propagated wake spatially at time $t + l$, detailing the different parts of the wake where the optimal control inputs are changed.

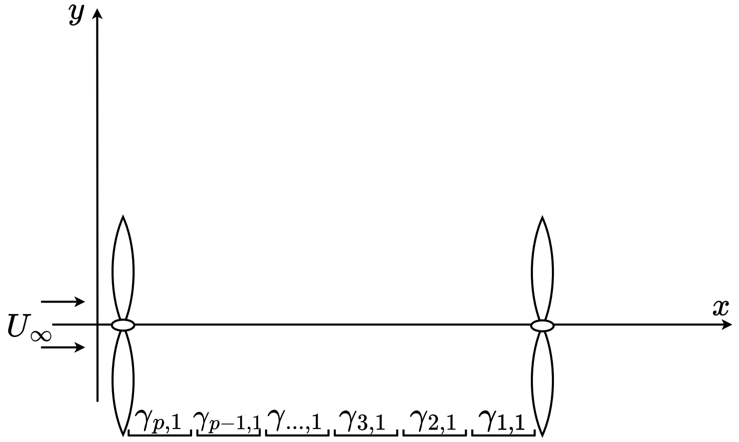

Figure 22: Propagated wake from a spatial point of view with the corresponding optimal control inputs

From Figure 22, $\gamma_{i,j}$ is the optimal yaw angle for turbine j at segment i. With this setup, the MPC controller is able to evaluate the effect of the upstream's turbine wake downstream and choose the best sparse control inputs at which the total wind farm energy over the horizon is maximised at a significantly lower computational cost.

Due to the lack of knowledge on future disturbances, the accuracy of the MPC controller's predicted future states is limited. Consequently, the controller's optimal control inputs will never be the true optimal input. This limitation is mitigated through the implementation of an Ornstein-Uhlenbeck model as stated in Section 3.3 into the controller as a representation of the uncertain nature of wind disturbances. While this approach might help drive the controller's optimal control input towards the true optimal input, it may also divert it away. Ultimately, this lack of knowledge on future disturbances is an unavoidable limitation to the controller's effectiveness.

To summarise, the construction of the MPC controller involves the propagation of states from the upstream turbine to the downstream turbine in an efficient manner, with disturbances generated using an Ornstein-Uhlenbeck model and obtains the sparse optimal control inputs based on obtaining maximum energy within the horizon.

5.4 Practical wake steering for wind farms using MPC controller and MHE algorithm

Under realistic conditions, the wake states fed into the controller will be estimated by the MHE algorithm since the true wake states are unknown. Therefore, to accurately test the wake steering controller for practical application, the estimates states from the MHE are utilised.

At each timestep, the MHE algorithm initially estimates the states of the wind farm wake using real-time measurements of streamwise velocity. These states are then provided to the controller. The controller then propagates these states with various control inputs and predicted disturbances to investigate the future power. Using an off-the-shelf optimiser such as `fmincon`, the controller provides the optimal control inputs for the horizon such that total farm energy is maximised. The earliest control inputs in the horizon are then applied to the turbines after which the process is repeated for the next timestep. Figure 23 provides a flowchart of this process.

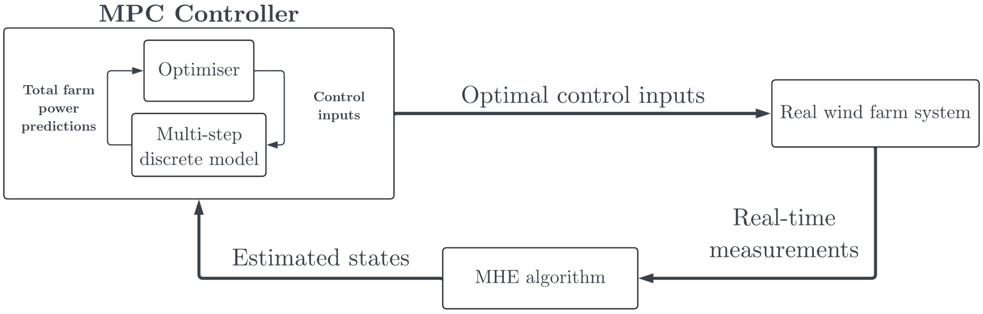

Figure 23: Wake steering for real wind farms using MHE algorithm and MPC controller

6 Results & Discussion

This Section details the performance of the MPC controller designed above in different scenarios and discusses the results. As there is no real wind measurement data from a wind farm, the multiple turbine discrete model was used to generate measurement data. This results an enhanced performance of the MHE algorithm and the MPC controller, as they are constructed from the same model. This can be avoided by testing the controller with wind measurement data generated by a different model, which can be part of future work to be done. The process of running the simulation are as follows:

1. Model the wind farm wake using the multiple turbine discrete model with the optimal control inputs - this is considered the 'real' wake
2. Obtain sensor data provided by the multiple turbine discrete model
3. Use the sensor data to estimate the wake states with the multiple turbine MHE algorithm
4. Feed the estimated wake states into the MPC controller to obtain the optimal control inputs for the next timestep

The value of P for the controller was set to 5, ensuring a relatively low computational cost while achieving acceptable levels of control accuracy.

6.1 Case 1: No disturbances with different turbine spacing

Observing the optimal yaw angle in this case, the MPC controller can be validated. This is because Case 1 is analogous to Figure 21, providing a means to validate the optimal yaw angle determined by the controller. Table 4 illustrates the optimal yaw angle observed from Figure 21 and from the MPC controller and the power increase from engaging in wake steering using the designed controller.

	Optimal yaw angle		Farm power output (MW)		
Turbine Spacing	Observed optimal yaw (°)	MPC controller optimal yaw (°)	No wake steering	With wake steering	% increase
2D	±25	25.03	3.3899	3.4458	1.65
5D	±18	17.94	4.1853	4.2253	0.956
8D	±6	5.95	4.6955	4.6961	0.013
10D	0	0	5.7014	5.7014	0

Table 4: Comparison of observed optimal yaw angle from Figure 21 and optimal yaw angle obtained by the MPC controller and the power increase from wake steering with $\sigma_0 = 0.361$ and $k_w = 0.08$

As observed from Table 4, the optimal yaw angles observed from Figure 21 closely match those obtained by the MPC controller. This verifies that the designed controller works as expected. Table 4 shows that with turbine spacing greater than $8D$, wake steering becomes ineffective in increasing total farm power as expected from the reasons previously mentioned in Section 5.2.4. The percentage increase in power output presented in Table 4 seem relatively modest compared to other literature on wake steering. Studies on scaled two-turbine arrays in wind tunnels by Adaramola et al. and Campagnolo et al. have demonstrated net power gains of 12% and 21% for 3D and 4D turbine spacing respectively [30] [31]. The reason behind the absurdly low power increase in Table 4 is due to the choice of parameters. These parameters highly affect the wake characteristic, therefore not only affect the optimal yaw angles but also the power increase from engaging in wake steering. Table 5 show the optimal yaw angle and the farm power output with and without wake steering for a different value of wake proportionality constant. The percentage power output increase in Table 5 is significantly higher when compared to Table 4, illustrating that the power increase from engaging in wake steering is highly dependent on the wake parameters used.

		Farm power output (MW)		
Turbine Spacing	MPC controller optimal control input	No wake steering	With wake steering	% increase
2D	0	2.8557	2.8557	0
5D	29.93	3.4329	3.2245	6.463
8D	25.10	3.9798	3.8059	4.569

Table 5: Optimal yaw angle obtained by the MPC controller and the power increase from wake steering with $\sigma_0 = 0.235$ and $k_w = 0.08$

Therefore, for more accurate studies on the effectiveness of the designed controller, parameter estimation must be pursued to accurately obtain the parameters that define the wake characteristic. More details about parameter estimation is done in [18].

6.2 Case 2: Disturbances with turbine spacing $5D$

Since the controller was successfully validated with no disturbances, disturbances can now be added into the model to verify the controller's response to unsteady inflow conditions. The controller was tested with three different scenarios of randomly generated disturbances using the Ornstein-Uhlenbeck model with the same wake parameters as Section 6.1. Figure 24 illustrates the disturbances that were utilised.

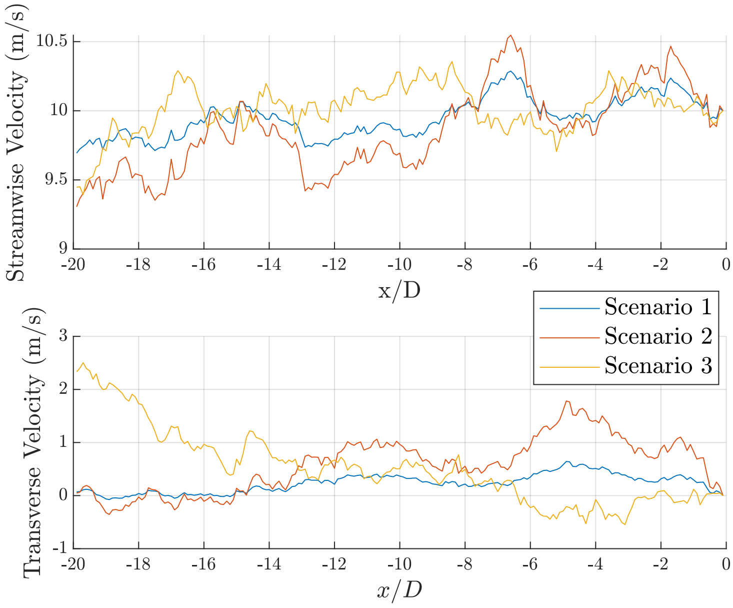

Figure 24: Three different disturbances generated

6.2.1 Scenario 1 disturbances

Observing Figure 25, engaging in wake steering using the controller leads to power performance gains on the wind farm. In the 75 seconds of simulation time, wake steering has increased the total wind farm power by 1.12% on average. This is relatively a low percentage increase, which again is due to the choice of wake parameters. Additionally, observing the optimal yaw inputs generated by the controller, show that the controller adjusts its inputs over time due to the effect of disturbances. From the power comparison plots, the power from wake steering is always higher than without, illustrating that the controller is effective in predicting future disturbances and adjusts its optimal control inputs accordingly. In scenario 1, the transverse disturbances exhibit relatively small variations, resulting in merely a variation of 1.83° in the controller's optimal yaw input for Turbine 1 over the simulation time. This small variation in optimal yaw angle is also attributed to the fact that turbines are spaced $5D$ apart, therefore eliminating the need for significant adjustments in the optimal yaw angle.

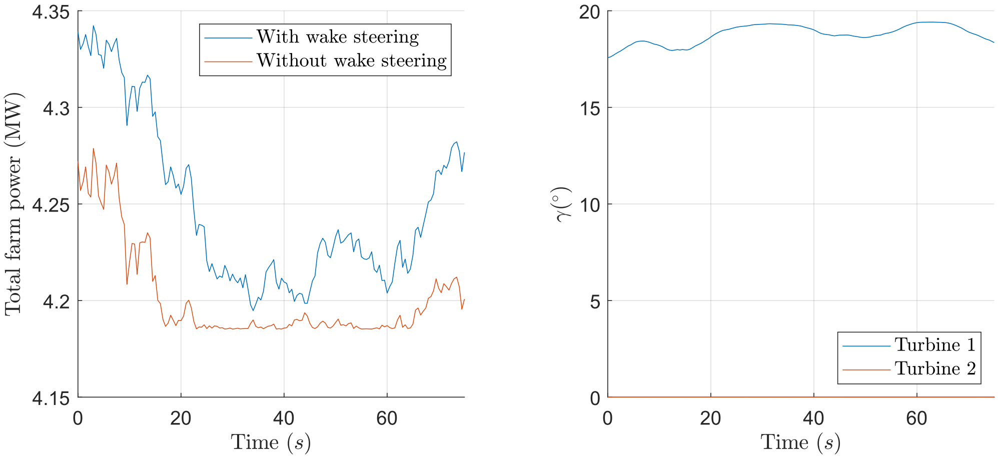

Figure 25: The total farm power with and without wake steering (left) and the optimal yaw angle inputs (right) for disturbance scenario 1

6.2.2 Scenario 2 disturbances

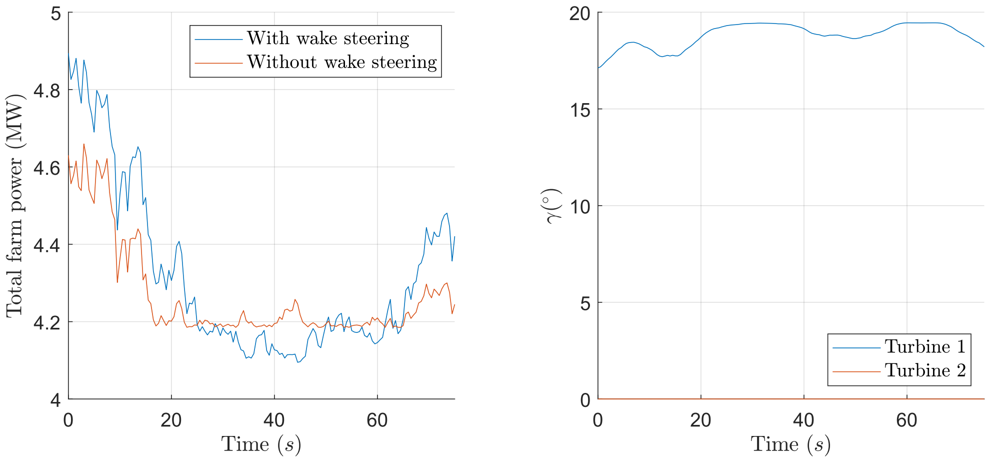

Figure 26: The total farm power with and without wake steering (left) and the optimal yaw angle inputs (right) for disturbance scenario 2

Scenario 2 disturbances are designed to have higher variation in transverse velocity disturbances than scenario 1 to investigate the controller's effectiveness with larger transverse disturbances. In this case, the controller is expected to have a larger variation in control input. This is observed in Figure 26 where there is now up to 2.33° variation of turbine 1 optimal control inputs over the simulation time. Likewise, the variation in optimal control inputs over time implies that the controller still takes into account the effect of disturbances. In scenario 2, the effect of wake steering has increased the total wind farm power by 1.17%. However, there is a period where the power of the wind farm with wake steering drops below the case without thus losing potential power gains due to ineffective control. This is due to the poor prediction of future disturbances by the controller resulting in an incorrect optimal control input. Therefore, the performance of the controller is highly dependent on its ability to predict these future disturbances.

6.2.3 Scenario 3 disturbances

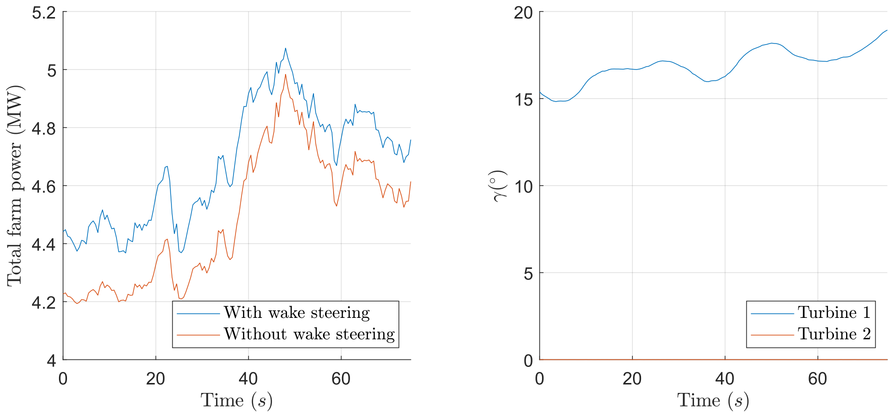

Figure 27: The total farm power with and without wake steering (left) and the optimal yaw angle inputs (right) for disturbance scenario 3

Scenario 3 disturbances are designed to have a similar variation in transverse velocity disturbance with scenario 2 to again test the controller's effectiveness with large transverse disturbances. Observing Figure 27, the variation in optimal inputs are large over the simulation time, with over 4.11° in variation. However, in this case, the power of the wind farm with wake steering is now always higher than without. This implies that the controller has effectively predicted the future disturbances in this situation. In this case, the action of wake steering increased total wind farm power production by 4.23%. This higher increase in power production percentage illustrates that the controller has been effective in predicting the future disturbances and has provided the optimal control inputs to efficiently steer the upstream turbine's wake away from the downstream turbine taking into account the influence of disturbances. This is another case that demonstrates that the effectiveness of the controller is highly dependent on its ability to predict future disturbances.

7 Future Work

7.1 Multiple turbine discrete model

In this project, the single turbine model was extended to multiple turbines with the effects of wake interactions modelled as disturbances. This assumption was made to seamlessly integrate the wake superposition model into the existing single turbine model. While a qualitative comparison was conducted, it is also crucial to perform a quantitative validation. A quantitative comparison could be performed by comparing wakes generated by the proposed model with those produced by FLORIS or other commercial software with the same wake parameters.

Additionally, improvements can also be made to the physics of the multiple turbine discrete model, particularly in addressing the assumption of a constant advective velocity, U_∞. This assumption is observed to break down over a large number of wake interactions where the advective velocity decreases significantly, leading to negative velocities in the near-wake region of downstream turbines. Therefore, a non-constant advection velocity wake model must be adopted to address this problem.

Furthermore, the current discrete wake models assume that the disturbances only affect the system at the turbine location and merely transport these disturbances downstream. This assumption gradually becomes less valid when modeling wakes over larger distances, such as when accounting for the wakes of a wind farm instead of an individual wind turbine. Therefore, it is crucial to extend the effect of disturbances to every point of the spatial grid for a more accurate wind farm wake model.

7.2 MPC controller

Due to time constraints, the effectiveness of the MPC controller was only assessed with a limited set of parameters and only for two turbines arranged in an array. Future work should include investigating the effectiveness of the MPC controller with a broader range of parameters. Additionally, future research could explore the application of yaw control to more turbines in diverse configurations to better simulate the conditions of commercial wind farms. This enables the development of a more practical wake steering controller. As the MPC controller is constructed using the discrete turbine model and tested with measurements obtained from the same model, thus its predictive accuracy over the horizon length is expected to be enhanced compared to using measurements from a real wind farm system. Future work should therefore involve a more fair investigation of the effectiveness of the controller, where LES data or real wind farm measurement data is utilised.

Furthermore, improvements can be made to the MPC controller with regards to its ability to predict future disturbances. As seen in Section 3.3, the controller's effectiveness highly depends on its ability to predict future disturbances. Currently, a random walk model is used to predict future disturbances. When the disturbance predictions are similar to future disturbances, the controller has been observed to produce effective control inputs as seen in Section 6.2.1 and Section 6.2.3. However, if disturbance predictions differ significantly from the actual future disturbances, the controller produces detrimental control inputs as seen in Section 6.2.2. A possible solution to deal with the random nature of disturbances, is by using a Stochastic MPC (SMPC) approach. The SMPC approach is designed to address the randomness of wind noise by using the probabilistic characteristics of wind noise to achieve optimal control inputs [16].

8 Conclusion

This project has fulfilled all three aforementioned objectives. Firstly, the existing single turbine wake model was successfully extended to accommodate multiple turbines, enabling the simulation of a wind farm flow field. This was achieved by utilising existing wake superposition theories that account for wake interactions between turbines. The multiple turbine model was qualitatively validated using flow fields available in other existing literature. While a qualitative validation was done, it is paramount to conduct a quantitative validation with other existing models available to investigate the accuracy of the results which can be part of future work.

Additionally, a MHE algorithm was also developed to effectively reconstruct the wakes of multiple turbines. This facilitates the reconstruction of the wind farm flow field with limited sensor information, thereby enabling the construction of a practical wake steering controller. The estimated wake was quantitatively compared to its actual counterpart to evaluate the estimation performance of the MHE algorithm. The error between estimated and the actual flow field is observed to be within tolerable limits and is eliminated in the far-wake region. Thus, with the high estimation performance of the MHE algorithm, it is adequate to be used for the purposes of building a practical wake steering controller.

The initial construction of a wake steering controller for a two-turbine configuration was also completed using a MPC approach. To effectively construct a wake steering controller, various wake parameters were explored to ensure sensible optimal solutions. The construction of the wake steering controller also involved defining a suitable cost function and employing innovative methods to address the time lag in wind farm systems, while keeping computational costs low. After the controller has been constructed it was tested against various scenarios. Due to the lack of real wind measurements from a wind farm, the measurements utilised were produced from the multiple turbine discrete model. The results of testing the controller showed gains of 1-4% in total wind farm power gain through wake steering. These are relatively low gains compared to literature, where 12% and 21% gains in total power for 3D and 4D turbine spacing respectively were observed in wind tunnel experiments [30] [31]. Investigating these modest power gains from using the designed controller, uncovered the fact that the selection of wake parameters highly influence the power gains. Thus, it is paramount to utilise the correct wake parameters to accurately assess the performance of the controller. Additionally, from testing the controller with different disturbances exposes a major limitation to the performance of the controller; which is its inability to perfectly predict future disturbances. With this limitation, it is impossible for the controller to obtain the real optimal control inputs given the effect of disturbances. Therefore, future work may involve the extension of the MPC approach to a SMPC approach to tackle the randomness of wind disturbances resulting in more certain power gains when employing the controller. Furthermore, while the controller shows promise in increasing power for a two-turbine wind farm, future studies should focus on extending the controller to operate with more turbines, aiming to replicate the scale of existing wind farms such as the Horns Rev offshore wind farm.

To conclude, this project has made significant contributions to the development of a robust, responsive and practical wake steering controller. However, with the progress so far limited to theoretical work, it is paramount to validate these advancements experimentally.

References

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

[2] International Energy Agency. Wind power for clean energy transitions. `https://www.iea.org/energy-system/renewables/wind`, 2023. Accessed: 2024-04-01.

[3] M. Gaumond, P.-E. Réthoré, S. Ott, A. Peña, A. Bechmann, and K. S. Hansen. Evaluation of the wind direction uncertainty and its impact on wake modeling at the horns rev offshore wind farm. *Wind Energy*, 17(8):1169–1178, 2014.

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

[5] Li R., Zhang J., and Zhao X. Dynamic wind farm wake modeling based on a bilateral convolutional neural network and high-fidelity les data. *Energy*, 258:124845, 2022.

[6] Simley E, Fleming P., Girard N., Alloin L., Godefroy E., and Duc T. Results from a wake-steering experiment at a commercial wind plant: investigating the wind speed dependence of wake-steering performance. Accessed: 2024-04-01.

[7] Bartłomiej P. Rak and R.B. Santos Pereira. Impact of the wake deficit model on wind farm yield: A study of yaw-based control optimization. *Journal of Wind Engineering and Industrial Aerodynamics*, 220:104827, 2022.

[8] N.O. Jensen. *A note on wind generator interaction.* Number 2411 in Risø-M. Risø National Laboratory, 1983.

[9] Ángel Jiménez, Antonio Crespo, and Emilio Migoya. Application of a les technique to characterize the wake deflection of a wind turbine in yaw. *Wind Energy*, 13(6):559–572, 2010.

[10] Richard M. Murray. State estimation, 2022. Accessed: 2024-04-28.

[11] Alexander Ronald, G. Campani, S. Dinh, and F.V. Lima. Challenges and opportunities on nonlinear state estimation of chemical and biochemical processes. *Processes*, 8(11), 2020.

[12] Shri Bhagwan, Anil Kumar, and J. S. Soni. A review on: Pid controller. *International Journal of Research in Mechanical, Mechatronics and Engineering*, 2016.

[13] Anthony K. H. Fundamentals of pid control. `https://pdhonline.com/courses/e331/e331content.pdf`, 2020. Accessed: 2024-05-23.

[14] MathWorks. What is model predictive control (mpc)? `https://uk.mathworks.com/help/mpc/gs/what-is-mpc.html`, 2024. Accessed: 2024-05-23.

[15] Dale E. Seborg. *Process dynamics and control.* Wiley, Hoboken, N.J, 3rd ed., international student version. edition, 2011.

[16] Peng Lu, Ning Zhang, Lin Ye, Ershun Du, and Chongqing Kang. Advances in model predictive control for large-scale wind power integration in power systems. *Advances in Applied Energy*, 14:100177, 2024.

[17] Carl R. Shapiro, Dennice F. Gayme, and Charles Meneveau. Modelling yawed wind turbine wakes: a lifting line approach. *Journal of fluid mechanics*, 841, 2018.

[18] Aashish Narang. Temporal wake modelling with exogenous disturbances, real-time wake reconstruction, and paramter estimation for wind turbines', June 2024.

[19] Ivan Markovsky and Sabine Van Huffel. Overview of total least squares methods. Technical Report 263855, University of Southampton, 2007. Accessed: yyyy-mm-dd.

[20] Sergey Obukhov, Emad M. Ahmed, Denis Y. Davydov, Talal Alharbi, Ahmed Ibrahim, and Ziad M. Ali. Modeling wind speed based on fractional ornstein-uhlenbeck process. *Energies*, 14(17), 2021.

[21] Jim Y.J. Kuo, David A. Romero, and Cristina H. Amon. A mechanistic semi-empirical wake interaction model for wind farm layout optimization. *Energy*, 93:2157–2165, 2015.

[22] Douwe J Renkema. Validation of wind turbine wake models. *Master of Science Thesis, Delft University of Technology*, 19, 2007.

[23] Dezhi Wei, Weiwen Zhao, Decheng Wan, and Qing Xiao. A new method for simulating multiple wind turbine wakes under yawed conditions. *Ocean Engineering*, 239:109832, 2021.

[24] Tony Burton, Nick Jenkins, Ervin Bossanyi, David Sharpe, Michael Graham, and Tony Burton. *Wind energy handbook.* Wiley, Hoboken, NJ, third edition / tony burton, nick jenkins, ervin bossanyi, david sharpe, michael graham. edition, 2021.

[25] Mustafa Cavcar. The international standard atmosphere (isa). *Anadolu University, Turkey*, 30(9):1–6, 2000.

[26] Zhiwen Deng, Chang Xu, Zhihong Huo, Xingxing Han, and Feifei Xue. Yaw optimisation for wind farm production maximisation based on a dynamic wake model. *Energies*, 16(9), 2023.

[27] Luis Martinez Tossas, Emmanuel Branlard, and Jason Jonkman. Wind turbine wakes under high thrust coefficients. In *APS Division of Fluid Dynamics Meeting Abstracts*, APS Meeting Abstracts, page W05.013, January 2020.

[28] Hao Liu, Jixing Chen, Jing Zhang, Yining Chen, Yafeng Wen, Xiaoyang Zhang, Zhongjie Yan, and Qingan Li. Study on atmospheric stability and wake attenuation constant of large offshore wind farm in yellow sea. *Energies*, 16(5), 2023.

[29] M. Gaumond, Pierre-Elouan Réthoré, S. Ott, A. Peña, A. Bechmann, and Kurt Schaldemose Hansen. Evaluation of the wind direction uncertainty and its impact on wake modeling at the horns rev offshore wind farm. *Wind Energy*, 17, 08 2014.

[30] Muyiwa Adaramola and P.-Å Krogstad. Experimental investigation of wake effects on wind turbine performance. *Renewable Energy*, 36:2078–2086, 08 2011.

[31] Filippo Campagnolo, Vlaho Petrović, Carlo Bottasso, and Alessandro Croce. Wind tunnel testing of wake control strategies. pages 513–518, 07 2016.

Appendix

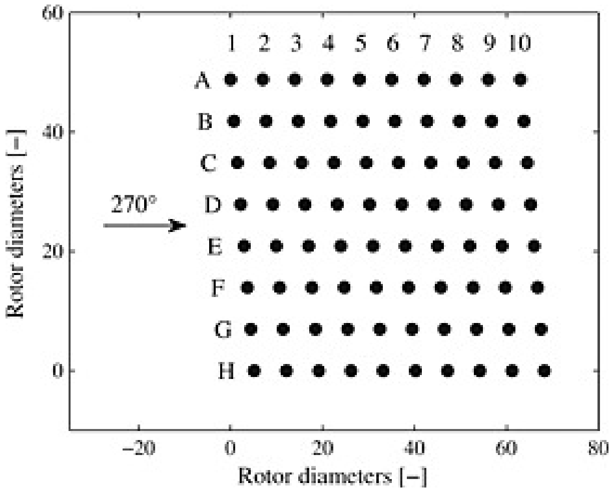

Figure 28: Layout of the Horns Rev offshore wind farm [29]