

Adjoint optimisation for wind farm flow control with a free-vortex wake model

Maarten J. van den Broek^{a,*}, Delphine De Tavernier^b, Benjamin Sanderse^c,
Jan-Willem van Wingerden^a

^a Delft Centre for Systems and Control, TU Delft, Mekelweg 2, 2628 CD, Delft, The Netherlands

^b Department of Flow Physics and Technology, Wind Energy, TU Delft, P.O. Box 5058, 2600 GB, Delft, The Netherlands

^c Scientific Computing, CWI, P.O. Box 94079, 1090 GB, Amsterdam, The Netherlands

ARTICLE INFO

Dataset link: <https://doi.org/10.4121/20278620>, <https://github.com/TUdelft-DataDrivenControl/vortexwake>, <https://doi.org/10.4121/20278590>

Keywords:

Free-vortex wake
Wake mixing
Wake redirection
Adjoint optimisation

ABSTRACT

Wind farm flow control aims to improve wind turbine performance by reducing aerodynamic wake interaction between turbines. Dynamic, physics-based models of wind farm flows have been essential for exploring control strategies such as wake redirection and dynamic induction control. Free-vortex methods can provide a computationally efficient way to model wind turbine wake dynamics for control optimisation. We present a control-oriented free-vortex wake model of a 2D and 3D actuator disc to represent wind turbine wakes. The novel derivation of the discrete adjoint equations allows efficient gradient evaluation for gradient-based optimisation in an economic model-predictive control algorithm. Initial results are presented for mean power maximisation in a two-turbine case study. An induction control signal is found using the 2D model that is roughly periodic and supports previous results on dynamic induction control to stimulate wake mixing. The 3D model formulation effectively models a curled wake under yaw misalignment. Under time-varying wind direction, the optimisation finds solutions demonstrating both wake steering and a smooth transition to greedy control. The free-vortex wake model with gradient information shows potential for efficient optimisation and provides a promising way to further explore dynamic wind farm flow control.

1. Introduction

Large, densely spaced wind farms are designed and constructed to make use of limited offshore parcels. Within these farms, aerodynamic interaction between wind turbines reduces power production and increases fatigue loading as turbulent, low-energy wakes travel through the farm and negatively affect downstream turbines. Wind farm topology is designed to minimise these interactions, but is inflexible to cope with dynamic, varying atmospheric conditions [1]. The purpose of wind farm control is to minimise the detrimental effects of aerodynamic interaction between wind turbines in a wind farm.

Control strategies for wind farm control can be roughly divided in three categories: wake redirection by yaw misalignment, induction control, and wake mixing strategies [2]. First, the use of yaw misalignment with respect to the free-stream wind direction to redirect wakes downstream has been shown to effectively improve performance under steady conditions in both wind tunnel experiments [3–5] and field studies [6–10]. Second, sinusoidal thrust variations, and consequent induction variations, through collective pitch control have

been found to improve wake recovery in an LES study [11] and in wind tunnel experiments [12]. Finally, recent developments in stimulating wake mixing have shown the potential to improve upon collective pitch variations with the helix approach, an individual pitch control strategy [13].

Control-oriented surrogate models are often at the core of wind farm control algorithms. Steady-state engineering wake models, such as those that have been implemented in FLORIS [14], are the current industry standard. These include, for example, the Gaussian model [15] or a steady representation of curled wake dynamics [16]. As steady-state wake representations are limited in realistic time-varying conditions, dynamic effects have been added to these engineering wake models to improve upon the steady-state results by including dynamic wake meandering [17] or using Lagrangian particles to incorporate wake dynamics [18].

Several studies have also developed physics-based dynamic models for wind farm flow control, especially using the adjoint method to efficiently calculate gradient information for a scalar objective function with a large number of parameters. The patterns found through

* Corresponding author.

E-mail addresses: m.j.vandenbroek@tudelft.nl (M.J. van den Broek), d.a.m.detavernier@tudelft.nl (D. De Tavernier), b.sanderse@cwil.nl (B. Sanderse), j.w.vanwingerden@tudelft.nl (J.W. van Wingerden).

<https://doi.org/10.1016/j.renene.2022.10.120>

Received 24 August 2022; Received in revised form 27 October 2022; Accepted 29 October 2022

Available online 4 November 2022

0960-1481/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

optimal control studies with adjoint large-eddy simulations [19,20] provided the basis for dynamic induction control methods, although these simulations are too computationally expensive for real-time control applications [11,21]. WFSim provides a 2D Navier–Stokes based wind-farm flow model for control [22], which has then been used for adjoint optimisation of induction control [23]. FRED [24] builds on the results from WFSim to simulate wind farm performance with the adjoint for gradient calculation [25,26]. However, the 2D physics inherent in this model lack the curled wake dynamics of a wind turbine under yaw misalignment [27–30] and could not accurately model the effects of wake redirection [26].

In contrast to conventional computational fluid dynamic approaches, free-vortex methods use the vorticity formulation of the Navier–Stokes equations to model wind turbine wakes with Lagrangian elements [31]. Within the field of wind energy, free-vortex wake models have been used to study floating wind turbines and wake dynamics [32] and to study dynamic wake control methods and analyse wake stability [33]. The latter uses the CACTUS code which has been shown to be mostly accurate for near-wake regions [34]. Even though vortex methods are generally more accurate in near-wake regions, a free-vortex ring method has been used to model far wake dynamics for both fixed-bottom and floating wind turbines [35]. Additionally, an actuator-disc model based on discretised vortex rings has been shown to capture the 3D dynamics of the kidney-shaped wake under yaw misalignment [36].

In this paper, we propose the use of the free-vortex wake method as a computationally efficient, physics-based surrogate wake model for control optimisation, especially coupled with the adjoint for efficient evaluation of the gradient. This work aims to extend the possibilities for optimisation of induction and yaw signals for dynamic wind farm flow control. For that purpose, the contribution of this paper is threefold: (i) a control-oriented free-vortex wake model of an actuator disc in 2D and 3D with the discrete adjoint for gradient computation, (ii) an economic model-predictive control implementation for dynamic wind farm flow control, and (iii) initial results that demonstrate dynamic induction control and yaw control under time-varying wind direction.

The remainder of the paper is structured as follows. A 2D and 3D free-vortex model of an actuator disc to represent a wind turbine wake is presented in Section 2. The non-linear optimisation problem for economic model-predictive control is formulated in Section 3 together with the discrete adjoint method for calculating the gradient. Results are discussed in Section 4, which provides an overview of operation under steady conditions followed by receding horizon control optimisation of time-varying axial induction and yaw signals. Finally, conclusions are presented in Section 5.

2. Control-oriented free-vortex wake model

The general formulation for the control-oriented free-vortex wake (FVW) representation is described in Section 2.1. Aspects specific to the 2D and 3D implementations are then defined in Section 2.2 and Section 2.3, respectively. The convergence and validation of the method for the numerical parameters used in this paper is provided in Appendix B.

2.1. General formulation

An actuator-disc representation of a wind turbine is implemented with the free-vortex method in both a two-dimensional (2D) and three-dimensional (3D) formulation. The free-vortex method is based on Lagrangian particles that advect downstream. These particles induce a velocity based on their associated vorticity. The resultant flow velocity may be calculated at any position based on the free-stream velocity and the sum of induced velocities. For a further description of the fundamentals, the reader is referred to aerodynamic literature, such as [31].

The use of the free-vortex wake method requires the assumption of inviscid and incompressible flow. The actuator disc is assumed to be uniformly loaded so it only releases vorticity along its edge [31]. For the 2D model, the wake is modelled by releasing pairs of vortex points at the edge of the actuator disc at every simulation time-step. The 3D code is based on the simulation of discretised vortex rings with vortex filaments, adapted from the model described by Berdowski et al. [36]. For convenience, all units have been non-dimensionalised by the rotor diameter and inflow speed.

A system with fixed dimensionality is preferred for control optimisation, therefore the wake models are set up with n_e elements per vortex ring and a fixed number of vortex rings n_r . The number of points to define the vortex elements n_p equals n_e in 2D and $n_e + 1$ in 3D. The spatial dimension of the simulation is n_d , which equals either two or three. The number of turbines modelled is n_t and the number of control parameters per turbine is n_c . For example, the total number of states is $n_s = 2n_r n_p n_d + n_r n_e + n_t n_c$ for a single wake modelled with the FVW, where additional virtual turbines are evaluated using the flow velocity without including their effect on the wake.

We set up the model as a non-linear state–space system in discrete time,

$$q_{k+1} = f(q_k, m_k), \quad (1)$$

$$y_k = g(q_k, m_k), \quad (2)$$

where for every discrete time step k the updated state $q_{k+1} \in \mathbb{R}^{n_s}$ and the output vector $y_k \in \mathbb{R}^{n_t}$ are a function of the current state $q_k \in \mathbb{R}^{n_s}$ and the control inputs $m_k \in \mathbb{R}^{n_c}$. The state vector is built up as

$$q = \begin{bmatrix} X \\ \Gamma \\ U \\ M \end{bmatrix}, \quad (3)$$

from the vortex element positions $X \in \mathbb{R}^{n_r n_p n_d}$, the vortex element circulations $\Gamma \in \mathbb{R}^{n_r n_e}$, the stored free-stream velocity $U \in \mathbb{R}^{n_r n_p n_d}$, and the control inputs from the previous time step $M \in \mathbb{R}^{n_c}$. The full control vector m is defined as

$$m = \begin{bmatrix} a_0 \\ \psi_0 \\ a_1 \\ \psi_1 \end{bmatrix}, \quad (4)$$

for a two-turbine configuration with axial induction a and turbine yaw angle ψ .

States corresponding to a ring are indicated with a subscript, rings are indexed with a superscript starting from 0. This allows, for example, the convenient relation of a point $x_i^{(b)} \in \mathbb{R}^{n_d}$ to the point in the same position in the previous ring $x_i^{(b-1)}$, or all points in a ring $X^{(b)} \in \mathbb{R}^{n_p n_d}$ to all points in the previous ring $X^{(b-1)}$.

For all rings except the first ($b \geq 1$), the position update is calculated from the position of the previous ring with simulation time step h , the stored inflow velocity $u_\infty \in \mathbb{R}^{n_d}$, and the total induced velocity $u_{\text{ind}} \in \mathbb{R}^{n_d}$,

$$x_i^{(b)} \Big|_{k+1} = x_i^{(b-1)} \Big|_k + h \left(u_{\infty, i}^{(b-1)} + u_{\text{ind}}(x_i^{(b-1)}, q) \right) \Big|_k. \quad (5)$$

The velocity u_{ind} induced at any point x is the sum of the contribution from all vortex elements in the system

$$u_{\text{ind}}(x, q) = \sum_{b=0}^{n_r-1} \sum_{j=1}^{n_e} u_{i_j}^{(b)}, \quad (6)$$

where $u_i \in \mathbb{R}^{n_d}$ is the velocity induced by a single vortex element. The generation of new vortex elements in the first ring $X^{(0)}$ and the velocity induced by a single vortex element u_i is defined for 2D and 3D in Section 2.2 and Section 2.3, respectively.

The vector Γ contains the vortex strength Γ for all elements in all rings. The vortex strength of the first ring is given according to

$$\Gamma_i^{(0)}(\mathbf{q}, \mathbf{m}) = \frac{d\Gamma}{dt} h = c'_t(a) \frac{1}{2} (\mathbf{u}_r \cdot \mathbf{n}(\psi))^2 h \quad \text{for } i = 1, 2, \dots, n_e. \quad (7)$$

In this expression, \mathbf{u}_r is the average wind speed at the rotor. The vector $\mathbf{n} \in \mathbb{R}^{n_d}$ is a unit vector orthogonal to the rotor disc, pointing in the downstream direction, with the rotation matrix $\mathbf{R}_z \in \mathbb{R}^{n_d \times n_d}$ and axis-aligned unit vector $\mathbf{e}_x \in \mathbb{R}^{n_d}$,

$$\mathbf{n}(\psi) = \mathbf{R}_z(\psi) \mathbf{e}_x. \quad (8)$$

The local thrust coefficient c'_t , is calculated from the axial induction a as

$$c'_t(a) = \begin{cases} \frac{4a(1-a)}{(1-a)^2} = \frac{4a}{1-a} & \text{if } a \leq a_t, \\ \frac{c_{t1} - 4(\sqrt{c_{t1}} - 1)(1-a)}{(1-a)^2} & \text{if } a > a_t, \end{cases} \quad (9)$$

where the induction a_t at the transition point is

$$a_t = 1 - \frac{1}{2} \sqrt{c_{t1}}, \quad (10)$$

and the parameter $c_{t1} = 2.3$. The thrust coefficient calculation is based on momentum theory with a transition to a linear approximation for high induction values that is an empirical correction based on the Glauert correction [37]. Vortex strength of subsequent rings is inherited downstream,

$$\Gamma_i^{(b)}|_{k+1} = \Gamma_i^{(b-1)}|_k \quad \text{for } i = 1, 2, \dots, n_e \text{ and } b = 1, 2, \dots, n_r - 1. \quad (11)$$

Ring zero is initialised at the turbine position with the free-stream velocity, which may vary over space and simulation time,

$$\mathbf{U}^{(0)}|_{k+1} = \mathbf{u}_\infty(\mathbf{X}^{(0)}, k). \quad (12)$$

The inflow velocity is then propagated downstream with the state update

$$\mathbf{U}^{(b)}|_{k+1} = \mathbf{U}^{(b-1)}|_k \quad \text{for } b = 1, 2, \dots, n_r - 1. \quad (13)$$

The vector \mathbf{M} is an augmentation of the system state to store controls for power calculation at the next time-step,

$$\mathbf{M}|_{k+1} = \mathbf{m}_k. \quad (14)$$

This avoids a direct feed-through of control actions to the output function.

The output vector \mathbf{y} contains the power of all turbines as

$$\mathbf{y} = \begin{bmatrix} P_0 \\ P_1 \end{bmatrix}, \quad (15)$$

for a two-turbine case. The power P at turbine i is calculated as

$$P_i = \frac{1}{2} c'_p(a) A_r (\mathbf{u}_r \cdot \mathbf{n}(\psi))^3, \quad (16)$$

with the local power coefficient c'_p , rotor area A_r , the disc-averaged velocity $\mathbf{u}_r \in \mathbb{R}^{n_d}$, and the yaw angle ψ . The local power coefficient is calculated with the induction factor a as

$$c'_p(a) = \frac{4a(1-a)^2}{(1-a)^3} = \frac{4a}{1-a}. \quad (17)$$

For the disc-averaged velocity, we distribute n_u points over a disc representing the turbine according to an equal-area distribution [38] and rotate the disc over the yaw angle. The rotor-disc averaged velocity is then

$$\mathbf{u}_r = \frac{1}{n_u} \sum_{i=1}^{n_u} (\mathbf{u}_\infty(\mathbf{x}_i, \mathbf{q}) + \mathbf{u}_{\text{ind}}(\mathbf{x}_i, \mathbf{q})), \quad (18)$$

where the local free-stream flow is calculated as an average from neighbouring points weighted by distance,

$$\mathbf{u}_\infty(\mathbf{x}, \mathbf{q}) = \sum_{i=0}^{n_p} \sum_{b=0}^{n_r} \bar{w}_i^{(b)} \mathbf{u}_{\infty,i}^{(b)}, \quad (19)$$

with normalised weights $\bar{w}_i^{(b)}$,

$$w_i^{(b)} = \exp(-10 \|\mathbf{x} - \mathbf{x}_i^{(b)}\|), \quad (20)$$

$$\bar{w}_i^{(b)} = \frac{w_i^{(b)}}{\sum_{i=0}^{n_p} \sum_{b=0}^{n_r-1} w_i^{(b)}}. \quad (21)$$

For calculation of power of a virtual turbine – one that does not act on the flow simulation, but is included for purposes of optimisation – we lower the disc-averaged velocity by the induction factor

$$\mathbf{u}_r^* = (1 - a) \mathbf{u}_r. \quad (22)$$

2.2. Two-dimensional model specifics

The $n_e = 2$ vortex elements of the first ring are initiated at the edge of the rotor disc with radius r

$$\mathbf{x}_0^{(0)}(\psi)|_{k+1} = \mathbf{R}_z(\psi_k) \begin{bmatrix} 0 \\ r \end{bmatrix}, \quad \mathbf{x}_1^{(0)}(\psi)|_{k+1} = \mathbf{R}_z(\psi_k) \begin{bmatrix} 0 \\ -r \end{bmatrix}, \quad (23)$$

where $\mathbf{R}_z(\psi)$ is the rotation matrix for a rotation of an angle ψ around the z -axis,

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix}. \quad (24)$$

The velocity \mathbf{u}_i induced at point \mathbf{x}_0 by a single vortex element located at \mathbf{x}_1 in 2D is calculated with the Biot–Savart law as

$$\mathbf{u}_i(\mathbf{x}_0, \mathbf{x}_1) = \begin{bmatrix} -r_y \\ r_x \end{bmatrix} \left(\frac{\Gamma}{2\pi} \frac{1}{\|\mathbf{r}\|^2} \right) \left(1 - \exp\left(-\frac{\|\mathbf{r}\|^2}{\sigma^2}\right) \right), \quad (25)$$

where the relative position \mathbf{r} is

$$\mathbf{r} = \mathbf{x}_1 - \mathbf{x}_0. \quad (26)$$

A Gaussian core with core size σ is included to regularise singular behaviour of the induced velocity close to the vortex element.

2.3. Three-dimensional model specifics

At every time-step, the vortex filaments that make up a new vortex ring discretised with n_e elements are distributed over a circle with radius r , with yaw angle ψ ,

$$\mathbf{x}_i^{(0)}(\psi)|_{k+1} = \mathbf{R}_z(\psi_k) \begin{bmatrix} 0 \\ r \cos(2\pi \frac{i}{n_e}) \\ r \sin(2\pi \frac{i}{n_e}) \end{bmatrix} \quad \text{for } i = 0, 1, \dots, n_e, \quad (27)$$

where $\mathbf{R}_z(\psi)$ is the rotation matrix for a rotation of an angle ψ around the z -axis,

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (28)$$

The induced velocity \mathbf{u}_i at a point \mathbf{x}_0 is calculated with Biot–Savart law from a single vortex element starting at \mathbf{x}_1 and ending at \mathbf{x}_2 , with vortex strength Γ ,

$$\mathbf{u}_i(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2) = \left(\frac{\Gamma}{4\pi} \frac{\mathbf{r}_1 \times \mathbf{r}_2}{\|\mathbf{r}_1 \times \mathbf{r}_2\|^2} \right) \left(\mathbf{r}_0 \cdot \left(\frac{\mathbf{r}_1}{\|\mathbf{r}_1\|} - \frac{\mathbf{r}_2}{\|\mathbf{r}_2\|} \right) \right) \left(1 - \exp\left(-\frac{\|\mathbf{r}_1 \times \mathbf{r}_2\|^2}{\sigma^2 \|\mathbf{r}_0\|^2} \right) \right), \quad (29)$$

where the relative positions \mathbf{r} are defined as

$$\mathbf{r}_0 = \mathbf{x}_2 - \mathbf{x}_1, \quad (30)$$

$$\mathbf{r}_1 = \mathbf{x}_1 - \mathbf{x}_0, \quad (31)$$

$$\mathbf{r}_2 = \mathbf{x}_2 - \mathbf{x}_0. \quad (32)$$

A Gaussian core with core size σ is included to regularise singular behaviour of the induced velocity close to the vortex filament.

3. Optimisation for power maximisation

The free-vortex wake model described in the previous section is implemented as a novel surrogate model for dynamic wind farm flow control. Wind turbine power maximisation is introduced in Section 3.1 in an economic model-predictive control setting. The associated non-linear optimisation problem is formulated in Section 3.2. The derivation of the discrete adjoint for calculation of the gradient is described in Section 3.3, followed by the choice of a gradient-based optimisation method to solve the non-linear problem in Section 3.4.

3.1. Economic model-predictive control

The conventional model-predictive control (MPC) approach is a model-based optimisation of control signals to drive an objective functional to zero, for example for optimal tracking of a reference signal. However, for maximisation of wind farm power production, the optimal objective value is not known a priori, leading to an economic problem formulation. The economic MPC (EMPC) approach considers optimisation of an objective to an unknown extremum. This optimisation problem is conventionally solved in a receding horizon setting with a finite prediction horizon. After optimisation, the first (set of) control(s) is implemented and the problem is shifted and solved again up to the new horizon [39].

One problem with optimisation to a finite horizon is that the optimisation considers the prediction horizon as the end of time. Therefore, control actions that prioritise gain within the horizon may be optimal, although they would have undesired consequences post-horizon. This is known as the turnpike effect [40], where a solution stays close to the optimal trajectory for most of the window but diverges towards the horizon. These finite horizon effects may be treated by terminal constraints or terminal conditions [39]. For example, the control signal has been kept constant towards the horizon to limit undesired effects in wind farm control [23] or a terminal condition on rotor kinetic energy has been used to regularise optimisation results for wind turbine control [41]. Given a sufficiently long prediction horizon, EMPC has been shown to also converge without terminal constraints [42].

In this paper, the turnpike effects are treated by considering sufficiently long prediction horizons within the receding horizon setting, so as not to require terminal constraints. The control problem is formulated in a non-linear EMPC setting without terminal conditions with the goal of maximising mean power production over time.

3.2. Objective function definition

A non-linear minimisation problem with a scalar objective function J is constructed to find the set of optimal controls $\mathbf{m}_{k_0+i} \in \mathbb{R}^{n_m}$, with $n_m \leq n_t n_c$ the number of free controls and $i = 0, 1, \dots, N_h$. The objective is the total power output over a prediction horizon of N_h steps from the current step k_0 ,

$$\min_{\mathbf{m}_k} \sum_{k=k_0}^{k_0+N_h} J(\mathbf{q}_k, \mathbf{m}_k) = \min_{\mathbf{m}_k} \sum_{k=k_0}^{k_0+N_h} \mathbf{Q} \mathbf{y}_k(\mathbf{q}_k, \mathbf{m}_k) + \Delta \mathbf{m}_k^T \mathbf{R} \Delta \mathbf{m}_k, \quad (33)$$

where \mathbf{y}_k contains the power of modelled and virtual turbines, $\Delta \mathbf{m}_k = \mathbf{m}_k - \mathbf{m}_{k-1}$ is the change in control value between time steps, and $\mathbf{Q} \in \mathbb{R}^{1 \times n_t}$ and $\mathbf{R} \in \mathbb{R}^{n_m \times n_m}$ are weights to balance power output and actuation cost. A linear sum of power is chosen for mean power maximisation because power is already a positive objective function. A quadratic functional would more heavily weight peaks in power production and be suboptimal for maximisation of mean power. The output weight is chosen negative ($\mathbf{Q} < 0$) so that power is maximised for minimisation of the objective. The input weight \mathbf{R} functions as a regularisation term and aids convergence to suitable control solutions by smoothing the optimisation landscape.

3.3. Discrete adjoint method for constructing the gradient

The gradient of the objective function is calculated following the discrete adjoint method [43] because the method scales well for a large number of input sensitivities. We take the non-linear state–space system in (1) and define the objective function $J_k = J(\mathbf{q}_k, \mathbf{m}_k)$ at time-step k , such that the total objective function J_{total} is accumulated over a number of steps N_h ,

$$J_{\text{total}} = J_{N_h} + \sum_{i=0}^{N_h-1} J_i, \quad (34)$$

where $i = 0$ at the current time-step $k = k_0$. This is the total objective function to be minimised in the optimisation problem in (33).

To derive the adjoint system, we extend the objective function with adjoint states and system constraint,

$$\bar{J}_{\text{total}} = J_{N_h} + \sum_{i=0}^{N_h-1} (J_i + \lambda_{i+1}^T (f_i - \mathbf{q}_{i+1})), \quad (35)$$

where the adjoint states λ can be chosen freely because $f_i - \mathbf{q}_{i+1} = 0$. Since $J_k = J(\mathbf{q}_k, \mathbf{m}_k)$, a differential change $\delta \bar{J}_{\text{total}}$ can be expanded in terms of changes in \mathbf{q} and \mathbf{m} as:

$$\begin{aligned} \delta \bar{J}_{\text{total}} = & \left(\frac{\partial J_{N_h}}{\partial \mathbf{q}_{N_h}} - \lambda_{N_h}^T \right) \delta \mathbf{q}_{N_h} + \frac{\partial J_{N_h}}{\partial \mathbf{m}_{N_h}} \delta \mathbf{m}_{N_h} \\ & + \sum_{i=0}^{N_h-1} \left(\left(\frac{\partial J_i}{\partial \mathbf{q}_i} + \lambda_{i+1}^T \frac{\partial f_i}{\partial \mathbf{q}_i} - \lambda_i^T \right) \delta \mathbf{q}_i + \left(\frac{\partial J_i}{\partial \mathbf{m}_i} + \lambda_{i+1}^T \frac{\partial f_i}{\partial \mathbf{m}_i} \right) \delta \mathbf{m}_i \right), \end{aligned} \quad (36)$$

We then choose the adjoint states to be

$$\lambda_{N_h}^T = \frac{\partial J_{N_h}}{\partial \mathbf{q}_{N_h}}, \quad \lambda_i^T = \frac{\partial J_i}{\partial \mathbf{q}_i} + \lambda_{i+1}^T \frac{\partial f_i}{\partial \mathbf{q}_i}, \quad \lambda_0^T = \mathbf{0}, \quad (37)$$

such that the variations due to \mathbf{q} in (36) are cancelled out. The adjoint states are solved for by propagation backwards in time, starting from the final adjoint state. The gradient of the objective function parts J_k to the input can then be calculated from these adjoint states

$$\nabla J_{N_h} = \frac{\delta J_{N_h}}{\delta \mathbf{m}_{N_h}}, \quad \nabla J_i = \frac{\partial J_i}{\partial \mathbf{m}_i} + \left(\frac{\partial f_i}{\partial \mathbf{m}_i} \right)^T \lambda_{i+1}. \quad (38)$$

The total gradient $\nabla \bar{J}_{\text{total}}$ with respect to all control parameters \mathbf{m}_i is then constructed as

$$\nabla \bar{J}_{\text{total}} = \sum_{i=0}^{N_h} \nabla J_i. \quad (39)$$

The partial derivatives of the state update and output function with respect to the full model state and controls are stored in memory during the forward simulation of the model. These partial derivatives are provided for the given model and objective function in Appendix A.

The evaluation of the gradient thus requires a single forward simulation with evaluation of the partial derivatives and a single backward pass to solve for the adjoint states and construct the gradient. In that sense, this method of gradient evaluation is considerably more efficient than finite difference methods as the computational cost of the discrete adjoint increases only minimally with the number of control parameters for which the derivative is required. The computational cost of gradient evaluation with the discrete adjoint primarily scales with the expense of the forward simulation and the associated partial derivatives.

3.4. Gradient-based optimisation methods

The availability of the gradient allows the use of gradient-based optimisation techniques for control optimisation. Exploration of the objective function shows that it is non-linear and non-convex, with almost flat regions and numerous local minima. Initial experiments were run with L-BFGS-B optimisation [44] as was also used in the

Table 1
FVW parameters for 2D and 3D case.

		2D	3D
Time-step	h	0.2	0.3
Core size	σ	0.1	0.16
Number of rings	n_r	60	40
Elements per ring	n_c	2	16

work by Munters and Meyers [21]. However, this optimiser appeared sensitive to initialisation at local maxima and to convergence to local minima.

The Adam optimiser [45] is a gradient-based method often used in machine learning for optimisation of neural network weights, where it is applied for gradient descent with noisy gradients in complex optimisation landscapes. It uses a momentum approach to accelerate gradient descent and has proven to be less sensitive to the choice of initial guess and local minima. Within this work, we use the Adam optimiser with the default parameters; a maximum step size $\alpha = 0.001$ and the default decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Tuning of these parameters may still improve performance. The yaw angle is on a different order of magnitude than axial induction. Therefore, it is scaled by a factor 10^{-2} in the optimisation, so that the step size covers a similar range of the allowable range of induction value and yaw angle.

4. Results and discussion

A brief overview of the 2D and 3D FVW under steady conditions is given in Section 4.1 to illustrate the test case configuration and provide a steady baseline for control performance. This is followed by two example cases to demonstrate the use of the FVW as a novel surrogate model for control optimisation in the receding horizon setting described previously; a 2D case with induction control is provided in Section 4.2 and a 3D case for yaw control under time-varying wind direction in Section 4.3. Finally, Section 4.4 discusses finite horizon effects in EMPC for wind farm control.

4.1. Steady state operation

We define a two-turbine case for evaluating control optimisation with the FVW, starting with steady-state control characteristics. The two turbines are spaced $5D$ apart, where D is the rotor diameter, aligned with the uniform unit inflow. The upstream turbine is modelled with the FVW and the virtual downstream turbine performance is evaluated using the flow velocity over the rotor area at the downstream position. The parameters for the FVW are provided in Table 1. An exploration of parameter sensitivity is supplied in Appendix B.

Fig. 1 shows a 2D FVW simulation with an induction factor $a = 0.33$ and without yaw misalignment. The pairs of vortex points provide the basis for the simulation and allow calculation of a dense velocity field. Disturbances in the far wake are the result of numerical instabilities. It is also notable that the wake is quite wide as is expected for planar flow.

The 3D FVW produces a vortex ring structure as shown in Fig. 2 for a simulation with yaw misalignment of $\psi = 30^\circ$ and induction factor $a = 0.33$. The figure shows the dense velocity field with the wake deficit calculated from the skeleton of vortex filaments. A kidney-shaped wake appears from the pair of counter-rotating vortices that are generated by a turbine operating under yaw misalignment, as shown in [36].

The model response to control signal variation is verified by examining power production in steady state. First, Fig. 3 shows the 2D and 3D FVW power curve for a variation in axial induction from $a = 0$ to $a = 0.5$. The maximum individual turbine power matches the expectation from momentum theory for the chosen parameters at the theoretical optimum induction of $a = 0.33$. Steady under-induction provides a power gain of 3.6% over greedy control. The 2D and 3D FVW show

remarkably similar behaviour in terms of power production for varying induction factor on the upstream turbine. The similarity in the power estimate shows an opportunity for doing induction control in 2D model studies. Additionally, 2D wind farm flow models have already been used for studies of induction control in a wind farm setting [23,46].

Second, a yaw sweep from $\psi = -45^\circ$ to $\psi = 45^\circ$ is illustrated in Fig. 4. This steady sweep shows a demonstrable lack of power gain from yaw misalignment in the 2D FVW. However, in 3D, yaw misalignment does lead to wake redirection and maximum power achieved in steady-state is 0.313 for a misalignment angle of 34° , providing a gain of 26.1% over greedy control.

The 3D FVW shows the formation of a kidney-shaped wake from a counter-rotating vortex pair when the turbine is operated under yaw misalignment. The subsequent deflection of the turbine wake leads to an increase of the combined power production. These dynamics are not present in 2D, which may explain the lack of wake redirection. This supports previous results that found 2D flow modelling ineffective in capturing the essential effects of wake steering [26].

The model is currently symmetric, which means there is no difference between positive or negative yaw misalignment. Experimental studies have found wake steering to be asymmetric due to the rotation induced by the rotor [28–30]. A normal actuator disc was chosen for simplicity, but a root vortex could be included to model the turbine as a rotating actuator disc to model the asymmetric aspect of wake redirection.

4.2. Induction control in two-dimensional flow

Given the similarity between 2D and 3D in power curves for variation of axial induction, an optimisation case for induction control is set up in 2D with a configuration as in Fig. 1 and parameters as in Table 1. Both turbines are aligned with the inflow wind direction and set to a $\psi = 0^\circ$ yaw angle. The downstream turbine is assumed to be performing at its greedy optimum with an induction factor $a = 0.33$, whereas the induction control signal of the upstream turbine is to be found by solving the optimisation problem.

The objective function (33) is constructed with the control signal $\mathbf{m}_k = [a_k]$, over a prediction horizon of $N_h = 100$ samples. The output contains the power of both turbines, $\mathbf{y}_k = [p_0 \ p_1]^T$. The objective function weights are set to $\mathbf{Q} = [-1 \ -1]$ and $\mathbf{R} = [10]$. This choice of input weight resulted in an adequate balance between input action and power production in an exploratory parameter sweep. At every time step, the optimisation is run for 50 iterations, after which the first value of the control signal from the optimisation is implemented in the receding horizon control scheme. Further iterations have not led to consistently better performance in terms of objective function value given the current optimiser configuration. The starting state for the optimisation case is the result of a steady simulation under greedy operating conditions to remove transient effects. This initial condition is shown in the first frame of Fig. 6.

The control signal produced in this economic MPC framework is illustrated in Fig. 5. The optimisation converges to a roughly periodic excitation with a dominant frequency of approximately $f = 0.20$. The signal features sharp downward peaks where the induction is lowered, thus reducing thrust and allowing more flow to pass through the rotor disc. In addition to the periodic excitation, the mean induction factor is lowered to $\bar{a} = 0.30$ below the greedy optimum of $a = 0.33$. The mean power produced in the final two-thirds of the simulation ($t > 20$) is 0.283, which is an increase of 6.0% over the maximum power achieved with steady induction control.

A series of snapshots of the wake produced with this control signal is shown in Fig. 6. The effects of periodic induction excitation are apparent in the flow field of the wake as coherent structures are formed that travel downstream.

We observe that optimisation with the 2D FVW leads to induction control signals that combine static under-induction with a strong

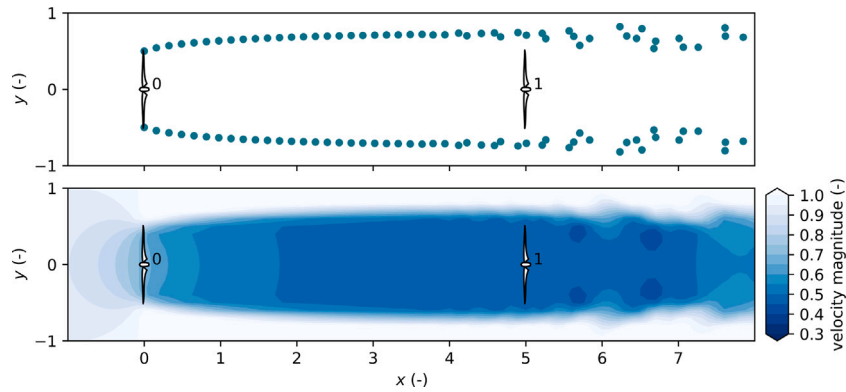


Fig. 1. Illustration of the 2D FVW for uniform unit inflow without yaw misalignment and with a constant induction factor $a = 0.33$. The pairs of vortex points (top) can be used to calculate the velocity at any point, allowing visualisation of a dense velocity field (bottom). The figure illustrates the two-turbine case where the second turbine performance is calculated from the flow velocity 5D downstream.

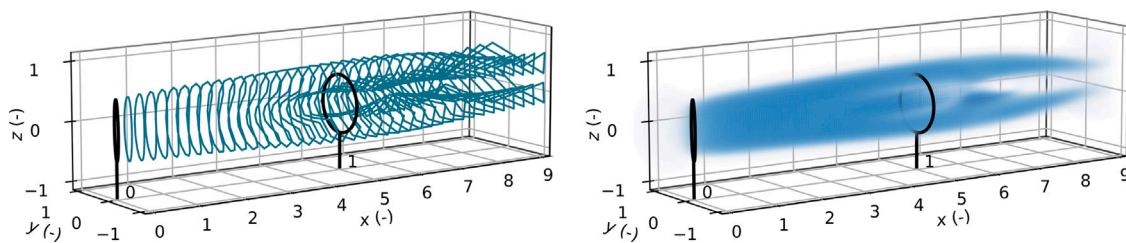


Fig. 2. The 3D FVW models the wake from a series of vortex rings discretised into vortex elements (left), allowing calculation of a velocity field showing the wake deficit (right). The kidney-shaped wake appears as a pair of counter-rotating vortices is formed under yaw misalignment. Simulation under uniform inflow with a yaw misalignment of $\psi = 30^\circ$ and induction $a = 0.33$. The figure illustrates the case where the upstream turbine is modelled with the 3D FVW and the downstream turbine performance is calculated from the flow velocity over a rotor disc 5D downstream.

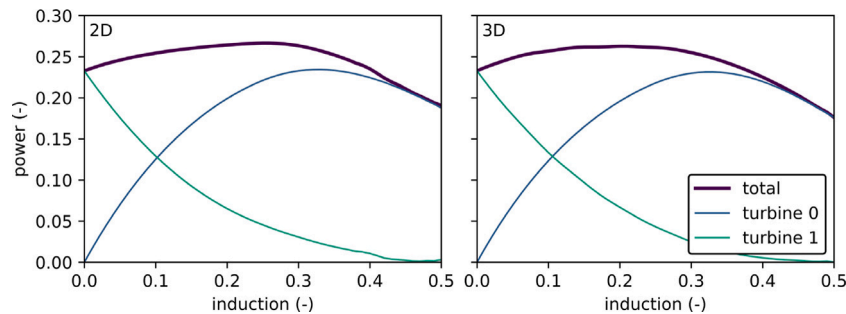


Fig. 3. Power production in steady state for varying induction on the upstream turbine, in 2D (left) and 3D (right), with turbine configuration as in Figs. 1 and 2 respectively. Total power is the sum of power from turbine 0 and turbine 1. Maximum greedy power production occurs for $a = 0.33$. Within this model, lowering the induction on the upstream turbine to $a = 0.27$ leads to a 3.6% gain in total power.

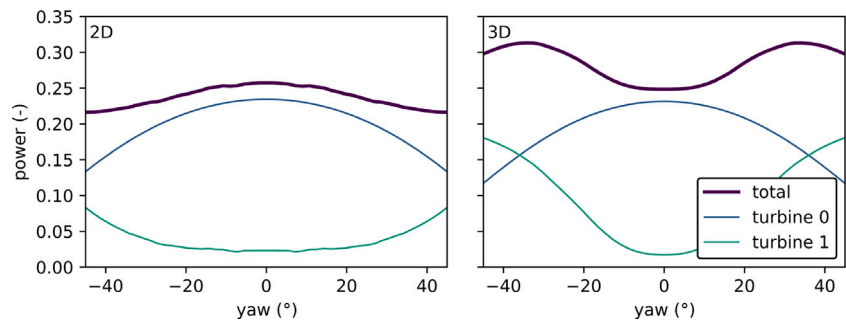


Fig. 4. Power production in steady state for varying yaw on the upstream turbine, in 2D (left) and 3D (right), with turbine configuration as in Figs. 1 and 2 respectively. The downstream turbine power is calculated from the flow velocity 5D downstream from the first turbine. Total power is the sum of power from turbine 0 and turbine 1. The 2D FVW does not have the dynamics to model wake steering effectively. In the 3D FVW, a 26.1% gain in total power is observed for a yaw misalignment angle of $\psi = 34^\circ$ compared to greedy control where $\psi = 0^\circ$.

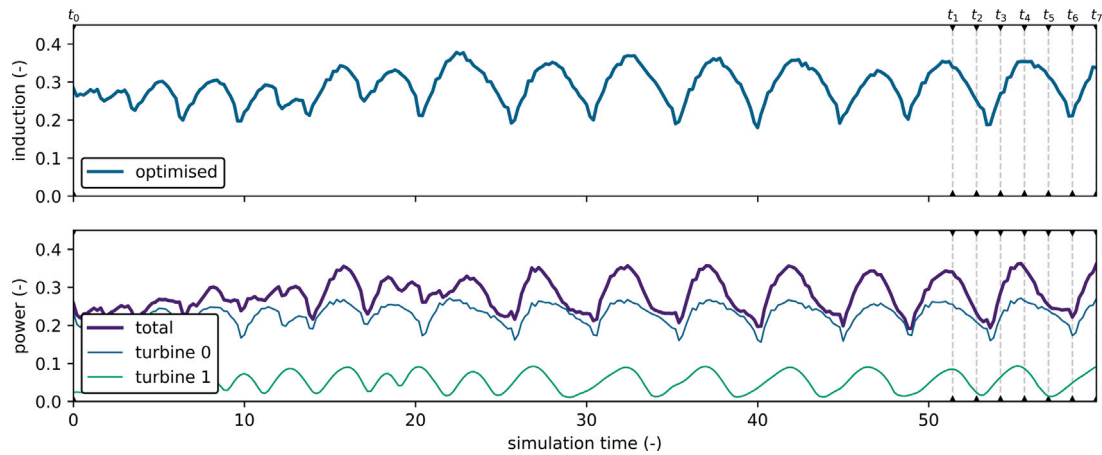


Fig. 5. Receding horizon optimisation of the axial induction control signal for total power yields dynamic behaviour that stimulates wake breakdown and improves time-average power production by 6.0% over steady under-induction. Control signal for turbine 0 is optimised whilst turbine 1 is virtually modelled to be operating at its greedy optimum, positioned 5D downstream in fully waked conditions. Total power is the sum of power from turbine 0 and turbine 1. Snapshots of the flow field at times t_0 to t_7 are illustrated in Fig. 6.

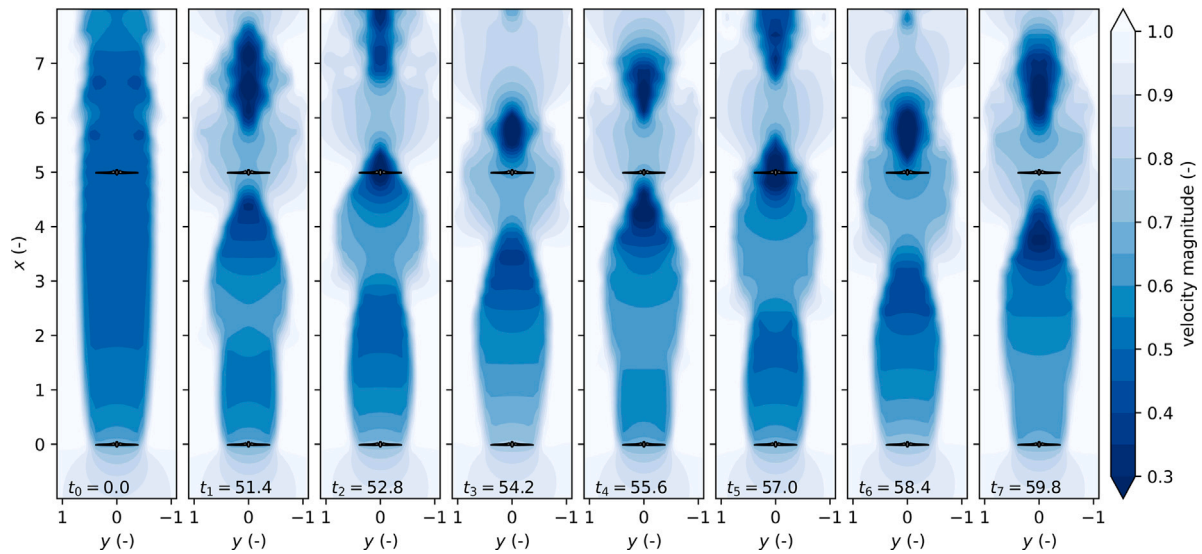


Fig. 6. Snapshots of the flow field from the 2D free-vortex wake simulation under uniform inflow in positive x direction with the induction signal applied to the turbine as shown in Fig. 5. The effects of the periodic induction excitation can be seen in the structure of the wake and appear to enhance wake breakdown.

dynamic component. Within the current model, this combination of periodic excitation and lowering of the mean induction factor outperforms a simple steady induction decrease in terms of mean power production. The sharp downward peaks in induction signal appear to stimulate breakdown of the wake and mixing with the free-stream flow. Note that mixing here does not refer to turbulent mixing as no turbulence is present in the FVW model.

The use of purely static induction control was previously shown not to be a very effective solution for improving wind farm power production [47,48]. These results are supported as, within the 2D FVW, the use of under-induction on its own is less effective than the dynamic induction signal acquired through optimisation. Further study will need to find out whether the combination of slight under-induction and periodic excitation is effective in a realistic wind farm scenario.

The periodic aspect of this control signal resembles the sinusoidal thrust signals that Munters and Meyers [11] found to improve wake mixing by stimulating shedding of vortex rings from the wind turbine. They find sinusoidal actuation at a non-dimensionalised frequency of $f = 0.25$ with a mean local thrust coefficient $c'_t = 2.0$ and amplitude $A = 1.5$ to be optimal for turbines operating in a small farm at 5D spacing. The signal found in the current work has a slightly lower frequency

at $f = 0.20$, and the induction signal corresponds to a lower mean thrust coefficient $c'_t = 1.75$ with amplitude $A = 0.87$. Their optimisation in a 3D LES environment with turbulent inflow is considerably more complex and more expensive than the 2D FVW, which runs well on a regular laptop. It is interesting to note that both studies consider a non-rotating actuator-disc wind turbine model. The differences between the two signals are worth exploring further and will be investigated in future work.

4.3. Yaw control with wind direction variation

Optimisation for yaw control requires the 3D FVW model because it captures the dynamics of the curled wake and therefore shows demonstrable power gain from wake steering, as shown in Fig. 4. The set-up for the optimisation case is as illustrated in Fig. 2 with the parameters listed in Table 1. A smooth wind direction change with unit magnitude from 0° to -20° is implemented to test the capabilities for yaw control under time-varying conditions. The downstream turbine is assumed to be performing at its greedy optimum with an induction factor $a = 0.33$ and a yaw angle that perfectly tracks the inflow direction. The yaw

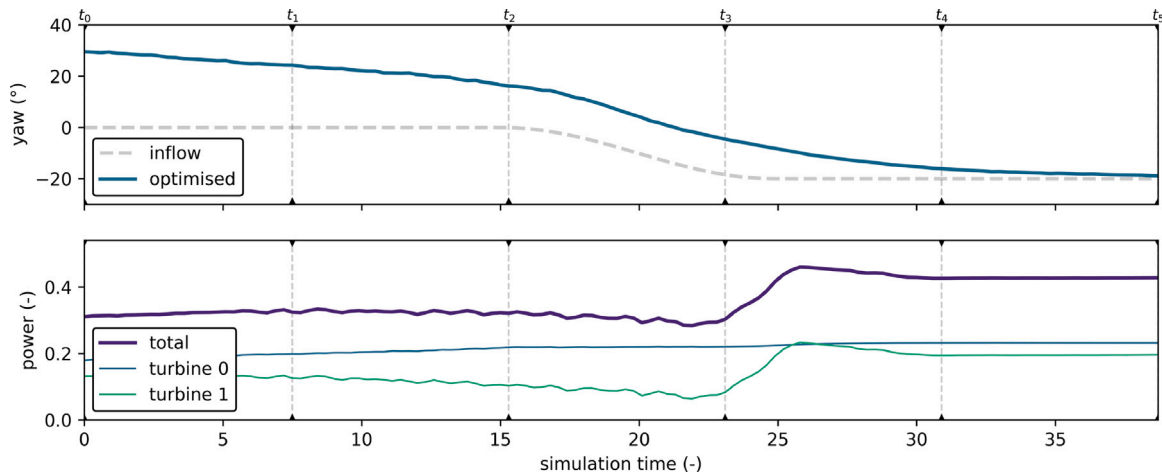


Fig. 7. Optimisation of the yaw control signal for total power finds a solution that utilises yaw misalignment to redirect the wake away from the downstream turbine. As the wind direction changes, the turbine is aligned with the flow to perform at its greedy optimum as the wake no longer impinges on the downstream turbine. Control signal for turbine 0 is optimised whilst turbine 1 is virtually modelled to be operating at its greedy optimum, positioned 5 D downstream in fully waked conditions. Total power is the sum of power from turbine 0 and turbine 1. Snapshots of the flow field at times t_0 to t_5 are illustrated in Fig. 8.

control signal of the upstream turbine is found as the solution of the optimisation problem.

The objective function (33) is constructed with the control signal $\mathbf{m}_k = [\psi_k]$, over a prediction horizon of $N_h = 60$ samples. The output contains the power of both turbines, $\mathbf{y}_k = [p_0 \ p_1]^T$. The objective function weights are set to $\mathbf{Q} = [-1 \ -1]$ and $\mathbf{R} = [0.025]$. This choice of input weight resulted in an adequate balance between input action and power production in an exploratory parameter sweep. It differs from the 2D case because the yaw control signal has a different magnitude than the induction signal. At every time step, the optimisation is run for 10 iterations, after which the first value of the control signal is implemented in the receding horizon control scheme. Given the slower variations in yaw angle compared to induction control, fewer iterations were required before further iterations no longer yielded consistent improvement in objective function with the current optimiser. The starting state for the yaw optimisation case is the result of a steady simulation with a 30° yaw misalignment on the upstream turbine to reach steady conditions with wake redirection. This initial condition is illustrated in the first frame of Fig. 8.

The control signal implemented in the receding horizon control is shown in Fig. 7 together with the inflow and the associated power production for both turbines. The result shows wake redirection through yaw misalignment is maintained for the first section where the wind direction has not yet changed. The turbine is slowly aligned with the inflow in anticipation of the wind direction change. As the wind direction changes, the upstream turbine is rotated into the wind until it is aligned with the free-stream inflow direction. The wake no longer interacts with the downstream turbine which makes its greedy optimum a good control solution.

A series of snapshots of the flow field averaged over rotor height are shown in Fig. 8. The snapshots illustrate how the change in wind direction propagates through the wake. It is visible that the turbine control transitions from wake steering with yaw misalignment to greedy control and alignment with the new wind direction.

The use of the 3D FVW as a novel dynamic surrogate model for online control optimisation contrasts with previous work under time-varying wind directions, where Campagnolo et al. [5] applied pre-optimised set-points in wind tunnel experiments and Doekemeijer et al. [49] used FLORIS to generate steady-state optimal yaw set-points in an online closed-loop controller. Howland et al. [50] presented dynamic yaw control using another, more simplified, physics-based model – a lifting line model with a Gaussian wake. They operated under unsteady inflow, but with an invariant mean wind direction. Similarly, unsteady

flow without direction changes was considered in the model-free yaw control work by Ciri et al. [51].

Especially for yaw control, the initial guess is critical for attaining good results with the optimisation algorithm. The optimiser struggles to find good solutions with an initial guess at zero misalignment from a configuration with greedy yaw control and two turbines with full wake interaction. This seems to be the result of a rather flat optimisation landscape in that configuration. Given some misalignment in the initial guess for the control signal, the optimiser will tend to find a wake steering solution. However, when the turbine is currently misaligned to one side, while the other is more effective, the controller is unlikely to switch because the gradient-based optimisation does not cover a large enough search space. A multi-start optimisation may be a solution to avoid having to predetermine which side to initialise.

Further work is required to validate the effectiveness of the dynamic yaw control results under realistic conditions and investigate whether these solutions improve upon wind farm control strategies with steady surrogate models. Additionally, a combination of yaw-based control and over-induction has been shown to improve wake steering results [52], which could be further explored in optimisation with the 3D FVW.

4.4. Finite horizon effects in economic MPC

In addition to the EMPC results in Sections 4.2 and 4.3, we illustrate the intermediate results that show the effects of optimisation on a finite horizon. Fig. 9 illustrates the optimisation result for a single optimisation window for both the induction control and the yaw control case. It illustrates the executed signal, the result of optimisation with a finite horizon at the current time step, and the actual future control signal that is applied in the receding horizon setting. Towards the finite horizon, the induction returns towards the greedy optimum as the effects no longer reach the downstream turbine. Right at the horizon, a final peak in induction occurs as the optimiser tries to generate a little more power with over-inductive behaviour. The effect of optimisation on a finite horizon for yaw control means that the turbine will rotate back to alignment with the inflow when the wake effects no longer propagate to the downstream turbine within the optimisation window.

For the cases presented in this paper, the horizon length is long enough that these end-of-window effects do not affect the control solution as the receding horizon control scheme prevents these finite

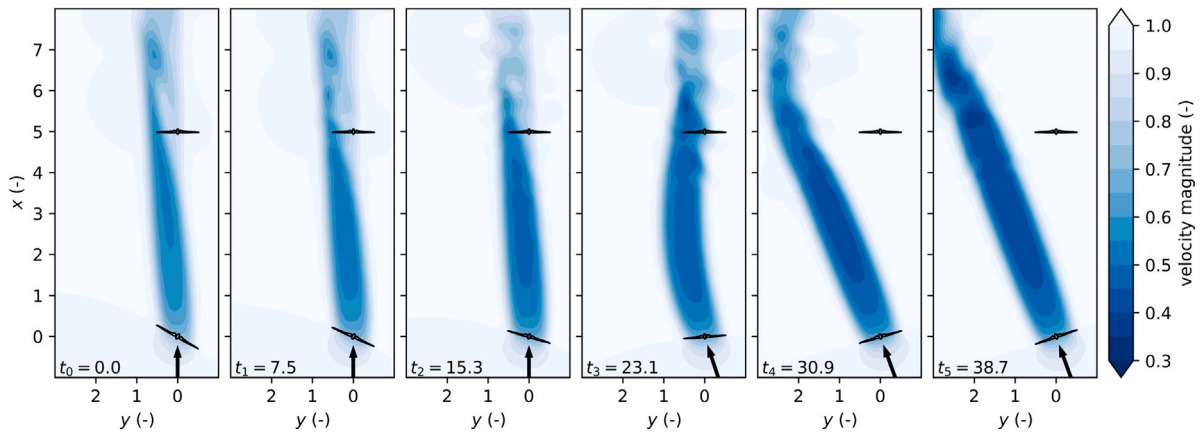


Fig. 8. A series of snapshots of the flow field averaged over rotor height for 3D free-vortex wake simulation under rotating inflow. The black arrow indicates the wind direction and the yaw signal applied to the turbine is shown in Fig. 7. This case illustrates a transition from wake steering to greedy control as the optimal operating point changes with inflow variation.

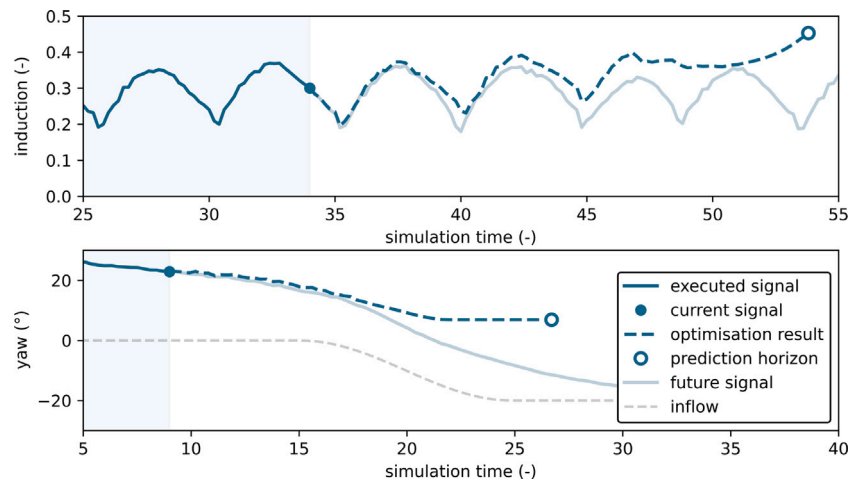


Fig. 9. The control signal for the finite horizon is found through optimisation at every time-step. The executed signal is the past control input leading up to the current state. An optimisation result is found up to the prediction horizon. The first step of the optimised signal is implemented, after which it is shifted and re-used at the next time-step to start re-optimisation. The future signal is the actual control input that ends up being applied in this receding horizon approach.

horizon effects from being implemented. However, especially in optimisation cases where the window is relatively short and input cost is high, the horizon effect needs to be properly treated.

Horizon length for single wind turbine optimisation should be long enough to push turnpike effects away from the signal to be executed. In a multiple-turbine setting, horizon length should be chosen at least long enough such that the effects of control signal variation are observed at the downstream turbine for long enough such that the optimiser finds a balanced control solution. Shorter horizons converge to greedy solutions because the effects do not propagate to downstream turbines within the optimisation window.

5. Conclusion

This work presents a control-oriented free-vortex wake (FVW) model of an actuator disc in 2D and 3D to represent a wind turbine wake. A main novelty in our work is the derivation of the discrete adjoint equations associated with the FVW model, which allows efficient gradient evaluation for use in gradient-based optimisation methods. The FVW model is computationally efficient enough that the experiments in this paper could be run on a regular laptop computer, without requiring high performance computing clusters. The evaluation of the gradient requires only a single forward simulation and backward integration of the adjoint states, which is on the order of ten times

slower than a simple simulation in the current implementation due to the calculation and storage of all partial derivatives.

The FVW model is implemented as a novel surrogate model for gradient-based control optimisation in an economic model-predictive control setting for maximising mean power production by reducing the negative effects of wind turbine wake interaction. This implementation allows generation of optimal control solutions and exploration of dynamical wake behaviour.

In a 2D simulation with receding horizon optimisation, dynamic induction control signals are found that combine slight under-induction with a roughly periodic excitation at a frequency of $f = 0.20$. This results in a 6.0% gain over the maximum power generated with steady induction control. The FVW provides a new, efficient model for exploring dynamic induction control, which has previously been studied in comparatively expensive LES studies.

The 3D free-vortex wake model exhibits curled wake dynamics under yaw misalignment and is therefore suitable as a novel physics-based surrogate model for dynamic wake steering control. The economic model-predictive control strategy finds yaw signals under time-varying wind direction that show both wake redirection and a smooth return to greedy control. It contrasts existing literature that bases yaw control on steady model results or considers dynamic control with unsteady flow, but invariant mean wind directions.

Future work will include validation of the model response, exploration of the dynamics of the optimisation problem and further

experiments with the optimisation of induction control and wake redirection.

Funding

This work is part of the research programme “Robust closed-loop wake steering for large densely spaced wind farms” with project number 17512, which is (partly) financed by the Dutch Research Council (NWO, The Netherlands).

CRedit authorship contribution statement

Maarten J. van den Broek: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Visualization. **Delphine De Tavernier:** Conceptualization, Methodology, Writing – review & editing. **Benjamin Sanderse:** Writing – review & editing, Supervision. **Jan-Willem van Wingerden:** Writing – review & editing, Conceptualization, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The software developed for this study is openly available in figshare with DOI <https://doi.org/10.4121/20278620> or on GitHub at <https://github.com/TUdelft-DataDrivenControl/vortexwake> [53]. Additionally, the data presented in this study is openly available in figshare with DOI <https://doi.org/10.4121/20278590> [54].

Appendix A. Partial derivatives

This section presents the partial derivatives required for the calculation of the gradient with the discrete adjoint. Derivatives of the model state update from Section 2 are provided in Appendix A.1 and those of the output and objective function from Section 3 are provided in Appendix A.2.

A.1. State update

The Jacobian matrix of the state update with respect to the previous state can be written as

$$\frac{\partial f(q_k, m_k)}{\partial q_k} = \frac{\partial q_{k+1}}{\partial q_k} = \frac{\partial}{\partial q_k} \begin{bmatrix} X \\ \Gamma \\ U \\ M \end{bmatrix}_{k+1} = \begin{bmatrix} \frac{\partial X_{k+1}}{\partial X_k} & \frac{\partial X_{k+1}}{\partial \Gamma_k} & \frac{\partial X_{k+1}}{\partial U_k} & \mathbf{0} \\ \frac{\partial \Gamma_{k+1}}{\partial X_k} & \frac{\partial \Gamma_{k+1}}{\partial \Gamma_k} & \frac{\partial \Gamma_{k+1}}{\partial U_k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial U_{k+1}}{\partial U_k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{A.1}$$

where those derivatives that are zero have been removed, leaving a number of sub-matrices to be constructed.

First, take the partial derivatives of the induced velocity of a single vortex element u_i in 2D, which is divided in three parts (u_0 , u_1 and u_2) to simplify calculation,

$$u_i(x_0, x_1) = \underbrace{\begin{bmatrix} -r_y \\ r_x \end{bmatrix}}_{u_0} \underbrace{\left(\frac{\Gamma}{2\pi \|r\|^2} \right)}_{u_1} \underbrace{\left(1 - \exp\left(-\frac{\|r\|^2}{\sigma^2}\right) \right)}_{u_2}. \tag{A.2}$$

For clarity, this section will refer to u_i as u .

The derivative of induced velocity with respect to positions is

$$\frac{\partial u}{\partial x_0} = -\frac{\partial u}{\partial r}, \quad \frac{\partial u}{\partial x_1} = \frac{\partial u}{\partial r}, \tag{A.3}$$

where the derivative to the relative position can be expanded as

$$\frac{\partial u}{\partial r} = \frac{\partial u_0}{\partial r} u_1 u_2 + u_0 \frac{\partial u_1}{\partial r} u_2 + u_0 u_1 \frac{\partial u_2}{\partial r}. \tag{A.4}$$

The required partial derivatives are

$$\frac{\partial u_0}{\partial r} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \tag{A.5}$$

$$\frac{\partial u_1}{\partial r} = \frac{\Gamma}{\pi} \left(\frac{r^T}{\|r\|^4} \right), \tag{A.6}$$

$$\frac{\partial u_2}{\partial r} = \frac{2r^T}{\sigma^2} \exp\left(-\frac{\|r\|^2}{\sigma^2}\right). \tag{A.7}$$

For all rings $b > 0$, the Jacobian matrix from element positions to positions is filled with the partial derivatives as

$$\left[\frac{\partial X^{(b)}}{\partial X^{(a)}} \right]_{ij} = h \frac{\partial u_i^{(b)}}{\partial x_j^{(a)}} \left(x_i^{(b)}, x_j^{(a)} \right) \tag{A.8}$$

for $i, j = 0, 1$, $a = 1, 2, \dots, n_r$ and $b = 1, 2, \dots, n_r - 1$.

If $a + 1 = b$, then the diagonal elements are substituted as

$$\left[\frac{\partial X^{(b)}}{\partial X^{(b-1)}} \right]_{ii} \leftarrow 1 + h \sum_{m=1}^{n_r-1} \sum_{n=0}^1 \frac{\partial u_i^{(b-1)}}{\partial x_j^{(m)}} \left(x_i^{(b-1)}, x_n^{(m)} \right). \tag{A.9}$$

The induced velocity u_i in 3D is similarly divided into three parts u_0 , u_1 , and u_2 ,

$$u_i(x_0, x_1, x_2) = \underbrace{\left(\frac{\Gamma}{4\pi \|r_1 \times r_2\|^2} \right)}_{u_0} \underbrace{\left(r_0 \cdot \left(\frac{r_1}{\|r_1\|} - \frac{r_2}{\|r_2\|} \right) \right)}_{u_1} \underbrace{\left(1 - \exp\left(-\frac{\|r_1 \times r_2\|^2}{\sigma^2 \|r_0\|^2} \right) \right)}_{u_2}. \tag{A.10}$$

The required full derivatives of induced velocity with respect to the position inputs can be expanded as

$$\frac{\partial u}{\partial x_0} = -\frac{\partial u}{\partial r_1} - \frac{\partial u}{\partial r_2}, \tag{A.11}$$

$$\frac{\partial u}{\partial x_1} = \frac{\partial u}{\partial r_1} - \frac{\partial u}{\partial r_0}, \tag{A.12}$$

$$\frac{\partial u}{\partial x_2} = \frac{\partial u}{\partial r_2} + \frac{\partial u}{\partial r_0}. \tag{A.13}$$

We expand the required derivatives according to the product rule into manageable parts,

$$\frac{\partial u}{\partial r_0} = \frac{\partial u_0}{\partial r_0} u_1 u_2 + u_0 \frac{\partial u_1}{\partial r_0} u_2 + u_0 u_1 \frac{\partial u_2}{\partial r_0}, \tag{A.14}$$

$$\frac{\partial u}{\partial r_1} = \frac{\partial u_0}{\partial r_1} u_1 u_2 + u_0 \frac{\partial u_1}{\partial r_1} u_2 + u_0 u_1 \frac{\partial u_2}{\partial r_1}, \tag{A.15}$$

$$\frac{\partial u}{\partial r_2} = \frac{\partial u_0}{\partial r_2} u_1 u_2 + u_0 \frac{\partial u_1}{\partial r_2} u_2 + u_0 u_1 \frac{\partial u_2}{\partial r_2}. \tag{A.16}$$

The partial derivatives of u_0 are

$$\frac{\partial u_0}{\partial r_0} = \mathbf{0}, \tag{A.17}$$

$$\frac{\partial u_0}{\partial r_1} = \frac{\Gamma}{4\pi} \left(\frac{\|r_1 \times r_2\|^2 \mathbf{I}_3 - 2(r_1 \times r_2)(r_1 \times r_2)^T}{\|r_1 \times r_2\|^4} \right) [r_2]_{\times}^T, \tag{A.18}$$

$$\frac{\partial u_0}{\partial r_2} = \frac{\Gamma}{4\pi} \left(\frac{\|r_1 \times r_2\|^2 \mathbf{I}_3 - 2(r_1 \times r_2)(r_1 \times r_2)^T}{\|r_1 \times r_2\|^4} \right) [r_1]_{\times}. \tag{A.19}$$

Here, $[a]_{\times}$ indicates the skew-symmetric constructed from a vector a , such that the cross product can be written in the form

$$a \times b = [a]_{\times} b = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, \tag{A.20}$$

allowing compact notation of the derivative

$$\frac{\partial \mathbf{a} \times \mathbf{b}}{\partial \mathbf{b}} = [\mathbf{a}]_{\times}. \quad (\text{A.21})$$

The partial derivatives of u_1 are

$$\frac{\partial u_1}{\partial \mathbf{r}_0} = \left(\frac{\mathbf{r}_1}{\|\mathbf{r}_1\|} - \frac{\mathbf{r}_2}{\|\mathbf{r}_2\|} \right)^T, \quad (\text{A.22})$$

$$\frac{\partial u_1}{\partial \mathbf{r}_1} = \frac{\|\mathbf{r}_1\|^2 \mathbf{r}_0^T - (\mathbf{r}_0 \cdot \mathbf{r}_1) \mathbf{r}_1^T}{\|\mathbf{r}_1\|^3}, \quad (\text{A.23})$$

$$\frac{\partial u_1}{\partial \mathbf{r}_2} = -\frac{\|\mathbf{r}_2\|^2 \mathbf{r}_0^T - (\mathbf{r}_0 \cdot \mathbf{r}_2) \mathbf{r}_2^T}{\|\mathbf{r}_2\|^3}. \quad (\text{A.24})$$

Finally, the partial derivatives of u_2 are

$$\frac{\partial u_2}{\partial \mathbf{r}_0} = -\exp\left(-\frac{\|\mathbf{r}_1 \times \mathbf{r}_2\|^2}{\sigma^2 \|\mathbf{r}_0\|^2}\right) \left(2 \frac{\|\mathbf{r}_1 \times \mathbf{r}_2\|^2}{\sigma^2 \|\mathbf{r}_0\|^4}\right) \mathbf{r}_0^T, \quad (\text{A.25})$$

$$\frac{\partial u_2}{\partial \mathbf{r}_1} = \exp\left(-\frac{\|\mathbf{r}_1 \times \mathbf{r}_2\|^2}{\sigma^2 \|\mathbf{r}_0\|^2}\right) \left(\frac{2(\mathbf{r}_1 \times \mathbf{r}_2)^T}{\sigma^2 \|\mathbf{r}_0\|^2}\right) [\mathbf{r}_2]_{\times}^T, \quad (\text{A.26})$$

$$\frac{\partial u_2}{\partial \mathbf{r}_2} = \exp\left(-\frac{\|\mathbf{r}_1 \times \mathbf{r}_2\|^2}{\sigma^2 \|\mathbf{r}_0\|^2}\right) \left(\frac{2(\mathbf{r}_1 \times \mathbf{r}_2)^T}{\sigma^2 \|\mathbf{r}_0\|^2}\right) [\mathbf{r}_1]_{\times}^T. \quad (\text{A.27})$$

For all rings $b > 0$, the Jacobian matrix is filled with the partial derivatives as

$$\left[\frac{\partial \mathbf{X}^{(b)}}{\partial \mathbf{X}^{(a)}} \right]_{ij} = \begin{cases} h \frac{\partial u_i^{(b)}}{\partial x_j^{(a)}}(\mathbf{x}_i^{(b)}, \mathbf{x}_j^{(a)}, \mathbf{x}_{j+1}^{(a)}) & \text{if } j = 0, \\ h \frac{\partial u_i^{(b)}}{\partial x_j^{(a)}}(\mathbf{x}_i^{(b)}, \mathbf{x}_{j-1}^{(a)}, \mathbf{x}_j^{(a)}) & \text{if } j = n_c, \\ h \frac{\partial u_i^{(b)}}{\partial x_j^{(a)}}(\mathbf{x}_i^{(b)}, \mathbf{x}_{j-1}^{(a)}, \mathbf{x}_j^{(a)}) + h \frac{\partial u_i^{(b)}}{\partial x_j^{(a)}}(\mathbf{x}_i^{(b)}, \mathbf{x}_j^{(a)}, \mathbf{x}_{j+1}^{(a)}) & \text{otherwise,} \end{cases} \quad (\text{A.28})$$

for $i, j = 0, 1, \dots, n_c$, $a = 0, 1, \dots, n_r - 1$ and $b = 1, 2, \dots, n_r - 1$.

If $a + 1 = b$, then the diagonal elements are substituted as

$$\left[\frac{\partial \mathbf{X}^{(b)}}{\partial \mathbf{X}^{(a)}} \right]_{ii} \leftarrow 1 + h \sum_{m=1}^{n_r-1} \sum_{n=1}^{n_c} \frac{\partial u_i^{(b-1)}}{\partial x_j^{(b-1)}}(\mathbf{x}_i^{(b-1)}, \mathbf{x}_{n-1}^{(m)}, \mathbf{x}_n^{(m)}). \quad (\text{A.29})$$

The position of the first ring depends only on the yaw angle, therefore

$$\frac{\partial \mathbf{X}^{(0)}}{\partial \mathbf{X}^{(a)}} = \mathbf{0}. \quad (\text{A.30})$$

Further derivatives are mostly independent from the dimension of the FVW model used.

The position update for all rings $b > 0$ depends on the vortex strength of all elements as

$$\left[\frac{\partial \mathbf{X}^{(b)}}{\partial \Gamma^{(a)}} \right]_{ij} = \frac{\partial x_i^{(b)}}{\partial u_i^{(b-1)}} \frac{\partial u_i^{(b-1)}}{\partial \Gamma_n^{(a)}} = h \frac{\partial u_i^{(b-1)}}{\partial \Gamma_j^{(a)}}(\mathbf{x}_i^{(b-1)}, \mathbf{x}_{j-1}^{(a)}, \mathbf{x}_j^{(a)}) \quad (\text{A.31})$$

for $i = 0, 1, \dots, n_c$, $j = 1, 2, \dots, n_c$ and $a = 0, 1, \dots, n_r - 1$,

where the partial derivative of induced velocity to vortex strength is

$$\frac{\partial u_i}{\partial \Gamma_j} = \frac{1}{\Gamma_j} \mathbf{u}_i. \quad (\text{A.32})$$

The points defining the vortex filament travel based on their stored free-stream velocity, so for all rings $b > 0$,

$$\frac{\partial \mathbf{X}_{k+1}^{(b)}}{\partial \mathbf{U}_k^{(b-1)}} = h \mathbf{I}_{n_p}. \quad (\text{A.33})$$

Vortex strength and free-stream velocity are passed from one ring to the next for all rings $b > 0$,

$$\frac{\partial \Gamma_{k+1}^{(b)}}{\partial \Gamma_k^{(b-1)}} = \mathbf{I}_{n_c}, \quad (\text{A.34})$$

$$\frac{\partial \mathbf{U}_{k+1}^{(b)}}{\partial \mathbf{U}_k^{(b-1)}} = \mathbf{I}_{n_p}. \quad (\text{A.35})$$

The initialisation of a new ring depends on the entire state of the model through the rotor velocity,

$$\frac{\partial \Gamma_{k+1}^{(0)}}{\partial \mathbf{X}_k} = \frac{\partial \Gamma^{(0)}}{\partial \mathbf{u}_r} \frac{\partial \mathbf{u}_r}{\partial \mathbf{X}}, \quad (\text{A.36})$$

$$\frac{\partial \Gamma_{k+1}^{(0)}}{\partial \Gamma_k} = \frac{\partial \Gamma^{(0)}}{\partial \mathbf{u}_r} \frac{\partial \mathbf{u}_r}{\partial \Gamma}, \quad (\text{A.37})$$

$$\frac{\partial \Gamma_{k+1}^{(0)}}{\partial \mathbf{U}_k} = \frac{\partial \Gamma^{(0)}}{\partial \mathbf{u}_r} \frac{\partial \mathbf{u}_r}{\partial \mathbf{U}}, \quad (\text{A.38})$$

where

$$\frac{\partial \Gamma}{\partial \mathbf{u}_r} = h c_i'(a) \mathbf{n}(\psi) (\mathbf{u}_r^T \mathbf{n}(\psi)). \quad (\text{A.39})$$

The Jacobian of rotor speed with respect to all element positions is calculated as

$$\frac{\partial \mathbf{u}_r}{\partial \mathbf{X}_k} = \frac{1}{n_u} \sum_{i=0}^{n_u-1} \left(\frac{\partial}{\partial \mathbf{X}_k} (\mathbf{u}_{\infty}(\mathbf{x}_i, \mathbf{q})) + \frac{\partial}{\partial \mathbf{X}_k} (\mathbf{u}_{\text{ind}}(\mathbf{x}_i, \mathbf{q})) \right), \quad (\text{A.40})$$

with

$$\frac{\partial \mathbf{u}_{\infty}(\mathbf{x}_i, \mathbf{q})}{\partial \mathbf{X}} = 20(\mathbf{x} - \mathbf{x}_i^{(b)}) \bar{w}_i^{(b)} - \bar{w}_i^{(b)} \sum_{i=0}^{n_u-1} \sum_{b=0}^{n_r} 20(\mathbf{x} - \mathbf{x}_i^{(b)}) \bar{w}_i^{(b)}, \quad (\text{A.41})$$

with the partial derivatives placed in 2D as

$$\left[\frac{\partial \mathbf{u}_{\text{ind}}(\mathbf{x}_i, \mathbf{q})}{\partial \mathbf{X}^{(a)}} \right]_{0j} = \frac{\partial \mathbf{u}}{\partial \mathbf{x}_j^{(a)}}(\mathbf{x}_i, \mathbf{x}_j^{(a)}) \quad (\text{A.42})$$

for $j = 0, 1, a = 0, 1, \dots, n_r - 1$,

and for the 3D FVW as

$$\left[\frac{\partial \mathbf{u}_{\text{ind}}(\mathbf{x}_i, \mathbf{q})}{\partial \mathbf{X}^{(a)}} \right]_{0j} = \begin{cases} \frac{\partial \mathbf{u}}{\partial \mathbf{x}_j^{(a)}}(\mathbf{x}_i, \mathbf{x}_j^{(a)}, \mathbf{x}_{j+1}^{(a)}) & \text{if } j = 0, \\ \frac{\partial \mathbf{u}}{\partial \mathbf{x}_j^{(a)}}(\mathbf{x}_i, \mathbf{x}_{j-1}^{(a)}, \mathbf{x}_j^{(a)}) & \text{if } j = n_c, \\ \frac{\partial \mathbf{u}}{\partial \mathbf{x}_j^{(a)}}(\mathbf{x}_i, \mathbf{x}_{j-1}^{(a)}, \mathbf{x}_j^{(a)}) + \frac{\partial \mathbf{u}}{\partial \mathbf{x}_j^{(a)}}(\mathbf{x}_i, \mathbf{x}_j^{(a)}, \mathbf{x}_{j+1}^{(a)}) & \text{otherwise,} \end{cases} \quad (\text{A.43})$$

for $j = 0, 1, \dots, n_c$, $a = 0, 1, \dots, n_r - 1$.

The Jacobian with respect to all vortex strengths is calculated as

$$\frac{\partial \mathbf{u}_r}{\partial \Gamma_k} = \frac{1}{n_u} \sum_{i=0}^{n_u-1} \frac{\partial}{\partial \Gamma_k} (\mathbf{u}_{\text{ind}}(\mathbf{x}_i, \mathbf{q})), \quad (\text{A.44})$$

with the partial derivatives

$$\left[\frac{\partial \mathbf{u}_{\text{ind}}(\mathbf{x}_i, \mathbf{q})}{\partial \Gamma_k^{(a)}} \right]_{1j} = \frac{\partial \mathbf{u}}{\partial \Gamma_j^{(a)}}(\mathbf{x}_i, \mathbf{x}_{j-1}^{(a)}, \mathbf{x}_j^{(a)}) \quad (\text{A.45})$$

for $j = 1, 2, \dots, n_c$ and $a = 0, 1, \dots, n_r - 1$.

The derivative of the free-stream contribution to the disc-averaged velocity over the rotor is

$$\frac{\partial \mathbf{u}_r}{\partial \mathbf{u}_{\infty, i}^{(b)}} = \frac{1}{n_u} \sum_{i=0}^{n_u-1} \bar{w}_i^{(b)}(\mathbf{x}_i, \mathbf{q}). \quad (\text{A.46})$$

The Jacobian matrix of the state update with respect to the inputs is defined as

$$\frac{\partial \mathbf{q}_{k+1}}{\partial \mathbf{m}_k} = \frac{\partial}{\partial \mathbf{m}_k} \begin{bmatrix} \mathbf{X} \\ \Gamma \\ \mathbf{U} \\ \mathbf{M} \end{bmatrix}_{k+1} = \begin{bmatrix} \frac{\partial \mathbf{X}_{k+1}}{\partial \mathbf{m}_k} \\ \frac{\partial \Gamma_{k+1}}{\partial \mathbf{m}_k} \\ \mathbf{0} \\ \frac{\partial \mathbf{M}_{k+1}}{\partial \mathbf{m}_k} \end{bmatrix}. \quad (\text{A.47})$$

The position of the first ring is directly controlled with the yaw angle

$$\left[\frac{\partial \mathbf{X}^{(0)}}{\partial \psi} \right]_i = \frac{\partial \mathbf{x}_i^{(0)}}{\partial \psi} \quad \text{for } i = 0, 1, \dots, n_c, \quad (\text{A.48})$$

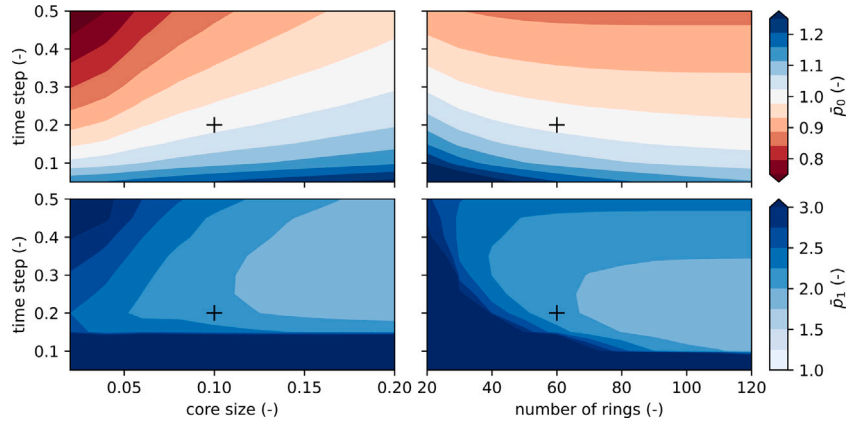


Fig. B.10. Variation of power for 2D FVW normalised by power estimate from momentum theory, for upstream (top) and downstream (bottom) turbine in a configuration as in Fig. 1. The cross indicates the parameter choice used in this paper. The base settings for this parameter sweep are: axial induction $a_0 = 0.3$ and $a_1 = 0.33$, number of rings $n_r = 60$, time step $h = 0.2$, and core size $\sigma = 0.1$.

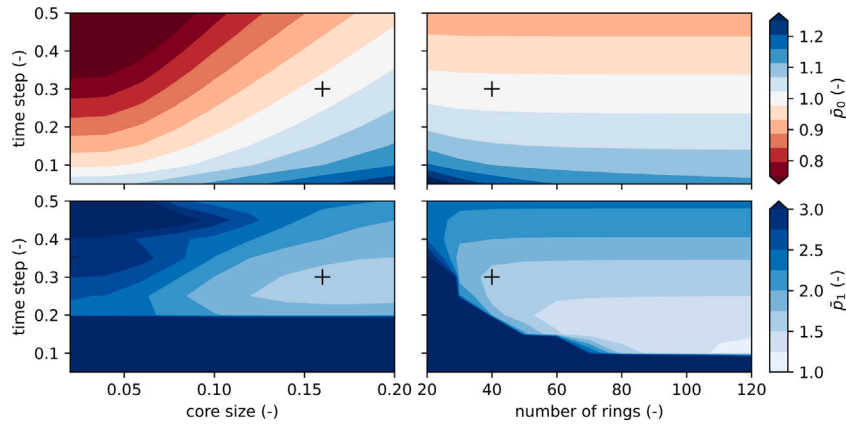


Fig. B.11. Variation of power for 3D FVW normalised by power estimate from momentum theory, for upstream (top) and downstream (bottom) turbine in a configuration as in Fig. 2. The cross indicates parameter choice used in this paper. The base settings for this parameter sweep are: axial induction $a_0 = 0.3$ and $a_1 = 0.33$, number of rings $n_r = 40$, number of elements per ring $n_e = 16$, time step $h = 0.3$, and core size $\sigma = 0.16$.

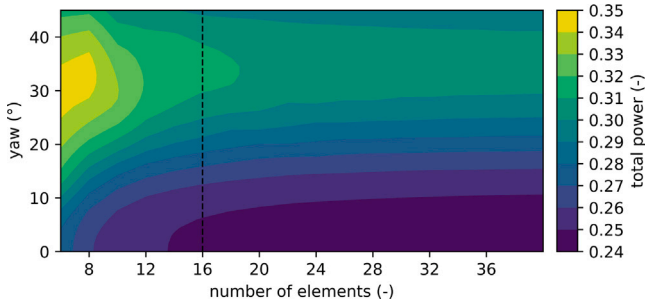


Fig. B.12. Total power estimate with the 3D FVW under yaw misalignment for varying number of vortex filaments in vortex ring discretisation. The dashed line indicates the number of elements used in this paper. Configuration as in Fig. 2.

which in 2D yields

$$\frac{\partial \mathbf{x}_0^{(0)}}{\partial \psi} = \frac{\partial \mathbf{R}_z(\psi_k)}{\partial \psi} \begin{bmatrix} 0 \\ r \end{bmatrix}, \quad \frac{\partial \mathbf{x}_1^{(0)}}{\partial \psi} = \frac{\partial \mathbf{R}_z(\psi_k)}{\partial \psi}(\psi_k) \begin{bmatrix} 0 \\ -r \end{bmatrix}, \quad (\text{A.49})$$

and in 3D yields

$$\frac{\partial \mathbf{x}_i^{(0)}(\psi)}{\partial \psi} = \frac{\partial \mathbf{R}_z(\psi_k)}{\partial \psi} \begin{bmatrix} 0 \\ r \cos(2\pi \frac{i}{n_e}) \\ r \sin(2\pi \frac{i}{n_e}) \end{bmatrix} \quad \text{for } i = 0, 1, \dots, n_e, \quad (\text{A.50})$$

given the derivative of the rotation matrix in the relevant dimension,

$$\frac{\partial \mathbf{R}_z(\psi)}{\partial \psi} = \begin{bmatrix} -\sin \psi & \cos \psi & 0 \\ -\cos \psi & -\sin \psi & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \frac{\partial \mathbf{R}_z(\psi)}{\partial \psi} = \begin{bmatrix} -\sin \psi & \cos \psi & 0 \\ -\cos \psi & -\sin \psi & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (\text{A.51})$$

The vortex strength of the first ring depends on the yaw angle as

$$\left[\frac{\partial \Gamma_i^{(0)}}{\partial \psi} \right]_i = \frac{\partial \Gamma_i^{(0)}}{\partial \psi}(a, \psi) \quad \text{for } i = 1, 2, \dots, n_e, \quad (\text{A.52})$$

where

$$\frac{\partial \Gamma_i^{(0)}}{\partial \psi} = hc'_t(a)(\mathbf{u}_r^T \mathbf{n}(\psi)) \left(\mathbf{u}_r^T \frac{\partial \mathbf{n}(\psi)}{\partial \psi} \right). \quad (\text{A.53})$$

The vortex strength of the first ring depends on the thrust coefficient as

$$\left[\frac{\partial \Gamma_i^{(0)}}{\partial a} \right]_i = \frac{\partial \Gamma_i^{(0)}}{\partial a}(a, \psi) \quad \text{for } i = 1, 2, \dots, n_e, \quad (\text{A.54})$$

where

$$\frac{\partial \Gamma_i^{(0)}}{\partial a} = \left(h \frac{1}{2} (\mathbf{u}_r \cdot \mathbf{n}(\psi))^2 \right) \frac{\partial c'_t}{\partial a}, \quad (\text{A.55})$$

$$\frac{\partial c'_t}{\partial a} = \begin{cases} \frac{4}{(1-a)^2} & \text{if } a \leq a_t, \\ \frac{2c_{t1}}{(1-a)^3} - \frac{4(\sqrt{c_{t1}-1})}{(1-a)^2} & \text{if } a > a_t. \end{cases} \quad (\text{A.56})$$

The derivative of saved controls is identity,

$$\frac{\partial \mathbf{M}_{k+1}}{\partial \mathbf{m}_k} = \mathbf{I}_{n_c}. \quad (\text{A.57})$$

A.2. Output and objective function

The input sensitivity of the objective function is calculated as

$$\frac{\partial J_k}{\partial \mathbf{q}_k} = \mathbf{Q} \frac{\partial \mathbf{y}_k}{\partial \mathbf{q}_k} - 2\Delta \mathbf{m}_k^T \mathbf{R}, \quad (\text{A.58})$$

$$\frac{\partial J_k}{\partial \mathbf{m}_k} = \mathbf{Q} \frac{\partial \mathbf{y}_k}{\partial \mathbf{m}_k} + 2\Delta \mathbf{m}_k^T \mathbf{R}. \quad (\text{A.59})$$

The partial derivative of the output to the controls is $\partial \mathbf{y}_k / \partial \mathbf{m}_k = 0$ because the power is calculated with controls saved in the state vector \mathbf{q}_k . The Jacobian of the output to the states is given as

$$\frac{\partial \mathbf{y}_k}{\partial \mathbf{q}_k} = \begin{bmatrix} \frac{\partial \mathbf{y}_k}{\partial \mathbf{u}_r} \\ \frac{\partial \mathbf{y}_k}{\partial X_k} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{u}_r}{\partial \Gamma_k} & \frac{\partial \mathbf{u}_r}{\partial U_k} & \frac{\partial \mathbf{u}_r}{\partial M_k} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{y}_k}{\partial \mathbf{M}_k} \end{bmatrix}.$$

The Jacobian of the power output to the saved control inputs for the two-turbine case presented in this paper is constructed as

$$\frac{\partial \mathbf{y}_k}{\partial \mathbf{M}_k} = \begin{bmatrix} \frac{\partial P_0}{\partial a_0} & \frac{\partial P_0}{\partial \psi_0} & 0 & 0 \\ 0 & 0 & \frac{\partial P_1}{\partial a_1} & \frac{\partial P_1}{\partial \psi_1} \end{bmatrix}, \quad (\text{A.60})$$

with the partial derivatives to the axial induction

$$\frac{\partial P_0}{\partial a_0} = \frac{1}{2} A_r (\mathbf{u}_{r_0} \cdot \mathbf{n}(\psi_0))^3 \frac{4}{(1-a_0)^2}, \quad (\text{A.61})$$

$$\frac{\partial P_1}{\partial a_1} = \frac{1}{2} A_r (\mathbf{u}_{r_1} \cdot \mathbf{n}(\psi_1))^3 (-8a_1(1-a_1) + 4(1-a_1)^2), \quad (\text{A.62})$$

and the partial derivatives to the yaw angle

$$\frac{\partial P_0}{\partial \psi_0} = \frac{3}{2} c_p'(a_0) A_r (\mathbf{u}_{r_0} \cdot \mathbf{n}(\psi_0))^2 (\mathbf{u}_{r_0}^T \frac{\partial \mathbf{n}(\psi_0)}{\partial \psi_0}), \quad (\text{A.63})$$

$$\frac{\partial P_1}{\partial \psi_1} = \frac{3}{2} c_p'(a_1) A_r (\mathbf{u}_{r_1}^* \cdot \mathbf{n}(\psi_1))^2 (\mathbf{u}_{r_1}^{*T} \frac{\partial \mathbf{n}(\psi_1)}{\partial \psi_1}). \quad (\text{A.64})$$

Taking the derivative of the output to the rotor velocity yields

$$\begin{bmatrix} \frac{\partial \mathbf{y}_k}{\partial \mathbf{u}_r} \\ \frac{\partial \mathbf{y}_k}{\partial \mathbf{u}_{r_0}} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_0}{\partial \mathbf{u}_{r_0}} & 0 \\ 0 & \frac{\partial P_1}{\partial \mathbf{u}_{r_1}} \end{bmatrix}, \quad (\text{A.65})$$

$$\frac{\partial P_0}{\partial \mathbf{u}_{r_0}} = \frac{3}{2} c_p'(a_0) A_r (\mathbf{u}_{r_0} \cdot \mathbf{n}(\psi_0))^2 \mathbf{n}(\psi_0)^T, \quad (\text{A.66})$$

$$\frac{\partial P_1}{\partial \mathbf{u}_{r_1}} = \frac{3}{2} c_p'(a_1) A_r (\mathbf{u}_{r_1} \cdot \mathbf{n}(\psi_1))^2 \mathbf{n}(\psi_1)^T (1-a_1). \quad (\text{A.67})$$

Appendix B. Convergence study for numerical methods

A variation of model parameters for the numerical methods was performed to show convergence for the model configuration chosen in this paper as presented in Section 4.1. The power output under steady conditions is compared to the reference power from momentum theory for validation, calculated as

$$p_{0,\text{ref}} = c_p'(a_0) A_r (u_\infty (1-a_0))^3, \quad (\text{B.1})$$

$$p_{1,\text{ref}} = c_p'(a_1) A_r (u_\infty (1-2a_0)(1-a_1))^3, \quad (\text{B.2})$$

and normalised as $\bar{p} = p/p_{\text{ref}}$. Turbine configuration is chosen with a 5D spacing as used in this paper and illustrated in Fig. 1.

The normalised power for a parameter sweep around the operating point from Table 1 is shown in Fig. B.10, evaluating combinations of time step h , vortex core size σ , and number of rings n_r . The chosen parameters appear to be a good match for the normalised power of the upstream turbine, \bar{p}_0 . There is a dependency between time step and core size, so they need to be chosen together. The combination of number of rings to model the wake needs to be chosen together with the time-step such that the wake covers the downstream turbine. As long as the wake adequately covers the downstream turbine position, the power

estimate is not very sensitive to variations in the parameters observed here. The 2D FVW does not exactly represent momentum theory, so it is unsurprising that downstream power is overestimated.

A similar sweep with the 3D FVW is shown in Fig. B.11. The trends in the results are similar to those shown in the parameter variations for the 2D FVW. For this configuration without yaw misalignment, results appeared not very sensitive to the number of elements used in the discretisation of the vortex rings.

The number of elements in ring discretisation is important for dynamics of wake redirection under yaw misalignment because finer discretisations better represent the curled wake. Fig. B.12 shows that power gain from yaw misalignment is observed even for coarse discretisations of the vortex filament rings. In the trade-off between computational limits and better convergence of the dynamics, a number of elements $n_e = 16$ was chosen for use in this paper.

References

- [1] J.W. van Wingerden, P.A. Fleming, T. Göcmen, I. Eguinoa, B.M. Doekemeijer, K. Dykes, M. Lawson, E. Simley, J. King, D. Astrain, M. Iribas, C.L. Bottasso, J. Meyers, S. Raach, K. Kölle, G. Giebel, Expert Elicitation on Wind Farm Control, *J. Phys. Conf. Ser.* 1618 (2) (2020) <http://dx.doi.org/10.1088/1742-6596/1618/2/022025>.
- [2] J. Meyers, C. Bottasso, K. Dykes, P. Fleming, P. Gebraad, G. Giebel, T. Göcmen, J.W. van Wingerden, Wind farm flow control: prospects and challenges, *Wind Energy Sci. Discuss. (March)* (2022) 1–55, <http://dx.doi.org/10.5194/wes-2022-24>.
- [3] F. Campagnolo, V. Petrović, J. Schreiber, E.M. Nanos, A. Croce, C.L. Bottasso, Wind tunnel testing of a closed-loop wake deflection controller for wind farm power maximization, *J. Phys. Conf. Ser.* 753 (3) (2016) <http://dx.doi.org/10.1088/1742-6596/753/3/032006>.
- [4] M. Bastankhah, F. Porté-Agel, Wind farm power optimization via yaw angle control: A wind tunnel study, *J. Renew. Sustain. Energy* 11 (2) (2019) <http://dx.doi.org/10.1063/1.5077038>.
- [5] F. Campagnolo, R. Weber, J. Schreiber, C.L. Bottasso, Wind tunnel testing of wake steering with dynamic wind direction changes, *Wind Energy Sci.* 5 (4) (2020) 1273–1295, <http://dx.doi.org/10.5194/wes-5-1273-2020>.
- [6] M.F. Howland, S.K. Lele, J.O. Dabiri, Wind farm power optimization through wake steering, *Proc. Natl. Acad. Sci. USA* 116 (29) (2019) 14495–14500, <http://dx.doi.org/10.1073/pnas.1903680116>.
- [7] P. Fleming, J. King, E. Simley, J. Roadman, A. Scholbrock, P. Murphy, J. K. Lundquist, P. Moriarty, K. Fleming, J. Van Dam, C. Bay, R. Mudafort, D. Jager, J. Skopek, M. Scott, B. Ryan, C. Guernsey, D. Brake, Continued results from a field campaign of wake steering applied at a commercial wind farm - Part 2, *Wind Energy Sci.* 5 (3) (2020) 945–958, <http://dx.doi.org/10.5194/wes-5-945-2020>.
- [8] P. Fleming, M. Sinner, T. Young, M. Lannic, J. King, E. Simley, B.M. Doekemeijer, Experimental results of wake steering using fixed angles, *Wind Energy Sci.* 6 (6) (2021) 1521–1531, <http://dx.doi.org/10.5194/wes-6-1521-2021>.
- [9] B.M. Doekemeijer, S. Kern, S. Maturu, S. Kanev, B. Salbert, J. Schreiber, F. Campagnolo, C.L. Bottasso, S. Schuler, F. Wilts, T. Neumann, G. Potenza, F. Calabretta, F. Fioretti, J.W. van Wingerden, Field experiment for open-loop yaw-based wake steering at a commercial onshore wind farm in Italy, *Wind Energy Sci.* 6 (1) (2021) 159–176, <http://dx.doi.org/10.5194/wes-6-159-2021>.
- [10] E. Simley, P. Fleming, N. Girard, L. Alloin, E. Godefroy, T. Duc, Results from a wake-steering experiment at a commercial wind plant: Investigating the wind speed dependence of wake-steering performance, *Wind Energy Sci.* 6 (6) (2021) 1427–1453, <http://dx.doi.org/10.5194/wes-6-1427-2021>.
- [11] W. Munters, J. Meyers, Towards practical dynamic induction control of wind farms: Analysis of optimally controlled wind-farm boundary layers and sinusoidal induction control of first-row turbines, *Wind Energy Sci.* 3 (1) (2018) 409–425, <http://dx.doi.org/10.5194/wes-3-409-2018>.
- [12] J.A. Frederik, R. Weber, S. Cacciola, F. Campagnolo, A. Croce, C. Bottasso, J.W. van Wingerden, Periodic dynamic induction control of wind farms: Proving the potential in simulations and wind tunnel experiments, *Wind Energy Sci.* 5 (1) (2020) 245–257, <http://dx.doi.org/10.5194/wes-5-245-2020>.
- [13] J.A. Frederik, B.M. Doekemeijer, S.P. Mulders, J.W. van Wingerden, The helix approach: Using dynamic individual pitch control to enhance wake mixing in wind farms, *Wind Energy* 23 (8) (2020) 1739–1751, <http://dx.doi.org/10.1002/we.2513>.
- [14] NREL, FLORIS. Version 3.0, 2022, Available at: <https://github.com/NREL/floris>.
- [15] M. Bastankhah, F. Porté-Agel, A new analytical model for wind-turbine wakes, *Renew. Energy* 70 (2014) 116–123, <http://dx.doi.org/10.1016/j.renene.2014.01.002>.
- [16] L.A. Martínez-Tossas, J. Annoni, P.A. Fleming, M.J. Churchfield, The aerodynamics of the curled wake: a simplified model in view of flow control, *Wind Energy Sci.* 4 (1) (2019) 127–138, <http://dx.doi.org/10.5194/wes-4-127-2019>.

- [17] G.C. Larsen, H. Madsen Aagaard, F. Bingöl, J. Mann, S.r. Ott, J.N. Sørensen, V. Okulov, N. Trolborg, N.M. Nielsen, K. Thomsen, T.J. Larsen, R. Mikkelsen, *Dynamic wake meandering modeling*, 2007.
- [18] M. Becker, B. Ritter, B.M. Doekemeijer, D. van der Hoek, U. Konigorski, D. Allaerts, J.-W. van Wingerden, The revised FLORIDyn model: Implementation of heterogeneous flow and the Gaussian wake, *Wind Energy Sci. Discuss.* (January) (2022) 1–25, <http://dx.doi.org/10.5194/wes-2021-154>.
- [19] J.P. Goit, J. Meyers, Optimal control of energy extraction in wind-farm boundary layers, *J. Fluid Mech.* 768 (2015) 5–50, <http://dx.doi.org/10.1017/jfm.2015.70>.
- [20] J.P. Goit, W. Munters, J. Meyers, Optimal coordinated control of power extraction in LES of a wind farm with entrance effects, *Energies* 9 (1) (2016) <http://dx.doi.org/10.3390/en9010029>.
- [21] W. Munters, J. Meyers, Dynamic Strategies for Yaw and Induction Control of Wind Farms Based on Large-Eddy Simulation and Optimization, *Energies* 11 (1) (2018) <http://dx.doi.org/10.3390/en11010177>.
- [22] S. Boersma, B.M. Doekemeijer, M. Vali, J. Meyers, J.W. van Wingerden, A control-oriented dynamic wind farm model: WFSim, *Wind Energy Sci.* 3 (1) (2018) 75–95, <http://dx.doi.org/10.5194/wes-3-75-2018>.
- [23] M. Vali, V. Petrović, S. Boersma, J.W. van Wingerden, L.Y. Pao, M. Kühn, Adjoint-based model predictive control for optimal energy extraction in waked wind farms, *Control Eng. Pract.* 84 (December 2018) (2019) 48–62, <http://dx.doi.org/10.1016/j.conengprac.2018.11.005>.
- [24] M.J. van den Broek, FRED - framework for wind farm flow regulation and estimation with dynamics, 2021, Available at: <https://github.com/TUDELFT-DataDrivenControl/FRED>,
- [25] M.J. van den Broek, J.W. van Wingerden, Dynamic Flow Modelling for Model-Predictive Wind Farm Control, *J. Phys. Conf. Ser.* 1618 (2) (2020) <http://dx.doi.org/10.1088/1742-6596/1618/2/022023>.
- [26] M.J. van den Broek, B. Sandefer, J.W. van Wingerden, Flow Modelling for Wind Farm Control: 2D vs . 3D, *J. Phys. Conf. Ser.* 2265 (3) (2022) <http://dx.doi.org/10.1088/1742-6596/2265/3/032086>.
- [27] M.F. Howland, J. Bossuyt, L.A. Martínez-Tossas, J. Meyers, C. Meneveau, Wake structure in actuator disk models of wind turbines in yaw under uniform inflow conditions, *J. Renew. Sustain. Energy* 8 (4) (2016) <http://dx.doi.org/10.1063/1.4955091>.
- [28] M. Bastankhah, F. Porté-Agel, Experimental and theoretical study of wind turbine wakes in yawed conditions, *J. Fluid Mech.* 806 (2016) 506–541, <http://dx.doi.org/10.1017/jfm.2016.595>.
- [29] J. Bartl, F. Mühle, J. Schottler, L. Sætran, J. Peinke, M. Adaramola, M. Hölling, Wind tunnel experiments on wind turbine wakes in yaw: Effects of inflow turbulence and shear, *Wind Energy Sci.* 3 (1) (2018) 329–343, <http://dx.doi.org/10.5194/wes-3-329-2018>.
- [30] P. Fleming, J. Annoni, L.A. Martínez-Tossas, S. Raach, K. Gruchalla, A. Scholbrock, M. Churchfield, J. Roadman, Investigation into the shape of a wake of a yawed full-scale turbine, *J. Phys. Conf. Ser.* 1037 (3) (2018) <http://dx.doi.org/10.1088/1742-6596/1037/3/032010>.
- [31] J. Katz, A. Plotkin, *Low-Speed Aerodynamics*, second ed., Cambridge University Press, 2001.
- [32] H. Lee, D.J. Lee, Effects of platform motions on aerodynamic performance and unsteady wake evolution of a floating offshore wind turbine, *Renew. Energy* 143 (2019) 9–23, <http://dx.doi.org/10.1016/j.renene.2019.04.134>.
- [33] K. Brown, D. Houck, D. Maniaci, C. Westergaard, Rapidly recovering wind turbine wakes with dynamic pitch and rotor speed control, in: AIAA Scitech 2021 Forum, (January) 2021, pp. 1–16, <http://dx.doi.org/10.2514/6.2021-1182>.
- [34] D.R. Houck, N. DeVelder, C.L. Kelley, Comparison of a mid-fidelity free vortex wake method to a high-fidelity actuator line model large eddy simulation for wind turbine wake simulations, *J. Phys. Conf. Ser.* 2265 (4) (2022) 042044, <http://dx.doi.org/10.1088/1742-6596/2265/4/042044>.
- [35] J. Dong, A. Viré, C.S. Ferreira, Z. Li, G. Van Bussel, A modified free wake vortex ring method for horizontal-axis wind turbines, *Energies* 12 (20) (2019) 1–24, <http://dx.doi.org/10.3390/en12203900>.
- [36] T. Berdowski, C.S. Ferreira, A. van Zuijlen, G. van Bussel, Three-dimensional free-wake vortex simulations of an actuator disc in yaw, in: *Wind Energy Symp.* 2018, (210029) 2018, <http://dx.doi.org/10.2514/6.2018-0513>.
- [37] T. Burton, D. Sharpe, N. Jenkins, E. Bossanyi, *Wind Energy Handbook*, Wiley, 2001.
- [38] L. Masset, O. Brüls, G. Kerschen, *Partition of the circle in cells of equal area and shape*, 2011.
- [39] L. Grüne, J. Pannek, *Nonlinear Model Predictive Control*, second ed., Springer, 2017, <http://dx.doi.org/10.1007/978-3-319-46024-6>.
- [40] R. Dorfman, P.A. Samuelson, R.M. Solow, *Linear Programming and Economic Analysis*, McGraw-Hill, New York, 1958.
- [41] S. Gros, *An Economic NMPC Formulation for Wind Turbine Control*, in: *Conf. Decis. Control*, 2013.
- [42] L. Grüne, Economic receding horizon control without terminal constraints, *Automatica* 49 (3) (2013) 725–734, <http://dx.doi.org/10.1016/j.automatica.2012.12.003>.
- [43] T. Lauß, S. Oberpeilsteiner, W. Steiner, K. Nachbagauer, The Discrete Adjoint Gradient Computation for Optimization Problems in Multibody Dynamics, *J. Comput. Nonlinear Dyn.* 12 (3) (2017) 1–10, <http://dx.doi.org/10.1115/1.4035197>.
- [44] R. Byrd, P. Lu, J. Nocedal, C. Zhu, A limited memory algorithm for bound constrained optimization, *J. Sci. Comput.* 16 (5) (1995) 1190–1208, <http://dx.doi.org/10.1137/0916069>.
- [45] D.P. Kingma, J.L. Ba, Adam: A method for stochastic optimization, in: *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, 2015, pp. 1–15, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [46] J.W. van Wingerden, L. Pao, J. Aho, P. Fleming, Active power control of waked wind farms, *IFAC-PapersOnLine* 50 (1) (2017) 4484–4491, <http://dx.doi.org/10.1016/j.ifacol.2017.08.378>.
- [47] J. Annoni, P.M. Gebraad, A.K. Scholbrock, P.A. Fleming, J.W. van Wingerden, Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model, *Wind Energy* (19) (2016) 1135–1150, <http://dx.doi.org/10.1002/we.1891>.
- [48] D. van der Hoek, S. Kanev, J. Allin, D. Bieniek, N. Mittelmeier, Effects of axial induction control on wind farm energy production - A field test, *Renew. Energy* 140 (2019) 994–1003, <http://dx.doi.org/10.1016/j.renene.2019.03.117>.
- [49] B.M. Doekemeijer, D. van der Hoek, J.W. van Wingerden, Closed-loop model-based wind farm control using FLORIS under time-varying inflow conditions, *Renew. Energy* 156 (2020) 719–730, <http://dx.doi.org/10.1016/j.renene.2020.04.007>.
- [50] M.F. Howland, A.S. Ghatge, S.K. Lele, J.O. Dabiri, Optimal closed-loop wake steering - Part 1: Conventionally neutral atmospheric boundary layer conditions, *Wind Energy Sci.* 5 (4) (2020) 1315–1338, <http://dx.doi.org/10.5194/wes-5-1315-2020>.
- [51] U. Ciri, M.A. Rotea, S. Leonardi, Model-free control of wind farms: A comparative study between individual and coordinated extremum seeking, *Renew. Energy* 113 (2017) 1033–1045, <http://dx.doi.org/10.1016/j.renene.2017.06.065>.
- [52] C. Cossu, Wake redirection at higher axial induction, *Wind Energy Sci.* 6 (2) (2021) 377–388, <http://dx.doi.org/10.5194/wes-6-377-2021>.
- [53] M.J. van den Broek, Free-vortex wake with discrete adjoint - model code, 2022, <http://dx.doi.org/10.4121/2027862>, Available at: <https://github.com/TUDELFT-DataDrivenControl/vortexwake>,
- [54] M.J. van den Broek, Free-vortex wake with discrete adjoint - dataset, 2022, <http://dx.doi.org/10.4121/20278590>,