POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E
DELL'INFORMAZIONE

Master in Aeronautical Engineering

Validation of a Superposed ROM for Wind Farm Control

Author:
Fabio Giovanni PARISI

Student ID:
894027

Supervisor:
Prof. Stefano CACCIOLA

Co-supervisors:
Dr. Filippo CAMPAGNOLO
Prof. Elisa CAPELLO

ACADEMIC YEAR 2018-2019

Declaration of Authorship

I, Fabio Giovanni PARISI, declare that this thesis titled, "Validation of a Superposed ROM for Wind Farm Control" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Milan, December 2019

Zwei Dinge erfüllen das Gemüt mit immer neuer und zunehmender Bewunderung und Ehrfurcht, je öfter und anhaltender sich das Nachdenken damit beschäftigt: der bestirnte Himmel über mir und das moralische Gesetz in mir.

Immanuel Kant

Sommario

Nell'ambito dell'energia eolica, il ruolo dei parchi eolici ha assunto un'attenzione sempre maggiore, promettendo una riduzione di costi significativa grazie ad un'economia di scala. il potenziale di tali parchi tuttavia non è ancora pienamente sfruttato dalle strategie di controllo attuali, che trascurano le interazioni aerodinamiche all'interno dei gruppi di turbine. Strategie di controllo coordinato (*wind farm control*) appaiono perciò sempre più interessanti poiché garantiscono un aumento della potenza prodotta e una migliore gestione dei carichi, portando in ultima analisi ad una riduzione del costo dell'energia. Ai fini di favorirne lo studio, risulta vantaggioso sviluppare modelli di parco eolico capaci di descrivere fedelmente la dinamica dell'impianto con un basso costo computazionale. Questi requisiti contrastanti sono soddisfatti da modelli ad ordine ridotto (ROMs) generati da precedenti simulazioni di fluidodinamica computazionale (CFD) dell'intero impianto. D'altra parte, queste ultime possono diventare eccessivamente onerose all'aumentare della complessità e delle dimensioni del parco eolico. Questa tesi presenta un modo possibile per risolvere tale problema, favorendo la generazione di un ROM assemblando dati CFD riguardanti la dinamica della corrente attorno ad una turbina singola, più gestibili rispetto ad una simulazione dell'intero parco. In primo luogo il lavoro si concentra sulla sovrapposizione di scie mediate nel tempo, ricavate utilizzando il software numerico SOWFA. A questo proposito, viene proposto un nuovo metodo di sovrapposizione, capace di ottenere risultati migliori rispetto agli altri schemi usualmente adottati. Inoltre, ai fini della sintesi di un modello ridotto, le potenze prodotte dalle turbine devono essere stimate. Per raggiungere questo obiettivo, due diverse procedure sono state adottate, ovvero lo stesso schema di sovrapposizione precedente e il software FLORIS. In ultimo, si esamina la generazione di un proiettore tramite approcci alternativi, ai fini di limitare il carico computazionale imposto dalla tecnica di *proper orthogonal decomposition* (POD), ampiamente utilizzata. Sebbene la presente analisi sia stata effettuata per facilitare lo sviluppo di un controllo del parco eolico basato sul ridirezionamento della scia in yaw, tutti i risultati possono essere estesi a differenti strategie di controllo.

Abstract

Within the wind energy sector, the role of wind farms is drawing an increasing attention as it promises a significant cost reduction due to an economy of scale. Yet, the potential of wind power plants is still not fully exploited by the current turbine control strategies, which neglect the aerodynamic interactions taking place in the cluster. Thus, wind farm control strategies are becoming more and more appealing as they would allow an increase in power production and a better load management, ultimately lowering the cost of energy. To foster the research of these novel control methods, it is required the development of wind farm models that can describe with high-fidelity the plant dynamic at a low computational costs. These contrasting requirements seem satisfied by reduced-order models (ROMs) generated from previous computational fluid dynamics (CFD) simulations of the entire plant. However, the latter may become a burden - or even unfeasible - when dealing with farms of increasing complexity and size. This thesis presents a possible way to circumvent this problem as it would allow the generation of a ROM assembling more manageable CFD data of the flow behaviour around individual turbines. At first, the work focuses on the superposition of time-averaged wakes, obtained using the SOWFA tool. In this regard, a novel method is proposed which is able to achieve better performance than the standard superposition schemes. Then, in order to synthesize a ROM, the power outputs of the superposed turbines should be predicted. In doing so, the thesis adopts two distinct procedures, namely the same superposition framework mentioned before and the FLORIS code. Finally, alternative approaches for the generation of the projector are investigated with the goal of lowering the computational burden demanded by the proper orthogonal decomposition (POD) technique, widely employed. Albeit the present research was carried on to promote the development of a wake steering farm control with a yaw-actuation, all the findings and the designed framework may be readily extended to other plant control strategies.

Acknowledgements

Giunti alla fine di questo lungo, faticoso ma appagante percorso desidero ringraziare tutte le persone che ne hanno preso parte per un tratto più o meno breve.

In primis, la mia riconoscenza va al mio relatore, il prof. Stefano Cacciola, per la disponibilità, la professionalità ed il supporto mostrato, nonostante i km di distanza.

In secondo luogo, desidero ringraziare il TUM Wind Energy Institute, presso cui la tesi è stata svolta, e tutte le persone che ne fanno parte, per avermi accolto. Un grazie particolare va a Filippo Campagnolo ed anche al prof. Carlo Bottasso, per i loro innumerevoli consigli che hanno permesso lo sviluppo di questa tesi.

In seguito, voglio ringraziare tutti i compagni di corso e amici incontrati durante il mio percorso accademico a Milano, Torino e Monaco: senza di loro non sarei certamente riuscito a sopravvivere a tutte queste avventure. Nello specifico, sono particolarmente grato ai miei compagni di ASP Alessandro, Domenico, Marco e Nicola, che mi hanno reso un miglior studente di ingegneria ma, soprattutto, una persona migliore.

Grazie poi ai miei amici storici che sono stati al mio fianco negli ultimi dieci anni e più. Sareste davvero troppi per elencarvi tutti e sono fiero di aver condiviso tutti i miei traguardi con voi.

Un grande e sentito ringraziamento va alla mia famiglia: ai miei genitori Dario ed Elena, a mia sorella Ilaria, a mio zio Carlo ed a tutti i miei parenti, per avermi sostenuto ed incoraggiato durante i miei studi.

Infine, un enorme grazie a Marika - i miei dolci tre quarti - che mi ha supportato e sopportato durante praticamente tutti questi cinque anni. Grazie per essermi stata vicina nei momenti difficili e per stimolarmi a migliorare continuamente nella vita quotidiana.

Grazie.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Wind Farm Control

The role of renewable energy sources (RES) has been regularly increasing during the last decades, under the constant push of new regulatory policies and a always lower levelized cost of energy (LCOE). In Europe, the targets set by the European Commission in 2014 and revised in 2018 aim for at least a 32% share of RES in the whole energy market for 2030 (European Commission, 2018). Clearly, wind energy plays a key role within this scenario, as nowadays more and more power can be produced with this renewable source, as depicted in Figure 1.1.

Often wind turbines are installed in clusters, known as wind power plants or wind farms, taking advantage of economies of scale to lower the LCOE. While this may be beneficial in economic terms, curtailing construction, maintenance and commissioning costs, aerodynamic wake interactions among the turbines come into being, affecting significantly the cluster performances. Indeed, the wake shed by the upstream turbines leads to lower wind speed and higher turbulence intensity impinging on the waked machines, reducing the net power production and increasing fatigue loads on the machines (Ahmad et al., 2019). Though the industry best practice prescribes to install turbines with spacing along the prevailing wind directions higher than those in the non-prevailing wind directions, it is evident that a more sophisticated way to manage the farm performance is sought.

In this regard, strategies for the coordinated control of wind power plants (termed *wind farm control*) appears to be a compelling opportunity to enhance the cluster energy production. Currently, farms are controlled in a greedy manner, according to which every turbine is individually optimized, neglecting the effects on the waked machines. Within a coordinated control strategy, each wind turbine would cooperate with the others to increase the net production; in other words, even though the turbines may work in individual sub-optimal conditions a benefit for the whole plant would be obtained. In the literature, different approaches have been proposed to implement wind farm control. Independently from the techniques used, all of them can be divided into quasi-static open-loop control, which may work adequately for slowly-varying wind conditions, and closed-loop control, that instead ensures a dynamic behaviour with faster response and a lower sensitivity to imperfect modelling assumptions. Albeit the latter scheme may be costlier and more complex, it is evident that its features make it particularly interesting for real-world engineering applications. As evidence of this trend, the European Union has funded since the end of 2016 the CL-Windcon project, precisely to assess the margins of closed-loop wind farm control.

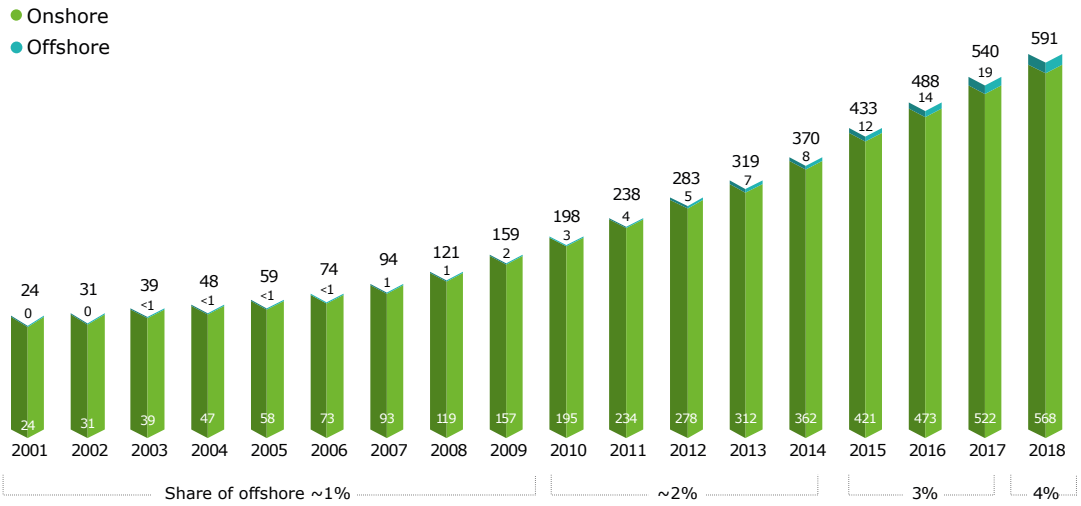
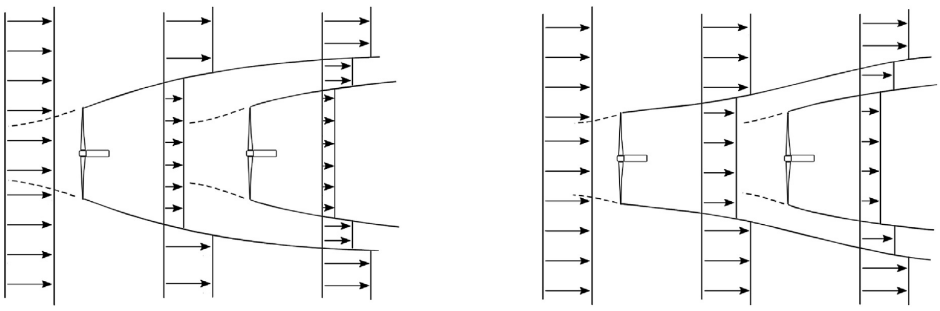


FIGURE 1.1: *Worldwide installed capacity of wind energy solutions (data in* GW*). Taken from (Ohlenforst et al., 2019).*

The first strategy that allows to control wakes in a farm layout (Active Wake Control, AWC) consists in the axial induction control (AIC). The main idea behind this strategy is that by properly regulating the axial induction factor of the upstream turbines the downstream wake deficit may be reduced, allowing an enhanced power production for the waked turbines. This scenario is depicted in Figure 1.2 and the down-regulation on the upstream machine may be achieved either increasing the blade pitch angle or adopting a sub-optimal tip-speed ratio. Evidently, the AIC



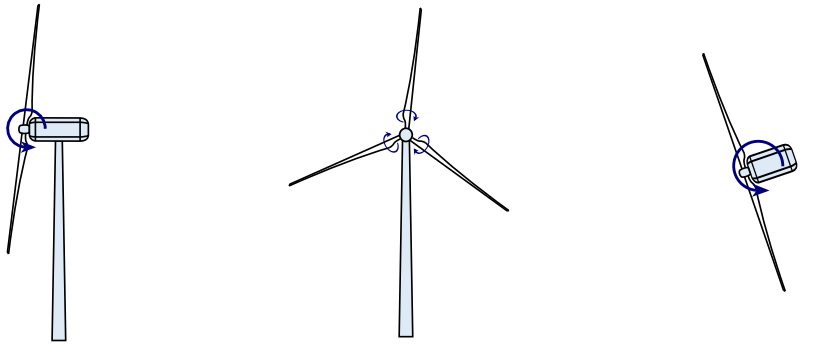
(a) *Without AIC (baseline case).* (b) *With AIC.*

FIGURE 1.2: *Velocity distribution without (a) and with (b) axial induction control. Taken from (Corten and Schaak, 2003).*

becomes appealing only when the reduced power production of the first turbines is compensated by the increase in production of the downstream machines (and one should even consider the cost of control in the overall power balance). Still, it should be underlined that the AIC advantages depends on several factors, as the atmospheric conditions, the wind direction and the wake characteristics, as reviewed by Boersma et al. (2017). Furthermore, beyond examining the velocity distribution, one should even consider the behaviour of the downwind turbulence intensity. The variation of the axial induction factor does lead to an increased velocity immediately after the turbine but it even reduces the turbulent mixing of the wake and, ultimately, the recovery rate. Hence, two counteracting effects would take place and the increase in production may be lower than expected. Overall, the effectiveness of

the AIC strategy has been thoroughly discussed in the literature, without reaching a clear agreement on the matter (Boersma et al., 2017). The recent work of Hoek et al. (2019) tested on-field the implementation of the AIC strategy on a commercial wind farm, eventually estimating an actual energy increase of 0.37%.

Another method to achieve AWC is by means of wake redirection techniques, so that the wake would be deflected downstream and the shading on the downwind turbines would be reduced. Boersma et al. (2017) listed tilt actuation, individual pitch control (IPC) and yaw actuation as possible ways to redirecting the wake. All these different strategies are sketched in Figure 1.3. The effects of the three redi-



(a) *Tilt-actuation.* (b) *IPC strategy.* (c) *Yaw actuation.*

FIGURE 1.3: *Wake redirection techniques.*

rection techniques has been numerically studied in-depth by Fleming et al. (2014). A tilt-based approach results quite effective for wake redirection (especially in the vertical direction) without significant consequences on the machine loads. Still, a tilt-actuation is far from being achieved in current standard turbines and it could even lead to problem for the tower-blades clearance; for instance, the adoption of large negative tilt angles (with the same convention in Fig.1.3a) would be prevented for upwind machines. On the other hand, the IPC could be already implemented on state-of-the-art wind turbines and it would even redirect effectively the wake, though with large increase in loads. Lastly, Fleming et al. (2014) investigated the adoption of a yaw steering technique, finding promising results as a substantial wake redirection and a load alleviation have been simultaneously achieved. Indeed, starting from these considerations, yaw-actuation has drawn the attention of the wind energy community. In recent years a plethora of works has focused on this method, either to get a better understanding dynamic of the curled wake dynamic (Bay et al., 2019; Annoni et al., 2018a) or to examine its implementation as a control strategy for wind power plants, developing accurate and low costs models (Fortes-Plaza et al., 2018) and optimization frameworks for real-time applications (Annoni et al., 2018b).

Lastly, other methods to increase the power production of wind farms have been lately discussed, as the Active Wake Mixing (AWM) technique. As its name suggests, the AWM aims at enhancing the recovery rate of the turbine wakes, such that shaded machines would feel higher inflow wind speeds. This strategy is carried on modulating the rotor thrust cyclically, for instance by means of a periodic collective motion of the blade pitches. Croce et al. (2019) inspected the effect of AWM on a single turbine, discovering a partial increase of the loads, notably when higher pitch amplitudes (more than 2°) were employed.

Clearly, the development of control methods is fostered by a suitable modelling of the wind power plant. Farm models are classically divided into two classes. The first one of those consists in engineering models, which aim at describing only the macroscopic features of the farm (as the velocity distribution) and due to their inherently low computational costs are widely exploited in the currently available commercial software. The other class is represented by models based on computational fluid dynamics (CFD) data, ensuring greater accuracy at the expense of larger computational resources. While engineering models may neglect important aspects of the wind farm dynamics, the latter category is evidently impractical for real-time usage. Hence, for wind farm modelling methods able to provide good accuracy with affordable computational costs are much sought.

Amidst different alternatives, Fortes-Plaza et al. (2018) developed a reduced-order model (ROM) compressing the huge amount of CFD data into a more manageable discrete-time linear time invariant system with nearly 20 states. Overall, the resulting ROM showed good performances when applied to a distinct validation dataset or either to wind-tunnel measurements. Being able to cast a model in the standard state-space form consists in an obvious advantage, as it allows the design of a controller exploiting all the practices and the know-how of the classical control engineering.

1.2 Objectives

The present work has been initiated as a natural continuation of the findings of Fortes-Plaza et al. (2018) and trying to overcome some limits of the latter. The ROM obtained by the authors provided acceptable results when used in operating conditions slightly different from those of the training dataset (i.e., modifying the spacing of the turbines), even thanks to the implementation of a Kalman filter. Yet, it is clear that changing significantly the working conditions - such as using a different inflow or even adding a new turbine - would affect the model behaviour, yielding inaccurate results. The simplest way to model a farm with newly modified parameters would certainly be the extraction of an additional ROM, but this would mean further computational resources being spent. Intuitively, this procedure lacks of flexibility and a more robust scheme is sought.

Essentially, this thesis is based on the following question:

> *Is it possible to exploit the knowledge of the flow behaviour around a single turbine to model the dynamic of a chosen farm?*

With an affirmative answer one could simulate the flow around a turbine by means of CFD methods and then suitably assemble the data together, eventually modelling the farm behaviour. In this way, costly simulations of entire power plants would be avoided and substituted by less demanding simulations run on individual turbines. Furthermore, this framework would achieve a better flexibility, relaxing the geometrical assumptions on the farm layout. Namely, after a proper way to assemble the data of single turbines has been found, one could reconstruct the behaviour of a farm for, virtually, whatever spacing and number of machines in the array. Notice how this approach would still require to run different simulations for distinct meteorological conditions, but all of them could be performed offline and once for all.

One may think to assemble the single turbine simulations exploiting the previous knowledge of the ROM states, handling a much more manageable system than the

original one. This would directly translate into operating with the modes used for the projection of the initial dataset onto a lower-order subspace (Fortes-Plaza et al., 2018). However, this may represent a risky operation mainly due to two reasons:

- The physical meaning of the assembling procedure would be lost
- The modes extracted by exactly the same dataset may differ because of the numerical algorithms involved or, more trivially, because of an opposite sign.

Rather, as a first step towards the generation of a farm model, it was chosen to adopt an approach focused on the physical features of the flow. While future developments for the handling of the modes may occur, within this procedure all the previous knowledge of the wake dynamic could be utilized.

The needs for a physic-preserving method basically translates into the adoption of a velocity superposition scheme: starting from the knowledge of the velocity distribution around a turbine given by the CFD data, the entire flow field in a wind farm is generated and then a ROM is extracted.

Hence, the present thesis aims at deriving a wind farm ROM by means of finding a proper method to superpose the velocity fields and, incidentally, a technique to estimate the power production of the machines in the plant. Indeed, the latter aspect consists in a mandatory requirement for the extraction of a reduced model and because the outputs of the wind turbines cannot be assessed by a CFD simulation of the farm (which one should avoid) an alternative scheme to predict the power production is investigated. Furthermore, this study even explores the possibility of generating a projector in additional and less demanding ways than those employed in the literature (Annoni, 2016; Fortes-Plaza et al., 2018).

This work shows the following innovative aspects:

i) A systematic approach for the superposition of CFD data is hereby introduced

ii) With regard to (i), a superposition scheme different from the standard ones used in engineering models is proposed

iii) A novel domain decomposition procedure is applied for the generation of a projector matrix for ROM accounting for input-output relations.

In fact, while the superposition of wakes is currently implemented in commercial software, this procedure has been only employed with engineering wake models that show inherently distinct features from CFD data and no work has focused consistently on the latter. As a matter of fact, only Bossuyt (2018) has partly addressed the opportunity of merging high-fidelity CFD wakes together, without introducing however a methodical framework for the purpose.

The whole research would study two- and three-turbines clusters. Whereas this number is clearly lower than the number of machines included in actual wind farms, it is evidently a useful for a starting analysis of the effectiveness and feasibility of the proposed framework. In addition, Annoni et al. (2018b) discussed how a large wind-farm could maximize its performance iteratively finding optimal working conditions for smaller subsets of machines, whose size ranged from 1 to 5 machines in maximum waked conditions, i.e. turbine arrays similar to the ones hereby examined.

1.3 Thesis Outline

In this thesis, firstly the concept of wind farm modelling would be introduced in Chapter 2. Indeed, the wake dynamics around an individual turbine and for a entire farm is discussed, updated with the current state-of-the-art findings. Hence, the remaining part of the section would be devoted to consider how the fluid behaviour is numerically modelled, with particular regard to the tools adopted in this thesis.

Then, Chapter 3 gives an overview of the requirements for the generation of a superposed ROM. In particular, after defining the different methods for the extraction of a reduced-order model and emphasizing their advantages and disadvantages, all the aspects necessary for the modelling would be touched in-depth.

The detailed aspects of the numerical simulations are reviewed in Chapter 4. The first part of the section would list down the turbulence models employed in the present study and next the simulation setup is investigated. The latter would chiefly address the adopted turbine model and the features of the spatiotemporal domain.

Afterwards, the results of the present thesis are exhibited in Chapter 5, where the outcomes of the different research directions followed would be thoroughly examined.

Lastly, Chapter 6 would end the document, reporting the conclusions and recommendations for future works.

CHAPTER 2

Wind Farm Modelling

This chapter provides the background information needed to understand how wind farm modelling works. In Section 2.1 the physics of the flow is introduced, analysing the wake regions behind a turbine and their interactions in a wind farm layout. Then, Section 2.2 focuses on the numerical modelling of the problem, briefly introducing the methods that can be generally used and discussing more in-depth the tools adopted in the thesis.

Where not otherwise specified, the coordinate reference system is such that:

- positive x describes locations downstream the turbine, with the hub located at $x = 0$
- z is the vertical coordinate of a point with respect to ground and with z_h as hub height
- y is the lateral coordinate forming a right-handed coordinate system.

Figure 2.1 shows a schematic representation of this reference frame.

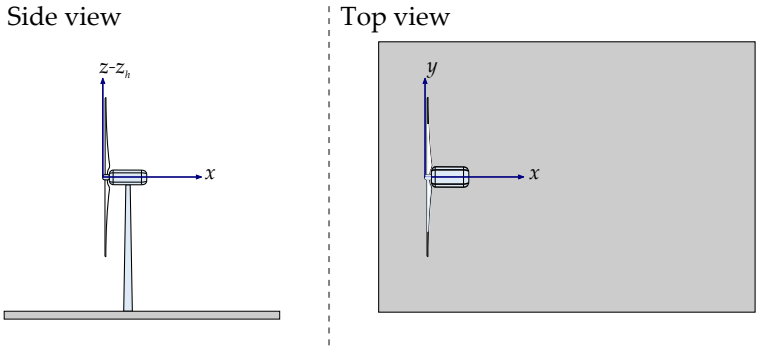


FIGURE 2.1: *Wind turbine reference frame.*

2.1 Wind Turbines Wake Dynamics

2.1.1 Flow Around a Wind Turbine

Wind turbines aim at extracting energy from the wind and, due to this interaction, both the upstream and downstream incoming airflow is altered.

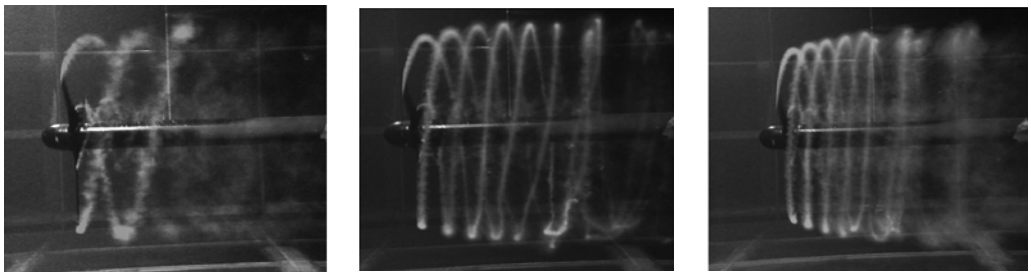
Even considering the one-dimensional momentum theory, written nearly 100 years ago by Betz (1926) and underlying several assumptions, one can understand

some of the features of the flow dynamic. Indeed, as the turbine is operating, extracting energy from the wind, due to the momentum conservation a wake region is generated. Generally, the latter is further divided in two parts: the *near-wake*, immediately after the turbine, with an extension of 2 to 4 rotor diameters; and the *far-wake*, downstream the previous one.

The Near-Wake

Due to the proximity to the machine, the near-wake is highly influenced by the geometry of the wind turbine. Several studies, among which those of Porté-Agel, Bastankhah, and Shamsoddin (2019) and Abraham, Dasari, and Hong (2019), showed the presence of coherent structures in this region.

Amidst these structures, tip and root vortices have been extensively examined in the literature: originated due to the pressure difference at the two sides of the blade (Porté-Agel et al., 2019), these structures have a helical shape. Based on a POD analysis, the study of Debnath et al. (2017) displayed how their dominant frequency is three-times the rotor rotational frequency for a three-blades turbine. While the pitch of the tip helical vortex is greater than that of the root vortex (Sherry et al., 2013), both decrease with the TSR (the tip-speed ratio, also indicated with λ), as well depicted in Figure 2.2 from the analysis of Okulov et al. (2014).



(a) $\lambda = 4$ (b) $\lambda = 6$ (c) $\lambda = 8$

FIGURE 2.2: *Tip and root vortices for low (a), near-optimal (b) and high (c) values of TSR. Taken from: (Okulov et al., 2014).*

Another helical vortical structure, the hub vortex, arises in the central part of the wake. This results in a periodic oscillating motion, that can be characterized by the Strouhal adimensional number $St = {}^{fL}/_{U_{hub}}$, where f is the shedding frequency under analysis, L a reference length and U_{hub} the streamwise velocity at hub height. So far, no agreement has been reached between several numerical and experimental studies on the St of the hub vortex, both employing the nacelle dimension and the rotor diameter D as reference length (Abraham et al., 2019; Howard et al., 2015). Using the latter $(St = {}^{fD}/_{U_{hub}})$ Porté-Agel et al. (2019) obtained the rather large range of Strouhal numbers 0.12 to 0.85 reviewing several works on the subject.

Abraham et al. (2019) discussed also the role of vortex shedding arising from the impingement of the airflow into the tower. Defining the Strouhal number with the freestream velocity and the diameter of the cylindrical tower for the reference length, they obtained $St = 0.2$, as expected from a flow past a bluff body.

Figure 2.3 depicts the coherent structures on a real-scale turbine, investigated with a peculiar Super Large-scale Particle Image Velocimetry (SLPIV) using natural snowfall (Abraham et al., 2019). While the region above hub height exhibits somehow regular tip and root vortices, the bottom part of the domain is characterized by less coherency, due to the interaction of vortices shed by the rotor blades, the tower

and the hub. The latter is a mechanism that influences greatly the overall wake recovery, as explained by Debnath et al. (2017): the interaction induces a breakdown of the tip vortices, promoting entrainment of the surrounding flow and, thus, recovery of the wake. This emphasizes the importance of including the tower in simulations to accurately predict the recovery. Analysing the mean flow in the near-wake,

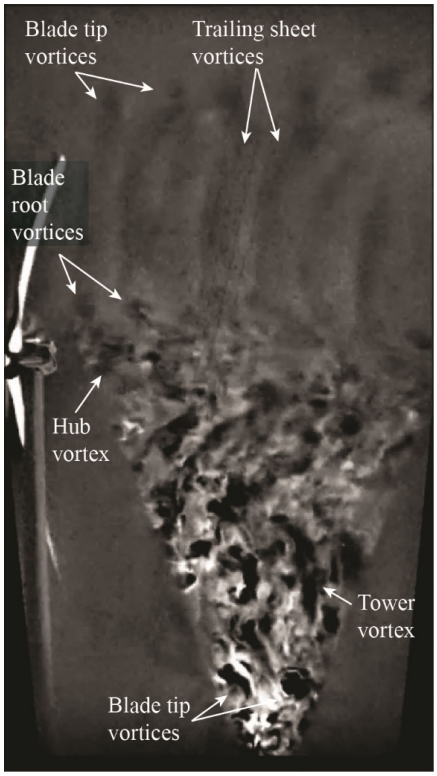
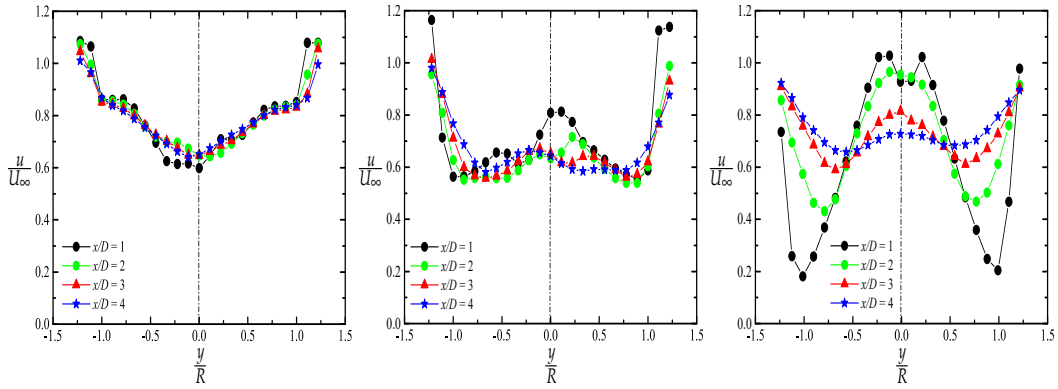


FIGURE 2.3: *Coherent structures behind a rotor (Abraham et al., 2019).*

a strong swirl component is present at the center of the wake, rotating in the opposite direction of the turbine blades, as expected by the conservation of angular momentum, and decaying travelling downstream. The streamwise velocity distribution shows a dependency on the TSR, as Krogstad and Adaramola (2012) stated; Figure 2.4 reproduces the velocity distribution along the spanwise coordinate at different downstream locations, all measured at hub height. Whereas for all the TSR the wake width increases downwind, in the partly stalled regime (2.4a) the defect is seen to remain unaltered propagating downstream, for higher TSR a speed-up region appears in the wake center. At optimal condition (2.4b) the speed-up is moderate and well discernible only at the first measurement station, allowing to consider the profile almost as uniform until the midsection of the wake (since the entire blade operates in the most efficient way); close to the runaway point (2.4c) the large TSR induces a much more noticeable overspeed and the profile assumes a double Gaussian shape, smeared out with downstream distance. Furthermore, Figures 2.4b and 2.4c highlight how the sharp gradients at the tips decrease for increasing x/D due to the entrainment of outer flow.



(a) $\lambda = 3.35$ (b) $\lambda = 5.79$ (c) $\lambda = 9.15$

FIGURE 2.4: *Velocity distribution at four near wake locations for low (a), optimal (b) and high (c) values of TSR. Taken from: (Krogstad and Adaramola, 2012).*

The Far-Wake

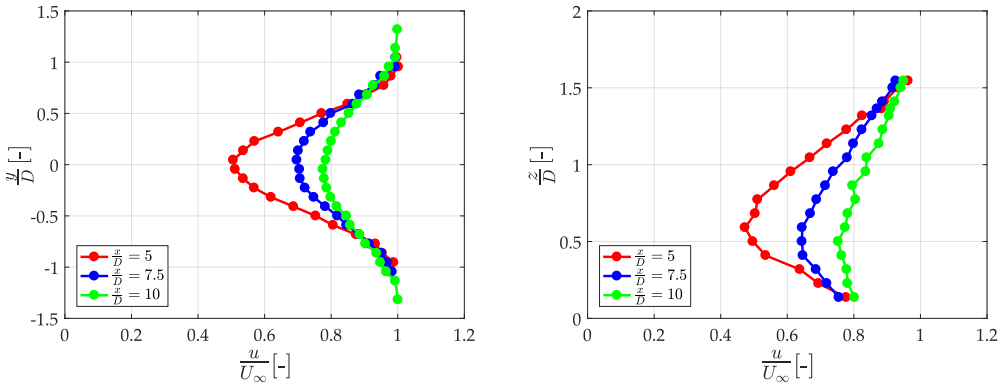
As mentioned before, the far-wake is considered to start approximately $2D$ to $4D$ downstream the turbine and the actual length of the previous near-wake is influenced by the incoming flow turbulence intensity, the TSR value and the mechanical shear induced by the wind turbine (Porté-Agel et al., 2019). However, it is rather difficult to separate the wake in two isolated regions, as the distinction between those is usually not clear. The far-wake is characterized by a large-scale oscillating motion, the *meandering*, and Howard et al. (2015) discussed how this feature was observed starting as early as $2D$ downstream, while the hub vortex signature was felt up to $4D$ after the turbine; namely, an overlapping between the two regions was detected.

The onset of the aforementioned meandering motion is still not well understood: Mao and Sørensen (2018) cited how this phenomenon could arise from the interaction of hub, root and tip vortices (evidence endorsed by Howard et al., 2015) or it can be either due to large-scale eddies in the free-stream. In the latter case, while the small eddies in the flow would be responsible for the wake recovery, larger vortical structures would advect periodically the mean wind field downstream (Larsen et al., 2008). Moreover the analysis of España et al. (2011) revealed that meandering motion was detected only when eddies larger than the rotor diameter were present in the incoming flow, supporting the second hypothesis stated.

In general, wake meandering appears to be influenced by incoming flow features but not from turbine operating conditions (Porté-Agel et al., 2019). This indeed is the main characteristic of the far-wake: as it moves downstream, the far-wake slowly "forgets" about the rotor, exhibiting a self-similar behaviour.

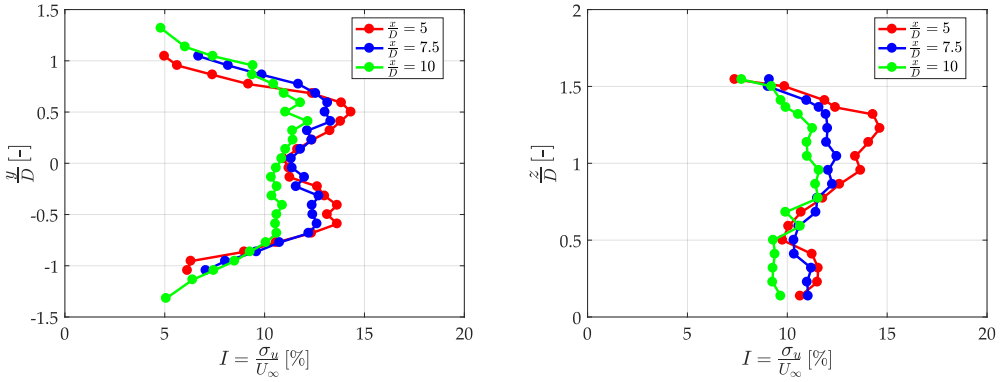
Figure 2.5 portrays the velocity distribution in the spanwise (a) and vertical (b) direction for three different locations, with measurements taken at the wind tunnel of Politecnico di Milano (GVPM). From the former is noticeable the Gaussian shape of the profile and that the wake width increases downstream (due to flow entrainment), while the defect progressively reduces. The wake recovery is still recognizable in the velocity distribution in the z direction but no symmetry is achieved due to the ground effect. The streamwise turbulence intensity $I = \sigma_u/U_\infty$ obtained from the same dataset, reveals a different shape in the spanwise coordinate (Fig.2.6a), as two peaks are registered at the edge of the wake. The plot along the z dimension in Figure 2.6b displays instead an asymmetric distribution, again due to the ground

which suppresses the turbulence. Porté-Agel et al. (2019) described how the maximum of I is expected at the top-tip position at 2 to 4 diameters downstream, exactly at the transition from near- to far-wake.



(a) *Velocity distribution at hub height.* (b) *Velocity distribution at $y = 0$.*

FIGURE 2.5: *Velocity distribution in the far-wake measured at an horizontal plane (a) and at a vertical one (b) at GVPM.*



(a) *I distribution at hub height.* (b) *I distribution at $y = 0$.*

FIGURE 2.6: *Turbulence intensity distribution in the far-wake measured at an horizontal plane (a) and at a vertical one (b) at GVPM.*

A glimpse to the entire wake and the role of γ, I and the ABL

Merging all the previously stated notions, we could obtain a picture of the flow field similar to the one in Figure 2.7. The top part underlines the role of turbulent eddies in the instantaneous snapshot of the flow, whereas in the bottom one a smoother time-averaged wake is portrayed. Notice also the presence of a turbulent boundary layer in the inflow.

Nevertheless, other variables may change the wake behaviour. Firstly, the effect of turbine misalignment with respect to the wind was not discussed. Yawing a turbine slightly affects the flow features for small yaw angles (for instance, less than 10°), but introduces major modifications with greater misalignment (e.g., more than 20°) (Porté-Agel et al., 2019). Basically, as the wind turbine is yawed out of the wind, the wake deflects, due to the conservation of momentum. Krogstad and Adaramola (2012) studied the velocity distribution under the effect of yaw in the

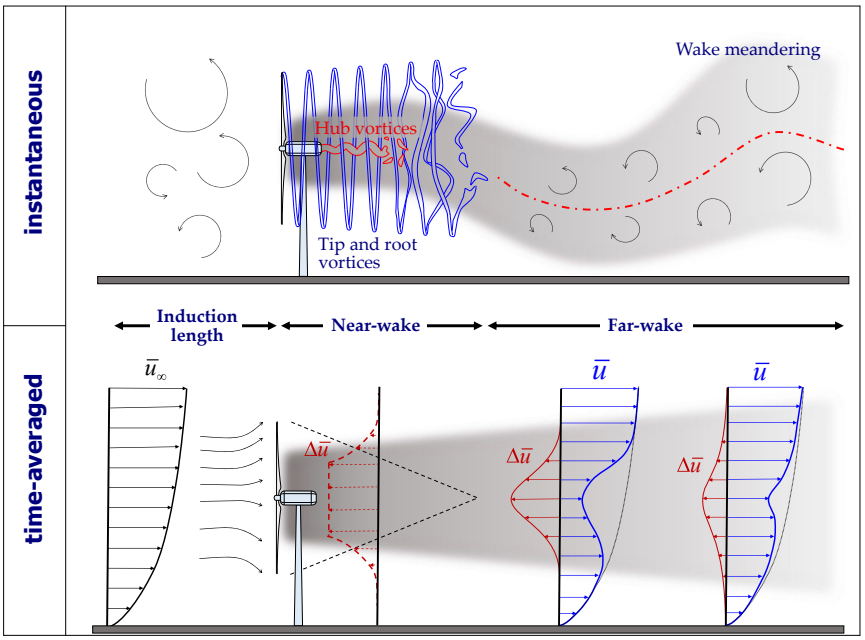
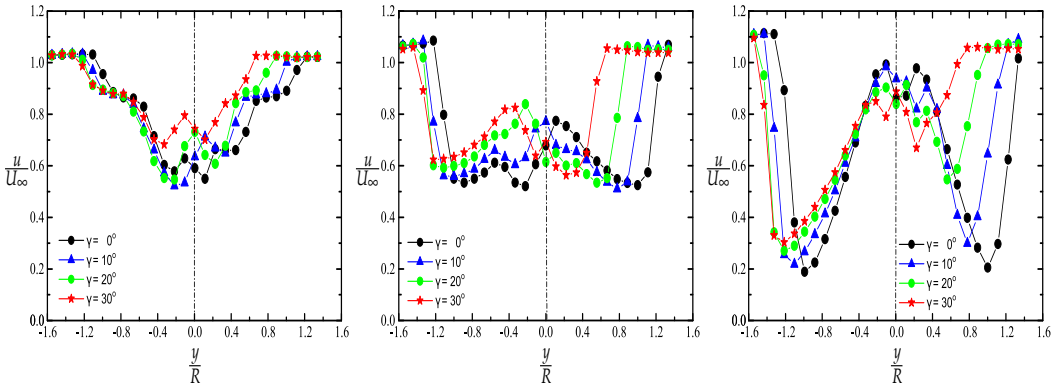


FIGURE 2.7: *Instantaneous (top) and time-averaged (bottom) flow field around a turbine with an incoming turbulent boundary layer. Adapted from: (Porté-Agel et al., 2019).*

near-wake ($x/D = 1$) at hub height for three values of TSR (Figure 2.8). It turned out that the wake width is reduced for increasing value of the yaw angle γ, as a smaller rotor swept area is seen by the incoming flow; this, in turn, causes a reduction in the extracted energy. The yaw clearly introduces an asymmetry in the profile (in a linear manner up to 20°) but it even influences the recovery: for high value of γ a smaller defect is observed, due to a greater entrainment from the surrounding. The entrainment shows as well an asymmetric behaviour, being more intense on the right side of the wake.



(a) $\lambda = 3.35$ (b) $\lambda = 5.79$ (c) $\lambda = 9.15$

FIGURE 2.8: *Velocity distribution in the near wake with varying yaw angles γ and low (a), optimal (b) and high (c) values of TSR. Taken from: (Krogstad and Adaramola, 2012).*

Another key parameter is the turbulence intensity of the incoming flow, I_0. As its value increases, the wake recovers faster: the lifetime of the tip vortices is shortened

and flow entrainment is enhanced (Porté-Agel et al., 2019). Still, the effect of I_0 has been proved to be negligible up to $2D$ downstream (Krogstad and Adaramola, 2012).

Finally, the wake behaviour is dependent on thermal stability of the Atmospheric Boundary Layer (ABL). To some extent its role can be related with that of the turbulence intensity of the incoming flow. Indeed, an unstable boundary layer is characterized by a lapse rate such that:

$$\frac{dT}{dz} < \left(\frac{dT}{dz}\right)\Big|_{adiabatic} \approx -\frac{1°\text{C}}{100\text{m}},$$

where T is the air temperature, that promotes the formation of convective turbulent cells. Hence, the latter structures would lead to an increment of turbulent mixing, wake recovery and wake meandering (Porté-Agel et al., 2019). On the other hand a stable boundary layer (SBL) would hinder a quick recovery, increasing wind shear due to thermal stratification. Usually, while SBL is typical of night condition (because of the cooling of the ground), the occurrence of an unstable - or convective - boundary layer (CBL) is recurrent during daytime. In the neutral boundary layer (NBL) condition the actual thermal gradient coincides with the adiabatic, but a similar situation has never been actually observed in the atmosphere (Allaerts and Meyers, 2015). Rather, the literature focuses on a conventionally-neutral boundary layer (CNBL), i.e. a neutral ABL developed inside a stably stratified fluid. Figure 2.9 depicts velocity and potential temperature profiles in a CNBL. The latter propriety is defined as the temperature θ of a particle brought adiabatically to a reference pressure p_0 from its initial conditions (p, T), namely for an ideal gas:

$$\theta = T\left(\frac{p_0}{p}\right)^{\frac{\mathcal{R}}{C_p}},$$

where $\mathcal{R}$ and C_P are the specific gas constant and the specific heat at constant pressure, respectively. It goes without saying that for a NBL the potential thermal gradient is null by definition. Looking at Figure 2.9, a steep inversion of θ is evident in the so-called *capping inversion*; as this inversion gets greater, higher negative buoyancy forces decelerate turbulent gusts from below, preventing turbulent entrainment and, hence, a thickening of the ABL.

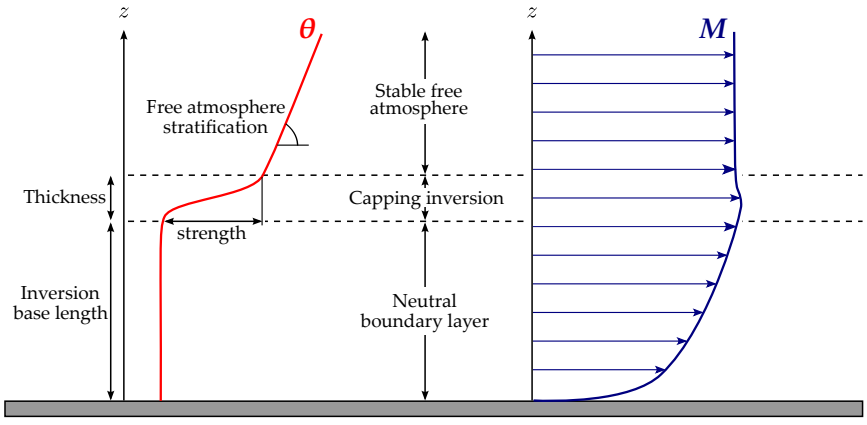


FIGURE 2.9: *Potential temperature (θ) and mean velocity (M) distributions in a conventionally-neutral boundary layer.*

2.1.2 Wake Interactions in a Wind Farm

As more turbines are put closer in a wind farm layout, the flow field becomes more complex. First of all, the whole cluster induces a blockage effect that leads to an upward and lateral ABL deflection, as represented in the induction region in Fig. 2.10. Then, when the first turbines are encountered an internal boundary layer (IBL) develops, due to the interactions of their wakes (entrance and development region); if the farm is large enough, the IBL reaches the ABL, starting to entrain flow from the outer atmosphere. According to this mechanism, momentum is brought inside the IBL, balancing the loss caused by the power extraction operated by the turbines: an equilibrium condition is reached in the fully-developed region. Towards the end of the wind farm, if the free atmosphere is strongly stratified, an acceleration of the flow is observed (exit region) after which the wake of the entire farm propagates downwards for several kilometers (Porté-Agel et al., 2019).

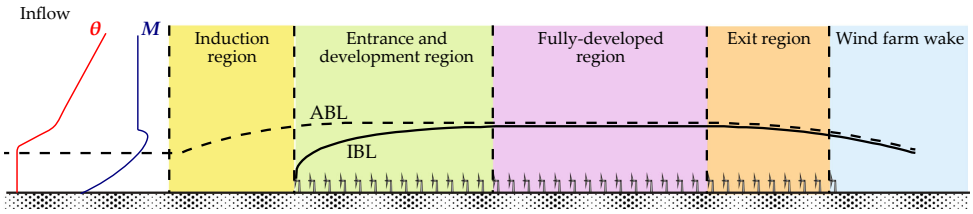


FIGURE 2.10: *Flow regions due to the interaction between a wind farm and a conventionally-neutral boundary layer with a strongly stratified atmosphere. Adapted from: (Porté-Agel et al., 2019).*

The mechanism governing the interaction of wakes is highly nonlinear and still not completely clear. Figure 2.11 shows the layout of Lillgrund, a Swedish offshore wind farm, whose power output under an inflow with $U = 9 \pm 0.5\text{m}\,\text{s}^{-1}$ and $I_0 = 6\%$ are plotted in Figure 2.12. Firstly, it is evident how a fully-developed flow is already reached around the fourth machine for the rows with no missing turbines, whereas the lack of machines in the array visibly delays the onset of the region. Furthermore, the figure highlights how, with an aligned incoming wind, the maximum deficit is felt by the second turbine, while the following ones suffer smaller losses.

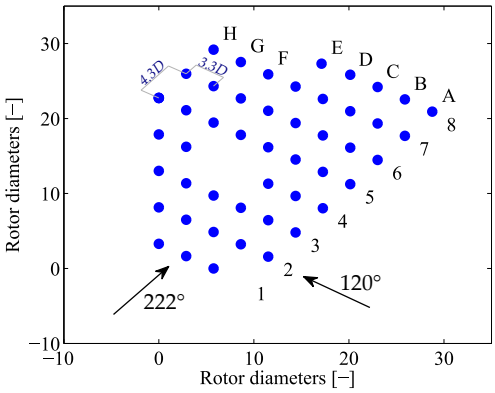


FIGURE 2.11: *Layout of the Lillgrund offshore wind farm.*

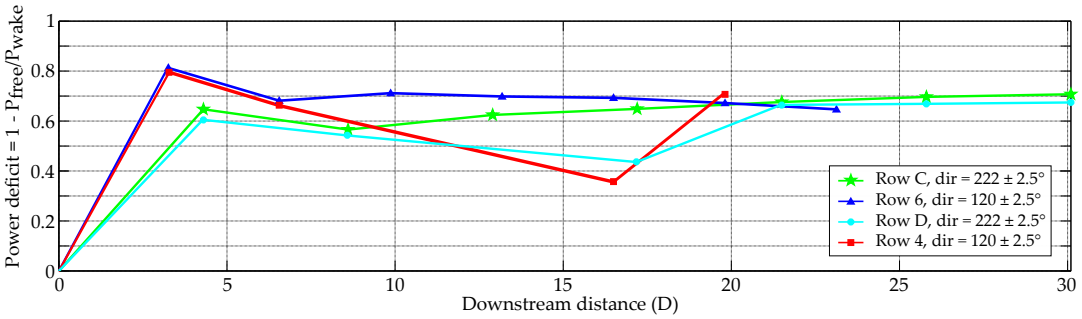


FIGURE 2.12: *Power deficit at different rows of the Lillgrund wind farm. Data from: (Göçmen et al., 2016) .*

Gunn et al. (2016) noticed how after two inline rotors the wake recovered faster due to the higher turbulent intensity, in agreement with the just reported experimental data. Yet, when the rotor were laterally displaced the situation qualitatively changed. With a $1D$ lateral spacing, the authors found a slow recovery, postulating that this happened because less energy was available between the two wakes and only the side of the wake shared with the outer flow was interested by an effective entrainment. Further displacing the rotors ($1.5D$ lateral spacing) the interaction between the wakes got limited as expected and the recovery rate was almost the same of the single turbine one.

2.2 Numerical tools

Being able to numerically model the flow in a wind farm is key to understand the interactions among turbines and, even most importantly, to predict power output and loads. Moreover, this can also allow parametric studies on the plants, namely analysing how the variation of a variable (e.g., changing the yaw angle) modify the overall performance.

The numerical methods used are classically divided into:

- Low-fidelity, describing only macroscopic features of the flow. Their accuracy is rather low, but they are computationally affordable.
- High-fidelity, that aims at modelling accurately the physics of the problem, at the expenses of computational cost.

Low-fidelity models assume an analytical expression for the wake, usually focusing on the velocity profile. The latter can be eventually coupled with a simple model for the turbulence intensity. Clearly the dynamics of the flow around the turbines is oversimplified and the vortical structures are missing.

In addition, some methods can be defined as mid-fidelity, as their performance and accuracy are a compromise between the two previous classes. Amongst those, the dynamic wake meandering (DWM) describes the wake as a passive tracer advected by the large eddies in the incoming turbulent flow (Göçmen et al., 2016), in agreement with the previously cited studies. Further details can be found in (Annoni, 2016).

In the following the two numerical tools used in the thesis will be discussed.

2.2.1 FLORIS

The FLOw Redirection and Induction in Steady State (FLORIS, NREL, 2019a) is an open-source control-oriented tool developed by NREL and TU Delft. It models turbine interactions in a plant to perform real-time optimizations or either the integration of supervisory controls. To do so, different wake models and superposition methods (Sec.3.2.2) are implemented.

Most of the wake models, and those present in FLORIS too, tries to achieve good performances in the far-wake domain, as indeed the majority of turbine spacing in modern wind farms falls in this case (Porté-Agel et al., 2019). However, several new analytical wake models focusing on the near-wake are emerging nowadays and they can be easily added to the FLORIS framework. Nevertheless, the present study will stick to the most known models that should be accurate enough for the selected layouts, defined in Sec.4.2.1.

The Jensen wake model

This is one of the most popular model due to its simplicity and robustness. Introduced by Jensen, it assumes a wake with linear expansion and a constant deficit over the downstream cross section, originated due to the interaction of an incoming steady flow with an actuator disc turbine uniformly loaded.

The streamwise velocity in the wake can be defined as:

$$u = u(x, r; a) \coloneqq U_\infty(1 - \Delta u(x, r; a)) \tag{2.1}$$

where x is the streamwise (axial) position, r the radial coordinate and a the axial induction factor (hereby considered a constant parameter due to the model hypotheses). Applying the conservation of mass, Jensen found an expression for the wake deficit Δu:

$$\Delta u = 2a\left(\frac{D}{D + 2\kappa x}\right)^2 \tag{2.2}$$

Hence, it is introduced a decay coefficient for the wake, κ, whose value is actually dependent on the topography, on the incoming turbulence intensity and on the turbine operating conditions. According to the model, the higher κ the faster the wake recovery. Values around 0.04 have been suggested in the literature for κ for off-shore cases.

In Figure 2.13 is shown the streamwise velocity predicted by the Jensen model for a turbine with null yaw ($\kappa = 0.04$, $a \approx 1/3$ and $U_\infty = 5.5\text{m}\,\text{s}^{-1}$). The discontinuity at the boundaries is clearly nonphysical and, with a uniform velocity on the cross-section, the velocity is underestimated in the wake center and overestimated near the edges.

The Gaussian wake model

The Gaussian wake model is presented, in its most recent formulation, in the work of Bastankhah and Porté-Agel (2016). This model is based on the observation that

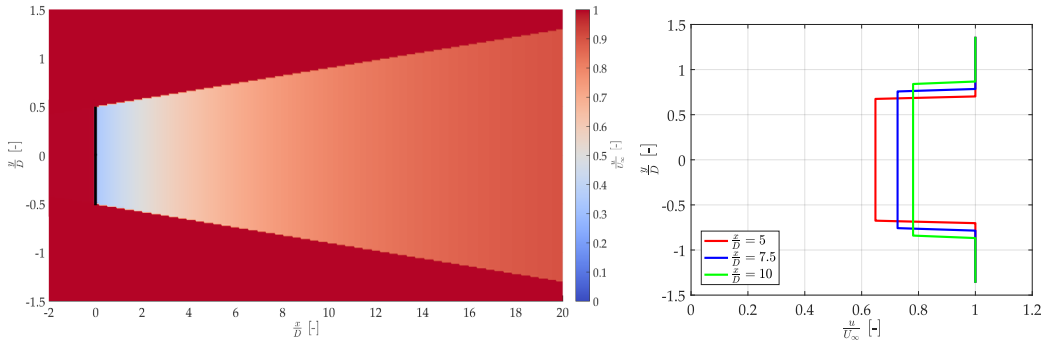


FIGURE 2.13: *Jensen wake at hub height in the whole plane (a) and sampled at 3 locations (b).*

after a certain downstream location self-similarity is reached, i.e. that:

$$\Delta u_c(x) := \max_{y^*} \Delta u(x, y^*) \tag{2.3}$$

$$\xi := \frac{y*}{\sigma(x)} \tag{2.4}$$

$$f(x, \xi) := \frac{\Delta u(x, y^*)}{\Delta u_c(x)} \text{ such that } f(x, \xi) = \tilde{f}(\xi) \tag{2.5}$$

So the velocity deficit Δu normalized by the maximum velocity deficit Δu_c and expressed as a function of y^*/σ collapse on a single curve, independently of the chosen x, where y^* represents the lateral distance from the wake center and σ the characteristic width of the wake.

The equations implemented in FLORIS which predict the far-wake for a non-yawed turbine are now reported. The null yaw angle, that is the actual operating condition studied with FLORIS in the present work, implies a null skew angle and a null wake deflection, leading to some simplifications. Eq.(2.6) is used to compute the velocity deficit in a 3D space, given the wake width in the y and z directions (Eq.2.7) and the potential core of the wake x_0 (Eq.2.9). The latter defines the position at which the far-wake starts and is a function of two tunable parameters, α^* and β^*, whereas the widths $\sigma_y = \sigma_z$ are dependent on two tunable coefficients describing the wake decay, k_a and k_b, as reported by Eq.(2.8).

$$\frac{\Delta u(x, y, z)}{U} = \left(1 - \sqrt{1 - \frac{C_T D^2}{8\,\sigma_y \sigma_z}}\right) \exp\left(-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)^2\right) \exp\left(-\frac{1}{2}\left(\frac{z - z_h}{\sigma_z}\right)^2\right) \tag{2.6}$$

$$\frac{\sigma_y(x)}{D} = \frac{\sigma_z(x)}{D} = k\,\frac{x - x_0}{D} + \frac{1}{\sqrt{8}} \tag{2.7}$$

$$k = k_a I_0 + k_b \tag{2.8}$$

$$\frac{x_0(I_0)}{D} = \frac{1 + \sqrt{1 - C_T}}{\sqrt{2}\left(\alpha^* I_0 + \beta^*\left(1 - \sqrt{1 - C_T}\right)\right)} \tag{2.9}$$

In Eq.(2.6) U stands for the incoming wind speed and z_h for the hub height.

Furthermore, these equations for the velocity are coupled with Eq.(2.10), proposed by Crespo and Hernandez (1996), that models the turbulence intensity added downstream a turbine, with 4 new parameters: (I_A, I_B, I_C, I_D).

$$I_+(I_0, x) = I_A a^{I_B} I_0^{I_C} \left(\frac{x}{D}\right)^{I_D} \tag{2.10}$$

After I_+ has been computed, the overall turbulence intensity can be evaluated as $I^2_{wake} = I_0^2 + I_+^2$.

For what concerns the near-wake, the deficit is computed with some minor modifications to Eq.(2.7) (NREL, 2019a).

The Gaussian model predicts a velocity distribution as the one in Figure 2.14, where the default tunable parameters in FLORIS were used ($k_a = 0.4, k_b = 0.004$ and $\alpha = 0.58, \beta = 0.077$) and $U_\infty = 5.5\text{m}\,\text{s}^{-1}, I_0 = 0.05, C_T = 0.89$.

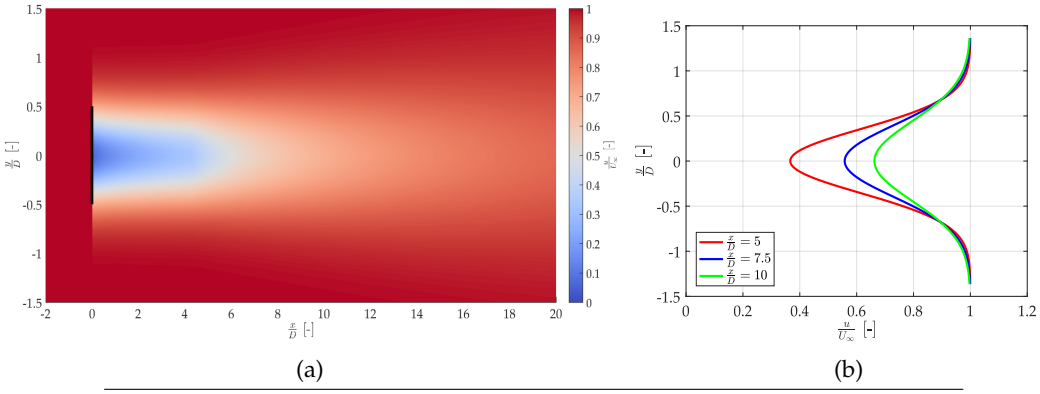


FIGURE 2.14: *Gaussian wake at hub height in the whole plane (a) and sampled at 3 locations (b).*

Overall the Gaussian model well predicts the wake behaviour and is extensively used in the literature to study wind farm performances (Annoni et al., 2018b; Annoni et al., 2018a; Bay et al., 2019). Yet, it still manifests some minor flaws.

Firstly, the equations may neglect some dependencies: Campagnolo et al. (2019) discussed for instance how expanding the decay rate to $k = k_a I_0 + k_b + k_c C_T$ would improve the model performance.

Secondly, even though the Gaussian model has after all a limited number of tunable parameters with respect to other wake models, a plethora of works describing significantly different coefficients exists and no universal set has been found. Indeed, the coefficients referenced by the literature have several validity ranges. The ones proposed by Crespo and Hernandez (1996) to model the turbulence intensity as in Eq.(2.10) were obtained empirically in the range of:

$$5 < \frac{x}{D} < 15, \;\; 0.07 < I_0 < 0.14 \quad \text{and} \quad 0.1 < a < 0.4\,.$$

Similarly, the default parameters for the wake decay rate (k_a, k_b) were interpolated by Niayifar and Porté-Agel (2015) in the range $0.065 < I_0 < 0.15$, whereas the default value of β was obtained from a theoretical comparison with laminar jet flows (Bastankhah and Porté-Agel, 2016). In a nutshell, a literature agreement is still far to be reached and, whenever possible, a previous tuning of the model should be performed, as done by Campagnolo et al. (2019).

This lack of robustness is indeed the major drawback of the analytical wake models, as trying to include new parameters in the set to better describe the physics

would further increase the number of tunable coefficients that should be found, making the process more cumbersome.

2.2.2 SOWFA

The other tool used is the Simulator fOr Wind Farm Applications (SOWFA), a high-fidelity numerical method that was developed at NREL. SOWFA is a CFD solver based on OpenFOAM coupled with FAST, an NREL's wind turbine simulator.

SOWFA studies wind turbines in the ABL, using an actuator line model coupled with FAST, which is an aeroelastic tool. In particular, the three-dimensional incompressible Navier-Stokes equations are solved relying on OpenFOAM, even accounting for the transport of potential temperature equations (considering buoyancy effect and Coriolis forces). According to the actuator line, the turbine blades are discretized as spanwise segments with constant airfoil shape, chord and twist. Then, given the incoming flow, the aerodynamic loads can be evaluated on each segment, obtaining the action of the wind on the blades; in turn, the turbine action on the wind can be easily derived and it is simulated and propagated downstream by OpenFOAM.

Wind turbine data can be collected thanks to FAST, that makes available at each time step the collection of quantities as the yaw position, the power output, the rotational speed, the thrust, the generator torque and so on. On the other hand, snapshots of the flow field can be taken at the desired contours/surfaces in the 3D discretized domain, thanks to OpenFOAM.

SOWFA has been already validated in several studies and even in works comparing experimental data collected at GVPM with numerical results obtained using a digital copy of the same wind tunnel (Wang et al., 2018a; Wang et al., 2018b; Wang et al., 2019). In general, its results are considered accurate enough for practical purposes. As it belongs to the high-fidelity class, this method is computationally demanding and simulations typically lasts for days even in supercomputer units, especially if wind farm with big size are studied.

CHAPTER 3

Generation of a superposed ROM via a POD approach

Currently, high-fidelity methods are the only choice to obtain an accurate representation of a wind power plant, but their computational complexity make them unsuitable for controller design; as a matter of fact, typical fluid dynamic problems embed millions of states. Hence, simpler and more flexible control methods are needed. In this perspective, a reduced-order model (ROM) constitutes a possible solution to the problem, lowering the computational cost while maintaining enough accuracy.

Section 3.1 presents some methods used in the literature to generate ROMs and Section 3.2 examines the procedure followed during to obtain a ROM through a superposition approach.

3.1 Reduced-Order Modelling Techniques

As stated before, ROMs allow a lower-dimensionality representation of the system. Generally, however, the purpose underlying the ROM generation may differ and distinct criteria may be selected for the choice of a proper technique. Yet, while the immediate goals of research can vary - e.g., a better understanding of the flow structures (Debnath et al., 2017; Pope, 2000) or either the modelling of a fluid dynamic problem - the ultimate goal at the basis of all engineering applications is control.

The present study partly traces the ones of Annoni (2016) and Fortes-Plaza et al. (2018), aiming at the control of a wind power plant, but no a priori assumption on the farm geometry is made. As a result, the sorting of the ROM methods would be based on similar criteria, namely:

- **Low computational cost**, reducing the huge number of states such that the resulting ROM would even be suitable for real-time applications
- **Input/Output handling**, as the constructed model should include controllable inputs and measurable outputs
- **Adjoint-free**, meaning that no model adjoint should be present
- **Linear parameter varying (LPV)**, so that the model could be used for nonlinear systems, whose dynamic change significantly for different operating conditions.

Specifically, the requirement of no model adjoint is needed to avoid the resolution of a new adjoint-problem (derived from the original one) as many times as the number of outputs. This indeed reduces the flexibility of the ROM technique, leaning towards an increase of computational cost.

From Sec.3.1.1 to 3.1.4 four techniques would be analysed in light of the selected criteria.

3.1.1 Balanced Truncation

Balanced truncation is a model reduction technique widely used in the field of control.

Eq.(3.1) defines the usual state-space representation of a time-invariant continuous-time linear system:

$$\begin{cases} \dot{x}(t) &= Ax(t) + B\mathsf{u}(t) \\ y(t) &= Cx(t) + D\mathsf{u}(t) \end{cases}, \tag{3.1}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state vector, $\mathsf{u}(t) \in \mathbb{R}^{n_u}$ is the input vector, $y(t) \in \mathbb{R}^{n_y}$ is the output vector and the state matrices A, B, C, D have the appropriate dimensions.

Given that, the definitions of observability and controllability can be recalled. A system is said to be controllable if, for any initial state $x(t_0) = x_0$, a suitable series of input $\mathsf{u}(t)$, with $t \in [t_0, t_1]$, allows the system to reach any other state $x(t_1) = x_1$ in a finite time interval. A system is defined as observable if, given the knowledge of $y(t)$ in a finite time interval $t \in [t_0, t_1]$, any initial state $x(t_0) = x_0$ can be identified. Less formally, controllability ensures that acting on the input the system dynamic would evolve in a desired manner and, by observability, this dynamic can be tracked simply measuring the output.

It goes without saying that the constructed ROM should be "strongly" observable and controllable, while the "least" observable and/or controllable states could be discarded during the model reduction process. In order to do that, it is necessary to define:

(i) how the degrees of controllability and observability can be assessed

(ii) whether states that are highly controllable but have a low degree of observability (or vice versa) should be truncated or not.

For what concerns (i), the controllability and observability Gramians W_c and W_o can be computed following the Lyapunov Equations (3.2) and (3.3), respectively.

$$AW_c + W_c A^\top + BB^\top = 0 \tag{3.2}$$

$$A^\top W_o + W_o A + C^\top C = 0\,. \tag{3.3}$$

The matrix W_c specifies to what extent each state is excited by an input. Given two states x_a and x_b, such that $\|x_a\| = \|x_b\|$, if $x_a^\top W_c x_a > x_b^\top W_c x_b$ the state x_a can be defined as "more" controllable than x_b, meaning that a lower control energy must be spent to bring the system from rest to x_a rather than to x_b. On the other hand, W_o measures how much each state excites the future outputs. Hence, initial states which lead to larger outputs are "more" observable and more dynamically important than "less" observable states (Rowley, 2006).

The Gramians are defined in different specific coordinates and, to solve the problem in (ii), a transformation is needed. As long as the system is both controllable and observable (i.e., the Gramians matrices are positive semidefinite), a matrix T can be

introduced (Rowley, 2006) such that:

$$T^{-1}W_cT^{-\top} = T^{\top}W_oT = \Sigma = diag(\sigma_1, ..., \sigma_n)\,. \tag{3.4}$$

The diagonal elements $\sigma_1 \geq ... \geq \sigma_n \geq 0$ are defined as Henkel singular values. The system is now expressed in *balanced* coordinates and the lower energy states can be discarded.

Whereas the balanced truncation consists in a technique accounting for I/O and adjoint-free, its computational cost required is extremely high as two Lyapunov equations should be solved. In addition, when including the LPV framework, the computations become even more expensive (Annoni, 2016).

3.1.2 Proper Orthogonal Decomposition

The proper orthogonal decomposition (POD), also known as Principle Component Analysis, is also a widely used technique in the field of model reduction. In the fluid dynamic community, POD was already employed near the 1970s as an eduction technique to detect coherent structures in turbulence (Pope, 2000). This method aims at finding, amongst the infinite set of possible orthogonal basis, the most *proper* one, namely the one minimizing an error functional.

Formally, given the nonlinear dynamics of the state

$$\dot{x}(t) = Ax(t) + B\mathsf{u}(t)\,,$$

a projector $\Pi_r : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_x}$ with a fixed rank r that minimizes the error in Eq.(3.5) is searched:

$$\int_0^{T_{max}} \|x(t) - \Pi_r x(t)\|^2\, dt\,. \tag{3.5}$$

A common way to compute the required POD basis is the method of snapshot. For a discretized time vector $t = t_0 : t_M$, the nonlinear field is collect into the column vector $x(t_j)$. Assembling the columns together we can find the snapshot matrices:

$$\begin{aligned} \mathcal{X} &= [\, x(t_0), x(t_1), ..., x(t_{M-1})\,] \in \mathbb{R}^{n_x \times (M-1)} \\ \mathcal{X}' &= [\, x(t_1), x(t_2), ..., x(t_M)\,] \in \mathbb{R}^{n_x \times (M-1)}\,, \end{aligned} \tag{3.6}$$

where usually $M \ll n_x$. Performing a singular value decomposition (SVD):

$$\mathcal{X} = U\Sigma V^{\top}\,, \tag{3.7}$$

the left-singular vectors $U \in \mathbb{R}^{n_x \times (M-1)}$, the singular values $\Sigma \in \mathbb{R}^{(M-1)\times(M-1)}$ and the right-singular vectors $V \in \mathbb{R}^{(M-1)\times(M-1)}$ of $\mathcal{X}$ are found. The matrix U can be seen as a collection of column vectors ϕ_k:

$$U = \begin{bmatrix} \phi_1 & \phi_2 & ... & \phi_{M-1} \end{bmatrix}\,, \tag{3.8}$$

that are indeed the POD modes. The energy - or weight - of the k-th mode is expressed by the respective element of the diagonal matrix Σ, arranged in descending order. Lastly, the rows of $V^{\top}$ provides the temporal evolution of the spatial POD modes.

Retaining the first r modes[1], $\mathcal{X}$ can be approximated as:

$$\mathcal{X} = U\Sigma V^\top = \begin{bmatrix} U_r & U_{M-r-1} \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 \\ 0 & \Sigma_{M-r-1} \end{bmatrix} \begin{bmatrix} V_r & V_{M-r-1} \end{bmatrix}^\top \approx U_r \Sigma_r V_r^\top , \quad (3.9)$$

where $(\cdot)_r$ and $(\cdot)_{M-r-1}$ denotes a partition of the matrices. Actually, $U_r\Sigma_r V_r^\top$ represents the *best* approximation with rank r of $\mathcal{X}$, as stated by the Eckart-Young theorem. The states can be projected onto a lower-order subspace by the matrix P such that:

$$P := U_r^\top \quad (3.10)$$

$$PP^\top = I_r , \quad (3.11)$$

as (3.11) is ensured by the orthogonality of U.

On a physical standpoint, POD maximizes the average energy of the reduced system, as Eq.(3.12) reports ($\langle\cdot\rangle$ represents the ensemble average).

$$\arg\min_{\phi_k} \langle \, \|x - \Pi_r x\|^2 \, \rangle = \arg\max_{\phi_k} \langle \, \|\Pi_r x\|^2 \, \rangle \quad (3.12)$$

From the definition of Eq.(3.10) follows that $\Pi_r = P^\top P$.

The POD method is not particularly computationally demanding, it has been extended to LPV (Annoni, 2016) and it is adjoint-free. On the other hand, no I/O is included in the standard formulation and, though POD is the best choice at approximating a given dataset, the dynamical performance of the reduced system may be not efficiently described, as stated by Rowley (2006).

It must be said that several other reduction methods are based on a POD approach. For instance, Rowley (2006) developed a balanced proper orthogonal decomposition, partly linking the balanced truncation with the POD method. The resulting technique exhibited better stability but required the solution of the adjoint-model to empirically compute the Gramians in Equations (3.2) and (3.3).

3.1.3 Dynamic Mode Decomposition

Dynamic mode decomposition (DMD) has been another popular choice for ROM generation after its introduction by Schmid and Sesterhenn (2008). As POD, it is based on a snapshot method, that can be applied both to numerical and to experimental data, however it attempts to represent data by an orthogonalization in time (i.e., isolating distinct frequencies) rather than in space. Indeed, although POD optimally captures the energy in a flow, some weakly-energetic, coherent structures may be neglected, worsening the dynamic performance of the ROM.

DMD tries to cast the nonlinear discrete-time dynamics of a state $x_{k+1} = f(x_k)$ into a linear relation:

$$x_{k+1} = Ax_k . \quad (3.13)$$

[1] Actually, the choice of r is a complex matter and while increasing the order would generally yields better performances, one should avoid the overfitting with the training dataset. In the literature, r is chosen such that the retained modes would contain 90% of the overall cumulative energy. However, in practice, the energy decrease with r much slower than expected and this criteria would lead to overestimating the order. Fortes-Plaza et al. (2018) chose r in the order of tens.

Collecting snapshots of the flow field, as in Sec.3.1.2, the matrices $(\mathcal{X}, \mathcal{X}')$ can be constructed (as in Eq.3.6). Hence, Eq.(3.13) can be simply rewritten as:

$$\mathcal{X}' = A\mathcal{X}\,, \tag{3.14}$$

from which A can be obtained as:

$$A = \mathcal{X}'\mathcal{X}^{\dagger}\,, \tag{3.15}$$

where † indicates the pseudo-inverse matrix, often the Moore-Penrose one.

A low-order representation of $x_k \in \mathbb{R}^{n_x}$ can be expressed as $z_k := P^{\top}x_k \in \mathbb{R}^r$, where P is the subspace projection matrix and $r \ll n_x$. Taking advantage of Eq.(3.13), the dynamic of the reduced vector is now:

$$z_{k+1} = P^{\top}APz_k := Fz_k\,, \tag{3.16}$$

and $F \in \mathbb{R}^{r\times r}$ constitutes the state matrix of the lower-order model.

One can clearly adopt the first r POD modes of $\mathcal{X}$ as projection matrix (as was done in Eq.3.10). Then:

$$A \approx PFP^{\top} = U_rFU_r^{\top}\,. \tag{3.17}$$

With the eigenvalue decomposition $F = T\Lambda T^{-1}$ and rearranging Eq.(3.17), the following is obtained:

$$A \approx (U_rT)\Lambda(T^{-1}U_r^{\top})\,, \tag{3.18}$$

where U_rT are actually the DMD modes and Λ provides the specific temporal frequency for the corresponding mode.

The main advantage of DMD is the frequency subdivision performed: DMD modes could be sorted by their spectral content or either by their growth rates, with better efficiency in the study of wake dynamics. Moreover, the method has low computational costs and is adjoint-free. While this approach does not feature I/O terms, an extension of it (DMD with control, DMDc) has been formulated (Annoni, 2016). However, DMD has not been extended to LPV and it is not robust to noise, as stated again by Annoni (2016).

3.1.4 Multiple Input-Multiple Output ROM

Lastly, the method proposed by Annoni (2016) to generate an input-output reduced order model (IOROM) is presented. In the following, a POD approach is linked to a system identification procedure, retracing the steps of a DMDc formulation.

The IOROM technique tries to cast a discrete-time nonlinear system with inputs and outputs:

$$\begin{cases} x_{k+1} &= f(x_k, \mathrm{u}_k) \\ y_k &= g(x_k, \mathrm{u}_k) \end{cases}, \tag{3.19}$$

in a discrete-time LTI system:

$$\begin{cases} x_{k+1} &= Ax_k + B\mathrm{u}_k \\ y_k &= Cx_k + D\mathrm{u}_k \end{cases}. \tag{3.20}$$

Then, snapshots of the states, inputs and outputs are taken and collected in the following matrices:

$$\begin{aligned}
\mathcal{X} &= [\, x_0, x_1, ..., x_{n_s-1} \,] \in \mathbb{R}^{n_x \times (n_s-1)} \\
\mathcal{X}' &= [\, x_1, x_2, ..., x_{n_s} \,] \in \mathbb{R}^{n_x \times (n_s-1)} \\
\mathcal{U} &= [\, \mathsf{u}_0, \mathsf{u}_1, ..., \mathsf{u}_{n_s-1} \,] \in \mathbb{R}^{n_u \times (n_s-1)} \\
\mathcal{Y} &= [\, y_0, y_1, ..., y_{n_s-1} \,] \in \mathbb{R}^{n_y \times (n_s-1)} .
\end{aligned} \tag{3.21}$$

In order to obtain a state-space representation in Eq.(3.20), a projection onto a lower-order subspace is needed to decrease computational costs. Thus, the reduced-order state would be $z_k := P^\top x_k$, resulting in a model:

$$\begin{cases} z_{k+1} &= (P^\top A P) z_k + (P^\top B)\mathsf{u}_k := F z_k + G \mathsf{u}_k \\ y_k &= (CP) z_k + D \mathsf{u}_k := H z_k + D \mathsf{u}_k \end{cases} . \tag{3.22}$$

New state-space matrices have been defined, precisely $F \in \mathbb{R}^{r \times r}$, $G \in \mathbb{R}^{r \times n_u}$ and $H \in \mathbb{R}^{n_y \times r}$, where r is the ROM order. Clearly, these matrices should approximate as well as possible (A, B, C) and Eq.(3.23) is the relation linking them together.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \approx \begin{bmatrix} PFP^\top & PG \\ HP^\top & D \end{bmatrix} = \begin{bmatrix} P & 0 \\ 0 & I_{n_y} \end{bmatrix} \begin{bmatrix} F & G \\ H & D \end{bmatrix} \begin{bmatrix} P^\top & 0 \\ 0 & I_{n_u} \end{bmatrix} . \tag{3.23}$$

It follows that the optimal choice of the state-space matrices of the reduced model minimizes the error in the Frobenius norm:

$$\underset{\left[\begin{smallmatrix} F & G \\ H & D \end{smallmatrix}\right]}{\arg\min} \left\| \begin{bmatrix} \mathcal{X}' \\ \mathcal{Y} \end{bmatrix} - \begin{bmatrix} P & 0 \\ 0 & I_{n_y} \end{bmatrix} \begin{bmatrix} F & G \\ H & D \end{bmatrix} \begin{bmatrix} P^\top & 0 \\ 0 & I_{n_u} \end{bmatrix} \begin{bmatrix} \mathcal{X} \\ \mathcal{U} \end{bmatrix} \right\|_F^2 . \tag{3.24}$$

Again, one can think about using the first r POD modes of $\mathcal{X}$ as projector and, indeed, Annoni (2016) and Annoni and Seiler (2017) adopted this choice. So, $P = U_r$ yields:

$$\begin{bmatrix} F & G \\ H & D \end{bmatrix}_{opt} = \begin{bmatrix} U_r^\top \mathcal{X}' \\ \mathcal{Y} \end{bmatrix} \begin{bmatrix} U_r^\top \mathcal{X} \\ \mathcal{U} \end{bmatrix}^\dagger . \tag{3.25}$$

Eq.(3.25) is the final step of the ROM identification and the information about the full states can be readily obtained by $x_k \approx U_r z_k$.

The IOROM procedure does not require a particularly high computational effort, avoiding the use of an adjoint. Moreover, an LPV extension of the presented framework has been discussed (Annoni, 2016) and can be carried on in a similar manner: snapshots of the states can be collected for several operating conditions and then stacked together in an all state data matrix $\mathcal{X}_{\text{all}}$, whose first r POD modes would be the needed projector. Lastly, even the IOROM is affected by the presence of noise in the snapshots; however, with respect to the DMD case, better performances can be obtained choosing the input term appropriately.

3.2 A superposed IOROM

As it was previously stated in Sec.1.2 a reduced-order model for the whole farm is sought. Fortes-Plaza et al. (2018) obtained a ROM for a 2-turbines array extracting

the POD modes from a CFD simulation of the entire farm. After a linearization around an equilibrium condition with null yaw angle for the upstream turbine, the IOROM was generated and its performance was inspected on different numerical and experimental test cases, yielding good results. Clearly, the whole framework could be applied as it is to a farm with another layout, however a similar procedure would be unfeasible to study systematically the behaviour of generic wind farms, entailing huge computational costs and times. Thus, this thesis studies an alternative method to generate an IOROM to model a wind farm, trying to avoid the limitations of the original procedure.

Examining the Eq.(3.25), here rewritten for a generic projector P

$$\begin{bmatrix} F & G \\ H & D \end{bmatrix}_{opt} = \begin{bmatrix} P^\top \mathcal{X}' \\ \mathcal{Y} \end{bmatrix} \begin{bmatrix} P^\top \mathcal{X} \\ \mathcal{U} \end{bmatrix}^\dagger ,$$

one can study the role of each term involved.

- $\mathcal{U}$ is the input signal that should properly excite the system. Its role is addressed in Sec.3.2.1 and, in the present study, this vector consists in the yaw angles of the first of the N_{WT} wind turbines of the farm;
- $\mathcal{X}$ and $\mathcal{X}'$ are the snapshot matrices. As Sec.3.2.2 reports, these terms are assembled superposing together CFD simulations of single wind turbines;
- $\mathcal{Y}$ is the power output of the wind turbines, so $\mathcal{Y} \in \mathbb{R}^{N_{WT} \times (n_s - 1)}$. In Section 3.2.3 two possible techniques are proposed for its computation;
- P is the matrix needed for the projection onto a lower-order subspace. While this matrix can be obtained from the POD modes of $\mathcal{X}$, Sec.3.2.4 explores alternative methods that can be used in the future to ease computational costs.

3.2.1 Choice of the Input Signal

The $\mathcal{U}$ vector, related to the yaw signal applied to the upstream turbine, should be designed accurately, in order to exert a suitable excitation for the system. Notice how in the IOROM formulation adopted by Fortes-Plaza et al. (2018) $\mathcal{U}$, besides the prescribed yaw angle, includes even other terms to model non-linearity of the system. Namely, for a given snapshot k:

$$\mathbf{u}_k = \boldsymbol{f}(\gamma_k) = \left(\gamma_k, \gamma_k^2, \gamma_k^3, \gamma_k^4, \cos^p(\gamma_k) - 1\right)^\top , \tag{3.26}$$

given the prescribed yaw angle γ_k. The sinusoidal term is added to better describe the relationship between yaw angle and power output of a turbine, as confirmed by experimental observations. The value of p depends on the type of turbine studied and for the one used in (Fortes-Plaza et al., 2018) $p = 1.787$.

Clear physical limitations to the input signal are the range of admissible yaw angles and the possible yaw rates. For the former, a range of $\gamma_{max} = \pm 30^\circ$ would cover the actual operating conditions of a turbine, whereas for the latter a yaw rate of $|\gamma_{/t}| = 30^\circ\,\mathrm{s}^{-1}$ has been chosen (Fortes-Plaza et al., 2018). Another constrain on $\mathcal{U}$ is represented by the maximum simulation time. This numerical issue appears due to the finite computational power available and generally one should aim at having an input vector as long as possible, to examine a higher number of operating conditions. Moreover, as the system under investigation is highly nonlinear, the input should suitably excite both the frequency and the amplitude domains.

The shape of the vector is of utmost importance in the field of system identification. Amongst all the possible signals (step, impulse, rectangular and so on) an APRBS (Amplitude-modulated Pseudo-Random Binary Sequence) was chosen. The latter consists in an extension of the PRBS signal, which is a deterministic approximation of a white noise in discrete time. The PRBS indeed only exerts a frequency excitation, as it consists in a pseudo-random binary sequence with length L (Fig.3.1). Upon this signal an amplitude excitation should be added, using more input ampli-

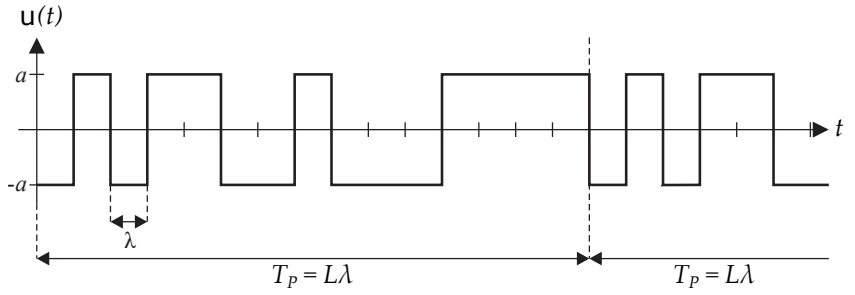


FIGURE 3.1: *A PRBS signal.*

tudes, as Figure 3.2 shows. Further details can be found in the work of Isermann and Münchhof (2014).

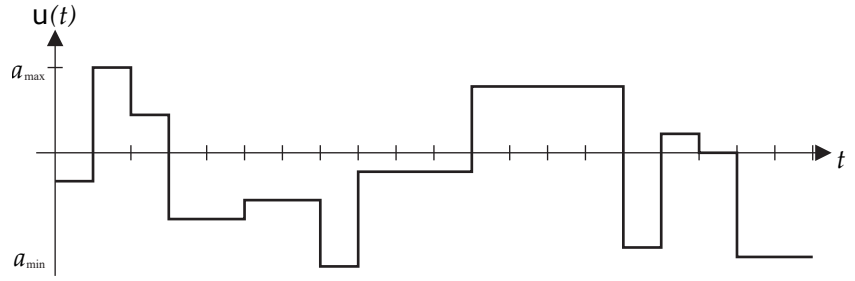


FIGURE 3.2: *An APRBS signal.*

For the present study an APRBS with an hold time $\lambda = 1$s was employed for the yaw-dynamic simulation. Its detailed shape is shown in Section 4.2.2.

3.2.2 Superposition of Single Wind Turbine Simulations

In this paragraph the core of the thesis is introduced, namely the superposition of single wind turbine simulations (SWTS) to obtain a picture of the flow field in the entire wind farm and, ultimately, the snapshot matrices.

The superposition of wakes is not a new field of research (Göçmen et al., 2016) and available commercial software already embed some methods to superpose the engineering wake models, aiming at a description of their interactions. Nevertheless, only a little attention has been paid to the superposition of CFD results, discussed in (Bossuyt, 2018).

For the whole paragraph the superposition methods would consider only the role of the streamwise velocity, that indeed is usually the most meaningful component and that moreover can be used on its own to identify the IOROM of the entire system (Fortes-Plaza, 2017). The spanwise and vertical velocities would be briefly addressed in the Sec.5.1.1.

It should be highlighted that in order to construct $\mathcal{X}$ the SWTS flow fields are superposed at each snapshot and, then, they are collected as column vectors in a matrix.

Finally, recall that the reference frame for the wind farm follows the same convention of that one in Fig.2.1 and it is located at the upstream turbine.

Superposition Methods in the Literature

Consider now a domain with a discrete mesh and a nodal point downstream the P-th turbine. The methods classically adopted for wake superposition would yield:

1. **Linear** $$\left(1-\frac{u}{U_\infty}\right)=\sum_{l=1}^{P}\left(1-\frac{u_l}{U_{0,l}}\right) \tag{3.27}$$

2. **Quadratic** $$\left(1-\frac{u}{U_\infty}\right)=\sqrt{\sum_{l=1}^{P}\left(1-\frac{u_l}{U_{0,l}}\right)^2} \tag{3.28}$$

3. **Maximum deficit** $$\left(1-\frac{u}{U_\infty}\right)=\max_{l=1:P}\left(1-\frac{u_l}{U_{0,l}}\right), \tag{3.29}$$

where:

- U_∞ is the streamwise freestream velocity upstream the first wind turbine
- u is the resulting superposed velocity in the node
- P is the number of previous wind turbines
- $U_{0,l}$ is the incoming wind speed for the l-th turbine
- u_l is the nodal velocity predicted by the SWTS of the l-th turbine.

From the previous definition one can notice that $U_\infty = U_{0,1}$. The superposition scheme is illustrated in Figure 3.3.

Some comments can already be made looking at the definition of the methods. First of all, as the Navier-Stokes equations are highly nonlinear, one would expect that a simple linear superposition as in Eq.(3.27) would not give accurate results. Indeed, the underlying assumptions of the linear method is a small perturbation hypothesis, i.e. the wake deficits are small enough that can be linearly superposed; when this is not actually true, the linear method would overestimate the deficit. On the contrary, the maximum deficit (*maxdef*) is the method, among the three, that would yield the smaller u, as it neglects every interaction amid the wakes and it considers only the strongest deficit. Lastly, the quadratic method assumes that the kinetic energy deficit of the superposed wake is given by the sum of the energy deficits of the previous wakes.

As all these methods were employed with analytical wake models, some minor changes should be applied when dealing with the superposition of CFD results. This is due to the fact that, while wake models predict smooth - or at least constant - profiles, the velocity distributions of CFD simulations are often noisy.

The domain of a SWTS can be expressed as the union of two distinct regions: the in-wake and the out-of-wake parts: $\Omega_{SWTS} = \Omega_{iw} \cup \Omega_{ow}$. The out-of-wake region of the superposed flow-field can be accordingly defined as:

$$\Omega_{ow}^{\text{sup}} = \Omega_{ow}^1 \cap \Omega_{ow}^2 \cap ... \cap \Omega_{ow}^P\,, \tag{3.30}$$

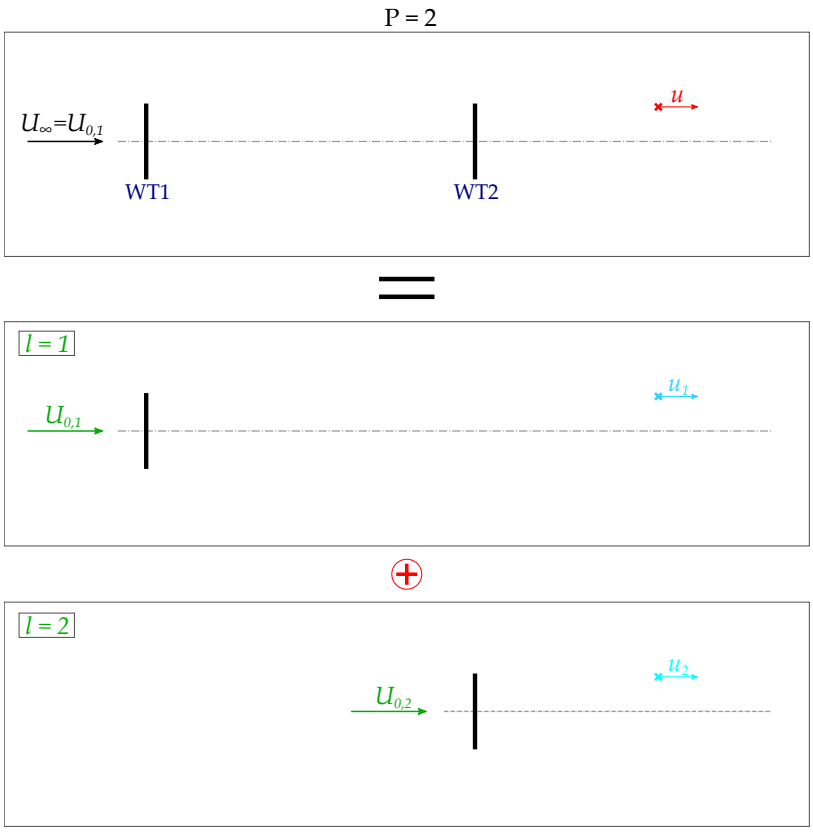


FIGURE 3.3: *Schematic view of the superposition procedure. The velocity in the farm domain is given as a function of the velocity in the same point of the SWTS. The $\oplus$ operator represents the chosen superposition method.*

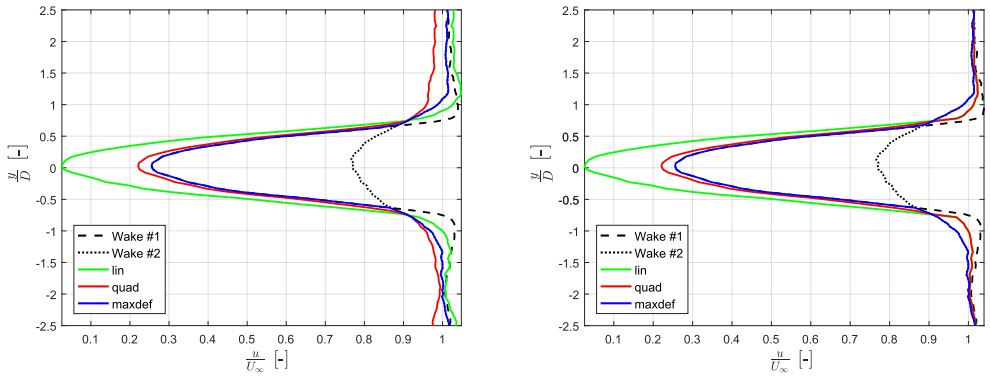
namely the intersection of all the out-of-wake SWTS regions. In Ω_{ow}^{sup} the first modification was introduced. To avoid the introduction of numerical errors due to the noisy CFD profiles, the linear and quadratic methods were corrected to return a velocity such that:

$$u = \frac{1}{P}\sum_{l=1}^{P}\frac{u_l}{U_{0,l}}\,. \tag{3.31}$$

For the maximum deficit method this adjustment was not necessary as it yielded acceptable results in the same region. Figure 3.4 compares the methods with and without the modification. In Fig.3.4a it is shown how - given small deviation of the profiles to be superposed - the quadratic (*quad*) and linear (*lin*) methods drift from the value $\frac{u}{U_\infty} = 1$ out of the wake, whereas the *maxdef* one appears to be less sensitive to this issue. Moreover, the drift of the former methods tends to become larger as the number of wakes increases. In Fig.3.4b the deviation of the methods is decreased implementing the expedient of Eq.(3.31).

Secondly, even the normalization of the wind profile was key. Whereas with wake models it suffices to consistently define the wind speed $U_{0,l}$ as the velocity where the rotor hub would lie, the extension to CFD data required an average over the whole rotor disc. This was performed evaluating the velocity at 1000 randomly generated points inside the rotor disc and averaging the respective values.

Finally, the procedure of domain decomposition employed in the code is described. Before starting with the superposition routine, the domain of the whole



(a) *No out-of-wake correction* (b) *Out-of-wake correction*

FIGURE 3.4: *Comparison of the superposition methods without (a) and with (b) the correction in the out-of-wake region. The profiles to be superposed (*Wake #1 *and* Wake #2*) are extracted from a CFD simulation at* $I_0 = 4\%$.

wind farm Ω^{wf} and the position of the turbines should be defined. Then, Ω^{wf} can be split in two different ways, as portrayed in Fig.3.5, and for the generic l-th subdomain it holds that[2]:

$$\Omega_l^{(a)}(x,y) \ : \qquad x_l - IL \leq x < x_{l+1} + \delta\,, \forall y \tag{3.32}$$

$$\Omega_l^{(b)}(x,y) \ : \qquad x_l - IL \leq x < x_{l+1} - IL\,, \forall y\,, \tag{3.33}$$

where (a,b) represent the two splitting methods as in the sketches, x_l is the streamwise position of the l-th turbine ($x_1 = 0$), IL is the induction length and δ represents a small offset (in the code $\delta = 0.3$m).

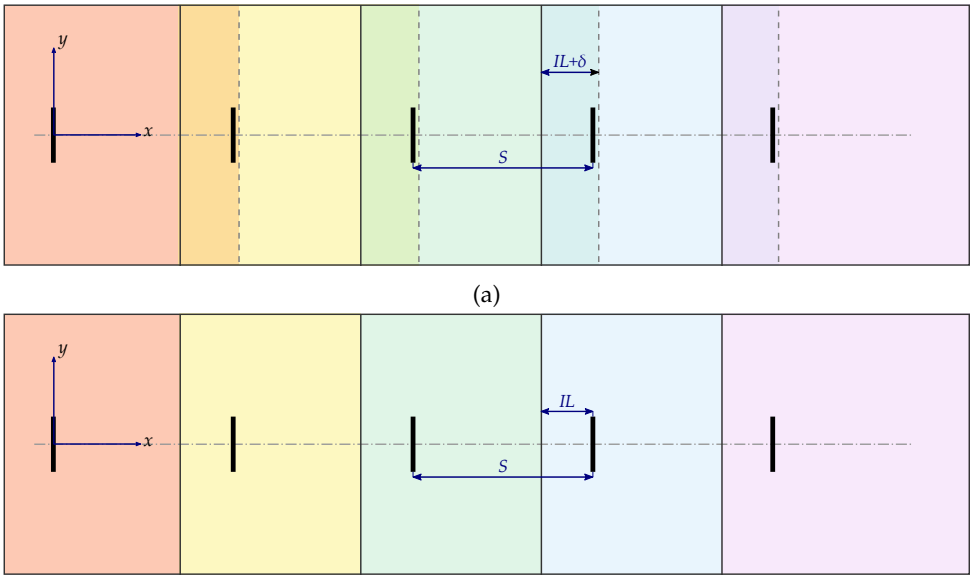


FIGURE 3.5: *Splitting of the wind farm domain for the l-th turbine: from the induction domain to slightly after the next turbine (a) and from the induction domain to the beginning of the next induction region (b).*

[2]For the sake of simplicity a 2D domain is now considered.

With the (a) splitting, one can evaluate the velocity in all $\Omega_l^{(a)}$ superposing the l previous SWTS. Then, the streamwise velocity profile at the horizontal line with coordinates $(x = x_{l+1}, y)$, i.e. where the $(l+1)$-th turbine should lie, is interpolated (and the role of δ is precisely that of extending a bit $\Omega_l^{(a)}$ in the streamwise direction to avoid a less accurate extrapolation) and $U_{0,l+1}$ is computed, as it would be needed for the superposition in the next subdomains. Finally, the velocities in $\Omega_l^{(a)}$ are interpolated in $\Omega_l^{(b)}$ (as in Fig.3.5b) and saved. During this last interpolation the nonphysical velocities obtained by superposing the l previous SWTS in the induction region of the $(l+1)$-th turbine are deleted.

The following algorithm sums up the steps of the superposition process at a given time instant. To obtain the velocity matrix at every time the procedure is simply repeated for each snapshot.

Algorithm 1: Superposition of SWTS at a given snapshot

Given:
  Number of turbines N_{WT},
  Dimensional turbine spacing S,
  Rotor diameter D,
  Length of the induction domain IL,
  Incoming wind speed $U_\infty = U_{0,1}$.
Selection of the proper SWTS with adimensional velocity distribution
Partition of Ω^{wf} into $\Omega^{(a)}$ and $\Omega^{(b)}$
for $l = 1 : N_{WT}$ **do**
  Interpolation of SWTS on $\Omega_l^{(a)}$
  Superposition of the interpolated velocity, obtaining a tmp vector
  Evaluation of the rotor averaged wind speed $U_{0,l}$
  Interpolation of tmp in $\Omega_l^{(b)}$, obtaining the nodal velocities $u[l]$
end
Collection of all the $u[l]$ elements in a column array u.

A Proposal for a New Method

As Section 5.1.1 would show, all the methods presented in the literature are not well performing, in agreement with the study of Gunn et al. (2016). Hence, a better superposition method is sought, such that the superposed flow field would yield better results.

As Gunn et al. (2016) and Shao et al. (2019) reported, all three methods tend to overestimate the defect in the far-wake region, namely the wake recover faster than predicted. This behaviour can be explained by the increase in turbulence intensity given by the previous wind turbines, as stated in Sec.2.1. However, experimental evidences showed that the turbulence of the incoming flow would influence the recovery rate only after a $2D$ distance from the rotor and, ultimately, a better performance of the methods should be noticed in the immediate near-wake. Starting from these observations, a correction to the quadratic method has been proposed, following the work of Shao et al. (2019):

$$\alpha\textbf{-Quadratic} \qquad \left(1 - \frac{u}{U_\infty}\right) = \alpha \sqrt{\sum_{l=1}^{P} \left(1 - \frac{u_l}{U_{0,l}}\right)^2}, \tag{3.34}$$

where the coefficient α is given by:

$$\alpha = 1 - \frac{2D}{S_{ave}}\,. \tag{3.35}$$

In Eq.(3.35), S_{ave} represents the average streamwise dimensional spacing of the previous P wind turbines and in this work, as only the case of constant spacing is taken into consideration, $S_{ave} = S$. The mixing coefficient is engineered such that for infinitely distant turbines the value $\alpha = 1$ is returned, as the machines would basically not interact. It is evident that the coefficient cannot be defined in the trivial case of $S = 0$.

One can notice that $\alpha < 1$, so the squared sum of the deficit would be *decreased* in value, leading to a smaller total deficit than that predicted by the quadratic method, closer to the experimental data. However, the α-quadratic method can be applied only when two or more wakes should be superposed. In fact, in the case in which a wake has to be superposed with a uniform flow field it is intuitive that the overall deficit would simply be the one of the wake itself. This result is actually obtained with the linear, quadratic and maximum deficit methods, but not with the newly proposed one. Hence, in the code, a further wake-region should be separately treated: the multiple wakes region. The greatest advantage of the α-quadratic method, however, lies in its flexibility, i.e. changing the value of α easily modify the speed at which the wake recovers and using $\alpha = 1$ the usual quadratic method is retrieved. Thus, in the distinct regions of the wake, the following scheme is applied:

Out-of-wake region
$$u = \frac{1}{P}\sum_{l=1}^{P}\frac{u_l}{U_{0,l}} \tag{3.36}$$

Single wake region
$$\left(1 - \frac{u}{U_\infty}\right) = \sqrt{\sum_{l=1}^{P}\left(1 - \frac{u_l}{U_{0,l}}\right)^2} \tag{3.37}$$

Multiple wakes region
$$\left(1 - \frac{u}{U_\infty}\right) = \alpha\sqrt{\sum_{l=1}^{P}\left(1 - \frac{u_l}{U_{0,l}}\right)^2}\,. \tag{3.38}$$

The newly designed method should give better results in the far-wake, at the cost of a worse performance in the near-wake, with respect to the standard methods. Thus, a 1D blending is introduced. The blending function is defined such that a value of $\alpha = 1$ is obtained just after the rotor and the value in Eq.(3.35) would be recovered in the far-wake. The link between the two values should be smooth and the chosen function is a $\tanh(\cdot)$, which has been used in engineering applications (for instance, for the two blending functions in the k-ω SST model by Menter et al., 2003).

Consider now the domain section between the l- and $(l+1)$-th turbines:

$$x_l < x < x_{l+1},\ \forall y\,.$$

Introducing a framework fixed on the l-th turbine simply shifting the wind farm reference frame by a streamwise distance of (lS), $\tilde{x} := x - lS$, the 1D blending is defined as:

$$F_{1\mathrm{D}}(\tilde{x}) = 1 - \tanh\left(\frac{\tilde{x}}{1.5D}\right)(1-\alpha) \qquad 0 < \tilde{x} \leq S\,. \tag{3.39}$$

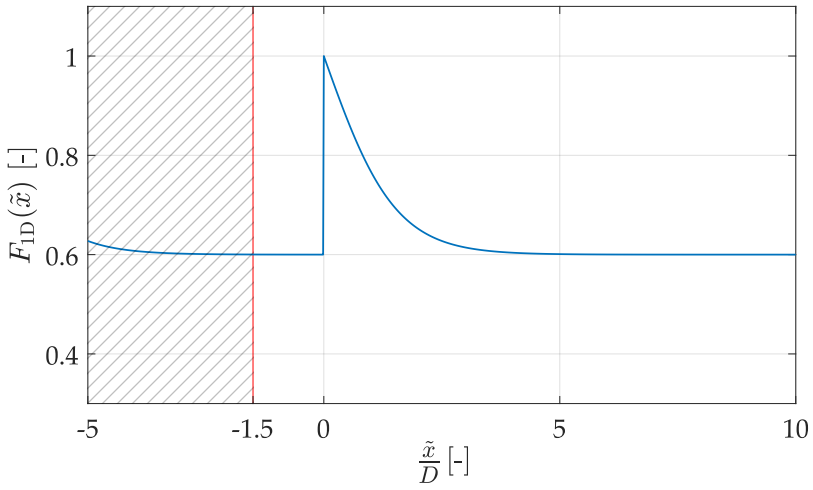


FIGURE 3.6: *One-dimensional blending function, with* $S = 5D$ ($\alpha = 0.6$).

Moreover, in the upstream region of the turbine, that is $-IL < \tilde{x} \leq 0$, the blending should return the value of α defined in Eq.(3.35) because this region is characterized by a fast wake recovery due to the interactions of previous turbines. F_{1D} is accordingly modified as:

$$F_{1D}(\tilde{x}) = \begin{cases} 1 - \tanh\left(\dfrac{\tilde{x}}{1.5D}\right)(1-\alpha) & 0 < \tilde{x} \leq S \\ 1 - \tanh\left(\dfrac{\tilde{x}+S}{1.5D}\right)(1-\alpha) & -IL < \tilde{x} \leq 0 \end{cases} . \tag{3.40}$$

Lastly, the function should also be extended in the region $\tilde{x} > S$. Analysing the physics of the problem, one could argue that turbulence influences the wake recovery in the same manner for the whole far-wake, even at high downstream distances. Thus, $F_{1D}(\cdot)$ is simply kept as it is, extending its domain:

$$F_{1D}(\tilde{x}) = \begin{cases} 1 - \tanh\left(\dfrac{\tilde{x}}{1.5D}\right)(1-\alpha) & \tilde{x} > 0 \\ 1 - \tanh\left(\dfrac{\tilde{x}+S}{1.5D}\right)(1-\alpha) & -IL < \tilde{x} \leq 0 \end{cases} , \tag{3.41}$$

resulting in a behaviour as the one in Figure 3.6.

The introduced 1D-blended method consists in:

1D-blended
$$\left(1 - \frac{u}{U_\infty}\right) = F_{1D}(x - x_P) \sqrt{\sum_{l=1}^{P} \left(1 - \frac{u_l}{U_{0,l}}\right)^2} . \tag{3.42}$$

The method engineered in this way would give a discontinuous flow field at the edges of the wakes, as in the single wake and in the multiple wakes regions two distinct values of α are used. Therefore, to avoid numerical issue in the extraction of the POD modes, even a blending in the spanwise direction is introduced. Several blending functions have been employed and then, in the post-processing phase, the best one is identified. To blend the two methods in the y-direction it is first required

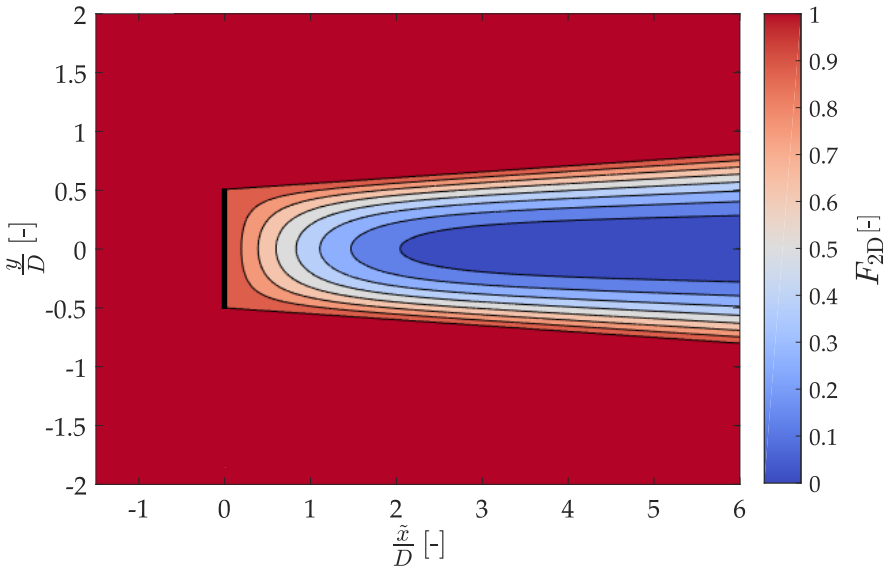


FIGURE 3.7: *Two-dimensional blending function, with* $S = 5D$ $(\alpha = 0.6)$ *and* $i = 2$.

to estimate the width of the wake at every streamwise location, i.e. $span(x)$; in this work it has been assumed a linear expansion of the wake. Then, the blending is performed with:

$$F_{2D}(\tilde{x}, \tilde{y}; i) = \begin{cases} F_{1D}(\tilde{x}) + \tilde{y}^i \dfrac{1 - F_{1D}(\tilde{x})}{span(x)^i} & \tilde{x} > -IL,\ |\tilde{y}| < span(\tilde{x}) \\ 1 & \text{otherwise} \end{cases}, \tag{3.43}$$

where $\tilde{y}$ is the spanwise distance from the wake center, F_{1D} is the one-dimensional blending in Eq.(3.41) and the blending functions chosen are even power function of y with $i = 2, 4, 6$. In Figure 3.7 it is given the behaviour of $F_{2D}(\tilde{x}, \tilde{y}; 2)$ with a span function $span(\tilde{x}) = \frac{D}{2} + 0.05\tilde{x}$, for $\tilde{x} > 0$, which mimics the values of α obtained behind a rotor.

The resulting 2D-blended method with power i is:

2D-blended *(i)* $$\left(1 - \frac{u}{U_\infty}\right) = F_{2D}(x - x_P, y; i) \sqrt{\sum_{l=1}^{P} \left(1 - \frac{u_l}{U_{0,l}}\right)^2}. \tag{3.44}$$

3.2.3 Power Output Prediction

After the matrices $\mathcal{X}$ and $\mathcal{X}'$ are obtained with a proper method, the vector of power output $\mathcal{Y}$ must be estimated, in order to proceed with the ROM extraction. In this regard, two different methods have been employed and subsequently compared, the first one being the prediction of power directly from the superposed flow field and the second one using of FLORIS.

In the superposition procedure, indeed, one can simply exploit the values $U_{0,l}$, i.e. the rotor averaged velocity at the turbine l obtained interpolating the whole flow field, to construct the power output of each wind turbine at every instant of time.

FLORIS, and broadly speaking all engineering models, offers an alternative to the former procedure. However, while being less complex and faster, its greatest drawback is represented by the assumption of a steady state. Thus, the power output should be firstly obtained in steady conditions, meaning that FLORIS should be run as many times as the number of amplitude levels in the APRBS input signal, every time changing the yaw angle of the upstream turbine accordingly. All these power values should be subsequently properly connected among them, during the transient in which the yaw angle of the first turbine is subject to a ramp signal. Overall, this method is clearly more cumbersome than the previous.

In the present study, a comparison between the two techniques is performed, examining which one gives better results and whether it is useful to advance in the definition of the FLORIS procedure.

In the end, as the whole ROM generation consists in a linearized analysis, it must be stressed that, even if each turbine output would be predicted with a large offset but constant in time, the resulting model would still be accurate.

3.2.4 Projection onto a low-order subspace

After $\mathcal{U}, \mathcal{X}, \mathcal{X}'$ and $\mathcal{Y}$ are assembled, the state-space matrices of the reduced model can be computed using Eq.(3.25). In this identification procedure, the projector matrix P would be obtained extracting the first r POD modes of $\mathcal{X}$.

This step requires high computational costs, as a SVD should be carried on, and, most importantly, it would limit the flexibility of the whole framework. Indeed, the modes would be computed *online*, in other words the SVD would be performed only after the snapshot matrix is known. Instead, an *offline* method able to estimate the projector without $\mathcal{X}$ would yield a more adaptable ROM.

To do so, an alternative method is suggested. Baiges, Codina, and Idelsohn (2013) and the more recent work of Xiao et al. (2019) discussed the role of a domain decomposition (DD) strategy applied to reduced-order modelling in fluid dynamic problems. Basically, DD aims at decomposing the whole domain in smaller region and, then, POD modes are evaluated in these subdomains. While this approach is beneficial for the ROM generation, as it eases the whole computational cost parallelizing the POD and it can help to better describe local effects, it still has to be performed after the computation of $\mathcal{X}$. To obtain an offline procedure one can consider to exploit the modes of the SWTS, applying a POD to the snapshot matrix of a single turbine (with an APRBS-like varying yaw angle), which is now assumed available. The POD modes have been chosen as a basis of the different domain subdivisions because, beyond being the best choice to approximate the SWTS flow under given freestream conditions, they actually ensures good performances while describing a flow around the same turbine for conditions different than those in the training dataset (as proved by Fortes-Plaza et al., 2018). What is sought with the selection of the POD modes is to describe in an accurate enough manner the flow around the farm turbines, considered alone, and to examine if their superposition could be able the flow field in the entire array.

If the wind farm domain is subdivided as in Fig.3.5b overlapping are avoided but the POD modes of the SWTS should be extracted after the whole flow field in Ω_{SWTS} is interpolated and restricted to the corresponding $\Omega_l^{(b)}$, making the method strongly dependent on the farm layout. Instead, a DD as the one in Figure 3.8 is considered. With this approach, although a clear overlap appears, POD modes can be earlier obtained on the entire Ω_{SWTS} and afterwards they are simply interpolated on the proper domain of influence of the single wind turbine on the wind farm, i.e.

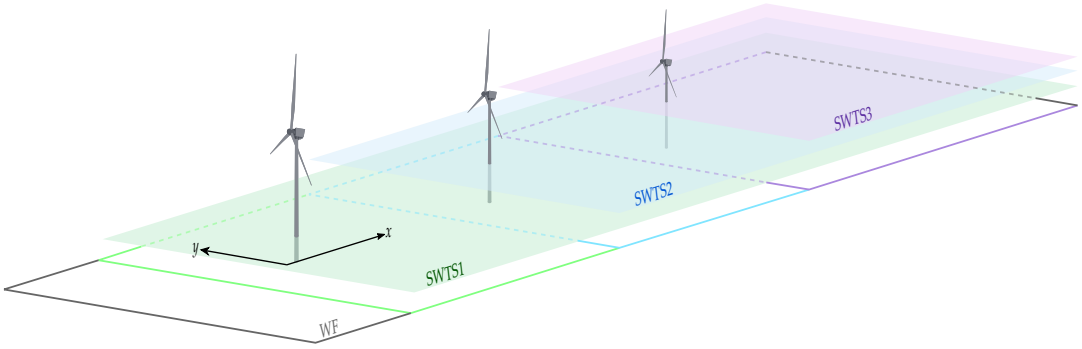


FIGURE 3.8: *Domain decomposed in N_{WT} regions. The coloured planes represent the domain of influence of each turbine. The height of the planes in the z direction has no physical meaning and it is introduced only for graphical purposes.*

for the l-th turbine the region

$$x \geq (l-1)S - IL, \forall y\,.$$

The interpolated modes are then assembled as column vectors of a projector for the whole farm. This matrix would be as the one in Eq.(3.45):

$$Q = \begin{bmatrix} \begin{Bmatrix} \varphi_1 \end{Bmatrix} & \begin{matrix} 0 \\ \end{matrix} & \cdots & \begin{matrix} 0 \\ 0 \\ \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ \end{matrix} \\ \begin{matrix} \\ \\ \end{matrix} & \begin{Bmatrix} \varphi_2 \end{Bmatrix} & & \begin{Bmatrix} \varphi_{N_{WT}-1} \end{Bmatrix} & \begin{Bmatrix} \varphi_{N_{WT}} \end{Bmatrix} \end{bmatrix}, \tag{3.45}$$

if the SWTS-domain is, at least, as long as the farm domain (like in Fig.3.8). If that is not the case, Q would be shaped as:

$$Q = \begin{bmatrix} \begin{Bmatrix} \varphi_1 \end{Bmatrix} & \begin{matrix} 0 \\ \end{matrix} & \cdots & \begin{matrix} 0 \\ 0 \\ \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ \end{matrix} \\ \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} & \begin{Bmatrix} \varphi_2 \end{Bmatrix} & & \begin{Bmatrix} \varphi_{N_{WT}-1} \end{Bmatrix} & \begin{Bmatrix} \varphi_{N_{WT}} \end{Bmatrix} \\ & \begin{matrix} 0 \\ 0 \end{matrix} & & \begin{matrix} 0 \end{matrix} & \end{bmatrix} \tag{3.46}$$

Supposing to use the same r number of POD modes for every SWTS, the projector would be

$$P \in \mathbb{R}^{N \times (r\, N_{WT})}\,,$$

where N is the number of points in the wind farm mesh and N_{WT} the number of wind turbines in the array.

Clearly, the orthogonality of P is lost due to the interpolation and, most importantly, to the overlapping of several modes. However, dealing with a set of linearly dependent vectors may be considered redundant as some vectors can be viewed as a linear combinations of others and so they do not actually add any additional information. Hence, one may seek a basis of linearly independent vectors which would describe the subspace without redundancy. As a consequence, an orthogonalization

of the matrix is performed following the modified Gram-Schmidt (MGS) algorithm, which gives more stable results than its original counterpart. This process leads to a new matrix $\hat{P}$, such that $\hat{P}^\top \hat{P} = I_{r\,N_{WT}}$.

CHAPTER 4

Simulation Environment

This chapter describes the simulation environment adopted in the thesis. In Sec.4.1 a brief introduction to the turbulence models is given, focusing especially on the two frameworks used. Subsequently, Section 4.2 defines the setup of the simulations, explaining firstly the turbine model adopted and then the computational details of the simulations run. For clarity, the latter section is further divided into two paragraphs:

- Sec.4.2.1 explains the framework of the time-averaged simulations, used to validate the superposition scheme. In this regard, the chosen layouts are presented and the criteria followed to select the SWTS is illustrated. Lastly, the procedure adopted for the inflow generation is presented.
- Sec.4.2.2 focuses on the description of the simulation with an APRBS (3.2.1) yaw signal, which consists in the starting point for the ROM extraction.

4.1 CFD Formulation

The Navier-Stokes equations are the governing equations for a viscous fluid. When dealing with an incompressible flow and Newtonian fluid, these equations can be written in their non-conservative and differential form, namely:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho = 0 \tag{4.1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{4.2}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \boldsymbol{u}\,, \tag{4.3}$$

where Eq.(4.1) is the mass conservation equation, (4.2) is the incompressibility constrain and (4.3) is the momentum conservation. The used variables are: the density ρ; a vectorial velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$; the pressure $p(\boldsymbol{x}, t)$ and the kinematic viscosity ν. Recall that for an incompressible flow, given an initial non-stratified fluid, the density would remain the same in the whole domain (as can be simply inferred integrating Eq.4.1) and the pressure is only a Lagrangian multiplier.

The Navier-Stokes equations are highly non-linear and a key parameter is given by the Reynolds number Re:

$$Re = \frac{U_M L_{ref}}{\nu} = \frac{\text{Inertial forces}}{\text{Viscous forces}} \tag{4.4}$$

where U_M is the macroscopic speed of the fluid and L_{ref} a proper reference length. As *Re* gets higher, instabilities are experienced by the laminar flow and the non-linear terms become more and more important. Ultimately, this scenario leads to the onset of turbulence.

Turbulent flows have a qualitative different behaviours than laminar flows. Turbulence entails no scale separation and it is a dissipative and (quasi-)random process. A turbulent field, as $\boldsymbol{U}(\boldsymbol{x}, t)$, shows a random nature (and distinct realizations behave differently) but its statistics - like the mean and standard deviation parameters - appear to be more stable and (supposedly) reproducible; moreover, in this chaotic motion, some deterministic coherent structures can be identified (as the ones described in Sec.2.1), explaining why turbulence can be labelled as *quasi*-random.

Starting from this, one can simply think to study only the statistical properties of the flow, for instance averaging the Equations (4.2) - (4.3). Yet, doing so the *closure problem of turbulence* emerges: no matter which and how many manipulations are done, there will be always more statistical unknowns then equations. As Davidson (2015) well stated:

> It seems that nature (God?) has a nice sense of irony. On the one hand we have a physical quantity, $\boldsymbol{U}$, which behaves in a random fashion, yet is governed by a simple, deterministic equation. On the other hand the statistical properties of $\boldsymbol{U}$ appear to be well behaved and reproducible, yet we know of no closed set of equations which describes them!

As the solution of the Navier-Stokes equations for all the scale of motions is prohibitive for basically all engineering applications due to the high computational costs, one should aim at adopting *turbulence models*, focusing on the statistical quantities of the flow. This latter approach is the one followed, in different ways, by CFD models.

In Sec.4.1.1 and 4.1.2 the models adopted in this thesis are examined. For both of them the PISO (Pressure Implicit with Splitting Operator) algorithm implemented in OpenFOAM was followed to solve the Navier-Stokes equations. Furthermore, the wind turbine nacelle and tower are modelled with an immerse boundary formulation, ensuring better performances in the near-wake (Wang et al., 2018b).

All the CFD simulations were run on the 'SuperMUC' cluster of the Leibniz Supercomputing Centre (LRZ, Germany).

4.1.1 Large-Eddy Simulation

In large-eddy simulation (LES) only the larger turbulent structures are computed, while the smaller ones are modelled. The whole framework can be summarized in four different steps (Pope, 2000):

(1) Filtering of the velocity

(2) Derivation of the filtered Navier-Stokes equations

(3) Closure of the problem

(4) Numerical solution of the filtered equations

In the first step 1, the velocity $\boldsymbol{U}(\boldsymbol{x}, t)$ is decomposed as:

$$\boldsymbol{U}(\boldsymbol{x}, t) = \widetilde{\boldsymbol{u}}(\boldsymbol{x}, t) + \boldsymbol{u'}(\boldsymbol{x}, t)\,, \tag{4.5}$$

where $\widetilde{\boldsymbol{u}}$ is the filtered velocity and $\boldsymbol{u}'$ is the residual component. The symbol $\widetilde{(\cdot)}$ identifies the filtering operator, particularly a low-pass filter with bandwidth Δ. It holds that:

$$\widetilde{\boldsymbol{u}}(\boldsymbol{x},t) = \int G(\boldsymbol{r},\boldsymbol{x})\,\boldsymbol{U}(\boldsymbol{x}-\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{r}\,, \tag{4.6}$$

given a normalized filter kernel such that:

$$\int G(\boldsymbol{r},\boldsymbol{x})\,\mathrm{d}\boldsymbol{r} = 1\,. \tag{4.7}$$

Different filters have been defined in the literature (Pope, 2000) and those which shows the property:

$$G(\boldsymbol{r},\boldsymbol{x}) = G(\boldsymbol{x}) \tag{4.8}$$

are known as uniform.

For uniform filters the filtering and differentiation operations commute and, following step 2, the incompressibility constrain implies that $\widetilde{\boldsymbol{u}}$ and $\boldsymbol{u}'$ are solenoidal (i.e., divergence-free). Filtering the momentum equations one notices the appearance of a new term, as the Eq.(4.9) in tensorial form shows:

$$\frac{\partial \widetilde{U}_j}{\partial t} + \frac{\partial \widetilde{U_i U_j}}{\partial x_i} = \nu \frac{\partial^2 \widetilde{U}_j}{\partial x_i \partial x_i} - \frac{1}{\rho}\frac{\partial \widetilde{p}}{\partial x_j}\,. \tag{4.9}$$

Thus, the residual-stress tensor can be defined as:

$$\tau_{ij}^{\mathrm{R}} := \widetilde{U_i U_j} - \widetilde{U}_i \widetilde{U}_j\,, \tag{4.10}$$

whose anisotropic part is equal to:

$$\tau_{ij}^{\mathrm{r}} = \tau_{ij}^{\mathrm{R}} - \frac{1}{3}\tau_{ii}^{\mathrm{R}}\delta_{ij}\,. \tag{4.11}$$

Therefore, Eq.(4.9) can be rewritten including τ_{ij}^{r}. The result is:

$$\frac{\partial \widetilde{u}_j}{\partial t} + \widetilde{u}_i \frac{\partial \widetilde{u}_j}{\partial x_i} = -\frac{1}{\rho}\frac{\partial \widetilde{p}}{\partial x_j} + \nu \frac{\partial^2 \widetilde{u}_j}{\partial x_i \partial x_i} - \frac{\partial \tau_{ij}^{r}}{\partial x_i}\,, \tag{4.12}$$

where $\widetilde{p}$ is redefined including the isotropic part of τ_{ij}^{R}:

$$\widetilde{p} := \widetilde{p}_{\mathrm{old}} + \frac{1}{3}\rho\tau_{ii}^{\mathrm{R}}\,.$$

To solve the problem a closure should be performed (step 3), modelling the residual-stress tensor. In the Smagorinsky model, which is the one employed in this study, the relation:

$$\tau_{ij}^{\mathrm{r}} = -2\nu_{\mathrm{r}}(\boldsymbol{x},t)\widetilde{S}_{ij} \tag{4.13}$$

is introduced, linking the residual stress with the filtered rate of strain. Still, closure has not been reached as the residual viscosity is yet to be determined. Adopting a mixing-length approach:

$$\begin{aligned} \nu_{\mathrm{r}} &= \ell_{\mathrm{s}}^2 \mathcal{S} \\ &= (C_{\mathrm{s}}\Delta)^2\,\mathcal{S}\,, \end{aligned} \tag{4.14}$$

where $\mathcal{S} = \left(\widetilde{S}_{ij}\widetilde{S}_{ij}\right)^{\frac{1}{2}}$ is the characteristic filtered rate of strain and ℓ_{s} is indeed the

Smagorinsky length scale. The latter is taken as proportional to the filter bandwidth Δ by means of the Smagorinsky coefficient C_s.

In the constant Smagorinsky model the value of C_s is fixed ($C_s = 0.13$ in the code) and no backscatter is allowed, namely energy is only transferred from the filtered to the residual motions.

Finally, step 4, the equations are closed and can be solved. In principle, the resolution of the system is uncoupled from the previous filtering procedures. However, in practice, the bandwidth Δ is often proportional to the grid size δ_x.

The detailed features of the LES framework adopted can be found in the more exhaustive discussion given by Wang et al. (2018a). Briefly, the LES approach has shown enough accuracy and its performances (in terms of power, loads and wake behaviour) has been verified with respect to wind tunnel measurements (Wang et al., 2018a; Wang et al., 2019). However, the required computational costs somehow prevent the application of this approach *"to cases with many wind turbines, long physical times, or when multiple operating conditions are of interest, which is for example the case in wind plant control research"* (Wang et al., 2018b), demanding the use of dense mesh to accurately resolve the flow features.

4.1.2 Scale-Adaptive Simulation

To ease the computational burden of LES, a scale-adaptive simulation (SAS) approach for wind turbines modelling was proposed by Wang et al. (2018b). This framework is based on the turbulence model introduced, and then refined, by Menter and Egorov (2010).

SAS are a class of models in which two independent scales from the source/sink terms are obtained, the first scale being computed from first-derivative of the velocity (as the strain rate tensor or the vorticity tensor) and the second one related to higher derivatives. Standard Reynolds-averaged Navier-Stokes (RANS) models typically provide only the first one of this scale, leaving the second scale out from their formulation. This ensures to SAS the capability to react more dynamically to resolved scales in the flow, that cannot be handled by usual RANS frameworks (Menter and Egorov, 2006). Thus, this scenario somehow positions SAS as a bridge between RANS and LES, avoiding to implement a blending between the methods dependent on the grid spacing, as it is often done in hybrid methods (such detached-eddy simulation, DES).

The SAS formulation used in the CFD code is derived from the KSKL model, afterwards cast in the most known k-ω SST framework (Menter and Egorov, 2010). This whole scheme is based on the work of Rotta, who - instead of modelling heuristically an equation either for ϵ (the turbulent kinetic energy dissipation) or for ω (the turbulent specific dissipation rate) - formulated an exact transport equation for the quantity kL, where k is the turbulent kinetic energy and L an integral length scale. Being a 2-equations model, Rotta coupled this transport equation with the k-equation, as done by almost all 2-equations models. The resulting modelling equations resemble those of the other models with the exception of a third-derivative of the velocity, which however causes a nonphysical behaviour in the logarithmic region and numerical complications. Yet, Menter and Egorov (2010) showed how some of the Rotta's assumptions could be relaxed, being too restrictive, and they formulated a two equations $k - \sqrt{k}L$ (k - square root k L, KSKL) model, featuring a second derivative of the velocity. After some manipulations, one can notice the presence of an additional source term, characterized by the von Kármán length scale L_{vK}. The latter is actually the term ensuring the scale-adaptivity of the method.

The k-ω SST model can be augmented with an additional term, Q_{SAS}, coming from the transformation of the terms in the KSKL model. The resulting equations, extarcted by (Wang et al., 2018b), read:

$$\begin{aligned}
\frac{\partial k}{\partial t} + \nabla \cdot (\overline{\mathbf{u}} k) &= P_k - c_\mu k \omega + \nabla \cdot \left(\left(\nu + \frac{\nu_t}{\sigma_k} \right) \nabla k \right) \\
\frac{\partial \omega}{\partial t} + \nabla \cdot (\overline{\mathbf{u}} \omega) &= \alpha_\omega \frac{\omega}{\rho k} P_k - \beta_\omega \omega^2 + Q_{\text{SAS}} \\
&\quad + \nabla \cdot \left(\left(\nu + \frac{\nu_t}{\sigma_\omega} \right) \nabla \omega \right) + (1 - F_1) \frac{2}{\sigma_{\omega_2}} \frac{1}{\omega} \nabla k \cdot \nabla \omega \,,
\end{aligned} \tag{4.15}$$

with $\overline{\mathbf{u}}$ as the resolved velocity. The coefficients (c_μ, σ_k) and $(\alpha_\omega, \beta_\omega, \sigma_\omega, \sigma_{\omega_2})$ entail closure of the k and the ω equations, respectively; F_1 is the function blending the k-ϵ and the k-ω models. The SAS term can be written in its extended form:

$$Q_{\text{SAS}} = F_{\text{SAS}} \cdot \max \left(\zeta_2 \kappa \mathcal{S}^2 \frac{L}{L_{vK}} - C \frac{k}{\sigma_\phi} \max \left(\frac{|\nabla \omega|^2}{\omega^2}, \frac{|\nabla k|^2}{k^2} \right), 0 \right) , \tag{4.16}$$

where $\mathcal{S}$ is the invariant of the strain rate tensor, $(\zeta_2, \sigma_\Phi, C)$ are tuned from experiments and F_{SAS} is a scaling coefficient that governs the damping of the flow (in the code, $F_{\text{SAS}} = 2$). Lastly, the von Kármán length scale is computed from Rotta's equation:

$$L_{vK} = \max \left(\frac{\kappa \mathcal{S}}{|\nabla^2 \overline{\mathbf{u}}|}, C_k \Omega_{CV}^{1/3} \right) , \tag{4.17}$$

being κ the von Kármán constant, C_k a model parameter and Ω_{CV} the cell volume.

The refined SST framework shows improvement in the prediction of the breakdown of turbulent structures and in the damping at high wave numbers. Wang et al. (2018b) discussed the behaviour of the SAS model compared to LES and experimental data. Overall, SAS provided mainly good results, in agreement with both numerical and experimental dataset, with a significantly lower computational time and the choice of coarser grids. However, using less dense mesh generates in turn numerical flaws, such as a worse resolution of tip vortices in low turbulence conditions. Hence, they advised the adoption of the SAS-framework in the initial stages of a study, taking advantage of its lower computational costs, switching then to LES to obtain more accurate final results.

4.2 Simulation Setup

Both the CFD simulations and the experimental data previously collected at the GVPM study the flow around one or more G1 (Generic 1m rotor diameter) wind turbine models. The G1 is a scaled model of a full-scale reference wind turbine, whose main technical data are summarized in Table 4.1.

TABLE 4.1: *Main technical data of the reference wind turbine.*

Rotor diameter	171.2 m
Rated rotor speed	10.4 rpm
Rated aerodynamic power	7.7 MW

Table 4.2 highlights instead the characteristics of the G1 turbine. In the simulations the G1 is controlled by the control-algorithms implemented in SOWFA.

TABLE 4.2: *Main technical data of the G1 turbine.*

Rotor diameter D	1.1 m
Hub height z_h	0.825 m
Number of blades	3
Rated rotor speed	850 rpm
Nacelle tilt angle	0 °
Rotor cone angle	0 °
Control	Variable speed, pitch and yaw

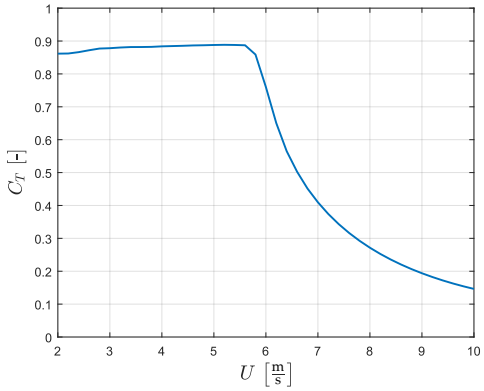


FIGURE 4.1: *C_T-U relation, with $\beta = \beta^*$ and optimal TSR in* R-II.

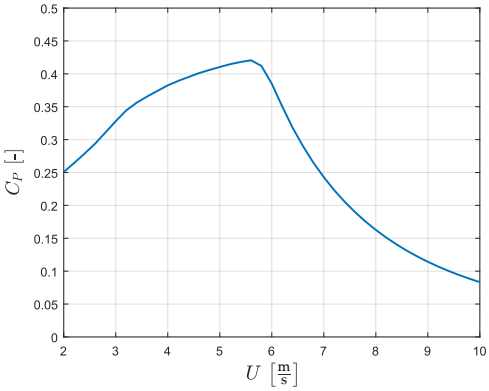


FIGURE 4.2: *C_P-U relation, with $\beta = \beta^*$ and optimal TSR in* R-II.

Starting from the data in the previous tables the following scaling factors can be defined:

$$\mu_L = \frac{171.2\,\text{m}}{1.1\,\text{m}} \approx 155.6 \tag{4.18}$$

$$\mu_t = \frac{850\,\text{rpm}}{10.4\,\text{rpm}} \approx 81.7 \tag{4.19}$$

$$\mu_u = \frac{\mu_L}{\mu_t} \approx 1.9\,, \tag{4.20}$$

respectively for length, time and velocity.

The present study focuses on *Region 2*, hence the role of the pitch angle β is not investigated and for all the simulations its value is constant and is the one maximizing power production, i.e. β^*. The C_T curve with $\beta = \beta^*$ and optimal TSR in R-II is shown in Figure 4.1, while the power coefficient curve is given in Fig.4.2.

In the following Sections 4.2.1-4.2.2 the computational domain and layout is extensively studied. For all the cases, numerical data are collected at some 2D planes, whose dimension and resolutions are given, but clearly the simulations considered the entire 3D domain. Then, it must be noticed that the upstream turbine is always located at a certain distance from the start of the domain. Therefore an initial time should be considered for the inflow to reach the turbines and to develop stationary wakes.

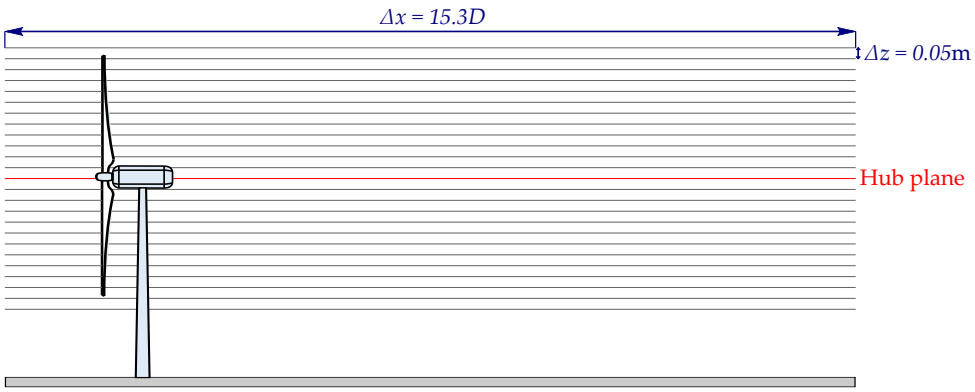


FIGURE 4.3: *Planes for data collection for the time-averaged wakes.*

4.2.1 Time-averaged Wakes with Constant Yaw Angle

Before employing the superposition framework for the construction of the whole matrix $\mathcal{X}$, the methods were firstly validated with time-averaged wakes, which consists in a simpler test-case. Ideally, this corresponds to ensure the accuracy of the methods while assembling a column vector of the snapshot velocity matrix and, afterwards, adopt the same framework for the remaining time instants.

For the time-averaged wakes case, numerical data were collected at several horizontal planes, spanning from $z = 0.2$m to $z = 1.4$m with steps of $\Delta z = 0.05$m (Fig.4.3). In Table 4.3 the properties of the grid used for the horizontal planes are described.

TABLE 4.3: *Grid properties of horizontal planes.*

Dimensions	Resolution
$\Delta x = 15.3D$	$\delta x \approx 0.003D$
$\Delta y = 12.5D$	$\delta y \approx 0.004D$

The simulations were carried out for $T = 18$s with a time step of $0.3 \cdot 10^{-3}$s. Then, the first 10s of simulations were discarded. Indeed, with an incoming wind speed $U_\infty = 5.5\,\mathrm{m\,s^{-1}}$ a time

$$t_{\text{end}} = \frac{\Delta x}{U_\infty} = 3.06\text{s}$$

is required for the inflow to reach the end of the computational domain and in the remaining 6.94s the stationary state is developed. To double check that this stationary state is actually reached one can even inspect the `.out` file generated by SOWFA, verifying that the rotational speed of the turbines has reached convergence and the pitch is kept constant at β^*. From 10s to T the flow field and the power output(s) are averaged.

To validate the superposition scheme it is required to define some reference cases, i.e. CFD simulations of different farm layouts, and to compute the SWTS that should be superposed. The latter were generated consistently with the CFD farm simulations. In other words, the SWTS are superposed aiming at the generation of a flow field as similar as possible to that of the farm simulations, without a direct comparison with experimental data. However, this procedure is only possible because the

CFD framework employed was already validated in (Wang et al., 2018b) and, whenever possible, available experimental data were kept as a reference to understand advantages and limitations of the proposed scheme.

To cut the computational burden, a SAS framework was adopted. This indeed ensured a higher flexibility when dealing with the SWTS and it was actually the only way to compute the flow in a large domain, as the one adopted, in reasonable time.

Defining the Reference Cases

Table 4.4 displays the layouts chosen for the wind farm simulations. For the three-turbines layouts both the streamwise (along x) and lateral (along y) spacing are uniform. In all the cases the inflow was characterized by freestream values of $U_\infty = 5.5\text{m}\,\text{s}^{-1}$ and $I_0 = 4\%$.

TABLE 4.4: *Layouts of the wind farm simulations.*

Layout	**Abbr.**	$\mathbf{N_{WT}}$	$\mathbf{S_{str}}$	$\mathbf{S_{lat}}$
Double aligned	DA	2	$5D$	0
Double misaligned	DM	2	$5D$	$0.5D$
Triple aligned	TA	3	$5D$	0
Triple misaligned	TM	3	$5D$	$0.5D$

A snapshot of the flow field at the hub plane of the double aligned and misaligned cases is given in Figures 4.4 and 4.5.

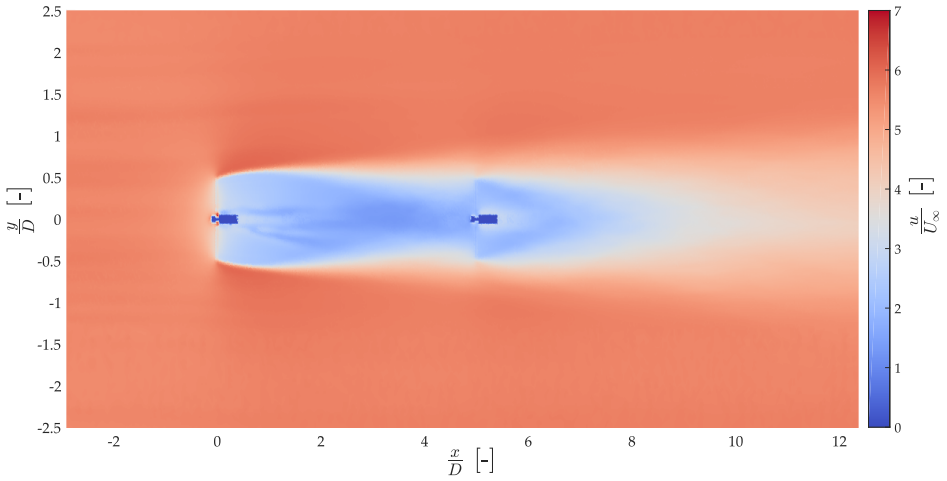


FIGURE 4.4: *Flow field at hub plane for the double aligned layout.*

These configurations were selected because it matches the layouts used in former experimental campaigns at the GVPM, whose data were available. However, this choice contributes to limit the analysis of the TA and TM layouts merely to the near-wake of the third turbine, as - despite the considerable length of the grid - only $3D$ of computational domain are available after the last machine. The flow field in the triple aligned layout is shown in Figure 4.6, while that in the TM layout is portrayed in Fig.4.7.

The reference cases were compared with the experimental data obtained in the GVPM with an inflow $I_0 \approx 5\%$, yielding good results. In Figure 4.8 are portrayed the

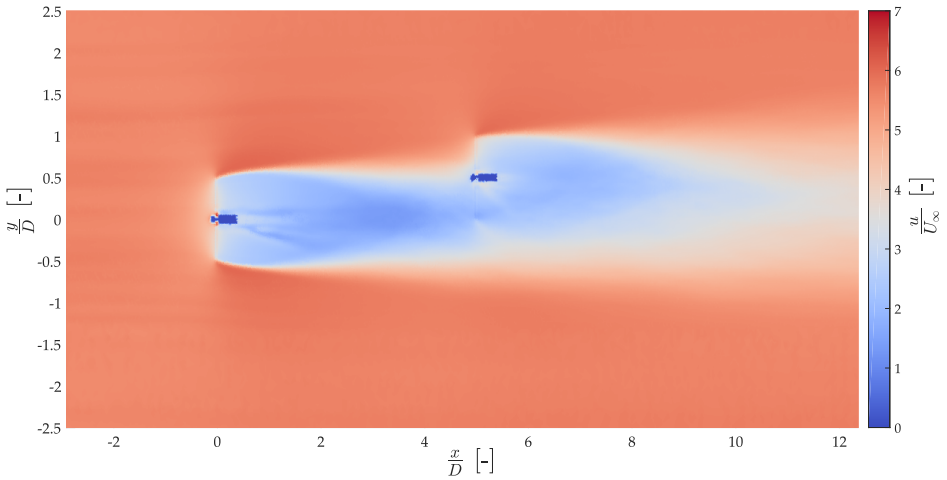


FIGURE 4.5: *Flow field at hub plane for the double misaligned layout.*

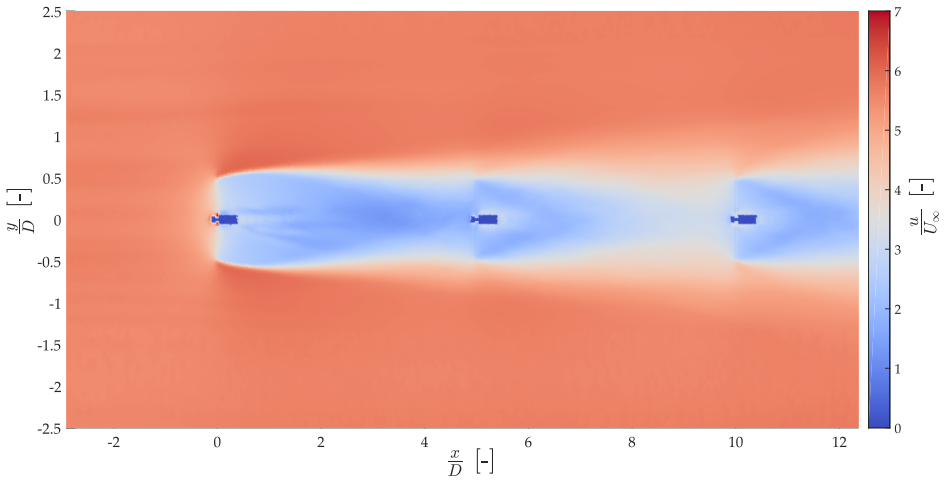


FIGURE 4.6: *Flow field at hub plane for the triple aligned layout.*

numerical and experimental velocity distributions in the DA and DM layouts, sampled at two different locations at hub height. Overall, the numerical model seems to predict adequately the velocity.

Choice of the SWTS

Throughout the previous chapters it was stated that in the superposition scheme *proper* SWTS simulations should be picked. This means that the parameters that characterize the wake behaviour have to be identified.

First of all, it is clear that the turbine features (as the yaw and pitch angles) in the SWTS needs to be the same of their corresponding counterparts in the wind farm simulations. Apart from that, one can consider the incoming wind speed and turbulence intensity as the most important parameters. However, as long as *Region 2* is concerned, the inflow velocity simply plays the role of a scaling factor and, on the other hand, the turbulence is the main variable influencing the wake shape.

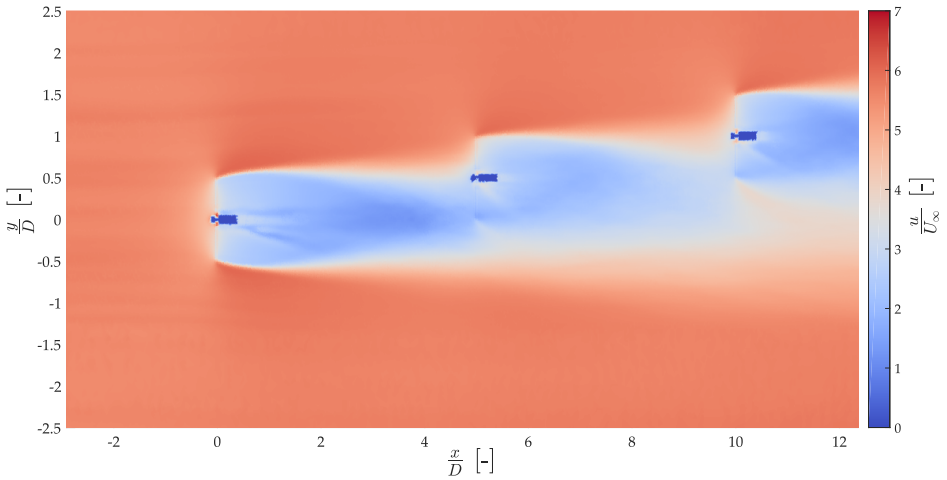
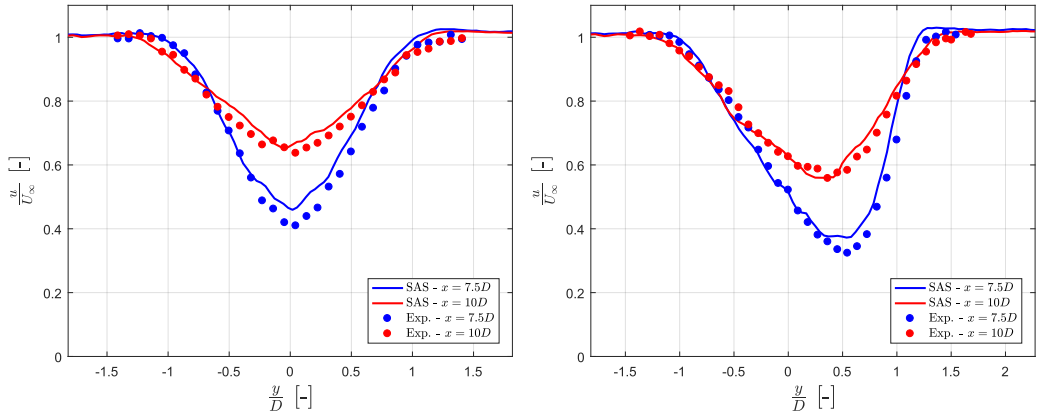


FIGURE 4.7: *Flow field at hub plane for the triple misaligned layout.*



(a) *Double aligned case.* (b) *Double misaligned case.*

FIGURE 4.8: *Comparison of velocity distributions predicted by SAS and measured in experiments for the DA (a) and the DM (b) layouts.*

This is in agreement with the theoretical results, as different recovery and expansion rates of the wake are expected with varying I, while the behaviour of the flow field adimensionalised with U_∞ remains the same. In addition, the choice of I as main parameter of influence is supported by Bossuyt (2018), whose work was founded on the superposition of SWTS obtained with varying incoming wind speed and did not achieve good results.

Therefore, different simulations were run changing the freestream turbulence. The first of the SWTS was simply carried on with an $I_0 = 4\%$, in agreement with the reference farm cases; this simulation is from now on labelled as S-I4. Then, in order to model correctly the downstream turbines, one needs to estimate the corresponding incoming turbulence. Instead of using a simple method as the one in Eq.(2.10), the time-averaged distribution of the turbulence intensity was extracted from the CFD simulations.

Starting from the S-I4 case, this distribution was evaluated at a vertical plane YZ $5D$ downstream the turbine, namely where the rotor of the second turbine of the reference farms should lie. From the turbulence field obtained (Fig.4.9), mean values

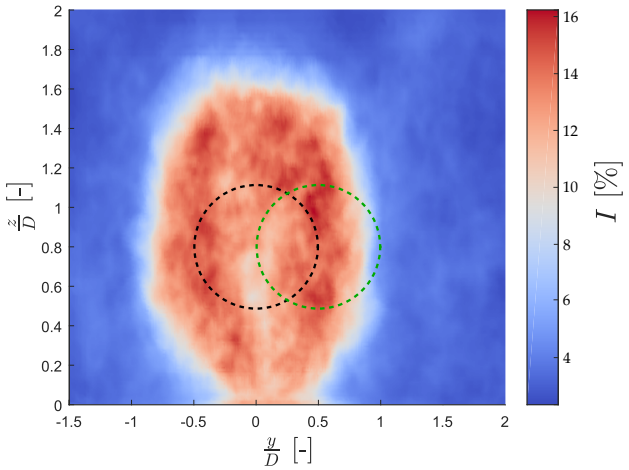


FIGURE 4.9: *I distribution at the vertical plane YZ $5D$ downstream the turbine, in the S-I4 simulation. The averaged-values in the area enclosed by the black and green circumferences are the turbulence felt by the second turbine in the DA and DM layouts, respectively.*

in the area enclosed by a circle with origin $(y, z) \equiv (0, 0)$ and by another circle with origin $(y, z) \equiv (0.5D, 0)$ - both with radius equal to $D/2$ - were extracted. These values are equal to $I = 13.3\%$ and $I = 11.8\%$ and they corresponds to the rotor-averaged turbulence intensities of the second wind turbine in the aligned and misaligned cases, respectively. Subsequently, simulations with inflow $U_\infty = 5.5\text{m}\,\text{s}^{-1}$ and incoming turbulence of $I_0 = 12\%$ (S-I12) and of $I_0 = 13\%$ (S-I13) should be run, with the goal of modeling the behaviour of the second turbine in the different layouts.

Then, even the turbulence of the flow impinging on the third turbine (for the TA and TM cases) had to be assessed. To do so, alongside the computation of the flow field, the turbulence field was evaluated in the DA and DM simulations at a vertical plane YZ, $10D$ distant from the upstream turbine. Given the distribution in the plane, the rotor-averaged I was calculated averaging the values enclosed in a circle with origin $(y, z) \equiv (0, 0)$, for the DA case, and in a circle with origin $(y, z) \equiv (1D, 0)$ (DM), both of whose with diameter D. This analysis yielded respectively values of $I = 11.2\%$ and $I = 10.7\%$. Hence, to model the third turbine behaviour in both the TA and TM layouts just one simulation with $U_\infty = 5.5\text{m}\,\text{s}^{-1}$ and $I_0 = 11\%$ (S-I11) was computed. Table 4.5 sums up the configuration and results of the simulations run to study the turbulence intensity distribution, whereas in Table 4.6 are shown all the SWTS computed accordingly to the former analysis. In the latter table, the S-I12 case does not appear as it was obtained from the interpolation of the S-I11 and S-I13 simulations, reducing the computational time.

Inspecting Tab.4.5 one can notice that, in the aligned case, the rotor-averaged turbulence felt by the third turbine is actually lower than that of the second one. This results seems counterintuitive as, instead, one would expect an increase of I, at least until deep-array effects are reached and the turbulence settles on a steady level. Indeed, experimental data (collected however at a slightly different freestream turbulence intensity) confirms that the third rotor in the TA layout should experience an incoming turbulence of 13.8%, a bit higher that the turbulence of the flow impinging on the second turbine.

TABLE 4.5: *Numerical set-up and results of the simulations for the I study.*

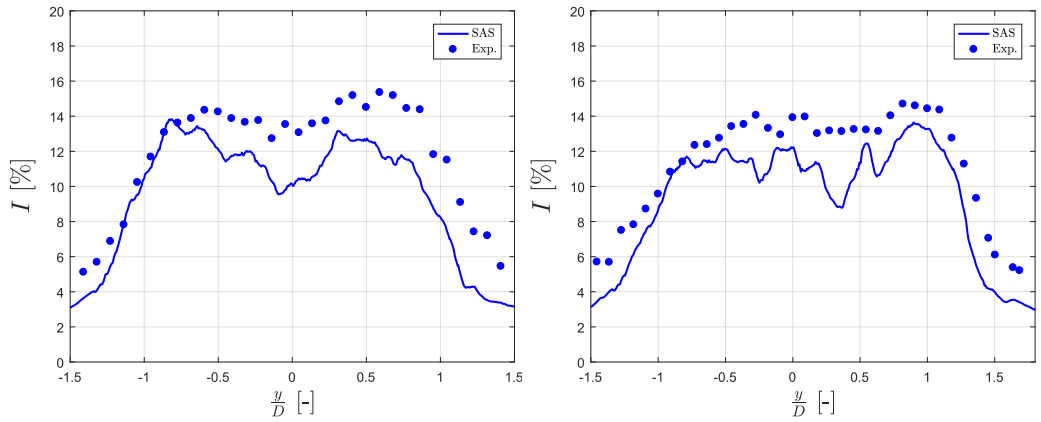
Case	N_{WT}	Distance of *YZ* plane (from upstream WT)	Rotor area origin (y_o, z_o)	Rotor-averaged *I*
S-I4	1	$5D$	(0,0)	13.3%
			$(0.5D, 0)$	11.8%
DA	2	$10D$	(0,0)	11.2%
DM	2	$10D$	$(1D, 0)$	10.7%

All the considered rotor area have a diameter D.

TABLE 4.6: *Summary of the SWTS carried on.*

Case	U_∞	I_0
S-I4	$5.5\,m\,s^{-1}$	4%
S-I11	$5.5\,m\,s^{-1}$	11%
S-I13	$5.5\,m\,s^{-1}$	13%

Observing Figure 4.10, one can notice that the turbulence intensity distribution are not well predicted by the SAS framework, especially in the DA case. It is postulated that the offset between numerical and experimental data is mainly due to the simulation setup, as the coarse grid (typical of SAS) induces artificial numerical diffusion, eventually influencing the *I* distribution, as previously mentioned.



(a) *Double aligned case.* (b) *Double misaligned case.*

FIGURE 4.10: *Turbulence intensity distribution at hub height predicted by SAS and from experimental data. Both the DA (a) and the DM (b) cases are shown.*

Yet, keeping in mind that the turbulence does not match experimental data, in the present study the third turbine of the TA layout was modeled with the S-I11 simulation, maintaining consistency between the CFD cases.

The S-I4, S-I11, S-I13 flow field at hub plane are respectively reported in Figures 4.11, 4.12 and 4.13.

Lastly, in Table 4.7 are then summarised the SWTS to be superposed in the 4 layout considered.

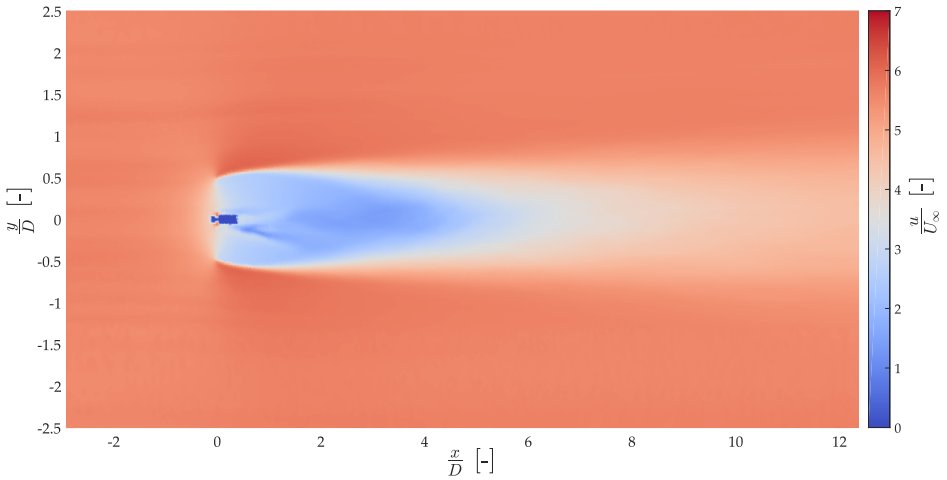


FIGURE 4.11: *Flow field at hub plane for the S-I4 simulation.*

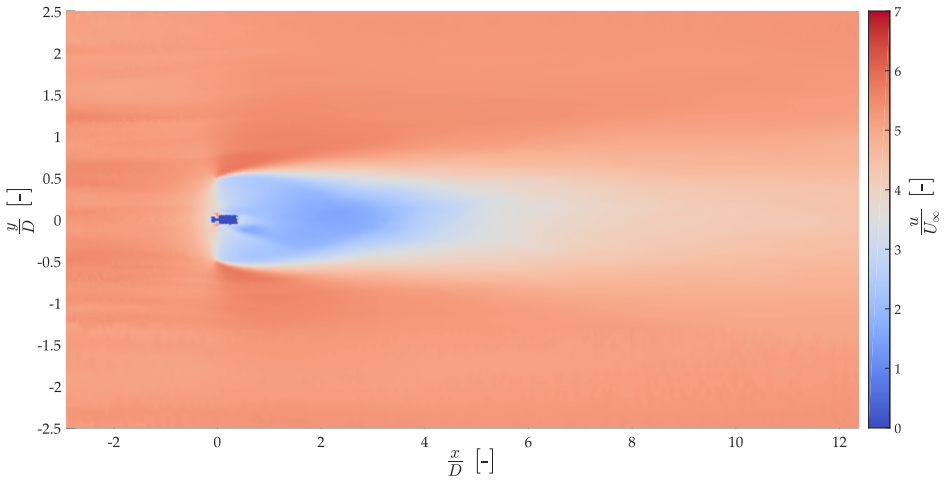


FIGURE 4.12: *Flow field at hub plane for the S-I11 simulation.*

TABLE 4.7: *SWTS modeling the single wind turbines in different layouts.*

	Modeling SWTS for		
	WT1	**WT2**	**WT3**
DA	S-I4	S-I13	-
DM	S-I4	S-I12	-
TA	S-I4	S-I13	S-I11
TM	S-I4	S-I12	S-I11

Generating the Inflow

As the inflow data required by the present study, and for the superposition scheme in general, are characterized by different values of I_0, a flexible and reliable method should be employed for the inflow generation. In this regard, for the present study the inflow was obtained thanks to TurbSim, an NREL's software (NREL, 2019b)

which provides a numerical simulation of a turbulent flow field.

The inflow was generated for the full-scale reference wind turbine and then it was scaled on the G1 case. The grid of the CFD simulations extended from nearly 0m to 3.84m in the z-direction and from -6.9m to 6.9m in the y-coordinate. Translating these lengths in the TurbSim environment, using μ_L as scaling factor, a grid height of 597m and a width of 2141m are obtained. Then, even the grid spacing had to be defined. As a resolution similar to that of the CFD simulations (Tab.4.3) would require too much of computational power, the grid spacing was defined ensuring to discretize the diameter with at least 10 points. For the temporal domain, a time step of 0.0245s was set (equal to the CFD time step multiplied by μ_t) and the usable time of the TurbSim simulation resulted equal to 1400s.

After defining the spatiotemporal properties of the domain, it was necessary to define the meteorological conditions. The wind speed was simply established scaling $U_\infty = 5.5\text{m}\,\text{s}^{-1}$ to $U_{\text{Ref}} = 10.9\text{m}\,\text{s}^{-1}$ by means of μ_u; the reference height in TurbSim was set to 124.5m. The turbulence spectrum chosen is of the Kaimal type. The Kaimal spectrum describes the wind turbulence on short time scales and its use has been suggested for the design of offshore turbines by DNV (2011). The spectral density can be written as:

$$S_U(f) = \sigma_u^2 \frac{4\frac{L_k}{\overline{U}}}{\left(1+6\frac{f\,L_k}{\overline{U}}\right)^{\frac{5}{3}}}, \tag{4.21}$$

where f denotes the frequency and L_k the integral scale parameter (whose formula is included again in DNV, 2011). An example of the Kaimal spectrum is shown in Fig.4.14a with $\overline{U} = U_{\text{Ref}} = 10.9\text{m}\,\text{s}^{-1}$ and $\sigma_u = 0.436\text{m}\,\text{s}^{-1}$ ($I = 4\%$) for the reference turbine at hub height. Recall that the Kaimal spectrum can only be used only to describe turbulent fluctuations, up to a frequency content of, say, $\sim 2 \cdot 10^{-3}$Hz corresponding to 10min long fluctuations, as 4.14b portrays. The IEC-Standard used was the IEC 61400-3 (offshore turbines), which determined the power-law exponent ($\alpha_{PL} = 0.14$) and the surface roughness length ($Z_0 = 0.03$m). Finally, the stability of

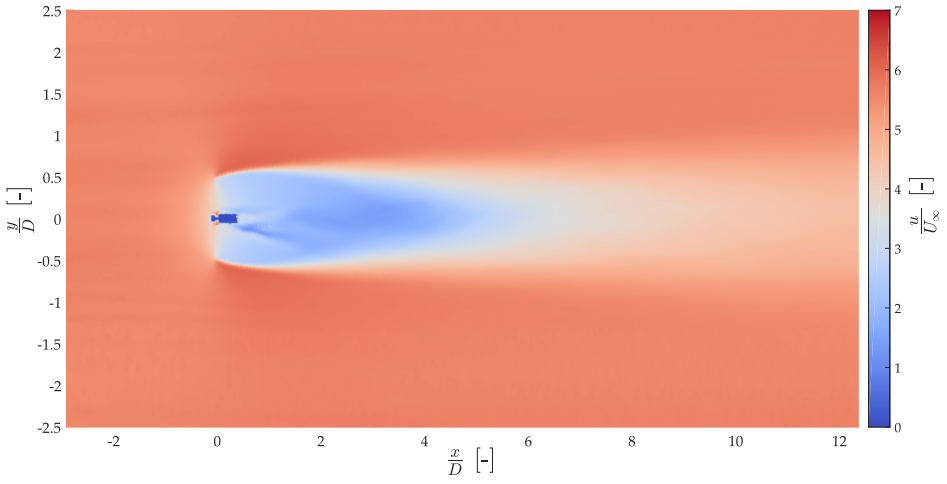
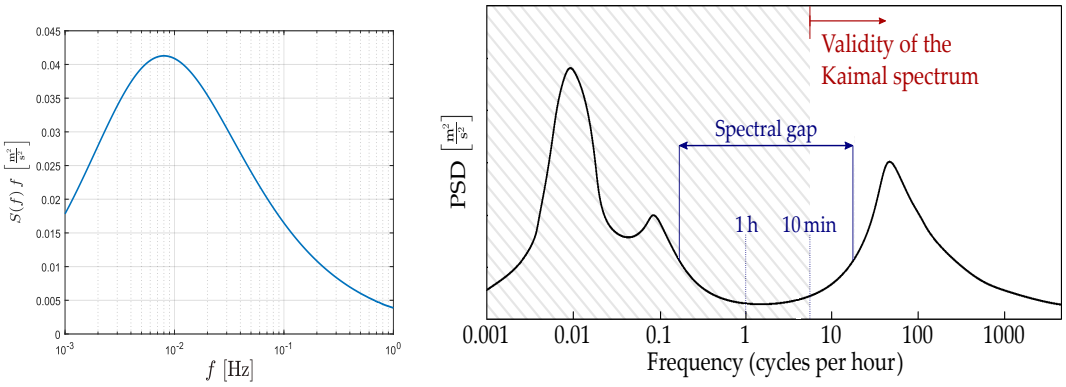


FIGURE 4.13: *Flow field at hub plane for the S-I13 simulation.*



(a) *Kaimal spectrum.* (b) *Van der Hoven spectrum.*

FIGURE 4.14: *Kaimal spectrum (a) and comparison with the full Van der Hoven spectrum (b), underlining the range of validity of the former.*

the ABL had to be determined. In this regard, one should set the gradient Richardson number Ri, that is the adimensional number defined as the ratio between the terms of buoyancy production and shear production in the turbulent kinetic energy equation. Its role related to the ABL is discussed more in-depth by Pelegrì and Gastel (2004). In a nutshell, in the TurbSim framework, a positive Ri describes a SBL condition, while negative values are characteristic of a CBL. However, when Ri exceeds 1 turbulent flows becomes non-turbulent. In the present study $Ri = 0.05$ was chosen.

In Table 4.8 are summed up the main parameters chosen for the inflow generation in TurbSim.

TABLE 4.8: *Main parameters of TurbSim input file.*

Grid height	597.0m
Grid width	2141.0m
Number of vertical grid points	38
Number of horizontal grid points	140
Turbulence model	Kaimal
IEC standard	61400-3
Reference height	124.5m
Reference velocity	$10.9\,\mathrm{m\,s^{-1}}$
Gradient Richardson number	0.05

4.2.2 A Single Wind Turbine Simulation with Varying Yaw

The simulation with the APRBS-like yaw input was run with the same incoming wind speed $U_\infty = 5.5\,\mathrm{m\,s^{-1}}$ but a turbulence intensity $I_0 = 6\%$.

Data were collected at two planes: one horizontal plane XY at hub height ($z = 0.8$m and a total number of points $\sim$ 380 000) and one vertical plane XZ going through the center of the turbine tower ($y = 0$m with $\sim$ 250 000 points). The grid properties of the XY plane, the only used in this thesis, are shown in Table 4.9.

The simulation was carried on with a time step of $0.3 \cdot 10^{-3}$s and data were sampled every 0.012s. Every snapshot of the flow field recorded the three-components velocity vector at each nodal point, resulting in nearly 1 890 000 variables stored for

TABLE 4.9: *Grid properties of the horizontal plane in the yaw-varying simulation.*

Dimensions	**Resolution**
$\Delta x = 15.2D$	$\delta x \approx 0.006D$
$\Delta y = 6.3D$	$\delta y \approx 0.010D$

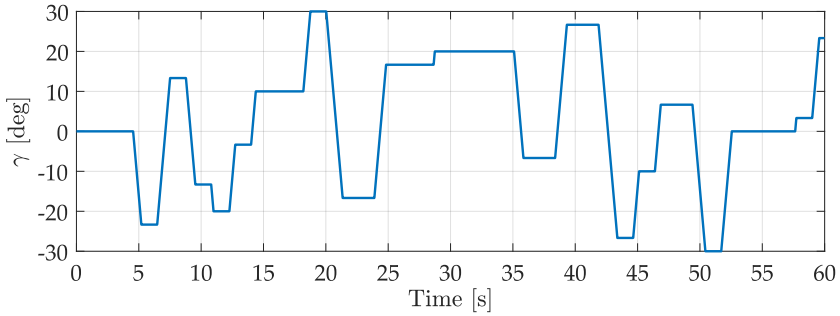


FIGURE 4.15: *Yaw signal applied to the turbine in the yaw-varying simulation.*

every instant. The simulation was run for $T = 60$s and the flow field during the first 5s were averaged, recovering a time-averaged wake. Indeed, observing Figure 4.15 the yaw-input of the turbine, one could notice an initial null value of γ, designed precisely to bring the system to a steady-state. The remaining part of the input has instead an APRBS shape.

The CFD simulation in this case adopted a LES scheme, previously validated by Wang et al. (2018a) and Wang et al. (2019). Comparing the grid properties for this case and the time-averaged one, it is noticeable how the mesh has more or less the same density. Still, in both the SAS and LES framework employed the grid is denser than that of (Fortes-Plaza et al., 2018).

The chosen turbulence differ from that used in the time-averaged cases ($I_0 = 4\%$) because this simulation was run before the inflow generation with TurbSim, adopting the same framework employed in Fortes-Plaza et al. (2018) to obtain a turbulent inflow. The latter scheme indeed consists in a precursor domain, which mimics the turbulence-generating system adopted in the GVPM. Further details about the precursor can be found in (Wang et al., 2019).

CHAPTER 5

Results

5.1 Superposition of the Flow Fields

The superposition of time-averaged wakes is investigated, firstly in the aligned layouts and then in the misaligned cases. For all of them, the accuracy of the methods has been evaluated by means of the relative absolute errors (RAE):

$$RAE^{\mathcal{P}} := \frac{\left\langle \left| u^{\mathcal{P}}_{i,\text{ref}} - u^{\mathcal{P}}_{i,\text{sup}} \right| \right\rangle}{U^{\mathcal{P}}_{\infty}} \cdot 100 \qquad i = 1 : N,\ \mathcal{P} = 1 : 25 \tag{5.1}$$

$$RAE^{\mathcal{P}}_{wake} := \frac{\left\langle \left| u^{\mathcal{P}}_{j,\text{ref}} - u^{\mathcal{P}}_{j,\text{sup}} \right| \right\rangle}{U^{\mathcal{P}}_{\infty}} \cdot 100 \qquad j = 1 : N_{wake},\ \mathcal{P} = 1 : 25\,, \tag{5.2}$$

where:

- $\mathcal{P}$ is the horizontal plane under examination
- N is the number of points where data is collected in the horizontal planes and, in general, $N = N(\mathcal{P})$
- N_{wake} is the number of points belonging to the in-wake region $\Omega^{\mathcal{P}}_{iw}$ and evidently: $N_{wake} < N,\ \forall \mathcal{P}$
- $u^{\mathcal{P}}_{i,\text{ref}}$ represents the velocity obtained in the i-th cell for the $\mathcal{P}$-th plane for the reference simulation
- $u^{\mathcal{P}}_{i,\text{sup}}$ represents the velocity obtained in the i-th cell for the $\mathcal{P}$-th plane for the superposed flow field
- $U^{\mathcal{P}}_{\infty}$ is the incoming wind speed at the plane $\mathcal{P}$ of the reference simulation.

The metric RAE_{wake} is more representative of the accuracy of the method than the general RAE, as it specifies the error in the wake, which is the more informative region for the extraction of the ROM. Clearly, it is expected $RAE^{\mathcal{P}}_{wake} > RAE^{\mathcal{P}}$ for every plane, as all the out-of-wake cells in which the superposition methods should return values close to 0 are excluded from the averaging operator.

Throughout all this paragraph, the superposition scheme would be applied to the streamwise velocity only, as it would be the component retained for the ROM extraction (due to memory requirements) and because the reconstruction of the field of the spanwise and vertical velocities would be less critical, as shown during the discussion. While within this paragraph only the graphical distributions of the RAE_{wake} errors are shown, Appendix A thoroughly reports the values of both the RAEs at all the horizontal planes.

5.1.1 Double Aligned Layout

Simply applying the usual superposition methods to the DA case, the flow fields in Figures 5.1 - 5.3 are obtained, in which the relative error field (Eq.5.3) is portrayed.

$$err_i^{\mathcal{P}} := \frac{u_{i,\text{ref}}^{\mathcal{P}} - u_{i,\text{sup}}^{\mathcal{P}}}{U_\infty^{\mathcal{P}}} \cdot 100 \tag{5.3}$$

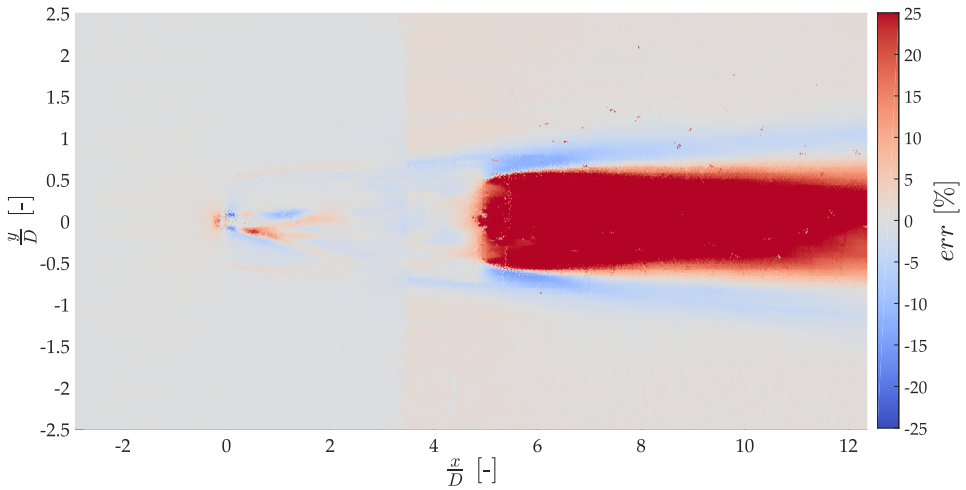


FIGURE 5.1: *Error field at the hub plane applying a linear superposition method to the double aligned layout.*

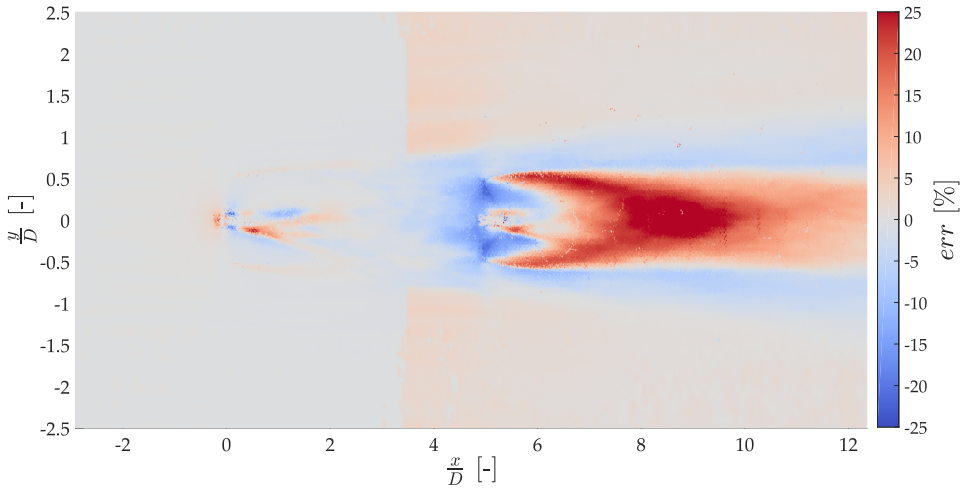


FIGURE 5.2: *Error field at the hub plane applying a maximum deficit superposition method to the DA layout.*

All the three methods visibly underpredict the wake behind the second turbine, as explained in Sec.3.2.2. In particular, the error of the methods is expected to be higher for the hub plane (the one in the Figures), where the wake deficit is stronger. The *RAE*s of the methods at the hub plane are shown in Table 5.1.

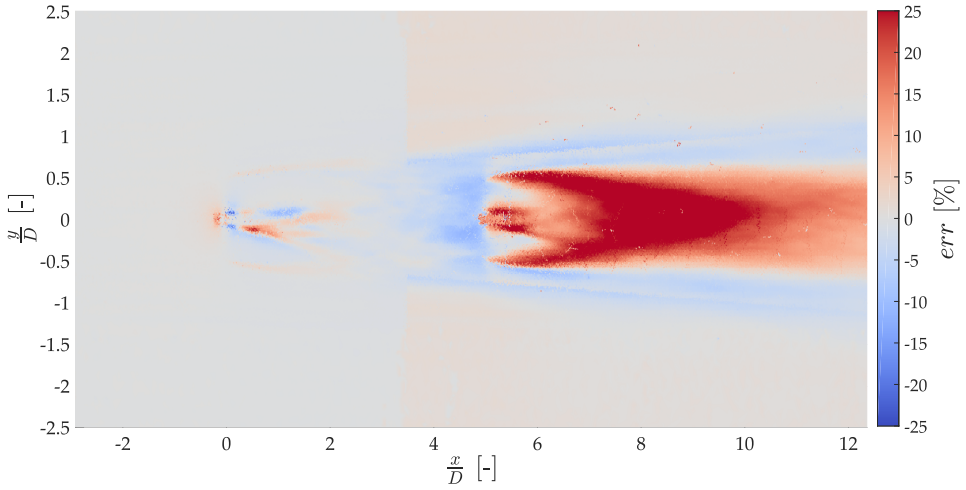


FIGURE 5.3: *Error field at the hub plane applying a quadratic superposition method to the double aligned layout.*

TABLE 5.1: *Relative absolute errors for the standard superposition methods evaluated at the hub plane for the double aligned layout.*

	lin	*maxdef*	*quad*
RAE [%]	6.97	3.34	3.89
RAE_{wake} [%]	14.81	6.46	7.97

Furthermore, the previous pictures display how the out-of-wake correction introduced in Eq.(3.31) refines the performance of the methods in this part of domain, yet without setting rigorously to zero the errors.

Introducing the α-*quad* method the superposed flow field shows improvements in the far-wake, at the cost of an expected deterioration of the accuracy in the near-wake (Fig.5.4).

With a streamwise blending (method *1d-blend*) the performance of the superposition scheme gets better in the near-wake region, still keeping the good accuracy demonstrated in the far-wake. The resulting flow field is given in Figure 5.5.

Lastly, the *2d-blend* method is implemented, smoothing the flow field at the wake boundaries, as Figure 5.6 shows (with a fourth power spanwise blending).

From all the superposed flow fields, one can notice how a region of relatively high error is present behind the tip of the rotor, which becomes more evident while applying the α-methods, as the error in the wake core is diminished. Rather than a numerical error introduced by the CFD due to the numerical resolution of tip vortices (which would induce a more diffused error region), it is postulated that the phenomenon is originated by the regulation of the downwind turbine. For the second turbine, the operational TSR would be enforced by a relation $\lambda = \lambda(\overline{U}) := \lambda^*$, in which $\overline{U}$ is the average wind speed impinging on the rotor. Yet, when the inflow has a gaussian shape, different actual values of λ are enforced at distinct rotor locations. In fact, the rotational speed of the rotor would be dictated by the regulation strategy given the average wind speed and the tip regions would have an actual $\lambda < \lambda^*$ (Fig.5.7), eventually extracting less energy from the flow and leading to a higher downwind velocity. This is also why Ciri et al. (2017) stated that

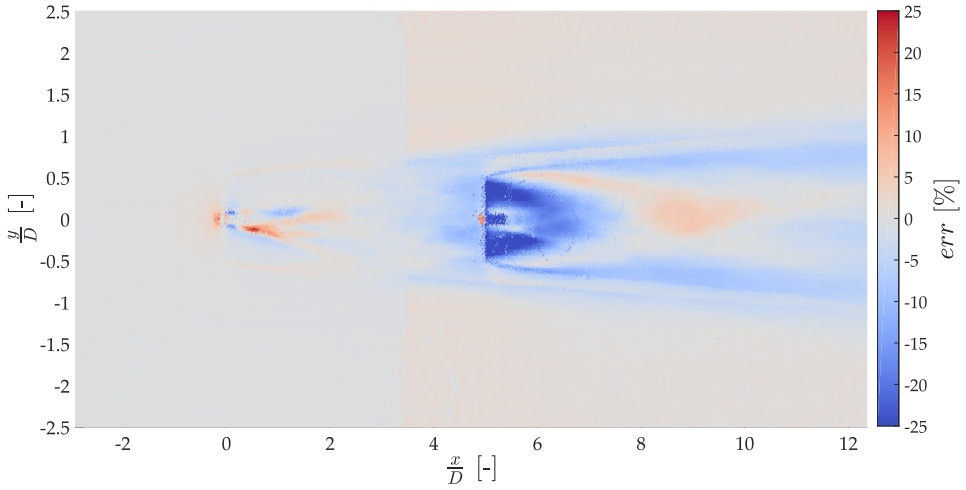


FIGURE 5.4: *Error field at the hub plane applying a α-quadratic superposition method to the double aligned layout.*

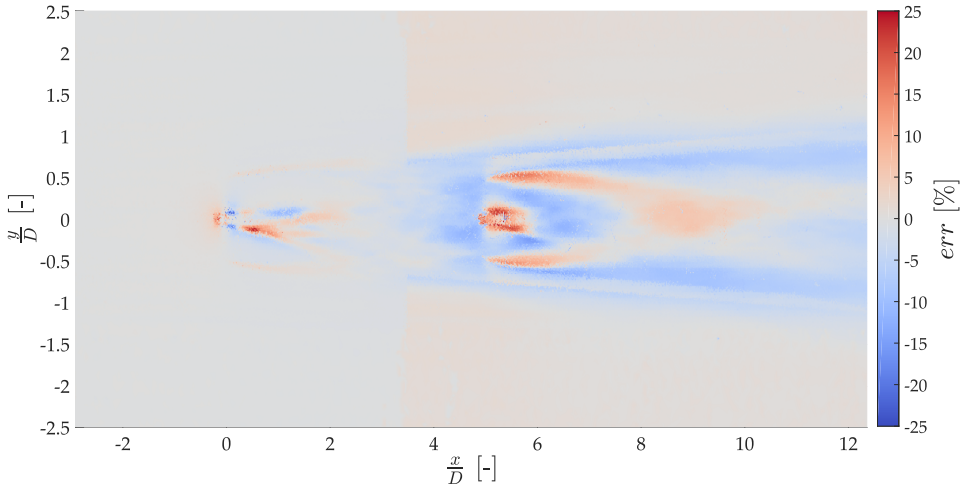


FIGURE 5.5: *Error field at the hub plane applying a 1D blended superposition method to the double aligned layout.*

the TSR may be not a fully representative indicator for waked turbines and, indeed, they showed how regulating the downwind turbines with a tip-speed ratio based on the average rotor speed leads to degraded performances, which may be improved changing the control algorithm. Eventually, while the second turbine is modelled by a SWTS with the same turbulent freestream conditions, a discrepancy between the superposed and reference cases appears due to different *shape* of the inflow profile. Still, this phenomenon would induce only a local error in the flow field and in the far-wake the velocity profile in both cases should be similar. That is particularly true in the aligned geometry: as one can see the error in the tip regions almost disappear at nearly $4D$ downstream the second turbine, thanks to the enhanced recovery rate of the farm layout.

Figure 5.8 sums up the RAE_{wake} of all the methods at the various horizontal planes.

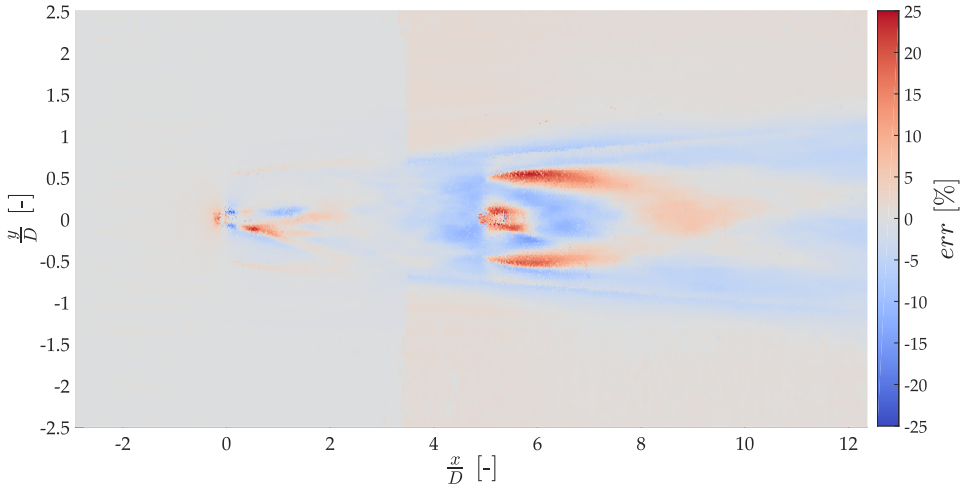


FIGURE 5.6: *Error field at the hub plane applying a 2D blended (y^4) superposition method to the double aligned layout.*

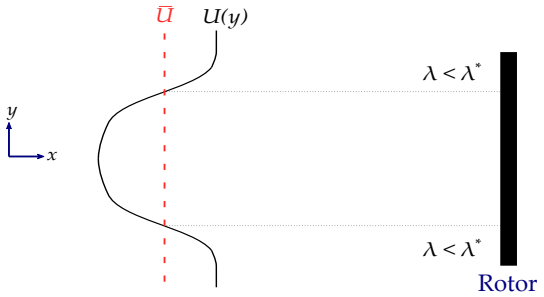


FIGURE 5.7: *λ values with a gaussian inflow and a regulation strategy based on the average wind speed.*

First of all, it emerges how the proposed method - in all of its derivations - achieves better performances than the usual schemes in the hub plane, which is the one used for the ROM extraction. Indeed, at hub height, the relative error in the wake region is almost halved, going from the 6.5% of the *maxdef* to about 3.3% for the 2D-blended schemes. Moreover, it should be highlighted that the performance of the quadratic method - often implemented in the commercially available software - is even worse than that of the maximum deficit. This confirms that the typical superposition procedure employed with the engineering wake models cannot be directly extended for superposing CFD data.

Secondly, one can notice that introducing a spanwise blending a better accuracy is attained, even though the main reason behind the introduction of the blending was the prevention of a discontinuous flow field.

Then, the Figure 5.8 points up how the proposed method, with and without blending, shows a parabolic trend in z, though with opposite concavity than that derived from the *lin*, *quad* and *maxdef* schemes. While the latter behaviour was due to the inability of the usual methods in predicting the increase recovery rate caused by the interactions of the wake (hence, inducing a large error near the hub plane), in the former case the coefficient α accelerates the wake recovery. Though this situation is beneficial in the hub plane, in the farther planes the α-schemes would overpredict

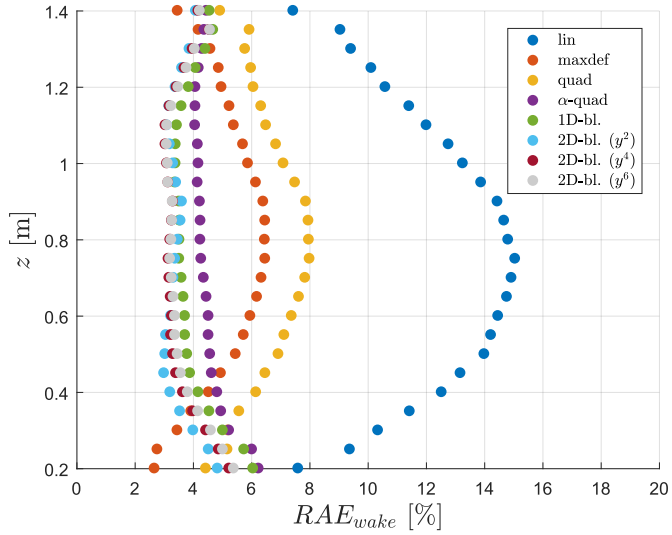


FIGURE 5.8: *Relative absolute errors in the wake region for the double aligned layout.*

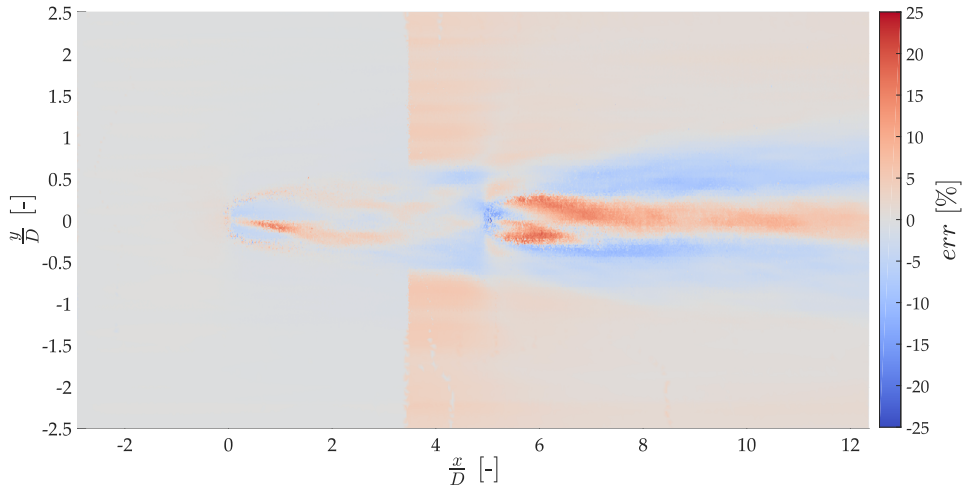


FIGURE 5.9: *Error field at the* $z = 0.3$m *plane applying a maximum deficit superposition method to the double aligned layout.*

the velocity, completely overturning the scenario. The different behaviour just explained is exemplified by Figure 5.9 - which depicts the error at the plane near the tip applying a maximum deficit scheme - and Fig.5.10, where the 2D-blended method has been adopted. In this regard, it looms ahead the opportunity to augment the method with a z-blending, if a better superposed flow field in the whole 3D domain is desired. As that is not the case for the present study, and the focus is mainly on the hub plane, no further modifications are added to the method.

The discussion has covered up to now only the superposition of the streamwise velocity. In the present study, the spanwise and vertical velocities for the wind farm have been obtained scaling properly the corresponding flow fields of the SWTS. Though this may seem a simplistic approach, the RAE in the entire domain are actually small: 0.76% for what concerns the spanwise velocity and 0.69% for the vertical

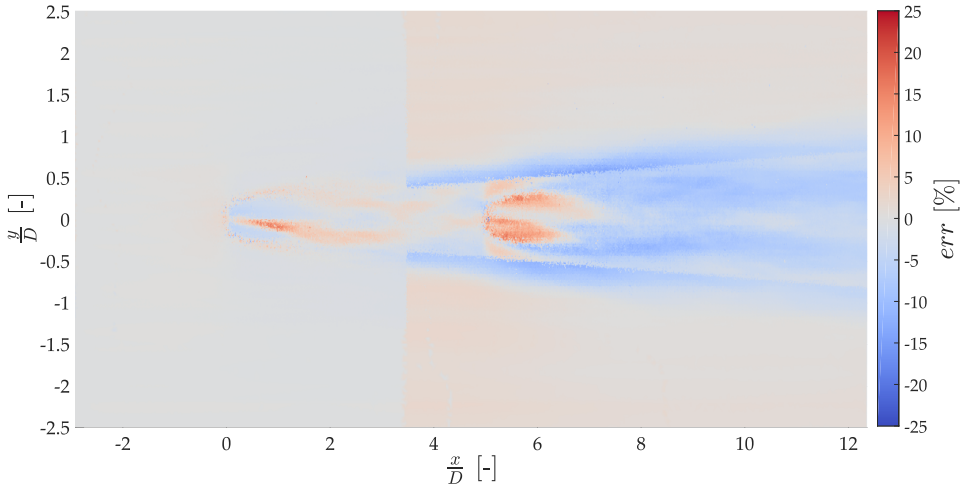


FIGURE 5.10: *Error field at the* $z = 0.3$m *plane applying a 2D blended* (y^4) *superposition method to the double aligned layout.*

one. Hence, the rest of the study would be focused on the application of the superposition framework to the streamwise velocity, which is moreover the component whose use is advocated for the ROM extraction whenever processing all the velocity components becomes unfeasible.

5.1.2 Triple Aligned Layout

With a triple aligned layout one can test the behaviour of $F_{1D}(\tilde{x})$ in the induction domain of the third turbine. In Figures 5.11, 5.12, 5.13 are shown the error fields at hub plane obtain respectively with the *maxdef*, *quad* and *2D-blended* (y^4) schemes. Analysing the figures one can notice how the out-of-wake correction leads to a better prediction of the flow field in the domain of influence of the third turbine.

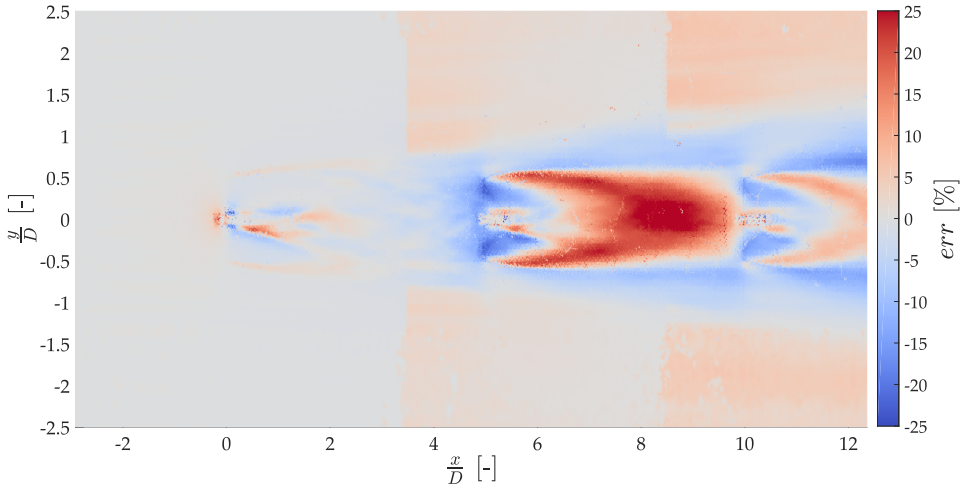


FIGURE 5.11: *Error field at the hub plane applying a maximum deficit superposition method to the triple aligned layout.*

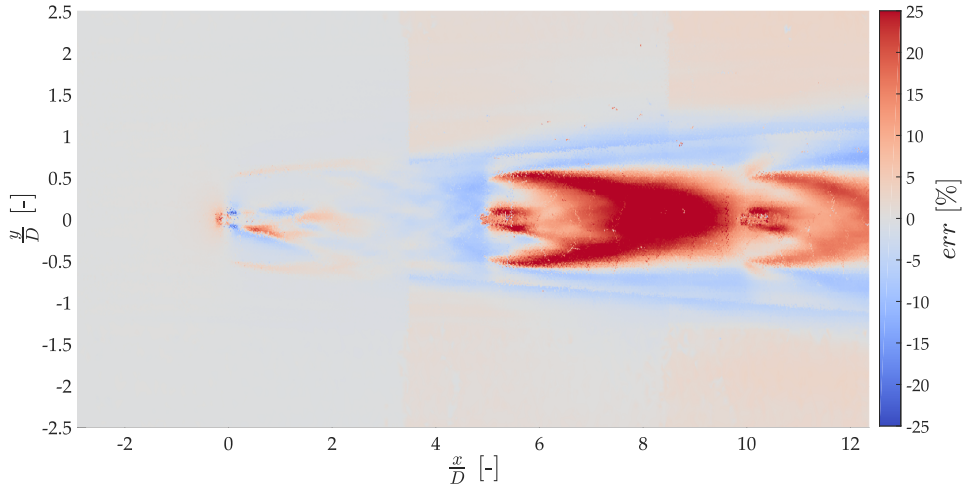


FIGURE 5.12: *Error field at the hub plane applying a quadratic superposition method to the triple aligned layout.*

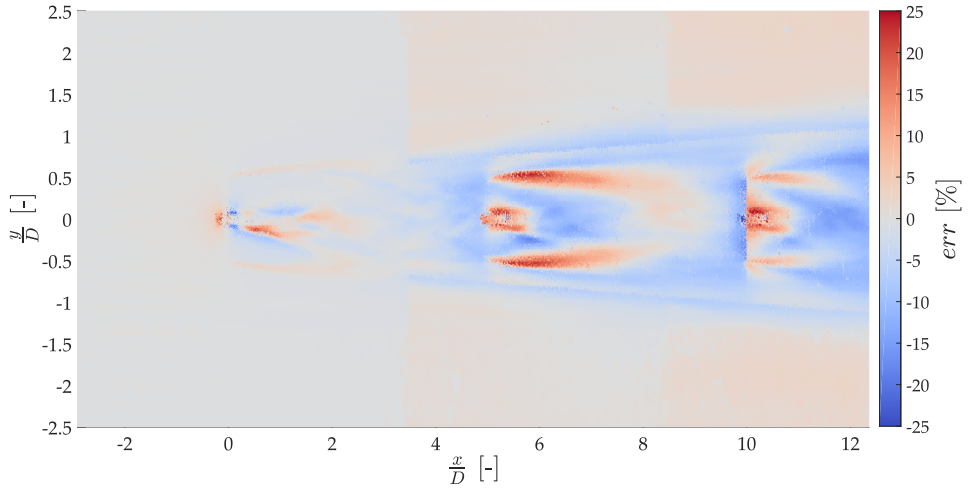


FIGURE 5.13: *Error field at the hub plane applying a 2D blended (y^4) superposition method to the triple aligned layout.*

The errors obtained in the wake region of the domain are displayed in Figure 5.14.

As it can be seen, in this case the improvement showed moving from the standard schemes to the proposed framework is reduced. This is mostly due to the computational domain, as unfortunately allows only a study in the near-wake of the third turbine.

5.1.3 Double Misaligned Layout

Starting from the considerations of Gunn et al., 2016, it is expected that introducing a lateral spacing ($S_{lat} = 0.5D$ in this layout) the scenario would qualitatively vary. Indeed, the authors showed that with a $S_{lat} = 1D$, the recovery rate predicted by

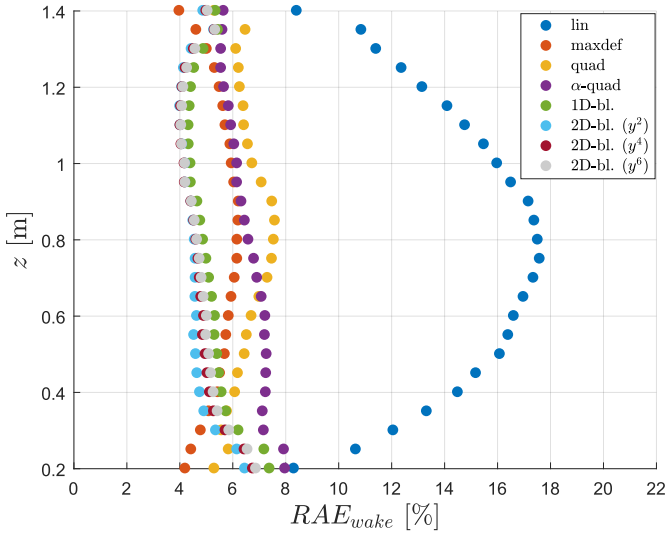


FIGURE 5.14: *Relative absolute errors in the wake region for the triple aligned layout.*

the quadratic would even be faster than the actual one, changing diametrically the behaviour of the method compared to before.

With the *quad* method the deficit is again overestimated (Fig.5.15) and, this time, using the same mixing coefficient as before ($\alpha = 0.6$, as obtained from Eq.3.35) does not seem to improve the error, leading to a large overprediction of the velocity distribution, as Figure 5.16 clearly shows.

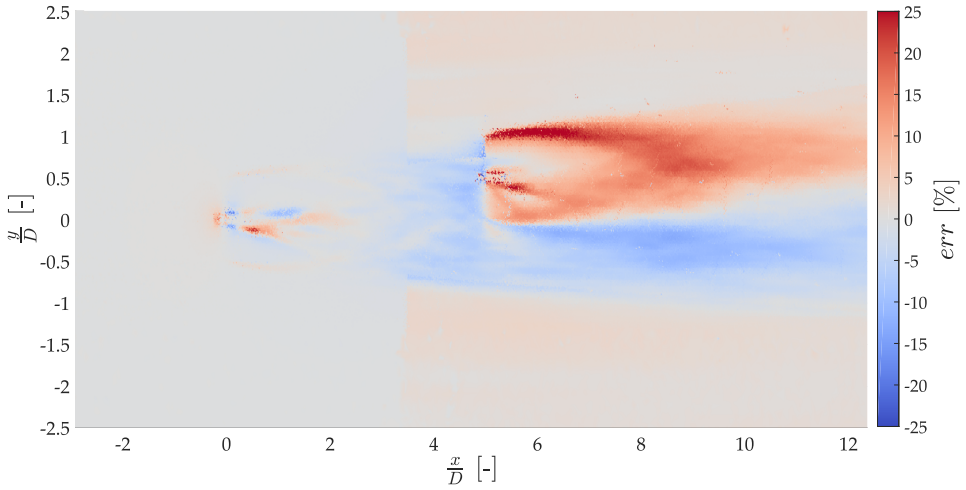


FIGURE 5.15: *Error field at the hub plane applying a quadratic superposition method to the double misaligned layout.*

The theoretical results highlight that:

- with $S_{lat} = 0$ the recovery rate is faster than that predicted by the quadratic method ($\Rightarrow \alpha < 1$)

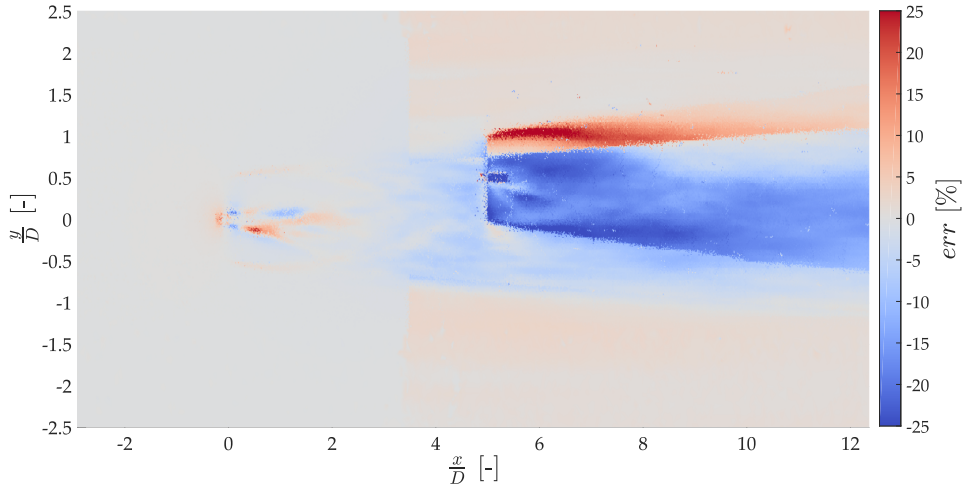


FIGURE 5.16: *Error field at the hub plane applying a α-quadratic superposition method ($\alpha = 0.6$) to the double misaligned layout.*

- with $S_{lat} = 1D$ the recovery rate is slightly slower than that predicted by the quadratic method ($\Rightarrow \alpha > 1$).

Hence, it is somehow expected that with a lateral spacing of $0.5D$ the situation would lie in between those just listed. In fact, the previous Figures point up how the optimal value of α would belong to $0.6 < \alpha_{opt} < 1$. In this regard, an optimization procedure has been carried on, finding the value of α which minimizes the RAE_{wake}. The range spanned consists in $\alpha = 0.6 : 0.05 : 1$.

The optimization results in a $\alpha_{opt} = 0.85$. The flow field for this method is depicted in Figure 5.17 (*α-quad* method) and Fig.5.18 (for the 2D-blended method).

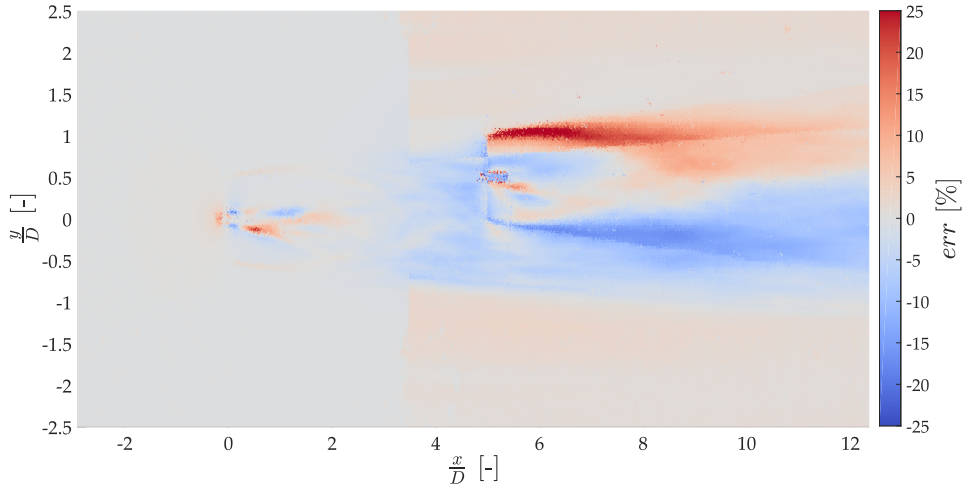


FIGURE 5.17: *Error field at the hub plane applying a α-quadratic superposition method ($\alpha = \alpha_{opt} = 0.85$) to the double misaligned layout.*

Fig.5.19 shows the error of all the schemes, including those belonging to the class of the α-frameworks, where the value $\alpha = \alpha_{opt}$ has been employed.

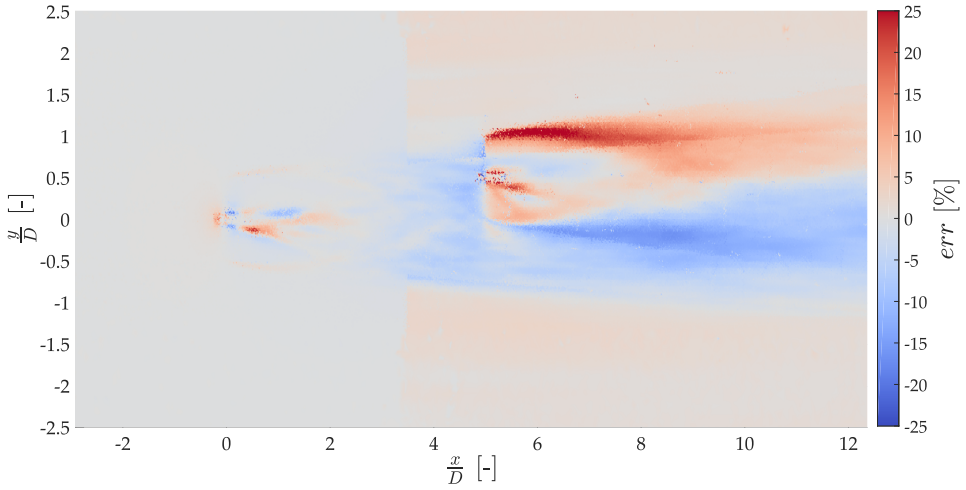


FIGURE 5.18: *Error field at the hub plane applying a 2D blended (y^4) superposition method to the double misaligned layout.*

Clearly, one can notice how the difference between the errors given by the *quad* and the blended methods is reduced, compared to the aligned cases. More than to a deficiency of the proposed scheme, this situation can be traced back to a better performance of the *quad* method: while the hub plane RAE_{wake} for the DA layout was equal to 8%, in the DM case its value is nearly 5.6%.

Moreover, it can be observed how - amongst the standard schemes - the quadratic method performs better than the *maxdef* one in this layout. This can be again explained by the same experimental observation mentioned before; indeed, as the *maxdef* predicts a smaller deficit than that returned by the linear and quadratic methods, its performance within the aligned layout is the best among the three. For a misaligned geometry however this is not true anymore: the recovery is slower and

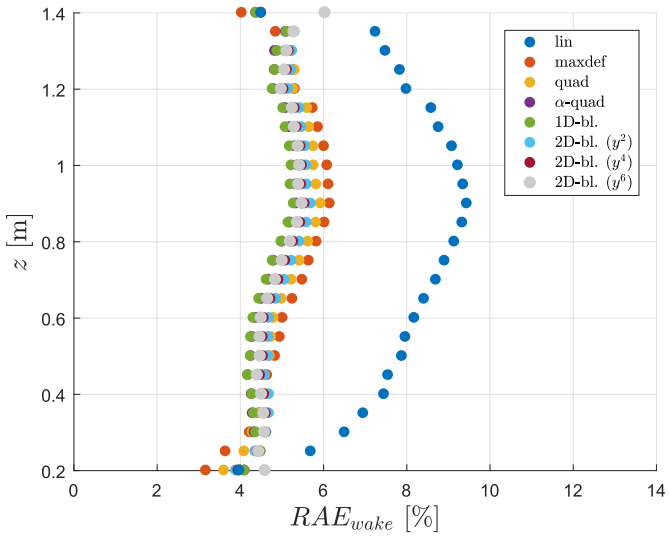


FIGURE 5.19: *Relative absolute errors in the wake region for the double misaligned layout.*

more accurately predicted by the *quad* scheme.

Examining the error of the superposed flow field, one can again notice a localized region of error behind the tip of the second turbine. However, though for the aligned layout this discrepancy smooths down pretty quickly travelling downstream, in this case the error visibly affects the domain even up to $7D$ of distance, as the recovery rate of the wake is slower than in the aligned case.

5.1.4 Triple Misaligned Layout

Even in this case an optimization procedure was followed, using the same range as before. One should expect that if the physics is correctly modelled the optimal value of the mixing coefficient should be the same in both the DM and TM layouts. Indeed, it has been found an optimal value of $\alpha_{opt} = 0.90$, slightly higher than the coefficient found for the double misaligned case. This small discrepancy can be also explained looking at the adopted domain: as only the near-wake of the third turbine is concerned and as the quadratic method ($\alpha = 1$) ensures low errors in this region, the optimization procedure would return an α_{opt} that would be closer to 1 than the value expected analysing also the far-wake domain.

The relative absolute errors of the methods are portrayed in Figure 5.20.

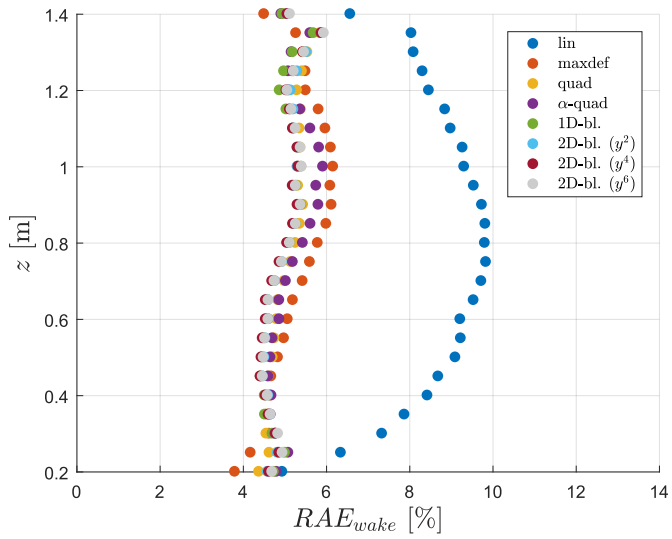


FIGURE 5.20: *Relative absolute errors in the wake region for the triple misaligned layout.*

In this case the advantage given by proposed method is not appreciable, partly due to the limitations in the considered domain and partly because the quadratic method is actually well performing in the misaligned case.

5.2 Prediction of Power Outputs

The aim of this paragraph is to find a reliable method able to predict the power outputs of the turbines, generating the output vector for the ROM extraction.

The reference values of the power have been extracted by the TA and TM simulations and are reported in Table 5.2 and 5.3, respectively. Notice how the power of the third turbine is higher than that of the second one, meaning that the deficit

TABLE 5.2: *Time-averaged power outputs for the triple aligned turbines.*

$\mathbf{P_1}$ [W]	$\mathbf{P_2}$ [W]	$\mathbf{P_3}$ [W]
42.24	8.77	14.75

TABLE 5.3: *Time-averaged power outputs for the triple misaligned turbines.*

$\mathbf{P_1}$ [W]	$\mathbf{P_2}$ [W]	$\mathbf{P_3}$ [W]
42.22	19.28	28.07

impinging on the rotor of the former machine is actually lower than that on the latter. Moreover, in the aligned case, the power of the downstream turbines are inferior than those of the corresponding machines in the misaligned layout. Indeed, as the upstream turbine has a null yaw angle, the maximum deficit is felt on the centerline, along which the turbines of the TA layout are located.

The accuracy of the methods is measured on the basis of the percentage error:

$$PE = \frac{\left|P_{k,\text{sup}} - P_{k,\text{ref}}\right|}{P_{k,\text{ref}}} \cdot 100\,, \tag{5.4}$$

given the power on the k-th turbine obtained from the reference case ($P_{k,\text{ref}}$) and from the superposition approach ($P_{k,\text{sup}}$).

Within the same superposition framework adopted to get the flow field in the whole domain, the power can be readily obtained from the usual relation in Eq.(5.5).

$$P_k = \frac{1}{2}\rho A U_{k,\text{RA}}^3 C_P(U_{k,\text{RA}}) \tag{5.5}$$

In the previous formula the standard air density $\rho = 1.225\text{kg}\,\text{m}^{-3}$ is assumed, A stands for the rotor area, $U_{k,\text{RA}}$ is the velocity averaged on the rotor disc and C_P is the power coefficient computed given the incoming wind speed, in agreement with Fig.4.2. To obtain the rotor-averaged velocity two different approaches have been followed: in the first one the speed has been estimated only using the velocity measurements belonging to the hub plane; in the second way, information coming from the whole flow field was used. Finally, recall that only the third power output depends actually on the superposition method, indeed:

- P_1 is obtained from the freestream velocity U_∞ of the farm
- P_2 is computed from S-I4 properly averaging the velocity distribution at a $5D$ downstream distance.

Hence, both P_1 and P_2 would have the same values for the superposition methods.

For the aligned layout, the estimated power using the hub plane (*hp*) measurements are shown in Table 5.4, whereas Table 5.5 presents the power computed processing data from all the horizontal planes (all planes, *ap*).

TABLE 5.4: *Estimated power outputs for the aligned layout using hub plane measurements.*

	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$\mathbf{P_1}$ [W]	42.17	42.17	42.17	42.17	42.17	42.17	42.17	42.17
$\mathbf{P_2}$ [W]	6.41	6.41	6.41	6.41	6.41	6.41	6.41	6.41
$\mathbf{P_3}$ [W]	1.19	6.27	4.52	13.83	13.79	12.40	13.47	13.70

TABLE 5.5: *Estimated power outputs for the aligned layout using all planes measurements.*

	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$\mathbf{P_1}$ [W]	41.81	41.81	41.81	41.81	41.81	41.81	41.81	41.81
$\mathbf{P_2}$ [W]	7.51	7.51	7.51	7.51	7.51	7.51	7.51	7.51
$\mathbf{P_3}$ [W]	1.56	7.35	5.36	14.72	14.69	13.51	14.45	14.62

The errors of the first two turbines are reported in Table 5.6, whereas Table 5.7 shows the errors of the different methods on the power prediction of the third turbine. From Tab.5.6 one can notice a high error on the power prediction of the second

TABLE 5.6: *Percentage errors of the predicted power for the first and second turbines in the aligned layout.*

	WT1	WT2
PE_{hp} [%]	0.17	26.92
PE_{ap} [%]	1.03	14.42

TABLE 5.7: *Percentage errors of the predicted power for the third turbine in the aligned layout.*

	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$PE_{P_3,hp}$ [%]	91.90	57.50	69.32	6.25	6.47	15.95	8.67	7.08
$PE_{P_3,ap}$ [%]	89.41	50.16	63.67	0.21	0.42	8.40	2.03	0.83

turbine. It is postulated that this can be due to how the power output is estimated: in fact, within the SOWFA framework the power is not computed by means of Eq.(5.5), but it is evaluated by an integral relationship, as in Eq.(5.6), referred to the simplest case of non-yawed turbine and symmetric inflow:

$$P_k = \frac{1}{2}\rho A \int_0^R U_k(r)^3 C_{P/r}(U_k, r)\,, \mathrm{d}r \tag{5.6}$$

where power is obtained integrating along the blade radius R the product of the local velocity cubed and the C_P distribution on the rotor span, which is similar to those portrayed in Fig.5.21. However, no computations were made as the actual distribution of the power coefficient along the span which accounted for drag terms was not available. For what concerns the third turbine, as it can be seen, the proposed method can achieve high accuracy for the power prediction, especially if the velocity distribution at different z is considered. The standard methods instead return a inaccurate power output, hindering the reconstruction of the $\mathcal{Y}$ vector.

Similarly, the misaligned case is analysed. Table 5.8 and 5.9 summarise the power predicted by the superposition schemes and Tab.5.10 and 5.11 display the percentage errors for the first two turbines and the third one, respectively.

The data underlines that also in this case the power output of the second turbine is affected by an error, which can be again due to the way the power is computed. However, even the predicted power of the third turbine significantly differs from the reference value. This can be explained observing for instance Fig.5.18, which shows

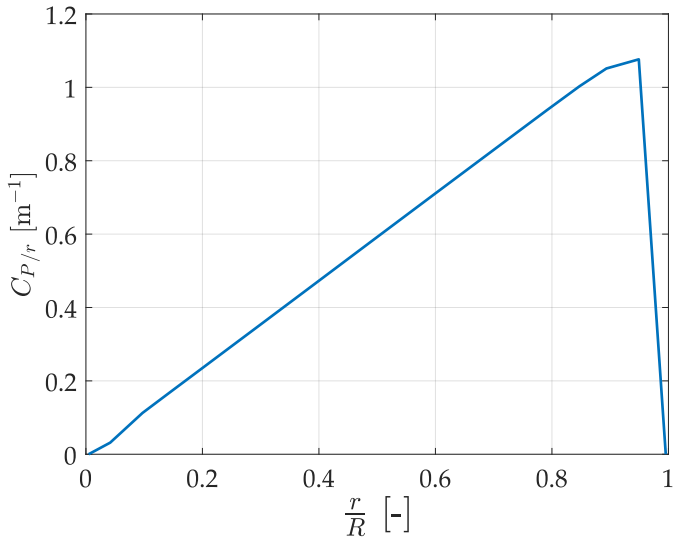


FIGURE 5.21: *Example of a $C_{P/r}$ trend over the blade span (no drag).*

TABLE 5.8: *Estimated power outputs for the misaligned layout using hub plane measurements.*

	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$\mathbf{P_1}$ [W]	42.15	42.15	42.15	42.15	42.15	42.15	42.15	42.15
$\mathbf{P_2}$ [W]	15.73	15.73	15.73	15.73	15.73	15.73	15.73	15.73
$\mathbf{P_3}$ [W]	13.71	16.52	16.13	18.47	18.46	17.92	18.27	18.38

TABLE 5.9: *Estimated power outputs for the misaligned layout using all planes measurements.*

	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$\mathbf{P_1}$ [W]	41.79	41.79	41.79	41.79	41.79	41.79	41.79	41.79
$\mathbf{P_2}$ [W]	18.23	18.23	18.23	18.23	18.23	18.23	18.23	18.23
$\mathbf{P_3}$ [W]	16.33	18.63	18.30	20.19	20.18	19.74	20.01	20.12

TABLE 5.10: *Percentage errors of the predicted power for the first and second turbines in the misaligned layout.*

	WT1	WT2
PE_{hp} [%]	0.16	18.42
PE_{ap} [%]	1.01	5.42

how, after the superposition of two turbine wakes, the rotor of an ideal misaligned third turbine would be invested by a wind speed lower than its actual value in a TM layout. From the same figure, it is also appreciable how the error is mainly originated by the tip region, for the reasons discussed before.

Overall, the prediction of the output by means of the superposition procedure appears to behave more accurately when the velocity distributions at several heights

TABLE 5.11: *Percentage errors of the predicted power for the third turbine in the misaligned layout.*

	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$PE_{P_3,hp}$ [%]	51.15	41.13	42.54	34.20	34.22	36.15	34.90	34.51
$PE_{P_3,ap}$ [%]	41.82	33.63	34.78	28.07	28.09	29.67	28.70	28.31

are available. However, the scheme adopted in the code seems to oversimplify the physics of the problem, causing a deviation of the predicted outputs from the reference values.

All the data obtained with the superposition framework are then compared with the power predictions attained with FLORIS, as discussed in Sec.3.2.3. As a first step, it was necessary to tune the adopted wake models to the available dataset. This procedure was done in agreement with (Campagnolo et al., 2019). Indeed, after the extraction from the SWTS of hub height velocity profiles at a distance from $5D$, $7.5D$ and $10D$, a cost function (SRE) has been defined as

$$SRE = \sum_{i=1}^{M} \sum_{d} \sum_{j=1}^{N^{i,d}} \left(\frac{\hat{v}_j^{i,d} - \tilde{v}_j^{i,d}}{\tilde{v}_j^{i,d}} \right)^2 , \tag{5.7}$$

where:

- M is the number of observation, namely the SWTS used
- d is the downwind distance of the observations ($d = 5D, 7.5D, 10D$)
- $N^{i,d}$ is the number of measurement points for the i-th observation at a d distance
- $\tilde{v}_j^{i,d}$ is the velocity experimentally measured at the point j
- $\hat{v}_j^{i,d}$ is the velocity predicted by the numerical model at the point j.

Yet, as the measurements are typically affected by noise and other errors, the output of this first minimization has been employed as a guess for a maximum likelihood estimation (MLE) of the model coefficients. This second cost function reads:

$$J = \frac{M}{2} \ln(2\pi) + \frac{M}{2} \ln \det(\boldsymbol{R}) + \frac{1}{2} \sum_{i=1}^{M} \boldsymbol{r}_i^T \boldsymbol{R}^{-1} \boldsymbol{r}_i \,, \tag{5.8}$$

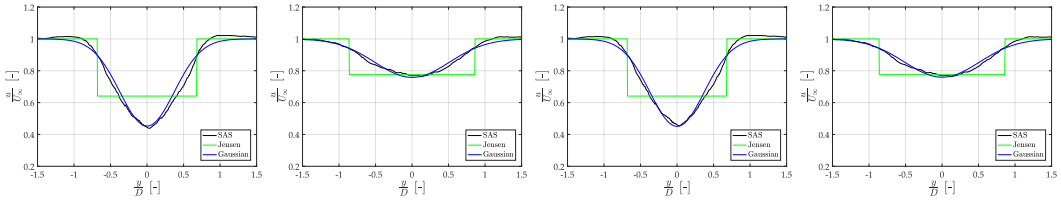
given the residual vector $\boldsymbol{r}_i$ of the i-th observation and the residual covariance matrix $\boldsymbol{R}$, computed as in the Eq.(5.9) and (5.10) respectively.

$$\boldsymbol{r}_i = \left[\dots, \left\| \left(\frac{\hat{\boldsymbol{v}}^{i,d} - \tilde{\boldsymbol{v}}^{i,d}}{\tilde{\boldsymbol{v}}^{i,d}} \right) \right\|, \dots \right]^T \qquad d = 5D, 7.5D, 10D \tag{5.9}$$

$$\boldsymbol{R} = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{r}_i \, \boldsymbol{r}_i^T \,. \tag{5.10}$$

Both the two steps for the tuning have been carried out thanks to the `fminsearch` function of MATLAB. The procedure returns a value of $\kappa = 0.036$ for the Jensen wake model, which well agrees with the one recovered by Campagnolo et al. (2019), and

the set of tuned coefficients ($k_a = 0.0036, k_b = 0.0273, \alpha^* = -0.0032, \beta^* = 0.1668$). In Figure 5.22 are shown some examples of the tuned wake models, compared with the SAS profiles extracted from the SWTS.



(a) $x = 5D$, *S-I4.* (b) $x = 10D$, *S-I4.* (c) $x = 5D$, *S-I13.* (d) $x = 10D$, *S-I13.*

FIGURE 5.22: *Comparison between SAS profiles and the tuned Jensen and Gaussian wake models.*

After properly tuning the wake models, the TA and TM layouts were studied in FLORIS. In this regard, the *α-quad* method has been implemented in the FLORIS framework. Recall that the latter evaluates the flow field in the 3D domain and, as a consequence, the results should be compared to those obtained by the superposition scheme utilizing the velocity distributions at all the horizontal planes.

The values of the power predicted by FLORIS for the aligned layout are shown in Table 5.12. Confronting this data with the reference values in Tab.5.2 the percentage errors are computed. Table 5.13 emphasizes how the Jensen wake model

TABLE 5.12: *Estimated power outputs for the misaligned layout using FLORIS.*

	Jensen Wake		**Gaussian Wake**	
	quad	*α-quad*	*quad*	*α-quad*
$\mathbf{P_1}$ [W]	41.44	41.44	41.44	41.44
$\mathbf{P_2}$ [W]	9.41	9.41	9.51	9.51
$\mathbf{P_3}$ [W]	6.64	15.99	7.09	16.12

TABLE 5.13: *Percentage errors for the predicted powers in the misaligned layout using FLORIS.*

	Jensen Wake		**Gaussian Wake**	
	quad	*α-quad*	*quad*	*α-quad*
PE_{P_1} [%]	1.89	1.89	1.89	1.89
PE_{P_2} [%]	7.26	7.26	8.40	8.40
PE_{P_3} [%]	54.97	8.43	51.92	9.31

underestimates the power outputs with respect to the Gaussian model. Comparing the FLORIS results with those obtained by means of the superposition scheme one can notice how even the power for the first two turbines have different values. For what concerns the first turbine power, the small difference may be explained due to the slightly different inflow condition: though the inflow velocity and the power law exponent in FLORIS have been taken computing their values from the data returned by TurbSim (i.e., $U_\infty = 5.562\text{m}\,\text{s}^{-1}$ and $\alpha_{PL} = 0.14$), the latter software adopts actually a combination of a power-law and logarithmic profiles (NREL, 2019b) and that

may induce the marginal discrepancies shown in the power. For the output of the second turbine, the difference between the power obtained post-processing SWTS data and that returned by FLORIS increases, up to 7%. Though this is quite a significant value, one can notice that the rotor averaged speed for the second rotor is equal to $U_{2,\mathrm{RA}} = 3.55\mathrm{m\,s^{-1}}$ in FLORIS and $U_{2,\mathrm{RA}} = 3.33\mathrm{m\,s^{-1}}$ for the other case, which corresponds to a nearly 6% difference that should be due the errors in the tuning procedure. Finally, for the third turbine, the tables show a good agreement of the wake models with the reference case, when the α-quadratic scheme is followed.

Considering then the misaligned case, FLORIS returns the power measurements displayed in Table 5.14. The respective percentage errors are computed in Tab.5.15.

TABLE 5.14: *Estimated power outputs for the misaligned layout using FLORIS.*

	Jensen Wake		**Gaussian Wake**	
	quad	*α-quad*	*quad*	*α-quad*
$\mathbf{P_1}$ [W]	41.44	41.44	41.44	41.44
$\mathbf{P_2}$ [W]	16.70	16.70	21.70	21.70
$\mathbf{P_3}$ [W]	15.65	17.99	21.38	23.50

TABLE 5.15: *Percentage errors for the predicted powers in the misaligned layout using FLORIS.*

	Jensen Wake		**Gaussian Wake**	
	quad	*α-quad*	*quad*	*α-quad*
PE_{P_1} [%]	1.84	1.84	1.84	1.84
PE_{P_2} [%]	13.38	13.38	12.56	12.56
PE_{P_3} [%]	44.24	35.90	23.82	16.27

Again, the Gaussian wake model appears to be better performing than the Jensen one and the percentage errors seems to be in agreement with those previously computed.

In conclusion, the reconstruction of the output vector $\mathcal{Y}$ appears to be a complicate matter and while the implemented superposition scheme shows some encouraging results, different flaws are still present, especially for the misaligned case. The discrepancies between the reference values and the ones obtained via superposition suggest the adoption of a more complex framework, with a more refined degree of modeling. In this scenario, FLORIS does not seem an interesting alternative to the superposition scheme implemented, as it does not involve a significant improvement of the accuracy and its extension to unsteady state of the systems appears excessively complicated.

5.3 Selection of the Projector

For what concerns the projector, three different choices have been adopted by the present work. The full-order system under examination has been obtained superposing different snapshots, mimicking the behaviour of a three aligned turbines array. As the focus of the paragraph stands solely in the selection of the projection

matrix, the snapshots have been superposed without actually studying the flow dynamics, in order to avoid to run further simulations and to parallelize the selection of the projector with the study of the superposition methods. Again, while the aim of the superposition schemes consists in finding the most consistent way to superpose SWTS in order to represent a larger flow field, hereby it is analysed how accurately a chosen matrix projects a full-system (which can even have no physical meaning) into a lower-order space, minimizing the loss of information.

The flow field has been obtained from the simulation with a yaw-dynamic input and an $I_0 = 0.06$ (Sec.4.2.2). Recall that the first 5 seconds were used to generate a time-averaged wake vector which has the shape:

$$\mathcal{X}_{ta} \in \mathbb{R}^{n_x \times 1},$$

where n_x is the number of desired states. Then, during the remaining 55s snapshots of the field were collected and assembled in a snapshot velocity matrix $\mathcal{V}$ with a number of columns equal to $n_s = 4622$, that is the number of effective snapshots used for the ROM extraction (indeed, with a sampling time $T_s = 0.012$s, $n_s T_s \approx 55$s).

The three turbines array was modelled as described in Table 5.16, namely it is assumed that the first turbine is operating with a yaw variable in time whereas the downwind machines are characterized by a constant null yaw angle. Hence, formally:

$$\text{Given a snapshot } i = 1 : n_s - 1,$$
$$\mathcal{X}(:, i) = \mathcal{V}(:, i)^{WT1} \oplus \mathcal{X}_{ta}^{WT2} \oplus \mathcal{X}_{ta}^{WT3},$$

where $\mathcal{X}$ represents the velocity snapshot matrix of the farm and the operator $\oplus$ indicates the superposition of the different flow fields.

TABLE 5.16: *Modelling choices for the three wind turbine for the selection of the projector.*

Modelling wake for WT1	Wake with an APRBS yaw, 55s long
Modelling wake for WT2	Time-averaged wake
Modelling wake for WT3	Time-averaged wake

Due to memory constraints, the states of the full-order system are the streamwise velocities at the grid points of the horizontal plane. Hence, the number of states n_x corresponds to the number of mesh points N.

Recalling that the IOROM extraction consists in a linearized procedure, it was also required to identify an initial equilibrium condition. In agreement with the modelling choices, the velocity snapshot $\mathcal{X}_{ini}$, obtained from the superposition of the time-averaged wake for the three turbines

$$\mathcal{X}_{ini} = \mathcal{X}_{ta}^{WT1} \oplus \mathcal{X}_{ta}^{WT2} \oplus \mathcal{X}_{ta}^{WT3},$$

has been determined as the equilibrium condition.

In this regard, the 2D-blended (y^4) superposition scheme was selected. The flow field taken as equilibrium condition is reported in Figure 5.23.

Then, the power output of the turbine collected during the simulation at varying yaw was taken as output vector for the farm, resulting in $\mathcal{Y} \in \mathbb{R}^{1 \times (n_s - 1)}$.

The first way to obtain a projector matrix is the standard procedure, i.e. an SVD is performed and the first modes are retained. The decomposition has been carried

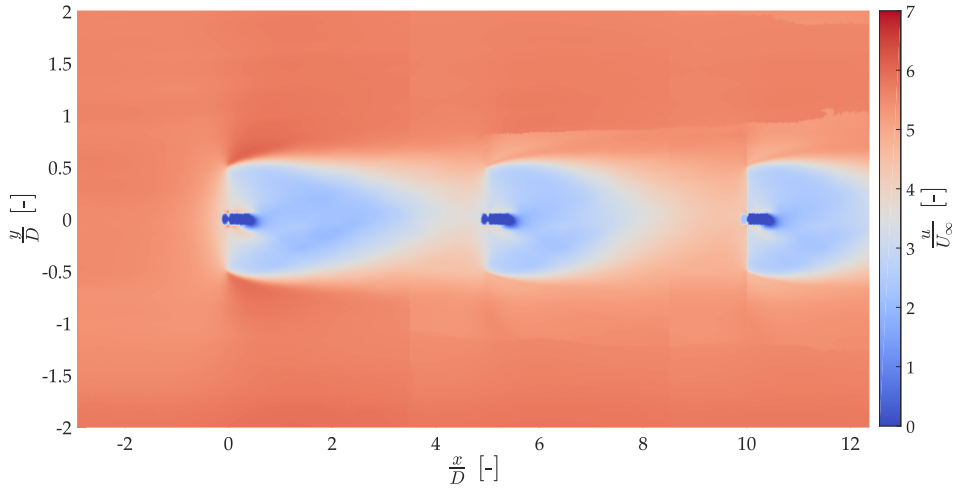


FIGURE 5.23: *Equilibrium condition for the modelled flow field.*

on by means of MATLAB's `svd` function. In particular, it has been performed an economy-size decomposition of the linearized matrix $\mathcal{X}_{lin} := \mathcal{X} - x_{ini}$. Notice that different results may be produced by distinct version of MATLAB, due to changes to its Math Kernel Library (MKL). All the data here reported were obtained with the MATLAB `2018b` version and the MKL `v.2018.0.1`. The resulting singular values are plotted in Figure 5.24a and their cumulative energy is shown in Fig.5.24b. The latter picture exemplifies how choosing the proper ROM order is not a trivial procedure and selecting a number of modes which constitutes nearly the 90% of the cumulative energy would generate a too high value of r. In agreement with (Fortes-Plaza et al., 2018), after the SVD the first 20 POD modes have been retained, resulting in a projector $P_{svd} = U_{20}$. Some of the selected POD modes can be observed in Fig-

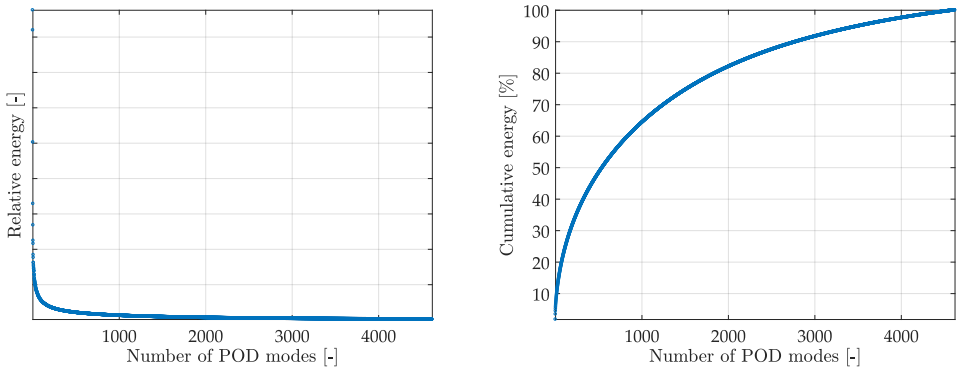


FIGURE 5.24: *Relative (a) and cumulative (b) energy of the POD modes of the superposed snapshot matrix.*

ure 5.25. Besides showing that the lower the mode index and the larger the flow spatial variations, the modal analysis underlines the importance of the dynamic of the downstream wakes. In fact, as those were modelled by a time-averaged quantity only a small modal contribution is present in the wake region of the second and third turbines, especially considering lower modes. Hence, to correctly capture the

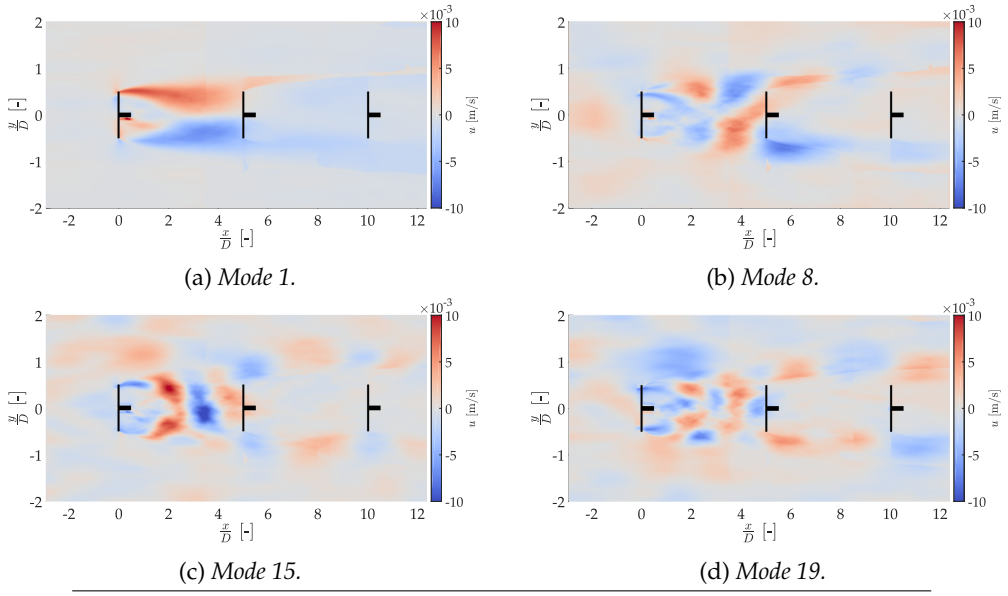


FIGURE 5.25: *Selection of the retained POD modes obtained for an SVD on the superposed snapshot matrix.*

flow dynamic it should be at least necessary to include the meandering of the downstream wakes.

The second choice for the projector is the interpolation of the POD modes of $\mathcal{V}$ into the domains of influence of the corresponding three turbines. In this regard, $\mathcal{V}$ has been previously decomposed with an SVD. Figure 5.26 displays the singular values and the corresponding cumulative energy. Notice how the trend of the energy

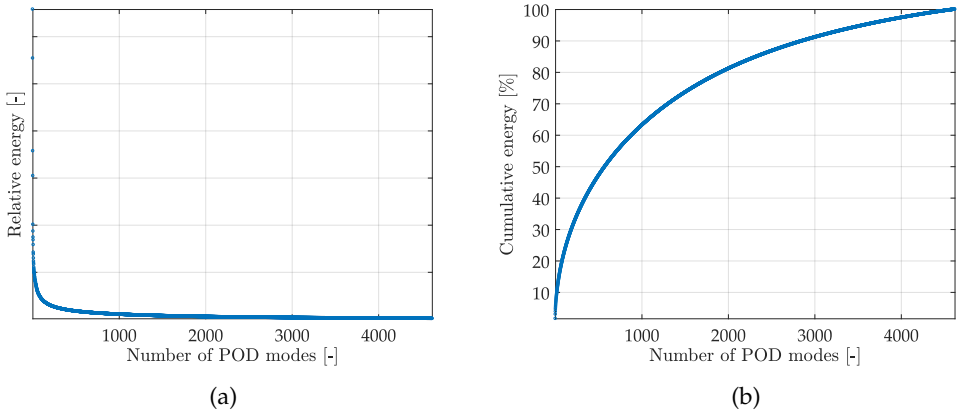


FIGURE 5.26: *Relative (a) and cumulative (b) energy of the POD modes of the snapshot matrix obtained from the yaw-varying simulation.*

is similar to those reported in Fig.5.24 and recall that in the latter case no actual physical meaning is associated to the snapshot velocity matrix. On the other hand, as the $\mathcal{V}$ has been obtained systematically sampling the flow field around a turbine, another requirement must be checked to ensure that the decomposition is physically consistent, i.e. the frequency content of the retained POD modes (stored in the right-singular vectors matrix) must comply with the Nyquist sampling theorem. Indeed, a discrete Fourier transform applied for the 20 modes revealed a significant frequency content only up to 20Hz, below the Nyquist frequency of nearly 42Hz associated to

the sampling period $T_s = 0.012$s. The dynamic of the modes is therefore properly captured.

Still, the just defined projector is not orthogonal and, as the third possible choice, another projector P_{MGS} is obtained applying the MGS algorithm to P_{noMGS}.

In Figure 5.28 a comparison between corresponding modes in the *noMGS* and *MGS* cases is shown. While in the former the modes are simply interpolated in the proper domain parts, in the latter case the orthogonalization procedure introduces some modal oscillations in region which are not actually influenced by the turbine (i.e. upstream the induction region). These oscillations become greater for higher POD modes.

After defining three different ways to compute P, the full-order system is projected onto a lower-order subspace. One can therefore analyse the performance of the three ROMs, looking at their predicted power outputs in Figure 5.27. Overall, the ROM extracted by means of the standard SVD approach appears to be better performing but it is even more remarkable how the outputs predicted by the assembled models coincide and incidentally the performance are not overly degraded. Hence, the MGS algorithm does not seem to influence the behaviour of the model but this scenario may be only a special case, due to the quite low conditioning number of P_{noMGS}, $cond(P_{noMGS}) \approx 3 \cdot 10^1$. For completeness, the graph shows also the power predicted by the ROM generated by the SVD method augmented with a Kalman filter (KF), which feeds actual power measurement back to the system. In this regard, the process noise and measurement noise covariance matrices have been defined according to (Fortes-Plaza et al., 2018). The time-averaged percentage errors

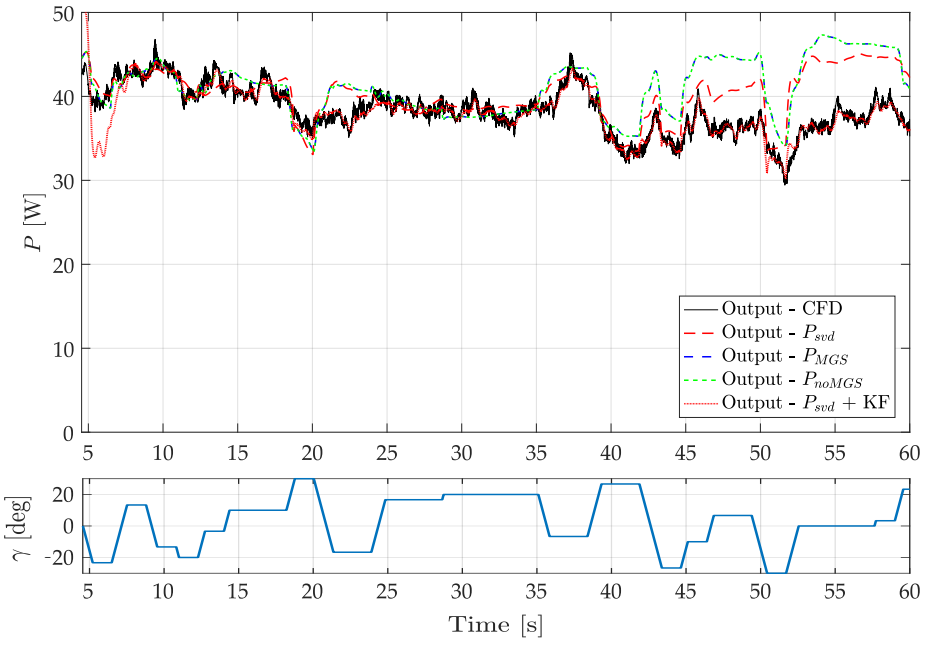
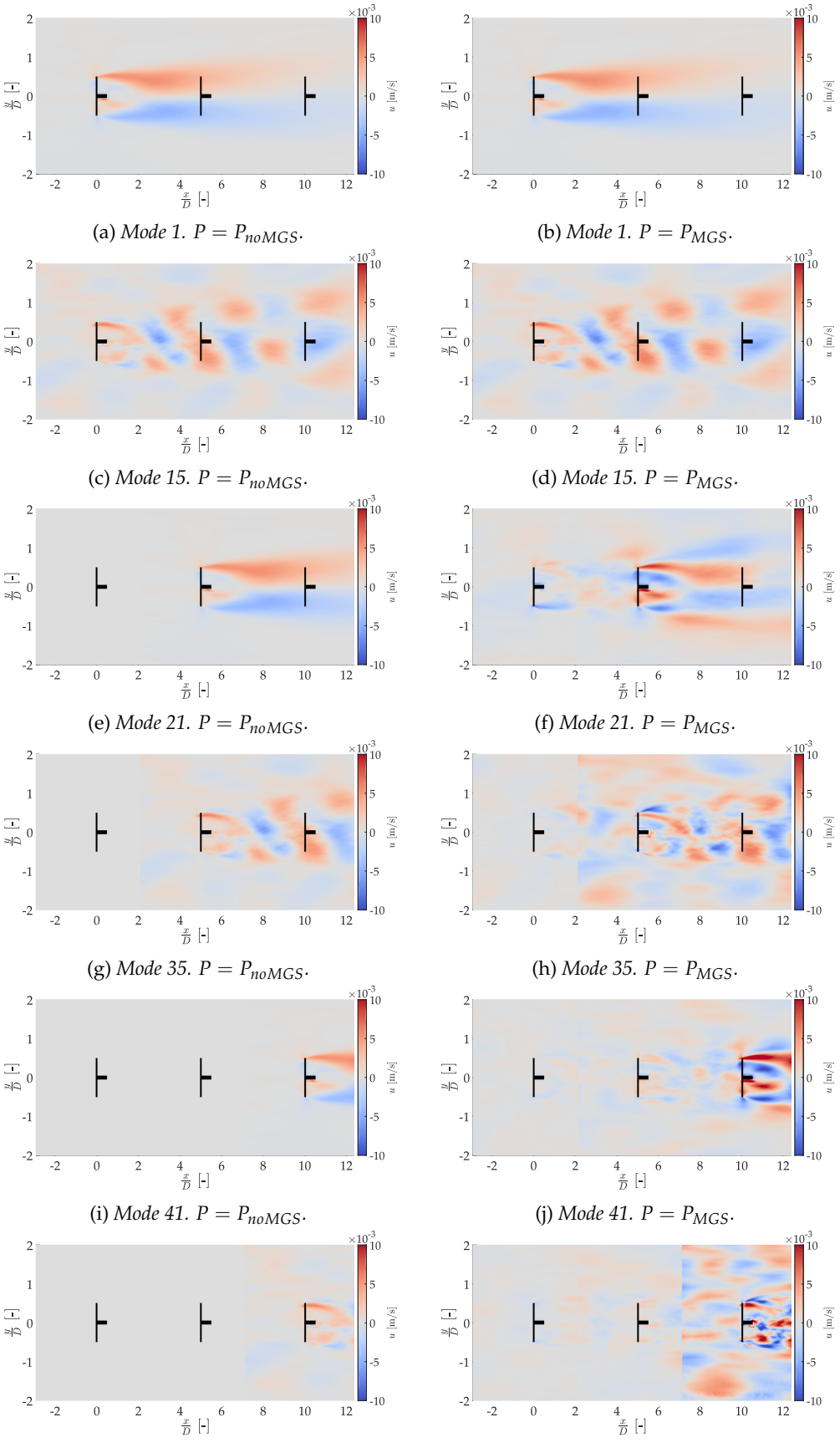


FIGURE 5.27: *Predicted power outputs for the different ROMs extracted. The input yaw signal is kept as a reference.*

between CFD and ROM-predicted power outputs are equal to 6.48% using a SVD method, 8.66% adopting both P_{noMGS} and P_{MGS} as projectors and 2.09% augmenting the SVD-based ROM with a Kalman filter.

Lastly, to benchmark the computational cost of the techniques, it is studied the behaviour of the number of floating point operations (FLOPS) needed by each of



(a) *Mode 1.* $P = P_{noMGS}$.

(b) *Mode 1.* $P = P_{MGS}$.

(c) *Mode 15.* $P = P_{noMGS}$.

(d) *Mode 15.* $P = P_{MGS}$.

(e) *Mode 21.* $P = P_{noMGS}$.

(f) *Mode 21.* $P = P_{MGS}$.

(g) *Mode 35.* $P = P_{noMGS}$.

(h) *Mode 35.* $P = P_{MGS}$.

(i) *Mode 41.* $P = P_{noMGS}$.

(j) *Mode 41.* $P = P_{MGS}$.

(k) *Mode 55.* $P = P_{noMGS}$.

(l) *Mode 55.* $P = P_{MGS}$.

FIGURE 5.28: *Comparison between the retained POD modes for the assembled projector, without* (left) *and with* (right) *MGS-orthogonalization.*

them to generate a projector for a wind farm with N_{WT} turbines. In doing so, one should also recall that the methods operate with matrices of different dimensions, namely:

- The SVD is performed on a N-by-n_s matrix, $\mathcal{X}$
- The assembly of the POD modes, with or without a MGS orthogonalization, works with a N-by-$(r\,N_{WT})$ matrix, Q, assuming that for each SWTS the same r number of POD modes have been selected.

Then, following the example of Trefethen and Bau (1997), the SVD algorithm would entail a number of FLOPS:

$$F_{svd,1} = 2Nn_s^2 + 11n_s^3 \tag{5.11}$$

$$\vee$$

$$F_{svd,2} = 4Nn_s^2 - \frac{4}{3}n_s^3\,, \tag{5.12}$$

where two different estimates are provided as the singular-value decomposition consists in an iterative process (hence, the number of FLOPS may vary). Instead, constructing the projector stacking the POD modes together (without orthogonalizing them subsequently) requires to evaluate the POD modes - previously evaluated - in N grid points for $r\,N_{WT}$ times. Hence:

$$F_{\text{noMGS}} = N(rN_{WT})\,. \tag{5.13}$$

Then, if the MGS is performed the overall number of FLOPS is increased to:

$$F_{\text{MGS}} = N(rN_{WT}) + 2N(rN_{WT})^2\,. \tag{5.14}$$

The number of points in the grid for a farm with N_{WT} turbines can be estimated as:

$$N\,(N_{WT}) = N_{SWTS} + (N_{WT} - 1)\cdot L(S)\,, \tag{5.15}$$

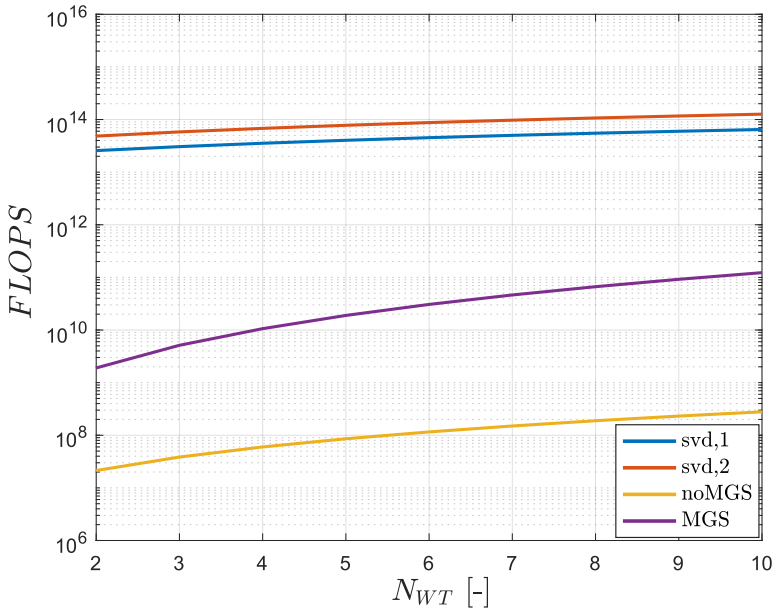


FIGURE 5.29: *Number of FLOPS required by the different techniques for the generation of a projector.*

assuming that the computational domain of the farm increases linearly with the number of turbines. The number of points of a SWTS has been computed using the available grid ($N_{SWTS} \approx 400\,000$) and the number of points that are added for each turbine L is a function of the spacing S. With a spacing of $5D$, $L \approx 100\,000$.

Figure 5.29 shows the FLOPS for each of the techniques. As it can be seen, the SVD procedure is actually quite demanding but its cost depend only slightly on the farm size. It must be underlined that in the calculation the number of snapshots has been kept constant ($n_s \approx 5000$), yet when dealing with large farms it is predictable an increase of the number of snapshots, to properly excite all the dynamic of the array and to account to the increasing time required to develop a stationary state. While the computation of P_{MGS} requires less FLOPS, a more steep growth is observed increasing the number of turbines, as expected from the Eq.(5.14). Lastly, the computation of the projector without recurring to orthogonalization techniques is by far the most economic algorithm.

CHAPTER 6

Conclusion and future work

This thesis aimed at finding a scheme to model a wind farm assembling the available data of a flow around a single machine and this purpose has been addressed in a multifaceted way. Moreover, although the present work was originated by previous studies focusing on a farm control strategy based on yaw steering, the findings may be extended even to different control techniques.

First of all, the superposition of flow fields was examined, benchmarking the effectiveness of several schemes in superposing time-averaged wakes. Within this framework, a new class of methods has been proposed, introducing a proper mixing coefficient. In fact, it has been found that the state-of-the-art schemes (which are current available in software dealing with engineering wake models) do not describe the interactions among different wakes in a suitable manner. What is even worse, is that the performance of these methods is significantly dependent on the power plant geometry. As an example, while the *maxdef* method is the most performing amongst the standard ones for aligned turbines, in the misaligned case the quadratic scheme appears to be more efficient. Hence, the suggested α-framework was designed to overcome these limits. In doing so, the mixing coefficient α was tuned keeping as a reference the physical behavior of interacting wakes. The resulting method actually achieved the best performances in all the studied layouts, still the gap with the standard methods is considerably smaller when dealing with misaligned turbines. Table 6.1 shows a comparison between the relative absolute errors evaluated in the wake domain at hub height for the 2D-blended and the quadratic methods. One

TABLE 6.1: *Comparison of the relative absolute errors between the 2D-blended (y^4) method and the best performing method amongst the standard ones. Data evaluated in the wake region of the hub plane.*

	Layouts			
Method	DA	TA	DM	TM
2D-bl.(y^4)	3.21%	4.63%	5.28%	5.05%
Best std. method	6.46%	6.18%	5.63%	5.26%

limitation of the superposition study stands in the definition of the blending and the mixing coefficient. Yet, the streamwise blending function was introduced to mimic at best the actual wake dynamic within a farm and the linear relation between α and the spacing (Eq.3.35) was dictated by the existing literature and validated in different real-scale farm layouts. Hence, all things considered, while the adopted mathematical framework may be sub-optimal, it was still found consistent and useful (as it is indicated by the error analysis). In addition, the analysis was limited only

to the near-wake region of the third turbine. While this represents a restriction on the validation of the proposed method, the error comparison is somewhat reassuring. Indeed, the α-framework achieved anyway better performances with respect to the classical superposition methods, even though the former was designed to yield accurate results in the far-wake region. Overall, the novel method appears well validated in the considered farm layouts and the framework could already be adopted to superpose CFD data in order to estimate the flow field in two- and three-turbines arrays, in a faster - but clearly less accurate - way than a complete CFD simulation. It must be also observed that in the entire study the domain of the SWTS was at least as long as those of the considered farm. While strictly speaking this is not a limitation - and the same condition is actually advocated for future studies - the thesis has not analysed the role of domain interruption, which would yield a discontinuity in the fields. Clearly, for long enough domain the influence of the turbine would be limited (e.g., the deficit would basically tend to 0) and the resulting discontinuity would only have a slight effect.

Then, this work investigated how to predict the turbine power outputs without running any simulation of the entire wind farm. The analysis underlines how the accurate reconstruction of the output vector is still far to be achieved. Indeed, both the superposition scheme and FLORIS yielded erroneous estimates of the produced power, particularly in the misaligned case. However, within the procedure of ROM extraction, the requirements on the accuracy of $\mathcal{Y}$ may be a bit relaxed. In fact, being a linearized analysis, an error constant and bounded in time would not influence particularly the model reliability. In addition, and even most importantly, the generation of a reduced model may be even carried on with a slightly offset $\mathcal{Y}$, incorporating a Kalman filter. The latter would indeed enhance the performance of the model, feeding to the system actual power measurements and eventually patching the inherent inaccuracy of the model (trained with a faulty output set).

Lastly, an alternative procedure to generate a projector matrix was considered. The assembly of the modes seems to yield accurate enough results, comparable with those that a standard SVD approach would give. In general, the studied scenario revealed that the orthogonalization procedure was not necessary, as the accuracy of the model remained the same but the computational cost increased. It is postulated that the MGS did not improve the model due to the already lower conditioning number of the P_{noMGS} projector. However, when dealing when larger farms (and therefore with higher modes) an increase of the conditioning number is expected, as the number of overlaps among modes would grow, and thus the projector may be re-orthogonalized to avoid numerical issues. Furthermore, the study of the modes highlighted that the dynamics of the waked turbines should be necessarily included to capture the flow behaviour in the downstream regions. Hence, one should consider at least the meandering of the wake but it may be even required to involve a time-varying yaw signal for the downwind turbines. It is clear that the latter scenario would entail a much more complicated picture of the flow and to properly describe the dynamic of the farm the number of snapshots should be increased. This in turn would further enlarge the computational cost of the SVD approach and, within this framework, the proposed algorithm for the generation of a projector would acquire even more weight. In conclusion, the assembly of the modes constitutes a compelling way to generate a ROM, especially when dealing with real-time applications.

6.1 Future work

Throughout the present work several aspects had emerged to further improve the topic .

Clearly, one way to continue the study would be the extraction of an actual ROM starting from the superposition of SWTS and its comparison with another reduced model generated from a farm simulation (following the standard approach given by Fortes-Plaza et al., 2018). This analysis could not be currently performed as:

1. The superposition framework was validated for a $5D$ spacing, sticking to the same spacing used for experimental data, whereas the available ROM was extracted for a two-turbines cluster with a $4D$ spacing

2. No simulations describing at least the meandering of the downwind wakes were available and - as mentioned before - the dynamic of the shaded turbines could have not been captured

3. The power prediction algorithm should be further improved.

Though, recall that the latter element could be bypassed with the employment of the actual $\mathcal{Y}$ and a Kalman filter. As a first step, it is advocated to compare the superposed and actual ROMs on a layout consisting of two aligned turbines spaced by $5D$, such that the superposition framework would be already validated. In this regard, both a yaw-varying simulation of the entire farm and a simulation capturing the meandering of a single turbine with a $I_0 = 13\%$ should be run.

Furthermore, in the future it would be necessary to test and validate the superposition framework in different layouts and this topic alone consists in a wide field of research. In fact, it is recommended to study the α-scheme with distinct streamwise and lateral spacing, comparing the superposed fields with those obtained in full simulations. This phase should be even coupled with the collection of experimental data to ensure the consistency of the results. Indeed, the thesis also pointed out the need to further improve the CFD model which was unable to predict the correct I distribution behind two aligned turbines. Overall, the study would suggest whether a linear relation between α and the spacing would hold. Moreover, notice how the whole framework was obtained under the assumption of a SBL, but - as mentioned in Sec.1.1 - an unstable condition of the ABL may induce to a different wake recovery and even this could be subject of future investigations.

After the modelling scheme is validated, one can generate a dataset of single turbine simulations at different I and tap into it to generate ROM for a chosen wind farm. In this regard, the Algorithm 2 is proposed. Notice how no further assumption is made on the $\mathcal{U}$ vector (e.g., the algorithm does not depend on the actuation strategy) and that the turbulence is estimated at a previous step than the actual superposition of the wakes. Finally, it must be observed that this approach would ensure a high degree of flexibility and would be key for the systematic generation of ROM, paving the way to a nonlinear control strategy.

Lastly, besides all the proposed future developments, even the field of control should be properly explored. Within this path of research, it is sought a more robust criteria for the ROM order selection. As the ROMs generated by this study and previous ones (Fortes-Plaza et al., 2018) show good agreement with the training set, though the chosen order was by far lower than that suggested by the standard energetic criteria, it is postulated that the cumulative energy may be not fully representative of the scenario. Furthermore, to actually close the topic of wind farm control, the design of a suitable controller is required. In particular, the controller

Algorithm 2: Generation of a superposed ROM for a generic farm.

Choice of the farm parameters: N_{WT}, spacing, $U_\infty = U_{0,1}$, I_0.
Design of the input signal to be applied, $\mathcal{U}$
for $l = 2 : N_{WT}$ **do**
    Estimation of the I at the l-th turbine: I_l
    Selection of the modelling SWTS for the l-th turbine, run at $I_0 \approx I_l$
end
for $i = 1 : n_s$ **do**
    Superposition of the flow fields in the entire domain following Alg.1
    Prediction of the power outputs of all the turbines
    Collection of the streamwise velocity in the column vector X_i
end
Assembly of all the column vectors in $\mathcal{X}$ and $\mathcal{X}'$
Construction of the output vector $\mathcal{Y}$
Generation of a proper projector P
Extraction of the ROM.

must maximize the power production of the plant for each instant, accordingly acting on the input.

APPENDIX A

RAEs of superposition methods

In the present Appendix, the relative absolute errors in every layouts are examined. The study is carried on looking at the *RAE*s of the different methods both at the entire domain and at the waked region of all the 25 horizontal planes.

A.1 Double Aligned Layout

TABLE A.1: *RAE for the double aligned layout. Data expressed in percentage.*

	Superposition methods							
Plane	*lin*	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$z = 0.20\,\mathrm{m}$	2.88	**1.48**	1.84	2.43	2.37	1.97	2.10	2.15
$z = 0.25\,\mathrm{m}$	3.59	**1.57**	2.16	2.45	2.35	1.94	2.06	2.11
$z = 0.30\,\mathrm{m}$	4.25	1.90	2.33	2.34	2.26	**1.89**	2.05	2.11
$z = 0.35\,\mathrm{m}$	4.59	2.03	2.43	2.20	2.05	**1.68**	1.85	1.91
$z = 0.40\,\mathrm{m}$	5.17	2.27	2.71	2.20	1.95	**1.57**	1.74	1.80
$z = 0.45\,\mathrm{m}$	5.55	2.46	2.89	2.16	1.86	**1.51**	1.67	1.74
$z = 0.50\,\mathrm{m}$	6.37	2.86	3.32	2.30	1.96	**1.63**	1.75	1.82
$z = 0.55\,\mathrm{m}$	6.22	2.88	3.28	2.20	1.87	**1.59**	1.67	1.72
$z = 0.60\,\mathrm{m}$	6.66	3.04	3.52	2.26	1.90	**1.69**	1.70	1.75
$z = 0.65\,\mathrm{m}$	6.59	3.10	3.55	2.20	1.86	1.69	**1.68**	1.72
$z = 0.70\,\mathrm{m}$	7.01	3.33	3.83	2.27	1.93	1.80	**1.74**	1.78
$z = 0.75\,\mathrm{m}$	6.80	3.27	3.76	2.16	1.84	1.78	**1.68**	1.71
$z = 0.80\,\mathrm{m}$	6.97	3.34	3.89	2.21	1.88	1.86	**1.74**	1.76
$z = 0.85\,\mathrm{m}$	6.75	3.28	3.80	2.17	1.85	1.86	**1.73**	1.74
$z = 0.90\,\mathrm{m}$	6.93	3.36	3.90	2.23	1.90	1.94	**1.80**	1.80
$z = 0.95\,\mathrm{m}$	6.39	3.12	3.58	2.11	1.77	1.79	**1.66**	1.66
$z = 1.00\,\mathrm{m}$	6.24	3.04	3.47	2.15	1.80	1.76	**1.66**	1.67
$z = 1.05\,\mathrm{m}$	5.76	2.86	3.22	2.07	1.74	1.65	**1.59**	1.61
$z = 1.10\,\mathrm{m}$	5.61	2.79	3.16	2.07	1.79	1.64	**1.62**	1.64
$z = 1.15\,\mathrm{m}$	5.12	2.63	2.96	2.00	1.79	1.61	**1.62**	1.64
$z = 1.20\,\mathrm{m}$	4.78	2.50	2.84	1.99	1.89	1.70	**1.71**	1.74
$z = 1.25\,\mathrm{m}$	4.52	2.46	2.79	2.04	2.00	**1.80**	1.83	1.86
$z = 1.30\,\mathrm{m}$	4.28	2.36	2.73	2.11	2.16	**1.92**	1.97	2.00
$z = 1.35\,\mathrm{m}$	3.91	2.11	2.66	**2.04**	2.16	2.12	2.11	2.12
$z = 1.40\,\mathrm{m}$	3.31	**1.79**	2.28	2.08	2.13	1.93	1.97	1.99

TABLE A.2: *RAE_{wake} for the double aligned layout. Data expressed in percentage.*

Plane	**Superposition methods**							
	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$z = 0.20\,\mathrm{m}$	7.60	**2.67**	4.44	6.25	6.06	4.84	5.24	5.39
$z = 0.25\,\mathrm{m}$	9.38	**2.77**	5.18	6.02	5.75	4.53	4.87	5.02
$z = 0.30\,\mathrm{m}$	10.35	**3.46**	5.19	5.23	5.02	4.00	4.44	4.61
$z = 0.35\,\mathrm{m}$	11.43	3.93	5.58	4.96	4.55	**3.55**	4.00	4.17
$z = 0.40\,\mathrm{m}$	12.53	4.53	6.16	4.83	4.18	**3.21**	3.65	3.81
$z = 0.45\,\mathrm{m}$	13.18	4.95	6.48	4.64	3.89	**3.00**	3.41	3.57
$z = 0.50\,\mathrm{m}$	13.99	5.46	6.93	4.58	3.79	**3.04**	3.31	3.46
$z = 0.55\,\mathrm{m}$	14.22	5.73	7.13	4.53	3.72	**3.07**	3.24	3.37
$z = 0.60\,\mathrm{m}$	14.47	5.96	7.38	4.53	3.72	**3.24**	3.27	3.37
$z = 0.65\,\mathrm{m}$	14.77	6.19	7.63	4.46	3.67	3.25	**3.22**	3.32
$z = 0.70\,\mathrm{m}$	14.93	6.34	7.84	4.36	3.60	3.32	**3.18**	3.26
$z = 0.75\,\mathrm{m}$	15.05	6.47	7.99	4.27	3.52	3.38	**3.15**	3.21
$z = 0.80\,\mathrm{m}$	14.81	6.46	7.97	4.25	3.53	3.47	**3.21**	3.25
$z = 0.85\,\mathrm{m}$	14.67	6.47	7.96	4.25	3.54	3.57	**3.27**	3.28
$z = 0.90\,\mathrm{m}$	14.45	6.40	7.87	4.23	3.52	3.61	3.31	**3.29**
$z = 0.95\,\mathrm{m}$	13.88	6.15	7.50	4.15	3.37	3.41	**3.14**	3.14
$z = 1.00\,\mathrm{m}$	13.26	5.88	7.10	4.18	3.39	3.31	**3.10**	3.12
$z = 1.05\,\mathrm{m}$	12.77	5.71	6.84	4.16	3.40	3.20	**3.05**	3.09
$z = 1.10\,\mathrm{m}$	12.01	5.39	6.50	4.06	3.44	3.10	**3.05**	3.10
$z = 1.15\,\mathrm{m}$	11.42	5.24	6.33	4.08	3.60	3.18	3.19	3.25
$z = 1.20\,\mathrm{m}$	10.60	4.97	6.07	4.08	3.85	**3.39**	3.43	3.48
$z = 1.25\,\mathrm{m}$	10.12	4.88	5.99	4.19	4.08	**3.61**	3.70	3.77
$z = 1.30\,\mathrm{m}$	9.42	4.59	5.78	4.31	4.42	**3.87**	3.98	4.05
$z = 1.35\,\mathrm{m}$	9.05	**4.18**	5.93	4.38	4.69	4.57	4.56	4.57
$z = 1.40\,\mathrm{m}$	7.43	**3.47**	4.92	4.46	4.56	4.09	4.18	4.23

A.2 Triple Aligned Layout

TABLE A.3: *RAE for the triple aligned layout. Data expressed in percentage.*

Plane	**Superposition methods**							
	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$z = 0.20\,\mathrm{m}$	3.26	**2.17**	2.27	3.15	2.95	2.65	2.75	2.78
$z = 0.25\,\mathrm{m}$	4.19	**2.39**	2.55	3.26	3.01	2.66	2.75	2.79
$z = 0.30\,\mathrm{m}$	5.10	2.65	2.64	3.25	2.88	**2.56**	2.70	2.75
$z = 0.35\,\mathrm{m}$	5.57	2.72	2.70	3.21	2.69	**2.37**	2.51	2.56
$z = 0.40\,\mathrm{m}$	6.22	2.84	2.86	3.33	2.66	**2.33**	2.48	2.54
$z = 0.45\,\mathrm{m}$	6.64	2.92	2.96	3.40	2.67	**2.33**	2.49	2.55
$z = 0.50\,\mathrm{m}$	7.58	3.22	3.30	3.67	2.84	**2.48**	2.64	2.70
$z = 0.55\,\mathrm{m}$	7.44	3.20	3.24	3.53	2.72	**2.39**	2.54	2.59
$z = 0.60\,\mathrm{m}$	7.95	3.24	3.43	3.66	2.80	**2.48**	2.61	2.66
$z = 0.65\,\mathrm{m}$	7.95	3.39	3.54	3.58	2.75	**2.47**	2.57	2.61
$z = 0.70\,\mathrm{m}$	8.52	3.59	3.85	3.67	2.82	**2.58**	2.66	2.69
$z = 0.75\,\mathrm{m}$	8.29	3.56	3.80	3.50	2.70	**2.52**	2.56	2.59
$z = 0.80\,\mathrm{m}$	8.54	3.58	3.93	3.48	2.69	**2.54**	2.57	2.58
$z = 0.85\,\mathrm{m}$	8.29	3.56	3.87	3.36	2.60	**2.48**	2.49	2.50
$z = 0.90\,\mathrm{m}$	8.52	3.66	3.95	3.41	2.62	2.51	**2.51**	2.52
$z = 0.95\,\mathrm{m}$	7.87	3.48	3.63	3.21	2.42	2.33	**2.32**	2.33
$z = 1.00\,\mathrm{m}$	7.76	3.41	3.50	3.23	2.42	2.33	**2.32**	2.33
$z = 1.05\,\mathrm{m}$	7.22	3.34	3.31	3.09	2.33	2.23	**2.21**	2.22
$z = 1.10\,\mathrm{m}$	7.13	3.35	3.34	3.12	2.39	**2.25**	2.25	2.26
$z = 1.15\,\mathrm{m}$	6.53	3.24	3.21	2.97	2.33	**2.17**	2.19	2.21
$z = 1.20\,\mathrm{m}$	6.10	3.14	3.13	2.88	2.34	**2.19**	2.20	2.21
$z = 1.25\,\mathrm{m}$	5.67	3.08	3.09	2.81	2.38	**2.22**	2.24	2.27
$z = 1.30\,\mathrm{m}$	5.29	2.97	3.06	2.82	2.54	**2.35**	2.38	2.41
$z = 1.35\,\mathrm{m}$	4.79	2.72	3.06	2.71	2.65	**2.58**	2.59	2.60
$z = 1.40\,\mathrm{m}$	3.90	**2.40**	2.61	2.76	2.63	2.44	2.48	2.51

TABLE A.4: *RAE_{wake} for the triple aligned layout. Data expressed in percentage.*

Plane	**Superposition methods**							
	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$z = 0.20\,\mathrm{m}$	8.32	**4.21**	5.31	7.98	7.40	6.47	6.77	6.88
$z = 0.25\,\mathrm{m}$	10.66	**4.44**	5.85	7.94	7.20	6.17	6.45	6.57
$z = 0.30\,\mathrm{m}$	12.07	**4.80**	5.59	7.18	6.22	5.37	5.73	5.87
$z = 0.35\,\mathrm{m}$	13.33	5.11	5.79	7.14	5.76	**4.93**	5.29	5.43
$z = 0.40\,\mathrm{m}$	14.51	5.44	6.10	7.25	5.59	**4.77**	5.14	5.28
$z = 0.45\,\mathrm{m}$	15.19	5.53	6.20	7.27	5.49	**4.66**	5.05	5.19
$z = 0.50\,\mathrm{m}$	16.09	5.71	6.46	7.29	5.42	**4.62**	4.98	5.11
$z = 0.55\,\mathrm{m}$	16.41	5.76	6.54	7.22	5.32	**4.54**	4.88	5.00
$z = 0.60\,\mathrm{m}$	16.62	5.85	6.72	7.24	5.34	**4.65**	4.92	5.03
$z = 0.65\,\mathrm{m}$	16.99	5.96	7.01	7.10	5.22	**4.59**	4.81	4.91
$z = 0.70\,\mathrm{m}$	17.36	6.08	7.32	6.93	5.11	**4.60**	4.76	4.83
$z = 0.75\,\mathrm{m}$	17.60	6.18	7.49	6.81	5.01	**4.60**	4.71	4.76
$z = 0.80\,\mathrm{m}$	17.52	6.18	7.56	6.60	4.89	**4.57**	4.63	4.66
$z = 0.85\,\mathrm{m}$	17.40	6.22	7.60	6.46	4.78	**4.51**	4.55	4.57
$z = 0.90\,\mathrm{m}$	17.18	6.23	7.50	6.33	4.67	4.44	**4.44**	4.46
$z = 0.95\,\mathrm{m}$	16.52	6.06	7.10	6.17	4.42	4.21	**4.19**	4.22
$z = 1.00\,\mathrm{m}$	15.99	5.96	6.74	6.17	4.41	4.21	**4.18**	4.21
$z = 1.05\,\mathrm{m}$	15.49	5.91	6.58	6.06	4.34	4.10	**4.07**	4.09
$z = 1.10\,\mathrm{m}$	14.78	5.73	6.43	5.95	4.34	**4.03**	4.03	4.06
$z = 1.15\,\mathrm{m}$	14.11	5.65	6.41	5.86	4.38	**4.02**	4.05	4.10
$z = 1.20\,\mathrm{m}$	13.17	5.51	6.27	5.68	4.43	**4.09**	4.11	4.14
$z = 1.25\,\mathrm{m}$	12.38	5.33	6.24	5.57	4.55	**4.16**	4.22	4.28
$z = 1.30\,\mathrm{m}$	11.42	5.03	6.13	5.57	4.90	**4.45**	4.53	4.59
$z = 1.35\,\mathrm{m}$	10.86	**4.63**	6.49	5.62	5.45	5.30	5.32	5.35
$z = 1.40\,\mathrm{m}$	8.42	**3.99**	5.31	5.66	5.36	4.89	5.01	5.06

A.3 Double Misaligned Layout

TABLE A.5: *RAE for the double misaligned layout. Data expressed in percentage.*

Plane	**Superposition methods**							
	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$z = 0.20\,\mathrm{m}$	1.97	1.83	**1.63**	1.80	1.80	1.78	1.79	1.80
$z = 0.25\,\mathrm{m}$	2.47	2.08	**1.89**	2.04	2.03	1.97	2.03	2.03
$z = 0.30\,\mathrm{m}$	3.01	2.43	**2.14**	2.16	2.17	2.16	2.17	2.17
$z = 0.35\,\mathrm{m}$	3.15	2.48	2.19	**2.12**	2.13	2.14	2.13	2.13
$z = 0.40\,\mathrm{m}$	3.41	2.53	2.26	2.13	2.13	2.16	2.13	**2.13**
$z = 0.45\,\mathrm{m}$	3.56	2.61	2.32	2.16	2.16	2.19	2.16	**2.16**
$z = 0.50\,\mathrm{m}$	4.04	2.90	2.59	2.38	2.37	2.42	2.38	**2.37**
$z = 0.55\,\mathrm{m}$	3.95	2.90	2.55	2.34	2.33	2.37	2.33	**2.33**
$z = 0.60\,\mathrm{m}$	4.20	2.93	2.61	2.41	2.38	2.42	2.38	**2.38**
$z = 0.65\,\mathrm{m}$	4.28	3.10	2.73	2.51	2.48	2.53	2.48	**2.48**
$z = 0.70\,\mathrm{m}$	4.62	3.34	2.96	2.69	2.67	2.74	2.67	**2.67**
$z = 0.75\,\mathrm{m}$	4.58	3.35	2.98	2.69	2.68	2.75	2.68	**2.68**
$z = 0.80\,\mathrm{m}$	4.80	3.45	3.12	2.82	2.82	2.90	2.82	**2.82**
$z = 0.85\,\mathrm{m}$	4.79	3.50	3.16	2.87	2.85	2.92	2.86	**2.85**
$z = 0.90\,\mathrm{m}$	5.02	3.65	3.32	3.03	3.00	3.07	3.00	**3.00**
$z = 0.95\,\mathrm{m}$	4.78	3.52	3.15	2.92	**2.86**	2.93	2.86	2.86
$z = 1.00\,\mathrm{m}$	4.77	3.50	3.13	2.95	**2.87**	2.95	2.88	2.88
$z = 1.05\,\mathrm{m}$	4.57	3.42	3.06	2.86	**2.80**	2.87	2.81	2.80
$z = 1.10\,\mathrm{m}$	4.60	3.47	3.12	2.90	**2.85**	2.92	2.86	2.85
$z = 1.15\,\mathrm{m}$	4.33	3.30	2.98	2.75	**2.72**	2.79	2.73	2.72
$z = 1.20\,\mathrm{m}$	4.01	3.03	2.79	2.57	**2.56**	2.61	2.56	2.56
$z = 1.25\,\mathrm{m}$	3.87	3.02	2.77	2.56	**2.56**	2.60	2.56	2.56
$z = 1.30\,\mathrm{m}$	3.72	2.93	2.73	2.56	**2.58**	2.60	2.58	2.58
$z = 1.35\,\mathrm{m}$	3.44	2.75	2.61	2.55	**2.56**	2.56	2.56	2.56
$z = 1.40\,\mathrm{m}$	2.96	2.31	2.25	2.24	2.24	**2.23**	2.24	2.24

TABLE A.6: *RAE_{wake} for the double misaligned layout. Data expressed in percentage.*

Plane	**Superposition methods**							
	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$z = 0.20\,\mathrm{m}$	4.60	**3.18**	3.61	4.11	4.10	4.05	4.09	4.09
$z = 0.25\,\mathrm{m}$	5.70	**3.65**	4.10	4.49	4.49	4.32	4.47	4.48
$z = 0.30\,\mathrm{m}$	6.51	**4.23**	4.30	4.34	4.38	4.35	4.37	4.37
$z = 0.35\,\mathrm{m}$	6.96	4.45	4.48	**4.30**	4.33	4.37	4.33	4.33
$z = 0.40\,\mathrm{m}$	7.46	4.61	4.61	**4.28**	4.29	4.37	4.29	4.29
$z = 0.45\,\mathrm{m}$	7.55	4.65	4.58	4.19	4.19	4.29	4.19	**4.19**
$z = 0.50\,\mathrm{m}$	7.89	4.84	4.71	4.26	4.25	4.36	4.26	**4.25**
$z = 0.55\,\mathrm{m}$	7.97	4.95	4.74	4.29	4.26	4.37	4.26	**4.26**
$z = 0.60\,\mathrm{m}$	8.19	5.02	4.80	4.38	4.32	4.42	4.32	**4.32**
$z = 0.65\,\mathrm{m}$	8.42	5.26	5.00	4.52	4.45	4.58	4.46	**4.45**
$z = 0.70\,\mathrm{m}$	8.70	5.49	5.24	4.68	4.63	4.78	4.64	**4.63**
$z = 0.75\,\mathrm{m}$	8.91	5.65	5.43	4.80	**4.78**	4.94	4.78	4.78
$z = 0.80\,\mathrm{m}$	9.14	5.83	5.63	5.01	**4.99**	5.16	5.00	4.99
$z = 0.85\,\mathrm{m}$	9.34	6.03	5.82	5.20	**5.16**	5.31	5.17	5.16
$z = 0.90\,\mathrm{m}$	9.45	6.15	5.93	5.35	**5.29**	5.43	5.29	5.29
$z = 0.95\,\mathrm{m}$	9.36	6.12	5.83	5.34	**5.21**	5.37	5.22	5.21
$z = 1.00\,\mathrm{m}$	9.23	6.09	5.76	5.38	**5.23**	5.39	5.24	5.24
$z = 1.05\,\mathrm{m}$	9.09	6.02	5.76	5.32	**5.19**	5.35	5.21	5.20
$z = 1.10\,\mathrm{m}$	8.77	5.87	5.66	5.19	**5.09**	5.25	5.10	5.09
$z = 1.15\,\mathrm{m}$	8.59	5.74	5.61	5.09	**5.03**	5.19	5.05	5.04
$z = 1.20\,\mathrm{m}$	8.00	5.31	5.28	4.80	**4.78**	4.90	4.79	4.78
$z = 1.25\,\mathrm{m}$	7.84	5.27	5.31	**4.83**	4.84	4.94	4.85	4.84
$z = 1.30\,\mathrm{m}$	7.49	5.08	5.24	**4.84**	4.89	4.94	4.90	4.89
$z = 1.35\,\mathrm{m}$	7.25	**4.86**	5.25	5.10	5.12	5.13	5.12	5.12
$z = 1.40\,\mathrm{m}$	6.04	**4.04**	4.41	4.38	4.38	4.35	4.39	4.38

A.4 Triple Misaligned Layout

TABLE A.7: *RAE for the triple misaligned layout. Data expressed in percentage.*

Plane	**Superposition methods**							
	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$z = 0.20\,\mathrm{m}$	2.25	2.22	**2.05**	2.20	2.18	2.21	2.17	2.17
$z = 0.25\,\mathrm{m}$	2.88	2.49	**2.25**	2.42	2.39	2.35	2.38	2.38
$z = 0.30\,\mathrm{m}$	3.51	2.80	**2.38**	2.48	2.44	2.44	2.44	2.44
$z = 0.35\,\mathrm{m}$	3.71	2.76	2.37	2.41	2.35	2.36	**2.34**	2.35
$z = 0.40\,\mathrm{m}$	4.04	2.74	2.42	2.44	2.37	2.38	**2.35**	2.36
$z = 0.45\,\mathrm{m}$	4.27	2.81	2.47	2.47	2.39	2.40	**2.37**	2.38
$z = 0.50\,\mathrm{m}$	4.83	3.07	2.71	2.68	2.59	2.60	**2.57**	2.58
$z = 0.55\,\mathrm{m}$	4.74	3.10	2.68	2.66	2.56	2.58	**2.54**	2.55
$z = 0.60\,\mathrm{m}$	4.96	3.14	2.76	2.79	2.66	2.69	**2.64**	2.65
$z = 0.65\,\mathrm{m}$	5.08	3.30	2.82	2.83	2.70	2.73	**2.68**	2.69
$z = 0.70\,\mathrm{m}$	5.39	3.53	3.00	3.02	2.87	2.91	**2.85**	2.86
$z = 0.75\,\mathrm{m}$	5.28	3.56	3.01	3.04	2.90	2.95	**2.88**	2.89
$z = 0.80\,\mathrm{m}$	5.39	3.66	3.10	3.19	3.04	3.09	**3.01**	3.03
$z = 0.85\,\mathrm{m}$	5.29	3.73	3.12	3.24	3.07	3.12	**3.05**	3.06
$z = 0.90\,\mathrm{m}$	5.45	3.88	3.25	3.43	3.22	3.25	**3.20**	3.21
$z = 0.95\,\mathrm{m}$	5.14	3.76	3.08	3.29	3.05	3.10	**3.03**	3.04
$z = 1.00\,\mathrm{m}$	5.06	3.77	3.07	3.37	3.11	3.15	**3.09**	3.10
$z = 1.05\,\mathrm{m}$	4.91	3.70	3.04	3.26	3.03	3.07	**3.02**	3.03
$z = 1.10\,\mathrm{m}$	4.96	3.75	3.14	3.27	3.06	3.11	**3.05**	3.06
$z = 1.15\,\mathrm{m}$	4.73	3.58	3.07	3.06	2.90	2.96	**2.89**	2.90
$z = 1.20\,\mathrm{m}$	4.51	3.38	2.99	2.89	2.79	2.86	**2.79**	2.79
$z = 1.25\,\mathrm{m}$	4.36	3.38	3.01	2.86	**2.81**	2.89	2.81	2.81
$z = 1.30\,\mathrm{m}$	4.27	3.34	3.07	**2.89**	2.91	2.96	2.91	2.91
$z = 1.35\,\mathrm{m}$	4.03	3.18	3.07	**2.97**	3.00	3.03	3.01	3.01
$z = 1.40\,\mathrm{m}$	3.43	2.74	2.69	**2.66**	2.68	2.68	2.68	2.68

TABLE A.8: *RAE_{wake} for the triple misaligned layout. Data expressed in percentage.*

Plane	**Superposition methods**							
	lin	*maxdef*	*quad*	*α-quad*	*1D-bl.*	*2D-bl.* (y^2)	*2D-bl.* (y^4)	*2D-bl.* (y^6)
$z = 0.20\,\mathrm{m}$	4.94	**3.80**	4.38	4.80	4.74	4.83	4.72	4.73
$z = 0.25\,\mathrm{m}$	6.35	**4.18**	4.63	5.09	5.00	4.90	4.98	4.99
$z = 0.30\,\mathrm{m}$	7.34	4.63	**4.56**	4.78	4.70	4.71	4.69	4.70
$z = 0.35\,\mathrm{m}$	7.88	4.62	4.57	4.66	4.52	4.57	**4.51**	4.52
$z = 0.40\,\mathrm{m}$	8.43	4.65	4.64	4.68	4.52	4.55	**4.49**	4.50
$z = 0.45\,\mathrm{m}$	8.69	4.68	4.61	4.61	4.42	4.46	**4.38**	4.41
$z = 0.50\,\mathrm{m}$	9.10	4.84	4.71	4.66	4.46	4.50	**4.42**	4.45
$z = 0.55\,\mathrm{m}$	9.23	4.98	4.75	4.71	4.50	4.55	**4.46**	4.48
$z = 0.60\,\mathrm{m}$	9.22	5.07	4.81	4.87	4.61	4.66	**4.56**	4.59
$z = 0.65\,\mathrm{m}$	9.54	5.20	4.84	4.87	4.60	4.67	**4.55**	4.58
$z = 0.70\,\mathrm{m}$	9.72	5.43	4.99	5.03	4.74	4.82	**4.69**	4.72
$z = 0.75\,\mathrm{m}$	9.84	5.60	5.14	5.19	4.91	5.01	**4.86**	4.89
$z = 0.80\,\mathrm{m}$	9.81	5.79	5.26	5.43	5.12	5.24	**5.07**	5.10
$z = 0.85\,\mathrm{m}$	9.82	6.00	5.36	5.62	5.27	5.36	**5.22**	5.25
$z = 0.90\,\mathrm{m}$	9.74	6.12	5.44	5.81	5.40	5.46	**5.34**	5.38
$z = 0.95\,\mathrm{m}$	9.54	6.09	5.32	5.76	5.27	5.37	**5.23**	5.26
$z = 1.00\,\mathrm{m}$	9.31	6.16	5.31	5.92	5.40	5.47	**5.36**	5.38
$z = 1.05\,\mathrm{m}$	9.27	6.11	5.38	5.83	5.35	5.44	**5.32**	5.34
$z = 1.10\,\mathrm{m}$	8.99	5.98	5.36	5.62	5.20	5.31	**5.17**	5.19
$z = 1.15\,\mathrm{m}$	8.85	5.81	5.39	5.37	5.04	5.16	**5.02**	5.03
$z = 1.20\,\mathrm{m}$	8.46	5.50	5.29	5.09	4.88	5.03	**4.87**	4.87
$z = 1.25\,\mathrm{m}$	8.31	5.50	5.41	5.08	**4.98**	5.14	4.98	4.98
$z = 1.30\,\mathrm{m}$	8.10	5.44	5.54	**5.16**	5.19	5.32	5.20	5.19
$z = 1.35\,\mathrm{m}$	8.04	5.27	5.85	**5.61**	5.69	5.75	5.71	5.70
$z = 1.40\,\mathrm{m}$	6.57	**4.50**	4.98	4.92	4.95	4.95	4.96	4.95

Bibliography

Abraham A, Dasari T, and Hong J (Oct. 2019). "Effect of turbine nacelle and tower on the near wake of a utility-scale wind turbine". In: *Journal of Wind Engineering and Industrial Aerodynamics* 193. DOI: 10.1016/j.jweia.2019.103981.

Ahmad T, Basit A, Anwar J, Coupiac O, Kazemtabrizi B, and Matthews P (Feb. 2019). "Fast Processing Intelligent Wind Farm Controller for Production Maximisation". In: *Energies* 12, p. 544. DOI: 10.3390/en12030544.

Allaerts D and Meyers J (June 2015). "Large eddy simulation of a large wind-turbine array in a conventionally neutral atmospheric boundary layer". In: *Physics of Fluids* 27, p. 065108. DOI: 10.1063/1.4922339.

Annoni J (May 2016). "Modeling for Wind Farm Control". PhD thesis. University of Minnesota.

Annoni J and Seiler P (2017). "A method to construct reduced-order parameter-varying models". In: *International Journal of Robust and Nonlinear Control* 27.4, pp. 582–597. DOI: 10.1002/rnc.3586.

Annoni J, Fleming P, Scholbrock A, Roadman J, Dana S, Adcock C, Porté-Agel F, Raach S, Haizmann F, and Schlipf D (Nov. 2018a). "Analysis of control-oriented wake modeling tools using lidar field results". In: *Wind Energy Science* 3, pp. 819–831. DOI: 10.5194/wes-3-819-2018.

Annoni J, Bay C, Taylor T, Pao L, Fleming P, and Johnson K (June 2018b). "Efficient Optimization of Large Wind Farms for Real-Time Control". In: pp. 6200–6205. DOI: 10.23919/ACC.2018.8430751.

Baiges J, Codina R, and Idelsohn S (Dec. 2013). "A domain decomposition strategy for reduced order models. Application to the incompressible Navier-Stokes equations". In: *Computer Methods in Applied Mechanics and Engineering* 267, pp. 23–42. DOI: 10.1016/j.cma.2013.08.001.

Bastankhah M and Porté-Agel F (2016). "Experimental and theoretical study of wind turbine wakes in yawed conditions". In: *Journal of Fluid Mechanics* 806, pp. 36. 506–541. DOI: 10.1017/jfm.2016.595.

Bay C, King J, Fleming P, Mudafort R, and Martínez Tossas L (May 2019). "Unlocking the Full Potential of Wake Steering: Implementation and Assessment of a Controls-Oriented Model". In: *Wind Energy Science Discussions*, pp. 1–20. DOI: 10.5194/wes-2019-19.

Betz A (1926). *Windenergie und ihre Ausnutzung durch Windmühlen*. Vandenhoek und Ruprecht. ISBN: 3922964117.

Boersma S, Doekemeijer B M, Gebraad P M O, Fleming P, Annoni J, Scholbrock A K, Frederik J A, and Wingerden J-W van (2017). "A tutorial on control-oriented

modeling and control of wind farms". In: *2017 American Control Conference (ACC)*, pp. 1–18. DOI: 10.23919/ACC.2017.7962923.

Bossuyt O H G (2018). "Modelling and validation of wind turbine wake superposition". MA thesis. Delft University of Technology, Technical University of Denmark.

Campagnolo F, Molder A, Schreiber J, and Bottasso C L (July 2019). "Comparison of Analytical Wake Models with Wind Tunnel Data". In: *Journal of Physics: Conference Series* 1256, p. 012006. DOI: 10.1088/1742-6596/1256/1/012006.

Ciri U, Rotea M, Santoni C, and Leonardi S (May 2017). "Large-eddy simulations with extremum-seeking control for individual wind turbine power optimization". In: *Wind Energy*. DOI: 10.1002/we.2112.

Corten G P and Schaak P (Oct. 2003). *Heat and Flux. Increase of Wind Farm Production by Reduction of the Axial Induction*.

Crespo A and Hernandez J (1996). "Turbulence characteristics in wind-turbine wakes". In: *Journal of Wind Engineering and Industrial Aerodynamics* 61 (1). DOI: 10.1016/0167-6105(95)00033-x.

Croce A, Cacciola S, Sartori L, Schwarzkopf M-A, Cosack N, Kanev S, and Elorza I (June 2019). "Assessment of controller key performance indicators Guidelines on controller application for the management of existing wind farms". In: CL-Windcon. URL: https://bit.ly/32WgCxj.

Davidson P A (2015). *Turbulence: an introduction for scientists and engineers*. 2 Edition. Oxford University Press. ISBN: 9780198722588.

Debnath M, Santoni C, Leonardi S, and Iungo G (Apr. 2017). "Towards reduced order modelling for predicting the dynamics of coherent vorticity structures within wind turbine wakes". In: *Philosophical Transactions of The Royal Society A Mathematical Physical and Engineering Sciences* 375. DOI: 10.1098/rsta.2016.0108.

DNV (Sept. 2011). *Offshore Standard DNV-OS-J101. Design of Offshore Wind Turbine Structures*. URL: https://rules.dnvgl.com/docs/pdf/DNV/codes/docs/2011-09/Os-J101.pdf.

España G, Aubrun S, Loyer S, and Devinant P (2011). "Spatial study of the wake meandering using modelled wind turbines in a wind tunnel". In: *Wind Energy* 14.7, pp. 923–937. DOI: 10.1002/we.515.

European Commission (Dec. 2018). "Directive (EU) 2018/2001 of the European Parliament and of the Council of 11 December 2018 on the promotion of the use of energy from renewable sources". In: *Official Journal of the European Union* L 328, pp. 82–209.

Fleming P, Gebraad P M O, Lee S, Wingerden J-W van, Johnson K, Churchfield M, Michalakes J, Spalart P, and Moriarty P (2014). "Evaluating techniques for redirecting turbine wakes using SOWFA". In: *Renewable Energy* 70, pp. 211 –218. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2014.02.015.

Fortes-Plaza A (2017). "Development of Reduced Order Models for Wind Farm Control". MA thesis. Technische Universität München.

Fortes-Plaza A, Campagnolo F, Wang J, Wang C, and Bottasso C L (June 2018). "A POD reduced-order model for wake steering control". In: *Journal of Physics: Conference Series* 1037, p. 032014. DOI: 10.1088/1742-6596/1037/3/032014.

Gunn K, Stock-Williams C, Burke M, Willden R, Vogel C, Hunter W, Stallard T, Robinson N, and Schmidt S (Sept. 2016). "Limitations to the validity of single wake superposition in wind farm yield assessment". In: *Journal of Physics: Conference Series* 749, p. 012003. DOI: 10.1088/1742-6596/749/1/012003.

Göçmen T, Laan M P van der, Réthoré P-E, Diaz A, Larsen G, and Ott S (July 2016). "Wind turbine wake models developed at the technical university of Denmark: A review". In: *Renewable and Sustainable Energy Reviews* 60, pp. 752–769. DOI: 10.1016/j.rser.2016.01.113.

Hoek D van der, Kanev S, Allin J, Bieniek D, and Mittelmeier N (2019). "Effects of axial induction control on wind farm energy production - A field test". In: *Renewable Energy* 140, pp. 994 –1003. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2019.03.117.

Howard K, Singh A, Sotiropoulos F, and Guala M (July 2015). "On the statistics of wind turbine wake meandering: An experimental investigation". In: *Physics of Fluids* 27. DOI: 10.1063/1.4923334.

Isermann R and Münchhof M (2014). *Identification of Dynamic Systems: An Introduction with Applications*. Springer Publishing Company, Incorporated. ISBN: 978-3-540-78879-9.

Krogstad P and Adaramola M S (2012). "Performance and near wake measurements of a model horizontal axis wind turbine". In: *Wind Energy* 15.5, pp. 743–756. DOI: 10.1002/we.502.

Larsen G C, Madsen H A, Thomsen K, and Larsen T J (2008). "Wake meandering: a pragmatic approach". In: *Wind Energy* 11.4, pp. 377–395. DOI: 10.1002/we.267.

Mao X and Sørensen J (Mar. 2018). "Far-wake meandering induced by atmospheric eddies in flow past a wind turbine". In: *Journal of Fluid Mechanics* 846. DOI: 10.1017/jfm.2018.275.

Menter F, Kuntz M, and Langtry R B (Jan. 2003). "Ten years of industrial experience with the SST turbulence model". In: *Heat and Mass Transfer* 4.

Menter F R and Egorov Y (2006). "SAS Turbulence Modelling of Technical Flows". In: *Direct and Large-Eddy Simulation VI*. Ed. by Eric Lamballais, Rainer Friedrich, Bernard J. Geurts, and Olivier Métais. Springer Netherlands, pp. 687–694. DOI: 10.1007/978-1-4020-5152-2_79.

— (July 2010). "The Scale-Adaptive Simulation Method for Unsteady Turbulent Flow Predictions. Part 1: Theory and Model Description". In: *Flow, Turbulence and Combustion* 85.1, pp. 113–138. DOI: 10.1007/s10494-010-9264-5.

Niayifar A and Porté-Agel F (June 2015). "A new analytical model for wind farm power prediction". In: *Journal of Physics: Conference Series* 625. DOI: 10.1088/1742-6596/625/1/012039.

NREL (2019a). *FLORIS. Version 1.0.0*. URL: https://github.com/NREL/floris.

— (2019b). *TurbSim. Version 1.06.00*. Last modified: 14-June-2016. URL: https://nwtc.nrel.gov/TurbSim.

Ohlenforst K, Sawyer S, Dutton A, Backwell B, Fiestas R, Lee J, Qiao L, Zhao F, and Balachandran N (2019). *Global Wind Report 2018*. Tech. rep. Global Wind Energy Council.

Okulov V, Naumov I, Mikkelsen R, Kabardin I, and Sørensen J (Apr. 2014). "A regular Strouhal number for large-scale instability in the far wake of a rotor". In: *Journal of Fluid Mechanics* 747, pp. 369–380. DOI: 10.1017/jfm.2014.174.

Pelegrì J and Gastel P (Dec. 2004). "Estimates of gradient Richardson numbers from vertically smoothed data in the Gulf Stream region". In: *Scientia Marina* 68.

Pope S B (2000). *Turbulent Flows*. 1st ed. Cambridge University Press. ISBN: 978-0-521-59125-6,978-0-521-59886-6.

Porté-Agel F, Bastankhah M, and Shamsoddin S (Sept. 2019). "Wind-Turbine and Wind-Farm Flows: A Review". In: *Boundary-Layer Meteorology*. DOI: 10.1007/s10546-019-00473-0.

Rowley C (Mar. 2006). "Model Reduction for Fluids, using Balanced Proper Orthogonal Decomposition". In: pp. 301–317. DOI: 10.1142/9789812774569_0016.

Schmid P and Sesterhenn J (Nov. 2008). "Dynamic Mode Decomposition of numerical and experimental data". In: *Journal of Fluid Mechanics* 656. DOI: 10.1017/S0022112010001217.

Shao Z, Ying W, Li L, Han S, and Yongqian L (Feb. 2019). "Multiple Wind Turbine Wakes Modeling Considering the Faster Wake Recovery in Overlapped Wakes". In: *Energies* 12, p. 680. DOI: 10.3390/en12040680.

Sherry M, Nemes A, Jacono D, Blackburn H, and Sheridan J (Nov. 2013). "The interaction of helical tip and root vortices in a wind turbine wake". In: *Physics of Fluids* 25. DOI: 10.1063/1.4824734.

Trefethen L and Bau III D (1997). *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics. ISBN: 0898713617,9780898713619.

Wang J, Wang C, Campagnolo F, and Bottasso C L (June 2018a). "A Large-Eddy Simulation Approach for Wind Turbine Wakes and its Verification with Wind Tunnel Measurements". In: *Wind Energy Science Discussions*, pp. 1–31. DOI: 10.5194/wes-2018-45.

— (July 2018b). "Scale-adaptive simulation of wind turbines, and its verification with respect to wind tunnel measurements". In: *Wind Energy Science Discussions*. DOI: 10.5194/wes-2018-47.

— (Jan. 2019). "Wake behavior and control: comparison of LES simulations and wind tunnel measurements". In: *Wind Energy Science* 4, pp. 71–88. DOI: 10.5194/wes-4-71-2019.

Xiao D, Heaney C, Mottet L, Hu R, Bistrian D A, Aristodemou E, Navon I, and Pain C (Feb. 2019). "A Domain Decomposition Non-Intrusive Reduced Order Model for Turbulent Flows". In: *Computers and Fluids* 182. DOI: 10.1016/j.compfluid.2019.02.012.