POLITECNICO DI MILANO
Master of Science Degree in Aeronautical Engineering
School of Industrial and Information Engineering

ENGINEERING MODELS ENHANCEMENT FOR WIND FARM WAKE SIMULATION AND OPTIMIZATION

Supervisor: Prof. Alessandro Croce
Tutor: Prof. Stefano Cacciola

Graduation Thesis:
Santiago Andrés Ramírez Castillo, Matr. 876875

Academic Year 2018-2019

Abstract

The wake of a single wind turbine is by itself a complex flow field, but for a wind farm the interactions of the wakes of all the rotors must be accounted as well. Control methods should investigate the different alternatives for improving the power production of the whole farm, mainly, by manipulating the properties of the interaction medium, the wakes. FLOw Redirection and Induction in Steady State - FLORIS is a useful and reliable software that uses engineering models for comprehending the behavior of a multiple wind turbine wakes flow field. Although it is already trustful for the calculation of these flows, it can still be enhanced to improve and add new capabilities, paying attention to not increasing indiscriminately its complexity. In the present work adjustments to the logic of the program and some of its components are proposed for improving the capacities of the software. The basic knowledge about the wind farm control methods and the general characteristics that should be covered by the wake submodels are explained. Additionally, FLORIS is reviewed in detail, including a comparison between both versions of the software: the one written in Python and FLORISSE M written in MATLAB®. For this a 2x1 wind farm with DTU 10MW turbines was used. The integrated optimization algorithm is explained and then augmented for optimizing additional variables of interest like the tilt angle and the hub height. The result of simple and combined optimizations for a 2x1-DTU 10MW are also presented. The principal existing issues found on the online available version of the software are the complexity of the input LUT for the yaw control method and the wrong consideration of the tilt rotations. This thesis proposes two different routines for determining the values of pP and pT, for enabling the use of simpler inputs. Additionally, adjustments to the Jimenez and Zones models are implemented for a correct tilt consideration, achieving the same wake behavior vertically and horizontally. The most significant contribution is to the power distribution, by the proper consideration of the sign of the thrust angle. The qualitative corrections are compared with experimental data found for a 3x1 power plant with TUM G1 turbines for evaluating their effectiveness. The new version of FLORIS will allow a more reliable consideration of the effects of the wakes over the downwind turbines, and with this, more accurate online control systems that include the program may be developed. Initial wind farm layout design and considerations are also benefited from additional variables optimization.

Sommario

La scia di una singola turbina eolica è un campo di moto complesso già considerata singolarmente, ma per un parco eolico anche le interazioni delle scie di tutti i rotori devono essere considerate. Metodi di controllo dovrebbero analizzare le diverse alternative per aumentare la produzione di potenza dell'intero parco, essenzialmente manipolando le proprietà del mezzo di interazione, cioè le scie. FLOw Redirection and Induction in Steady State – FLORIS è un software pratico e attendibile che sfrutta modelli ingegneristici per caratterizzare il comportamento di un campo di moto dominato dalla scia di turbine multiple. Nonostante sia già affidabile per il calcolo di questi flussi, può ulteriormente essere esteso migliorando funzionalità esistenti e aggiungendone di nuove, facendo attenzione a non incrementare indefinitamente la sua complessità. In questo lavoro di tesi sono presentate alcune modifiche apportate alla logica del programma e ad alcune sue componenti con l'obiettivo di incrementarne le capacità. Sono presentate le nozioni di base sui metodi di controllo solitamente applicati ai parchi eolici e le caratteristiche generali dei modelli che descrivono la scia di una turbina eolica. Inoltre, FLORIS è descritto in dettaglio, includendo un paragone tra le due versioni del software: quella scritta in Python e FLORISSE M scritta in MATLAB. Per questa un parco eolico 2x1 con turbine DTU 10MW è stata usata. L'algoritmo di ottimizzazione integrata è spiegato ed esteso per ottimizzare ulteriori variabili di interesse come l'angolo di tilt e l'altezza del mozzo. Sono presentati i risultati ottenuti per una ottimizzazione singola e combinata della stessa configurazione 2x1 di turbine DTU 10MW. Le principali limitazioni riscontrate nella versione del software disponibile, sono legate alla complessa gestione degli input LUT per il metodo di controllo dell'angolo di imbardata e all'erronea introduzione delle modifiche nella inclinazione. Per superare questi limiti sono state studiate due differenti routine in grado di determinare i valori di pP e pT, cosi da semplificare la procedura di input delle variabili di interesse. In aggiunta, alcune migliorie sono apportate ai modelli di Jimenez e Zones, ora implementati con una corretta considerazione dell'angolo di inclinazione. Questo permette di ottenere un comportamento invariabile della scia nella direzione orizzontale e verticale. Un'altra significativa modifica riguarda il calcolo della distribuzione di potenza, influenzata dalla corretta implementazione dell'effetto del segno dell'angolo di spinta. Gli effetti delle correzioni apportate sono evidenziati attraverso il confronto tra i risultati forniti dal software e quelli ricavati sperimentalmente per un parco eolico 3x1 con turbine TUM G1. La nuova versione di FLORIS permette una più affidabile modellazione degli effetti della scia della prima turbina sulle successive, rendendo così possibile lo sviluppo di un sistema di controllo online più accurato che includa FLORIS stesso. Anche la progettazione e lo sviluppo di un parco eolico possono beneficiare di un processo di ottimizzazione reso più attendibile dalla migliore gestione delle variabili.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The usage of green energies has been increasing during the last decade. Wind energy is the one that harness power from the wind movement, when it induced rotation to a wind turbine. An electrical generator then transforms this mechanical movement in to electrical power. Regardless that there is still the research for new more effective and, if it is possible, cheaper sources, the ones that already exist can always be updated, principally to increase their energy production, reduce their cost or both. For wind turbines these objectives may be achieved with two different focus targets: the single turbine or the whole wind farm. There are two main approaches: one can intervene during the machines' design and construction steps (cheaper and stronger materials, more efficient manufacturing processes, among others) for reducing the associated costs or use turbine control methods for *Cost Of Energy (COE)* reduction. Various wind turbine control strategies are treated in [1]. Wind farm control methods may be found in [2].

Wind turbine power production optimization focuses in the improvement of an isolated turbine. However, for optimizing the production of a whole wind farm, the interaction between the rotors must be taken into account. The wake of the generators is the principal medium of this interaction. The complete velocity field has to be analysed, including the multiple interactions between all the present wakes and the additional complexity that comes with the manipulation of the wakes when a control method is used. This is why the models use for simulating such a complex flow field should be accurate enough to satisfactorily represent all these phenomena. In [3, 4, 5] can be found examples of CFD studies using LES. Additionally, engineering models are also being investigated and developed, for example in [6, 7, 8], for low computational cost applications. These last specializes in modelling one specific aspect of the wake, for example, the Jimenez model is in charge of recreating the wake deflection and the Jensen model is a classic example for modeling the wake velocity deficit.

FLOw Redirection and Induction in Steady State - FLORIS is a software developed originally by TU Delft for wind farm power production analysis. Written in MATLAB® or in Python, it focuses in the simulation of the wakes generated by the interaction of the ambient flow with the turbines. The architecture of the program permits to simulate the most important characteristics of the wake with a selection of submodels, and it focuses in how a yaw induced wake deflection affects the complete power plant. Despite the implementation is not yet complete, it has the potential of considering additional control variables that are not the focus of the current analysis and optimization studies.

1.0.1 Thesis Review

During this work, an initial insight of how FLORIS works is provided and it will give the necessary theoretical background for its submodels comprehension. Its principal objective is to enhance and add some capabilities inside the software for augmenting the fidelity of its results, expecting to make it suitable for non-conventional control variables optimization and to show, by a comparison with experimental data, the functionality of the upgrades. This would expand the applications of the program because the overall complexity remains acceptable. The proposed upgrades will allow an improvement to the wind farm production by the development of online control systems for automated live control and presents a relatively simple tool for the initial analysis of the power plant's possible layout.

The **Introduction** gives a general overview of the problematic to be treated in this thesis and its general content. Next, the **Chapter 2** introduces and explains qualitatively the main control strategies and the modelling approaches for a wind farm. **Chapter 3** provides a detailed description of both versions of FLORIS, including the mathematical theory of the available submodels. A comparison between the two versions of the software, including some results, is also presented and it is explained how the selection of the submodels to use was done. In the **Chapter 4** it can be found all the contributions and upgrades proposed for FLORIS for the consideration of turbines with different hub heights and tilt deflections. In the final part of this same chapter, a quantitative and qualitative comparison between numerical and experimental results has been made. **Chapter 5** addresses the optimization aspects of the software and show the results of simple and combined non-conventional control variables optimizations. Finally, the **Conclusions** of this thesis are presented.

Chapter 2

Wind Farm Control and Modelling

The considerations necessary for wind farm control methods have a different focus than the strategies for controlling a single generator. The turbine is no longer isolated, so the interactions between the machines must be considered, and the main source of these interactions is the wake that arrives to the downwind machine. For the modelling, something similar happens. Rather than focusing on how the flow affects the rotor, it should be considered how the rotor affects the flow, how it is perturbed. In this section, the principal control methods used for wind farm production optimization will be discussed and the possible approaches for its modelling are exposed.

2.1 Control

Wind turbine control methods are used mainly to increase the turbine's energy production and for reducing the stresses suffered by its parts, maintaining the structural health. However, wind farm control methodologies priorities the overall power plant production over the single machine efficiency. This section will address three control strategies: *Yaw Redirection*, *Induction Control* and *Active Wake Mixing.*

2.1.1 Yaw Redirection

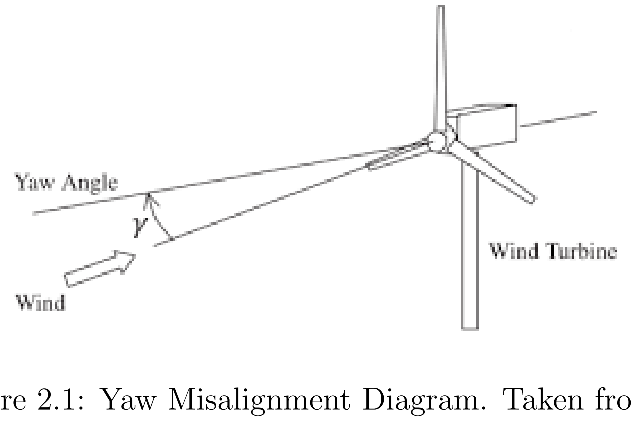
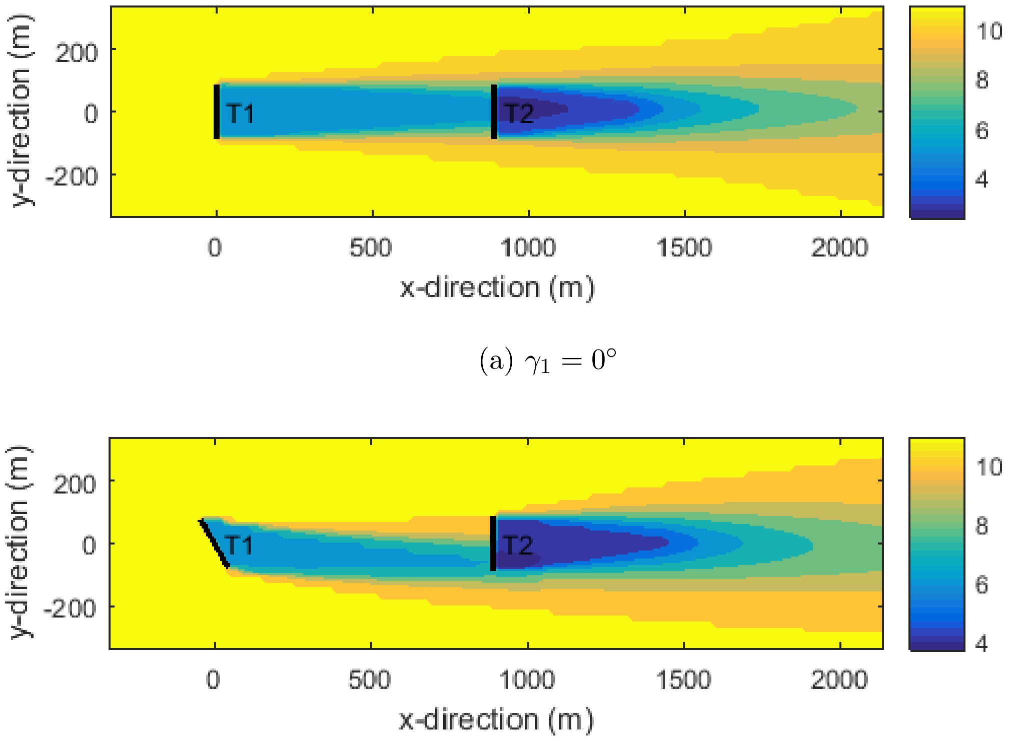


Figure 2.1: Yaw Misalignment Diagram. Taken from [9].

For purposes of this work, lets define as *Yaw Angle* θ the one between the origin turbine horizontal orientation and the actual horizontal orientation, and as *Yaw*

Misalignment γ the angle between the turbine actual horizontal orientation and the incident wind direction. A case in which both angles are equivalent is shown in the Figure 2.1. To illustrate further the difference between these quantities let's consider an example: A change of the wind direction causes an initial yaw misalignment γ_0, which demands an equivalent yaw deflection. Once the turbine orientation is again normal to the flow, γ will be equal to 0 and the yaw angle will be $\theta = \gamma_0$.

In principle, yaw control is used to reduce the yaw misalignment, take it to zero if possible, as the maximum energy production of a wind turbine is reached when it is perpendicular to the incoming velocity field. This is the main application of this control strategy talking about a single turbine performance. However, studies such as [4, 10] have shown that it might desirable to use Yaw Redirection in the upwind turbine to induce a performance increase on a downwind turbine.



(a) $\gamma_1 = 0°$

(b) $\gamma_1 = 30°$

Figure 2.2: Yaw Induced Wake Deflection

Figure 2.2 shows an example of a yaw induced wake deflection. In 2.2a the second turbine is working under the worst possible regime as it has to produce energy under the complete wake of the previous generator. Yaw Redirection has been use in the case 2.2b. The objective is to take the second turbine to a working regime under partial wake. This way the velocity profile seen by this machine is more similar to the ambient flow, causing a significant increase on its production and on the farm's overall energy generation as well. Its important to have caution, the *first turbine yaw misalignment* γ_1 can't be indiscriminately increased, as the energy production of the power plant can be deteriorated because the significant reduction of the first machine's production.

It is of interest for modern research [11] to develop an online yaw misalignment control method which can quickly and accurately calculate the turbines yaw angles array for optimizing the wind farm power production after a change in the wind direction.

2.1.2 Induction Control

Similarly to Yaw Redirection, the performance of the upwind rotor is compromised to generate better working conditions for the downwind turbines. Corten and Shaak in [12] show the effects of this control method. Reducing the axial induction factor of the upwind turbine a_1 will reduce the velocity deficit of its wake and the downwind turbine will experience a faster incoming velocity profile. The first generator production will decrease but the increase of the second one will compensate the power production shortage.

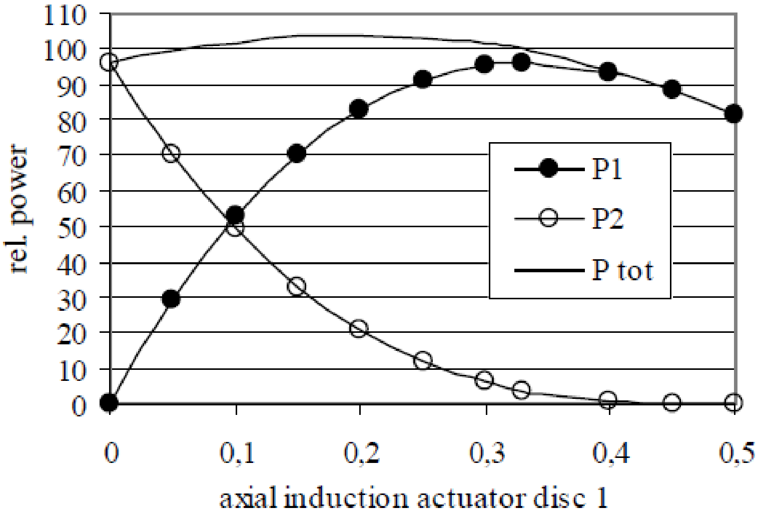


Figure 2.3: a_1 Reduction Effects on the Power Distributions. Taken from [12].

Figure 2.3 show the power distributions in function of a_1 of the individual turbines and of the whole power plant for a 2x1 wind farm. The highest energy production of the first turbine is when $a_1 = 1/3$, the optimal value. Nevertheless, the peak of the power plant generation is when $a_1 = 1/5$.

One way to reduce a is by the variation of the blade's pitch angle. The pitch angle β is the angle between the airfoil chord and the blade section speed. The angle of attack (AoA) α is the angle between the airfoil chord and the relative wind speed. At high pitch angles (low AoA), the flow is less disturbed, less energy is being harness by the turbine, its wake is in the closest state to the original velocity field, and therefore, a is decreased. This has a negative impact on the turbine but benefits the machine behind it, as this last one will profit from a cleaner and faster velocity field.

2.1.3 Active Wake Mixing

Differently than the previous control methods (which are static), Active Wake Mixing, also called wake meandering, is a dynamic method which consist in periodically deflecting the wind turbine wake. Goit and Meyers in [3] use individual turbines as dynamic actuators to improve the wind farm production within a Large Eddy Simulation (LES). As indicated in [13], given that this is a LES-based method, the computational cost is yet to high for practical implementation. Nonetheless the objective of this method is to reduce the velocity deficit of the wake as the Induction Control, the wake meandering does not causes a significant reduction of the upwind turbine production as the working regime is never far from the optimal.

2.2 Modelling

Considering a single wind turbine, there are different areas in need of modelling, mainly: aerodynamics, structures and control. Aero-servo-elastic models have to be implemented in order to take into account all the different phenomena that will affect the machine. These models should be capable to estimate with sufficient certainty the turbine's behaviour within different operating conditions. Simulating these last is the role of wind farm aerodynamics modelling.

Because it is not under the influence of the perturbation of another wind farm element, the first wind turbine may be modelled as an isolated machine, but its wake is going to change the initial conditions for the ones behind. It should be the role of the aerodynamics model to determine the operating conditions of each turbine inside the power plant. These are basically represented by the velocity profile given to the *ambient flow* V_∞ and the *hub height* z_{hub} of the machine. For the elements under the influence of a wake, the velocity field gains complexity. This modelling may be achieve by means of high fidelity methods, *Computational Fluid Dynamics-CFD*, or with low fidelity methods, *Engineering Models*.

2.2.1 Computational Fluid Dynamics-CFD

CFD is a high accuracy method that simulates the flow field interaction with a body inside a meshed control volume. The value of the fluid properties is calculated by means of a numerical method, being Finite Volumes the most commonly used. It is a conservative method, that calculates the fluxes at the element's interfaces. In cases where turbulent flow must be considered, depending on the demands of the specific problem, there are three different modelling methods for taking into account such a complex phenomenon.

Reynolds Averaged Navier-Stokes-RANS expresses the conservation laws using the Reynolds decomposition, which separates the variables into their time averaged and fluctuating parts. The whole flow field is modelled, making this the cheaper but most inaccurate method.

Large Eddy Simulation-LES solves numerically the Navier-Stokes equations for the big scales phenomena (large eddies) and models them for the small scale motions. Its computational cost increases in comparison with the RANS, but the precision augments as well.

Direct Numerical Simulation-DNS solves numerically the governing equations for the complete flow field. Evidently is the most accurate approach of the three, however, the high computational cost causes that this particular method is use almost exclusively for simple flow field calculations.

2.2.2 Engineering Models

The complexity of these methods decreases significantly in comparison with a CFD and the computational cost is practically minimum. This comes with a significant information loss because various phenomena are not considered. Nevertheless, they present satisfactory results and can be used in situations that required constant updating computations, as just the specific characteristics of the particular flow field are modeled. In the interest of wind farm power production analysis and optimization, there are four main aspects to take into account in order to simulate a wind turbine wake: *wake deflection*, *wake velocity deficit*, *wake combination* and *added turbulence*. As an additional observation, it is important that all these models guarantee that the upwind turbine will not be affected by the downwind turbine operation.

Wake Deflection model should describe the wake displacement cause by any misalignment between the wind direction and the turbine orientation. Figure 2.2 shows the case of a horizontal wake displacement due a yaw misalignment, but there can be vertical wake displacement caused by a *tilt angle* φ as well. Depending of the aerodynamic model used for the turbine, the horizontal and vertical deflection may have the same behaviour.

Wake Velocity Deficit model is in charge of estimating the deceleration of the flow caused by its interaction with the turbine. For the recreation of the downwind velocity profile, aspects like wake expansion, velocity reduction, and velocity recovery must be taken into account. The various models represent in different ways these aspects and therefore generate a different velocity field. In the optics of engineering models, there is always a possibility to make a model more accurate, but the equilibrium between accuracy and calculation complexity must not be left aside.

Wake Combination model needs to represent the new velocity profile resulting from the interaction of two or more previous fields. One must not get confused by the name, as it does not only describe the interaction of various wakes but also the one between the ambient flow and a wake. This may be achieved in various ways, the ones that will be mention further in this work will be via velocities summation and kinetic energy summation.

Added Turbulence model is use to consider the increment of the wake turbulence generated by the flow-body interaction.

This thesis will concentrate in the Yaw Redirection control as it is the most promising mechanism for optimizing the power production of the whole farm. For the modelling, it is of interest to enhance Engineering Models that would allow the development of online control systems given their reduced computational cost.

Chapter 3

FLORIS - Overview

The results and commentaries exposed in this chapter have been obtained and make reference to the software versions available at the date: 04/02/2019 for FLORIS Python version and 07/02/2019 for FLORISSE M (MATLAB® version). The velocity fields visualization shown have been generated with FLORISSE M.

3.1 General Description

FLOw Redirection and Induction in Steady State - FLORIS is a free source reduced order engineering model used for the simulation of wind turbine wakes, their interactions and wind farm energy production optimization. There are two available versions: the original version developed by the *Delft University of Technology-TU Delft* written in MATLAB® [14] and a version written in Python developed by the same academic institution in cooperation with the *National Renewable Energy Laboratory-NREL* of the U.S. government [15]. The translation to a new coding language has been done thinking about spreading the model usage without the necessity of a particular commercial software. Nevertheless, the wide use of MATLAB® within the academic community encourages that this version is also taken up to date.

The software main capabilities are finding the working conditions of the turbines inside a wind farm (although a simulation of a single wind turbine is also possible), generating the resulting velocity field and visualizations and optimize the plant's power production by computation of optimal yaw misalignment for each turbine. The *Object Oriented Programming (OOP)* implementation leaves room for easily adding features and adjusting the existing ones for the user's particular interests. Even though it is not a *high fidelity method*, it has been proven that it is capable to find satisfactory results in confront with experimental data [16].

Nonetheless the software version written in Python is a sort of translation of the original one, the constant work for improving the instrument and that the versions are managed by different institutions has generated differences in the programs. The essence is the same, the main goal is still shared but the logic behind and some of the models differ, and therefore, the results obtained as well, even with the same input data. Is highly recommended to take the time to consider which software version adjusts better to the user needs.

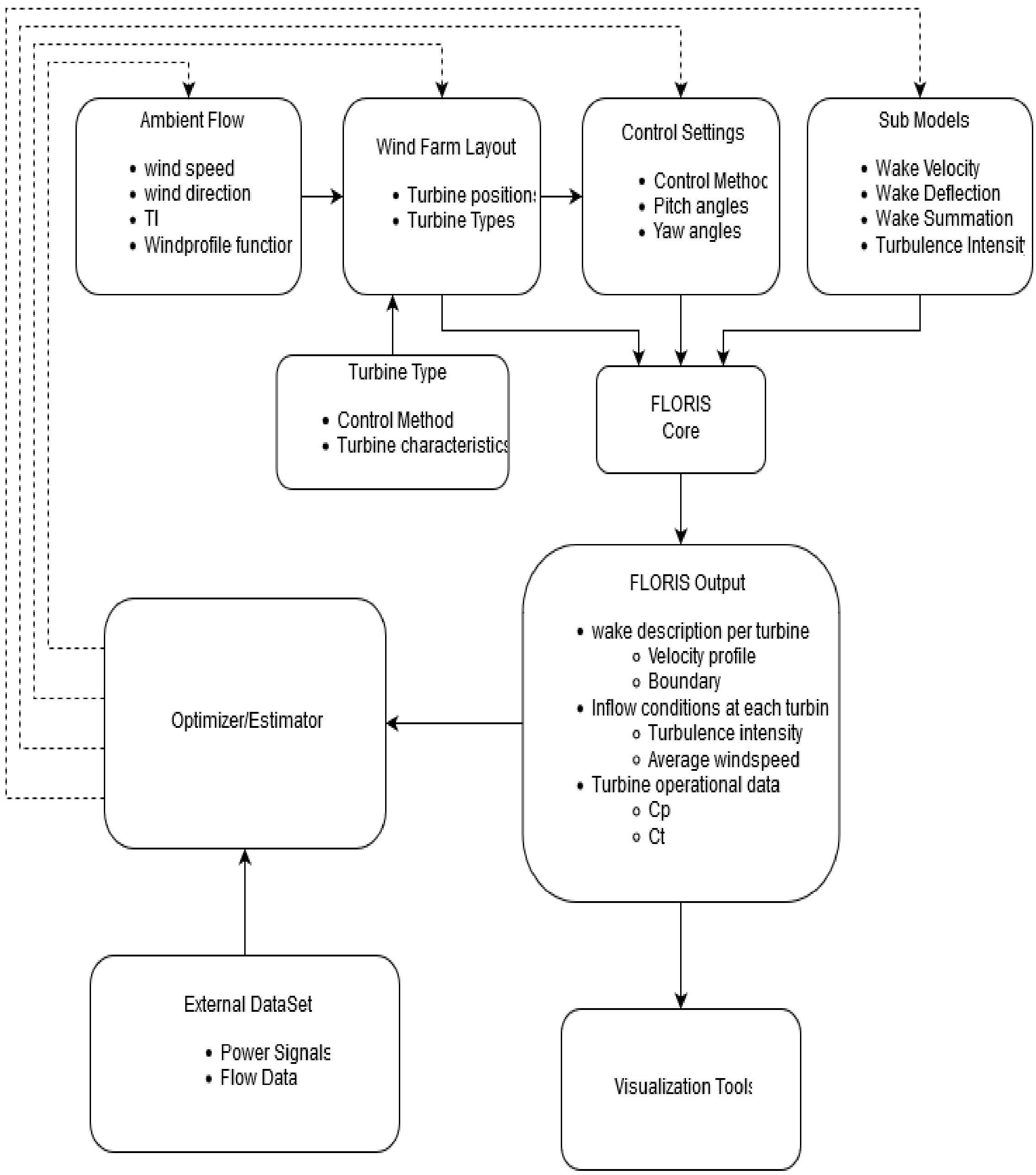


Figure 3.1: FLORISSE M version program architecture. Taken from [17].

3.2 Start Up

The inputs necessary for doing a simulation can be categorized: one must choose the different sub models to use, these models' parameters, the inflow conditions, the wind farm layout, the turbine to use and the control variables. Let's refer as *case inputs* to any input that will be modified for the analysis of different scenarios; the ambient conditions, the turbines and the wind farm characteristics. These last includes the farm's layout and control variables. It will be refer as *specific inputs* to the ones whose objective is the tuning of a specific model. Fig. 3.1 illustrates the FLORISSE M architecture. Even if for the Python version is different, both follow a similar logic. Flow properties, such as density, wind direction, wind velocity, turbulence intensity and shear factor are requested. It is important to clarify that the ground is consider by the velocity profile used, there is no other way to simulate it. Thus, if no shear factor is provided, FLORIS does not take into account the ground, as the input velocity profile will be uniform. For describing a specific turbine it is necessary to provide the principal physical characteristics of the machine: *turbine radius R* and *hub height z_{hub}*, with a data array that contains the *power coefficient C_p* and the *thrust coefficient C_t* values for various wind speeds. The C_p and C_t are non-dimensional forms of the power P and thrust force T given by the equations:

$$C_p = \frac{P}{\frac{1}{2}\rho U^3 A} \tag{3.1}$$

$$C_t = \frac{T}{\frac{1}{2}\rho U^2 A} \tag{3.2}$$

where ρ is the air density, U is the wind speed and A is the rotor area. As it is needed for one of the software's main features, FLORIS uses corrective functions for calculating these coefficients under yaw conditions:

$$C_p = C_{p_0} \cos^{pP} \gamma \tag{3.3}$$

$$C_t = C_{t_0} \cos^{pT} \gamma \tag{3.4}$$

where the 0 subscript refers to the value of the coefficient for $\gamma = 0$ and pP and pT are the corresponding corrective exponents. These two should be specified for each *turbine type*[1]. The layout construction needs the coordinates of each turbine given in meters. The standard set up of the control variables value is zero. For the calculation of the power, the equation of the *actuator disk model* is implemented:

$$P = \frac{1}{2}\rho A U_\infty^3 C_p e \tag{3.5}$$

where P is the produced power, ρ is the air density, A is the turbine area, U_∞ is the wind speed and e is the generator efficiency. The power output of the complete wind farm is the sum of the individual turbines power generation.

3.3 Sub Models

As mention in Chapter 2, there are four main aspects to consider in order to ensure a correctly wake simulation using engineering models: *wake deflection*, *wake velocity deficit*, *wake combination* and *added turbulence*. This section will show the mathematical aspects of the models implemented in both versions to simulate these characteristics. Unless it is specified, both software versions use the specific model. More detailed differences between their application inside a specific version will be addressed in a dedicated section inside this chapter. The results and visualizations generated with FLORIS shown in this section where obtained with the wind farm illustrated by the Figure 3.2 using the DTU 10 MW reference turbine. The most relevant features of this generator are mention in the next section and the detailed characteristics can be found in [18]. In this section, only the specific content of the theory used by FLORIS will be addressed. The complete information of each sub model can be found in the respective reference.

[1] Turbine type will be used to make reference to distinct turbines. For example: DTU 10 MW and TUM G1 are turbine types. Unless it is specified, turbine will make reference to a turbine element inside a wind farm.

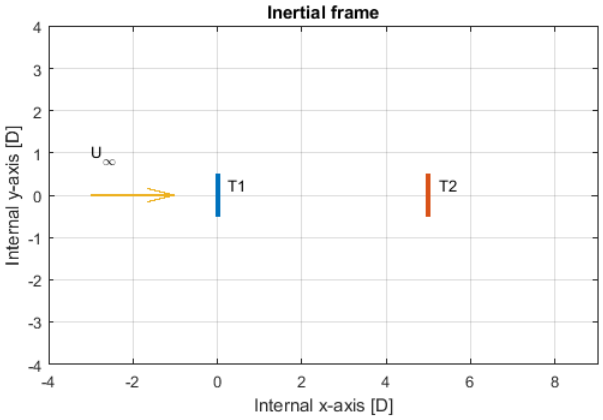
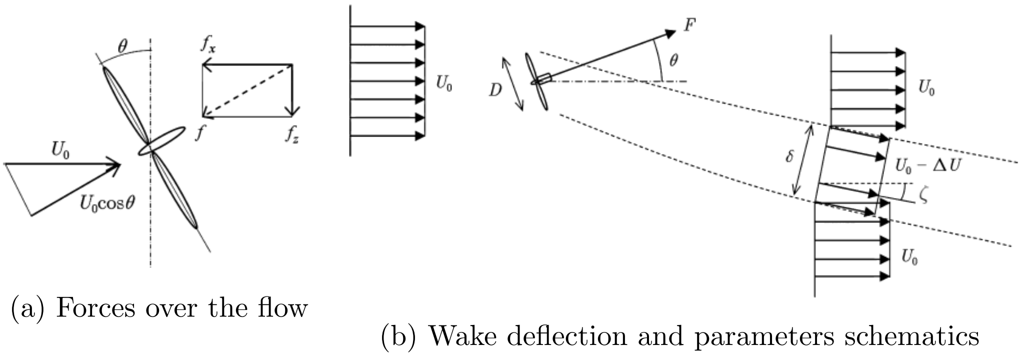


Figure 3.2: 2x1 Wind Farm Layout

3.3.1 Wake Deflection Models

Inside FLORIS are included two different wake deflection models: *Jimenez Model* and *Self-Similar Deflection Model.*

Jimenez Model



(a) Forces over the flow

(b) Wake deflection and parameters schematics

Figure 3.3: Jimenez Model Schematics. Taken from [4].

Jimenez *et al.* at [4] proposes a simple model for the wake deflection cause by the turbine forces acting over the flow, using the *top-hat model* for accounting the velocity deficit. These equations were found under the next assumptions: the control volume is far enough so that the pressure field may be approximated as uniform, the resulting error is due the assumption that $U_\infty - \Delta U \approx U_\infty$ is negligible and that α is sufficiently small such that $\cos\alpha \approx 1$ and $\sin\alpha \approx \alpha$. The complete derivation of such forces can be found in the article. The forces exerted by the turbines over the flow can be expressed as:

$$f_x \approx -\rho U_\infty \Delta U \frac{\pi\delta^2}{4} \tag{3.6}$$

$$f_z \approx -\rho U_\infty^2 \frac{\pi\delta^2}{4}\alpha \tag{3.7}$$

where f_x is the streamwise force, f_z is the transverse force, ΔU is the induce velocity change, δ is the wake width and ζ is the skew angle between the wind direction and the wake direction. Figure 3.3 illustrates these quantities.

ζ is the variable of interest for FLORIS. The article proposes an expression for ζ as a function of δ, γ (in the article, it is in function of the yaw angle, but for this case and according the definitions previously given, $\theta = \gamma$) and the C_t:

$$\zeta = \left(\frac{D}{\delta}\right)^2 \cos^2\gamma \sin\gamma \frac{C_t}{2} \tag{3.8}$$

where D is the turbine diameter. However, in the implementation this function is considered right just behind the turbine[2], where $D = \delta$, thus:

$$\zeta|_{x=0} = \zeta_0 = \cos^2\gamma \sin\gamma \frac{C_t}{2} \tag{3.9}$$

The initial wake deflection expression is accompanied with an equation to describe the wake displacement:

$$dy = \frac{15\zeta_0 \left(\frac{K_{dY}x}{R} + 1\right)^4 + \zeta_0^2}{\frac{15K_{dY}}{R}\left(\frac{K_{dY}x}{R} + 1\right)^5 - \zeta_0 R \frac{15+\zeta_0^2}{15K_{dY}}} \tag{3.10}$$

where K_{dY} is wake deflection recovery factor.

Self-Similar Deflection Model

Bastankhah and Porté-Agel in [5] mention that research does not pay much attention to the far wake region. They proposed a wake deflection model that will treat differently the wake displacement depending on the streamwise position of interest, separating the wake into two sections: *near wake* and *far wake*. Here it mentions how, according to vortex theory, apart from the component normal to the turbine of the induce velocity, there is also a tangential component:

$$U_T = a \tan\frac{\chi}{2} \tag{3.11}$$

where a is the axial induction factor and χ is the angle between the deflected wake and the rotor axis. The value of χ at the rotor disk can be approximated as:

$$\chi = \zeta + \gamma = (0.6a + 1)\gamma \tag{3.12}$$

Furthermore, the C_t can be related with a using the model proposed by Glauert [19]:

$$C_t = \frac{4aU_R}{U_\infty} = 4a\sqrt{1 - a(\cos\gamma - a)} \tag{3.13}$$

where U_R is the wind speed at the rotor. In the paper, the authors develop an expression that does not required to be solve numerically for every value of C_t. By stating that a is negligible in comparison with $2\cos\gamma$, especially for low values of γ, and knowing that $\sqrt{1-x}$ generally asymptotes to $1 - \frac{1}{2}x$, the equation (3.13) can be approximated to the expression:

$$C_t \approx 4a(a - a\cos\gamma) \tag{3.14}$$

[2] Equation 20 at [4].

and by solving this for a in function of C_t, one obtains:

$$a \approx \frac{1}{2\cos\gamma}(1 - \sqrt{1 - C_t \cos\gamma}) \tag{3.15}$$

By rewriting equation (3.12) for ζ and replacing in it the approximate expression for a (3.15), the model's wake skew angle equation is obtained:

$$\zeta \approx \frac{0.3\gamma}{\cos\gamma}(1 - \sqrt{1 - C_t \cos\gamma}) \tag{3.16}$$

This equation is intended to determine the wake skew angle at the generator, nonetheless, it is use for determining the deflection in the complete wake, as it estimates this angle satisfactorily even for few downstream diameters away. For the wake displacement it depends if it is near wake or far wake. x_0 is the stream wise position where the far wake begins. This distance can be determine by the generalization of the model proposed by Lee and Chu [20] for the variation of the shear layer width, by not only taking into account that it is affected by the velocity difference between the *potential core* and the unperturbed flow, but also considering that the incoming turbulence increases the shear layer growth. As defined in the article, the *potential core* is a region of uniform velocity U_C after the exit of a jet when this last is in coflow. Then, the variation of the width of the shear layer can be represented with:

$$\frac{1}{U_\infty}\frac{ds}{dt} = \frac{U_S}{U_\infty}\frac{ds}{dx} = \alpha I + \beta\frac{U_s}{U_\infty} \tag{3.17}$$

where s is the characteristic width of the shear layer, t is time, U_S is the characteristic velocity of the shear layer (which is equal to $0.5(U_\infty + U_C)$), I is the turbulence intensity, U_s is the relative velocity of the shear layer (which is equal to $0.5(U_\infty - U_C)$), and α and β are constants. The velocities ratio U_C/U_∞ resulting from replacing the expressions of U_S and U_s in (3.17) can be found by applying the Bernoulli equation in both upwind and downwind side of the turbine, then solving the system of equations for the pressure drop across the rotor and then replacing it for $0.5\rho U_\infty^2 C_t$. The complete procedure will not be shown in this work, its final result is:

$$\frac{U_C}{U_\infty} = \sqrt{1 - C_t} \tag{3.18}$$

Apart from this, there are other quantities that must be define in order to solve the differential equation. One of them is the relationship between U_R and U_∞. A function of U_R in terms of C_t can be found by replacing (3.15) in (3.13):

$$\frac{U_R}{U_\infty} = \frac{C_t \cos\gamma}{2(1 - \sqrt{1 - C_t \cos\gamma})} \tag{3.19}$$

Another variable of interest is the wake width in y direction at the beginning of the far wake region σ_{y0}. As these calculations are only for yawed turbines, the frontal area of the generator in confront with the wind direction will be assumed to be elliptical and therefore the wake will have the same cross-section geometry. Knowing this, and calculating the streamwise momentum deficit flow rate at $x = 0$ and x_0, is possible to develop expressions for σ_{y0} and the wake width in z direction[3]

[3] In this article, z is the direction normal to the ground.

at the beginning of the far wake σ_{z_0}:

$$\frac{\sigma_{z_0}}{D} = \frac{1}{2}\sqrt{\frac{U_R}{U_\infty + U_C}} \tag{3.20}$$

$$\frac{\sigma_{y_0}}{D} = \frac{\sigma_{z_0}}{D}\cos\gamma \tag{3.21}$$

As the reader may have already notice by now, it is of interest to express the relevant variables as functions of the C_t. For the wake widths, this is achieved by substituting the velocity ratios defined by (3.18) and (3.19) into equation (3.20):

$$\frac{\sigma_{z_0}}{D} = \sqrt{\frac{1+\sqrt{1-C_t\cos\gamma}}{8(1+\sqrt{1-C_t})}} \tag{3.22}$$

With all the previous definitions is now possible to find x_0 in terms of C_t. For solving (3.17) one can integrate as:

$$\int_0^{\sigma_{y0}} ds = \frac{U_\infty}{U_S}\left(\alpha I + \beta\frac{U_s}{U_\infty}\right)\int_0^{x_0} dx \tag{3.23}$$

This integration, and the substitution for a function in terms of C_t yields:

$$\frac{x_0}{D} = \frac{\cos\gamma(1+\sqrt{1-C_t})}{\sqrt{2}[4\alpha I + 2\beta(1-\sqrt{1-C_t})} \tag{3.24}$$

The constants α and β values are determined by comparison of this equation results with measured data. For this particular study and the data here presented, $\alpha = 0.58$ shows satisfactory results. Equation (3.24) shows good accordance with the values reported in jet flows studies when $2\beta = 0.154$. The normalized wake deflection δ/d in the near wake region is given by:

$$\frac{\delta}{d} = \zeta_{C0}\frac{x}{d} \tag{3.25}$$

where ζ_{C_0} is the skew angle at the center of the wake. While the deflection at $x > x_0$ is given by:

$$\frac{\delta}{D} = \zeta_{C_0}\frac{x_0}{D} + \frac{\zeta_{C_0}}{14.7}\sqrt{\frac{\cos\gamma}{k_y k_z C_t}}(2.9+1.3\sqrt{1-C_t}-C_t)\times ln\left[\frac{(1.6+\sqrt{C_t}\left(1.6\sqrt{\frac{8\sigma_y\sigma_z}{D^2\cos\gamma}}-\sqrt{C_t}\right)}{(1.6-\sqrt{C_t}\left(1.6\sqrt{\frac{8\sigma_y\sigma_z}{D^2\cos\gamma}}+\sqrt{C_t}\right)}\right] \tag{3.26}$$

where k_y and k_z are the wake growth rates in the spanwise and vertical directions. This is the last equation necessary to build this wake deflection model.

Is relevant to mention that FLORIS calculates the axial induction factor in terms of the C_t by applying (3.15), regardless of the submodels chosen.

3.3.2 Wake Velocity Deficit Models

The models included in both versions of FLORIS are: *Jensen Model*, *self-similar Deficit Model* and *Zones Model.*

Jensen Model

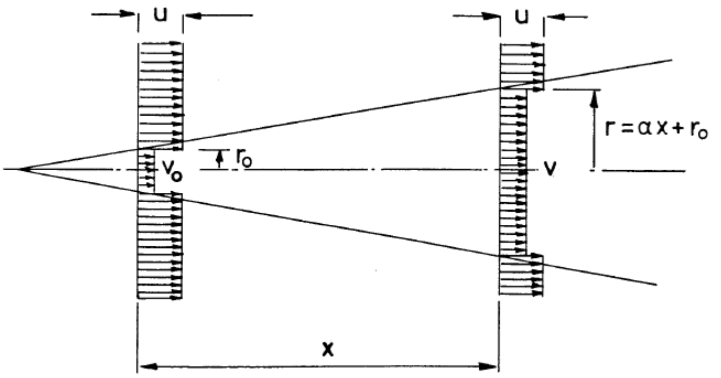


Figure 3.4: Jensen Deficit Model Schematics. Taken from [6].

Jensen in [6] addresses the wake as a turbulent wake, by neglecting the contribution of the generated vortices behind the turbine. This approach entails to the assumption of a spread of momentum deficit that is linear to the streamwise distance x. The momentum balance with such characteristics is:

$$\pi r_0^2 v_0 + \pi(r^2 - r_0^2)u = \pi r^2 v \tag{3.27}$$

where r_0 is the wake's half width just behind the rotor, v_0 is the flow velocity just behind the rotor, r is the wake's half width at distance x from the turbine. Figure 3.4 shows this quantities and the linear wake expansion equation, where α is the wake expansion coefficient. Solving (3.27) for v one obtains:

$$v = u\left[1 - G\left(\frac{r_0}{r_0 + \alpha x}\right)^2\right] \tag{3.28}$$

The value of coefficient G in the original publication appears as 2/3, nonetheless, the source provided in the FLORISSE M documents has what seems to be a hand-made annotation where $G = 2a$. This last value of G is the one implemented by FLORIS. An additional observation is that, because this model is use for the computation of a deficit, the dedicated files of the software only have implemented the second component of equation (3.28).

Figure 3.5 shows the visualization generated via FLORIS using the Jensen Deficit Model. Additional sub models used are: Jimenez for wake deflection, Quadratic Ambient Velocity model for wake combination, and Crespo-Hernandez for added turbulence.

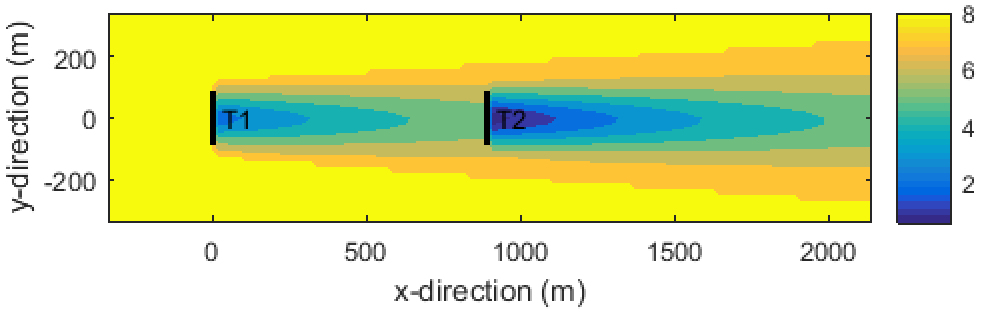


Figure 3.5: FLORIS-Generated Jensen Deficit Model Visualization.

Self-Similar Deficit Model

This model is described by Bastankhah and Porté-Agel in [5] as well. Is known that bluff-bodies wakes become *self-similar* at a specific downwind distance and can be characterized with a *Gaussian Distribution*. This model receives its name from the fact that turbine's wakes also show this behaviour. As matter of fact, in the MATLAB® version, the Jensen deficit model implementation multiplies the wake intensity reduction factor (the denominator of the velocity deficit term in equation (3.28)) by a Gaussian variable to take into account this phenomena. Other studies have already shown that the wake's velocity profile of not yawed turbines quickly achieve self-similarity. Indeed, deflected wakes of yawed turbines also might be consider self-similar. Thus, in general it might be assumed that the lateral velocity profiles in the far wake are self-similar. Analogous with the wakes of bluff-bodies, the generator self-similar wake can be acceptably described (with exception of the extremes) by a Gaussian distribution define by the expression:

$$\frac{\Delta\bar{U}}{\Delta\bar{U}_c} = e^{-0.5(y_*/\sigma_y)^2} \tag{3.29}$$

where $\Delta\bar{U}$ is the velocity deficit, $\Delta\bar{U}_c$ is the maximum velocity deficit, y_* is the lateral distance from the wake center and σ_y is the wake's lateral width. Wind turbine velocity profile self-similarity occurs when the profiles of $\Delta\bar{U}/\Delta\bar{U}_c$ against y_*/σ_y at different streamwise distances collapses into a single curve.

Figure 3.6 illustrates the variables taken into account by both the Self-Similar models, actually, the Gaussian variation of the wake lateral velocity profile can be seen. x_0 is the streamwise distance at which the far wake starts. Thus, the far wake is the region where the lateral velocity distribution becomes self-similar. The figure also helps to illustrate some concepts explained at subsection 3.3.1. With equation (3.24) this distance may be computed. As for the wake deflection, the velocity deficit is treated differently according if it is near wake or far wake region. It is also visible in Figure 3.6 the free shear layer generated by the velocity difference between the potential core and its surroundings, where the velocity increases progressively from U_C at the center to U_∞ with the lateral distance. This means that the velocity is dependent of the lateral distance r, therefore, the lateral velocity distribution for the near wake region $0 \leq x \leq x_0$ can be represented with:

$$\frac{U}{U_\infty} = \begin{cases} 1 - C_0 & \text{if } r \leq r_{pc} \\ 1 - C_0 e^{-(r-r_{pc})^2/2s^2} & \text{if } r \geq r_{pc} \end{cases} \tag{3.30}$$

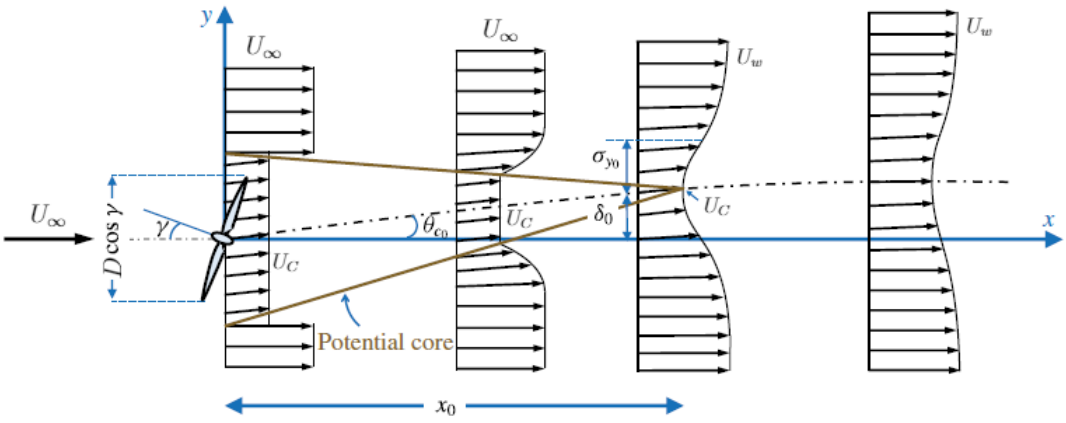


Figure 3.6: Self-Similar Model Wake Schematics. Taken from [5].

where C_0 is the velocity deficit at the wake center at x_0 normalized with the wind speed, r_{pc} is the radius on the lateral plane of the potential core and s is the characteristic width of the shear layer. For the far wake, the velocity profile is described by the expression:

$$\frac{\Delta\bar{U}}{\bar{U_\infty}} = \left(1 - \sqrt{1 - \frac{C_t\cos\gamma}{8(\sigma_y\sigma_z/D^2)}},\right) e^{-0.5[(y-\delta)/\sigma_y]^2} e^{-0.5[(z-z_h)/\sigma_z]^2} \tag{3.31}$$

where δ is the wake deflection. In other hand, the wake widths σ_y and σ_z are given by the equations:

$$\frac{\sigma_y}{D} = k_y\frac{x-x_0}{D} + \frac{\cos\gamma}{\sqrt{8}} \tag{3.32}$$

$$\frac{\sigma_z}{D} = k_z\frac{x-x_0}{D} + \frac{1}{\sqrt{8}} \tag{3.33}$$

where k_y and k_z are the wake expansion coefficients in the respective direction.

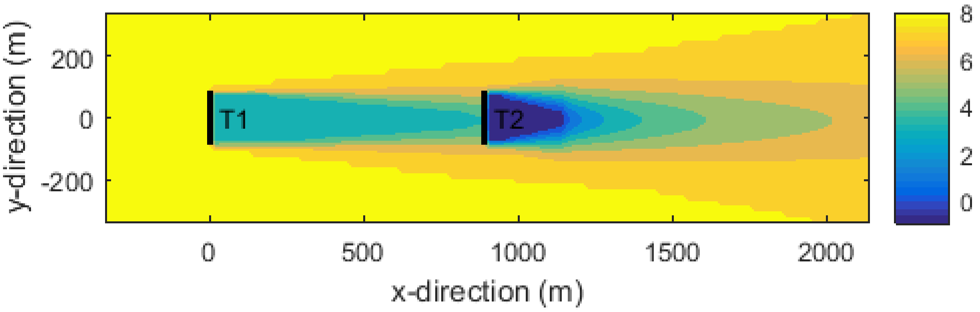


Figure 3.7: FLORIS-Generated Self-Similar Deficit Model Visualization.

Figure 3.7 shows the visualization generated via FLORIS using the Self-Similar Deficit Model. Additional sub models used are: Jimenez for wake deflection, Quadratic Ambient Velocity model for wake combination, and Crespo-Hernandez for added turbulence.

Zones Model

Gebraad *et al.* in [7] presents a parametric model based on the Jensen Deficit model and the Jimenez Deflection model. This model was augmented to empirically fit with their measurements obtained in the experiments described in the article. The base deficit model was created under the assumption of a linear expansion wake with respect to the downwind distance from the generator and a wake with uniform lateral velocity. The approach proposed by the authors, in order to improve the model's data with the experiments, is to divide the wake in three sub-zones, each of them yet expanding linearly with the distance but each with its own expansion factor. Sub index q will be used to denote each region: $q = 1$ corresponds to the near wake, $q = 2$ corresponds to the far wake and $q = 3$ corresponds to the mixing zone. The diameter of wakes zones $D_{w,q}$ are given by:

$$D_{w,q} = max(D + 2k_e m_{e,q} x, 0) \tag{3.34}$$

where k_e and $m_{e,q}$ are the wake expansion coefficients. The last is given for each zone.

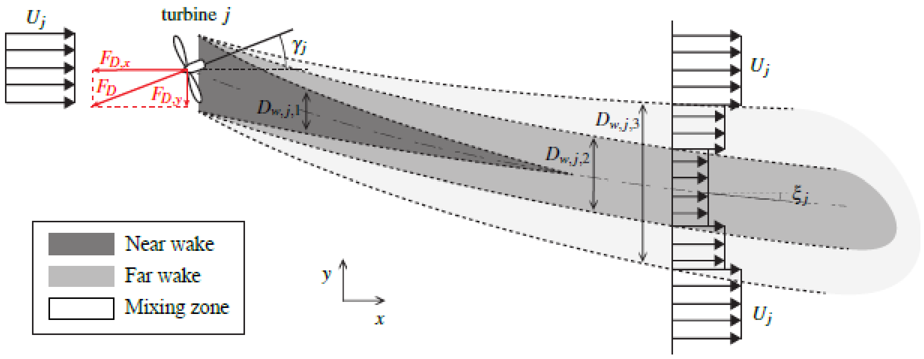


Figure 3.8: Zones Model Schematics. Taken from [7].

Figure 3.8 helps to visualize the division of the sub zones proposed in the article. In this work it is proposed that the velocity deficit of the wake decays quadratically with the downwind distance from the turbine, instead of with the wake expansion as Jensen had suggested in his model. For the new model, the velocity profile of the wake is expressed by:

$$U_w(x, r) = U[1 - 2ac(x, r)] \tag{3.35}$$

where r is the lateral distance from the wake center, and, similarly to Jensen's model, the second term inside the parenthesis is the velocity deficit. c is the wake decay coefficient who, depending on the zone, has different values given by the function:

$$c_q(x) = \left[\frac{D}{D + 2k_e m_{U,q}(\gamma)x}\right]^2 \tag{3.36}$$

where $m_{U,q}$ is a multiplier function of independent variable γ given by the expression:

$$m_{U,q} = \frac{M_{U,q}}{\cos a_U + b_U \gamma} \tag{3.37}$$

in which a_U, b_U and $M_{U,q}$ are parameters with their value adjusted to ensure the best accordance possible with the experimental data specified in the article; $a_U = 5$,

$b_U = 1.66$, $M_{U,1} = 0.5$, $M_{U,2} = 1$ and $M_{U,3} = 5.5$. The last of them receives the name of relative recovery of wake zones inside FLORIS. The logic used for determining the value of c in equation (3.35) is described by the next conditional:

$$c(x,r) = \begin{cases} c_1 & \text{if } r \leq D_{w,1}/2 \\ c_2 & \text{if } D_{w,1}/2 < r \leq D_{w,2}/2 \\ c_3 & \text{if } D_{w,2}/2 < r \leq D_{w,3}/2 \\ 0 & \text{if } r > D_{w,3}/2 \end{cases} \tag{3.38}$$

As an additional note, this model receives different names in both versions of the software; in FLORISSE M is call "Zones", while in the Python version it is called "floris". Nonetheless, they are the same model.

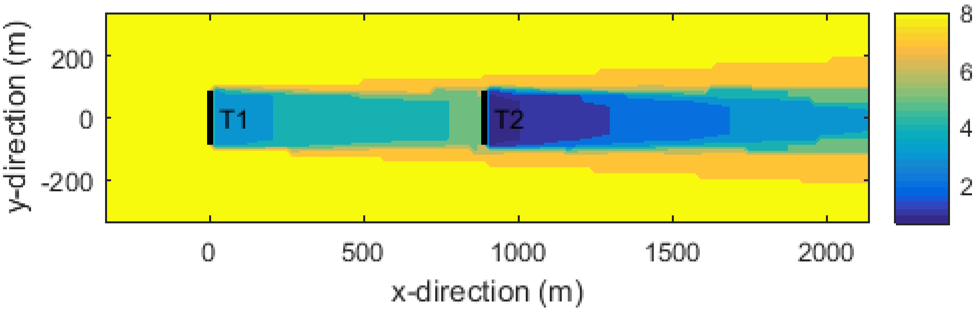


Figure 3.9: FLORIS-Generated Zones Deficit Model Visualization.

Figure 3.9 shows the visualization generated via FLORIS using the Zones Deficit Model. Additional sub models used are: Jimenez for wake deflection, Quadratic Ambient Velocity model for wake combination, and Crespo-Hernandez for added turbulence.

Larsen Model

This velocity deficit model is exclusively implemented on the MATLAB® version. This model was developed by Larsen in [8] under the assumption that the wake generated by a wind turbine can be satisfactorily represented with the Prandtl's turbulent boundary layer equations, given that these are turbulent wakes. The wake is also assumed to be symmetric. These equations, in cylindrical coordinates, are expressed as:

$$\frac{\partial}{\partial x}(u_x r) + \frac{\partial}{\partial r}(u_r r) = 0 \tag{3.39}$$

$$(U_\infty + u_x)\frac{\partial u_x}{\partial x} + u_r\frac{\partial u_x}{\partial r} = \frac{1}{r}\frac{\partial}{\partial r}\left[l^2 r\left(\frac{\partial u_x}{\partial r}\right)^2\right] \tag{3.40}$$

where u_x and u_r are the axial and radial velocity respectively, r is the radius of the wake and l is the mixing length.

The article proposes a first order wake model and a second order wake model. The one implemented in FLORIS is the first one. To find the simplified equations, it is assumed that for high Reynolds numbers the velocities perpendicular to the streamwise direction are geometrically similar, also that the size of the boundary of the turbulent wake is proportional to x^n, the wind shear is neglected, and the flow is stationary and incompressible. After some steps, it is shown in the paper that n is found to be equal to 1/3. After all the necessary computations it is found that the momentum equation is reduced to:

$$U_\infty \frac{\partial u_x}{\partial x} = \frac{1}{r}\frac{\partial}{\partial r}\left[l^2 r \left(\frac{\partial u_x}{\partial r} \right)^2 \right] \tag{3.41}$$

by only taking into account the terms of order of magnitude $x^{-\frac{5}{3}}$. Equations (3.39) and (3.41) are the system of equations of the first order proposal. Following the procedure described in the reference for solving the system, the equation of velocity deficit is found:

$$u_x = -\frac{U\infty}{9}(C_t A x)^{\frac{1}{3}} \left[r^{\frac{3}{2}} (3c_1^2 C_t x)^{-\frac{1}{2}} - \left(\frac{35}{2\pi} \right)^{\frac{3}{10}} (3c_1^2)^{-\frac{1}{5}} \right]^2 \tag{3.42}$$

For the sake of completeness, the simplified expression for the radial velocity u_r is found to be:

$$u_r = \frac{U_\infty}{3}(C_t A)^{\frac{1}{3}} x^{-\frac{5}{3}} r \left[r^{\frac{3}{2}} (3c_1^2 C_t A x)^{-\frac{1}{2}} - \left(\frac{35}{2\pi} \right)^{\frac{3}{10}} (3c_1^2)^{-\frac{1}{5}} \right]^2 \tag{3.43}$$

There are two unknowns in these equations: the constant c_1 and x_0 that is the position of the generator. Two conditions must be satisfied: first, a measure of the velocity at a point $x_0 + \Delta x$ over the axis of symmetry must be given, this to simplify the equations and because it is believe that probably the influence of the geometry of the body that generates the wake, after some downwind distance, is less on the axis than outside the axis. The second condition: is assumed that the width of the wake in the position of the rotor it's equal to the diameter. However, the second condition is implemented differently inside the software. As it has been derive in [21], assuming a cylindrical control volume with constant cross-section geometry, the area of the rotor and the area just behind the rotor can be related via the axial induction factor, this means that the diameters can be related via the C_t. It is implemented the relationship of both diameters defined by the expression:

$$D_a = D\sqrt{\frac{1 + \sqrt{1 - C_t}}{2\sqrt{1 - C_t}}} \tag{3.44}$$

where D_a is the diameter of the wake just behind the turbine. This last is the measure used inside the equations that determines the value of the system's unknowns. Thus, x_0 and c_1 can be found via:

$$x_0 = \left[\left(\frac{D_a}{2\alpha_1} \right)^{-3} \left(\frac{\beta_1}{U_\infty - U_m} \right)^{\frac{3}{4}} - 1 \right]^{-1} \Delta x \tag{3.45}$$

$$c_1 = \left(\frac{D_a}{2\alpha_1} \right)^{\frac{5}{2}} x_0^{-\frac{5}{6}} \tag{3.46}$$

where U_m is the required measure velocity, and constants α_1 and β_1 are given respectively by:

$$\alpha_1 = \left(\frac{105}{2\pi}\right)^{\frac{1}{5}} (C_t A)^{\frac{1}{3}} \tag{3.47}$$

$$\beta_1 = \frac{1}{9}\left(3^{-\frac{2}{5}}\right)\left(\frac{35}{2\pi}\right)^{\frac{3}{5}} (C_t A)^{\frac{1}{3}} \tag{3.48}$$

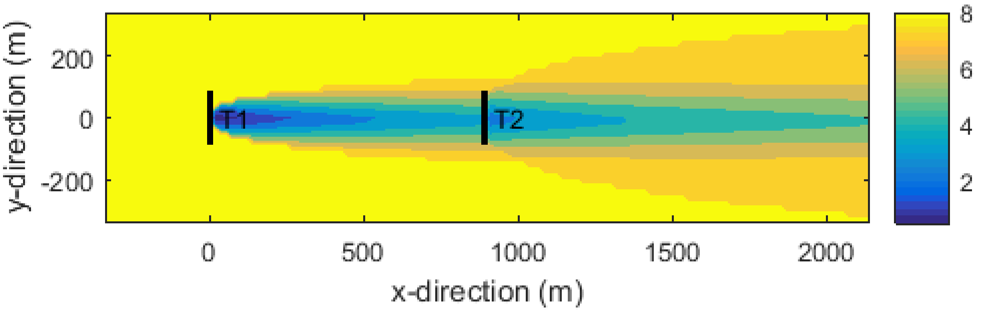


Figure 3.10: FLORIS-Generated Larsen Deficit Model Visualization.

Figure 3.10 shows the visualization generated via FLORIS using the Larsen Deficit Model. Additional sub models used are: Jimenez for wake deflection, Quadratic Ambient Velocity model for wake combination, and Crespo-Hernandez for added turbulence.

3.3.3 Wake Combination Models

These models are indeed the principal difference between both versions of FLORIS, not only the models used are different but also how the software implements this wake characteristic is particular of each version. As said in Chapter 2, this model does not only take into account the interaction between two different wakes, but also the one between the ambient flow and just one wake.

Python Version

This version uses a simple sum for simulating this phenomenon. As once the upwind flow field is calculated, if there is an interaction between this and a wake, both the velocities are added into a new variable called combined wake velocity U_W. This last is then subtracted from the initial flow field U_∞ to find the velocity resulting from the interaction of both fields. How the sum is done varies according the selected model.

Freestream Linear Superposition-FLS. This model makes the addition directly. Defining as U_{ff} the velocity of the upwind flow field and as U_w the velocity of any wake, the resulting speed caused by the interaction of the flows is:

$$U_W = U_{ff} + U_w \tag{3.49}$$

Sum Of Squares Freestream Superposition-SOSFS. As it name insinuates, this model uses the Pythagorean equation, where the resulting velocity is equal to the sum of the squares of the fields interacting:

$$U_W^2 = U_{ff}^2 + U_w^2 \tag{3.50}$$

MATLAB® Version

The models of this version are base in the proposition described by Katic *et al.* in [22]. The wake is assumed to have linear expansion as the original Jensen deficit model, and the wake combination is treated by the hypothesis that the kinetic energy deficit of a combined wake is assumed to be equal to the sum of the individual energy deficit of the wakes involved. This concept is presented mathematically in the article as:

$$\left(1 - \frac{U_W}{U_M}\right)^2 = \left(1 - \frac{U_{w1}}{U_M}\right)^2 + \left(1 - \frac{U_{w2}}{U_M}\right)^2 \tag{3.51}$$

where U_M is a velocity that depends of the chosen model (the one proposed in the article will be with $U_M = U_\infty$) and $U_{w,j}$ are the wake speeds. In the equation is assumed that the interaction between the wakes 1 and 2 gives as result the speed U_W.

As described at [23], for the power calculation, the wakes are combined according to the relative overlap area for finding a rotor-average and time-average turbine wind speed. This makes necessary the definition of the volumetric flow rate through the rotor plane $Q_{R,j}$ to describe the wake of the upwind turbine j. Thus, in polar coordinates, the flow rate can be expressed as:

$$Q_{R,j} = \int_0^R \int_0^{2\pi} \left(r \left[1 - \frac{U_{w,j}(r,\beta)}{U_\infty(r,\beta)} \right] \right) d\beta dr \tag{3.52}$$

where β is the azimuth angle and r is the radial distance from the center of the turbine.

Quadratic Ambient Velocity. With the definition of $Q_{R,j}$, this model uses the ambient wind speed for the calculation. The rotor effective wind speed of a given turbine is then:

$$U_R = U_\infty \left[1 - \sqrt{\sum_{j=1}^{N_t} \left(\frac{Q_{R,j}}{\pi R^2} \right)^2} \right] \tag{3.53}$$

For this model, $U_M = U_\infty$ in equation (3.52).

Quadratic Rotor Velocity. The difference between this and the other model is simply the velocity used outside the parenthesis. This uses the average speed of the upwind rotor U_{uw}:

$$U_R = U_{uw} \left[1 - \sqrt{\sum_{j=1}^{N_t} \left(\frac{Q_{R,j}}{\pi R^2} \right)^2} \right] \tag{3.54}$$

For this model, $U_M = U_{uw}$ in equation (3.52).

3.3.4 Added Turbulence Model

FLORIS uses the **Crespo-Hernandez model**. The equation is found in [24], it was developed by relating the turbulent kinetic energy k with the standard deviation of the wind direction, by assuming that the anisotropy of the wake is similar to the basic atmospheric flow. The region just after the rotor is not considered. The software only uses the far wake added turbulence contribution described on the article. With parameters $5 < x/D < 15$, $0.07 < I_0 < 0.14$ and $0.1 < a < 0.4$, the model is given by the equation:

$$I_+ = 0.73a^{0.8325}I_0^{0.0325}\frac{x}{D}^{-0.32} \tag{3.55}$$

where I_0 is the ambient turbulence intensity and x is a downwind streamwise position. The equation was found with the best least square fitting with their numerical data using UPMWAKE, a 3D code used to predict the turbulent characteristic of individual wakes. Other added turbulence models are presented in [25].

Although both versions of FLORIS have exclusively implemented this model, the Python version has included it as a default preset and not like an optional choice like the other submodels. Meanwhile, FLORISSE M has included it with its own class and module, giving flexibility to the program if another added turbulence model is going to be implemented and used.

3.4 MATLAB® vs Python

The differences between both versions of the software will be addressed in this section. The results here to be shown have been found with the layout illustrated by Figure 3.2 composed of two aligned DTU 10MW wind turbines, with a $5D$ separation between each generator. The main characteristics of this generator are shown in the Table 3.1. This turbine was chosen because is the only one inside FLORISSE M who has available the "yawAndRelPowerSetpoint" control option. Table 3.2 shows the flow conditions used in the simulation referred in this section. Shear factor zero means that a uniform incoming velocity profile was used. All the simulations where done with the Crespo-Hernandez added turbulence model (the only one available) and for wake combination, in FLORISSE M was used QAV and in Python was used SOSFS. These models won't be specified again during this section for simplicity.

DTU 10MW	
Rotor Diameter D [m]	173.80
Hub Height z_h [m]	119.00
Rated Wind Speed U_{rated} [m/s]	11.40
Rated Power P_{rated} [MW]	10.00
Rotor Efficiency e	1.00
pT	1.88
pP	1.88

Table 3.1: Principal Characteristics of the DTU 10MW wind turbine.

Case Data	
Wind Speed U_∞ $[m/s]$	8.000
Air Density ρ $[kg/m^3]$	1.225
Wind Direction [°]	0.000
Turbulence Intensity TI	0.060
Shear Factor	0.000

Table 3.2: Ambient conditions of the evaluated case.

3.4.1 Main Differences

Is worth to clarify that the objective of this comparison was to confirm that both versions won't be able to find the same numerical results even with all the initial parameters being the same. Afterwards, to decide which one would be analyze in detailed and enhance it. This means that the differences here to be mention are probably not all that exists, but the most relevant for accomplishing this goal. The main differences found between both programs are resume in Table 3.3. There are some other features that where noticed, but they can be consider intrinsic of one mentioned in the table. Each listed element will have a dedicated paragraph to further explanation. The characteristics with * have been mention in former sections but will have breve further comments and the characteristics with ** have been already treated and won't be treated again.

Characteristic	MATLAB®	Python
Unique Input File	NO	YES
Ambient Velocity Profile Object	YES	NO
Asymmetric Wake with Jimenez	YES	NO
Larsen Deficit Model*	YES	NO
Modified Jensen Model**	YES	NO
Constants Included in SS-Equation	NO	YES
Velocities Combination Models**	NO	YES
Kinetic Energy Combination Models**	YES	NO
Added Turbulence Models Object**	YES	NO
Control Methods Choice Option	YES	NO
Thrust Angle ψ	YES	NO
Graphic Problem with Big Domains	NO	YES

Table 3.3: Main differences between versions of FLORIS.

Unique Input File

The Python version counts with a template in .json format of how all the inputs, both case and specific inputs, should be given to the program. This facilitates the setting of the simulations. In other hand, FLORISSE M does not has anything similar. This version counts with a group of templates to run simulations under different ambient velocity profiles, where the only information needed are some of the case inputs, for example, there is no need to specified the yaw or tilt angles because their default value is defined somewhere inside the code, while in the input file of the Python version this is included.

The disadvantages of having all inputs in a single file might be that it's easier to change something with huge impact by mistake and that all the information of all the models must be included, even for those that are not going to be used. This last can be easily solve as the information of the unused model can be set randomly. Nevertheless, this is not a recommended practice as often is desirable to simulate the same scenarios with different models for result's comparison. In other hand, the disadvantage of having all the inputs spread in different files is that the initial setting of a simulation can be considerably slower and is easier to overlook any necessary change.

There are also some data that one version consider input and the other does not. The default setting of the control variables is an example. Additionally, Python ask for an initial value of the TSR and for the axial induction factor to use in the Crespo-Hernandez model, while MATLAB® calculates and update these quantities. Another difference is the intricacy of the C_p and C_t inputs. FLORISSE M requires, for some of the control methods options, Look Up Tables (LUT) which are complex numerical arrays of the value of the coefficients in function of two or more variables instead than just as function of the wind speed. For example, for the control method "*yawAndRelPowerSetpoint*" it is use a LUT of the coefficients where the rows are the wind speed, the columns are the set point (which is believe to be the efficiency) and every 2D array is given for a specific yaw angle. Although this might be a little problematic, it gives precision to the results as there is no need to correct the coefficients with the cosine functions given by the equations (3.3) and (3.4). Python instead only needs an array of these dimensionless quantities in function of the wind speed, correcting their value for deflected turbines with the aforementioned equations. Another detail is that the wind directions are not the same. The positive streamwise direction in FLORISSE M is 0 degrees while in Python is 270 degrees.

Ambient Velocity Profile Object

As mentioned in Section 3.1 the program was developed using OOP with the objective of giving flexibility to add, update or correct the models that it includes. Inside the case inputs, it can be found all the flow properties necessary to simulate the ambient velocity profile. MATLAB® defines the ambient inflow as an object. The user has the possibility of writing different scripts for different velocity functions that describe the velocity profile. For example, the file "*ambient_inflow_log.m*" uses a logarithmic distribution for the velocity profile, while the file "*ambient_inflow_uniform.m*" uses a uniform distribution. The last of them will not be in need of the shear factor, so it does not ask to specify it. The advantage of this implementation is the flexibility and that there is no need to rewrite anything in the main code, just create another object for a different inflow profile. Instead, the Python version does not has anything similar to this. Inside the file "*flow_field.py*", the velocity profile is calculated automatically with a logarithmic function. If the user wants to simulate a uniform case, the specified shear factor (here named "wind_shear") with a value of zero in the inputs file.

Asymmetric Wake with Jimenez

Jimenez is one of the two available wake deflection models in both versions. The implementation of the equation 3.8 is really similar, on Python the equation has a $\cos\gamma$ while in FLORISSE M there is no $\cos\gamma$. This is due the fact that the values of the coefficients, when they arrive to this step, are already corrected with the cosine function. The main difference between the versions is that in MATLAB® is implemented an initial wake deflection, independent of the yaw misalignment of the turbine. With this, it is possible to simulate the asymmetry of the wake cause by the sense of rotation of the generator when $\gamma = 0$.

With this initial deflection angle, the algorithm suffers a change. First, the Jimenez equation is applied to find the model's predicted deflection. The Rodrigues' rotation formula:

$$v_{rot} = v\cos\theta + (k \times v)\sin\theta + k(k \cdot v)(1 - \cos\theta) \tag{3.56}$$

where v is a vector, k is a unitary vector that represents the rotation axis and θ is the angle of rotation, is used for rotating the wake direction. The pre-imposed deflection γ_0 is considered using the rotational matrix with respect to the z-axis:

$$R_z = \begin{bmatrix} \cos\gamma_0 & -\sin\gamma_0 & 0 \\ \sin\gamma_0 & \cos\gamma_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.57}$$

A vector is generated by multiplying R_z by the vector v_{rot} and the angle is found by using the arccos of the first element of the resulting vector.

This procedure is present exclusively in the Jimenez model of the MATLAB® version. This means that the Self-Similar model and both models in the Python version have symmetric wakes when $\gamma = 0$. This consideration improves the Jimenez model's physical representation as in reality there is a contribution to the wake shape due the turbine's sense of rotation. Table 3.4 shows the result of both turbines found by the different FLORIS versions. The simulations where done with; deflection model: Jimenez and deficit model: Jensen.

1st Turbine					
Version	Power $[MW]$	C_p	C_t	a	U_∞ $[m/s]$
MATLAB	3.5810	0.4573	0.8066	0.2801	8.0000
Python	3.5807	0.4573	0.8066	0.2801	8.0000
Error [%]	0.0083	0.0000	0.0000	0.0000	0.0000

2nd Turbine					
Version	Power $[MW]$	C_p	C_t	a	U_∞ $[m/s]$
MATLAB	1.4769	0.4282	0.9218	0.3602	6.0872
Python	1.4123	0.4257	0.9286	0.3664	6.0088
Error [%]	4.3740	0.5838	0.7377	1.7213	1.2879

Table 3.4: Comparison between the results found by the different FLORIS versions.

The error calculation was done with the equation:

$$Err_{Rel} = 100 * \left| \frac{d_t - d_e}{d_t} \right| \tag{3.58}$$

where, in a general case, d_t is the theoretical value and d_e is the experimental value. In this context d_t is the FLORISSE M result and d_e is the Python result. It is clear that, even if they are really similar, the results for the second turbine are different, not just because of the asymmetry present in FLORISSE M but also because the models are different. Thus a perfect confrontation with the exact same simulation condition is not possible. However, the results for the first generator can be considered equal (the 0.0083% error can be neglected). This happens because the computation if these results on the first turbine found by the flow do not depend of the submodels used by FLORIS, just depend of the case inputs. Due this, the results of the first turbine will not be shown in cases where the only change is a submodel variation.

The asymmetry here mention can be correctly evidenced with a confront of the results found using the different wake deflection models of a single version, in this case FLORISSE M. Table 3.5 show the results obtained of the second turbine for three different wind directions: 0°, 5.7° and −5.7°. This angles where chosen by assuming a straight wake. Under these conditions, the turbine behind will be working under complete or half wake. The wake deficit model used was Jensen. The error was calculated applying equation (3.58) where d_t is the Jimenez data and d_e is the Self-Similar data. On the Jimenez generated data, is evident the asymmetry of the wake. Depending of the wind direction, even if the magnitude of the deviation is the same, the power generated varies. Meanwhile, independent of the sense of arrival of the wind, for the self-similar deflection model, the results are the same due the

Jimenez Deflection Model					
Wind Direction [°]	Power $[MW]$	C_p	C_t	a	U_∞ $[m/s]$
0.0	1.4769	0.4282	0.9218	0.3602	6.0872
5.7	1.9602	0.4443	0.8767	0.3244	6.6075
−5.7	2.1895	0.4509	0.8581	0.3117	6.8220

self-similar Deflection Model					
Wind Direction [°]	Power $[MW]$	C_p	C_t	a	U_∞ $[m/s]$
0.0	1.4709	0.4279	0.9225	0.3608	6.0800
5.7	2.0668	0.4475	0.8679	0.3183	6.7094
−5.7	2.0668	0.4475	0.8679	0.3183	6.7094

Error [%]					
Wind Direction [°]	Power	C_p	C_t	a	U_∞
0.0	0.4063	0.0701	0.0759	0.1666	0.1182
5.7	5.4382	0.7202	1.0038	1.8804	1.5422
−5.7	5.6040	0.7540	1.1421	2.1174	1.6505

Table 3.5: FLORISSE M Deflection Models Comparison: different wind directions with Jensen Deficit model.

symmetry of the wake. The error of the produced power increase significantly when the second turbine is working in partial wake again because of the nature of the velocity profile that it sees, the asymmetry makes that a half of the wake is more harmful than the other half.

Larsen Deficit Model*

As mention before, the only version of the software with the Larsen Deficit model already implemented is FLORISSE M. The details can be found in Section 3.3.2. Results for the same example case mention in the paragraph before using this deficit model are presented in Table 3.6.

Jimenez Deflection Model					
Wind Direction [°]	Power $[MW]$	C_p	C_t	a	U_∞ $[m/s]$
0.0	1.0349	0.3914	0.9816	0.4255	5.5709
5.7	1.9652	0.4444	0.8763	0.3241	6.6124
-5.7	2.3458	0.4552	0.8462	0.3039	6.9588

self-similar Deflection Model					
Wind Direction [°]	Power $[MW]$	C_p	C_t	a	U_∞ $[m/s]$
0.0	1.0234	0.3902	0.9834	0.4275	5.5559
5.7	2.1517	0.4499	0.8611	0.3136	6.7877
-5.7	2.1517	0.4499	0.8611	0.3136	6.7877

Error [%]					
Wind Direction [°]	Power	C_p	C_t	a	U_∞
0.0	1.1112	0.3066	0.1834	0.4700	0.2693
5.7	9.4901	1.2376	1.7346	3.2397	2.6511
-5.7	8.2744	1.1643	1.7608	3.1918	2.4588

Table 3.6: FLORISSE M Deflection Models Comparison: different wind directions with Larsen Deficit model.

Constants Included in SS-Equation

The specific relationship here to be refer to is the equation (3.24). Recalling, this is the expression that finds the distance x_0 where the far wake region starts in both Self-Similar deflection and deficit models. On the denominator, there are two constants to be determine, α and β, who's tuning with experimental data, will improve the accuracy of the model's prediction. Bastankhah and Porté-Agel in [5] at Chapter 7 write in another way equation (3.24), where the constants this time are α^* and β^*:

$$\frac{x_0}{d} = \frac{\cos\gamma(1+\sqrt{1-C_t})}{\sqrt{2}[\alpha^* I + \beta^*(1-\sqrt{1-C_t})]} \tag{3.59}$$

This is the actual equation that is find in FLORISSE M, neither the 4 or the 2 are included inside the equation. If the user wishes to modified this constants he must take into account that the numerical value must be pre-multiplied by 4 or 2 respectively if he is considering equation (3.24). In Python the equation has the numerical constants so there is no additional consideration necessary. The numerical values of α^*0 and β^* are also reported in [5]: $\alpha^* = 2.32$ and $\beta^* = 0.154$.

Control Methods Choice Option

For executing a simulation with FLORISSE M, the user must choose a control method for the selected turbine. Gathering all the methods available in the already implemented wind turbines, there are four different options; *"pitch"*, *"axialInduction"*, *"greedy"* and *"yawAndRelPowerSetpoint"*. FLORIS does not have the capability of applying a dynamic control as it is only able to simulate steady conditions. The differences between the control methods is the nature of the input LUT. If a given turbine type has all four options, it would count with three different LUT. For "pitch" control method, the LUT will be a 2D-array of both C_p and C_t, where the rows are the various wind speeds and the columns are various pitch angles. If the selection is "greedy", it should be provided a 1D array, of the coefficients in function of different wind speeds. As mention before, the case of "yawAndRelPowerSetpoint" is the more complex of them all, being in need of the 3D array, where the rows are again the wind speed, the columns are a given set point, (which is believed to be the turbine's efficiency) and each 2D array is given for different yaw angles. The "axialInduction" does not need any LUT input, the program will use a default constant value of $a = 1/3$ for the calculations, making this the only control method available for all turbines type by default. The Python version does not count with any characteristic similar to this, however, they might be included by changing manually the input array of C_p and C_t for the appropriate values of the LUT with the coefficients data.

Thrust Angle ψ

The thrust angle ψ is exclusive of FLORISSE M. Some of the equations have replaced the yaw misalignment with ψ. This angle is calculated after considering the yaw and tilt rotations given to the turbines, and calculating the thrust direction caused by these deflections:

$$R_zR_y = \begin{bmatrix} \cos\gamma\cos\varphi & -\sin\gamma & \cos\gamma\sin\varphi \\ \sin\gamma\cos\varphi & \cos\gamma & \sin\gamma\sin\varphi \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \tag{3.60}$$

Equation (3.60) show the rotation matrix for a 2D deflection, where γ is the yaw misalignment and φ is the tilt angle. The rotations are made in that order, first around z-axis (yaw) and then with respect the y-axis (tilt). The resulting matrix is then multiplied by a vector $[1\ 0\ 0]^T$ for obtaining the thrust direction vector E:

$$E = \begin{bmatrix} \cos\gamma\cos\varphi & -\sin\gamma & \cos\gamma\sin\varphi \\ \sin\gamma\cos\varphi & \cos\gamma & \sin\gamma\sin\varphi \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\gamma\cos\varphi \\ \sin\gamma\cos\varphi \\ -\sin\varphi \end{bmatrix} \tag{3.61}$$

The dot product of the resulting vector with again $[1\ 0\ 0]^T$ gives the cosine of ψ:

$$\cos\psi = E \cdot [1\ 0\ 0]^T \tag{3.62}$$

Thus the arccos of the right side gives the thrust angle:

$$\psi = \arccos(E \cdot [1\ 0\ 0]^T) = \arccos(\cos\gamma\cos\varphi) \tag{3.63}$$

So, mathematically, ψ should be the resulting angle of two previous rotations with respect to the new rotation axis, and physically, it is the angle between the thrust direction and the streamwise direction. This is calculated for all the turbines in the farm, and it is found for considering the tilt angle of the turbine as well in the overall wake deflection. In Python there is no similar procedure or quantity, it operates with the usual variables.

Graphic Problem with Big Domains

During the test runs, a wind farm of 5x1 turbines, with 5D of separation between them was simulated. Although it was capable of calculating the quantitative results, the Python version showed an error at the flow field visualization. This issue was not examined in depth, so there is no certainty of the maximum size of the flow field that this version is able to manage for generating its visualization successfully. Is worth to mention that for a farm of 4x1 generators, with the same spacing between the turbines, the program show no inconvenience. FLORISSE M did not show any trouble for any of the attempted domain sizes.

3.5 Version and Models Selection

Both versions are functional and capable of generating results that are congruent with the physics behind the power production of wind farms with yaw deflections, nonetheless, for the interest of the future projects, the version that was chosen to use for further analysis was FLORISSE M. For the submodels selection there are additional aspects to take into account. Clarifying, the analysis was qualitative, there were not used any numerical data to compare with the software results. The wake combination models are really similar, both use the same equation, and QRV can be considered a modification of QAV because this last is the one that uses the same variables found in the reference paper. For all of these, the **QAV model** was chosen. Being the only option already present in FLORIS, the added turbulence model used was **Crespo-Hernandez**.

About the wake deflection model, the key analysis for the election were the simulations at different wind directions. For completeness, and to complement Table 3.5 and Table 3.6, the remaining wake velocity deficit results for that example case are presented in Table 3.7 and Table 3.8. The comparison of both wake deflection models shows that as a whole, the results are similar. The biggest error is of 9.5063% in the power production at a wind direction of 5.7° for the Self-Similar deficit model. Although it is not a negligible error, it can be acceptable as in absolute values the difference is not even 0.2 MW. However, the implementation inside the **Jimenez model** of the effect of the sense of rotation of the generator on the wake is an important physical phenomena that should be taken into account. Thank to this, Jimenez was the selected model.

Jimenez Deflection Model					
Wind Direction [°]	Power $[MW]$	C_p	C_t	a	U_∞ $[m/s]$
0.0	0.9158	0.3785	1.0000	0.4457	5.4084
5.7	1.7157	0.4365	0.8984	0.3406	6.3578
−5.7	2.0514	0.4470	0.8691	0.3191	6.6949

self-similar Deflection Model					
Wind Direction [°]	Power $[MW]$	C_p	C_t	a	U_∞ $[m/s]$
0.0	0.9062	0.3774	0.1	0.4457	5.3946
5.7	1.8788	0.4418	0.8837	0.3295	6.5270
−5.7	1.8788	0.4418	0.8837	0.3295	6.5270

Error [%]					
Wind Direction [°]	Power	C_p	C_t	a	U_∞
0.0	1.0483	0.2906	0.0000	0.0000	0.2552
5.7	9.5063	1.2142	1.6362	3.2590	2.6613
−5.7	8.4138	1.1633	1.6799	3.2592	2.5079

Table 3.7: FLORISSE M Deflection Models Comparison: different wind directions with self-similar Deficit model.

Jimenez Deflection Model					
Wind Direction [°]	Power $[MW]$	C_p	C_t	a	U_∞ $[m/s]$
0.0	1.0137	0.3892	0.9849	0.4293	5.5431
5.7	1.9807	0.4449	0.8750	0.3232	6.6275
−5.7	2.3120	0.4543	0.8488	0.3057	6.9297

self-similar Deflection Model					
Wind Direction [°]	Power $[MW]$	C_p	C_t	a	U_∞ $[m/s]$
0.0	1.0137	0.3892	0.9849	0.4293	5.5431
5.7	2.1437	0.4497	0.8617	0.3141	6.7805
−5.7	2.1437	0.4497	0.8617	0.3141	6.7805

Error [%]					
Wind Direction [°]	Power	C_p	C_t	a	U_∞
0.0	0.0000	0.0000	0.0000	0.0000	0.0000
5.7	8.2294	1.0789	1.5200	2.8156	2.3086
−5.7	7.2794	1.1025	1.5198	2.7814	2.1531

Table 3.8: FLORISSE M Deflection Models Comparison: different wind directions with Zones model.

Regarding the velocity deficit model, the choice was taken based principally in the distribution of the wind farm's power in function of the yaw misalignment of the first generator. Figure 3.11 shows the power distribution generated with all four velocity deficit models. The yaw misalignment step is of 5 degrees and all the simulation data is still the one find in Table 3.2. One general characteristic is that all the models predict a higher power production for positive deflections. From this,

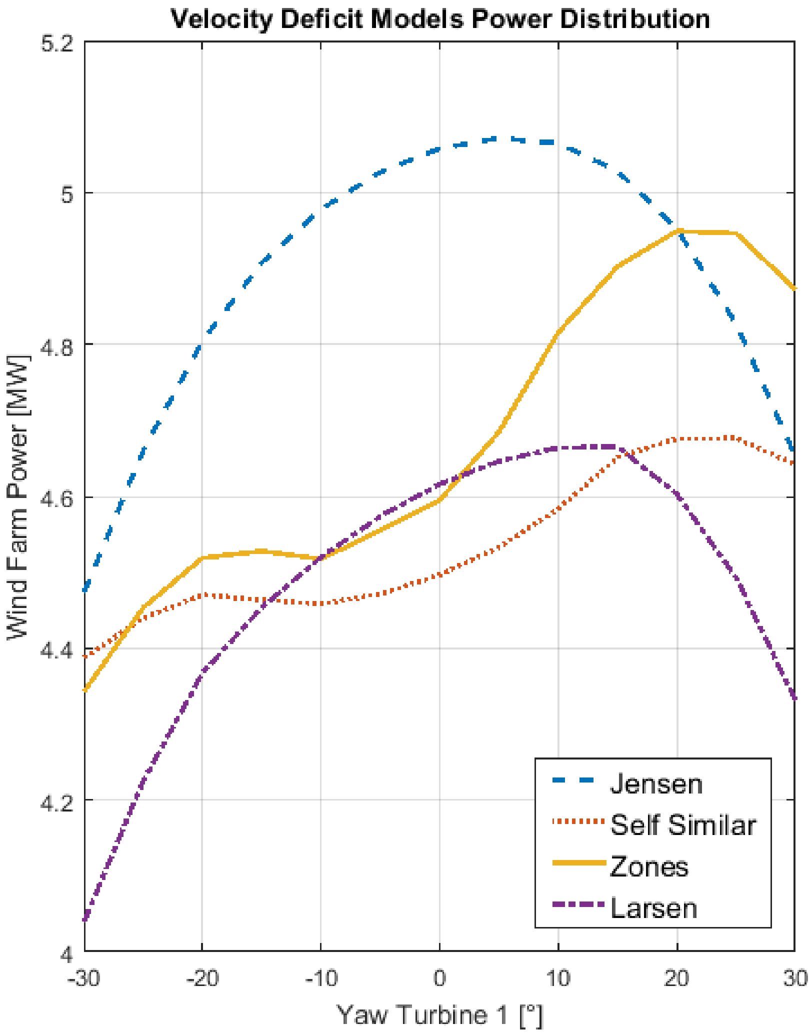


Figure 3.11: Velocity Deficit Models Power Distribution.

it can be known that the initial deflection imposed by the Jimenez model is towards the negative side (imposing the origin point in the very center of the rotor), or in other words, it is like a small positive rotation of the generator. Two separate trends can be seen, the Jensen and Larsen models show a semi parabolic distribution with simple cusp towards the right side, while the Self-Similar and Zones model show two local maxima for both the deflection directions. The expected behavior of such distribution is to have two cusps, one on each side, but one should be higher than the other given the asymmetry caused by the rotation of the turbine. According to this criteria, the Jimenez and Zones are good candidates. Having a comparison with experimental data presented by Schreiber *et al.* in [16], the final choice was to work with the **Zones model.**

Table 3.9 shows the relative error of all the possible combinations of deflection and deficit models with respect to the results of the selected, Jimenez and Zones. The rows of zeros correspond to these models' calculations. There is an exception, this deficit model with the Self-Similar deflection model has the same results. This might be caused by the constants used in the deficit model, with this definition of the zones inside the perturbed flow the asymmetry of the wake is probably not significant at the arrival of the second generator. Let's remember that the power production model uses the average wind speed seen by the generator, so a valid conclusion might be: in that specific zone and in that location, the wake is still symmetric. The Jensen model at 0° of wind direction show the highest error of the whole case, but it shows better concordance than the Self-Similar deficit model with other inflow directions.

Relative Error [%]-Incoming Wind Direction: 0°						
Deflection	Deficit	Power	C_p	C_t	a	U_∞
Jimenez	Jensen	45.6940	10.0206	4.4067	16.0960	9.8158
	self-similar	9.6577	2.7492	1.5332	3.8202	2.4300
	Zones	0.0000	0.0000	0.0000	0.0000	0.0000
	Larsen	2.0913	0.5653	0.3351	0.8852	0.5015
self-similar	Jensen	45.1021	9.9435	6.3357	15.9562	9.6859
	self-similar	10.6047	3.0319	1.5332	3.8202	2.6790
	Zones	0.0000	0.0000	0.0000	0.0000	0.0000
	Larsen	0.9569	0.2569	0.1523	0.4193	0.2309

Relative Error [%]-Incoming Wind Direction: 5.7°						
Deflection	Deficit	Power	C_p	C_t	a	U_∞
Jimenez	Jensen	1.0350	0.1349	0.1943	0.3713	0.3018
	self-similar	13.3791	1.8881	2.6743	5.3837	4.0694
	Zones	0.0000	0.0000	0.0000	0.0000	0.0000
	Larsen	0.7826	0.1124	0.1486	0.2785	0.2278
self-similar	Jensen	4.3469	0.5844	0.8114	1.5161	1.2358
	self-similar	5.1446	0.6968	0.9943	1.9493	1.5164
	Zones	8.2294	1.0789	1.5200	2.8156	2.3086
	Larsen	8.6333	1.1238	1.5886	2.9703	2.4172

Relative Error [%]-Incoming Wind Direction: $-5.7°$						
Deflection	Deficit	Power	C_p	C_t	a	U_∞
Jimenez	Jensen	5.2984	0.7484	1.0957	1.9961	1.5542
	self-similar	11.2716	1.6069	2.3916	4.4175	3.3883
	Zones	0.0000	0.0000	0.0000	0.0000	0.0000
	Larsen	1.4619	0.1981	0.3063	0.5563	0.4199
self-similar	Jensen	10.6055	1.4968	2.2502	4.1558	3.1791
	self-similar	18.7370	2.7515	4.1117	7.8207	5.8112
	Zones	7.2794	1.0125	1.5198	2.7814	2.1531
	Larsen	6.9334	0.9685	1.4491	2.6178	2.0492

Table 3.9: FLORISSE M Models Comparison with respect to Jimenez and Zones results.

Chapter 4

FLORIS - Update: Different Hub Heights and Tilt Angle Inclusion

It is of interest for future developments as well to investigate the effects of non-conventional control variables variation, like the tilt angle, in the production of the wind farm. In this chapter, the version of FLORIS selected was modified in order to enhance the capabilities of the software to simulate cases with additional parameters. With this, the program will be capable to consider additional phenomena that may improve an online control system performance and the initial considerations of a wind farm layout design. The distinction between Python version and MATLAB® is not characterized in this chapter since the focus is exclusively on this last software. A main file was developed, it includes all the case inputs with all their possible options with the objective of increasing the software fluid usability. The specific inputs are let out in order to avoid any possible unintentional modifications, this guarantees that the user that intentionally manipulates them is sure about these changes. Apart from the yaw angle, analyzing the tilt deflection and the effects on the farm if the turbines have different z_0 was also of interest. With this objective, some small changes where done to the core functions to make them suitable for the treatment of the generator's height as other control variables. Table 4.1 shows the ambient conditions used in the analyses shown hereafter. The generator used is the DTU 10MW, its main features are described in Table 3.1, and the wind farm layout is illustrated by Figure 3.2.

Case Data	
Wind Speed U_∞ $[m/s]$	11.4
Air Density ρ $[kg/m^3]$	1.225
Wind Direction [°]	0.000
Turbulence Intensity TI	0.060
Shear Factor	0.000
Deflection Model	Jimenez
Deficit Model	Zones
Combination Model	QAV
Added Turbulence Model	Crespo-Hernandez

Table 4.1: Ambient conditions used for the analysis.

The main contributions to the program were the creation of the parametric class, "yawAndRelPowerSetpoint" control method input analysis and the adaptation of the selected models for a better tilt situation simulation. The improvements here proposed are then compared with wind tunnel data.

4.1 Parametric Class

Following the programming methodology of OOP, a parametric class script is proposed. Although the original version of FLORIS has a dedicated class for visualizations, it does not include the plots of distributions as the program was written for analyzing just one steady case at a time. However, the main file proposed has the capability of running FLORIS more than one time and gathering various results for the construction of the parametric curves. Specifically, the parametric graphs show the effect over the power production of the farm in function of the variation of yaw misalignment γ, tilt angle φ or hub height z_0.

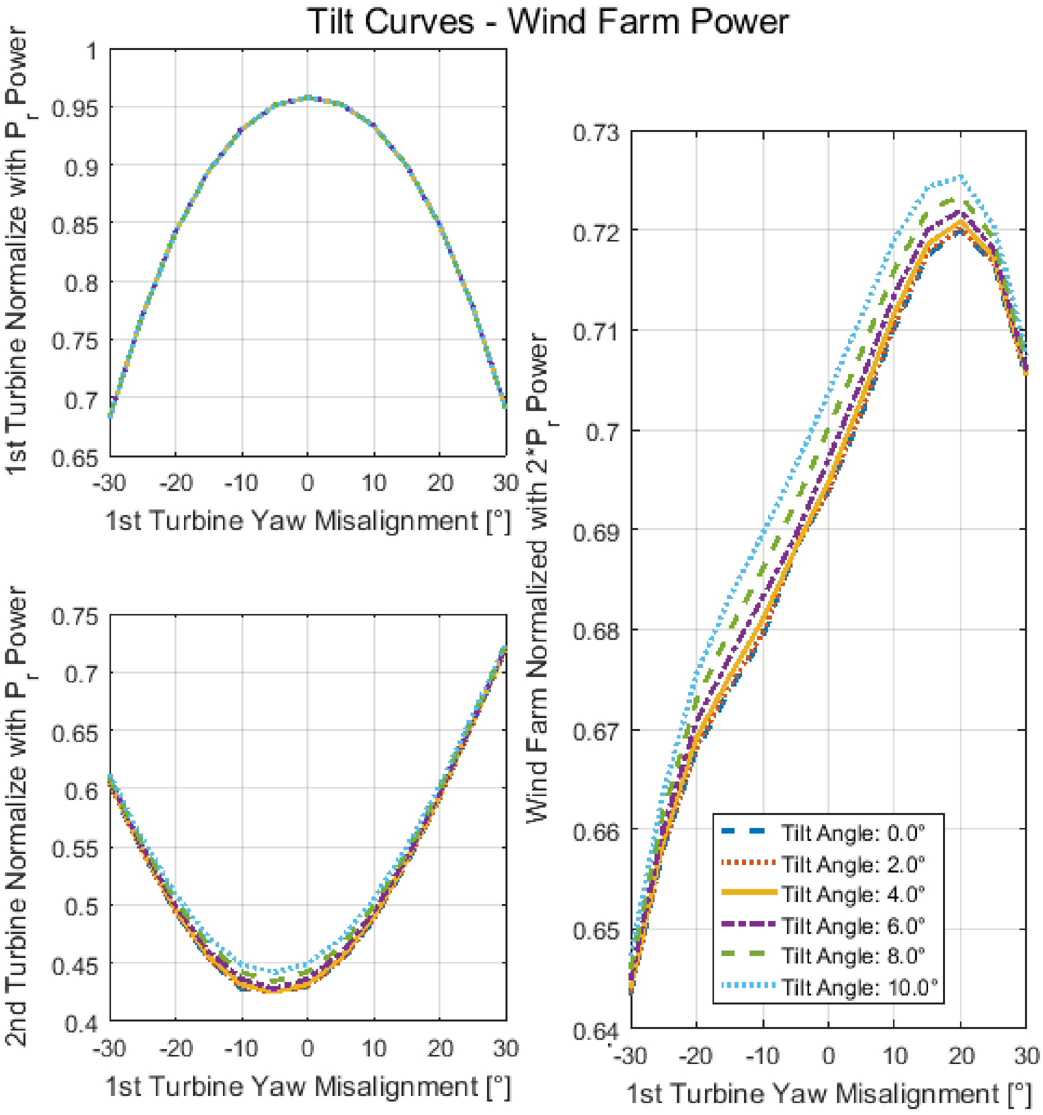


Figure 4.1: Example of Tilt Curves; Individual Turbines and Farm Normalized Power Distribution.

Figure 4.1 illustrates the power distribution of both turbines individually and of the whole farm. The wind farm power is determined by the summation of the individual energy production of the generators. These distributions were generated with the default values of the program. The step of the first turbine's yaw misalignment $\Delta\gamma_1$ is of 5°. The tilt angle is the constant value shown in the legend and it was applied only to the first rotor, meaning that the second turbine has no deflection at all.

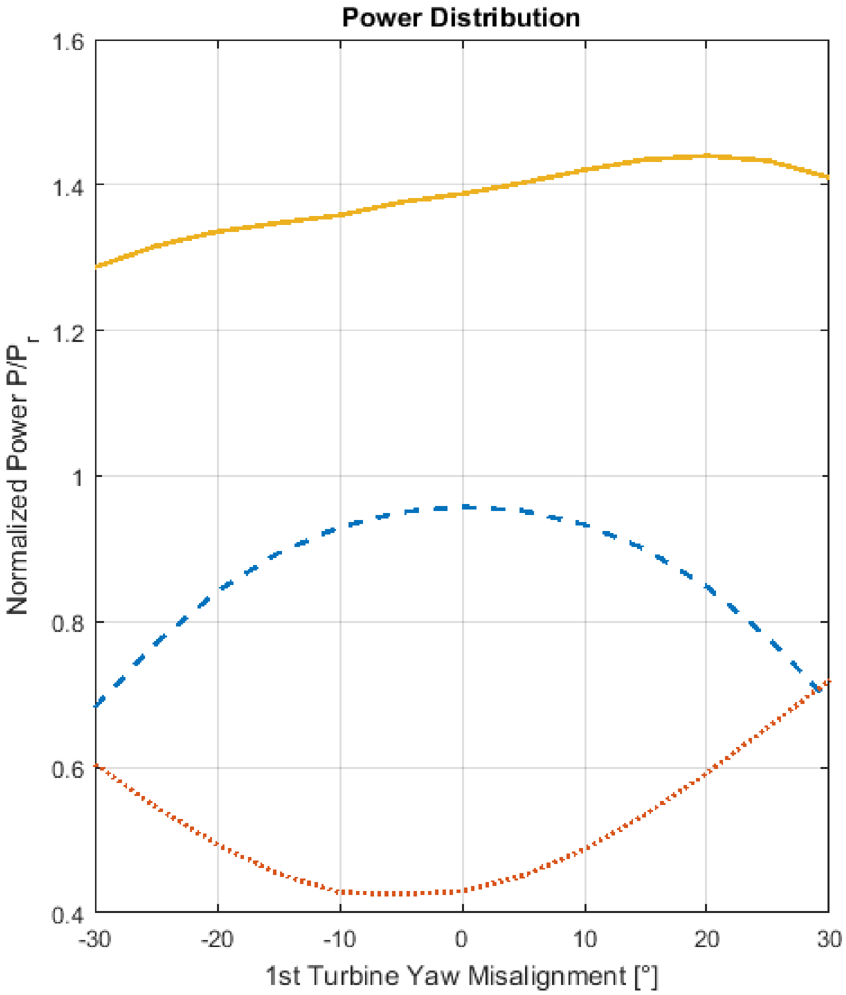


Figure 4.2: Power Distributions. The blue dashed line is the power production of the First Turbine, the red doted line is the power production of the Second Turbine and the yellow continuous line is the power production of the whole Wind Farm. $\gamma_1 = -30 : 5 : 30[°]$, $\gamma_2 = 0$.

Another example of the functionality of the parametric class is Figure 4.2. This graphic is commonly used to make easier the visualization of the individual contribution of the turbine elements to the farm's total production. The power generated P is normalized with the rated power P_r of the turbine type. Although they are not illustrated here, this object is also capable to generate: height curves in function of the variation of the first rotor yaw misalignment, and yaw curves in function of the variation of the tilt angle and the hub height.

4.2 Input Analysis

The main problem with the inputs, using the *"yawAndRelPowerSetpoint"* control method, is their complexity. Configured as the default version, the initial data for the turbine's C_p and C_t are LUT made up by 3D arrays, where the rows are the different wind speed, the columns are the servo variable (is possibly the efficiency) and each matrix is in function of yaw angle. These tables are then used to interpolate, by means of the *griddedInterpolant* function implemented in MATLAB® , a function for both these coefficients in terms of wind speed, servo and yaw angle. The result function is the one used in calculations made by FLORIS. It is important to highlight that the software has the extrapolation disable. As mentioned in the last chapter, there is also a possibility of adjusting the power calculation, originally, with the yaw angle. In this version of FLORIS, the correction is in fact in function of the thrust angle:

$$C_p = C_{p_0} \cos^{pP} \psi \tag{4.1}$$

$$C_t = C_{t_0} \cos^{pT} \psi \tag{4.2}$$

pP and pT values are the default, respectively, 1.88 for this turbine and 2 in general. These equations are used with all the different control methods included, remembering that between turbine types, with the default version of the software, only pP changes.

For the possible improvements, two approaches are presented. The first is to transform the 3D array input into a 1D array input and apply the adjustment. The new inputs vector will be a column vector with the coefficients values for different wind speed values for null yaw angle. In the calculation of the generated power, the coefficients value used will be the results of the adjustment functions, considering both the effects of the yaw and tilt angle. The great advantage of this method is the simplicity introduced in the input processing, but it comes at a price of lower precision. A routine was developed for optimizing the value of the exponents pP and pT. Again, two approaches where used. The first is to optimize each component of the input matrix and then finding the mean of the results. For convenience this method is going to be called *Mean Optimization.* At this point, the mentioned matrix is extracted from the 3D array for a value of $servo = 100$. The optimized value is calculated by using *fmincon* (MATLAB® integrated function) for minimizing the relative error (again, using equation (3.58)) of each of the array components, sum all the results and computing the algebraic mean. With this method, the exponents values found are:

$$pP = 3.1020 \qquad pT = 1.0433 \tag{4.3}$$

The values are considerably different between them, but considering the similarity of the default values, the pT is empirically multiplied by three. In this way, they have almost the same difference than the default values but, somehow, both results of the optimization are used. This way, one obtains:

$$pP = 3.1020 \qquad pT = 3.1299 \tag{4.4}$$

The second one minimizes a single value. Using *fmincon*, the quadratic error computed by transforming the matrix of relative errors into a column vector V_{Err} of

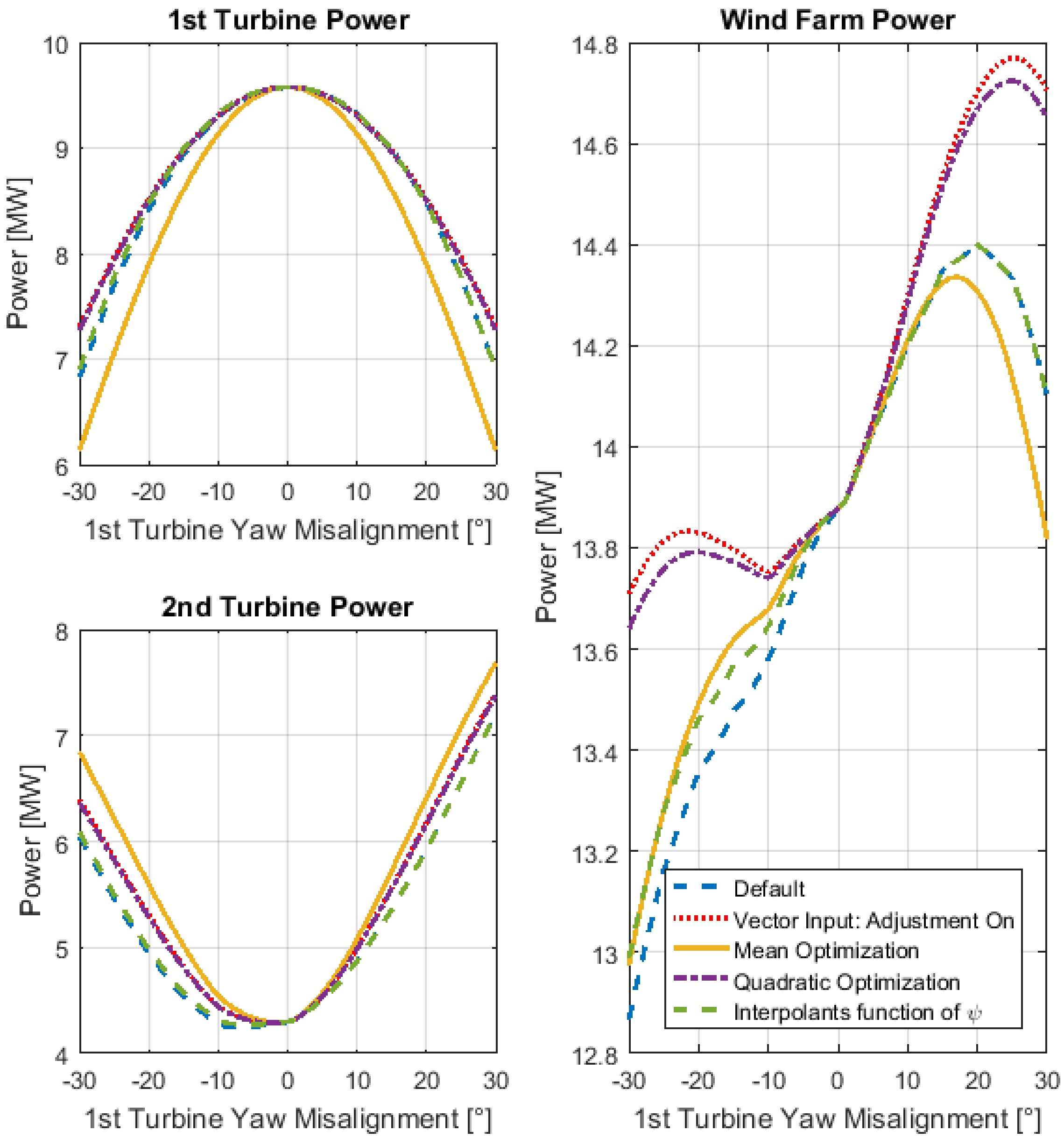


Figure 4.3: Inputs Comparison: Power Distribution in function of the first turbine yaw misalignment γ_1. $\Delta\gamma_1 = 5°$, $\gamma_2 = 0$.

$m * n$ elements, where m is the number of rows and n is the number of columns, just by re positioning each column as rows after the previous one. This vector is pre-multiplied by its transpose and divided by the total amount of elements to obtain a single value:

$$Err_{Quad} = \frac{V_{Err}^T V_{Err}}{m * n} \tag{4.5}$$

Err_{Quad} is the quantity to optimize. For convenience, this is going to be called *Quadratic Optimization*. The results of this method are:

$$pP = 1.9132 \qquad pT = 1.9188 \tag{4.6}$$

which can be notice to be quite similar to the default values, so there is no expectancy of a big difference between the results obtained with one and other.

The second approach is to use the 3D array input but using the interpolant functions in terms of the thrust angle, instead of the yaw misalignment. Applying this change is highly recommended for the use of this particular control method, because in here the corrective cosine functions are not used as the LUT coefficient values have already consider the rotor's deflection, and they would impose a double correction according to the thrust angle. Using the interpolation in function of the yaw misalignment cause a wrong power calculation when the turbine deflection is not in yaw, as the yaw value would remain constant. The next section gives more details about this phenomena. This method advantage is the high precision of the simulation but at cost of a complex input data.

Figure 4.3 show the power distribution in terms of the yaw misalignment variation of the first generator. The values for the individual turbines are represented along with the whole wind farm, with all the possibilities of input's settings mention during this section. As the legend indicates, dashed-blue corresponds to the Default settings, pointed-red is just the C_{p_0} and C_{t_0} vectors with the adjustments corresponding to the "greedy" control method comparing the implementation with the one in other turbines, continuous-yellow is the distribution generated with the values of pP and pT via the Mean Optimization method, dash-pointed-purple is the results calculated with the values of the correcting exponents found with the Quadratic Optimization method, and dashed-green addresses the change of the yaw misalignment for the thrust angle dependency of the interpolation function. There are illustrated two foreseen small variations: the first is between the Default and ψ inside the interpolant functions results. As a matter of fact, the variation in the positive side of the wind farm power production is imperceptible and it is negligible talking about the turbines individually. This also confirms the good accuracy of the applied suggestion. The second one is between the input vector with the cosine adjustment and the results found with the Quadratic Optimization, given the similarity of pP and pT in both computations. Both exponents generation methods show, in general, satisfactory results but each of them is more suitable to represent a different case: the Mean Optimization shows a good recreation of the wind farm production, while the Quadratic Optimization demonstrates an adequate agreement in the individual analysis of each turbine.

Figure 4.4 shows the power distributions of each generator and of the whole farm, as function of the tilt angle of the first turbine. The colors and line type are the same as the ones for Figure 4.3. In contrast, there is just one similarity, between the cosine adjustment results with the values of pP and pT given by the Quadratic Optimization and the original setting. The Mean and Quadratic Optimization methods show accuracy in the same cases than in the yaw misalignment dependent case: the first has better precision at analyzing the whole wind farm but the second is recommended for studying each turbine element. The unusual

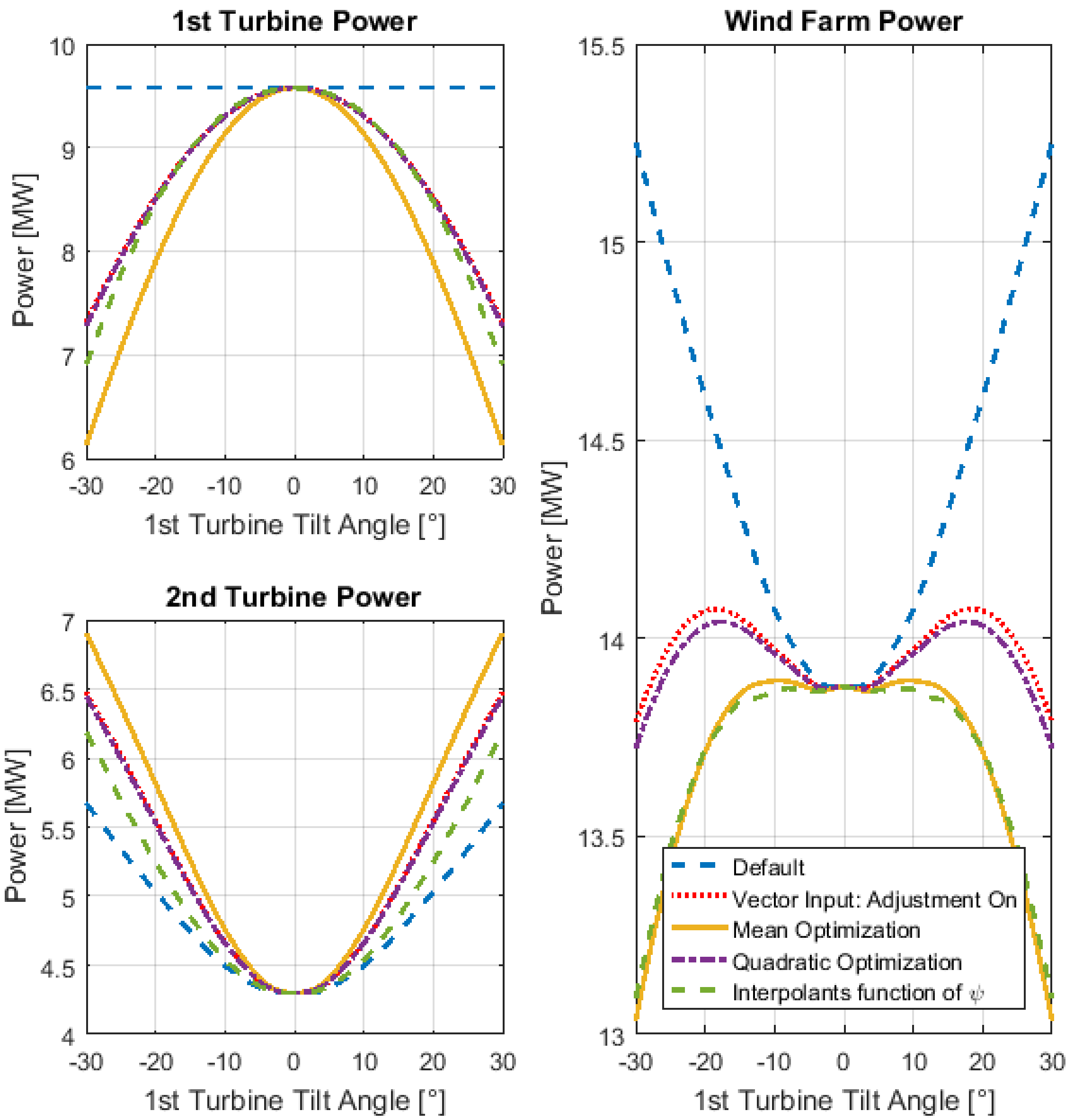


Figure 4.4: Inputs Comparison: Power Distribution in function of the first turbine tilt angle φ_1. $\varphi_1 = -30 : 5 : 30[°]$, $\varphi_2 = 0$.

behavior of the power distributions with the Default configuration is due the fact that the adjustment with the cosine functions is not used. This because it will calculate a wrong result as it would correct again the coefficients values as it has been explained earlier. The problem is that this is the only way in which the power calculation takes into account the tilt deflection because the interpolant functions use the yaw misalignment, which, for this study case, is set to a constant value of 0. The effect of the tilt angle should be equivalent or really similar to the green curve of Figure 4.4, as this one uses the complex array input data for accuracy and does not oversees the tilt angle with the dependency of ψ implemented in the interpolations. Another characteristic to highlight is that, talking specifically of the second turbine, the minimum is not displaced to the left as in the case of the yaw misalignment.

This leads to a symmetric distribution for the farm, as it is not expected for the first turbine to show any asymmetric attitude under the influence of a uniform ambient velocity profile.

4.3 Tilt Consideration

It has been already mention in Chapter 3 that the power generation calculation done by FLORIS used the actuator disk model described by Equation (3.5). Under this theory, the rotor is considered a circle, there is no difference between the horizontal or vertical sections of the turbine. Additionally, the ambient velocity profile is uniform, and the software does not consider the tower or the ground. With this in mind, there is no reason why the power distribution, given by a yaw and a tilt deflection, should be different. Implementing such behavior is very important for a better understanding of combined rotations.

It is easy to realize, looking Figures 4.3 and 4.4, that this is not the case: the computation, with the original state of FLORIS, generates distributions for the two considered deflections that are really different. Figure 4.5 illustrates the comparison between these distributions. The input system used is with the original LUT with the interpolant functions in terms of the thrust angle. With this, there is no need to activate the cosine functions adjustment for the tilt consideration. The first turbine power production is identical, independent of the nature of the rotations. This means that the problem is inside the wake modeling, as it demonstrates that the power calculation has already the expected behavior. This last is reinforced with the concordance in all three graphics on the point where there is no deflection. In section 3.4.1 it has been already explained how the Jimenez model manages to recreate the asymmetry of the wake due the generator's sense of rotation. A correction to this model is proposed by using a 2D rotational matrix R_zR_y (Equation (3.60)) instead of the implemented 1D rotational matrix R_z (Equation (3.57)). The angles used would be the pre-imposed deflections: γ would be horizontal and φ would be vertical. A value of 1.5° was used for both of them. The red curves in Figure 4.6 illustrate the effect of this change. The tilt variation power distribution is no longer symmetric, as a matter of fact, the trend is really similar to the pure yaw case. For the negative side, the curves almost overlap, however, the positive side results are still considerably different.

Looking at the equations of each submodel, there is no dependency of any turbine rotation inside the Wake Combination and Added Turbulence models, just in the Velocity Deficit one. Equation (3.36) represents the wake decay coefficient for the various zones in the chosen deficit model, in which there is a function of the yaw misalignment in the denominator. This means that in any case where γ remains constant, the contribution of the tilt deflection to this coefficient is going to be neglect. This can be solved by replacing the yaw misalignment with the thrust angle inside that function, resulting in:

$$m_{U,q} = \frac{M_{U,q}}{\cos a_U + b_U \psi} \tag{4.7}$$

The effects of this change are represented by the blue curves in Figure 4.6. The first characteristic that is noticed is that only one curve can be clearly distinguished.

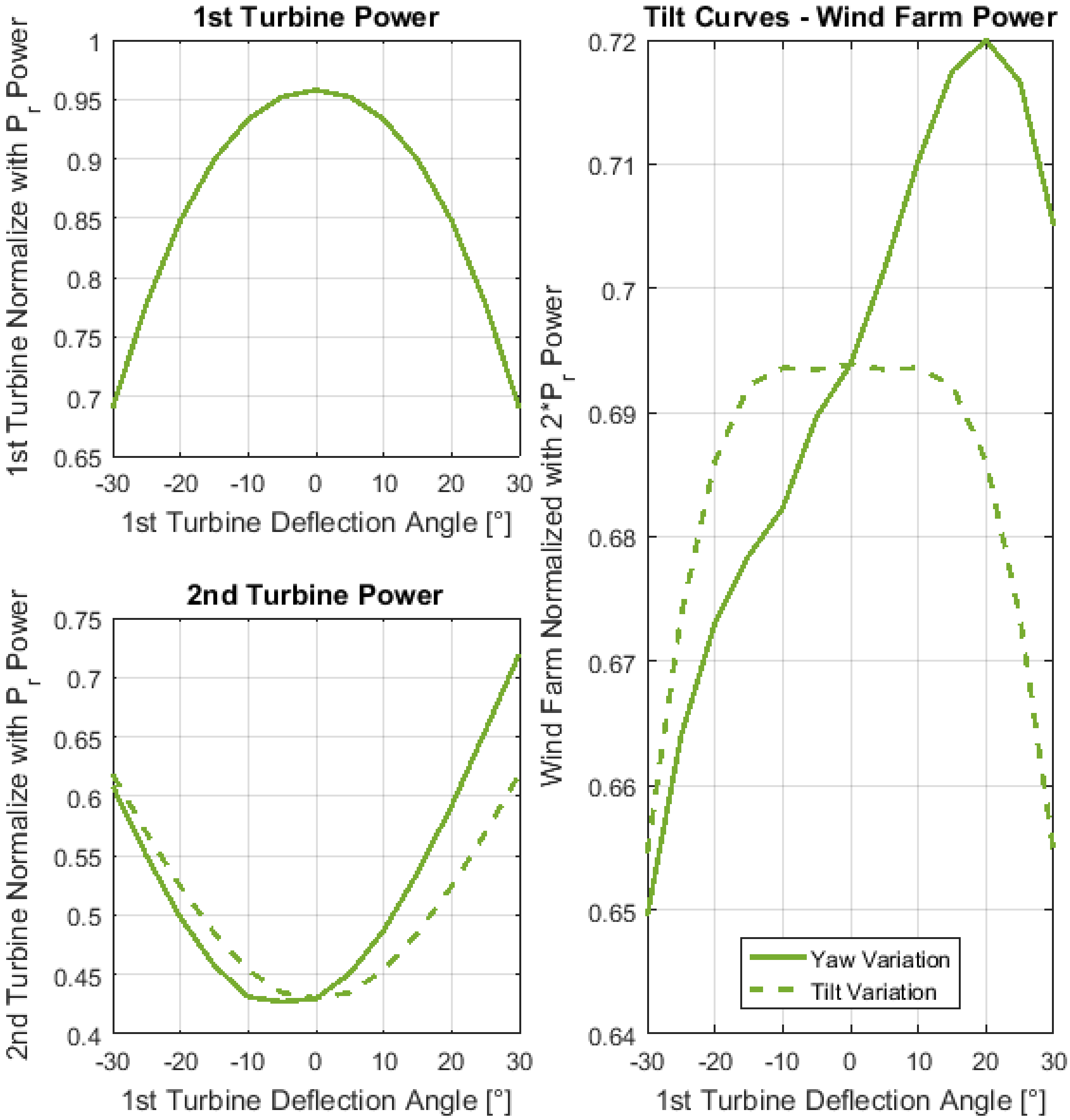


Figure 4.5: Yaw and Tilt Deflection Power Distributions Comparison.$\gamma_1 = -30 : 5 : 30[°]$, $\gamma_2 = 0$, $\varphi_1 = -30 : 5 : 30[°]$, $\varphi_2 = 0$.

This because the two distributions, in function of the yaw misalignment and of the tilt angle variation, are in fact superimposed. The first turbine power generation does not change as expected, the only generators affected by such alterations should be the ones working under a partial or complete wake. For clarification, the right side suffers no change by adding the deficit model correction to the Jimenez adjustment, so the curves are equivalent and that is why this portion of the red continuous curve is not seen. With the models' adjustments, the condition of equivalent effects given a horizontal or a vertical rotation of the rotor has been accomplished.

4.3.1 Quaternion Aided Sign Determination

Confronting the curves resulting of the given modifications with the ones generated with the original equations, it can be noticed that the behavior of such corrections depends on the sign of the deflection. The Jimenez correction causes a small vertical

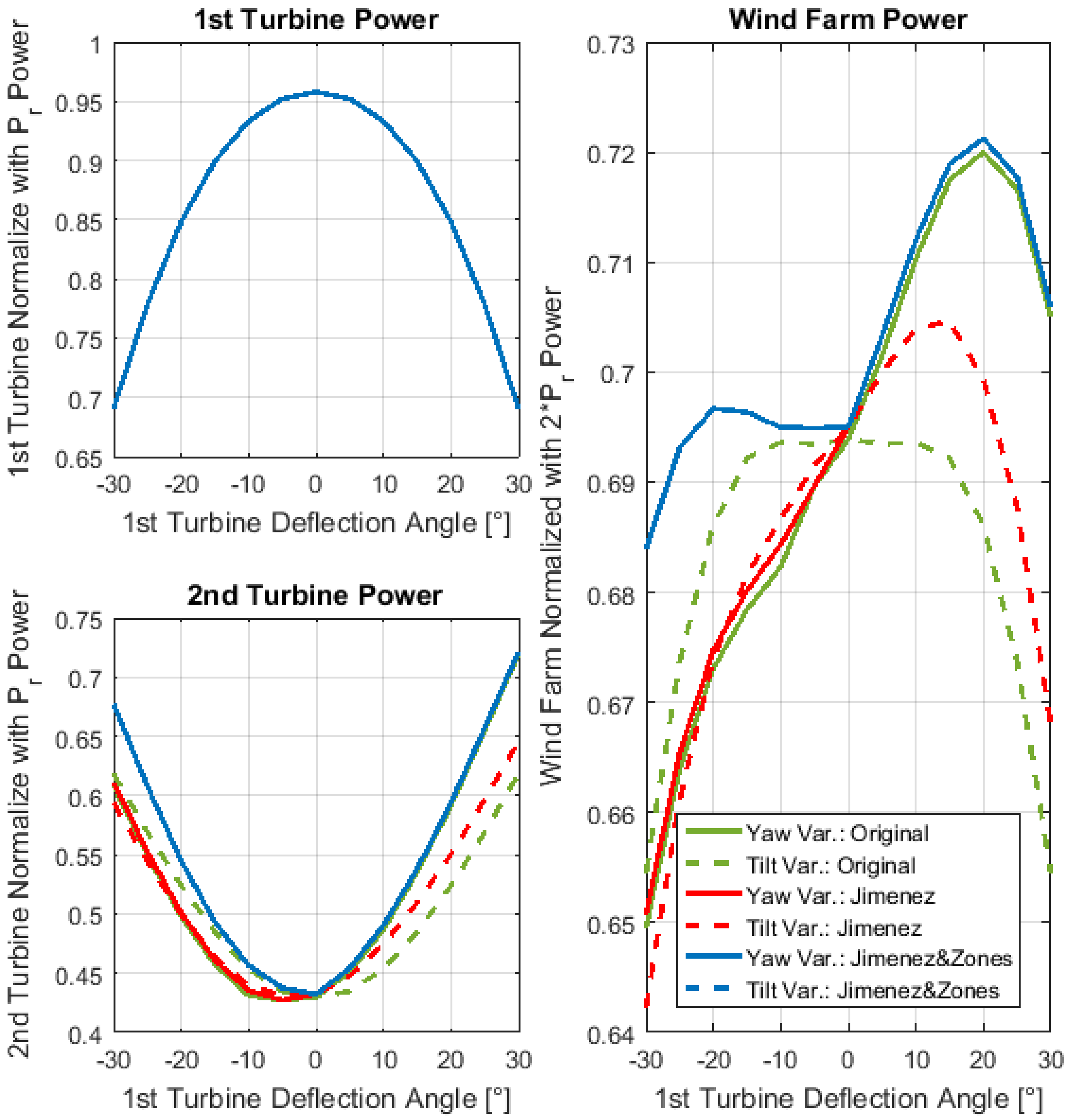


Figure 4.6: Yaw and Tilt Deflection Power Distributions Comparison: With Models' Modification. $\gamma_1 = -30 : 5 : 30[°]$, $\gamma_2 = 0$, $\varphi_1 = -30 : 5 : 30[°]$, $\varphi_2 = 0$.

wake deflection that results in a greater mean wind speed seen by the second turbine. This is reflected by the modest upward translation seen in the positive half of the distribution. Nevertheless, the trend is conserved. The behavior of the negative side is completely different, which insinuates that there may be an imperfection in the rotor's deflection sign.

Let's recall the procedure described in section 3.4.1 to determine the thrust angle ψ. This method uses the Euler angles inside a rotational matrix in order to find the direction cosines vector. Then, the first component of the array is equaled to the cosine of the thrust angle, and finally this expression is solved for ψ. The problem lays in the fact that the cosine is an even function. The principal property of such functions is that $f(x) = f(-x)$. This means that the already implemented reasoning for the thrust angle computation is in fact incapable of differentiating among a negative or a positive deflection.

Using the Euler angles is not the only method that exist for dealing with 3D rotations. In other areas like automation and game development the rotations via quaternions multiplication are commonly used [26, 27]. A quaternion is an extension of complex numbers, that has one real part and three imaginary parts. They are normally expressed as:

$$q = a + bi + cj + dk \tag{4.8}$$

where a, b, c and d are real numbers, and i, j and k are the imaginary components of the respective axis. This kind of numbers might be seen as a vector defined by the imaginary parts, that suffers a rotation defined by the real part. With this in mind, equation (4.8) can be written in terms of a rotation angle ψ [28]:

$$q = \cos\psi + \sin\psi(xi + yj + zk) \tag{4.9}$$

where x, y and z are components of a unitary vector.

Determining an angle with the arccosine function will always result in a positive deflection, in equation (4.9) this means that both cosine and sine functions are given for a positive angle. Any rotation can be positive around a specific axis, which is in matter of fact, what a quaternion represents. With this in mind, the problem would not be the sign of the deflection, actually, it would be the direction of the resulting axis of rotation w. The thrust direction E, given by equation (3.61), can be interpreted as the result of rotating the streamwise unit vector x_{st} by ψ degrees, around an axis w that is perpendicular to both E and x axis. Thus, w can be determined by the computation of the cross product of E and x_{st}. Due the pre-established reference frame inside FLORIS, if it is imposed that always the z component of E must be positive, the sign of the rotation can be determined. This has been imposed for pure yaw and combined rotations. In the case of pure tilt rotations, the imposed condition is that the y is always positive. This will be called *the reference axis sign condition.* If the unitary vector of E is introduced into an expression of the type of equation (4.9), and it does not fulfill the reference axis sign condition, it means that all the imaginary components must suffer a change of sign, which can only be achieved if $\sin\psi$ is negative, thus, ψ must be negative. The magnitude of the thrust angle is determined by equation (3.63).

Figure 4.7 show the effect of introducing the sign correction on the thrust angle. The green curves have the original models (where just Jimenez is in function of ψ), the red distributions count with the Jimenez modification for inducing a vertical asymmetric wake, and for the blue curves is already implemented the Zones model adjustment. The continuous lines are the power production in function of the first turbine yaw misalignment and the dashed lines are the power generation in terms of the first rotor tilt angle. The red continuous curve is overlapped by the blue distribution. The first characteristic to highlight is that the Zones model correction does not longer affect the yaw dependent distribution. In Figure 4.6 can be seen how the positive deflection side does not change, but in the negative side of the x axis some difference can be highlighted. The fact that the power generation does not longer change by considering the sign of ψ, reflects that this modification is

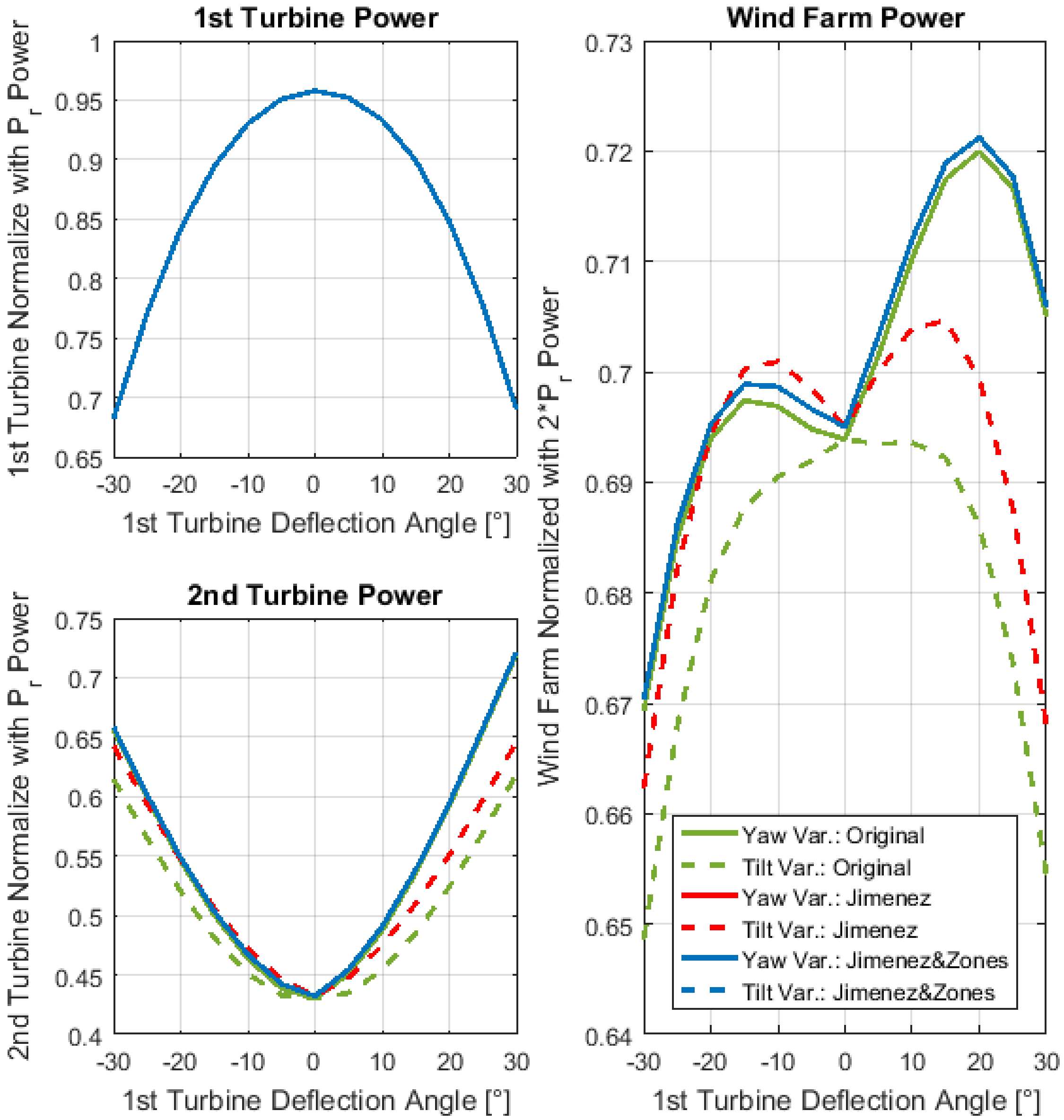


Figure 4.7: Yaw and Tilt Deflection Power Distributions Comparison: With Sign Correction. $\gamma_1 = -30:5:30[°]$, $\gamma_2 = 0$, $\varphi_1 = -30:5:30[°]$, $\varphi_2 = 0$.

well implemented. This because its objective was to adjust the dependency to the tilt angle, not to modify the power generation for a pure yaw rotation. It has been previously mention that the expected behavior of the power generation distribution of the whole farm should consist of two local maxima, one for positive rotations and the other for negative rotations, but looking to the curves presented in Figure 4.5, this was not the case. There is no reason for which the total wind farm energy generation should be bigger when the second turbine is under a complete wake, rather than when it is under a partial wake for all the negative deflections considered. With the sign correction, this is no longer the case. Although for considerable first turbine negative rotations the power of the farm decreases significantly, the peaks are present in both sides of the curves. Thus, there is at least one positive deflection and one negative deflection, for which the global energy production is higher. The expected trend, and the equal effect of yaw and tilt deflection have been achieved.

4.4 Experimental Comparison

Until now, the modifications have been done looking to improve the qualitative aspect of the software. The objective of this section is to realize a quantitative confrontation with experimental data that would demonstrate that those adjustments are correct and fulfill their goal.

The wind tunnel data used was obtained from the experiments executed for the Deliverable 3.1 CL-WINDCON [29]. These consist of three G1-TUM wind turbines with variable yaw of the first two elements and additionally, wind direction variation, simulated by means of a turning floor table. The main characteristics of this generator are shown in Table 4.2[1]. These and additional data may be found in [29]. The layout is a 3x1 farm, with $5D$ between the turbines and an offset of $1.5D$ from the center of the table towards the left with respect to the incoming wind direction. The presence of a shear factor greater than zero means that the velocity profile is logarithmic.

G1-TUM	
Rotor Diameter D [m]	1.1000
Hub Height z_h [m]	0.8250
Rated Wind Speed U_{rated} [m/s]	6.0000
Rotor Efficiency e	1.0000
pT-Low TI	2.1741
pP-Low TI	1.4248
pT-High TI	2.1790
pP-High TI	1.4317

Table 4.2: Principal Characteristics of the G1-TUM wind turbine.

The data comparison have been done for all the cases of *Test 1: Validation of wind farm open-loop control algorithms, low TI"* and for all the cases of *Test 4: Validation of wind farm open-loop control algorithms, high TI.* However not all results are going to be shown. These experiments have been made for various wind direction, for each wind direction there are five yaw misalignment of the first turbine γ_1 and five yaw misalignment of the second turbine γ_2. The ambient conditions used are shown in Table 4.3.

Within the various measures available in the experimental data set, the ones that were use are: the torque Q, the rotor speed Ω, the Relative Effective Wind Speed-REWS U_{eff}, the yaw misalignment γ and the air density ρ. All these variables are given for each turbine, and their final values are determined with the arithmetic mean of all the given measures after filtering the initial spurious data. The experimental power calculation for each rotor has been done with two methods. First it was computed with the actuator disk model (equation (3.5)) for generating

[1] pT and pP are found on the experimental data set.

Case Data	
Wind Speed U_∞ $[m/s]$	≈ 5.70
Air Density ρ $[kg/m^3]$	≈ 1.16
Low Turbulence Intensity TI	≈ 0.05
High Turbulence Intensity TI	≈ 0.10
Shear Factor	0.20

Table 4.3: Ambient conditions used in the experiments and simulations.

a comparison of energy production under the same model. The power coefficient C_p necessary for the calculation has been interpolated from a table of this variable, generated experimentally, in function of the wind speed for a constant air density. The second method is given by the equation:

$$P = \Omega Q \tag{4.10}$$

which, taking advantage that both experimental measures are available, shows the actual generated power. The REWS have been corrected with the cosine of the yaw misalignment of the turbine:

$$U_{R,i} = \frac{U_{REWS,i}}{\cos\gamma_i} \tag{4.11}$$

where the subscript i represents the turbine element, $U_{REWS,i}$ is the REWS of turbine i and $U_{R,i}$ is the wind speed seen by rotor i.

For the FLORIS simulation, the ambient case inputs used are the ones shown in Table 4.3. The wind speed varies for each test and its value is equal to the corrected REWS of the first turbine. For each wind direction, the second yaw misalignment is constant, so the power distribution is in function of γ_1. All the analyzed tests have $\gamma_3 = 0$. The specific inputs have not been changed. For clarification, as all these scenarios consist in pure yaw rotations, $\psi_i = \gamma_i$.

In overall, the tests that have just positive γ_1 values show almost no change. This was expected, the most significant effect on the FLORIS generated distributions is given by the consideration of the sign of ψ, so in the scenarios where this is not going to be applied, the variation is exclusively due the modeling of a vertically asymmetric wake within the Jimenez deflection model adjustment. The cases with the highest wind direction θ_W angle show the smallest variation with the corrections implemented to FLORIS. High floor table rotation caused a reduced interaction of the downwind turbines with the upstream wakes. The trend of the distributions does not suffer a remarkable modification with the proposed adjustments, and what is noticeable is that the effect seems to be a translation towards the experimental results, leading to an improvement in the quantitative comparison of the two data sets. This is probably because the majority of the evaluated values of γ_1 are distant from the zero-deflection point, the zone where the greatest shape variation occurs, as it has been illustrated by the figures of the Section 4.3. Some of the high TI cases show that for a particular turbine, especially turbine two, the original version of the software actually calculates better results, however, for the whole farm distribution, the adjusted FLORIS improves the agreement with wind tunnel data. The third turbine is the most problematic, the error of this generator is always the greatest and also is the element for which the trend of the software generated curves and the experimental curves differs the most. This is also an expected phenomenon because of the interaction of multiple wakes. It is the region with the most complex flow structure in the field.

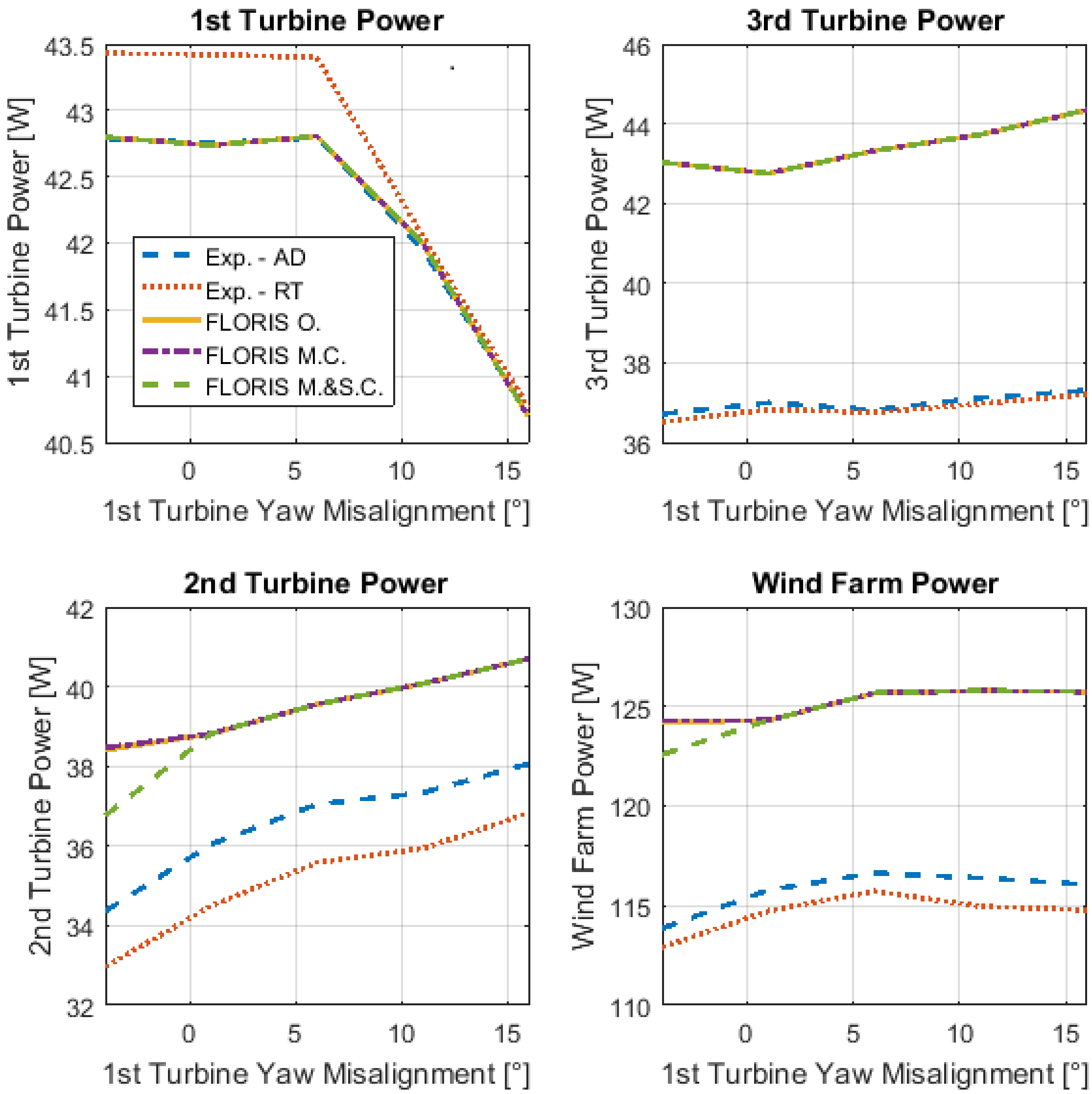


Figure 4.8: Power Distributions of all the components of the wind farm. Test 1.1: $\theta_W = -11.54°$, $\gamma_1 = -4 : 5 : 16[°]$, $\gamma_2 = 16°$, $TI = 5\%$.

Figure 4.8 show the power curves, in terms of the yaw misalignment of the first turbine, for one case of Test 1.1. As in the legend, the blue dashed line correspond to the experimental power calculation with the actuator disk model equation, the red pointed curve is again the experimental power but calculated with the rotor speed and the torque of the given turbine, the yellow continuous distribution was generated with the original version of FLORIS, the purple dash-pointed line was found adding to the software the modification of both models and the green dashed curve was computed by including the correction of the thrust angle sign inside FLORIS as well. In all the cases, the wind farm generated power calculation was computed as the sum of the production of all the turbines.

Relative Error [%]- AD vs FLORIS Original				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-4.0000	0.0172	11.4776	17.1911	9.0002
1.0000	0.0172	7.4133	15.5541	7.2710
6.0000	0.0172	6.5722	17.6768	7.6591
11.0000	0.0172	7.1301	17.7912	7.9540
16.0000	0.0172	6.7174	18.8727	8.2624

Relative Error [%]- AD vs FLORIS Models Correction				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-4.0000	0.0172	11.6610	17.1911	9.0556
1.0000	0.0172	7.4235	15.5541	7.2742
6.0000	0.0172	6.5793	17.6768	7.6613
11.0000	0.0172	7.1341	17.7912	7.9553
16.0000	0.0172	6.7174	18.8727	8.2624

Relative Error [%]- AD vs FLORIS Models and Sign Correction				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-4.0000	0.0172	6.6482	17.1911	7.5427
1.0000	0.0172	7.4235	15.5541	7.2742
6.0000	0.0172	6.5793	17.6768	7.6613
11.0000	0.0172	7.1341	17.7912	7.9553
16.0000	0.0172	6.7174	18.8727	8.2624

Relative Error [%]- RT vs FLORIS Original				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-4.0000	1.4918	16.2107	17.8195	9.9222
1.0000	1.5625	12.2473	16.0908	8.2549
6.0000	1.4081	10.9160	17.8579	8.4994
11.0000	0.2617	11.2582	18.2646	9.2975
16.0000	0.1621	10.3160	19.1801	9.4683

Relative Error [%]- RT vs FLORIS Models Correction				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-4.0000	1.4918	16.4019	17.8195	9.9780
1.0000	1.5625	12.2580	16.0908	8.2581
6.0000	1.4081	10.9234	17.8579	8.5016
11.0000	0.2617	11.2624	18.2646	9.2988
16.0000	0.1621	10.3160	19.1801	9.4683

Relative Error [%]- RT vs FLORIS Models and Sign Correction				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-4.0000	1.4918	11.1762	17.8195	8.4523
1.0000	1.5625	12.2580	16.0908	8.2581
6.0000	1.4081	10.9234	17.8579	8.5016
11.0000	0.2617	11.2624	18.2646	9.2988
16.0000	0.1621	10.3160	19.1801	9.4683

Table 4.4: Relative Error Tables. Test 1.1: $\theta_W = -11.54°$, $\gamma_1 = -4 : 5 : 16[°]$, $\gamma_2 = 16°$, $TI = 5\%$.

This particular test is quite interesting because it includes the majority of the characteristics described above. With the first turbine power production curves, two things can be highlighted: the actuator disk model, with the cosine correction for yawed rotors, can reproduce satisfactorily the actual energy production of a wind farm, and, as has been mention before, this evaluation via FLORIS remains constant regardless of the changes of the models. The high wind direction tests are the only ones who have at least one negative value of γ_1. This particular point shows a considerable improvement for the distribution generated with the models and ψ sign correction, in quantitative and qualitative terms. Table 4.4 show the relative errors, calculated with the experimental data as d_t and the FLORIS data as d_e using equation (3.58), of all the three versions of FLORIS with respect to both experimental power computation results for the Test 1.1. There are two aspects to remark: first, the models' correction actually deteriorates the result. Nevertheless, the error is just augmented by a order of magnitude of approximately 10^{-2}, and additionally, the negative deflection point, with also the sign correction, improves its accuracy by approximately 5% for the second turbine and 1.5% for the whole farm results. This error increase can be considered acceptable and negligible since it leads to more significant improvements in the results. The second aspect will be that it demonstrates once again that the ψ sign correction is adequate, it does not induce any change in the positive deflection power calculations.

Figure 4.9 and Table 4.5 illustrate the results for one of the scenarios of Test 4.8. This is one of the cases that show the greatest improvements of the calculations of FLORIS with the corrections proposed. As anticipated, the resulting curve trend does not suffer great variations: the green dashed and purple point-dashed curves look like translations of the yellow continuous distribution. What happens in turbine two is noticeable: the correction of the models alone has a better agreement with the RT experimental data in the point of the highest yaw misalignment, but in the same point, the computation with both corrections has the better concordance with the AD results. However, for all the values of γ_1, the green distribution is the closest one to both wind tunnel data curves. This is also one of the cases where the FLORIS calculations follow a trend satisfactorily similar to the experimental ones for turbine number three. Speaking about the relative error between the results of the software original version and of the updated FLORIS, there is a reduction of almost 16% for the second turbine at $\gamma_1 = -6$. The biggest error reduction for the farm's production is of approximately 8% and the smallest is of around 7.5%. Also, there is one point, on the distributions of the second turbine, where the error is almost 0%. Is also worth to mention that, in comparison with the low turbulence test, the higher TI is reflected in more notorious differences of the behavior of the FLORIS generated distributions in contrast to the experimental ones. Specifically in the first turbine power distribution, the trend of all the curves that use the actuator disk model differs the most compared with the RT experimental results. This can be attribute to the higher velocity fluctuations present in the measured flow given the greater turbulence.

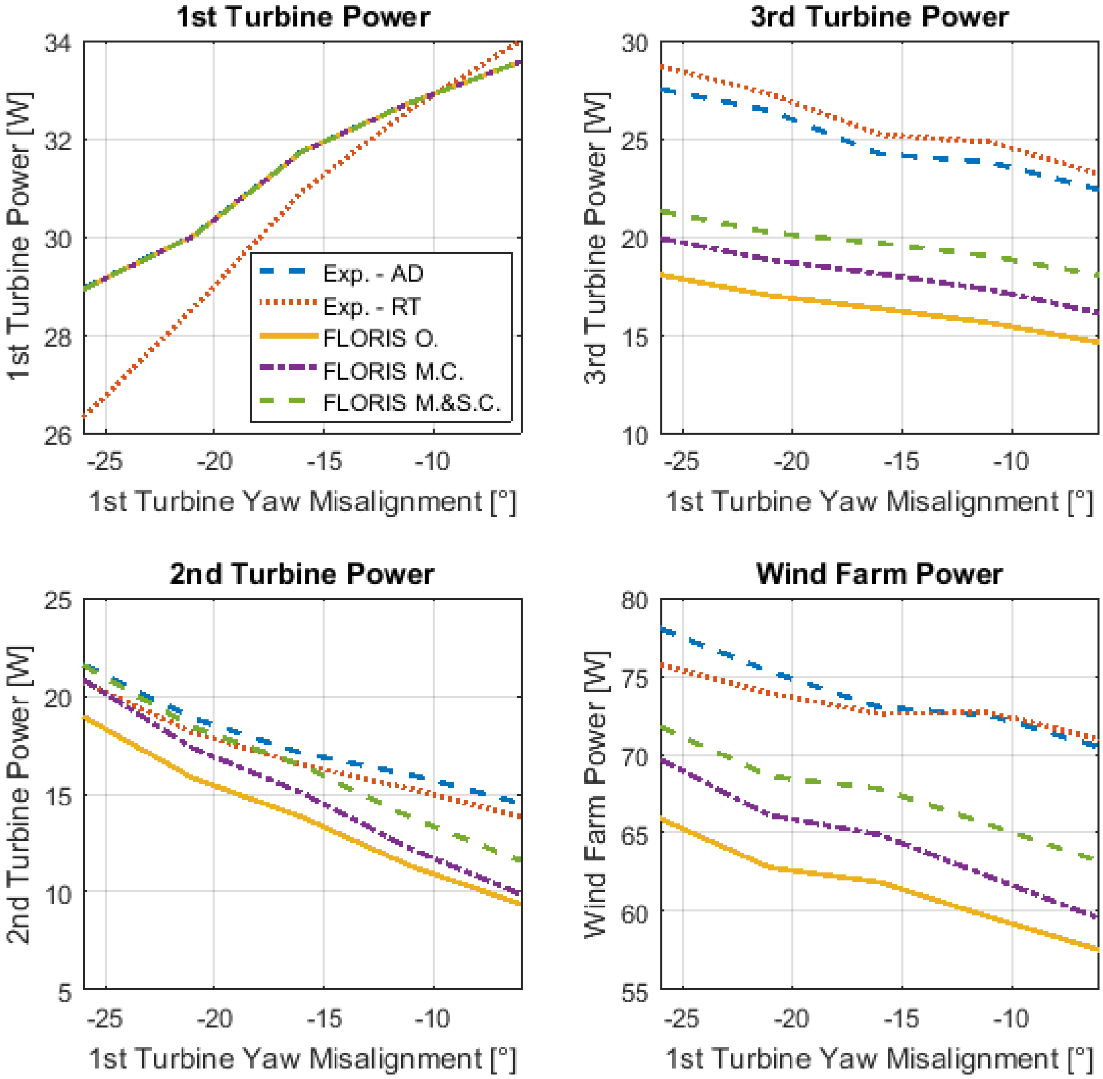


Figure 4.9: Power Distributions of all the components of the wind farm. Test 4.8: $\theta_W = 2.29°$, $\gamma_1 = -26 : 5 : -6[°]$, $\gamma_2 = -20°$, $TI = 10\%$.

Relative Error [%]- AD vs FLORIS Original				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-26.0000	0.0513	12.1969	34.3961	15.5198
-21.0000	0.0328	16.1811	35.6787	16.5835
-16.0000	0.0199	19.0639	32.5727	15.2653
-11.0000	0.0093	29.1986	34.4014	17.7243
-6.0000	0.0027	35.3834	34.7465	18.3308

Relative Error [%]- AD vs FLORIS Models Correction				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-26.0000	0.0513	3.4633	27.6265	10.7183
-21.0000	0.0328	8.1140	28.7858	12.1443
-16.0000	0.0199	11.7767	25.2309	11.1275
-11.0000	0.0093	23.7280	27.2739	14.1804
-6.0000	0.0027	31.9024	28.1231	15.5092

Relative Error [%]- AD vs FLORIS Models and Sign Correction				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-26.0000	0.0513	0.1319	22.5539	8.0086
-21.0000	0.0328	2.3734	23.5149	8.8569
-16.0000	0.0199	3.6658	18.8196	7.1056
-11.0000	0.0093	13.7152	20.1221	9.6284
-6.0000	0.0027	20.1741	19.5379	10.3652

Relative Error [%]- RT vs FLORIS Original				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-26.0000	9.9294	8.5765	37.0640	12.9361
-21.0000	5.2137	12.7940	37.7159	15.0428
-16.0000	2.6246	16.2074	35.2209	14.7868
-11.0000	0.3953	25.8640	37.1084	17.9214
-6.0000	1.2403	32.2373	36.9367	18.9337

Relative Error [%]- RT vs FLORIS Models Correction				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-26.0000	9.9294	0.5172	30.5697	7.9877
-21.0000	5.2137	4.4009	31.0414	10.5215
-16.0000	2.6246	8.6630	28.1675	10.6257
-11.0000	0.3953	20.1357	30.2750	14.3860
-6.0000	1.2403	28.5868	30.5356	16.1329

Relative Error [%]- RT vs FLORIS Models and Sign Correction				
1st Turbine Yaw Misalignment γ_1 [°]	P_1	P_2	P_3	P_{WF}
-26.0000	9.9294	3.9859	25.7034	5.1951
-21.0000	5.2137	1.5717	25.9374	7.1734
-16.0000	2.6246	0.2658	22.0079	6.5811
-11.0000	0.3953	9.6513	23.4184	9.8449
-6.0000	1.2403	16.2875	22.2385	11.0269

Table 4.5: Relative Error Tables. Test 4.8: $\theta_W = 2.29°$, $\gamma_1 = -26 : 5 : -6[°]$, $\gamma_2 = -20°$, $TI = 10\%$.

Chapter 5

FLORIS - Optimization

FLORIS counts with a couple of optimization routines as well. One of them is developed to optimize the wind farm power production by variations of some control variables. The second module function is to optimize the parameters of the different submodels for a better agreement with a given experimental data. However, only the first optimizer was studied in this work.

5.1 Power Production Optimizer

The power production optimizer uses the MATLAB® integrated function *fmincon*. This is a gradient-based method designed for continuous objective and constraint functions with continuous first derivative. It finds the minimum of constrained multivariable functions [30]. In this application, the power production of the whole wind farm is the cost function, while the control variables are the ones to optimize. The only constraints imposed for these computations are the lower and upper boundary of the optimization variables. This module gives the possibility of optimizing three different control variables: the yaw misalignment, pitch angle and the axial induction factor. The yaw feature is available for all the control methods inside the program, however, the other two options are enabled exclusively when the specific control methods, "pitch" and "axialInduction" respectively, are used. Is also possible to optimize the yaw misalignment and one of the other variables simultaneously.

In this routine there is a possibility of including uncertainty of the ambient wind direction using a Gaussian probability distribution. For doing this, the program asks for the wind direction standard deviation and discretization bins. It is also mentioned that is recommended five or more bins. The probability distribution is given by the equation:

$$\theta_{W,probability} = \left(\frac{1}{\theta_{W,std}}\right)\left(\frac{1}{\sqrt{2\pi}}\right) e^{\frac{\theta_{W,range}}{2(\theta^2_{Wstd})}} \tag{5.1}$$

where $\theta_{W,std}$ is the standard deviation and $\theta_{W,range}$ are the discretization bins. This expression is then normalized with the sum of the bins. The discretization is done within a linear space. The final wind farm power is then calculated by sum of the product between the elements of $\theta_{W,probability}$ and the sum of the power production of the individual turbines. In the program this sum is called *deterministic cost* *function*. If just one bin is chosen, the normalized wind direction probability is equal to one, thus the farm's power computation is the deterministic one.

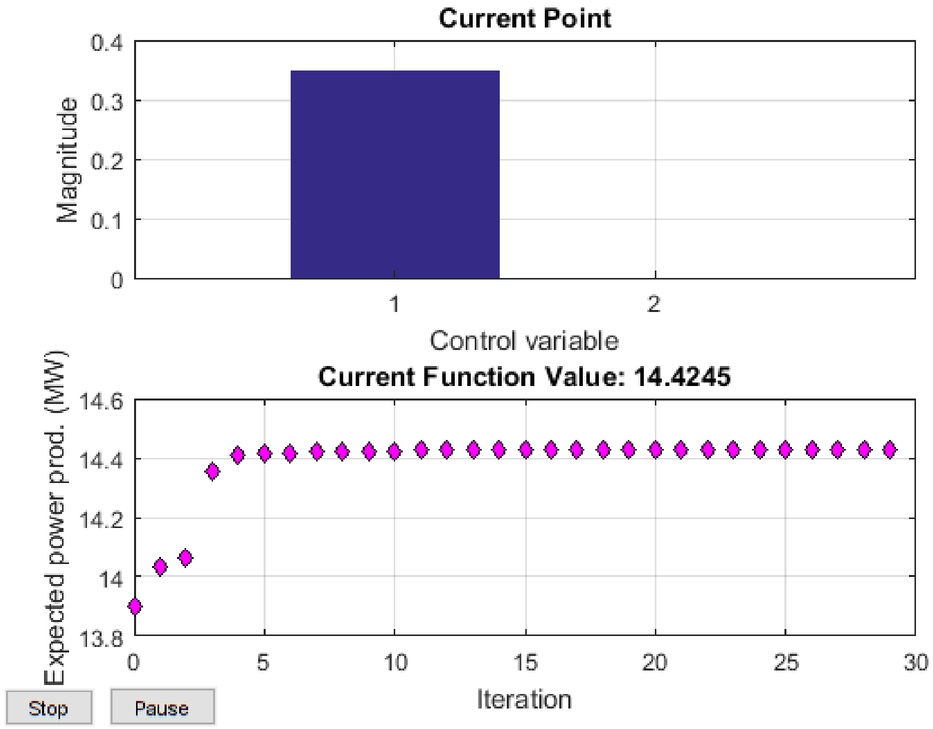


Figure 5.1: Example of the dynamic graphic of the optimization process. The optimization variables are γ_1 and γ_2.

Figure 5.1 shows the final result of a yaw misalignment optimization. Both graphs are updated for each iteration. The bar graphic shows the variation of the optimization variables while each point of the bottom graphic is the result of the cost function in every iteration. The layout used is the 2x1 farm illustrated by Figure 3.2 and the turbine type is DTU 10MW. The control method used was "yawAndRelPowerSetPoint", so neither pitch angle or axial induction factor optimization were analysed, and just one discretization bin was used. The lower and upper boundaries set for γ are $b_l = -30°$ and $b_u = 30°$ respectively. The FLORIS version used has already implemented the adjustments described in Chapter 4. Table 5.1[1] shows

Yaw Misalignment γ Optimization			
Variable	Baseline	Optimized	Relative Variation [%]
γ_1 [°]	0.0000	20.0000	-
γ_2 [°]	0.0000	$\approx$ 0.0000	-
P_1 $[MW]$	9.5781	8.4817	-11.4474
P_2 $[MW]$	4.3217	5.9429	37.5124
P_{WF} $[MW]$	13.8998	14.4245	3.7751

Table 5.1: Yaw Misalignment γ Optimization Results.

[1] Every time a $\approx$ appears in a table in this chapter means that the result can be neglected and approximated to zero. For example, in this case, the optimized value of γ_2 found is 5.8065×10^{-8}.

the quantitative results of the γ optimization. The method implemented find results in line with parametric distributions of the power production. As a matter of fact, confronting the results of γ_1 with the Figure 4.7 demonstrates that the actual peak energy production is at $\gamma_1 = 20°$.

5.2 Added Features

As it has been said in the last chapter, the tilt angle φ and the hub height z_{hub} are variables of interest of this work and they are included as control variables to optimize. The optimization of z_{hub} is not practical for online control systems, nevertheless, it can be an interesting step in the design step of the wind farm. This being said, it is not an actual control variable, but it will be referred as one because its treatment inside FLORIS is like the one given to γ or φ. The normalization of the hub height, for avoiding optimization issues, is done with the rotor's diameter D, and the results of this variable are given in terms of:

$$z_{opt} = z_{hub_0} \pm hD \tag{5.2}$$

where z_{hub_0} is the original design hub height and h is a real number. It is possible to optimize these two variables separately or with one of the other two. In the last section the result of the optimization of the yaw misalignment alone were shown.

Tilt Angle Optimization

The first results presented are for the tilt angle individual optimization with the same settings described in the section before. The lower and upper boundaries set are $b_l = -10°$ and $b_u = 10°$ respectively.

Tilt Angle φ Optimization			
Variable	Baseline	Optimized	Relative Variation [%]
φ_1 [°]	0.0000	10.0000	-
φ_2 [°]	0.0000	$\approx$ 0.0000	-
P_1 $[MW]$	9.5781	9.3337	-2.5513
P_2 $[MW]$	4.3217	4.9067	13.5354
P_{WF} $[MW]$	13.8998	14.2404	2.4503

Table 5.2: Tilt Angle φ Optimization Results.

Given that the power production in function of the first turbine tilt angle has the same distribution than in function of the yaw misalignment of the same rotor, the optimized values of φ should be the same than for γ, i.e., 20 and 0 degrees. However, the upper boundary is chosen to be 10°, so the result will be the closest possible rotation.

Hub Height Optimization

The lower and upper boundaries set for this case are $b_l = 0$ and $b_u = 1$. For the hub height optimization, the actual variable to optimized is the value of h in the equation (5.2).

Hub Height z_{hub} Optimization			
Variable	Baseline	Optimized	Relative Variation [%]
$z_{hub,1}$ $[\pm D]$	0.0000	1.0000	-
$z_{hub,2}$ $[\pm D]$	0.0000	$\approx$ 0.0000	-
P_1 $[MW]$	9.5781	9.5781	0
P_2 $[MW]$	4.3217	9.5188	120.2553
P_{WF} $[MW]$	13.8998	19.0969	37.3895

Table 5.3: Hub Height z_{hub} Optimization Results.

Considering the included adjustment of the Jimenez deficit model, it is correct that the first turbine is the one for which the hub height increases. This because the mentioned correction induces a small upward deflection of the wake. Then, if $z_{hub,2}$ was higher, the second turbine would work under a greater percentage of the upstream wake. As optimal solution, this method shifts the height value to the higher boundary, in fact, the best scenario possible is to displace the wake completely away from the downwind rotor.

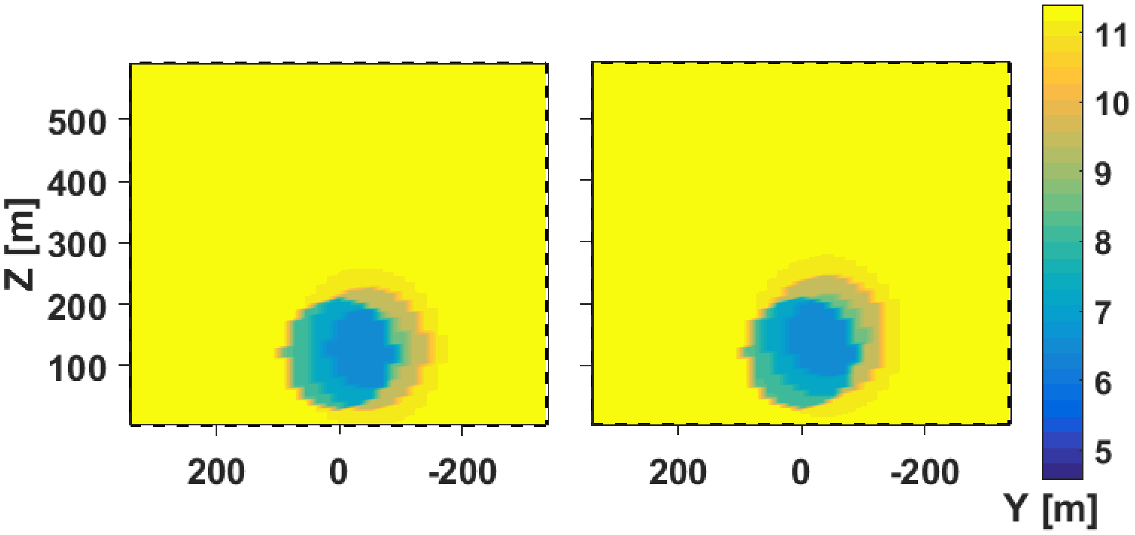
Yaw Misalignment and Tilt Angle Combined Optimization

The lower and upper boundaries are set like in the individual respective optimization case.

Yaw Misalignment γ and Tilt Angle φ Combined Optimization			
Variable	Baseline	Optimized	Relative Variation [%]
γ_1 $[°]$	0.0000	17.4099	-
γ_2 $[°]$	0.0000	$\approx$ 0.0000	-
φ_1 $[°]$	0.0000	10	-
φ_2 $[°]$	0.0000	$\approx$ 0.0000	-
P_1 $[MW]$	9.5781	8.4817	-11.4474
P_2 $[MW]$	4.3217	6.0236	39.3803
P_{WF} $[MW]$	13.8998	14.5053	4.3558

Table 5.4: Yaw Misalignment γ and Tilt Angle φ Combined Optimization Results.

One thing to remark is that, with these results, $\psi = 20°$. This insinuates that the distribution in function of the thrust angle, by rotating the reference frame such that z axis becomes congruent with w, has the same distribution shown in Figure 4.7. This recalling that the turbine is modeled as a perfect circle. It would not be adequate to simply leave both tilt angles equal to 0 and rotate 20° in yaw because the most harmful vertical part of the wake would still affect turbine two. Figure 5.2 shows the cross section of the wakes at the x position of the second rotor. The blue circular zone over $y = 0$ represents the second turbine, and the other shades are the wake of the first generator. The darker blue is the slowest area because of the presence of the center of the upwind wake. By a considerable deflection in both control angles, the downwind rotor would work under a faster zone of the partial wake, which can be seen in Figure 5.2b, as there is a bigger light blue area over the north-east part of the second rotor's wake.



(a) γ Optimization (b) γ and φ Combined Optimization

Figure 5.2: Cross Section Comparison at $x = 5D$ from the first Rotor: Yaw Misalignment Optimization vs Yaw Misalignment and Tilt Angle Combined Optimization.

Yaw Misalignment and Hub Height Combined Optimization

The lower and upper boundaries are set like in the individual respective optimization case.

Yaw Misalignment γ and Hub Height z_{hub} Combined Optimization			
Variable	Baseline	Optimized	Relative Variation [%]
γ_1 [°]	0.0000	≈ 0.0000	-
γ_2 [°]	0.0000	≈ 0.0000	-
$z_{hub,1}1$ [$\pm D$]	0.0000	10	-
$z_{hub,2}$ [$\pm D$]	0.0000	≈ 0.0000	-
P_1 [MW]	9.5781	9.5781	0.0000
P_2 [MW]	4.3217	9.5188	120.2553
P_{WF} [MW]	13.8998	19.0969	37.3895

Table 5.5: Yaw Misalignment γ and Hub Height z_{hub} Combined Optimization Results.

The power results are equal to the hub height single optimization. The optimized values of both turbines' yaw misalignment are approximately zero, meaning that the program evaluates if even a small rotation has any effect. Physically, this means that the horizontal asymmetry of the small section of the first turbine's wake that still interferes with the inflow of the second rotor, is not very notorious, so the program does not detect any variation of the power calculation at a given yaw deflection. The power distribution, in terms of the yaw misalignment, should be approximately or completely constant.

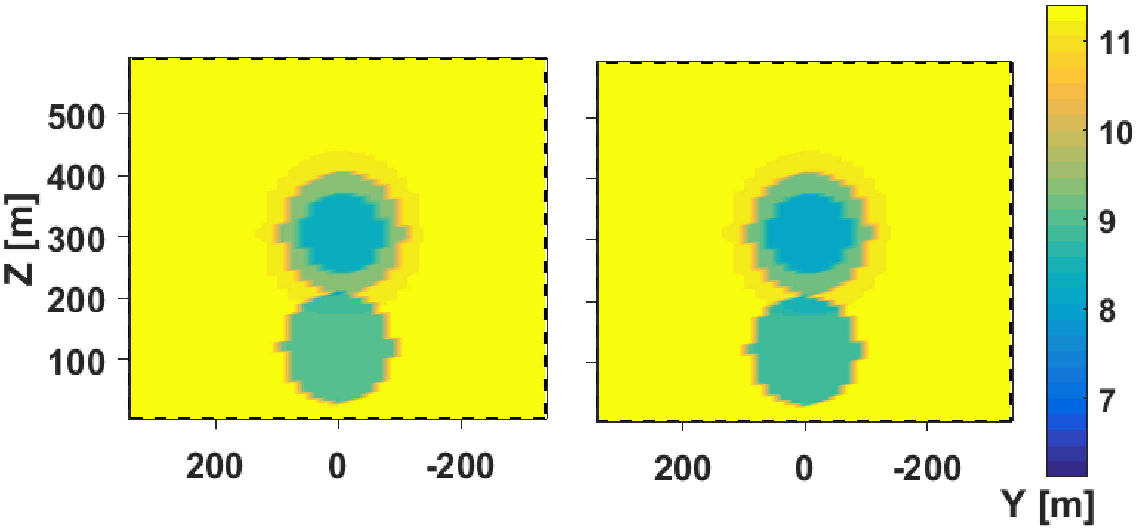
Tilt Angle and Hub Height Combined Optimization

The lower and upper boundaries are set like in the individual respective optimization case.

Tilt Angle φ and Hub Height z_{hub} Combined Optimization			
Variable	Baseline	Optimized	Relative Variation [%]
φ_1 [°]	0.0000	2.5000	-
φ_2 [°]	0.0000	$\approx$ 0.0000	-
$z_{hub,1}1$ [$\pm D$]	0.0000	10	-
$z_{hub,2}$ [$\pm D$]	0.0000	$\approx$ 0.0000	-
P_1 [MW]	9.5781	9.5664	-0.1226
P_2 [MW]	4.3217	9.5329	120.5822
P_{WF} [MW]	13.8998	19.0993	37.4067

Table 5.6: Tilt Angle φ and Hub Height z_{hub} Combined Optimization Results.

As expected, the z_{hub} is augmented to the limit, but there is a small positive tilt rotation of the first rotor. This is also congruent as it will increase the upwards deflection of the wake. Figure 5.3 shows the comparison of the cross section of the wake at $x = 5D$ between a Height Optimization, and a Tilt and Height Combined Optimization. In Figure 5.3a can be seen that in the superior part of the second turbine, there is still a small portion of significant velocity of the upwind wake that interacts with the rotor. Instead, in Figure 5.3b, the modest increase of φ_2 causes that the interaction described happens no more. The first turbine is rotated just the adequate amount to avoid this phenomenon.



(a) z_{hub} Optimization

(b) φ and z_{hub} Combined Optimization

Figure 5.3: Cross Section Comparison at $x = 5D$ from the first Rotor: Hub Height Optimization vs Tilt Angle and Hub Height Combined Optimization.

Chapter 6

Conclusions

The principal requirements of wind farm control methods and how to model the wake main phenomena have been presented. Each submodel has been explained with mathematical detail, exposing the relevant reasoning and assumptions use for their development. By means of an initial evaluation of simple layout and a known turbine, both versions available online of the software were confronted in a quantitative way. The most relevant differences between both versions of their calculation logic were expose and the qualitative comparison shows that the use of the thrust angle ψ inside FLORISSE M had the greatest impact on the differences of the tabulated results. A main file and a module for producing parametric curves have been developed. The main issues of the input complexity and tilt rotation consideration has been addressed with an adjustment proposed for solving both. It has been found that ignoring the sign of the thrust angle ψ, like in the original version does, has a huge impact in the behavior of the curves in function of the turbine's rotations, particularly for negative rotations. For positive deflections, the consideration of the vertical asymmetry of the wake has a negative impact but given the small order of magnitude of the error's increase, this can be neglected. The effects of these modifications over the power distribution of each turbine and of the complete power plant have been graphically illustrated. By confronting FLORIS results with wind tunnel data, it was confirmed the effectiveness of the recommended adjustments. Various experimental cases show important decrease of the relative error, reductions of approximately 16% have been achieved. Non-conventional control variables have been successfully optimized, additionally, results of combined optimizations are presented and it have been explained the physical reasons behind these results.

The main file will facilitate future implementation of FLORIS to other programs for further design investigations. The optimization results, especially for the combined cases, can be compared against the computations of more sophisticated algorithms, and the submodels parameters' optimization option could be tested for increasing the agreement between experimental and simulated results. The efficiency of the proposed new version of the software should be analyzed for more complex wind farm layouts. With the tilt consideration now available, new models could be developed by analyzing the resulting power curves of combined rotations and/or in function of the thrust angle, instead of being just yaw misalignment dependent.

Bibliography

[1] Jackson G. Njiri and Dirk Söffker. "State-of-the-art in wind turbine control: Trends and challenges". In: *ELSEVIER: Renewable and Sustainable Energy Reviews* (2015).

[2] Christopher J Spruce. "Simulation and Control of Windfarms". PhD thesis. University of Oxford, 1993.

[3] J. P. Goit and J. Meyers. "Optimal control of energy extraction in wind-farm boundary layers". In: *Journal of Fluid Mechanics: 768, 5–50* (2015).

[4] Ángel Jiménez, Antonio Crespo, and Emilio Migoya. "Application of a LES technique to characterize the wake deflection of a wind turbine in yaw". In: *Wind Energ.; 3:559–572* (2010).

[5] Majid Bastankhah and Fernando Porté-Agel. "Experimental and theoretical study of wind turbine wakes in yawed conditions". In: *J. Fluid Mech. Vol. 806, pp. 506-541.* (2016).

[6] N. O. Jensen. "A note on wind generator interaction". In: *Risø-M, No. 2411* (1983).

[7] P. M. O. Gebraad et al. "A Data-Driven Model for Wind Plant Power Optimization by Yaw Control". In: *American Control Conference (ACC)* (2014).

[8] Ángel Jiménez, Antonio Crespo, and Emilio Migoya. "A Simple Wake Calculation Procedure". In: *Risø-M, No. 2760* (1988).

[9] Simon De Zutter et al. "Modeling of active yaw systems for small and medium wind turbines".

[10] J.G. Schepers J.W. Wagenaar L.A.H. Machielse. "Controlling Wind in ECN's Scaled Wind Farm". In: *European Wind Energy Association (EWEA)* (2012).

[11] C.L. Bottasso and J. Schreiber. "Online model updating by a wake detector for wind farm control". In: *Annual American Control Conference (ACC)* (2018).

[12] G.P. Corten and P. Schaak. "Heat and Flux: Increase of Wind Farm Production by Reduction of the Axial Induction". In: *BT4.2: Development of Measurement Methods.* European Wind Energy Conference, EWEC. (Madrid, Spain, June 16–19, 2003). 2003.

[13] Wim Munters and Johan Meyers. "Towards practical dynamic induction control of wind farms: analysis of optimally controlled wind-farm boundary layers and sinusoidal induction control of first-row turbines". In: *Wind Energ. Sci., 3, 409–425, 2018* (2018).

[14] TU Delft DCSC Data Driven Control Group Revision. *FLORISSE M.* 2019. URL: `https://github.com/TUDelft-DataDrivenControl/FLORISSE_M`.

[15] NREL. *FLORIS. Version 1.0.0.* 2019. URL: `https://github.com/NREL/floris`.

[16] J. Schreiber et al. “Verification and Calibration of a Reduced Order Wind Farm Model by Wind Tunnel Experiments”. In: *Journal of Physics: Conf. Series 854* (2017).

[17] TU Delft DCSC Data Driven Control Group Revision. *FLORISSE M Documentation.* 2018. URL: `https://florisse-m.readthedocs.io/en/latest/index.html`.

[18] Christian Bak et al. *Description of the DTU 10 MW Reference Wind Turbine, DTU Wind Energy Report-I-0092.* 2013.

[19] H. Glauert. *A General Theory of the Autogyro.* HM Stationery Office, 1926.

[20] J. H. Lee and V. Chu. *Turbulent Jets and Plumes: A Lagrangian Approach.* Springer, 2012.

[21] Sten Frandsen et al. “Analytical modelling of wind speed deficit in large wind farms”. In: *Wind Energy, 9:39–53* (2006).

[22] I.Katic, J.Højstup, and N.O.Jensen. “A simple model for cluster efficiency”. In: *European Wind Energy Association Conference and Exhibition* (1986).

[23] CL-Windcon. *D1.2: Description of the reference and the control-oriented wind farm models.* 2018.

[24] A. Crespo and J. Hernández. “Turbulence characteristics in wind-turbine wakes”. In: *Journal of Wind Engineering and Industrial Aerodynamics* (1996).

[25] Amin Niayifar and Fernando Porté-Agel. “A new analytical model for wind farm power prediction”. In: *Journal of Physics: Conference Series 625* (2015).

[26] Janez Funda, Russel H. Taylor, and Richard P. Paul. “On Homogeneous Transforms, Quaternions, and Computational Efficiency”. In: *IEEE Transactions on Robotics and Automation, Vol. 6 , No. 3* (1990).

[27] Fletcher Dunn and Ian Parberry. *3D Math Primer for Graphics and Game Development.* A K Peters/CRC Press, 2011.

[28] 3Blue1Brown. *Quaternions and 3d rotation, explained interactively.* URL: `https://www.youtube.com/watch?v=zjMuIxRvygQ`.

[29] CL-Windcon. *D3.1: Definition of wind tunnel testing conditions.* 2017.

[30] MathWorks. *fmincon.* URL: `https://la.mathworks.com/help/optim/ug/fmincon.html?lang=en#busp5fq-7`.

Appendix A

Additional Graphs of Experimental Comparison

This appendix illustrate the graphic comparison between the FLORIS generated distributions and the experimental data of additional cases evaluated in [29].

The same explanations given in the Chapter 4 apply for this content.

They are organized in ascendant order of the magnitude of the wind direction θ_W. The chosen high turbulence scenarios are the closest equivalent to those with low turbulence.

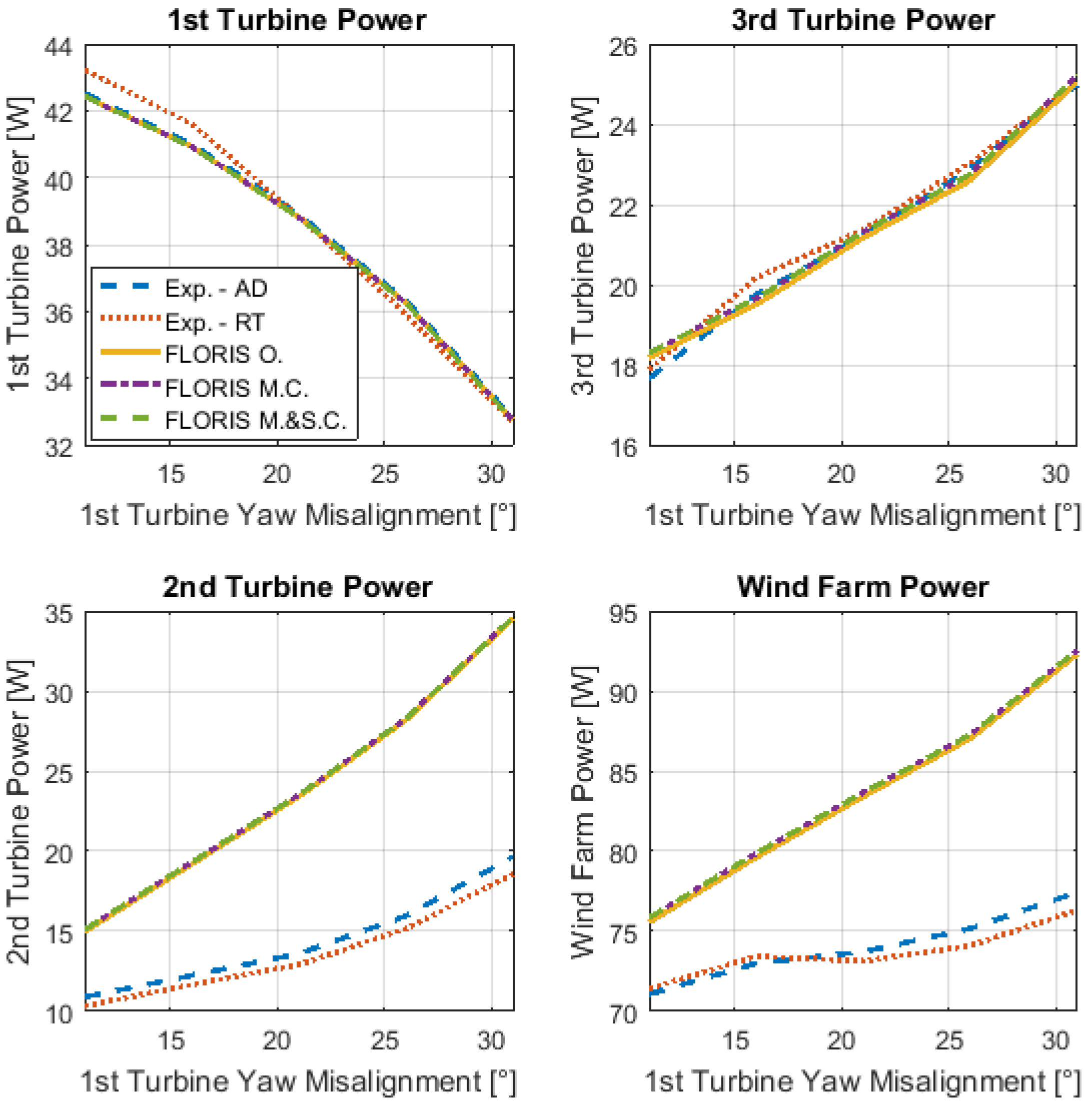


Figure A.1: Power Distributions of all the components of the wind farm. Test 1.7: $\theta_W = 0.0°$, $\gamma_1 = 11 : 5 : 31[°]$, $\gamma_2 = 19°$, $TI = 5\%$.

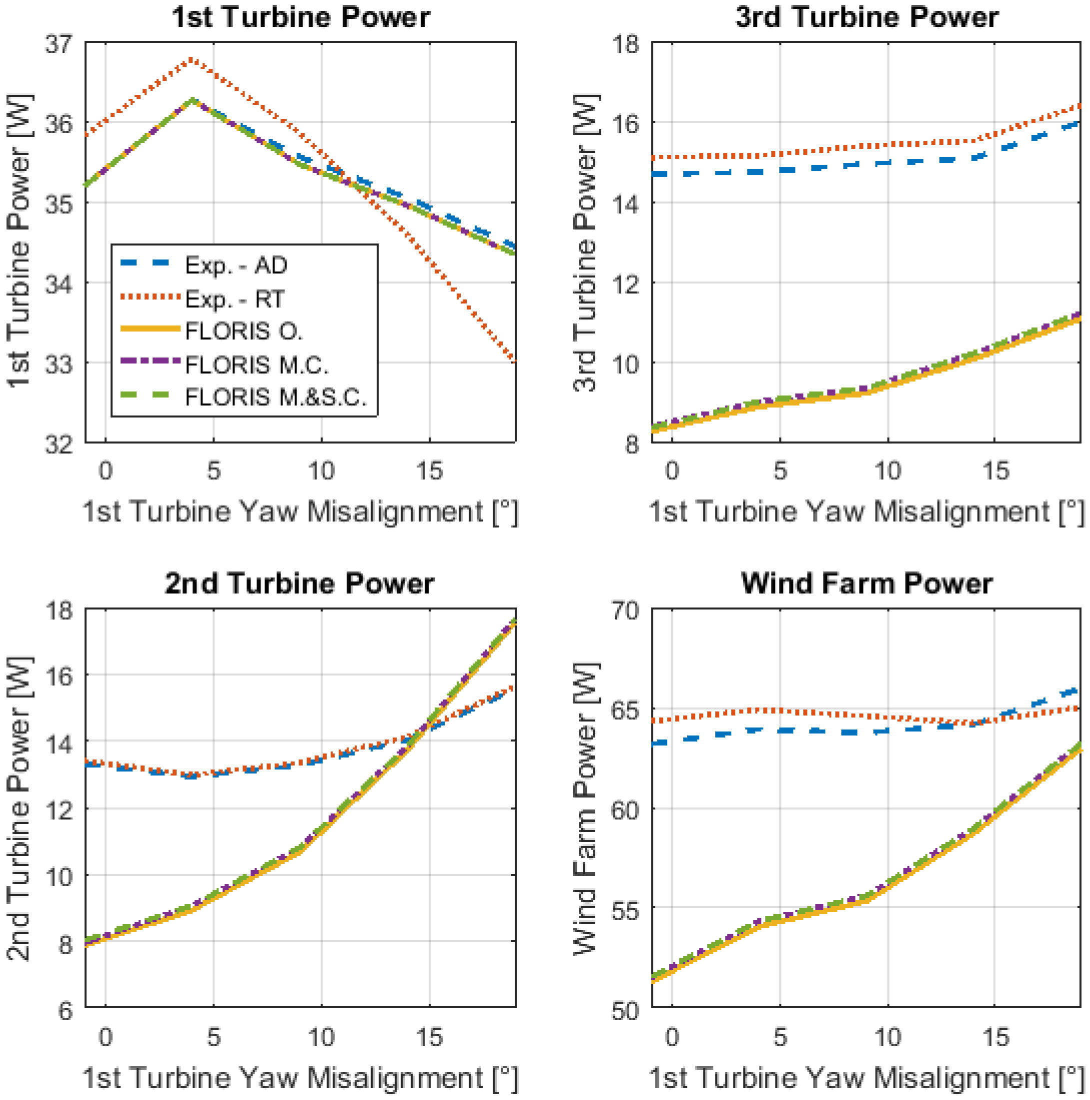


Figure A.2: Power Distributions of all the components of the wind farm. Test 4.7: $\theta_W = 0.0°$, $\gamma_1 = -1 : 5 : 19[°]$, $\gamma_2 = 9°$, $TI = 10\%$.

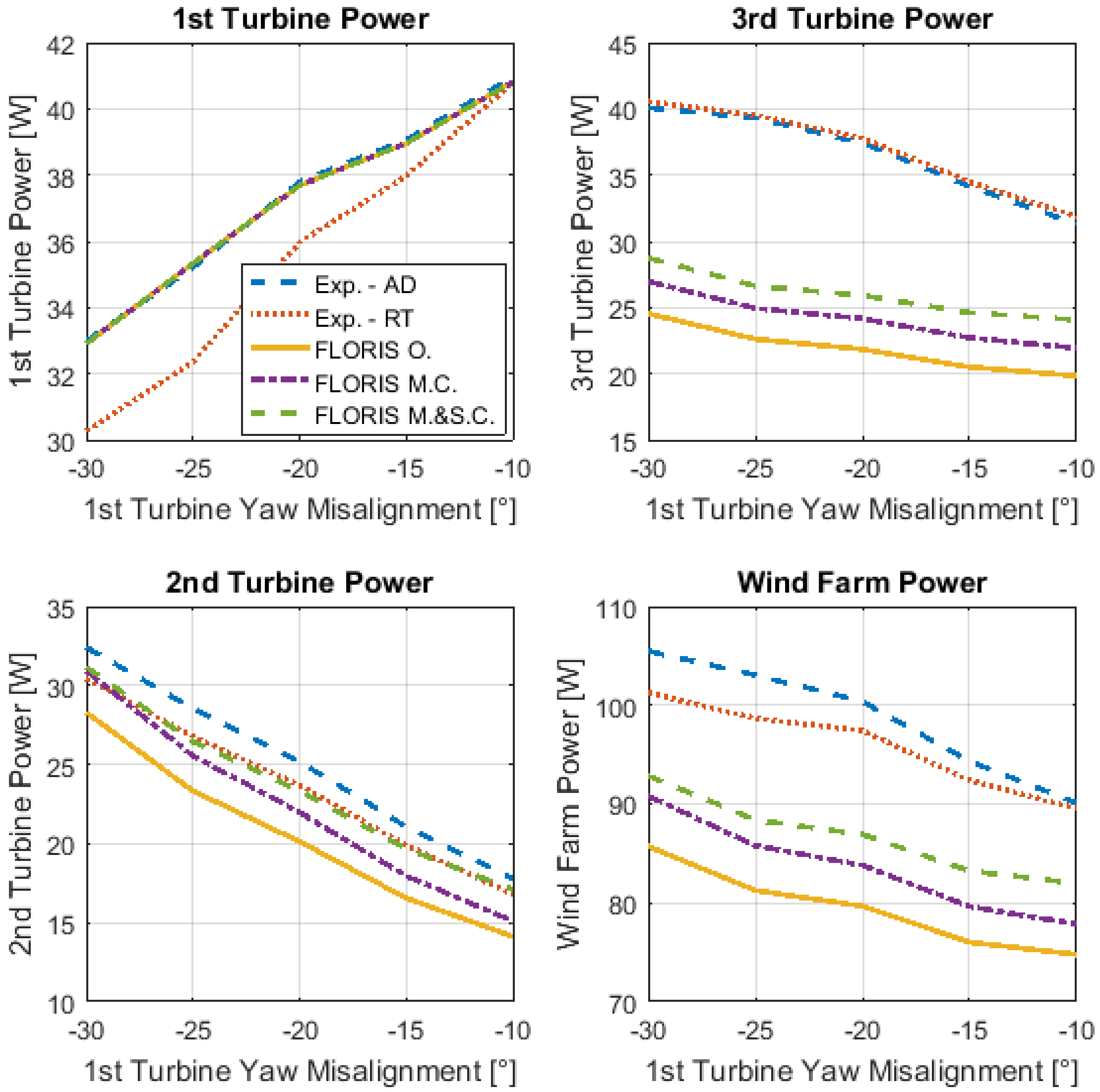


Figure A.3: Power Distributions of all the components of the wind farm. Test 1.8: $\theta_W = 2.29°$, $\gamma_1 = -30 : 5 : -10[°]$, $\gamma_2 = -20°$, $TI = 5\%$.

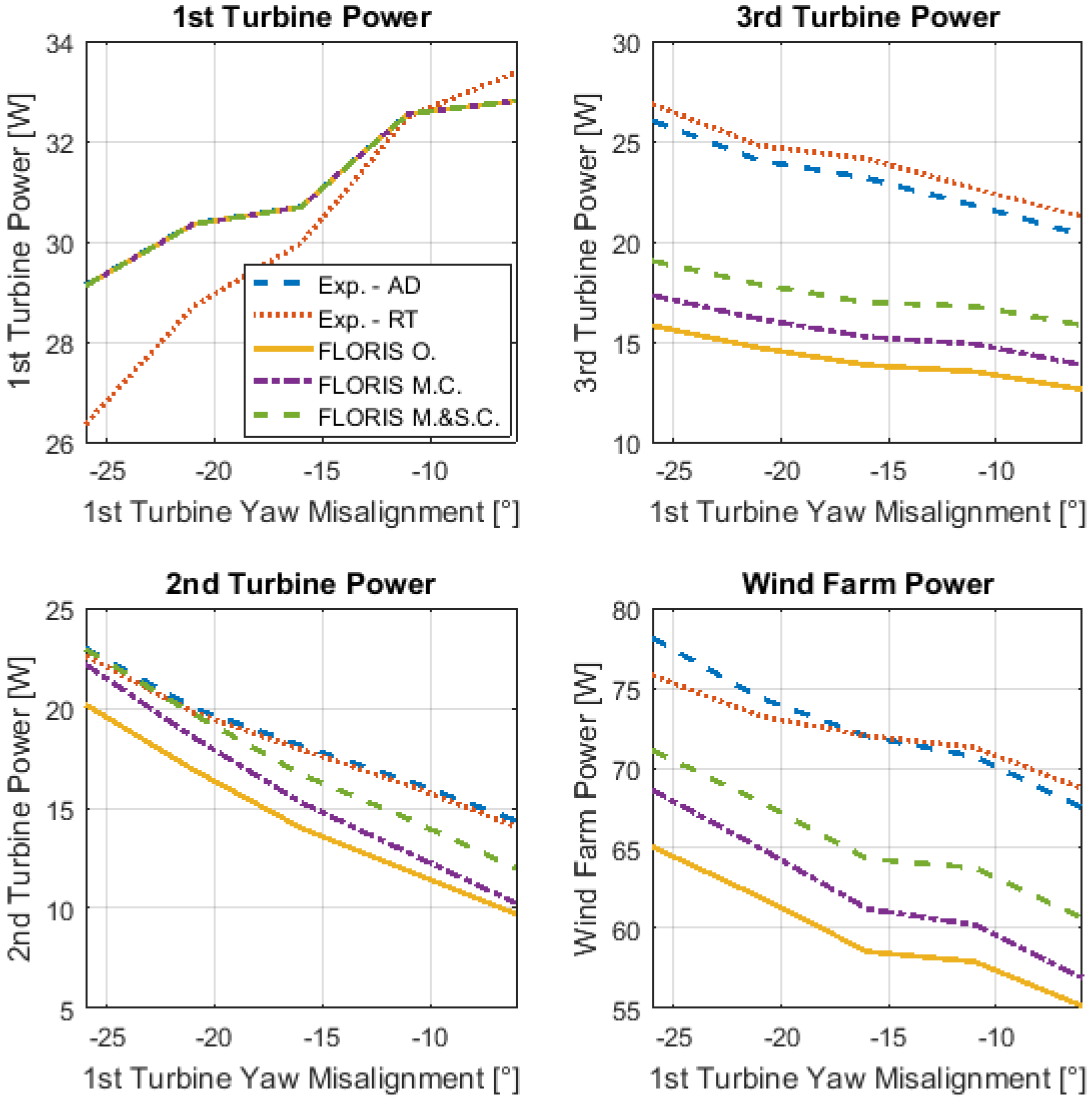


Figure A.4: Power Distributions of all the components of the wind farm. Test 4.8: $\theta_W = 2.29°$, $\gamma_1 = -26 : 5 : -6[°]$, $\gamma_2 = -15°$, $TI = 10\%$.

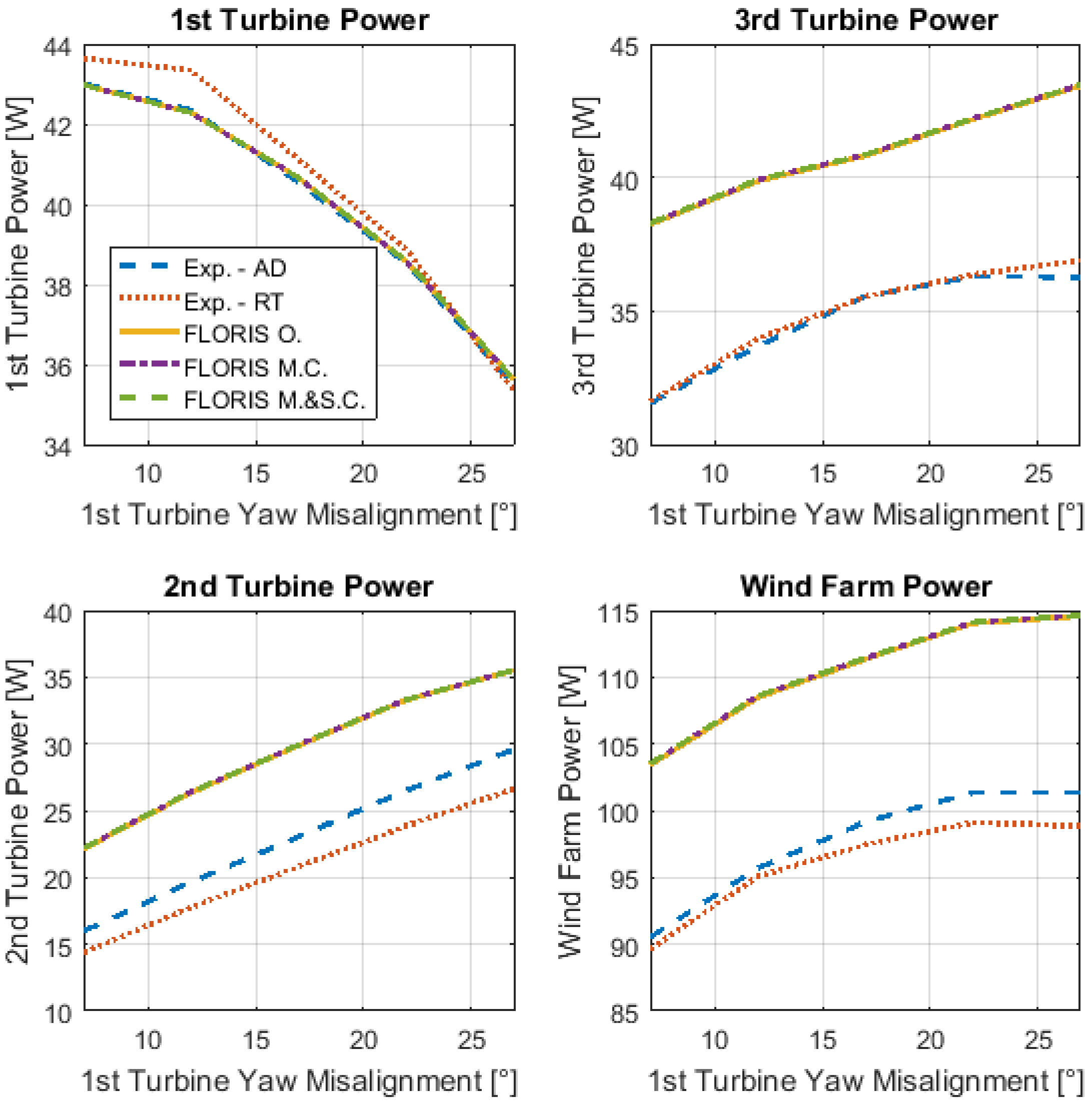


Figure A.5: Power Distributions of all the components of the wind farm. Test 1.4: $\theta_W = -4.58°$, $\gamma_1 = 7:5:27[°]$, $\gamma_2 = 25°$, $TI = 5\%$.

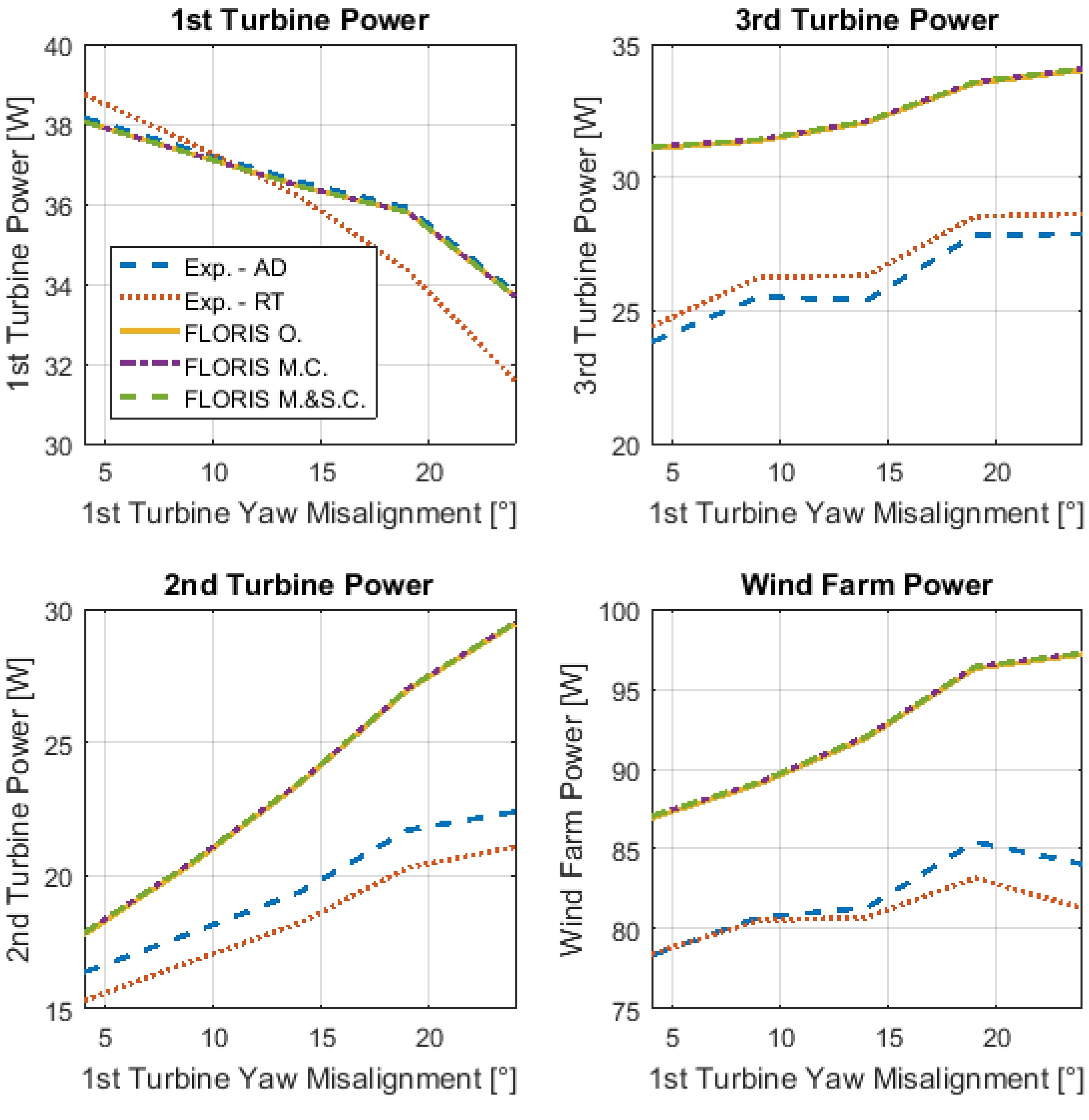


Figure A.6: Power Distributions of all the components of the wind farm. Test 4.4: $\theta_W = -4.58°$, $\gamma_1 = 4 : 5 : 24[°]$, $\gamma_2 = 22°$, $TI = 10\%$.

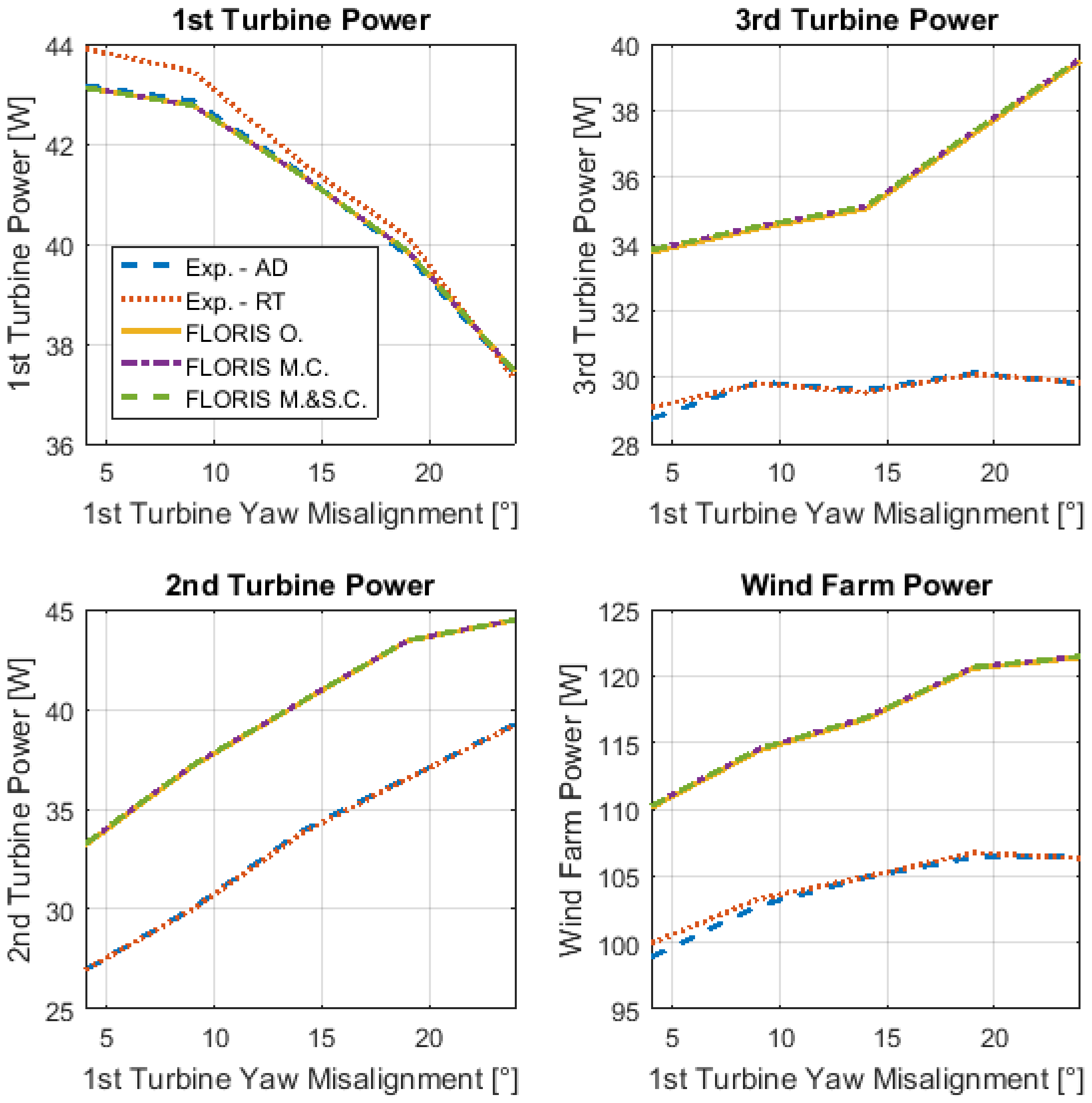


Figure A.7: Power Distributions of all the components of the wind farm. Test 1.3: $\theta_W = -6.89°$, $\gamma_1 = 4 : 5 : 24[°]$, $\gamma_2 = 5°$, $TI = 5\%$.

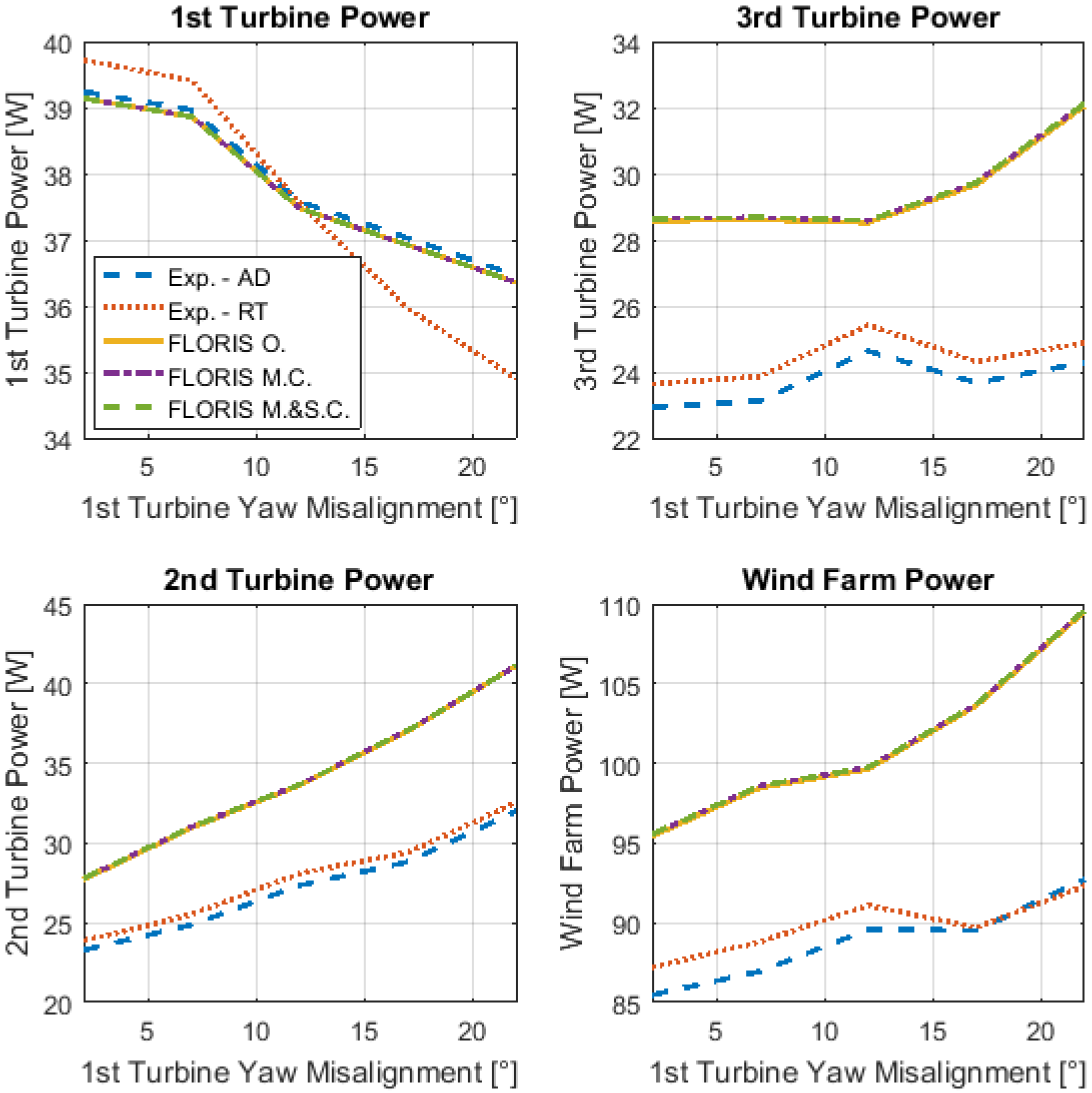


Figure A.8: Power Distributions of all the components of the wind farm. Test 4.3: $\theta_W = -6.89°$, $\gamma_1 = 2:5:22[°]$, $\gamma_2 = 3°$, $TI = 10\%$.

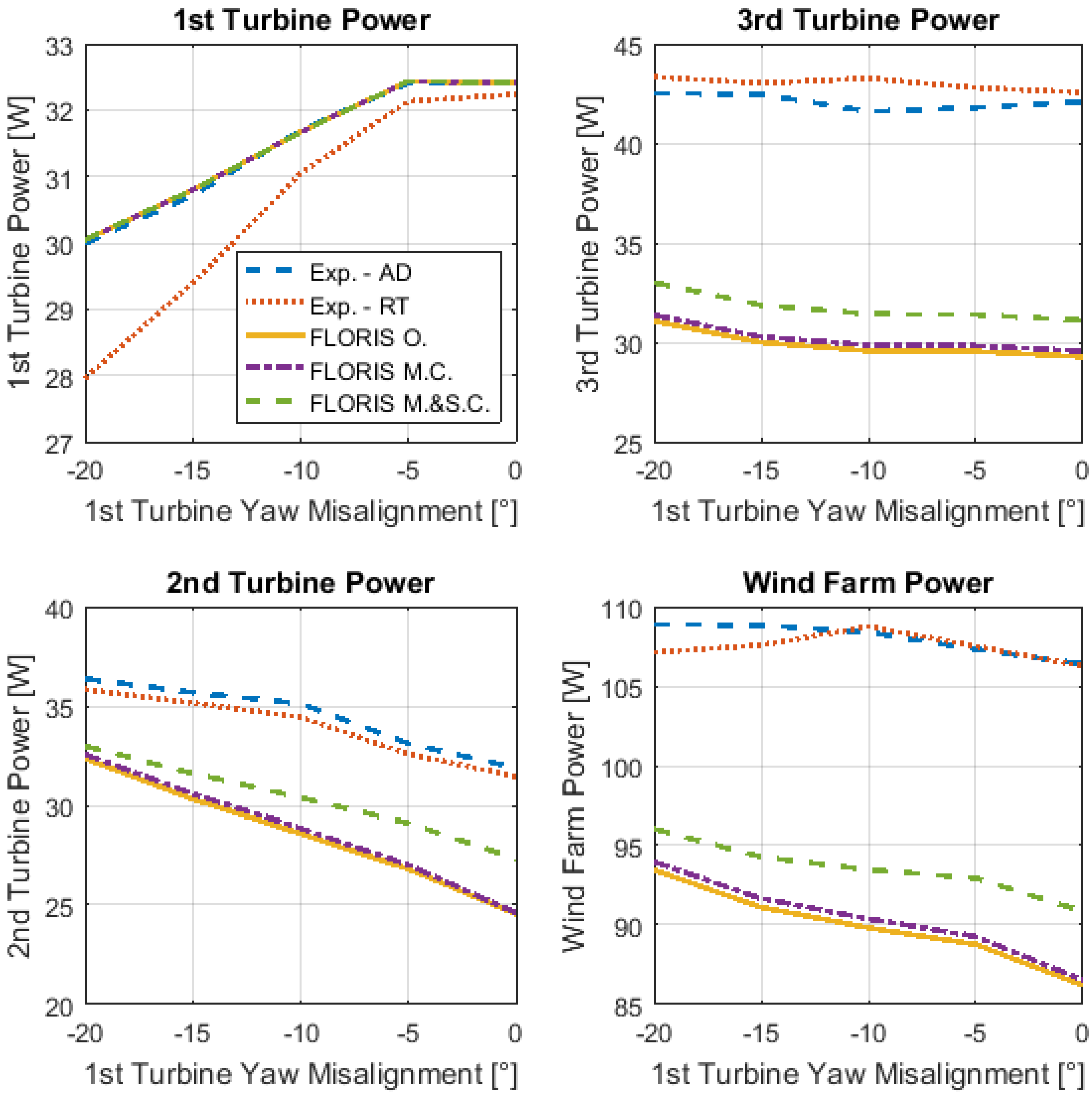


Figure A.9: Power Distributions of all the components of the wind farm. Test 1.11: $\theta_W = 9.21°$, $\gamma_1 = -20 : 5 : 0[°]$, $\gamma_2 = -10°$, $TI = 5\%$.

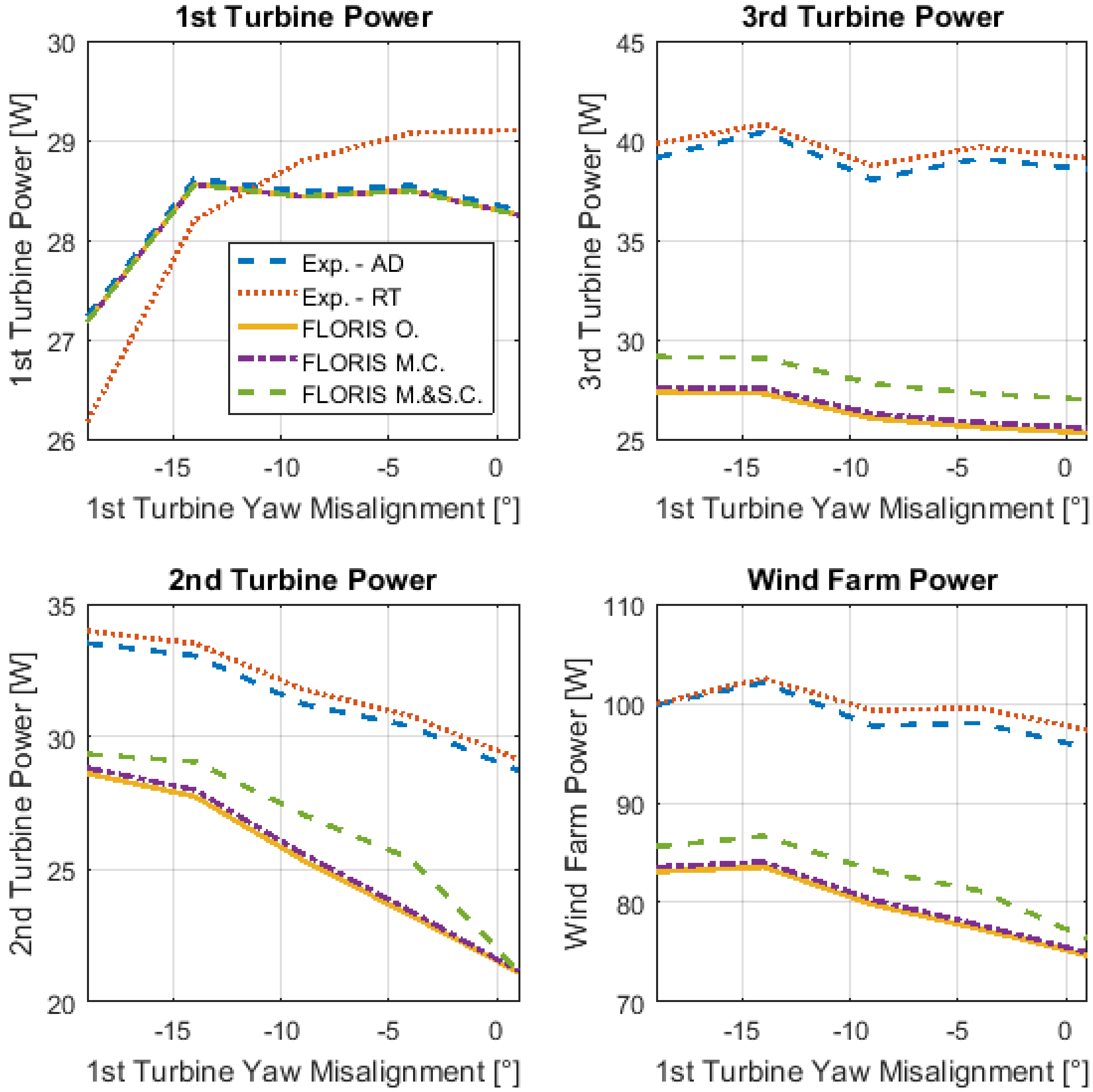


Figure A.10: Power Distributions of all the components of the wind farm. Test 4.11: $\theta_W = 9.21°$, $\gamma_1 = -19 : 5 : -1[°]$, $\gamma_2 = -9°$, $TI = 10\%$.

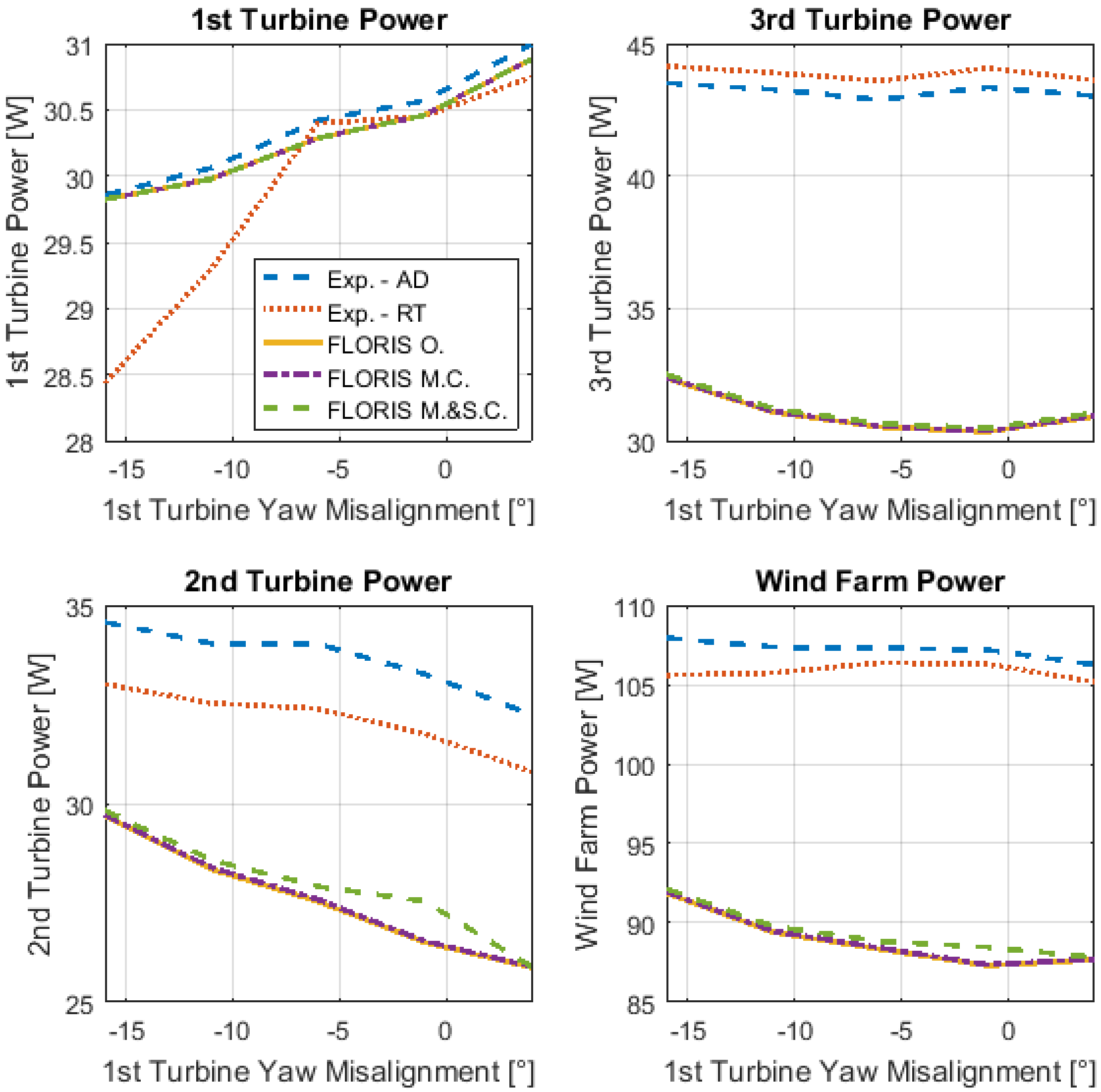


Figure A.11: Power Distributions of all the components of the wind farm. Test 1.12: $\theta_W = 11.54°$, $\gamma_1 = -16 : 5 : 4[°]$, $\gamma_2 = -16°$, $TI = 5\%$.

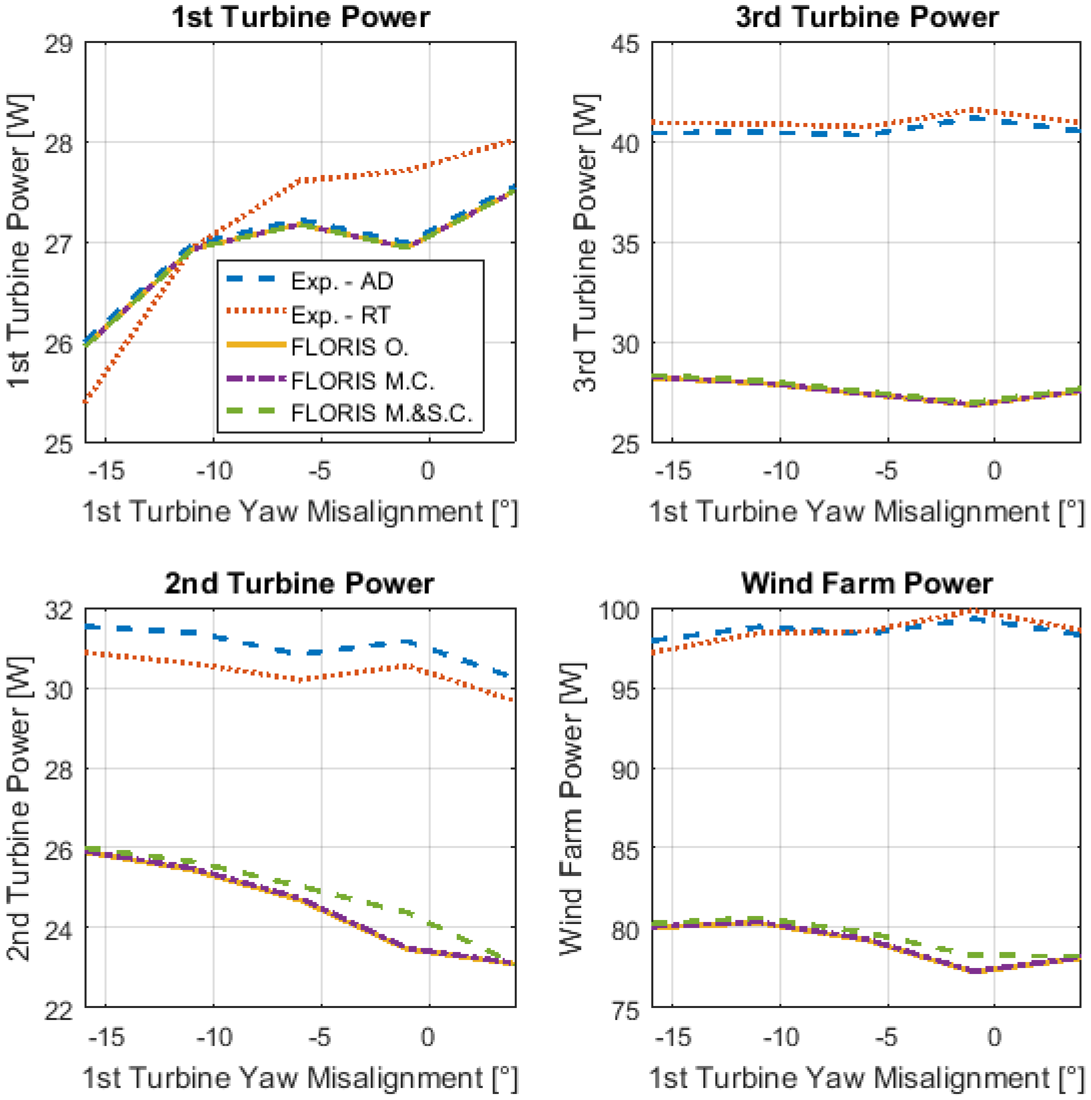


Figure A.12: Power Distributions of all the components of the wind farm. Test 4.12: $\theta_W = 11.54°$, $\gamma_1 = -16 : 5 : 4[°]$, $\gamma_2 = -16°$, $TI = 10\%$.