Evaluating techniques for redirecting turbine wake using SOWFA

P. Fleming¹, P. Gebraad², S. Lee¹, J.W. van Wingerden², K. Johnson¹, M. Churchfield¹, J. Michalakes¹, P. Spalart³, P. Moriarty¹

¹National Renewable Energy Laboratory, Golden CO, USA, paul.fleming@nrel.gov
²Delft University of Technology, Delft, The Netherlands, J.W.vanWingerden@TUDelft.nl
³Boeing Commercial Airplanes, Seattle WA

Abstract

Wind plant control is an active field of research in which controllers seek to maximize overall wind-plant performance in terms of power production, loading, or both. Such control strategies are often different from those that are optimal for an individual turbine. One type of wind-plant control method is to redirect the wake of an upstream turbine in order to avoid downstream turbines. In this paper, we investigate several possible methods for redirecting a turbine’s wake, including some existing and some novel approaches. The methods are compared in terms of their ability to redirect turbine wake, the effect on turbine power capture, and turbine loading using the high-fidelity wind plant simulation tool SOWFA.

1 Introduction

Wind turbine wakes are complex and difficult to model. When turbines are located together in wind power plants, wake interaction between turbines can decrease energy capture and increase turbine loads. Therefore, recent research has focused on the design of wind plant controllers to mitigate these effects. Often in the literature, the controllers are based on modifying an individual turbine’s axial induction factor by adjusting pitch and torque. Example studies that use this approach to optimize the global wind plant power capture include [1, 2, 3].

An alternative approach to wind plant power optimization is to redirect the wake using yaw misalignment rather than to optimize induction. In this method, when two turbines are aligned in the wind direction, the upstream turbine intentionally misaligns its yaw angle so as to apply skew to the wake such that it avoids the downstream turbine. This method has been studied experimentally in [4] and in computational fluid dynamics (CFD) simulation in [5] with encouraging results. In a similar way, vertical wake redirection obtained by changes in rotor tilt angle has been investigated in [6] using a CFD model. Note that both [5] and [6] use an actuator disk model of the turbine and [6] assumes laminar flow.

In this paper, we examine the potential of turbine controllers to redirect the turbine wake. This study is done through experiments with the high-fidelity wind plant modeling tool SOWFA.
Wake redirection methods to be examined include both the yaw- and tilt-angle based methods discussed above. We further add an additional novel, to the best of our knowledge, approach. The method attempts to employ individual pitch control (IPC) to achieve a horizontal or vertical wake skew by intentionally inducing a yaw or tilt moment. IPC is typically used to remove these moments, but we use it in reverse to apply them. The four techniques to be evaluated are illustrated in fig. 1.

The contributions of this paper are: first, we introduce a novel approach to achieving wake redirection (IPC). Second, we use high-fidelity simulations to evaluate the methods both in terms of their capability to redirect wakes and in terms of the effect on turbine power and loads.

![Figure 1](image.png)

Figure 1: Techniques considered for redirecting the wake. Note that arrow directions in yaw and tilt case correspond to positive increases.

The remainder of this paper is organized as follows. Section 2 describes the high-fidelity wind plant simulation tool, SOWFA, used in this study. Section 3 provides details of the simulation experiment setup. Section 4 provides the results and analysis of the experiment. Conclusions are given in section 5.

## 2 SOWFA

SOWFA [7] is a CFD tool coupled with the National Renewable Energy Laboratory’s (NREL’s) FAST turbine simulator tool [8] for studying wind plant behavior. The CFD solver is based on the OpenFOAM CFD toolbox [9]. Specifically, a large-eddy simulation (LES) is used, which directly resolves the larger, energy-containing turbulent scales, to simulate the atmospheric boundary layer and the turbulence contained within it. Then, actuator line turbine models are placed
in the flow to create wakes that interact with one another, and the actuator lines are coupled with FAST. Extensive details are given by Churchfield et al. [10], and are summarized here.

The flow is computed using an unstructured, collocated, variable, finite-volume formulation that is second-order accurate in time and space. The filtered momentum equation is solved along with an elliptic equation for pressure that enforces continuity. Buoyancy effects are included through a Boussinesq term in the momentum equations necessitating the solution of a temperature transport equation. Velocity-pressure decoupling that would normally occur with a collocated incompressible method is avoided through Rhie-Chow [11] interpolation and the Pressure-Implicit Splitting Operation algorithm [12] is used solve the equation set. The linear systems that arise when discretizing the implicit equations are solved using preconditioned iterative solvers.

Coriolis forces account for the Earth’s rotation. The lower surface boundary conditions based on Monin-Obuhkov [13] similarity theory is used, which is common practice in the atmospheric LES community. The upper boundary is a stress-free, rigid lid. First, a laterally periodic atmospheric boundary layer precursor simulation with no turbines is performed to generate the turbulent atmospheric boundary layer. Once that simulation has reached quasi-equilibrium, planes of inflow data are saved every time step. These data are then used as inflow boundary conditions for the non-periodic wind turbine simulation, and the downstream boundaries are outflow.

Sørensen and Shen’s [14] actuator line method is used to model the interaction of the wind turbine blades with the wind. The basic idea is that each blade is represented as a line, and each line is discretized into segments. For each segment, the blade airfoil type, twist, and chord are known. The velocity vector experienced by that segment can be sampled from the LES flow field giving the velocity magnitude and angle of attack. Airfoil lift and drag tables are then used to compute the force vector at each actuator line segment. The forces are then projected, using a three-dimensional Gaussian at each actuator line segment, onto the flow field as volumetric body forces that enter the momentum equation. Large-scale structures like the rotor wake and blade tip, root, and bound vortices are resolved.

FAST is two-way loosely coupled to the actuator line model. The LES model samples the velocity along the actuator line segments and returns those values to FAST. FAST, which normally computes those velocities using blade element momentum theory, operates instead in blade-element mode because the LES solver computes induction caused by the rotor. The blade forces computed with FAST are returned to the LES solver and imposed as the body forces described above.

Validation of the SOWFA tool is an ongoing process. In [15], SOWFA was used to simulate the 48-turbine Liligrund wind plant, and the results were then compared with field data, with good agreement throughout the first five rows. Additionally, [10] includes documentation of experiments testing SOWFA’s capability to simulate the inertial range in the turbulent energy spectra and the log-layer mean flow.
3 Simulation experiment

In this study, numerous simulations are run within SOWFA of a single turbine subject to the four proposed methods shown in fig. 1, which are applied in individual simulations with varying settings of yaw misalignment, tilt angle, or IPC moment set-point. The wind inflow is the same for all simulations. From the simulations, we extract the turbine’s average power over the simulation, as well as the metrics of loading for several components. From the flow, we use a correlation method to identify the wake-center at all locations downstream from the turbine. The results allow for a trade-off analysis of wake redirection potential vs. turbine effects.

Note that this study is limited to a single turbine and an examination of the flow behind it. However, an important consideration is the effect of changes made by an upstream turbine on a downstream turbine. Additionally, it is also important to learn if a reduction in the power output of the upstream turbine is compensated for by an increase in the power output of the downstream turbine. These issues are addressed in a related paper [16].

We simulate an NREL 5-MW baseline turbine [17] in turbulent inflow. The inflow, which is based on the study reported in [10], is that of a neutral atmospheric boundary layer. This inflow was selected because it had previously been validated and represents a realistic scenario. The inflow is generated in a precursor atmospheric LES on a domain that is 3 km by 3 km in the horizontal and 1 km in height. The horizontally averaged wind speed is driven to 8 m/s at the turbine hub height and is controlled through a time-varying mean driving pressure gradient. The wind comes from the southwest (300°) so that the elongated turbulent structures in the surface layer are not “trapped” by the periodic boundaries, continually cycling through in the same location. In the baseline case, the turbine rotor axis is aligned with the wind direction. The surface temperature flux is set to zero, although a capping inversion initially at 750 m above the surface is used both to slow boundary layer growth and because it is a real feature of atmospheric boundary layers. The surface aerodynamic roughness is set to 0.001 m, which is typical of flow over water. Details on the positioning of the turbine and meshing of the domain are given in fig. 2.

The yaw and tilt wake redirection strategies are tested for a range of settings. Each setting is tested in a simulation with 1,000s of simulated time. SOWFA requires significant computational power in order to run high-fidelity simulations: using a sample time of 0.02s, the time steps take an average 2.5s to calculate on the Sandia National Laboratories/NREL Red Mesa supercomputer [18], using distributed computation with 256 processors. This yields an execution time of 34.4h for each simulation.

In each case, the turbine uses the baseline controller defined in [17] for pitch and torque control. The IPC implementation is based on the design first presented in [19] using the parameters as specified in [20]. It is adapted so that IPC can be used in below-rated conditions and to induce an asymmetric moment, rather than remove one. Details on this IPC implementation are given in the Appendix.
Figure 2: Overview of the experimental setup in the baseline case.

- **Block mesh generation for OpenFOAM CFD solver**

<table>
<thead>
<tr>
<th>outer mesh:</th>
<th>number of cells</th>
<th>size of cell</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-direction</td>
<td>250</td>
<td>12m</td>
</tr>
<tr>
<td>y-direction</td>
<td>250</td>
<td>12m</td>
</tr>
<tr>
<td>z-direction</td>
<td>83</td>
<td>12.05m</td>
</tr>
</tbody>
</table>

  In a 2394 x 630 x 405 m box surrounding the turbine wake, the cell sizes are halved in x, y and z direction, and halved again in a 2142 x 376 x 279 m box, yielding:
  - cell size in the inner mesh: 3m x 3m x 3m
  - total number of cells: 14,921,616

- **B - NREL 5MW reference turbine**

  D=126.4m
4 Analysis and discussion

Following completion of the experiments, slices are extracted from the simulation outputs and a method is used to determine the mean wake center based on ideas of de Mare [21], which has been further developed at NREL. We take a horizontal slice of the mean velocity field at the turbine hub height and a vertical slice aligned with the mean wind and passing through the turbine centerline. From the horizontal slice, we take the mean velocity along lines within the slice plane and perpendicular to the flow at successive downstream locations. When plotted, the velocity along each line is a mean velocity profile. In the near wake, the velocity profiles are double-Gaussian in shape, and in the far wake, they resemble a normal Gaussian. The profiles at each downstream location are correlated with a Gaussian of similar width and depth. The point of maximum correlation is taken as the wake center position at each downstream location giving the lateral wake deflection. We follow the same process to find the vertical wake deflection using the vertical slice of mean velocity; however, we first subtract the vertical profile of mean velocity to remove the effect of vertical shear that is present in the atmospheric boundary layer. Fig. 3 shows the output of the wake center-line identification algorithm for several cases in the horizontal and vertical planes.

The cases shown in fig. 3 are representative of the collected results fully summarized in table 1. Wake center tracking results in the yaw and tilt simulations demonstrate significant displacements of wake center, in agreement with the previous literature. The IPC methods produce some redirection. While it does not achieve as much redirection as yaw or tilt angle adjustments, the skew is in some cases significant. Also noted, that while the intention of the IPC algorithms were to approximate yaw misalignment through an IPC-induced yaw moment, or tilt via a tilt moment, the results show that the largest vertical skew is given when a yaw moment is targeted and the largest horizontal skew for a high tilt moment.

In addition to measurements of the wake, data were collected from the FAST turbine output. The data included time series of output power, blade out-of-plane (OOP) bending moment, drivetrain torsion, tower fore-aft and side-side bending, and the yawing and tilting moments experienced at the yaw bearing. Using a root-sum-square combination, the separate tower and yaw moments are combined into a single moment. An average power output is computed, as well as the damage equivalent load (DEL) for each load signal. The DEL is a standard metric of fatigue damage; see [22] for an example implementation. These results are summarized in table 1. Note that the measurement of wake displacement is taken at 7 rotor diameters from the turbine, which is a typical location for a downstream turbine.

Reviewing table 1, there is a positive result for yaw-based wake skew. One can see that when the turbine yaws in the positive direction, wake redirection and load reduction for all components are simultaneously achieved for a number of operating points. Using a yaw misalignment to reduce turbine loads has been studied in the literature and these results are consistent with those findings. [23] There is a loss of power, however the intention is that this reduction should be compensated for by a larger gain in a downstream turbine. [24]
Table 1: Full results of experiment. Turbine wake redirection is summarized by the wake center 7 rotor diameters downstream from the turbine, bold indicates the larger offset.

<table>
<thead>
<tr>
<th>Amount</th>
<th>Horizontal Wake-offset abs(x/D)</th>
<th>Vertical Wake-offset abs(y/D)</th>
<th>Power</th>
<th>Blade OOP</th>
<th>Drivetrain</th>
<th>Tower</th>
<th>Yaw Bearing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.08</td>
<td>0.03</td>
<td>0.0%</td>
<td>+0.0%</td>
<td>+0.0%</td>
<td>+0.0%</td>
<td>+0.0%</td>
</tr>
<tr>
<td>-40°</td>
<td>0.33</td>
<td>0.02</td>
<td>-39.4%</td>
<td>+5.6%</td>
<td>-17.3%</td>
<td>-20.3%</td>
<td>-13.9%</td>
</tr>
<tr>
<td>-35°</td>
<td>0.32</td>
<td>0.03</td>
<td>-31.2%</td>
<td>+8.9%</td>
<td>-10.7%</td>
<td>-17.7%</td>
<td>-0.7%</td>
</tr>
<tr>
<td>-30°</td>
<td>0.27</td>
<td>0.00</td>
<td>-23.8%</td>
<td>+9.5%</td>
<td>-6.1%</td>
<td>-15.2%</td>
<td>-6.7%</td>
</tr>
<tr>
<td>-25°</td>
<td>0.24</td>
<td>0.02</td>
<td>-16.8%</td>
<td>+14.2%</td>
<td>-0.8%</td>
<td>-14.9%</td>
<td>-8.0%</td>
</tr>
<tr>
<td>-20°</td>
<td>0.17</td>
<td>0.02</td>
<td>-11.2%</td>
<td>+11.7%</td>
<td>+2.3%</td>
<td>-7.7%</td>
<td>-9.8%</td>
</tr>
<tr>
<td>-15°</td>
<td>0.10</td>
<td>0.02</td>
<td>-6.0%</td>
<td>+7.2%</td>
<td>+1.5%</td>
<td>-10.5%</td>
<td>-3.5%</td>
</tr>
<tr>
<td>-10°</td>
<td>0.05</td>
<td>0.02</td>
<td>-2.6%</td>
<td>+8.6%</td>
<td>+1.0%</td>
<td>-10.3%</td>
<td>-2.8%</td>
</tr>
<tr>
<td>-5°</td>
<td>0.03</td>
<td>0.02</td>
<td>-0.4%</td>
<td>+5.3%</td>
<td>-5.9%</td>
<td>-3.1%</td>
<td></td>
</tr>
<tr>
<td>5°</td>
<td>0.17</td>
<td>0.05</td>
<td>-0.3%</td>
<td>-4.0%</td>
<td>-1.6%</td>
<td>-0.1%</td>
<td>-5.1%</td>
</tr>
<tr>
<td>10°</td>
<td>0.24</td>
<td>0.06</td>
<td>-2.6%</td>
<td>-5.2%</td>
<td>+1.3%</td>
<td>+0.1%</td>
<td>-9.7%</td>
</tr>
<tr>
<td>15°</td>
<td>0.30</td>
<td>0.06</td>
<td>-5.7%</td>
<td>-9.9%</td>
<td>-0.4%</td>
<td>-3.2%</td>
<td>-11.1%</td>
</tr>
<tr>
<td>20°</td>
<td>0.35</td>
<td>0.05</td>
<td>-10.6%</td>
<td>-13.3%</td>
<td>-1.0%</td>
<td>-6.0%</td>
<td>-14.4%</td>
</tr>
<tr>
<td>25°</td>
<td>0.43</td>
<td>0.08</td>
<td>-16.2%</td>
<td>-14.4%</td>
<td>-6.2%</td>
<td>-7.4%</td>
<td>-14.8%</td>
</tr>
<tr>
<td>30°</td>
<td>0.49</td>
<td>0.05</td>
<td>-23.3%</td>
<td>-17.1%</td>
<td>-10.2%</td>
<td>-12.7%</td>
<td>-8.1%</td>
</tr>
<tr>
<td>35°</td>
<td>0.51</td>
<td>0.05</td>
<td>-31.3%</td>
<td>-20.1%</td>
<td>-14.9%</td>
<td>-7.9%</td>
<td>-13.5%</td>
</tr>
<tr>
<td>40°</td>
<td>0.54</td>
<td>0.10</td>
<td>-40.1%</td>
<td>-24.4%</td>
<td>-22.2%</td>
<td>-7.1%</td>
<td>-1.5%</td>
</tr>
<tr>
<td>-15°</td>
<td>0.05</td>
<td>0.14</td>
<td>-6.0%</td>
<td>-14.7%</td>
<td>+10.4%</td>
<td>-2.0%</td>
<td>-8.6%</td>
</tr>
<tr>
<td>-12°</td>
<td>0.10</td>
<td>0.14</td>
<td>-3.6%</td>
<td>-12.2%</td>
<td>+7.8%</td>
<td>-3.6%</td>
<td>-7.7%</td>
</tr>
<tr>
<td>-9°</td>
<td>0.08</td>
<td>0.09</td>
<td>-1.9%</td>
<td>-10.0%</td>
<td>+5.0%</td>
<td>-2.0%</td>
<td>+2.5%</td>
</tr>
<tr>
<td>-3°</td>
<td>0.10</td>
<td>0.02</td>
<td>0.1%</td>
<td>-0.6%</td>
<td>+0.7%</td>
<td>+5.5%</td>
<td>-3.2%</td>
</tr>
<tr>
<td>3°</td>
<td>0.11</td>
<td>0.10</td>
<td>-0.3%</td>
<td>+1.3%</td>
<td>+0.6%</td>
<td>-1.8%</td>
<td>-1.3%</td>
</tr>
<tr>
<td>6°</td>
<td>0.11</td>
<td>0.10</td>
<td>-0.5%</td>
<td>+4.1%</td>
<td>-2.0%</td>
<td>-3.3%</td>
<td>-4.9%</td>
</tr>
<tr>
<td>9°</td>
<td>0.13</td>
<td>0.13</td>
<td>-1.6%</td>
<td>+8.3%</td>
<td>-2.0%</td>
<td>-4.8%</td>
<td>-3.1%</td>
</tr>
<tr>
<td>12°</td>
<td>0.14</td>
<td>0.16</td>
<td>-3.2%</td>
<td>+5.8%</td>
<td>-2.8%</td>
<td>+3.0%</td>
<td>-0.9%</td>
</tr>
<tr>
<td>15°</td>
<td>0.11</td>
<td>0.18</td>
<td>-5.4%</td>
<td>+7.0%</td>
<td>-3.9%</td>
<td>-3.4%</td>
<td>+4.1%</td>
</tr>
<tr>
<td>IPC-Yaw Max</td>
<td>0.05</td>
<td>0.08</td>
<td>-9.2%</td>
<td>+138.1%</td>
<td>+1.7%</td>
<td>+5.9%</td>
<td>+33.2%</td>
</tr>
<tr>
<td>IPC-Yaw Min</td>
<td>0.14</td>
<td>0.08</td>
<td>-15.2%</td>
<td>+64.7%</td>
<td>+6.2%</td>
<td>+5.2%</td>
<td>+22.4%</td>
</tr>
<tr>
<td>IPC-Tilt Max</td>
<td>0.17</td>
<td>0.03</td>
<td>-11.8%</td>
<td>+91.0%</td>
<td>+1.4%</td>
<td>+10.0%</td>
<td>+26.5%</td>
</tr>
<tr>
<td>IPC-Tilt Min</td>
<td>0.05</td>
<td>0.06</td>
<td>-13.2%</td>
<td>+130.4%</td>
<td>+89.3%</td>
<td>+1.9%</td>
<td>+34.2%</td>
</tr>
</tbody>
</table>
Tilt similarly demonstrates potential for wake redirection with, mostly, positive load impacts. Observing the loads in the 12 degree case, all are reduced with the exception of blade bending, which has gone up 10%. It should be pointed out that currently there is no means of modifying tilt angle in the field. However, the effect of causing the higher-speed higher-altitude winds to be pulled downward might be rewarding enough to justify further investigation, although given that positive tilt angles would cause the blades to come closer to the tower on upwind turbines, this is appropriate more for downwind machines. [25]

The results for the IPC-based methods are mixed. Significant wake skew is achieved for some cases, however, because the method is maximizing an asymmetric rotor moment, the blade loads are substantially increased. This leads to the notion that while it may be possible to achieve wake redirection with IPC, this particular IPC algorithm is not a good method. Finding an IPC controller which achieves wake skew with reduced blade loads would be very useful because
IPC is already possible to implement on many existing turbines (unlike changes to tilt), and can be adjusted more quickly than yaw angle.

In considering the results, the authors now believe the initial concept for IPC-based wake redirection followed in this paper was fundamentally flawed. Specifically, while IPC can reproduce the rotor moments generated by yaw or tilt misalignment, this moment is not what can create skew. In the left of fig. 4, the conceptualization of yaw-misalignment induced wake redirection from [5] is redrawn. In it, the thrust force of the turbine is shown to act along the axis of the rotor shaft. When the wind inflow is at an angle to this direction, the thrust can be divided into components $f_x$ and $f_y$. $f_x$ is parallel to the flow and slows the wind, while $f_y$ is perpendicular and applies the force which causes wake redirection. A moment produced in IPC is based on an uneven plane of thrust force. This yields a moment from the turbine’s perspective, but no perpendicular force on the flow. However, IPC can cause a perpendicular net force on the flow. Observing the right section of fig. 4, when the blade torque through a rotor rotation is uneven (in the sense that rightward torque is not matched by leftward torque), a skew can result. This is because the reaction force on the flow is now also unbalanced, and a net perpendicular force is applied resulting in wake skew. Notice that the IPC configuration draw in fig. 4 will yield a tilt moment on the turbine (because the blade thrust is most different between the top and bottom azimuth positions) and a horizontal wake skew (because the flow reaction forces are most different in the horizontal force directions), which agrees with the results presented earlier.

This analysis hopefully indicates that while the IPC algorithm first developed, which seeks to
apply skew by maximizing yaw or tilt moment, is problematic, an alternative implementation may be possible. This implementation, the subject of future work, should attempt to maximize the wake skew through torque imbalance while minimizing rotor moments.

5 Conclusions

In this paper, the NREL wind plant simulation tool, SOWFA, was used to simulate and investigate several methods for wake redirection. Wake redirection is one proposed method for improving wind plant overall performance. For yaw misalignment, simulations showed significant redirection effects coupled with reductions in loading across measured components, a positive result. Tilt angle adjustment was shown to also achieve wake redirection while reducing all turbine loads except for blade OOP bending. Although modifying tilt angle is not currently a controllable feature of wind turbines, knowledge of the capability of this effect might be useful, especially if the effect of pulling in faster wind yields greater overall power gains.

IPC-based methods also demonstrated an ability to affect wake skew, however this was achieved with a substantial increase in blade loading. Analysis presented indicates that the IPC algorithm employed in this paper, while enough to prove the concept that IPC can redirect wake, it too simplistic for actual implementation and future work will focus on the determination of more optimal designs.

6 Acknowledgements

The authors are very grateful to Wesley Jones and the NREL High Performance Computing team for their crucial help and support in completing this simulation study. This work was supported by the U.S. Department of Energy under Contract No. DE-AC36-08GO28308 with the National Renewable Energy Laboratory.

References


M. de Maré. personal communication, 2012.

M. Buhl Jr. MCrunch theory manual for version 1.00.


**Appendix**

**Appendix A: Implementation of IPC with induced yaw or tilt moments**

This section explains how individual pitch control (IPC) was implemented to allow yaw and tilt moments to be induced by the IPC. The implementation also can be used in below-rated operation with varying rotor speeds. Let \( \varphi \) denote the rotor speed in rad/s, let \( \{ M_{y,i} \}_{i=1}^3 \) denote blade root vibrations of each of the three blades, let \( M_{r,yaw}, M_{r,tilt} \) denote setpoints for the induced yaw and tilt moments, and let \( s \) denote the Laplace operator. Then the 1P and 2P IPC additive adjustments to the pitch, \( \{ \delta \theta_{1,jP} \}_{i=1}^3 \), are given by:

\[
\begin{bmatrix}
\delta \theta_{1,jP} \\
\delta \theta_{2,jP} \\
\delta \theta_{3,jP}
\end{bmatrix} = \mathcal{L}(s) I_{3 \times 3} P_{jP}(\varphi + \delta jP) \begin{bmatrix}
\frac{K_{jP,yaw}}{s} \\
0 \\
\frac{K_{jP,tilt}}{s}
\end{bmatrix}
\]

\[
\times \left( \frac{2}{\pi} P_{jP}(\varphi) N_{jP}(s) I_{3 \times 3} \begin{bmatrix}
M_{y,1} \\
M_{y,2} \\
M_{y,3}
\end{bmatrix} - \begin{bmatrix}
M_{r,yaw} \\
M_{r,tilt}
\end{bmatrix} \right)
\]

for \( j = 1, 2 \), with Coleman transformation matrices:

\[
P_{jP}(\varphi) = \begin{bmatrix}
\cos(j \varphi) & \sin(j \varphi) \\
\cos(j (\varphi + 2\pi/3)) & \sin(j (\varphi + 2\pi/3)) \\
\cos(j (\varphi + 4\pi/3)) & \sin(j (\varphi + 4\pi/3))
\end{bmatrix},
\]

and with inverse notch filters \( N_{jP} \), and low-pass filter \( \mathcal{L} \):

\[
N_{jP}(s) = K_{jP} \frac{2 \zeta_{jP} \omega_{jP} s}{s^2 + 2 \zeta_{jP} \omega_{jP} s + \omega_{jP}^2}, \quad \mathcal{L}(s) = \frac{\omega_L^2}{s^2 + 2 \zeta_L \omega_L s + \omega_L^2},
\]

with \( \omega_{jP} = j \varphi \), and parameters \( K_i, \zeta_i, \omega_L, \delta_{jP} \) as specified in [20]. The filters are used in a Tustin discretized form with a sample time of 0.02s. The pitch angles are saturated to a 5 degree amplitude, and the pitch rates are limited to 8 deg/s. In the IPC induced moment test cases, \( M_{r,yaw} \) or \( M_{r,tilt} \) are chosen large enough such that the pitch angles vary with maximum amplitude, in order to find the maximum effect of IPC action on the wake.
High-fidelity simulation comparison of wake mitigation control strategies for a two-turbine case

P. Fleming¹, P. Gebraad², S. Lee¹, J.W. van Wingerden², K. Johnson¹,
M. Churchfield¹, J. Michalakes¹, P. Spalart³, P. Moriarty¹

¹National Renewable Energy Laboratory, Golden CO, USA, paul.fleming@nrel.gov
²Delft University of Technology, Delft, The Netherlands, J.W.vanWingerden@TUDelft.nl
³Boeing Commercial Airplanes, Seattle WA, USA

Abstract
Wind turbines arranged in a wind plant impact each other through their wakes. Wind plant control is an active research field that attempts to improve wind plant performance by modifying individual turbine controllers to take into account these turbine-wake interactions. In this paper, we use high-fidelity simulations of a two-turbine fully-waked scenario to investigate the potential of several wake mitigation strategies, including modification of yaw and tilt angle of an upstream turbine to induce wake skew, as well as repositioning of the downstream turbine. The simulation results are compared through change relative to a baseline operation in terms of overall power capture and loading on the upstream and downstream turbine. Results demonstrate improved power production for all methods while analysis of control options, including individual pitch control, show potential to minimize the increase of, or reduce, turbine loads.

1 Introduction
Wind turbines influence nearby turbines aerodynamically as they extract energy from and enhance turbulence in the wind. Recently, wind turbine control systems researchers have been examining how to compensate for these influences so that wind power plant efficiency can be increased and turbine loads can be reduced. In the literature, such compensation is often achieved by adapting the axial induction of individual turbines through pitch and torque control [1, 2, 3]. In these methods, an axial induction factor is found that results in lower individual turbine power for an upstream tower, but higher total plant power because of increased power capture by downstream turbines.

An alternative approach to wind plant control focuses on redirecting wakes around downstream turbines. One published method of achieving this redirection is through intentional yaw misalignment in an upstream turbine. In [4], field-tests are carried out in a scaled wind plant test field, whereas in [5], computational fluid-dynamics (CFD) simulations of a wind turbine are performed. In a related fashion, [6] investigates the ability to redirect the wake vertically using...
adjustments of the tilt angle, by performing CFD simulations of laminar flow. In both [5] and [6], the CFD simulations use actuator disk models of the wind turbines. These papers indicate good potential for the wake re-direction methods.

In this paper, we describe two-turbine simulation experiments of wake-redirection-based wind plant control using a high-fidelity wind plant simulator, the NREL SOWFA (Simulator for Off/Onshore Wind Farm Applications) tool [7]. SOWFA, described in Section 2, couples a CFD solver with the aero-elastic turbine simulator FAST [8]. Using SOWFA, controllers can be compared in terms of their effects on power production and loading of upstream and downstream turbines. This is an important capability because a controller’s effects on power production must be weighed against its effects on turbine loads. Additionally, the tool allows for simulations in sheared and turbulent inflow.

In addition to evaluating the yaw-based and tilt-based redirection methods, we investigate an additional concept for wake avoidance. This method repositions the downstream turbine, which is assumed to be floating and therefore repositionable, as proposed in [9]. Additional analysis of the yaw and tilt methods can be found in [10], which analyzes the ability of a single turbine to redirect the wake.

The control methods are evaluated using a simulation of two 5-MW turbines, 7 rotor diameters apart, aligned in a turbulent inflow. For each method, the setpoint (yaw misalignment angle, tilt angle or downstream turbine position) is varied, and a 1000-second simulation is run. Additionally, because these techniques might move the downstream turbine from full wake to partial wake, which induces loading [11] due to uneven wind speed distribution across the rotor plane, we consider the cases where the downstream turbine is and isn’t using independent pitch control (IPC) to mitigate these loads. The results are compared in terms of total power output and key component loads on both the upstream and downstream turbine across all cases.

The contributions of the paper are: (1) an analysis and comparison of three methods of wind-plant wake mitigation using a high-fidelity simulator in terms of power and loading of an upstream and downstream turbine, and (2) an investigation into the use of IPC for mitigating the effects of partial wake on turbine loads.

The remainder of this paper is organized as follows. Section 2 provides an overview of the simulation tool SOWFA used in this work. Section 3 provides a description of the simulation scenario in terms of dimension, inflow properties, turbine properties, and individual control laws. Section 4 presents and analyzes the results of the study. Section 5 provides some discussion points and considerations for future work. Finally, the conclusions are given in section 6.

2 SOWFA

In this work, a parametric study of the proposed control actuation methods is performed using a high-fidelity tool: Simulator for Off/Onshore Wind Farm Applications (SOWFA) [7], which is
a large-eddy simulation (LES) framework coupled with NRELs aero-elastic turbine code FAST [8] for studying wind turbines embedded in the atmospheric boundary layer (ABL). The kernel of the LES framework is based on open-source OpenFOAM libraries [12], which solves the incompressible Navier-Stokes equations that are augmented with a buoyancy term (based on Boussinesq approximation) and Coriolis acceleration to simulate atmospheric boundary layers under various conditions. The transport of potential temperature is solved in parallel to account for buoyancy. This set of governing equations are discretized over an unstructured finite volume mesh with a second-order central differencing scheme. The collocated formulation of the velocity and pressure variables is decoupled using the Rhie-Chow [13] interpolation method to avoid numerical instability. The time advancement method follows the predictor-corrector pressure-implicit splitting operation (PISO) of Issa [14] with three sub-step iterations to maintain second-order accuracy. The Moeng model [15] is adapted to estimate the local time-varying shear stress. The sub-grid scale (SGS) turbulence closure employs the standard Smagorinsky [16] formulation with a constant of 0.135.

The turbine blades are represented by the actuator line (AL) method of Sørensen and Shen [17], in which the blades are discretized along the radial line where the lift and drag forces are computed based on the incoming flow and the airfoil geometry at the actuator points. These force vectors are projected on to the computational domain space using a three-dimensional Gaussian filter which, as a collective whole, produces the wake structures similar to those from blade-geometry resolved simulations at significantly reduced computational cost. In SOWFA, the incoming flow velocity data at the actuator points from the flow solver are fed into FAST, from which the computed aerodynamic lift and drag forces and the shifted actuator points (caused by the blade deflections) are projected on to the momentum equation as a body force term completing the two-way coupling cycle. The structural loading responses induced by the aerodynamic forces are collected as FAST outputs, which are later presented in this study. Further details on SOWFA can be found in [18].

With SOWFA, simulations of proposed wind plant control schemes can be analyzed. Because the simulation includes high-fidelity modeling of the atmosphere and the turbine structure, it is possible to study simultaneously a controller’s impact on power and turbine loads.

3 Simulation setup

As described in the introduction, the objective of this paper is to compare three strategies of wake mitigation using SOWFA (yaw misalignment, tilt misalignment, and repositioning of the downstream turbine). To this end, an “open-loop” two-turbine simulation study is performed in which the yaw, tilt, and position setpoints are swept and held fixed for separate 1000-second simulations with constant wind direction. Later research will develop active closed-loop control strategies for time-varying wind directions.

A scenario was developed to simulate two NREL 5-MW baseline turbines [19] in turbulent inflow. The turbulent scenario is that of a neutral boundary layer, which was selected based on a
Figure 1: Overview of the simulation setup in the baseline case.
previously published study [18]. The inflow is generated in a precursor atmospheric LES on a
domain that is 3 km by 3 km in the horizontal and 1 km in height. The horizontally averaged
wind speed is driven to 8 m/s at the turbine hub height and is controlled through a time-varying
mean-driving pressure gradient. The wind comes from the southwest (300°) so that the elongated
turbulent structures in the surface layer are not "trapped" by the periodic boundaries, continually
cycling through in the same location. In the baseline case, the turbine rotor axis is aligned with
the wind direction. The surface temperature flux is set to zero, although a capping inversion
initially at 750 m above the surface is used both to slow boundary layer growth and because it
is a real feature of atmospheric boundary layers. The surface aerodynamic roughness is set to
0.001 m, which is typical for flow over water. Details on positioning of the turbines and meshing
of the domain are given in fig. 1.

Screenshots from time averaged slices of the flow for the different control strategies are provided
in fig. 2. SOWFA requires significant computational power to run high-fidelity simulations. Us-
ing a sample time of 0.02s, the time steps take an average 2.5 s to calculate on the Sandia/NREL
Red Mesa supercomputer [20] using distributed computation with 256 processors. This yields
an execution time of 34.4 h for each simulation.

In each case, the turbines use the baseline pitch and torque controllers defined in [19]) independ-
dently. Because wake redirection can move the downstream turbine from full wake to partial
wake, inducing loads [11], we compare the use of IPC with the standard collective pitch con-
trol by switching on load-reducing IPC on the downstream turbine for the last 400 s of the simulation. The IPC implementation is based on the design first presented in [21], using the parameters as specified in [22], with some adaptations to be able to use the load-reducing IPC in below-rated conditions. A supervisory wind plant controller [23] collects the data from the individual turbines.

4 Results and Analysis

Following completion of the runs, the data were collected from each case and post-processed as follows. First, the 1000 s of time domain data for each were broken into segments. The first 200 s of each run were discarded because the wake was not fully developed. The last 100 s were also discarded because of system problems that left some files incomplete. Finally, the remaining time histories were divided into a segment from 200 s to 600 s, in which the downstream turbine is not running IPC, and a segment from 700 s to 900 s, when it is and the IPC startup transients have vanished. Although it should be possible to start IPC smoothly, because the transition was not our research focus, we start the controller rather abruptly. In the baseline case, IPC is never enabled, to provide a basis for comparison.

From these two blocks of time (200-600 s and 700-900 s), several metrics are computed. First, the average power is computed for each turbine. Next, loads are computed for blade out-of-plane (OOP) bending, drivetrain torsion, tower bending and yaw bearing moment. In the case of the tower load, a combined load is computed from the separate fore-aft and side-side loads using a root-sum-square combination. This is likewise done to combine the separate $M_y$ and $M_z$ loads on the yaw bearing. Individual loads are provided in the appendix. For each of these load signals, a damage equivalent load (DEL) is computed. The DEL is a standard metric of fatigue damage (see [24]). These results are summarized in figs. 3 and 4.

Fig. 3 shows the comparison of methods in the case where the downstream turbine does not use IPC. For each method, there is a sweep across possible setpoints (yaw angle, tilt angle or reposition of downstream turbine position). The top row shows the total power output of each case, with the horizontal line indicating the baseline level, and the numbers above each bar denoting percent change from the baseline case. The remaining rows of the figure indicate percent change in DEL for the components examined.

Starting with yaw-based control, in the best case, the method shows an increase in power of 4.6%. Additionally, the simulations show that, depending on the yaw direction, the blade bending is either increased or decreased in the upstream turbine. This is in agreement with other published studies on this behavior in sheared inflow (see [25] for a detailed investigation into this effect.) The other loads experience smaller variations, with the yaw bearing load showing a similar yaw angle dependence. The downstream turbine, however, experiences a rise in blade OOP bending, drivetrain torsion, and yaw bearing moment. This change is most likely due to the movement from full to partial wake overlap.
Figure 3: Summary of results of two-turbine simulation. The three columns are divided by control action. The top row shows the combined power output for each case, compared to the baseline case on the far left. The remaining rows indicate the percent change in load compared to the baseline.
In the tilt case, a maximum power gain of 7.1% is observed. At this peak of power capture, it is seen that for the upstream turbine, the blade and yaw bearing loads have gone up while the drivetrain and tower loads have declined a small amount. As in the yaw case, the downstream turbine experiences an increase in blade loads, most likely due to partial-wake overlap. There is also a decrease in tower loads and increase in yaw bearing loads.

Tilt misalignment shows larger potential power production improvements than yaw when considering large positive tilt angles. With a positive tilt angle, the rotor would face downwards, and for conventional upwind turbine designs this would cause the blades to hit the tower. Therefore, a positive tilting mechanism is more suitable to downwind-facing turbines [26]. Both negative and positive tilt angles will redirect the wake away from the downstream turbine rotor, but the positive tilting has the advantage that it will redirect the wake towards the ground, allowing high velocity air from higher altitudes to flow towards the upper part of the downstream rotor, resulting in higher power production of the downstream turbine. Additional details on the wake displacements that can be achieved using yawing or tilting can be found in [10].

Repositioning of (floating) turbines produces the most substantial gains in power if the rotor of the downstream turbine is moved more than 25 m out of the rotor axis of the upstream turbine. (In this case there is up to 41% improvement when the downstream turbine is moved a full rotor diameter.) Observing the loads for the downstream turbine, there is little change for small alterations in position, significant change for the displacements yielding partial overlap, and then no change again when the turbine is moved a full rotor diameter.

Looking at the loads across experiments, the upstream turbine either experiences an increase or decrease in blade OOP bending, depending on the angle chosen. Also, for the upstream turbine, yaw- and tilt-angle adjustments either decrease or minimally increase the drivetrain, tower, and yaw load. A possible explanation for this effect is that these methods generally reduce the power capture of the upstream turbine, and derating can be considered a load mitigation strategy. For the downstream turbine, all loads generally increase somewhat, and this is most likely due to moving from full to partial wake overlap.

Fig. 4 performs the same analysis, but now for the case where the downstream turbine is using IPC to mitigate the effect of partial wake overlap. Note that these results are based on 200 s of simulation versus 400 s in Fig. 3, and are from a different point in the simulation. The results are dramatic: the blade loads, tower loads, and yaw bearing loads are consistently reduced when compared to the baseline case (which does not use IPC). The drivetrain loads are the exception, but the lack of a clear pattern indicates that perhaps this is a somewhat stochastic load. Missing from the current controller is a drivetrain damper, a very common element in industrial controllers that could be used to minimize the changes in drivetrain loads. Overall, the results indicate a very strong motivation for the use of IPC, in general, and as a way to eliminate the negative impacts of using wake-mitigation strategies.

The appendix provides the full results listed in absolute values (not relative to baseline).
Figure 4: Comparison of data as in fig. 3, except with the downstream turbine operating with IPC in all cases except for the baseline.
5 Discussion

It is important to acknowledge the shortcomings of this current work. First, because of computational/time constraints, results are based on simulations of one inflow case. Future work will simulate multiple wind inputs, with varied atmospheric conditions, which will better establish the robustness of these results. Additionally, the experiments are not of a closed-loop control, but of settings that take advantage of a stationary wind direction. This is a good first step, but to be applicable to real wind plants, future work must establish control loops that can achieve similar results for changing wind directions. Finally, future work will include simulations of greater number of turbines to analyze the use of these techniques in a larger wind plant.

6 Conclusions

First, this report shows very good potential for all methods considered. For each case, there are operating points that couple improved power capture with, mostly, reduced loading. It is important to point out though, that existing technology can only implement yaw misalignment. However, given that tilt and repositioning are capable of yielding more power capture, perhaps the effects could be considered in the design of future turbines. Second, we show that there is very good potential for employing IPC to mitigate partial wake effects. This improves the benefit of the wake redirection or repositioning techniques by reducing the partial-wake-induced loads on the downstream turbine.

Acknowledgements

The authors are very grateful to Wesley Jones and the NREL High Performance Computing team for their crucial help and support in completing this simulation study. This work was supported by the U.S. Department of Energy under Contract No. DE-AC36-08GO28308 with the National Renewable Energy Laboratory and by the NWO Veni Grant no. 11930 “Reconfigurable floating wind farm”.

References


<table>
<thead>
<tr>
<th>Case</th>
<th>Power (MW)</th>
<th>OOP Bending (kN/m)</th>
<th>Drive train (kN/m)</th>
<th>Fore-Arm (kN/m)</th>
<th>Side-Side (kN/m)</th>
<th>Yaw Mx (kN/m)</th>
<th>Yaw My (kN/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 yaw = -30</td>
<td>1.3</td>
<td>1.1</td>
<td>2.6</td>
<td>738.8</td>
<td>982.5</td>
<td>252.9</td>
<td>5595.1</td>
</tr>
<tr>
<td>T1 yaw = -25</td>
<td>1.4</td>
<td>1.0</td>
<td>2.5</td>
<td>867.6</td>
<td>1124.8</td>
<td>271.6</td>
<td>5672.2</td>
</tr>
<tr>
<td>T1 yaw = 0</td>
<td>1.5</td>
<td>0.9</td>
<td>2.5</td>
<td>735.3</td>
<td>1018.6</td>
<td>219.3</td>
<td>5346.8</td>
</tr>
<tr>
<td>T1 yaw = 10</td>
<td>1.6</td>
<td>0.8</td>
<td>2.5</td>
<td>798.9</td>
<td>981.1</td>
<td>185.4</td>
<td>5489.2</td>
</tr>
<tr>
<td>T1 yaw = 5</td>
<td>1.7</td>
<td>0.8</td>
<td>2.5</td>
<td>811.0</td>
<td>1018.0</td>
<td>169.4</td>
<td>5582.1</td>
</tr>
<tr>
<td>T1 yaw = 0</td>
<td>1.4</td>
<td>1.0</td>
<td>2.5</td>
<td>705.8</td>
<td>1042.1</td>
<td>151.6</td>
<td>5425.9</td>
</tr>
<tr>
<td>T1 yaw = 10</td>
<td>1.7</td>
<td>1.0</td>
<td>2.6</td>
<td>678.6</td>
<td>1044.4</td>
<td>136.8</td>
<td>5346.8</td>
</tr>
<tr>
<td>T1 yaw = 15</td>
<td>1.6</td>
<td>1.0</td>
<td>2.6</td>
<td>639.2</td>
<td>1055.9</td>
<td>122.6</td>
<td>5189.3</td>
</tr>
<tr>
<td>T1 yaw = 20</td>
<td>1.5</td>
<td>1.1</td>
<td>2.7</td>
<td>626.1</td>
<td>1132.7</td>
<td>109.8</td>
<td>5029.3</td>
</tr>
<tr>
<td>T1 yaw = 25</td>
<td>1.4</td>
<td>1.2</td>
<td>2.8</td>
<td>608.5</td>
<td>1131.6</td>
<td>97.1</td>
<td>4877.9</td>
</tr>
<tr>
<td>T1 yaw = 30</td>
<td>1.3</td>
<td>1.3</td>
<td>2.9</td>
<td>607.6</td>
<td>1130.4</td>
<td>85.1</td>
<td>4732.9</td>
</tr>
<tr>
<td>T1 yaw = 35</td>
<td>1.2</td>
<td>1.4</td>
<td>2.6</td>
<td>609.6</td>
<td>1080.7</td>
<td>73.4</td>
<td>4601.1</td>
</tr>
<tr>
<td>T1 yaw = 40</td>
<td>1.0</td>
<td>1.5</td>
<td>2.5</td>
<td>598.9</td>
<td>1057.9</td>
<td>61.6</td>
<td>4472.3</td>
</tr>
<tr>
<td>T1 tilt = -15</td>
<td>1.8</td>
<td>0.9</td>
<td>2.5</td>
<td>862.2</td>
<td>906.4</td>
<td>25.1</td>
<td>5579.6</td>
</tr>
<tr>
<td>T1 tilt = -12</td>
<td>1.6</td>
<td>0.9</td>
<td>2.5</td>
<td>645.5</td>
<td>916.5</td>
<td>21.3</td>
<td>5469.3</td>
</tr>
<tr>
<td>T1 tilt = -9</td>
<td>1.7</td>
<td>0.9</td>
<td>2.5</td>
<td>657.5</td>
<td>967.0</td>
<td>17.3</td>
<td>5496.8</td>
</tr>
<tr>
<td>T1 tilt = -6</td>
<td>1.7</td>
<td>0.9</td>
<td>2.5</td>
<td>695.4</td>
<td>970.6</td>
<td>13.4</td>
<td>5449.0</td>
</tr>
<tr>
<td>T1 tilt = -3</td>
<td>1.7</td>
<td>0.8</td>
<td>2.5</td>
<td>722.1</td>
<td>979.9</td>
<td>9.5</td>
<td>5392.3</td>
</tr>
<tr>
<td>T1 tilt = 0</td>
<td>1.7</td>
<td>0.9</td>
<td>2.6</td>
<td>760.9</td>
<td>1032.6</td>
<td>5.6</td>
<td>5330.9</td>
</tr>
<tr>
<td>T1 tilt = 3</td>
<td>1.7</td>
<td>0.9</td>
<td>2.6</td>
<td>783.4</td>
<td>1043.4</td>
<td>1.7</td>
<td>5269.5</td>
</tr>
<tr>
<td>T1 tilt = 6</td>
<td>1.7</td>
<td>0.9</td>
<td>2.6</td>
<td>803.6</td>
<td>1137.3</td>
<td>-2.3</td>
<td>5209.2</td>
</tr>
<tr>
<td>T1 tilt = 9</td>
<td>1.7</td>
<td>0.9</td>
<td>2.6</td>
<td>831.7</td>
<td>1232.7</td>
<td>-8.3</td>
<td>5150.5</td>
</tr>
<tr>
<td>T1 tilt = 12</td>
<td>1.7</td>
<td>1.0</td>
<td>2.6</td>
<td>827.2</td>
<td>1292.1</td>
<td>-14.3</td>
<td>5093.8</td>
</tr>
<tr>
<td>T1 tilt = 15</td>
<td>1.7</td>
<td>1.1</td>
<td>2.7</td>
<td>822.8</td>
<td>1265.1</td>
<td>-20.3</td>
<td>5037.1</td>
</tr>
<tr>
<td>T1 tilt = 18</td>
<td>1.7</td>
<td>1.2</td>
<td>2.7</td>
<td>838.8</td>
<td>1282.3</td>
<td>-26.3</td>
<td>4980.4</td>
</tr>
<tr>
<td>T1 tilt = 21</td>
<td>1.7</td>
<td>1.2</td>
<td>2.7</td>
<td>854.3</td>
<td>1287.7</td>
<td>-32.3</td>
<td>4923.8</td>
</tr>
<tr>
<td>T1 tilt = 24</td>
<td>1.7</td>
<td>1.3</td>
<td>2.7</td>
<td>812.7</td>
<td>1270.2</td>
<td>-38.3</td>
<td>4867.1</td>
</tr>
<tr>
<td>T1 tilt = 27</td>
<td>1.3</td>
<td>1.4</td>
<td>2.7</td>
<td>811.3</td>
<td>1222.6</td>
<td>-44.3</td>
<td>4810.4</td>
</tr>
<tr>
<td>T1 tilt = 30</td>
<td>1.3</td>
<td>1.5</td>
<td>2.7</td>
<td>785.1</td>
<td>1212.3</td>
<td>-50.4</td>
<td>4753.7</td>
</tr>
<tr>
<td>T1 tilt = 33</td>
<td>1.3</td>
<td>1.5</td>
<td>2.7</td>
<td>784.9</td>
<td>1209.4</td>
<td>-56.5</td>
<td>4697.0</td>
</tr>
<tr>
<td>T1 tilt = 36</td>
<td>1.3</td>
<td>1.6</td>
<td>2.7</td>
<td>784.9</td>
<td>1206.5</td>
<td>-62.6</td>
<td>4640.3</td>
</tr>
<tr>
<td>T2 moved -100m</td>
<td>1.7</td>
<td>1.5</td>
<td>2.5</td>
<td>739.5</td>
<td>1125.1</td>
<td>41.7</td>
<td>5718.7</td>
</tr>
<tr>
<td>T2 moved -50m</td>
<td>1.7</td>
<td>1.5</td>
<td>2.5</td>
<td>739.5</td>
<td>1125.1</td>
<td>41.7</td>
<td>5718.7</td>
</tr>
<tr>
<td>T2 moved 0m</td>
<td>1.7</td>
<td>1.5</td>
<td>2.5</td>
<td>740.6</td>
<td>1002.0</td>
<td>38.8</td>
<td>5661.8</td>
</tr>
<tr>
<td>T2 moved 5m</td>
<td>1.7</td>
<td>1.6</td>
<td>2.5</td>
<td>735.7</td>
<td>948.1</td>
<td>35.9</td>
<td>5604.0</td>
</tr>
<tr>
<td>T2 moved 10m</td>
<td>1.7</td>
<td>1.7</td>
<td>2.5</td>
<td>738.7</td>
<td>991.0</td>
<td>33.1</td>
<td>5546.2</td>
</tr>
<tr>
<td>T2 moved 15m</td>
<td>1.7</td>
<td>1.8</td>
<td>2.5</td>
<td>738.2</td>
<td>982.3</td>
<td>30.3</td>
<td>5488.4</td>
</tr>
<tr>
<td>T2 moved 20m</td>
<td>1.7</td>
<td>1.9</td>
<td>2.5</td>
<td>738.2</td>
<td>986.5</td>
<td>27.5</td>
<td>5430.6</td>
</tr>
<tr>
<td>T2 moved 25m</td>
<td>1.7</td>
<td>2.0</td>
<td>2.5</td>
<td>737.5</td>
<td>984.8</td>
<td>24.7</td>
<td>5372.8</td>
</tr>
<tr>
<td>T2 moved 30m</td>
<td>1.7</td>
<td>2.1</td>
<td>2.5</td>
<td>737.4</td>
<td>994.9</td>
<td>21.9</td>
<td>5315.0</td>
</tr>
<tr>
<td>T2 moved 35m</td>
<td>1.7</td>
<td>2.1</td>
<td>2.5</td>
<td>737.4</td>
<td>1052.3</td>
<td>19.1</td>
<td>5257.2</td>
</tr>
<tr>
<td>T2 moved 40m</td>
<td>1.7</td>
<td>2.2</td>
<td>2.5</td>
<td>740.1</td>
<td>1047.6</td>
<td>16.3</td>
<td>5204.4</td>
</tr>
<tr>
<td>T2 moved 45m</td>
<td>1.7</td>
<td>2.3</td>
<td>2.5</td>
<td>740.1</td>
<td>1047.6</td>
<td>13.5</td>
<td>5151.7</td>
</tr>
<tr>
<td>T2 moved 50m</td>
<td>1.7</td>
<td>2.3</td>
<td>2.5</td>
<td>740.1</td>
<td>1047.6</td>
<td>10.7</td>
<td>5098.9</td>
</tr>
<tr>
<td>T2 moved 55m</td>
<td>1.7</td>
<td>2.4</td>
<td>2.5</td>
<td>740.1</td>
<td>1047.6</td>
<td>8.0</td>
<td>5046.1</td>
</tr>
<tr>
<td>T2 moved 60m</td>
<td>1.7</td>
<td>2.5</td>
<td>2.5</td>
<td>740.1</td>
<td>1047.6</td>
<td>5.2</td>
<td>4993.3</td>
</tr>
</tbody>
</table>

---

**Table 1: Full results table**

ICOWES 2013 Conference
17 June 2013 - 19 June 2013, Lyngby, Denmark
Maximum power-point tracking control for wind farms

P. M. O. Gebraad and J. W. van Wingerden
Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628CD Delft, The Netherlands

ABSTRACT

This paper presents a data-driven adaptive scheme to adjust the control settings of each wind turbine in a wind farm such that an increase in the total power production of the wind farm is achieved. This is done by taking into account the interaction between the turbines through wake effects. The optimization scheme is designed in such a way that it yields fast convergence, so that it can adapt to changing wind conditions quickly. The scheme has a distributed architecture in which each wind turbine adapts its control settings through gradient-based optimization, using information that it receives from neighbouring turbines. The novel control method is tested in a simulation of the Princess Amalia Wind Park. It is shown that the distributed gradient-based approach performs the optimization in a more time-efficient manner compared to an existing data-driven wind farm power optimization method that uses a game theoretic approach. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS
wind farms; distributed control; optimization

1. INTRODUCTION

The aim of control algorithms in modern wind turbines is to adjust the control degrees of freedom of the turbine, such as the generator torque and the pitch angles of the rotor blades, to changing wind conditions, with the aim of maximizing the energy capture of the wind turbine while keeping the structural loads on the turbine within acceptable limits. Nowadays, wind turbines are often placed with other turbines in wind farms to reduce use of space and costs of installation and maintenance. However, placing wind turbines in a wind farm introduces aerodynamic interaction between the turbines that affect the power production and loads on each turbine in the farm. These interactions effects are not taken into account in the current practice of wind turbine control design.

The aerodynamic interaction follows from the fact that a wind turbine converts the kinetic energy of an incoming wind flow into electrical energy, which results into the formation of a wake of turbulent slow moving air downstream of the rotor. As the wake travels downstream, the wake expands, and recovers to the free stream conditions because of mixing with the surrounding air. If another turbine is standing in the path of a wake that has not fully recovered, the reduced wind speed in this wake results in a lower electrical power production of the downstream turbine. Adjusting the control parameters of a wind turbine affects the extraction of power from the wind flow, and therefore the velocity deficit in the wake it produces. Therefore in wind farms, in which turbines are placed relatively close to each other, the wake effect causes a coupling between the control parameters of upstream turbines and the power productions and loads on downstream turbines. Cooperative control strategies that take into account the wake effect can be used to optimize the total power production of the wind farm. This is done by reducing the power production of the upstream turbines, in order to reduce the velocity deficit in the downstream wind flow, which increases the power production of the downstream turbines [1, 2, 3]. In a similar manner, it is possible use cooperative control to distribute the structural loads acting on the individual wind turbines in the wind farm more equally.
One approach to deal with wake interactions is to derive a model that describes the dynamics of the wind farm, and to use this model to synthesize control laws \[4, 5, 6\]∗, or to directly calculate control actions using model predictive control techniques \[7, 8, 9\]. However, because complicated nonlinear, time-varying models are needed to accurately describe the interaction between the aerodynamics of the wind flow in the farm and the turbine dynamics, such a model-based control strategy is difficult to apply in real-time. Therefore, this paper aims to develop a wind farm control method that is data-driven rather than model-based in the sense that it makes direct use of measured data in order to optimize the control parameters of the wind turbines and adapt them to time-varying wind conditions, without using a predefined model that predicts the effect of each control action. Thereby the objective of the optimization is to maximize the total power production of the wind farm in below-rated wind conditions, although the method may be extended to perform load control, by including static load measures in the objective function, as was done in \[7, 5\].

The method that in this paper is chosen to perform the data-driven adaptive control is known as Maximum Power-Point Tracking (MPPT). In previous work \[10\], the MPPT method was used to optimize the power of a single wind turbine using a real-time closed-loop scheme, where the change of the power production of the turbine as a result of control changes are measured, and subsequently the control parameters are adapted in a direction that yields a power improvement. In this paper, the MPPT method is extended in such a way that it optimizes the total power of a wind farm, by letting the wind turbines exchange information about their power production with other wind turbines in the wind farm†.

In this work, the MPPT wind farm control method is made adaptive to time-varying wind speeds, by designing the algorithm in such a way that its objective function is the efficiency by which the wind farm converts the kinetic energy of the incoming wind into electrical energy, rather than the total power production of the wind farm. The wind farm control also needs to be adaptive to other changing wind conditions, such as a changing wind direction, or a changing turbulence intensity which will affect the amount of mixing with the surrounding air and thereby the wake recovery. To accomplish this, it is required that the optimization takes place in a time-efficient manner. To this end, gradient-based techniques are used to perform the optimization, and the algorithm is designed using a distributed architecture in which the control parameters of a wind turbine are adapted based on information from the nearest neighboring turbines only. This architecture is illustrated in Figure 1.

This paper presents the new MPPT wind farm control method, and demonstrates its features through simulation examples in which the performance of the method is compared with a benchmark algorithm using the game theoretic

∗ The wind farm model that is used in \[5\] to synthesize control laws does not include interaction between turbines through the wakes.
† A preliminary version of this work appeared in \[11\].
approach of [12]. The simulation results are generated using the Jensen wake model, to which a delay structure is added to simulate the dynamics of the wake travelling through the wind farm.

The paper is organized as follows. A full explanation of the MPPT wind farm control approach is given in Section 2. The game theoretic wind farm control approach is explained in Section 3. An explanation of the wind farm model used in the simulation examples is given in Section 4. In Section 5, the simulation examples are described in detail and the results are given. Finally, in Section 6 the conclusions are presented.

2. MAXIMUM POWER-POINT TRACKING CONTROL FOR WIND FARMS

In this section, the MPPT wind farm control method is presented in two variants: the Gradient-Ascent MPPT (GA-MPPT) method and the Quasi-Newton MPPT (QN-MPPT) method. Both methods make use of gradient-based optimization techniques to find the control settings that yield a maximum total power production of the wind farm. As these gradient-based techniques can be categorized as being local optimization techniques, they may converge to a local maximum instead of a global maximum of the total power [13]. Further, in developing the MPPT methods simplifying assumptions are made in order to be able to perform the optimization using a distributed control architecture, in which each turbine uses information from the nearest neighbouring downstream turbine only. This gradient-based, distributed optimization approach is taken to improve the time-efficiency of the optimization.

The GA-MPPT and QN-MPPT methods are explained for a single row of wind turbines in Sections 2.1 and 2.2, respectively. In Section 2.3, the two methodologies are extended in such a way that they can be used on any wind farm configuration.

2.1. Gradient-Ascent MPPT control of a row of wind turbines

Consider a row of $n$ wind turbines standing in the wake of each other, in a wind field with an incoming free stream speed $V_{\infty}$, as depicted in Figure 2. The turbines have power productions $\{P_i\}_{i=1}^n$ and certain control settings $\{a_i\}_{i=1}^n$ that influence the power production of the turbines. In the simulation examples in this paper, it is assumed that the control variable $a_i$ is the axial induction factor of turbine $i$. This is a generalization in the sense that in a modern turbine the axial induction factors can be influenced by adapting either the blade pitch angles, or by scaling the generator torque which changes the tip speed ratio of the rotor [14]. Hence, in practice, one would use the MPPT scheme as presented here as a supervisory controller adjusting the reference signals for the blade pitch angles for each turbine, or the generator torque scaling factors used to adjust the rotor speed of each turbine. Alternatively, one could make use of knowledge of the power and thrust characteristics of a turbine rotor, if available, to find the pitch and torque that yield a desired axial induction provided by a wind farm controller, as was shown in [15]. In both cases, each of the local turbine controllers would track the reference signals while fulfilling some requirements for safe operation (by limiting pitch rates for example).

Due to wake interaction, changing a control parameter $a_i$ influences $\{P_j\}_{j=1}^n$, the power productions of turbine $i$ and the turbines downstream of turbine $i$. Further, the power production of the turbines is dependent on the kinetic energy of the incoming wind field. The kinetic power of wind with air density $\rho$ passing through an area $A$ with speed $V_{\infty}$ is given by [14]:

$$P_V = \frac{1}{2} \rho AV_{\infty}^3.$$ (1)

Figure 2. A row of $n$ wind turbines with power productions $\{P_i\}_{i=1}^n$ and control parameters $\{a_i\}_{i=1}^n$, in a wind field with free-stream speed $V_{\infty}$. 

Wind Energ. 2012; 00:1–18 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls
If constant air density is assumed, the efficiency by which the row of turbines converts the energy of the incoming wind field \( P_V \) into electrical energy can be maximized by solving the following optimization problem:

\[
\max_{\{a_i\}_{i=1}^n} \sum_{i=1}^n \tilde{P}_i(a_1, a_2, \ldots, a_i), \quad \text{with } \tilde{P}_i = \frac{P_i}{V_{\infty}}.
\]  

(2)

Note that in the above optimization problem, the power production of each turbine is divided by a factor that is proportional to the power of the incoming wind field. To perform the optimization in a local sense, the control parameters \( \{a_i\}_{i=1}^n \) can be iteratively updated using a gradient-ascent optimization method [13], resulting in the following Gradient-Ascent Maximum Power-Point Tracking (GA-MPPT) control update law:

\[
a_i(k+1) = a_i(k) + K \sum_{j=i}^n \frac{\partial \tilde{P}_j}{\partial a_i}(k),
\]

(3)

for \( i = 1, \ldots, n \), with index \( k \) denoting the iterations, and with a small scalar \( K > 0 \) being a scaling factor for the size of the steps on \( a_i \). This design variable \( K \) can be used to tune the convergence properties of the gradient-ascent optimization.

In order to perform the optimization in a real-time data-driven manner, the gradients \( \frac{\partial \tilde{P}_j}{\partial a_i} \) can approximated from the past iterations through first-order backward differencing:

\[
\frac{\partial \tilde{P}_j}{\partial a_i}(k) \approx \frac{\tilde{P}_j(k) - \tilde{P}_j(k-1)}{a_i(k) - a_i(k-1)}.
\]

(4)

A difficulty of the approach described above is that it takes a substantial amount of time to obtain the gradients of the objective function. For example, suppose that in iteration \( k, a_1 \) is changed by a certain step, then to find the gradient \( \partial P_n/\partial a_1(k) \) using the above update rule, one would have to wait for the air in the wake of turbine 1 to travel to turbine \( n \) to find the effect of the control update on \( P_n \), which is the power of the last turbine in the row. Because of the large distances between the turbines (typically 7 to 8 rotor diameters), the time this takes is very long for a large wind farm. During this travelling time, the speed of the incoming wind field is likely to have changed. Also other wind conditions may change over time, such as the wind direction, and the turbulence intensity in the free stream flow, which affects the amount of wake recovery in between the turbines. To make the algorithm adaptive to time-varying wind conditions, two changes are made to the control scheme described above:

1. To overcome the problem of a changing speed of the incoming wind field, the delays related to the wind travelling from one turbine to the next are taken into account in the definition of the efficiencies \( \tilde{P}_i \). Let \( T_{V_{\infty} \rightarrow i} \) denote the time it takes for the wind field to travel from \( x_{V_{\infty}} \) (the location where the incoming free-stream wind speed is measured) to \( x_i \) (the location of a turbine \( i \)). Then to compensate for the wind travelling delays, the efficiency \( \tilde{P}_i \) can be found from the power \( P_i \) at a time instant \( t \), by:

\[
\tilde{P}_i = \frac{P_i(t)}{V_{\infty}^{\triangledown}(t)^3},
\]

(5)

with:

\[
V_{\infty}^{\triangledown}(t) = V_{\infty}(t - T_{V_{\infty} \rightarrow i}).
\]

(6)

2. Changes in wind conditions such as wind direction and turbulence intensity change the way in which the power of each turbine is dependent on the control parameters of upstream turbines. A speed-up of the algorithm is needed for the optimization to be able to track these changes. A practical approach to speed-up the optimization scheme is to only take into account the influence of a turbine’s control settings on the power of the turbine itself, and on the power of the neighbouring downstream turbine. This then results in the following control update scheme:

\[
a_i(k+1) = a_i(k) + K \left[ \frac{\partial \tilde{P}_i}{\partial a_i}(k) + \frac{\partial \tilde{P}_{i+1}}{\partial a_i}(k) \right].
\]

(7)

Only taking into account the effect on the downstream neighbouring turbine can be a good approximation as in practical cases there is a substantial reduction of the velocity deficit in the wake as the air travels from one turbine to the next, because of mixing with the free stream air. Thus, the effect of a control setting change is far larger on the nearest downstream neighbouring turbine than on turbines further downstream.
2.2. Quasi-Newton MPPT control of a row of wind turbines

In order to further improve the convergence properties of the algorithm, one can use a Quasi-Newton optimization method to perform the local optimization of the control variables. When the vector notation $a = [a_1, \ldots, a_n]^T$ is used for the set of control parameters, the Quasi-Newton update law reads:

$$a(k + 1) = a(k) + KB(k) J(k).$$

As in the previous method, the scalar parameter $K > 0$ again determines the step-size. The matrix $J(k)$ represents an approximation of the Jacobian of the objective function, in which again only the gradients corresponding to the effect of a turbine’s control parameter on its own power and on its downstream neighbouring turbine are taken into account:

$$J(k) = \begin{bmatrix}
\frac{\partial P_1}{\partial a_1}(k) + \frac{\partial P_2}{\partial a_1}(k) \\
\frac{\partial P_2}{\partial a_2}(k) + \frac{\partial P_3}{\partial a_2}(k) \\
\vdots \\
\frac{\partial P_{n-1}}{\partial a_{n-1}}(k) + \frac{\partial P_n}{\partial a_{n-1}}(k) \\
\frac{\partial P_n}{\partial a_n}(k)
\end{bmatrix}. \quad (9)$$

As in the previous method, the gradients $\partial P_i/\partial a_i$ are approximated using (4). The matrix $B$ represents an approximation of the inverse Hessian of the objective function. To avoid the inversion of an approximate Hessian matrix, the matrix $B$ is directly approximated using the Davidon-Fletcher-Powell formula [16]:

$$B(k) = B(k-1) + \frac{\Delta a(k)\Delta a(k)^T}{\Delta J(k)\Delta J(k)^T} - \frac{B(k-1)\Delta J(k)\Delta J(k)^T B(k-1)^T}{\Delta J(k)\Delta J(k)^T}, \quad (10)$$

with:

$$\Delta a(k) = a(k) - a(k-1), \quad (11)$$

$$\Delta J(k) = J(k) - J(k-1). \quad (12)$$

As a starting point of the above iterations, $B(0)$ should be set to a symmetric positive definite matrix, for example, the identity matrix.

2.3. MPPT control of a wind farm

This section presents the scheme to perform the real-time closed-loop MPPT control on a wind farm of an arbitrary, but known spatial configuration, taking into account the delays between the control update and the power responses of the different turbines. Let $F = \{1, 2, \ldots, N\}$ denote a set of indices that number the wind turbines in a wind farm, with $N$ denoting the total number of turbines. Let $G \subset F$ be the set of turbines that are directly influencing downstream turbines through wake interaction, and let $d(i)$ be the index of the nearest neighbour downstream turbine that a turbine $i \in G$ is directly influencing. Further, $L = \{i \in F | d(i) \notin F\}$ is the set of turbines that are not influencing other turbines. In Figure 3 an example is given of how to define the sets $F$, $G$, and $L$ and the mapping $i \mapsto d(i)$ for a given wind farm configuration and wind direction. It is assumed that the sets $F$, $G$, $L$ can be updated using information of the wind farm configuration and the wind directions in the wind farm. Notice that an estimate of the wind direction is available in most wind turbines, as it is used to align the rotor axis with the wind direction through yaw control.

Using the above definitions, the GA-MPPT control update law for the wind farm is written as:

$$a_i(k + 1) = a_i(k) + K \left[ \frac{\partial P_i}{\partial a_i} + \frac{\partial P_{d(i)}}{\partial a_i} \right] \quad \forall i \in G, \quad (13)$$

and the QN-MPPT control update law for the wind farm is:

$$a_i(k + 1) = a_i(k) + Ks_i(k) \quad \forall i \in G, \quad (14)$$

with $s_i(k)$ being the $i$-th element of the search direction vector $s(k)$, defined as:

$$s(k) = B(k) J(k), \quad (15)$$
Figure 3. The above picture shows the top view of a 4-by-4 wind farm in a south-eastern wind flow. The dotted arrows show which turbine is directly influencing which other nearest neighbour downstream turbine through wake interaction. In this case, the set of indices numbering each turbine is $F = \{1, 2, \ldots, 16\}$. The indices of turbines $i$ that are influencing other turbines are collected in the set $G$. The index of the neighbouring downstream turbine that a turbine $i \in G$ is directly influencing is given by $d(i)$. The mapping $i \rightarrow d(i)$ is given in the table below the picture. The indices of turbines that do not influence other turbines are collected in the set $L = \{4, 8, 12, 13, 14, 15, 16\}$. The notation $R(i)$ is used for the full set of turbines that are standing in the wake of a turbine $i \in G$ (it is used to define the wind farm simulation model in Section 4).

\[
\begin{array}{cccccccccc}
\text{turbine index } i \in G & 1 & 2 & 3 & 5 & 6 & 7 & 9 & 10 & 11 \\
\text{turbine } d(i) & 5 & 7 & 8 & 10 & 11 & 12 & 14 & 15 & 16 \\
\text{set } R(i) & \{1, 6, 11, 16\} & \{2, 7, 12\} & \{3, 8\} & \{5, 10, 15\} & \{6, 11, 16\} & \{7, 12\} & \{9, 14\} & \{10, 15\} & \{11, 16\}
\end{array}
\]

\[
\begin{array}{cccccccccc}
\text{turbine index } j \in F & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 10 \\
\text{turbine } i(j) & 1 & 2 & 3 & 4 & 5 & 1 & 2 & 3 & 9 \\
\end{array}
\]

with:

\[
J(k) = \begin{bmatrix}
J_1(k) \\
\vdots \\
J_G(k)
\end{bmatrix}
\]

\[
J_i(k) = \left\{ \begin{array}{ll}
\frac{\partial \tilde{P}_i}{\partial a_i}(k) + \frac{\partial P_{d(i)}}{\partial a_i}(k) & \text{for } i \in G, \\
0 & \text{for } i \notin G,
\end{array} \right.
\]  

(16)

and $B(k)$ being generated by the update rule (10). In both MPPT methods, turbines in the set $L$ are controlled to operate in such a way that the power of the turbine itself is maximized:

\[
a_i = a_i^{\text{opt}} \forall i \in L,
\]

(17)

where $a_i^{\text{opt}}$ is the control setting that yields maximum power production for the turbine $i$ itself. In the case that $a_i$ is the axial induction factor, $a_i^{\text{opt}} = 1/3$, [14].

After a control variable $a_i$ is updated, there is a time delay before this change has an effect on the turbine $i$ itself, and another time delay before the change has an effect on the neighbouring downstream turbine $d(i)$. The scalar $T_{s,t}$ denotes the largest settling time of the responses of the power $P_i$ of each turbine $i \in G$ to the change of their own control variables $a_i$. The scalar $T_{s,d} \in \mathbb{R}$ is an upper bound for the time interval that it takes for each control variable $\{a_i | i \in G\}$ to have its full effect on the power production of the neighbouring downstream turbine, $P_{d(i)}$. The interval $T_{s,d}$ includes the maximum wake travelling time between a turbine $i \in G$ and its downstream neighbouring turbine $d(i)$. Therefore, the interval $T_{s,d}$ can be assumed to be larger than $T_{s,t}$. Notice that if the control variables $\{a_i, i \in G\}$ are updated simultaneously to an iteration $a_i(k)$, at a time instant denoted by $t_{\text{upd}}$, the gradient updates can be scheduled according to the following update rules:

\[
\frac{\partial \tilde{P}_i}{\partial a_i}(k) = \tilde{P}_i \left( t_{\text{upd}} + T_{s,t} \right) - \tilde{P}_i \left( t_{\text{upd}} \right) \\
+ \tilde{P}_i \left( t_{\text{upd}} \right) (a_i(k) - a_i(k-1)),
\]

(18)
The pseudocode below shows the Gradient-Ascent MPPT wind farm control algorithm. The variables $\pi_i$ and $P_i$ are used to store past values of the control variables and the power of the turbines.

For all $i \in G$ do
  measure $P_i(t)$, estimate $V_{\infty,i}^{del} (t)$, $\tilde{P}_i \leftarrow P_i(t) V_{\infty,i}^{del} (t)^{-3}$
  $\tilde{P}_i \leftarrow \tilde{P}_i$
end for

for all $i \in G$ do
  measure $P_{d(i)}(t)$, estimate $V_{\infty,d(i)}^{del} (t)$, $\tilde{P}_{d(i)} \leftarrow P_{d(i)}(t) V_{\infty,d(i)}^{del} (t)^{-3}$
  $\tilde{P}_{d(i)} \leftarrow \tilde{P}_{d(i)}$
end for

$\pi_i \leftarrow \pi_i + K \left( \frac{\partial P_{d(i)}}{\partial a_i} + \frac{\partial P_{d(i)}}{\partial a_i} \right)$

for all $i \in G$

$\tau \leftarrow 0$
end if
end loop

\[
\frac{\partial \tilde{P}_{d(i)}}{\partial a_i}(k) = \frac{\tilde{P}_{d(i)} \left( v^{upd}_k + T_{s,d} \right) - \tilde{P}_{d(i)} \left( v^{upd}_k + T_{s,t} \right) \left( a_i(k) - a_i(k-1) \right)}{a_i(k) - a_i(k-1)},
\]

for all $i \in G$. In Algorithm 1 the complete GA-MPPT wind farm control scheme is given. In this algorithm, $\Delta t$ denotes the interval between two samples. In line 9 of Algorithm 1, the value of $\Delta t$ is employed to update a scalar time counter $\tau \in \mathbb{R}$ that schedules the updates of the gradients. The size of the initial step on the control settings $\{a_i \in G\}$ is $(\Delta a)_{init}$, where $(\Delta a)_{init} > 0$ is a scalar that is to be chosen beforehand. A Boolean variable $LocGr$ is used to memorize whether or not the gradients $\frac{\partial \tilde{P}_{d(i)}}{\partial a_i}$ have been updated after the last control update. Estimates of the speed of the incoming wind field $V_{\infty}$ are to be produced using measuring [17] or filtering techniques [18, 19], and these estimates of $V_{\infty}$ are to be stored for a certain time window to be able to calculate the lagged variables $V_{\infty,i}^{del} (t)$ in lines 4, 12, and 20 of Algorithm 1. In a straightforward manner, Algorithm 1 can be adapted to give the QN-MPPT method.

### 3. Benchmark Wind Farm Control Algorithm: The Game Theoretic Approach

In the simulation examples of Section 5, the MPPT approaches are compared with a Game Theoretic (GT) wind farm control approach with full communication between the turbines, presented in [12]. Like the MPPT method, the GT approach of [12] is data-driven, since it only needs measurements of the power and the control parameters to track the point of maximum power. A similar game-theoretic approach was taken in [20], but the latter uses knowledge of the model to efficiently perform the optimization. An important difference between the MPPT method and the GT optimization approach with full communication as presented in [12], is that this GT approach aims to optimize the settings of each turbine by...
evaluating their effect on all the turbines in the wind farm.\textsuperscript{1} Furthermore, the GT approach performs this optimization by making random perturbations to the control variables and holding the settings if they yield an improvement of the wind farm total power production, so as to iteratively find the global maximum of the wind farm total power. To evaluate the effect of each control variable change on the total power production of the wind farm, the algorithm has to wait until the wake has travelled through the entire wind farm. This waiting time is denoted by $T_{s,p}$.

In Algorithm 2, the control scheme of the GT approach is given as it is implemented in our simulations. The algorithm has two parameters that are used to set the exploration rate of the randomized optimization:

- a scalar $E \in [0, 1]$ that defines the probability of using a new random setting for $a_i$, instead of keeping the settings that yielded the largest total power so far,
- a scalar $K \in [0, 1]$ that defines the size of the interval in which the random steps on the control settings are chosen.

The range $[a_{\min}, a_{\max}]$ is the set of allowable values for the control settings, which for the axial induction factor is given by $[0, 1/3]$. The algorithm is somewhat different than the one presented in [12], since in the exploration it makes small random perturbations in each iteration, rather than taking random values in the full range $[a_{\min}, a_{\max}]$. This change is made to improve the convergence speed of the algorithm, and reduce oscillations of the power signal.

Algorithm 2 The pseudocode below shows a wind farm control algorithm similar to the Game Theoretic approach of [12]. The values of $R_1$ and $R_2$ are drawn randomly using a uniform distribution. The variables $\pi_i$ and $P_i$ are used to store past values of the control variables and the power of the turbines.

1: given $T_{s,p}$, $K \in [0, 1]$, $E \in [0, 1]$ and set $F$
2: $\tau \leftarrow 0$
3: $a_i \leftarrow a_i^{\text{opt}} \forall i \in F$
4: $\overline{P} \leftarrow \sum_{i=1}^{N} P_i(t)$
5: $\overline{a}_i \leftarrow a_i$
6: loop
7: $\tau \leftarrow \tau + \Delta t$
8: if $\tau > T_{s,p}$ then
9: if $\sum_{i=1}^{N} P_i(t) > \overline{P}$ then
10: $\pi_i \leftarrow a_i \forall i \in F$
11: $\overline{P} \leftarrow \sum_{i=1}^{N} P_i(t)$
12: end if
13: for all $i \in F$ do
14: $R_1 \leftarrow$ random value $\in [0, 1]$
15: if $R_1 < E$ then
16: $R_2 \leftarrow$ random value $\in [a_{\min}, a_{\max}]$
17: $a_i \leftarrow \min \{ \overline{a}_i + K R_2, a_{\max} \}$
18: else
19: $a_i \leftarrow \overline{a}_i$
20: end if
21: end for
22: $\tau \leftarrow 0$
23: end if
24: end loop

4. WIND FARM SIMULATION MODEL

In Section 5, the control methods are evaluated in simulations of a wind farm. The simulation model used in this case study is the Jensen model, that was first introduced in [21], extended with a delay model to include the wake travelling dynamics. The Jensen model is a relatively simple engineering model that gives an estimate of the velocity profile in the wind farm as a function of the incoming wind field and the set of axial induction factors of each turbine $\{a_i | i \in F\}$. The Jensen model is also used to include wake velocity deficit effects in the well-known Aeolus SimWindFarm wind farm model [22].

\textsuperscript{1}[12] also presents a form of the Game-Theoretic method with limited communication between the turbines, but it is shown in the same paper that this particular method has slower convergence than the full communication GT approach.
P. M. O. Gebraad and J. W. van Wingerden

Maximum power-point tracking control for wind farms

Figure 4. The wake expansion parameters in the Jensen model. The value of $A_{i,j}^{ol}$ is set to a maximum value of 1 when the turbines are in each full wake, i.e., when $A_i$ and $A_j$ fully overlap.

Consider a single turbine $i$ with a rotor diameter $D_i$, with its rotor axis aligned with the wind direction. Assume an incoming uniform wind field with a free-stream speed $V_\infty$. Let $(x, r)$ be a point in the wake of the turbine, where $x$ is the distance to the rotor disk plane of the turbine, and $r$ is the distance to the centerline of the wind turbine rotor axis (see Figure 4). The Jensen model estimates the wind speed in the point $(x, r)$ to be:

$$V_{w,i}(x, r, a_i) = V_\infty [1 - \delta V_{w,i}(x, r, a_i)]$$

with the fractional velocity deficit $\delta V_{w,i}(x, r, a_i)$ given by:

$$\delta V_{w,i}(x, r, a_i) = \begin{cases} 
  2a_i \left[ \frac{D_i}{D_{w,i}(x)^2} \right]^2 & \text{for } r \leq \frac{D_{w,i}(x)}{2}, \\
  0 & \text{for } r > \frac{D_{w,i}(x)}{2},
\end{cases}$$

where $D_{w,i}$ is the diameter of the wake, which is assumed to have a circular cross-section. The diameter is assumed to expand proportional to the distance $x$:

$$D_{w,i}(x) = D_i + 2\kappa x,$$

where parameter $\kappa$ represents a tunable wake expansion coefficient. In the simulation examples of Section 5, this parameter is set to $\kappa = 0.084$, to fit the offshore wind farm power data provided in [23].

The model is extended to include multiple turbines with interacting wakes. Then the effective wind speed $V_j$ for a turbine $j \in F$ is found by combining the estimated wind velocity deficits created by each upstream turbine:

$$V_j = V_\infty [1 - \delta V_j],$$

with:

$$\delta V_j = 2 \sum_{i \in F, x_i < x_j} a_i \left[ \frac{D_i}{D_{w,i}(x_j - x_i)} \right]^2 \left[ \frac{A_{i,j}^{ol}}{A_j} \right]^2,$$

where $A_j$ is the rotor swept area of a turbine $j$, and $A_{i,j}^{ol}$ is the overlapping area of the rotor swept disk of a turbine $j$, and the wake generated by an upstream turbine $i$ at the rotor plane of turbine $j$ (see Figure 4), which are calculated using basic trigonometric relationships.

When the effective wind speed at each turbine is known, the power of each turbine is calculated as [14]:

$$P_j = \frac{1}{2} \rho A_j C_P(a_j) V_j^3,$$

where $\rho$ is the air density and $C_P$ is the power efficiency coefficient, which is expressed as a function of the axial induction factor:

$$C_P(a_j) = 4a_j(1 - a_j)^2.$$

In the above form, the Jensen model is a static model, in which a change in the axial induction factor has an immediate effect on the total power. To be able to evaluate the time-efficiency of the different wind farm control algorithms described...
in this paper, simplified wake travelling dynamics are added to the model. This is done by including estimated delays corresponding to the wake travelling from one turbine to the next in the Jensen model, following an approach similar to that presented in [24]. In this approach, an estimate of the wake travel time between a turbine \( i \) and its nearest downstream neighbour \( d(\bar{\bar{i}}) \), denoted by \( T_{i \to d(\bar{\bar{i}})} \), is made by assuming a constant speed in between the turbines that is equal to the average of the wind speed just behind the turbine \( i \) and the wind speed just in front of the downstream turbine \( d(\bar{\bar{i}}) \):

\[
T_{i \to d(\bar{\bar{i}})} = \frac{x_{d(\bar{\bar{i}})} - x_i}{\frac{1}{2} [V_i [1 - 2a_i] + V_{d(\bar{\bar{i}})}]}.
\]  

(27)

Before a change in the incoming wind field at location \( x_{V_\infty} \), has its effect on a turbine \( i \), the wind has to travel from location \( x_{V_\infty} \) to the turbine location \( x_i \), see also Figure 3. To incorporate this effect, the delays approximated with equation (27) are incorporated in the model in such a way that a change in the incoming wind field has a delayed effect on the turbines. In this model, a uniform incoming free stream wind field with a speed \( V_\infty(t) \) is prescribed as the wind speed just in front of a wind turbine \( f \), i.e. \( x_{V_\infty} = x_f \), where \( f \) is the turbine that is standing upstream of all other turbines, i.e.:

\[
f = \arg \min_{i \in F} (x_i).
\]  

(28)

For example, for the wind farm configuration of Figure 3, \( f = 1 \). The notation \( \bar{\bar{R}}(j) \) is used for the complete row of turbines \( \bar{\bar{R}}(i) \) that a certain turbine \( j \) is part of, i.e. \( \bar{\bar{R}}(j) \) is the largest set \( \bar{\bar{R}}(i) \) for which it holds that \( j \in \bar{\bar{R}}(i) \). Further, let \( i(j) \) be the first member of the set \( \bar{\bar{R}}(j) \) (see Figure 3 for an example of the mapping \( j \to i(j) \)), then by summing the different delays an expression for \( T_{V_\infty \to j} \), being the total wind travelling delay for a turbine \( j \), is found:

\[
T_{V_\infty \to j} = \frac{x_{i(j)} - x_f}{V_\infty (t - T_{i(j) \to j})} + T_{i(j) \to j},
\]  

(29)

with:

\[
T_{i(j) \to j} = \sum_{u \in R_j, u < x_i} T_{u \to d(u)}.
\]  

(30)

Substituting expression (29) in equation (6) yields:

\[
V_{\infty,j}^{del}(t) = V_\infty \left( \frac{t - x_{i(j)} - x_f}{V_\infty (t - T_{i(j) \to j})} - T_{i(j) \to j} \right).
\]  

(31)

Moreover, for a change in control variable \( a_i \) to have an effect on the downstream turbine \( d(\bar{\bar{i}}) \), the wake has to travel from turbine \( i \) to turbine \( d(\bar{\bar{i}}) \). To incorporate this effect, a delay structure is added to the model for the wake velocity deficit, which for the MPPT method is given by:

\[
\delta V_{j}^{del}(k) = 2 \sum_{i : x_i < x_j} \left( \delta V_{w,i,j}^{del}(k) \frac{A_{i,j}}{A_i} \right)^2,
\]  

(32)

with:

\[
\delta V_{w,i,j}^{del}(k) = a_i (k - \Delta_{i,j}) \left[ \frac{D_i}{D_{w,i}(x_j - x_i)} \right]^2,
\]  

(33)

where \( \Delta_{i,j} \) is the discrete delay as a consequence of the wake travelling from a turbine \( i \) to a turbine \( j \). For example, \( \Delta_{i,j} = 1 \) if \( j = d(\bar{\bar{i}}) \), \( \Delta_{i,j} = 2 \) if \( j = d(d(\bar{\bar{i}})) \), and so on. Then at time \( t_{upd} + T_{s,t} \), the power of each turbine changes in response to the change in its own control variable. Hence, the power estimate is updated by:

\[
V_j \left( t_{upd} + T_{s,t} \right) = V_{\infty,j}^{del} \left( t_{upd} + T_{s,t} \right) \left[ 1 - \delta V_j^{del}(k) \right],
\]  

(34)

\[
P_j \left( t_{upd} + T_{s,t} \right) = \frac{1}{2} \rho A_j C_P (a_j(k)) \left[ V_j \left( t_{upd} + T_{s,t} \right) \right]^3.
\]  

(35)

The settling time of the turbines with respect to a change in their own control variables is assumed to be \( T_{s,t} = 5s \).

In the MPPT method, the gradients are updated simultaneously after each wake has reached the next downstream turbine, which results in the following settling time \( T_{s,d} \) used in scheduling the gradient updates:

\[
T_{s,d} \approx \max_{i \in G} (T_{i \to d(\bar{\bar{i}})}).
\]  

(36)
At time $t_{upd}^k + T_{s,d}$ the wakes have travelled from one turbine to the next, and the velocity deficit in front of each turbine changes. Therefore, the wind velocities and powers are updated by:

$$V_j(t_{upd}^k + T_{s,d}) = V_{\infty,j}^d( t_{upd}^k + T_{s,d} ) \left[ 1 - \delta V_j^{del}(k+1) \right], \tag{37}$$

$$P_j(t_{upd}^k + T_{s,d}) = \frac{1}{2} \rho A_j C_P (a_j(k)) \left[ V_j(t_{upd}^k + T_{s,d}) \right]^3. \tag{38}$$

Then in the MPPT method, after the wake has travelled to the next turbine, the control variables are updated, hence:

$$t_{upd}^{k+1} = t_{upd}^k + T_{s,d}. \tag{39}$$

The wake travelling modelling results in the fact that after each control update in the MPPT optimization, initially the total power will decrease as a consequence of the fact that some turbines will decrease their own power extraction, but when the wakes of those turbines travel to the next row of turbines, the total power will increase as a consequence of the reduced velocity deficits in the wakes.

In the model used to evaluate the GT method, a similar behaviour is incorporated, but part of the delay structure is omitted because in the GT method the wake will have travelled through the full wind farm before a control update takes place. Under the same assumptions as used above to estimate $T_{s,d}$, an estimate of $T_{s,p}$ is obtained, which denotes the largest time it takes for a change in a control variable $a_i$ of a turbine $i \in G$ to have its effect on the power of all of its downstream turbines, by summing each of the turbine-to-turbine wake travel times. This time interval $T_{s,p}$ is used to schedule the control updates in the GT method. To find an expression for $T_{s,p}$, the notation $R(i)$ is used for a set that includes the index of a turbine $i \in G$ and the indices of the full row of turbines in the set $G$ that are affected by that turbine $i$, i.e., $R(i) = \{ i, d(i), d(d(i)), \ldots \}$ (see Figure 3 for an example of the mapping $i \mapsto R(i)$). Using this notation, the approximation is given by:

$$T_{s,p} \approx \max_{i \in G} \left( \sum_{j \in R(i)} T_{i \rightarrow j}(d(i)) \right). \tag{40}$$

The estimates for the delays in the model are fairly rough, but since the underlying assumptions are similar for the estimation of $T_{s,d}$ and $T_{s,p}$, these estimates can be used to make a relative comparison of the time-efficiency of each optimization method.

## 5. SIMULATION EXAMPLES

This section presents the results of simulation experiments that compare the performance of the different wind farm control methods presented in this paper in terms of the time-efficiency of the power optimization, the power production increases that can be achieved, and the adaptability of each method to varying wind conditions.

In the simulation examples, the MPPT approaches presented in 2 are compared to the Game-Theoretic (GT) approach described in Section 3, which is a global optimization approach for maximizing the total power production of the wind farm. It is shown that as a consequence of the simplifying assumptions that are taken in the MPPT optimization scheme (in which the effect on nearest neighbouring turbines is taken into account only), the power production increases that can be obtained with the MPPT approaches may be somewhat smaller than those that can be obtained with the GT approach, although the differences are small. The distributed gradient-based optimization approach of the MPPT wind farm control method yields a large improvement of the time-efficiency however, which is important for making real-time implementation possible.

In each of the simulation experiments, the wind farm model as presented in Section 4 is used to simulate the Princess Amalia Wind Park, an offshore wind farm that is located 23 km off the coast of The Netherlands, under different wind conditions. The Princess Amalia Wind Park consists of 60 wind turbines with a rotor diameter of 80 m and a rated power of 2 MW. The locations of the turbines in the The Princess Amalia Wind Park are shown in Figure 5, together with a compass rose that defines the different wind directions as mentioned further on in the simulation examples. In each of the cases, a constant air density $\rho = 1.225 \text{ kg} \cdot \text{m}^{-3}$ is assumed.

In Section 5.1, the time-efficiency and gain of the MPPT power optimization methods is compared to that of the GT approach in a simulation of the wind farm with a constant incoming wind speed and a single wind direction. Thereby each of the parameters of the optimization algorithms are set to deliver fast convergence properties for this specific case. In Section 5.2, it is evaluated whether the same settings will also result in good convergence properties for other wind directions, by repeating the simulation experiment of Section 5.1 for a range of wind directions. These results are then used in Section 5.3 to make an estimate of the energy production increase that can be achieved using these methods. In the final experiment in Section 5.4, it is shown that the proposed MPPT algorithms are able to deal with a time-varying incoming wind speed.
5.1. Comparative simulation study of the wind farm control approaches for a constant incoming wind speed

In the first simulation example, a uniform wind field with a direction of 25° is assumed, with a constant, below-rated speed of \( V_{\infty} = 8 \, \text{ms}^{-1} \). The parameters determining the iteration step-size of each of the methods are tuned to yield fast convergence towards the optimum. With the turbine powers \( P_i \), expressed in the megawatt unit, this resulted in setting \( K = 7.7 \) for the GA-MPPT approach, \( K = 5.12 \) for the QN-MPPT approach, and \( K = 0.06 \) and \( E = 0.1 \) for the GT approach. In both MPPT methods, the size of the initial step on the control settings is set to \((\Delta \theta)_{\text{init}} = 0.01 \).

The simulation results are given in Figure 6a. After the control updates take place, first a drop in total power production is observed, since the production of the upstream turbines decreases, and then after the wake has travelled to the downstream turbines, an increase of the total power is achieved. The results show that each of the control methods will iteratively improve the total electrical power production. The GA-MPPT approaches converge to a slightly lower total power than the GT and the QN-MPPT approach. This is because the GT is guaranteed to converge to a global optimum of the wind farm total power, and the QN-MPPT approach also finds this optimum in this case, but the GA-MPPT converges to a local optimum that is close to this global optimum. The MPPT approaches increase power much quicker than the GT approach. This is because the MPPT methods use gradient information to converge to the local optimum in a faster way. Also, it is because the distributed optimization approach of the MPPT methods consider the wake effect on the nearest neighbouring turbines only, which makes that the MPPT algorithms is able to update the control settings more frequently, since the turbine-to-turbine settling time \( T_{\text{set}} \) is much shorter than the total wind farm settling time \( T_{\text{set}} \).

5.2. Evaluation of the power gain of the wind farm control approaches for different wind directions

In the second case study, it is evaluated how the power increase that is achieved with the MPPT and GT control methods changes with the direction of the incoming wind. The simulation experiment of Section 5.1 is repeated for the wind directions 0°, 5°, 10°, ..., 355°. In each simulation the incoming wind speed is kept constant at \( V_{\infty} = 8 \, \text{ms}^{-1} \). The power increase that is achieved after each of the control methods have converged is shown in Figure 7. It can be seen that the power increase that is achieved is highly dependent on the wind direction, as the spatial configuration of the turbines in the Princess Amalia Wind Park is optimized for more frequently occurring wind directions. As in the previous case study, the MPPT methods converge much faster than the GT method. This is shown in Figure 7e, in which the convergence times of the methods are given for each wind direction.

The optimization results for different wind directions are obtained without adjusting the parameter \( K \) for each direction (the same settings are used as in the example of Section 5.1). In Figure 7d it can be seen that while the QN-MPPT may yield a slightly higher power increase for the 25° wind direction (for which the control parameters are tuned), for other wind directions the GA-MPPT yields a higher power increase. This is because the convergence properties of the QN-MPPT approach are sensitive to the tuning of the \( K \) parameter, and the QN-MPPT approach may need retuning of the step-size scaling parameter \( K \) to different wind directions to have good converge properties, while on the other hand, the GA-MPPT is more easy to use, in the sense that it does not need adjustment of the \( K \) parameter to have good convergence properties for different wind directions. This is also illustrated in Figure 6b and 6c. In Figure 6b the power timeseries for each of the optimization methods is shown for the wind direction 60°, where for each method the same tuning is used as previously for the 25° wind direction. It can be seen that this results in bad convergence properties for the QN-MPPT method, as it does not come as close to the global optimum found by the GT method as the GA-MPPT method does. When the QN-MPPT method is retuned to \( K = 0.044 \), better convergence properties can be obtained, the results are show in Figure 6c. Also, in Figure 6c, the GT method is retuned such that faster convergence occurs, by setting \( E = 0.2 \) and keeping \( K = 0.06 \).

5.3. Estimation of the annual energy gain of the wind farm control methods

In the wind farm model of Section 4, the fractional velocity deficit in the wake (i.e. the relative amount of wake recovery) is independent of the incoming wind speed. Therefore, the settings finally found by the GT and MPPT optimization algorithms yield the same power increase for different below-rated wind speeds. This can also be confirmed by rerunning the experiments of Section 5.2 with different below-rated incoming wind speeds (the results are omitted for brevity). By using wind measurements at a nearby location, an estimate is made of the increase of the energy annually produced in below-rated conditions that can be achieved using the different optimization methods. These wind measurements were made by the NoordzeeWind meteorological mast at a nearby location in the North Sea during the period from July 1st, 2005 to June 30th, 2006, [26]. The measurements are available at [25]. The measurements consist of 10 minute averages of the wind direction and the free stream wind speeds. With the wind farm model of Section 4, the energy production of the Princess Amalia Wind Park is calculated for each of the wind directions, for the case in which each turbine is controlled individually, and the results are shown in Figure 5c. When the final power increase that is obtained with each of the control methods is added to these productions, and the results are summed over the year, a rough estimate can be made of the annual production with each of the methods. In this way it is estimated that on a yearly basis, the energy produced in
below-rated wind conditions can be increased with 1.36% using the GA-MPPT method, with 1.19% using the QN-MPPT method, and with 1.42% using the GT method. Notice that in these calculations, the convergence time of each of the methods is not taken into account, and that it is thus assumed that each of the control methods is able to quickly track the changing wind conditions. Given the large convergence times of the GT method when compared to the MPPT method, it is less likely that the GT method is able to perform this tracking, and thus it is less likely that the estimated energy production increase can be achieved in practice using the GT method as described in Section 3 in an online implementation.

5.4. Simulation of the MPPT approaches with a varying wind speed

In the final case study, it is shown in simulation that indeed the total power can be optimized under varying wind speeds using the MPPT methods. The results are obtained by simulating the Princess Amalia Wind Park with a varying incoming wind speed signal, obtained from a part of the December 2010 wind speed measurements of the NoordzeeWind meteorological mast in the North Sea [25], shown in the top plot of Figure 8. It should be noted that this wind speed is smoothed, as it is an interpolation of 10 minute averages of the measured wind speed. In the lower two plots in Figure 8, it can be seen that a power increase of about 4% can be obtained with the MPPT techniques, using $K = 0.25$ for the QN-MPPT approach, and $K = 13$ for the GA-MPPT approach. In both MPPT approaches, the initial step size is set to $(\Delta t)_{\text{min}} = 0.01$. Notice that in this time-varying wind speed case, the algorithm is able to continue the optimization under changing wind velocity, making use of the fact that the objective function is defined as the sum of the efficiencies $\tilde{P}_i$ as defined in (6), rather than the sum of turbine power productions. In Figure 9 it is shown how this objective function is optimized within the first 20 minutes of the simulations.

6. CONCLUSIONS

In this paper, two data-driven MPPT control algorithms for wind farms are presented that optimize the control settings of each turbine in the farm in a real-time closed-loop manner. The control algorithms achieve a power production increase of the wind farm by taking into account the interaction between the turbines through the wake effect. A speed-up of the optimization is achieved by using gradient-based optimization techniques with a distributed approach in which we take into account the effect on neighbouring turbines only. Using information on the spatial configuration of the wind farm in this way, results in a much faster convergence of the power optimization than is achieved with the existing game-theoretic method with full communication between the turbines presented in [12]. This is demonstrated in the first simulation example in Section 5.1.

As the gradient information used in the GA-MPPT and QN-MPPT is calculated from measured data, the method is adaptive to changing wind conditions such as a changing wind direction. For the optimization algorithms to be able to track time-varying wind conditions it is needed that the optimization takes place in a time-efficient manner. Therefore, the gradient-based distributed MPPT approaches may be a more likely candidate for practical application than the GT approach.

It is shown in the second simulation example in Section 5.2, that the GA-MPPT method is more robust to changing wind conditions than the QN-MPPT approach, as the latter optimization method may need an adjustment of the iteration step-sizes to adapt to changing wind conditions.

The final simulation example in Section 5.4 showed that by letting the control algorithm optimize the energy conversion efficiency of the wind farm rather than the total power, the optimization scheme is made adaptive to time-varying incoming wind speeds. To define this efficiency, the algorithm needs an estimate of the effective incoming wind speed, and an estimate of the delays related to the wind field travelling from one turbine to the next. In the simulation experiments with the Jensen model, these wind speeds and delays are assumed to be exactly known. Future research aims at applying the MPPT wind farm methods on a more advanced simulation model that describes the wake and turbine dynamics in more detail, such as the model presented in [27, 28]. To be able to do this, measuring [17] or filtering techniques [18, 19] are to be applied that produce estimates of the speed of the incoming wind field, and models are to be developed that give estimates of the wake travelling delays.

Further, in future work the presented optimization methodology can be extended to above-rated wind conditions by including constraints in the optimization problem that give upper bounds on the power production of each turbine in the wind farm. Also, the optimization methodology may be used to perform balancing of the loads on each turbine, using an approach similar to the one presented in [7], in which the objective function is extended with static turbine load measures.
Figure 5. Properties of the Princess Amalia Wind Park offshore wind farm.

(a) Locations of the wind turbines (with rotor diameter $D = 80\,\text{m}$)

(b) Wind farm electrical power production for different wind directions, with $V_\infty = 8\,\text{m/s}$ without optimization, as predicted by the model presented in Section 4.

(c) Total annual wind farm electrical energy produced in below-rated wind conditions for different wind directions, without optimization, as predicted by the model presented in Section 4, using the wind data of the NoordZeeWind met mast [25].
Figure 6. Results of power optimization control with the GA-MPPT, QN-MPPT, and the GT approach in the wind farm simulation described in Section 5.1, where the incoming wind speed is kept constant at $V_{\infty} = 8$ m s$^{-1}$. On the left, the ‘o’-markers on the total power curves correspond to the time instances at which the control updates take place. On the right, the results are shown on a larger time range to show the convergence of the GT approach, and to show the effect of the randomization in this GT method, the distribution of the results of 100 experiments is shown.
Maximum power-point tracking control for wind farms

P. M. O. Gebraad and J. W. van Wingerden

Figure 7. Results of the power optimization the Princess Amalia Wind Park for different wind directions.
Figure 8. Results of wind farm simulation experiments with a varying incoming wind speed. The varying incoming wind speed $V_\infty$ is shown in the top plot. In the middle plot, it is shown how the different MPPT control methods yield an increase of the produced power when compared to the case where locally optimal control is used. The relative increase in power for each time instant is shown in the bottom plot.

Figure 9. The optimization of the objective function using the GA-MPPT and QN-MPPT methods as it takes place within the first 20 minutes of the wind farm simulation experiments with varying wind speeds as described in Section 5.4.
REFERENCES

A Data-Driven Model for Wind Plant Power Optimization by Yaw Control


Abstract—This paper presents a novel parametric model that will be used to optimize the yaw settings of wind turbines in a wind plant for improved electrical energy production of the whole wind plant. The parametric model has a limited number of parameters that are estimated based on data. This data-driven model predicts the effective steady-state flow velocities at each turbine, and the resulting electrical energy productions, as a function of the axial induction and the yaw angle of the different rotors. Moreover, we show how this model can be used for optimization of the yaw settings using a game-theoretic approach. In a case study we demonstrate that our novel parametric model fits the data generated by a high fidelity computational fluid dynamics model of a small wind plant, and that the data-driven yaw optimization control has a great potential to increase electrical energy production of the wind plant.

I. INTRODUCTION

Each wind turbine in a wind plant influences the electrical energy productions and loads of other turbines through wakes of slow moving, turbulent air, that form downstream of the turbine’s rotor. If another turbine is standing in the path of a wake that has not fully recovered to free stream conditions, the reduced wind speed in this wake results in a lower electrical energy production. The amount of wake interaction depends on time-varying atmospheric conditions (e.g. wind speed, turbulence, atmospheric stability, wind direction), and on the operating point of each turbine (rotor speed, pitch angles of the blades, yaw angle) that can be adjusted by changing the control settings of each turbine.

The goal of this paper is to develop an ‘internal model’ for a wind plant controller. The internal model predicts the effect of control settings on the steady-state wake interaction effects in the wind plant. The wind plant controller can use this model to improve electrical energy production of the wind plant by model-based optimization of the control settings. Previous work has mainly aimed at reducing wake interaction by adjusting the axial induction of turbines to improve wind plant performance (cf. [1]–[10], for example), which can be achieved by adjusting pitch and torque. In this work, we aim at using the yaw degree of freedom of the wind turbine to deflect the wake of turbines away from downstream turbines, which was shown to have great potential in [11]–[13]. Previous work on yaw optimization for wind plants, [14], did not include validation of the optimized settings using high-fidelity numerical simulations.

High-fidelity models, based on a coupling of detailed turbine dynamics models with accurate wind flow models, such as Computational Fluid Dynamics (CFD) based models [15]–[17], have an important role in wind plant controls development, as they allow the algorithms to be tested in a controlled environment. Because of their computational complexity, they are less suited as internal models however. Therefore, we have aimed at developing a novel simplified parametric model for which the parameters can be identified using turbine power measurements. In this paper, a high-fidelity CFD wind plant model is used to develop the simplified parametric model. Then that parametric model is used for model-based optimization of the yaw settings in a wind plant using a game-theoretic approach, and finally these model-predictive optimized settings are validated in the high-fidelity wind plant simulation. The low number of parameters in the simplified parametric model make that it has the potential to be tuned online, based on real-time turbine power measurements, in order to adapt to changing atmospheric conditions.

The rest of this paper is organized as follows. The experiments performed in the high-fidelity CFD simulator to obtain identification data for the parametric model are described in more detail in section II. The simplified parametric model is presented in section III. Section IV presents the game-theoretic approach to calculate optimal yaw control settings based on the simplified model. In section V, a simulation study is presented to validate these model-predictive optimized settings in a high-fidelity wind plant simulation.

II. EXPERIMENTS IN SOWFA, A HIGH-FIDELITY CFD WIND PLANT SIMULATOR

The Simulator for Onshore/Offshore Wind plant Applications (SOWFA) is a CFD simulator of the 3-dimensional wind flow around one or more turbine rotors in the atmospheric boundary layer. The rotating rotor blades are modeled through an actuator line approach [18]. The actuator lines are coupled with the FAST turbine aeroelastics simulator tool [19] that calculates the loads, power and rotor speed of each turbine, in addition to the forces that each turbine blade exerts on the flow. Each turbine can be controlled using an individual control algorithm implemented in FAST, but also...
through a supervisory or distributed controller. See [17], [20], [21] for more details.

In [12], [13], SOWFA simulation results were presented which show how effective the yaw techniques are at wake redirection, 2) what the effect of yaw wake redirection techniques is on the electrical energy production and loads of downstream turbines that are standing in the wake of the yawing turbine, 3) the effect of repositioning a turbine from full wake to partial wake on electrical energy production and loads. More in particular, in [13], experiments are described with a setup of two NREL 5-MW baseline turbines [22] that are aligned in the wind direction with a downwind spacing of 7 rotor diameters (7D). The simulated turbulent inflow has a mean hub-height free-stream wind speed \( U_\infty \) of 8m/s and a turbulence intensity of 6%. The data of the following two experiments performed in [13] are used in this paper (cf. Figure 1a): in Experiment 1, the upstream turbine (T1) is yawed to redirect its wake away from the downwind turbine (T2), resulting in an energy production decrease of T1, but an energy production increase of T2; in Experiment 2, T2 is moved in the cross-wind direction to reduce the overlap of its rotor with the wake of T1. See Figure 1b for the time-averaged power data from these experiments.

SOWFA high-fidelity CFD simulations are typically run for a few days on a cluster with a few hundred processors [12], [13]. Due to the complexity and computational costs of the SOWFA model, it is not suitable as an internal model for a wind plant controller. However, the data generated by SOWFA can be used to develop simplified models that can be directly used by the controller. In Section III, we describe how the power data from Experiments 1 and 2 are used to identify such a simplified control-oriented model. In Section V, SOWFA is used to evaluate the control techniques based on the simplified model in a high-fidelity simulation.

III. A DATA-DRIVEN PARAMETRIC WIND PLANT MODEL

The model presented here is a combination of the Park / Jensen model [23], [24], combined with a model for wake deflection through yaw [25]. Further, augmentations were made to the model in order to empirically fit the model with the power measurements obtained in the Experiments 1 and 2.

A. Turbine power

Let \( \mathcal{F} = \{1, 2, \cdots, N\} \) denote a set of indices that number the wind turbines in a wind plant, with \( N \) denoting the total number of turbines. When the effective wind speed at a turbine \( j \), denoted as \( U_j \), is known, the steady-state power of each turbine is calculated as [26]:

\[
P_j = \frac{1}{2} \rho A_j C_P(a_j, \gamma_j) U_j^3 \quad \forall j \in \mathcal{F}
\]  

(1)

where \( \rho \) is the air density, \( A_j \) is the rotor area, and \( C_P \) is the power coefficient of the turbine. In non-yaw idealized conditions, the power coefficient is related to the axial induction factor of each turbine, defined as \( a_j = 1 - U_{j,D}/U_j \) with \( U_{j,D} \) being the wind speed at the rotor, and \( U_j \) the free stream wind speed in front of turbine \( j \), as \( C_P(a_j, \gamma_j) = 4a_j[1 - a_j]^2 \) [26]. In the model presented here, a correction was applied on this relationship to account for the effect of the yaw misalignment angle \( \gamma_j \) on the rotor power coefficient, following the example of the experimental studies in [27]. Further, a constant scaling of the \( C_P \) value, \( \eta \), was used to account for other losses and match the maximum \( C_P = 0.482 \) value for the NREL 5-MW reported in [22]. This resulted in:

\[
C_P(a_j, \gamma_j) = 4a_j[1 - a_j]^2 \eta \cos(\gamma_j) p_P.
\]

(2)

While [27] found a parameter value \( p_P = 2 \) to fit data from wind tunnel tests, the parameters settings listed in Table I were found to fit the yaw-power plot of the upstream turbine (T1) in SOWFA Experiment 1, see Figure 1b, assuming an idealized axial induction of \( a_j = 1/3 \).

In the remainder of this section, it will be described how the effective wind speeds \( U_j \) at each turbine are estimated by the model, using a model for steady-state wake behavior.

B. Wake deflection

Yawing a turbine rotor causes the thrust force that the rotor exerts on the flow, \( F_\theta \), to rotate in such a way that a cross-wind component is induced [25], cf. Figure 2a, which causes the wind flow to deflect in the opposite direction of the yaw rotation. Since the wake deflection is induced by
the thrust force, the yaw deflection is a function of the thrust coefficient of the turbine \( C_T = 2 F_D / (\rho A_j U_j^2) \), which for non-yawed conditions can be related to the axial induction factor \( a_j \) of the rotor, as follows [26]:

\[
C_T(a_j) = 4a_j[1 - a_j].
\]

(3)

The following relationship between the yaw angle of a turbine \( j \) and the angle of the centerline of its wake, \( \xi_j \), was derived in [25]:

\[
\xi_j(x) \approx \left( \frac{\tilde{C}_T(a_j, \gamma_j)}{1 + 2k_d x / D} \right)^2
\]

(4)

with:

\[
\tilde{C}_T(a_j, \gamma_j) = \frac{1}{2} \cos^2(\gamma_j) \sin(\gamma_j) C_T(a_j)
\]

(5)

where \( x \) is the axial distance from the rotor. By integrating this the wake centerline angle over \( x \), the lateral offset of the wake center w.r.t the hub of a turbine \( j \), denoted as \( y_{w,j} \), can be found:

\[
y_{w,j}(x) = \int_0^x \tan(\xi_j(x)) dx.
\]

(6)

This integral can be approximated by integrating the second order Taylor series approximation of \( \xi_j(x) \), yielding:

\[
y_{w,j}(x) \approx \frac{C_T(a_j, \gamma_j) \left[ 15 \left( \frac{2k_d}{D} + 1 \right)^4 + \tilde{C}_T(a_j, \gamma_j)^2 \right]}{\left( \frac{30k_d}{D} \right) \left( \frac{2k_d}{D} + 1 \right)^5} x
\]

(7)

Further, in the experiments described in [12], it was found that a small lateral wake deflection occurs in when the turbine is not yawed (i.e., \( \gamma_j = 0 \)). This deflection can be explained by vertical shear in the boundary layer and wake rotation: in reaction to a rotor rotating clockwise, low speed flow in the lower part of the boundary layer will be rotated up and to the right, and high speed flow in the upper part of the boundary layer will be rotated down and to the left, and as a result the wake deflects to the right. Since in Experiments 1 and 2 the wake behavior was tested for a single mean wind velocity with a limited velocity variance due to turbulence, the exact dependence of the wake deflection on rotor speed could not be derived from the power data obtained. Therefore, this rotation induced wake lateral offset was parametrized through a simple linear function:

\[
y_{w,\text{rotation}}(x) = a_d + b_d x.
\]

(8)

Combining the rotation induced and yaw induced components, the lateral position of the wake center with respect to the hub of a turbine \( j \) is given by:

\[
y_{w,j}(x) = y_{w,\text{rotation}}(x) + y_{w,yaw,j}(x).
\]

(9)

C. Wake expansion

In its original form, the Park / Jensen model [23], [24] assumes a wake that is expanding proportionally to the axial downstream distance to the rotor, and a wind velocity in the wake that is uniform in the lateral direction. In this work we expand the model in order to better match the data from Experiment 1 and 2, by dividing the wake in three areas that also expand proportional to the distance to the rotor, but each with their own expansion factor (see Figure 2a). The diameters of the wake areas behind a turbine \( j \) are given by:

\[
D_{w,j,q}(x) = \max(D_j + 2k_e m_{e,q} x, 0)
\]

(10)

with index \( q = 1, 2, 3 \) numbering the different areas, \( D_j \) being the rotor diameter of turbine \( j \), and with parameters \( m_{e,q}, k_e \) being coefficients defining the expansion of the areas. The different wake areas can be referred to as the ‘near wake’ (\( q = 1 \)), ‘far wake’ (\( q = 2 \)) and ‘mixing zone’ (\( q = 3 \)), in accordance with the terms that are commonly used in literature to describe wake characteristics [18].

D. Wind velocity in a single wake

The Park-Jensen model assumes that the time-averaged velocity deficit in the wake decays quadratically with the expansion of the wake. An extension made in the model presented here in order to better fit the power data from Experiments 1 and 2, is that the wake is divided into three zones, as described in the previous section, and that the
velocity deficit decays quadratically with the distance to the rotor, rather than being directly related to the wake expansion. The velocity profile behind a turbine $j$ is modeled as:

$$U_{w,j}(x,r) = U_j [1 - 2a_j c_j (x,r)]$$

(11)

with $U_j$ the free-stream speed in front of the turbine, and $x$, $r$ the axial and lateral distances to the wake center, and with wake decay coefficient:

$$c_j(x,r) = \begin{cases} c_{j,1} & \text{if } r \leq D_{w,j,1}/2 \\ c_{j,2} & \text{if } D_{w,j,1}/2 < r \leq D_{w,j,2}/2 \\ c_{j,3} & \text{if } D_{w,j,2}/2 < r \leq D_{w,j,3}/2 \\ 0 & \text{if } r > D_{w,j,3}/2 \\ \end{cases}$$

(12)

with the local wake decay coefficient for each zone given by:

$$c_{j,q}(x) = \left[ \frac{D_j}{D_j + 2k_w m_{U,q} (\gamma_j)x} \right]^2.$$  

(13)

Following a similar approach to that in Section III-A, the wake decay rates are adjusted for the rotor yaw angle in order to fit the data from Experiments 1 and 2, by empirically deriving the following relationship between $m_{U,q}$ and $\gamma$:

$$m_{U,q} (\gamma_j) = \frac{M_{U,q}}{\cos(\alpha_U + b_U \gamma_j)}$$

(14)

for $q = 1, 2, 3$ with parameters $M_{U,q}$, $\alpha_U$, $b_U$.

E. Combining wakes to find turbine effective wind velocities

Generally, we cannot assume the free-stream inflow into the wind plant to be known, but by inverting Eq. 1, we can estimate the effective wind speeds at the front turbines from turbine power measurements. Then, the wind speeds at the downstream turbines are estimated by combining the effect of the wakes, weighting the wake areas by their overlap with the rotors using the root-sum-square method of [24]. This results in the following formulations. Let $\mathcal{W} \subset \mathcal{F}$ denote the set of front, upstream turbines that are not influenced by other turbines through wake interaction, and $\mathcal{D} = \{ i \in \mathcal{F} | i \notin \mathcal{W} \}$ the set of turbines that is influenced by other turbines. Further, $i(j)$ denotes the index of a turbine in the set $\mathcal{W}$ that influences a turbine $j \in \mathcal{D}$ through its wake, and $\{x_i,j|j \in \mathcal{F} \}$ are the positions of the turbines along the wind direction. Then, the effective wind speeds at each turbine $j \in \mathcal{F}$ are estimated by:

$$U_j = \left\{ \begin{array}{ll} f_1(j) & \forall j \in \mathcal{W} \\ f_2(j) & \forall j \in \mathcal{D} \end{array} \right.$$ 

(15)

with functions:

$$f_1(j) = \left[ \frac{2P_j}{\rho A_j \mathcal{C}_p(a_j, \gamma_j)} \right]^{1/3}, \quad f_2(j) = U_{i(j)} X(j)$$

(16)

with:

$$X(j) = 1 - 2 \left( \sum_{i \in \mathcal{F}, x_i < x_j} a_{i} \sum_{q=1}^{3} c_{j,q} (x_j - x_i) \min \left( \frac{A_{ol}}{A_j}, 1 \right) \right)^{2}$$

(17)

where $A_{ol}$ denotes the overlapping area of different wake zones, see Figure 2b. These overlapping areas are calculated from the wake center and wake diameter predictions described in Section III-B and III-C, using basic geometry.

F. Fitting the wake parameters

By fitting to the power data from turbine 2 in Experiments 1 and 2, the parameters for wake deflection, expansion and decay were tuned ‘manually’ (cf. Figure 1b, Table I). The results were validated by comparing the resulting wake velocity profiles for a single yawed turbine with the corresponding data generated by SOWFA in the experiments described in [12], see Figure 3 for this comparison. In our example, the parameter $k_w$ is the most significant parameter when adjusting the model to different wind farm power data sets, measured at different atmospheric conditions, e.g. [28].

IV. WIND PLANT YAW OPTIMIZATION USING A GAME-THEORETIC APPROACH

In this paper, we use the game-theoretic (GT) approach of [7] to perform an optimization of the yaw settings in a wind plant based on the simplified model, and validate the results in SOWFA. The GT approach performs the optimization by making random perturbations to the yaw settings and holds the settings as a baseline setting if they yield an improvement of the wind plant total power, so as to iteratively find the global maximum of the wind plant total power. In Algorithm 1, the (simplified) optimization scheme of the GT approach is given as it is implemented in our simulations. The algorithm is somewhat different than the one presented in [7] in the way the exploration distribution and the exploration rate are defined. Instead of choosing $\gamma$ randomly according to a uniform distribution over the full range of allowable values $[\gamma^{\text{min}}, \gamma^{\text{max}}]$, a normal distribution around the baseline setting $\gamma_b$ is used to choose the new setting $\gamma$. Further, an annealing strategy is used: the probability of updating a yaw setting rather than keeping it the same as in the previous iteration, $E$, is reduced as the number of iterations increases, such that a higher density of perturbations results for the earlier iterations than for later iterations. This change is made to improve the convergence speed of the algorithm while avoiding the guarantee on convergence to the global optimum of the model.

V. YAW OPTIMIZATION SIMULATION STUDY

We perform an evaluation of the parametric model and the GT optimization strategy on a wind plant consisting of two rows with 3 NREL 5-MW baseline turbines each, with a 5 rotor diameter spacing in the down-wind direction, and 3 rotor diameters in the cross-wind direction. First, the wind plant is simulated in SOWFA with ‘greedy’ control settings for the yaw, i.e. with the rotors pointed into the direction of the mean wind inflow without any offset. Then, the optimized yaw settings are calculated using the game-theoretic approach described in Section IV on the basis of the predictions given by the model presented in Section III. A second SOWFA simulation is performed with the optimized yaw settings, in which it is validated whether the electrical energy production improvement predicted by the simplified model is indeed achieved. Because of the high mesh and time resolution required ($30 \cdot 10^6$ cells, 0.02s sample time), the computational cost of the CFD simulations is high: 59
Algorithm 1 The pseudocode below shows a Game Theoric approach for wind plant control, performing optimization of the yaw angles for increased electrical energy production. Index \( k \) denotes the iterations of the optimization. The variables \( \gamma_i \) and \( P_i \) are used to store past values of the control variables and the turbine powers. \( \mathcal{U} \) denotes a uniform distribution, \( \mathcal{N} \) denotes a uniform distribution, with \( \sigma \) the standard deviation.

\[
\begin{align*}
1: & \quad \gamma_i \leftarrow 0 \quad \forall i \in \mathcal{F} \\
2: & \quad k \leftarrow 0 \\
3: & \quad \text{update } P_i \forall i \in \mathcal{F} \text{ using (15),(1)} \\
4: & \quad \bar{P} \leftarrow \sum_{i=1}^{N} P_i(t) \\
5: & \quad \bar{\gamma}_i \leftarrow \gamma_i \\
6: & \quad 	ext{loop} \\
7: & \quad \quad k \leftarrow k + 1 \\
8: & \quad \quad \text{update } P_i \forall i \in \mathcal{F} \text{ using (15),(1)} \\
9: & \quad \quad \text{if } \sum_{i=1}^{N} P_i(t) > \bar{P} \text{ then} \\
10: & \quad \quad \quad \bar{\gamma}_i \leftarrow \gamma_i \forall i \in \mathcal{F} \\
11: & \quad \quad \quad \bar{P} \leftarrow \sum_{i=1}^{N} P_i(t) \\
12: & \quad \quad \text{end if} \\
13: & \quad \quad \text{for all } i \in \mathcal{F} \text{ do} \\
14: & \quad \quad \quad \mathcal{B}_1 \leftarrow \text{random value } \sim \mathcal{U}(0,1) \\
15: & \quad \quad \quad E \leftarrow 1/\beta k^5 \\
16: & \quad \quad \quad \text{if } \mathcal{B}_1 < E \text{ then} \\
17: & \quad \quad \quad \quad \mathcal{B}_2 \leftarrow \text{random value } \sim \mathcal{N}(0, \sigma^2) \\
18: & \quad \quad \quad \quad \gamma_i \leftarrow \min \left( \bar{\gamma}_i + \mathcal{B}_2, \gamma_{i}^{\min}, \gamma_{i}^{\max} \right) \\
19: & \quad \quad \quad \text{else} \\
20: & \quad \quad \quad \quad \gamma_i \leftarrow \bar{\gamma}_i \\
21: & \quad \quad \text{end if} \\
22: & \quad \text{end for} \\
23: \end{align*}
\]

Algorithm 1: The pseudocode below shows a Game Theoretic approach for wind plant control, performing optimization of the yaw angles for increased electrical energy production. Index \( k \) denotes the iterations of the optimization. The variables \( \gamma_i \) and \( P_i \) are used to store past values of the control variables and the turbine powers. \( \mathcal{U} \) denotes a uniform distribution, \( \mathcal{N} \) denotes a uniform distribution, with \( \sigma \) the standard deviation.

In Figure 4, we compare the convergence properties of the GT optimization using the uniform distribution of the search actions and an annealing strategy, with the GT approach using a uniform distribution of the search actions over the search range. In the normal distribution GT approach, the search action distribution standard deviation is set as \( \sigma = 35 \), and the annealing parameters as \( \beta = 0.050 \), \( \tau = 1 \). It is shown that a speed-up of the GT approach can be achieved by using the normal distribution with these settings. Each iteration of GT (a simulation of the steady-state of the wind plant with the parametric model) takes MATLAB about 0.6 milliseconds to calculate on a 1.6GHz PC.

Figure 5b shows the results of the SOWFA simulations. It can be seen that after a period in which the wake develops and travels from one turbine to the next, the electrical energy production will decrease significantly in each of the cases due to wake interaction, but that the optimized yaw settings will reduce the wake interaction by redirecting the wake of upstream rotors away from downstream rotors (see also figure 5a), resulting in a 13% wind plant power increase with respect to the ‘greedy’ case. The steady-state power predictions have a decent fit with the turbine powers.
predicted by the extended model. 

For these SOWFA simulations, an inflow with a 6% turbulence intensity and an 8 m/s mean velocity was used, which is the same inflow condition as used in Experiments 1 and 2. Note that a different spacing between the turbines in the wind direction is used in Experiments 1 and 2 to obtain the parametric model, namely 7 rotor diameters, and in that sense we use the model for extrapolation.

The experiments are repeated with a 5° and 10° rotation of the complete wind plant configuration with respect to the wind direction, cf. Figure 5c and 5d, in order to show the effect of changes in wind direction on the wake interaction in the wind plant. For a 10° rotation, there is far less to gain from the yaw optimization, since also in the ‘greedy’ case there is little overlap of the wakes with the downstream turbines. This motivates the use of the parametric model as an internal model of a controller that adjusts the yaw settings to the wind direction.

VI. CONCLUSIONS AND FUTURE WORK

The parametric model presented in this paper has the possibility to be used for optimization of the yaw settings for improved electrical energy production in a small simulated wind plant. The model was found to be able to predict turbine powers for both the ‘greedy’ and ‘optimized’ settings, for changing configurations of the wind plant. Ongoing
work includes implementation of the extended model as an internal model for a wind plant controller that will adjust the yaw reference settings online and adjusts them to the wind direction. As the inflow conditions (e.g., wind speed and turbulence intensity) change, the wake properties are affected, and the model parameters should be updated online. The parametric model presented in this paper has a relatively simple formulation, with a small number of parameters that can be identified using power measurements of the different turbines in the wind plant, which allows for the development such a fully data-driven approach for wind plant optimization control.

REFERENCES