FLORIDyn - A dynamic and flexible framework for real-time wind farm control

Becker, M.; Allaerts, D.; Van Wingerden, J. W.

DOI
10.1088/1742-6596/2265/3/032103

Publication date
2022

Document Version
Final published version

Published in
Journal of Physics: Conference Series

Journal of Physics: Conference Series

PAPER • OPEN ACCESS

FLORIDyn - A dynamic and flexible framework for real-time wind farm control

To cite this article: M Becker *et al* 2022 *J. Phys.: Conf. Ser.* **2265** 032103

FLORIDyn - A dynamic and flexible framework for real-time wind farm control

M Becker[1], D Allaerts[2], J W van Wingerden[1]

[1] Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, NL

[2] Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, NL

E-mail: marcus.becker@tudelft.nl

Abstract. This paper presents a new framework of the FLOw Redirection and Induction Dynamics (FLORIDyn) model. It is able to dynamically simulate the wake behaviour in wind farms under heterogeneous and changing environmental conditions at a low computational cost. The novelty of this work is the improved segregation of wake dynamics and wake influence: the framework creates Observation Points (OPs) at each turbine, which propagate wind field states and turbine states downstream and follow the wind direction of the free stream velocity. These observation points cover the dynamic aspects of the simulation. The OPs, along with the stored states, are now used to derive so-called Temporary Wind Farms (TWF), which approximate the effective intra-farm wind conditions at a given location. Within these TWF, the flow conditions are homogeneous and steady state. This way, arbitrary wake models can be used to calculate the farm influence on the location. The FLORIDyn framework also provides interfaces to flow field estimators, which is tested with an effective wind speed estimator. A nine turbine case is used to highlight the quality and performance of the simulation result. Compared to its predecessor, the new FLORIDyn framework decreases the computational cost by one to two orders of magnitude, which makes it a promising candidate for real-time model predictive dynamic wind farm control.

1. Introduction

As wind turbines extract kinetic energy from the wind and transform it into electric energy, they leave an area of decreased wind speed in the flow field. This area is called the wake and can influence downstream turbines in a wind farm by decreasing the amount of energy these can extract from the wind. The wake is shaped by uncontrollable environmental influences and controllable turbine states. An optimization of the latter can increase the power generated by the entire wind farm: for instance an increase of the misalignment of the wind turbine with the wind direction can redirect the wake and reduce the influence on downstream turbines [1]. This control strategy is known as wake steering which is considered to be a viable solution for wind farm control [2]. One way to implement wake steering is to use a surrogate wind farm model to find the optimal control settings. The parametric FLORIS (FLOw Redirection and Induction in Steady state) model has made this approach feasible by being computationally cheap, easy to tune and implement, whist being accurate enough to lead to performance improvements in higher fidelity simulations and wind tunnel experiments [3, 4]. A downside of FLORIS is that it neglects the dynamic behaviour of the wake and the surrounding flow and only a few implementations are capable of simulating heterogeneous environments and thus optimally suitable for dynamic wind farm control [5].

There have been a few proposals to address the lack of dynamics, one of which is the FLOw Redirection and Induction Dynamics (FLORIDyn) model, presented in 2014 by Gebraad et al. [6]. The approach is to use so called Observation Points (OPs) which travel downstream, starting at the rotor plane and inherit the turbine state and the wind field state. With this information and the boundaries of the FLORIS wake, the OPs calculate their path and propagate downstream. This way turbine state changes propagate with the OPs downstream and have a delayed effect at other turbines. While the model pioneered in its methodology and shows promising results, it also has some flaws: Firstly, the wind direction is fixed and the model does not allow for heterogeneous conditions. Secondly, the used Zone FLORIS model has since been overhauled and more capable and accurate parametric models have been developed. Thirdly, due to the OP travel behaviour, parts of the wake could overlap causing inconsistent edge cases.

In 2022, Becker et al. addressed a number of shortcomings of the Zone FLORIDyn model and extended it with new features [7]. These are, among others, the implementation of the 3D Gaussian FLORIS model, the inclusion of heterogeneous and changing flow conditions and a new method for Observation Point (OP) distribution in the wake. While the Gaussian FLORIDyn model is able to keep the computational cost low, the simulation times grow exponentially and become unfeasible for model based control approaches for a wind farm with a large number of turbines. The OPs also discretize the Gaussian shape and limit the function by their distribution. The design is quite interconnected with the Gaussian model and does not provide a simple interface to switch the parametric model as desired.

This paper aims to structurally rework aspects of the Gaussian FLORIDyn model to solve or reduce the issues it has, while maintaining its strengths. The presented work evolves a concept from the Gaussian FLORIDyn model: To simulate a wake in heterogeneous conditions, the implementation decoupled wake and flow properties in a wake and a world coordinate system. We develop this approach further to introduce Temporary Wind Farms (TWFs) which approximate the behaviour of static turbine wakes under heterogeneous conditions. However, this formal addition has significant implications, as it changes the way FLORIDyn interacts with the underlying FLORIS model and the resource requirements of FLORIDyn. FLORIS is now treated much more as a generic model with an interface. This allows an exchange of the used wake model. This new structural framework can create a dynamic wake from any (steady state) wake representation, also from those which include secondary wake steering effects. This will also allow to include the most recent developments of the FLORIS models and such alike. The approach also lowers the computational cost in our simulations and allows a scale in which dynamic real time wind farm control with a large number of turbines becomes more viable.

The remainder of the paper is structured as follows: Section 2 presents the new FLORIDyn framework. A nine turbine case is presented in Section 3, along with the computational cost of the framework. Section 4 concludes the work presented in this paper.

2. Development of a generic FLORIDyn framework

First, the propagation of OPs and states is described (Sec. 2.1), the novel framework is then described in Section 2.2. How to couple the framework with measurement data is discussed in Section 2.3. Section 2.4 concludes the methodology by highlighting specific aspects of the used implementation which diverge from previous models.

2.1. Propagation of Observation Points and states

This section describes the mathematics[1] of the OPs, which represent the wake of a turbine. In the new FLORIDyn framework, OPs only follow the centerline. This is in contrast to previous versions, where OPs were distributed across the entire wake area. For each turbine, a new OP is created every time step, while the oldest one is disregarded. Each OP has three sets of states: the location of the OP, denoted as $\mathbf{x}_{\mathrm{OP}}$, the turbine state $\mathbf{x}_{\mathrm{T}}$ and the wind field state $\mathbf{x}_{\mathrm{WF}}$. The propagation follows the concept introduced in [7] where the downwind step is calculated in the wake coordinate system $\mathcal{K}_1$ and then translated to the world coordinate system $\mathcal{K}_0$:

$$\mathbf{x}_{\mathrm{OP},0}(k+1) = \mathbf{x}_{\mathrm{OP},0}(k) + \mathbf{R}_{01}(\mathbf{x}_{\mathrm{WF},\varphi}) \underbrace{\left[\mathbf{x}_{\mathrm{OP},1}(k+1) - \mathbf{x}_{\mathrm{OP},1}(k)\right]}_{\text{step in } \mathcal{K}_1}, \tag{1}$$

where $\mathbf{x}_{\mathrm{OP}} = [\mathbf{x}_{\mathrm{OP},0}, \mathbf{x}_{\mathrm{OP},1}]^\top$ denotes the location state of the OP with $\mathbf{x}_{\mathrm{OP},0} = [x_0, y_0, z_0]^\top$ in the world coordinate system and $\mathbf{x}_{\mathrm{OP},1} = [x_1, y_1, z_1]^\top$ in the wake coordinate system. Note, that x_1 denotes the downwind direction and y_1, z_1 the crosswind coordinates, y_1 from right to left, z_1 from down to up as shown in Figure 1. The wind field state is written as $\mathbf{x}_{\mathrm{WF}}$ and $x_{\mathrm{WF},\varphi}$ only refers to the wind direction φ. The rotational matrix $\mathbf{R}_{01}$ rotates the OP propagation step in $\mathcal{K}_1$ around the z-axis in mathematical positive direction and therefore transforms vectors from $\mathcal{K}_1$ to $\mathcal{K}_0$:

$$\mathbf{R}_{01}(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The positional update in the wake coordinates is calculated as follows:

$$x_{\mathrm{OP},1,\mathrm{x}}(k+1) = x_{\mathrm{OP},1,\mathrm{x}}(k) + \Delta t\, x_{\mathrm{WF,u}} \tag{3}$$

$$\mathbf{x}_{\mathrm{OP},1,\mathrm{y,z}}(k+1) = \delta(x_{\mathrm{OP},1,\mathrm{x}}(k+1), \mathbf{x}_{\mathrm{T}}, \mathbf{x}_{\mathrm{WF}}) \tag{4}$$

where Δt is the simulation time step, $\mathbf{x}_{\mathrm{WF,u}}$ is the free wind speed and δ denotes the deflection function. When an OP is created, $\mathbf{x}_{\mathrm{OP},1} = \mathbf{0}$ and $\mathbf{x}_{\mathrm{OP},0} = \mathbf{t}_0$ where $\mathbf{t}_0$ is the world location of the wind turbine rotor center.

The description of $\mathbf{x}_{\mathrm{T}}$ and $\mathbf{x}_{\mathrm{WF}}$ is purposefully kept generic as these states may vary with the used parametric model. The turbine state $\mathbf{x}_{\mathrm{T}}$ can be summarized as all states that are turbine specific and are needed to calculate the wake shape. Examples are the yaw angle γ or the axial induction a. The wind field state $\mathbf{x}_{\mathrm{WF}}$ contains all states necessary to propagate the wake and to calculate it. In the presented formulation the free wind speed and the wind direction are mandatory, but $\mathbf{x}_{\mathrm{WF}}$ can also include the ambient turbulence intensity for instance. States in $\mathbf{x}_{\mathrm{WF}}$ are also considered to be measurable by sensors at other locations than the turbine and could be corrected by state estimation methods.

2.2. Extrapolation from OPs and the creation of temporary wind farms

The reduction of the number of OPs leads to a sparse wake description. Where other FLORIDyn formulations have the possibility to justify a nearest neighbour interpolation due to a high density

[1] The notation of Section 2.1 and 2.2 is as follows: small italic letters denote scalars (e.g. x), bold small letters denote column vectors (e.g. $\mathbf{x}$) and bold capital letters denote matrices (e.g. $\mathbf{R}$). Coordinate systems are denoted by $\mathcal{K}$. Square brackets organize equations or define matrices and vectors, round brackets are function inputs or properties. Lower indices of vectors first specify the parent object or type, then the coordinate system and lastly the extracted value. Depending on the context, some part of the index might be missing, but the order remains. If the extracted value is specified, the vector might reduce to a scalar and is written accordingly, e.g. $\mathbf{x}_{\mathrm{OP},0} \to x_{\mathrm{OP},0,y}$. Lower indices of matrices denote which coordinate system they transform into which, e.g. $\mathbf{R}_{01}$ transforms $\mathcal{K}_1$ into $\mathcal{K}_0$, such that $\mathbf{x}_0 = \mathbf{R}_{01}\mathbf{x}_1$.

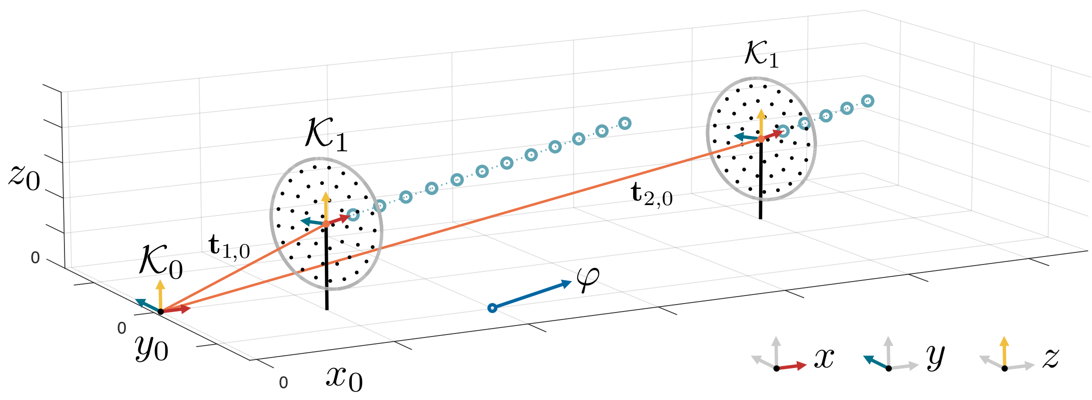

Figure 1. Visualization of the coordinate system $\mathcal{K}_0$ and the two $\mathcal{K}_1$ systems of two turbines. The figure also shows the OPs flowing downstream as well as the turbine location vectors $\mathbf{t}_{T_i,0}$ and the wind direction φ.

of OPs, the current description cannot. Therefore a method to extrapolate the wake influence is presented. The general problem is posed as follows:

What is the influence of the turbines $\mathcal{T}$ at the world location $\mathbf{l}_0$?

For each turbine $T_i \in \mathcal{T}$ we choose the two OPs (OP^1 and OP^2) in front and behind the closest point on the centerline of T_i to $\mathbf{l}_0$. Then we linearly interpolate between OP^1 and OP^2 to obtain OP^* in such a way that the distance to $\mathbf{l}_0$ is minimal:

$$w = \frac{\left[\mathbf{x}_{\mathrm{OP}^2,0} - \mathbf{x}_{\mathrm{OP}^1,0}\right]^\top \left[\mathbf{l}_0 - \mathbf{x}_{\mathrm{OP}^1,0}\right]}{\left[\mathbf{x}_{\mathrm{OP}^2,0} - \mathbf{x}_{\mathrm{OP}^1,0}\right]^\top \left[\mathbf{x}_{\mathrm{OP}^2,0} - \mathbf{x}_{\mathrm{OP}^1,0}\right]} \tag{5}$$

which returns the weight $w \in [0, 1]$. The $\mathcal{K}_0$ location of OP^* is then given by

$$\mathbf{x}_{\mathrm{OP}^*,0} = \mathbf{x}_{\mathrm{OP}^1,0} + w\left[\mathbf{x}_{\mathrm{OP}^2,0} - \mathbf{x}_{\mathrm{OP}^1,0}\right] = (1-w)\mathbf{x}_{\mathrm{OP}^1,0} + w\,\mathbf{x}_{\mathrm{OP}^2,0}\;. \tag{6}$$

This is visualized in Figure 2. The weight is then also used to interpolate the other states $\mathbf{x}_{\mathrm{OP}^*,1}$, $\mathbf{x}_{\mathrm{T}^*}$ and $\mathbf{x}_{\mathrm{WF}^*}$, equivalent to Eq. (6). In the edge cases where the first or last OP of T_i is the closest OP, no interpolation is performed and $\mathrm{OP}^* = \mathrm{OP}^{\mathrm{Edge}}$. This means, that if only one OP is used, the framework extrapolates from that OP and directly returns the underlying parametric wake model. After applying Eq. (6), every turbine in $\mathcal{T}$ is represented by an OP^* close to $\mathbf{l}_0$.

The next step is to locate $\mathbf{l}_0$ in $\mathcal{K}_1$ of turbine T_i, based on the states represented by OP^*. This is done by rotating the vector $\mathbf{l}_0 - \mathbf{x}_{\mathrm{OP}^*,0}$ (pointing from OP^* to $\mathbf{l}_0$) from the world frame $\mathcal{K}_0$ to the wake frame $\mathcal{K}_1$ of T_i with the inverted rotational matrix: $\mathbf{R}_{01}^{-1} = \mathbf{R}_{01}^\top = \mathbf{R}_{10}$. This vector is then added to the $\mathcal{K}_1$ position of OP^*:

$$\mathbf{l}_{T_i,1} = \mathbf{x}_{\mathrm{OP}^*,1} + \mathbf{R}_{10}(x_{\mathrm{WF}^*,\varphi})\left[\mathbf{l}_0 - \mathbf{x}_{\mathrm{OP}^*,0}\right]\;. \tag{7}$$

As a result, $\mathbf{l}_{T_i,1}$ can be calculated for all turbines $T_i \in \mathcal{T}$. The vector $\mathbf{l}_{T_i,1}$ stems from the origin of $\mathcal{K}_1^*$, the turbine coordinate system of T_i based on the states of OP^*. This means that if the wind field states of OP^1 and OP^2 (which OP^* is derived from) have changed since their initialization, the origins of $\mathcal{K}_1^*$ and $\mathcal{K}_1$ are not at equal locations in $\mathcal{K}_0$. Figure 2 illustrates this.

With this information we can create a Temporary Wind Farm (TWF). The TWF approximates the environment around $\mathbf{l}_0$ and can be seen as the effective wind farm at the

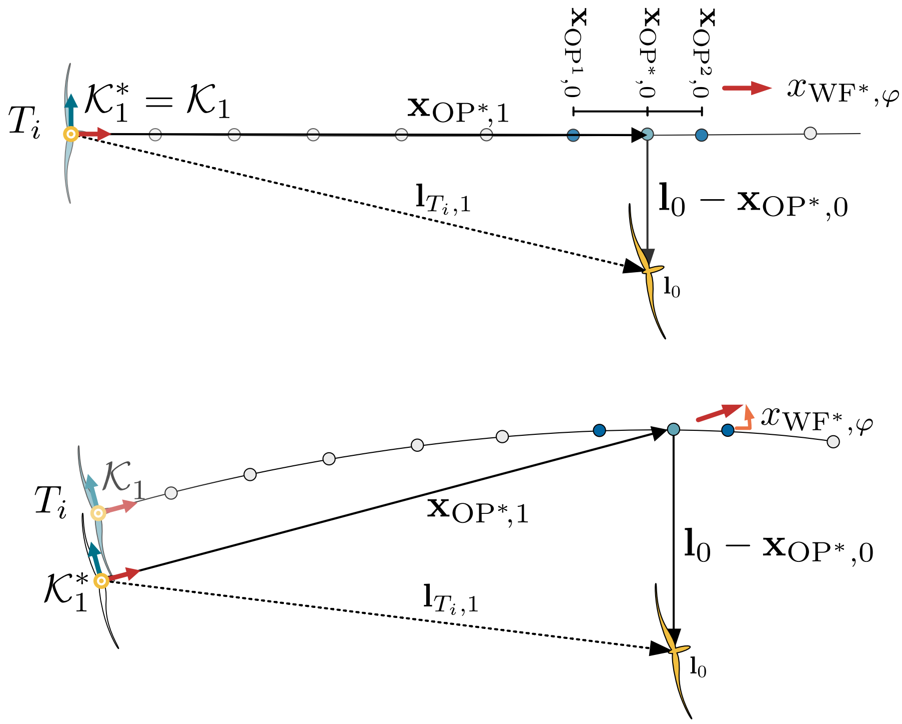

Figure 2. The top figure shows the interpolation of OP* and how it is used to calculate the position of $\mathbf{l}_0$ in $\mathcal{K}_1^*$ of T_i. The wind direction is steady and $\mathcal{K}_1$ of the turbine is equal to $\mathcal{K}_1^*$ of OP*. During a wind direction change, the center line propagates to its new steady state, which leads to a change of $x_{\mathrm{WF}^*,\varphi}$ and therefore to a temporary mismatch between the locations of the origins of $\mathcal{K}_1^*$ and $\mathcal{K}_1$ in $\mathcal{K}_0$ (lower figure).

requested location: This includes all turbines in $\mathcal{T}$ as perceived by the OPs* close to $\mathbf{l}_0$. By using the data from the OPs*, locations of the turbines in the TWF can differ from their real world locations, as shown in Figure 2. A new coordinate system $\mathcal{K}_2$ is created which is characteristically similar to $\mathcal{K}_1$: The wind direction is fixed along the x_2 axis, y_2 is the crosswind direction and the z_2 axis is pointing upwards. First, we place $\mathbf{l}_0$ in $\mathcal{K}_2$: its location can be chosen arbitrarily, but for convenience we will choose $\mathbf{l}_2 = [0, 0, l_{0,z}]^\top$ which ensures that $z_2 = 0$ is ground level if elevation is not part of the simulation. The wind turbines can be located by reversing the vectors $\mathbf{l}_{T_i,1}$:

$$\mathbf{t}_{T_i,2} = \mathbf{l}_2 - \mathbf{R}_{21}(0)\,\mathbf{l}_{T_i,1} \ . \tag{8}$$

Since $\mathcal{K}_1$ and $\mathcal{K}_2$ share the same wind direction, $\mathbf{R}_{21}(0)$ is a 3×3 identity matrix. The turbine states are given by the respective OP*. The wind field states however are not entirely defined, only the wind direction is fix. In practice however, we can assume that the wind field states of the different OP* will be very similar as they are also in local proximity to each other. In our implementation we averaged between the two closest OP*, weighted by the distance to $\mathbf{l}_0$. The TWF is now complete and approximates the wake and wind field conditions around $\mathbf{l}_0$ as a wind farm in homogeneous conditions. The TWF in $\mathcal{K}_2$ can be evaluated in an arbitrary wake model to return the influence of $\mathcal{T}$ at $\mathbf{l}_2$ which approximates the influence of $\mathcal{T}$ at $\mathbf{l}_0$.

Note that due to the way the TWF are derived, existing steady state wake models can be used to their full extent: The presented framework does not require a certain wake shape or wake merging method. With multiple turbines in one TWF, secondary wake steering effects could be captured as well. This ensures that novel developments in these wake models can also be tested in a dynamic, heterogeneous environment.

Another aspect is that the computational load can be scaled by limiting the size of $\mathcal{T}$. If downstream turbines do not have an influence on upstream turbines in the used wake model, they can be disregarded from $\mathcal{T}$. The same goes for turbines with a significant upwind or crosswind distance to $\mathbf{l}_0$. This allows a split of an entire wind farm into many smaller wind farms which can be evaluated in parallel.

2.3. Interfaces and the Immersion and Invariance estimator

The presented framework simplifies the state architecture of the simulation and purposefully treats turbines as sensors and actuators providing information to the OPs. The OPs in return provide an estimate of the intra wind farm flow. Estimators of flow field metrics can convert measurement data into metrics which the FLORIDyn framework can use and store as states in the OPs.

In this work, we present one implementation with the Immersion and Invariance (I&I) estimator as described in [8]. It returns the effective wind speed at the rotor plane $\hat{u}_{\mathrm{eff,R}}$ based on the generator torque and the rotor torque. A required component is the power coefficient table C_{P} which is depend on factors such as the blade pitch and tip-speed-ratio. The used look-up tables were obtained in FAST. The estimator is derived for wind turbines without yaw misalignment, which limits its use until it is adapted.

Since FLORIDyn uses the free wind speed to propagate the OPs, $\hat{u}_{\mathrm{eff,R}}$ has to be converted to $\hat{u}_{\mathrm{free,R}}$. For free stream turbines we neglect induction and blockage effects and assume that $\hat{u}_{\mathrm{eff,R}} = \hat{u}_{\mathrm{free,R}}$. The FLORIDyn framework delivers an estimate of the reduction of the wind speed for downstream turbines, which can be used to calculate $\hat{u}_{\mathrm{free,R}}$ by dividing $\hat{u}_{\mathrm{eff,R}}$ by the reduction.

2.4. Implementation

This section briefly summarizes the details of the implementation used in this work. The implementation follows in its core [7] but diverges in some aspects highlighted here.

Rotor plane discretization The rotor plane is discretized to calculate the influence of turbine wakes and sum it, weighted by the areas of the subdivided plane. We propose the use of the Isocell algorithm which splits a circle into n equally sized and regularly distributed parts [9]. The downside of this algorithm is that it can only provide sets for certain values of n. This limitation is found to be acceptable as the steps between possible values of n are relatively small ($n = 3, 12, 27, 48, 75, 108, 147, ...$). In this work 48 rotor points (RP) were used, distributed in the y_1, z_1 plane and translated, according to the yaw orientation and turbine position, into $\mathcal{K}_0$. Only the rotor center is used as location to set up the TWF. The resulting distribution can be seen in Figure 1.

Thrust and power coefficient In [7] look-up tables were used to get values for the thrust and power coefficient. As these look-up tables turned out to be incomplete and needed to be limited to be useful, we decided to fall back to the actuator disc method and calculate the coefficients based on the axial induction factor.

Parameter set The presented FLORIDyn framework was also used in a parameter uncertainty quantification study [10]. The study was conducted with the FLORIS parameter set found by [11] as starting point (see Table 1) and is also used in this study. Additionally, the efficiency term η was set to be neutral, the yaw correction coefficient p_p was set following [7].

Added turbulence Influence of the added turbulence is implemented as presented in [12]. This formulation only defines a downwind development of the added turbulence and no crosswind

Table 1. Parameters used with the aspects they mainly influence

Wake expansion		Added turbulence				Near wake length		Power	
k_a	k_b	$k_{f,a}$	$k_{f,b}$	$k_{f,c}$	$k_{f,d}$	α^*	β^*	η	p_p
$5.37 \cdot 10^{-1}$	$-8.48 \cdot 10^{-4}$	7.84	4.57	$4.3 \cdot 10^{-1}$	$-2.46 \cdot 10^{-1}$	1.088	$2.22 \cdot 10^{-1}$	1	2.2

part. A recommendation is to include all turbines within 2D crosswind distance and 15D upwind distance [13]. Under changing environmental conditions, this leads to sudden changes in the added turbulence levels as turbines enter or leave the proposed area.

Instead of looking upstream for influencing turbines, one can also look downstream to identify the turbines influenced by the wake. In the previous FLORIDyn implementation the distributed OPs were used to determine the area of influence [7]. In the here presented Framework, OPs do not cover the wake area anymore and rather extrapolate their influence from the centerline. Following this approach the added turbulence value is weighted by the Gaussian distribution of the wake. To achieve a similar width as in [7], the width factor is multiplied by three.

3. Case study

The presented case is a nine turbine simulation case in Section 3.1. The wind field is turbulent and changes the flow direction during the simulation. Section 3.2 discusses the computational cost aspect of the framework in comparison to the previous implementation.

3.1. Nine turbine case

The wind farm layout is a regularly spaced 3×3 layout in a 3×3 km domain. The turbines are placed 900 m apart from each other and 600 m from the domain edge. The validation simulation is done in SOWFA, where the DTU10MW reference turbines are simulated with the Actuator Disc Method and the domain is discretized in $10 \times 10 \times 10$ m cells without refinement areas and the time step is set to 0.5 s. The flow is turbulent with an ambient turbulence intensity of $\approx 6\%$ and a mean wind speed of 8.2 ms^{-1}. During the simulation the wind direction uniformly changes from 255° to 195°, with a constant change rate from simulation time $t = 600$ s to 900 s. The turbines remain perpendicular to the wind direction during the change. This case was also used in [7].

Figure 3 shows the power generated by the four downstream turbines during the simulation. The figure compares four different versions of the same simulation: The FLORIDyn framework with and without the I&I estimator, the validation data (here SOWFA) and what FLORIS would return. The grey area in the plots highlights the time during which the wind direction changes.

Looking at the difference between steady state and dynamic models, we can see that FLORIS is not able to accurately predict the timing of the wake influence of upstream turbines on downstream turbines due to the wind direction change. The power generated also shows single peaks where the validation data shows multiple occurrences of interaction. The FLORIDyn framework returns identical results during steady state. Furthermore, FLORIDyn is able to predict the timing of the wake interaction. The model does lack information about the wind speed changes, which can lead to significant differences in power generated, compared to the validation data. These differences become minimal when the I&I estimator is added. Due to the dynamic nature of the I&I, the wind speed estimate is experienced with a minor phase shift and does not contain higher frequency components from the validation data. These results could

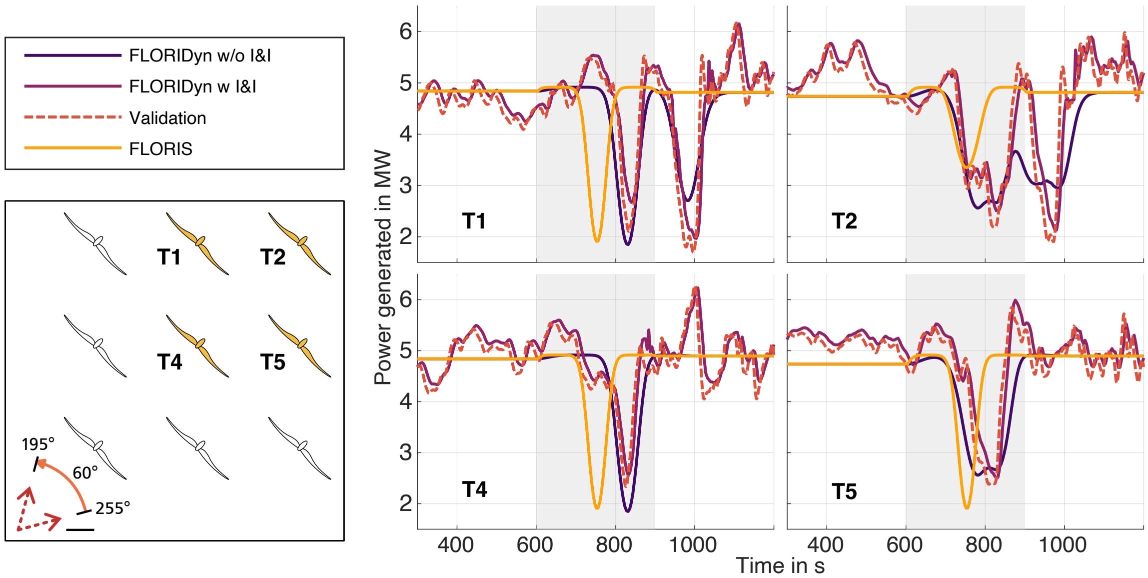

Figure 3. Generated power by the FLORIDyn framework without and with the I&I estimator, in comparison to the validation data and FLORIS. The grey area indicates the time in which the wind direction turns. Only the four down stream turbines are plotted here, their location is marked on the left side as well as the change of wind direction.

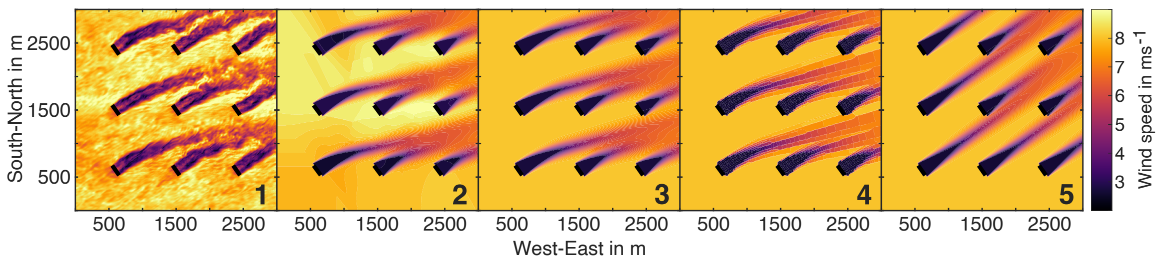

Figure 4. Flow field comparison during a wind direction change between SOWFA (1) and the new FLORIDyn framework, with I&I estimator (2) and without (3), the Gaussian FLORIDyn model (4), as well as FLORIS (5). The snapshots are taken at simulation time $t = 700$ s.

potentially be even more improved by adequate tuning of the I&I parameters. The downside of the use of the I&I estimator is that it removes the FLORIDyn influence from the calculation of the power generated as it directly provides the effective wind speed. But it is in return possible to use the turbine as a sensor and to provide a wind field state estimate to the OPs.

The flow fields at hub height in Figure 4 compare the validation data with the FLORIDyn framework with and without I&I estimator, the Gaussian FLORIDyn model and the FLORIS model. FLORIS shows the current wind direction, as the model immediately reacts to the new wind direction and the far wake does not slowly adapt, as is the case with the validation data and all FLORIDyn variants. The Gaussian FLORIDyn shows a mosaic-like flow field due to the nearest neighbour interpolation employed to get the influence of the OPs. The novel FLORIDyn framework allows an accurate use of the FLORIS wake, adapted to the wind direction change. Adding the I&I estimator results in a very similar wake shape but a much more detailed flow field which can incorporate locally different wind speeds based on realistic turbine data.

3.2. Performance

Figure 5 shows the computational performance of the presented FLORIDyn framework in seven test cases, four of which have also been tested with the I&I estimator, and three which have been simulated in the previously published Gaussian FLORIDyn model [7]. The measurements tell how long it takes to simulate one time step. They are only roughly representative, as the performance will vary with the wind farm layout, the environmental conditions, the implementation and the hard- and software[2]. The times capture only the simulation and no visualization, nor initialization. Each measurement is the mean of 250 or more simulation time steps. The 9_1 turbine case is the same case as discussed in Section 3.1. The $1, 2, 3$ and 9_2T cases feature a turbulent wind field but steady wind direction. The 23 and 54T cases are derived from real wind farm layouts. The cases run with a constant wind speed and direction.

One observation is that the new framework, in comparison to the previous Gaussian FLORIDyn, decreases the computational cost by one to two orders of magnitude, depending on the case. The new implementation also scales better, to a point where a 54 turbine case can be simulated at a similar speed as a three turbine case in the previous implementation. This improvement allows more optimization steps in the same time, which, in return allows the application of more demanding control strategies. The TWF concept is also by design suited for parallel computation, something which was not performed in this study.

Another observation is that the added computational cost of the I&I estimator varies between the cases. This is due to the fact that the estimator runs with the frequency of the measurement data from the turbines. The 2 and 9_1 T case have 2 Hz data, the 1 T case 5 Hz and the 3 T case 25 Hz. The FLORIDyn framework runs at 0.25 Hz in all cases. Unfortunately we do not have the data to run the I&I estimator in the $9_2, 23, 54$ turbine cases.

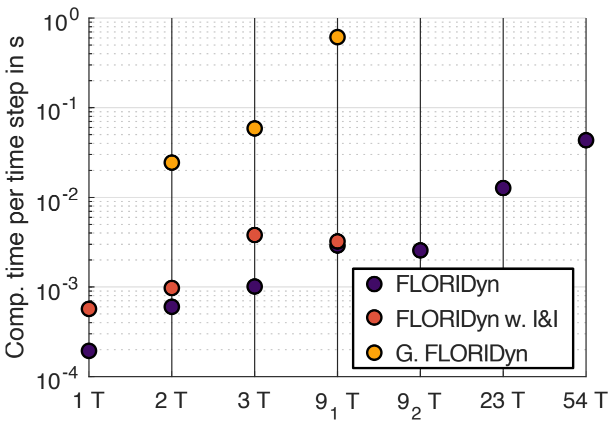

Figure 5. Computational performance of the new FLORIDyn framework with and without the I&I estimator in comparison to the Gaussian FLORIDyn model. Note the logarithmic time scale and the number of turbines for the different cases. The FLORIDyn framework values are mean value of 10 consecutive simulations.

4. Conclusion

This paper presents a formal definition of the FLORIDyn framework. It decouples the wake propagation under heterogeneous, changing conditions from the wake shape description. This allows the use of wake models which are designed for steady state conditions. The new interface to a generic wake model is achieved by creating temporary wind farms which approximate intra wind farm flow at any given location. They translate the heterogeneous environmental conditions into a steady state homogeneous space in which the wake model is evaluated. This allows the dynamic use of most recent steady state wake model implementations with minimal effort and makes the concept usable for future generations of engineering wake models.

[2] Simulations performed in Matlab 2021a, single threaded on a laptop (Intel Core i7 vPro 10th Gen, 16GB RAM).

This paper also demonstrates how measurements from a turbine can be included into the FLORIDyn framework: The I&I effective wind speed estimator is used to convert the turbine into a sensor for the simulation, which significantly decreases the differences in the power generated and improves the intra wind farm flow estimate.

Compared to the previous Gaussian FLORIDyn model for heterogeneous conditions, the FLORIDyn framework is able to decreased computational effort by one to two orders of magnitude. This is a necessary step in order to provide real time control inputs in a realistic wind farm scenario with a large number of turbines. The temporary wind farms also provide a basis for parallel computing, something which was not part of this research.

For future work we propose to implement the framework into a model based closed loop control strategy. Within the FLORIDyn framework a suitable engineering model can be used to evaluate desired quantities of interest under changing environmental conditions. This could be for instance a wake steering strategy to maximize power while taking loads into account. Additionally other flow field estimators can also be implemented to improve the connection of FLORIDyn with data from wind farms.

Acknowledgement This work is part of the research programme Robust closed-loop wake steering for large densely space wind farms with project number 17512 and partly financed by the Dutch Research Council (NWO).

References

[1] T. Knudsen, T. Bak, and M. Svenstrup. Survey of wind farm control-power and fatigue optimization: Survey of wind farm control. *Wind Energy*, 18(8):1333–1351, August 2015.

[2] J. W. van Wingerden, P. A. Fleming, T. Göçmen, 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, and G. Giebel. Expert elicitation on wind farm control. *Journal of Physics: Conference Series*, 1618:022025, sep 2020.

[3] P. M. O. Gebraad, F. W. Teeuwisse, J. W. van Wingerden, P. A. Fleming, S. D. Ruben, J. R. Marden, and L. Y. Pao. A data-driven model for wind plant power optimization by yaw control. In *2014 American Control Conference*, pages 3128–3134, June 2014. ISSN: 2378-5861.

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

[5] B. M. Doekemeijer, D. van der Hoek, and J. W. van Wingerden. Closed-loop model-based wind farm control using floris under time-varying inflow conditions. *Renewable Energy*, 156:719–730, 2020.

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

[7] M. Becker, B. Ritter, B. Doekemeijer, D. van der Hoek, U. Konigorski, D. Allaerts, and J. W. van Wingerden. The revised FLORIDyn model: Implementation of heterogeneous flow and the Gaussian wake. *Wind Energ. Sci. Discuss. [preprint]*, January 2022.

[8] Y. Liu, A. K. Pamososuryo, R. M. G. Ferrari, and J. W. van Wingerden. The Immersion and Invariance Wind Speed Estimator Revisited and New Results. *IEEE Control Systems Letters*, 6:361–366, 2022.

[9] L. Masset, O. Bruls, and G. Kerschen. Partition of the circle in cells of equal area and shape. *Open Repository and Bibliography*, page 6, May 2011.

[10] V. V. Dighe, M. Becker, T. Göçmen, B. Sanderse, and J. W. van Wingerden. Uncertainty quantification of a dynamic wind farm control model: FLORIDyn. *Torque 2022*.

[11] B. M. Doekemeijer, D. van der Hoek, and J.W. van Wingerden. Closed-loop model-based wind farm control using FLORIS under time-varying inflow conditions. *Renewable Energy*, 156:719–730, August 2020.

[12] A. Crespo and J. Hernández. Turbulence characteristics in wind-turbine wakes. *Journal of Wind Engineering and Industrial Aerodynamics*, 61(1):71–85, June 1996.

[13] A. Farrell, J. King, C. Draxl, R. Mudafort, N. Hamilton, C. J. Bay, P. Fleming, and E. Simley. Design and analysis of a spatially heterogeneous wake. *Wind Energy Science*, page 25, 2020.