POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

Towards a Digital Twin for Optimal Control Strategies in Wind Farms

LAUREA MAGISTRALE IN AERONAUTICAL ENGINEERING - INGEGNERIA AERONAUTICA

Author: LORENZO FIRRINCIELI

Advisor: PROF. ALESSANDRO CROCE

Co-advisor: PROF. ANDREA MATTA

Academic year: 2023-2024

Summary

This work shows an ongoing activity with the main goal of developing a Digital Twin model to be applied to a wind farm optimal control strategy. Currently, the control technique is focused on the Wake Steering for a given layout of the cluster.

1. Introduction

A Digital Twin (DT) is defined as a virtual representation of a physical asset enabled through data and simulators for several purposes, among which real-time prediction, monitoring, informed decision-making, control and optimization [1]. Wind farms are aimed to maximize the power output reducing collateral costs such us maintenance costs. The maximization of the power output can be achieved only by means of smart strategies and optimization techniques about the layout of the wind turbines and the wake inside the cluster. Due to several reasons (technical, environmental, bureaucratic constraints), sometimes the layouts are rigid, therefore the only way to optimize the energy production is to operate specific coordinated control techniques on the single turbines. The main techniques are Wake Steering (or Wake Redirection) and Wake Recovery; the former is the most studied in literature and is based on specific desired yaw misalignments of upwind turbines in order to move the wake away from the downwind turbines, while the latter is represented by some specific actions (for instance dynamic axial induction control through IPC or derating wind turbines) to have a faster recovery of the downwind speed.

Since Wake Steering has shown very promising results for "freeze" layouts, the goal of this study is to give a wind farm designer a compact and reliable tool able to predict the control strategy for the yaw of each turbine in a wind farm with a given wind history.

In order to choose the proper simulation tool, the first step has been doing an exhaustive numerical validation of the wind tunnel campaign for the EU Project CL-Windcon [2], performed by Politecnico di Milano. Among all of them, FLORIDyn [3] has been preferred as a basis for its low computational costs, its simplicity and the possibility to manage a dynamic simulation. The expected result is an improvement in power output with respect to the greedy strategy (therefore, without any coordinated control techniques), having the advantage of avoiding a static and rigid Look-Up Table, extracted for instance via FLORIS [4], for single wind speeds.

2. Software Validation

The wind tunnel campaign was performed between 2017 and 2019, including seven entries with several tests to characterize the wake shed and to validate open-loop control techniques for the EU Project. The tests to perform this validation were three: the first two were with constant wind speed and constant wind direction, while the third one was with constant wind speed but variable wind direction to validate open-loop control techniques.

Regarding the static tests, the first test was with two aligned turbines, while the second one with a slight lateral displacement of the downwind turbine under a yawed condition; the last one represented a dynamic condition to use a Look-Up Table pre-tuned by FLORIS (statically) for different wind directions.

The goals of the validation were to find the best tool to describe sufficiently well these tests, comparing the simulated wakes at specific distances with the real data and to improve the static open-loop strategy of wake steering.

The tools chosen for the validation were FLORIS, FAST.Farm [5] and FLORIDyn: they are low and medium-fidelity open-source software; more sophisticated analyses were omitted in order to privilege the relatively low computational cost for a sufficient level of accuracy.

The static tests showed a very good level of fidelity for all the tools, also capturing the asymmetric wake due to the yawed condition for the second test. The worst result came from FAST.Farm, especially regarding the computational time: while FLORIS and FLORIDyn's output were ready in 1 second, FAST.Farm required more than 30 minutes, due to a forced choice of a very small-time step to describe correctly the wake dynamics inside the tool. Moreover, since the default parameters of FAST.Farm were calibrated with SOWFA for NREL5MW, the accuracy was negatively affected by this unprecise calibration since the wind turbine model of the campaign had a rotor diameter of a 1.1 m and a rated rotor speed of 850 rpm (TUM G1 Model).

The dynamic test was performed only in FLORIDyn (see Figure 1), because FLORIS incorporates only steady-state engineering wake models, while FAST.Farm was not suitable because of its complex and expensive implementation compared to FLORIDyn.

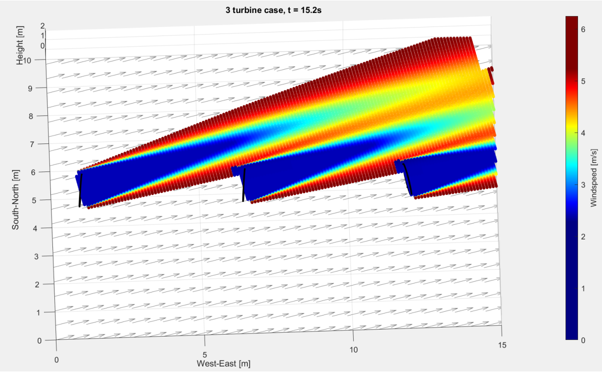

Figure 1: FLORIDyn: Screenshot of Test 3

The control strategy (LUT) was implemented in the code of the software and validated.

The wake output was affected by error due to the small layout and a very dynamic yaw condition. That final result was not considered as determinant, because it was defined as a pathological case: the wake calculation in FLORIDyn uses points (Observation Points) that contain flow information, travelling at the undisturbed speed; longitudinally the layout was included in 11 meters (10 diameters), therefore the parcels travelled the whole cluster in less than three seconds (considering the wind speed of test 3), while in a more realistic scenario 10 diameters are more than 1 km, so the travel time of the observation points is much higher and the inaccuracy from the numerical model has a smaller impact.

3. Towards a Digital Twin

The new tool has been developed in FLORIDyn due to its good performance shown in the validation chapter and a very simple and intuitive structure of the code.

The structure of this "primitive" digital twin can be divided into four categories: Inputs, Initialization, Optimum Control and Simulation. Furthermore, the inputs are classified in "General Inputs" (environment and layout) and "Initialization Inputs"; while the former ones are the classical inputs that change for each case study, the latter ones are an intrinsic characteristic of this code.

The methodology is based on a Dynamic Look-Up Table, built by an automatic initialization process, where only few inputs (the "Initialization Inputs") are managed by the designer. The initialization consists in a series of simulations, each of which involves only one turbine experiencing a set of yaw misalignments for a specific wind condition. The designer determines the number of simulations and the computational time, choosing the resolution of the vectors that represent the input of the simulations. In fact, N, D, τ are respectively the number of elements of the initialized vectors of wind speed, wind direction and the yaw misalignments. The last one is from -30 to 30 degrees and represents all the conditions that the single turbine must experience in one simulation of the initialization. The other two are shaped in function of the wind condition from the general outputs and, therefore, determine the number of simulation $(N \times D)$.

This process leads to the construction of a database for the wake steering technique, where the element (d, n) contains important information extracted from the simulation for wind speed n and wind direction d: a column vector $\tau \times 1$ stores the power production for each yaw misalignment, while a matrix $\tau \times (M-1)$ represents the wake. Specifically, each element of this matrix represents the wake at a specific relative position of a downwind turbine with respect to the closest upwind turbine for a yaw misalignment of the latter: the relative positions are automatically calculated immediately after knowing the layout of the farm (M is the number of wind turbines that influence each other, identifying upwind and downwind turbines for a given wind history).

A higher resolution of the database, obviously increases the accuracy of the optimal control that will be explained in the next lines, but the computational cost could become a decision factor: for instance, an initialization with $N = 5$, $D = 10$, $\tau = 13$ requires less than 5 minutes, while another one with $N = 30$, $D = 31$, $\tau = 27$ approximately 4 hours.

This strategic structure allows the implementation of a very basic algorithm to develop the control technique of Wake Steering: the goal is to maximize the overall power production of the farm, neglecting other constraints such as ultimate loads and fatigue, finding the optimal combination of yaw misalignment of all the turbines excepting for the last one (its wake is harmless, so the Greedy control strategy is always the best one).

Thanks to the already explained methodology, it is possible to extract the whole set of combinations of the total power of the farm and look for the one that maximizes the power output.

In fact, there is a for-loop to create iteratively the optimum yaw for the upwind turbines using the database: for each time step of the wind history, the structure of the database is extracted for the specific wind speed and wind direction; for the first upwind turbine, the wind input will be the undisturbed flow condition, while for the downwind turbines there will be "τ-combinations" of wind speed and wind direction due to different yaws of the upwind turbine. The total combinations for each time step will be $\tau^{(M-1)}$, so the main limitation is the exponential complexity of this tool: if M increases, the tool can be very inefficient, so for this reason at this moment only small groups of turbines of a cluster are considered, classifying *a priori* the upwind-downwind disposition.

Some tests (see Figures 2, 3) have been performed to show that even if the algorithm is not fully optimized, the control technique from this new tool improves not only the greedy technique (each turbine uses an individual yaw control to have zero yaw misalignment for each wind direction), but also the LUT of FLORIS, used in the third test of the validation process, when the velocity varies rather than remaining constant (see Figure 3).

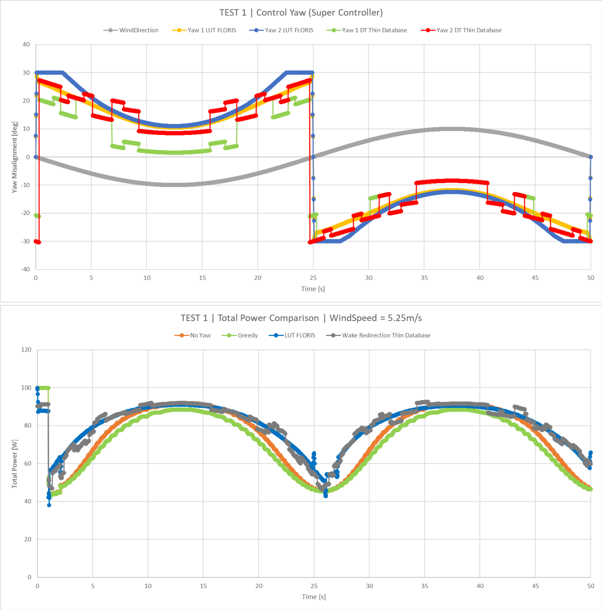

Figure 2: Control Strategy and Power Production (Constant Wind Speed)

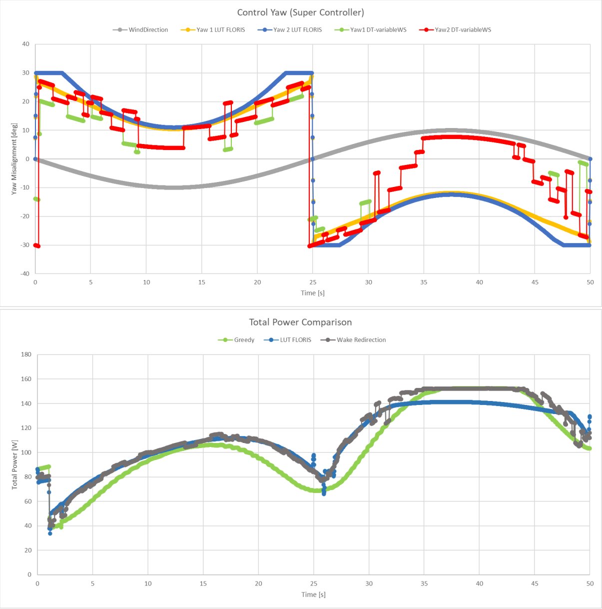

Figure 3: Control Strategy and Power Production (Linear Wind Speed Increase)

4. Conclusion

This promising tool is not restricted to a single wind speed like open-loop strategy “LUT FLORIS”, therefore with variable wind speed the ongoing Digital Twin (DT) is the only optimization tool to be robust and to still perform a good strategy. In fact, comparing the energy production with the greedy strategy, for a constant wind speed and sinusoidal wind direction (see Figure 2) the new tool presents +11.65%, while LUT FLORIS achieves a similar result with +11.94%; instead, the situation changes when there is a ramp wind speed around the rated speed: the former continues to significantly improve the greedy strategy (+8.54%), whereas the latter, if not updated again, reduces its effectiveness (+6.13%) (see Figure 3).

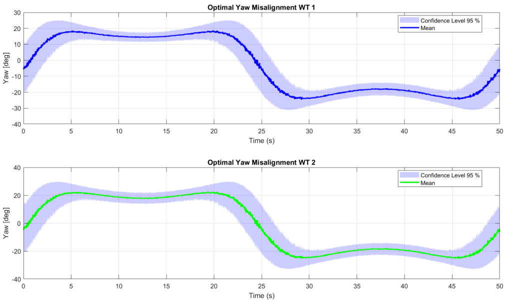

Figure 4: Robustness Analysis, Confidence Level with 1000 Simulations

Moreover, a Monte Carlo analysis with 1000 simulations was conducted to verify that the confidence level for specific initialization inputs was limited. The results (see Figure 4) show that the ongoing Digital Twin also demonstrates robust behavior when white noise is applied to the input wind conditions.

The next steps will aim to solve the inefficient algorithm for big clusters, overcoming the exponential complexity through the introduction of alternatives such as Heuristics, Dynamic Programming, and Metaheuristics. The last ones, like genetic algorithms, can be particularly suitable because they explore solutions efficiently. Another promising direction is the use of a surrogate model for initialization, reducing computational costs and enabling integration with Deep Reinforcement Learning to learn an optimal Wake Steering control policy offline, avoiding costly re-optimizations during online deployment.

Another useful future development could be an interface between FLORIS and the future DT to achieve a pre-initialization to obtain the optimal layout thanks to the Genetic Random Search algorithm, already implemented in FLORIS, and then start the loop of the new tool presented in this work. For this purpose, the most comfortable choice could be using FLORISSE that basically represents FLORIS via MATLAB, since FLORIDyn and the ongoing DT are in MATLAB.

A more sophisticated addition could be the implementation of different mode control optimization techniques such as wake recovery through derating and wake recovery through dynamic induction factor.

In conclusion, the current tool can be defined as a Physics-Based Model, because it uses numerical wake models, validated through experimental data from the wind tunnel campaign. However, future implementations may incorporate features such as Neural Networks, which could serve as a foundation for developing a Prescriptive DT, providing optimized recommendations and automated decision-making to design a coordinated control strategy for wind farms.

References

[1] Florian Stadtmann, Adil Rasheed, Trond Kvamsdal, Kjetil André Johannessen, Omer San, Konstanze Kölle, John Olav Tande, Idar Barstad, Alexis Benhamou, Thomas Brathaug, Tore Christiansen, Anouk-Letizia Firle, Alexander Fjeldly, Lars Frøyd, Alexander Gleim, Alexander Høiberget, Catherine Meissner, Guttorm Nygård, Jørgen Olsen, Håvard Paulshus, Tore Rasmussen, Elling Rishoff, Francesco Scibilia, and John Olav Skogås. Digital twins in wind energy: Emerging technologies and industry-informed future directions. 11:110762–110795. Conference Name: IEEE Access.

[2] CL-Windcon | Closed loop wind farm control - Available at: `https://www.clwindcon.eu/`.

[3] Marcus Becker, Bastian Ritter, Bart Doekemeijer, Daan Van Der Hoek, Ulrich Konigorski, Dries Allaerts, and Jan-Willem Van Wingerden. The revised FLORIDyn model: implementation of heterogeneous flow and the gaussian wake. 7(6):2163–2179.

[4] NREL/floris, available at: `https://github.com/NREL/floris`.

[5] Jason Jonkman and Kelsey Shaler. FAST.Farm User's Guide and Theory Manual.