Wind Farm Energy Production Optimization Via Wake Steering

L A M Lima¹,², A K Blatt¹ and M N Machuca¹

¹ Voltalia - Rua do Passeio 78, 14º andar, Centro, 20021-290, Rio de Janeiro, RJ, Brazil
² Electrical Engineering Program, COPPE-UFRJ, Cidade Universitaria, Centro de Tecnologia, bloco H, 21941-972, Rio de Janeiro, RJ, Brazil
E-mail: l.lima@voltalia.com

Abstract. The wake steering technique consists in misaligning turbines with respect to the incoming wind with the goal of displacing their wake region and reducing the wake wind speed deficit, thus increasing the power production of other turbines downwind. In this paper, an algorithm is proposed to estimate the amount of production gain a wind farm could achieve by employing this technique. Details about data treatment are provided and an analytical wake model that can represent the wakes displacement is described. A minimization algorithm to calibrate the wake model parameters based on SCADA data is presented, and the complete math needed to solve this problem is developed. Similarly, to find the optimal yaw angle values that maximize the wind farm production, a maximization problem is described along with the full development of the equations behind it. Insights on an efficient computational implementation to solve these problems are shown, based on matrix representation of the described variables. A case study is presented for a small wind farm owned by Voltalia, consisting of five turbines. Results point to an AEP gain in the order of 1%, a relevant value over the course of a project’s lifetime.

1. Introduction
Wakes are a natural phenomenon related to wind farms. To generate power, a turbine must extract kinetic energy from the incoming wind, generating wake regions of lower wind speed behind it. Depending on site layout and wind direction, turbines on the back rows of a wind farm may operate partially or fully inside these wake regions, which results in reduced power production. A wind farm’s layout is usually designed to minimize wake losses according to historically prevalent wind directions. Nevertheless, some projects have other issues, such as land or environmental restrictions and social acceptance, which may have a big impact on a wind farm’s layout. Therefore, as various wind directions occur during regular operation, some operating points could result in high wake losses. Besides that, the increasing turbine diameter is leading to novel wind farms’ layout design [1], usually with a decreased relative distance (d/D) between turbines, which may lead to relevant wake losses even for the prevalent wind direction.

Recently, many studies have focused on optimizing wind farm generation using control strategies. Most of them are based on sacrificing the production of some turbines to increase that of others, with an overall positive gain. Other advantages are that the turbines’ structural loading may be reduced [2] and that most of these techniques do not require extra equipment in order to be implemented. There are mainly two strategies to achieve that: induction control
[3] and wake steering [4]. The former consists in reducing the turbine’s axial induction factor by pitching the blades or by operating with a sub-optimal rotating speed. The latter is based on yawing turbines to misalign them with respect to the incoming wind, which leads to a deflection in the produced wake region. Figure 1 illustrates the yaw misalignment strategy and its effect on the turbine’s wake trajectory. Both techniques reduce the amount of energy extracted from the wind by the front turbines, reducing the wind speed drop inside the wake region and therefore increasing the production of downwind turbines. By deflecting the wakes away from the back turbines, the wake steering technique further contributes to their production increase.

Figure 1: Sample wake displacement due to turbine yawing, with wind coming directly from the West.

Simulation studies [5, 6], wind-tunnel experimental studies [7] and full-scale experimental measurement campaigns [8] have demonstrated the wake’s displacement and overall wind farm power increase that result from wake steering. Since it could improve a wind farm’s Annual Energy Production (AEP) without the need of extra equipment, it has drawn the attention of several companies and research institutes, as even a 1% gain on a large wind farm would be significant. As far as the authors are aware of, this technique is not yet commercially applied in any wind farm.

2. Objectives
This paper describes an algorithm used to calculate the production gain of a wind farm when applying the wake steering technique. Details and insights on the implementation are provided, describing the steps followed to achieve a computationally fast algorithm. Simulation results for one of Voltalia’s wind farms show a potential production gain around 1%.

3. Methodology
The overall goal of the algorithm is to calculate the amount of AEP gained from the application of wake steering on a given wind farm. For that, some basic information is needed to setup the algorithm, such as turbines location, rotor diameter, power and thrust curves. A series of steps is then performed, as described in the following sections.

The proposed methodology is inspired by Howland et al. [8]. Main contributions of this paper are detailed in the following sections and include: a technique to enable the usage of a different free wind speed value for each wind turbine instead of a single value for the entire wind farm; the detailed and comprehensive description of the derivative equations used in optimization; and a proposal of matrix representation for fast computation of the results.

3.1. Data Treatment
The available SCADA data must be treated to be properly used in the following stages of the algorithm. The dataset consists of data points in a 10-min frequency, containing information about the turbines’ measurement of wind speed, wind direction and power.
First, a simple outlier removal is performed based on the power curve. For each turbine, the measured wind speed is grouped into 0.5m/s bins. Then, the corresponding power measurements for each bin are cleaned up based on the bin’s standard deviation: data points outside a 2.5 std. range are removed.

The measurements are subject to noise and error, especially for the wind speed and direction measurements, and even more if said turbine is in a wake region. The average measured wind direction is expected to be the same for all wind turbines; it is calculated individually and, choosing a specific turbine as reference, the direction offset is removed from the others. For example, if the reference turbine has an average wind direction of 90° and another turbine has an average of 92°, then 2° are subtracted from all direction measurements of that turbine.

Next, the wind farm overall free wind speed and direction are calculated. For each data point, an algorithm calculates which turbines are not inside any wake region (are upwind, or in front) by analyzing the turbine coordinates and the wind direction. These turbines are assumed to be in the free wind region. The algorithm calculates the average wind speed and direction from these turbines, which are assumed to be the overall values of the free wind flow for this data point. This is used mainly for data binning.

The turbines’ power curves are calibrated based on operational data. An algorithm gathers the data points where a given turbine is in front according to the wind direction, and performs wind speed binning. The power values are averaged for each bin, and a correction factor is applied to the manufacturer provided power curve. This way, a more realistic representation of each turbine is possible, accounting for turbine-specific ageing and possible degradation.

Finally, a direction offset removal is performed for the wind farm as a whole. Individual turbine’s wind direction offset was treated before, but if the reference turbine itself has an offset, this could lead to wrong direction values for the entire wind farm. Therefore, an extra step is taken to deal with this issue. The turbines’ power data are compared with the wind direction. Based on their coordinates, the algorithm can identify a possible offset and correct it. For example, if turbine 2 is directly behind turbine 1 for a wind direction of 60° (based on their coordinates), it is expected that the lowest power output of T2 will be for that direction, since it will be in the center of T1’s wake. By analyzing the SCADA data, if the lowest power for T2 happens at 58°, then an offset of 2° is applied to all measured wind directions.

By applying all these corrections, the dataset is ready to be used as input for the wake steering calculations. It is treated with the removal of outliers and offsets, the power curves are corrected and the overall free wind speed and direction are calculated for each data point.

### 3.2. Wake Model

One of the challenges of wake steering is having a fast and accurate analytical wake model that can represent the wake displacement due to yaw. Some wake models are not capable of capturing this behavior (Jensen [9], Frandsen et al. [10]...) but others are (Shapiro et al. [4], Floris [11], Bastankhah and Porté-Agel [7]...). The chosen model is the one presented by Shapiro et al. [4], due to ease of implementation and good accuracy. It represents the wind speed loss profile inside the wake region as a Gaussian distribution, with a wake centerline that may be displaced according to the turbine’s yaw. The main equations for this model can be found on [4, 8] and are reproduced below. They are defined on the turbine’s reference frame, shown in Figure 2.

The streamwise velocity deficit on a given point with coordinates $(x, y)$ caused by the wake of a given turbine $j$ is:

$$
\Delta u_j(x, y) = -\delta u_j(x) \frac{D_j^2}{8\sigma_{0,j}} \exp \left( -\frac{(y - y_{c,j}(x))^2}{2\sigma_{0,j}^2 d_{w,j}(x)} \right),
$$

where $\delta u_j(x)$ is the streamwise velocity deficit, $D_j$ is the turbine diameter, $\sigma_{0,j}$ is the width of
the Gaussian distribution, \( y_{c,j}(x) \) is the wake centerline position at coordinate \( x \) and \( d_{w,j} \) is the normalized wake diameter. If the given point is located in front of the turbine (i.e. negative value of \( x \)), the wind speed drop is zero because the point is not actually inside the wake region.

The streamwise velocity deficit is calculated as (2)-(4), where \( \delta u_0 \) is the initial streamwise velocity deficit, \( \text{erf} \) is the Gaussian Error Function, \( a_j \) is the axial induction factor, \( u_{\infty,j} \) is the turbine’s free windspeed, \( C_{T,j} \) is the turbine’s thrust coefficient (as provided by the manufacturer) and \( \gamma_j \) is its yaw angle. The wake centerline is calculated as (5)-(7), where \( \delta v(x) \) is the transverse velocity and \( \delta v_0 \) is the initial transverse velocity.

\[
\delta u_j(x) = \frac{\delta u_{0,j}}{d_{w,j}(x)} \frac{1}{2} \left[ 1 + \text{erf} \left( \frac{x}{\sqrt{2}} \right) \right],
\]

(2)

\[
\delta u_{0,j} = 2a_j u_{\infty,j},
\]

(3)

\[
d_{w,j}(x) = \frac{1}{2} \left[ 1 - \frac{1}{1 - C_{T,j} \cos^2(\gamma_j)} \right],
\]

(4)

\[
y_{c,j}(x) = \int_{x'=0}^{x} \frac{-\delta v(x')}{u_{\infty,j}} dx',
\]

(5)

\[
\delta v(x) = \frac{\delta v_{0,j}}{d_{w,j}(x)} \frac{1}{2} \left[ 1 + \text{erf} \left( \frac{x}{\sqrt{2}} \right) \right],
\]

(6)

\[
\delta v_0 = \frac{1}{4} C_{T,j} u_{\infty,j} \cos^2(\gamma_j) \sin(\gamma_j).
\]

(7)

The wake centerline integral is calculated discretely, and the number of steps is defined according to available computational power.

The normalized wake diameter is defined as a function of the wake spread coefficient \( k_{w,j} \):

\[
d_{w,j}(x) = 1 + k_{w,j} \log (1 + \exp[2(x/D_j - 1)])).
\]

(8)

Finally, the power coefficient and generated power for a given turbine are calculated as:

\[
C_{p,j} = 4\eta_j a_{p,j} (1 - a_{p,j}^2) \cos^p(\gamma_j),
\]

(9)

\[
a_{p,j} = 1/2 \left( 1 - \sqrt{1 - C_{T,j}} \right),
\]

(10)

\[
P_j = 0.5\rho A_j C_{p,j} u_j^3,
\]

(11)

where \( \eta_j \) is the power curve calibration parameter (as mentioned in the previous section), \( \rho \) is the air density, \( A_j \) is the turbine’s rotor swept area and \( u_j \) is the turbine speed. The value of the exponent \( p \) is chosen as \( p = 2 \) [8].

An advantage of the proposed algorithm is that it is common for methodologies to consider a uniform wind field [8], with the same free wind speed for all turbines. The proposed algorithm analyzes which turbines are inside any wake region or not, and computes their individual free wind speed accordingly to be used in the calculations. Figure 3 shows an example of this logic. The algorithm understands that turbine 1 is not inside any wake region, therefore its wind speed measurement represents the free wind speed it is experiencing. As for turbine 2 that is not the case: its wind speed measurement includes a deficit caused by the wake. Since T2 is inside T1’s wake region, the algorithm considers that T2’s free wind speed would be the same as T1’s. The wind direction is assumed to be the same for all turbines, as described in section 3.1.

To calculate a turbine’s resulting wind speed, a discrete integration of the wind speed over its rotor area is performed as in (12). The rotor is discretized in individual ‘rotor pieces’;

\[
\int_{A} u_{\text{rotor}} \ dx.
\]
resulting wind speed is calculated for each rotor piece and then averaged over the rotor area. In case a given rotor piece is subject to the wake of more than one upwind turbine, the individual wake velocity deficits are linearly superposed, as in (13).

\[ u_i = u_{\infty,i} - \frac{1}{A_i} \sum_{p=1}^{np} \Delta u_{ip}(x_{ip}, y_{ip})h_{ip} \frac{D_i}{np} \]  
\[ \Delta u_{ip}(x_{ip}, y_{ip}) = \sum_{j=1}^{m} \Delta u_{ipj}(x_{ipj}, y_{ipj}). \]  

Here, \( np \) is the number of rotor pieces, \( h_{ip} \) is their height, \( \Delta u_{ipj} \) is the velocity deficit caused by the wake of turbine \( j \) on a particular rotor piece (subscript \( p \)) of turbine \( i \), \( x_{ipj} \) and \( y_{ipj} \) are the coordinates of piece \( p \) of turbine \( i \) on the reference frame of turbine \( j \) and \( m \) is the number of turbines in the wind farm. The rotor pieces are defined along a linear path across the middle of the rotor area, and their amount \( (np) \) is chosen according to available computational power.

Figure 4 represents the rotor pieces of a given turbine, using an arbitrary turbine 3 as example for the naming convention. Each tick in the rotor represents an individual rotor piece and the height of piece 2 is shown as an example. The 3\(^{rd} \) subscript, \( j \), is omitted, since it would only mean the index of another turbine that would cause wind deficit in these rotor pieces.

\[ u_{\infty} \]
\[ p_{3,np} \]
\[ T_3 \]
\[ p_{3,2} \]
\[ p_{3,1} \]
\[ h_{3,2} \]

Figure 4: Illustration example of the rotor pieces and their height.

### 3.3. Wake Parameters Calibration

The described wake model has two parameters: \( k_{w,i} \) and \( \sigma_{0,i} \). These change according to surface roughness, turbulence intensity, atmospheric stability and other conditions. Since SCADA data is available, these parameters can be calibrated to properly reflect the site-specific conditions. They are calibrated separately for each bin of wind speed and wind direction. Since the terrain around the wind farm is not uniform, the resulting wake parameters can be different depending on the incoming wind direction. An analytical approach is implemented based on gradient descent technique to quickly solve the minimization problem. The objective function \( F \) is defined as the mean absolute error:

\[ F = \text{mean absolute error} \]
\[ F = \frac{1}{m} \sum_{i=1}^{m} |\dot{P}_i - P_i| = f (\sigma_{0,1}, \sigma_{0,2}, ..., k_{w,1}, k_{w,2}, ...) , \]  

where \( P_i \) is the calculated turbine power and \( \dot{P}_i \) is the SCADA power value.

To minimize \( F \), its partial derivatives with respect to each wake parameter must be calculated analytically, according to the wake model presented in the previous section. Considering a generic wake parameter \( \beta \) that could represent both \( \sigma_0 \) and \( k_w \):

\[ \frac{\partial F}{\partial \beta} = -\sum_{i=1}^{m} \text{sign}(\dot{P}_i - P_i) \frac{\partial P_i}{\partial \beta} , \]  

\[ \frac{\partial P_i}{\partial \beta} = \frac{\partial P_i}{\partial u_i} \cdot \frac{\partial u_i}{\partial \beta} = 3 \frac{\rho A_i C_p i u_i}{\partial \beta} , \]  

\[ \frac{\partial u_i}{\partial \beta} = -\frac{1}{A} \sum_{p=1}^{np} \left( \sum_{j=1}^{m} \frac{\partial \Delta u_{ipj}}{\partial \beta} \right) h_{ipj} D_i , \]  

From this step on, the derivatives are different according to the target wake parameter. For \( \sigma_0 \), the derivatives are:

\[ \frac{\partial \Delta u_{ipj}}{\partial \sigma_{0,j}} = \begin{cases} 0, & j \neq i, \\ -\frac{2}{\sigma_{0,j}} \frac{\delta u_j(x_{ip}) D_j^2}{8\sigma_{0,j}^2} \exp \left( -\frac{(y_{ipj} - y_{c,j}(x_{ip}))^2}{2\sigma_{0,j}^2 d_{w,j}(x_{ip})} \right) \left[ 1 - \frac{(y_{ipj} - y_{c,j}(x_{ip}))^2}{2\sigma_{0,j}^2 d_{w,j}(x_{ip})} \right] , & \text{if } j \neq i, \end{cases} \]  

For \( k_w \), the derivatives are:

\[ \frac{\partial \Delta u_{ipj}}{\partial d_{w,j}} = \begin{cases} 0, & j \neq i, \\ \delta u_j(x_{ip}) D_j^2 \left[ -\frac{(y_{ipj} - y_{c,j}(x_{ip}))^2}{2\sigma_{0,j}^2 d_{w,j}(x_{ip})} \right] \exp \left( -\frac{(y_{ipj} - y_{c,j}(x_{ip}))^2}{2\sigma_{0,j}^2 d_{w,j}(x_{ip})} \right) , & \text{if } j \neq i; \end{cases} \]  

\[ \frac{\partial \Delta u_{ipj}}{\partial y_{c,j}} = \frac{(y_{ipj} - y_{c,j}(x_{ip})) D_j^2}{8\sigma_{0,j}^2} \exp \left( -\frac{(y_{ipj} - y_{c,j}(x_{ip}))^2}{2\sigma_{0,j}^2 d_{w,j}(x_{ip})} \right) , \]  

\[ \frac{\partial \Delta u_{ipj}}{\partial u_j} = \frac{D_j^2}{8\sigma_{0,j}} \exp \left( -\frac{(y_{ipj} - y_{c,j}(x_{ip}))^2}{2\sigma_{0,j}^2 d_{w,j}(x_{ip})} \right) , \]  

\[ \frac{\delta u_{0,j}}{\partial d_{w,j}} = \frac{\delta u_{0,j}}{d_{w,j}(x_{ip})} \left[ 1 + \text{erf} \left( \frac{x_{ipj}}{\sqrt{2} D_j/2} \right) \right] , \]  

\[ \frac{\delta u_{c,j}}{\partial k_{w,n}} = -\frac{1}{u_{\infty,j}} \int_{x_{ipj}}^{x_{ipj}/D_j} \frac{\delta v_{ij}}{d_{w,j}(x_{ip})} \frac{\partial d_{w,j}}{\partial k_{w,j}} dx' , \]  

\[ \frac{\delta u_{c,j}}{\partial k_{w,n}} = -\frac{\delta u_{c,j}}{d_{w,j}(x_{ip})} \left[ 1 + \text{erf} \left( \frac{x_{ipj}}{\sqrt{2} D_j/2} \right) \right] , \]  

\[ \frac{\delta d_{w,j}}{\partial k_{w,n}} = \log \left( 1 + \exp[2(x_{ipj}/D_j - 1)] \right) . \]
For (18) and (19), if $j = i$, it would mean to calculate the effect of a turbine’s wake on itself, which does not make sense; therefore, the derivatives are set to zero.

The algorithm runs iteratively to calibrate the wake parameters. On each iteration, the derivatives are calculated and the parameters are updated according to gradient descent rule

$$
\beta(t + 1) = \beta(t) - \alpha \frac{\partial F}{\partial \beta(t)},
$$

until they converge. An adaptive approach is used, where $\alpha$ is increased if two consecutive updates are on the same direction (same sign) or decreased otherwise (it would mean the algorithm is converging around a result).

To speedup the calculations, all partial derivatives are implemented in a way to enable matrix operations, avoiding for-loops to calculate the derivatives for different turbines. The variables’ matrix structure is as follows, using the wind speed deficit $\Delta u_{ipj}$ as an example:

- Lines ($1^{st}$ dim.) represent the turbine to which the rotor pieces belong ($i$);
- Columns ($2^{nd}$ dim.) represent the rotor pieces themselves ($p$); and
- Height ($3^{rd}$ dim.) represents the turbine ($j$) whose wake causes a velocity deficit in the rotor piece $p$.

For example, the wind speed deficit caused by turbine 5’s wake on rotor piece number 42 of turbine 3 would be located on line 3, column 42, height 5 of this variable. If a given variable is not calculated separately for each rotor piece (e.g. turbine speed $u_i$, diameter $D_i$), it would have only 1 column. To calculate the wake centerline of turbine $j$ at each rotor piece position of turbine $i$ ($y_{c,j}(x_{ipj})$), integration is required from turbine $j$’s $x$-coordinate to the rotor piece $x$-coordinate as described in (5). Therefore, a 4-dimensional variable is needed, and the length of its $4^{th}$ dimension is equal to the number of integration steps used to calculate $y_{c,j}(x_{ipj})$. For example, the $x$-coordinate of the 70th integration step needed to calculate the wake centerline of turbine 7 on the position of turbine 2’s rotor piece number 26 would be located on line 2, column 26, height 7 and $4^{th}$-dimension index 70 of the variable that represents the $x$-coordinates. This adaptation makes the calculations very fast, which is crucial since they must be run repeatedly until the values converge, and also for numerous combinations of wind speed and direction bins.

With the wake parameters calibrated, it is assumed that the wake model is able to produce results that represent well the analyzed wind farm.

### 3.4. Yaw Optimizer

The next step is to calculate the optimal turbine yaw angles that increase the total generated power. It is unfeasible to brute-force test all possible yaw combinations, so again an optimization problem must be developed using the wake model equations and partial derivatives. The objective function to be maximized is the sum of all turbines’ power calculated according to the described wake model:

$$
F = \sum_{i=1}^{m} P_i.
$$

The partial derivatives with respect to the individual turbines’ yaw angles are:

$$
\frac{\partial F}{\partial \gamma_j} = \sum_{i=1}^{m} \frac{\partial P_i}{\partial \gamma_j},
$$

$$
\frac{\partial P_i}{\partial \gamma_j} = 0.5 \rho A_i u_i^3 \frac{\partial C_{P,i}}{\partial \gamma_j} + 1.5 \rho A_i C_{P,i} u_i^2 \frac{\partial u_i}{\partial \gamma_j},
$$

$$
\frac{\partial C_{P,i}}{\partial \gamma_j} = \begin{cases} 
-8 \eta_{a,p,i}(1 - a_{p,i})^2 \cos(\gamma_i) \sin(\gamma_i), & \text{if } j = i; \\
0, & \text{otherwise},
\end{cases}
$$

$$
\frac{\partial u_i}{\partial \gamma_j} = -\frac{1}{A} \sum_{p=1}^{n_p} \left( \sum_{j=1}^{m} \frac{\partial \Delta u_{ipj}}{\partial \gamma_j} \right) h_{ip} D_i \frac{D_i}{n_p},
$$
\[
\frac{\partial \Delta u_{ipj}}{\partial \gamma_j} = \frac{\partial \Delta u_{ipj}}{\partial d_{w,j}} \frac{\partial d_{w,j}}{\partial \gamma_j} + \frac{\partial \Delta u_{ipj}}{\partial y_{c,j}} \frac{\partial y_{c,j}}{\partial \gamma_j} + \frac{\partial \Delta u_{ipj}}{\partial d_{u,j}} \frac{\partial d_{u,j}}{\partial \gamma_j} + \frac{\partial \Delta u_{ipj}}{\partial y_{p,j}} \frac{\partial y_{p,j}}{\partial \gamma_j},
\]
(32)

\[
\frac{\partial \delta u_j}{\partial \gamma_j} = \frac{\partial \delta u_j}{\partial d_{w,j}} \frac{\partial d_{w,j}}{\partial \gamma_j} + \frac{\partial \delta u_j}{\partial y_{c,j}} + \frac{2 \delta u_j}{\partial d_{w,j}} \frac{\partial d_{w,j}}{\partial \gamma_j},
\]
(33)

\[
\frac{\partial \delta u_j}{\partial y_{c,j}} = \frac{1}{d_{w,j}(x_{ipj})^2} \left[ 1 + \text{erf} \left( \frac{x_{ipj}}{\sqrt{2D_j}/2} \right) \right],
\]
(34)

\[
\frac{\partial \delta u_j}{\partial x_{ipj}} = \frac{\delta u_j(x_{ipj})}{d_{w,j}(x_{ipj})^2} \frac{1}{\exp \left[ \left( \frac{x_{ipj}}{\sqrt{2D_j}/2} \right)^2 \right]},
\]
(35)

\[
\frac{\partial \delta v_j}{\partial \gamma_j} = \frac{\delta v_j}{\delta v_{0,j}} \frac{1}{\delta v_{0,j}} \left[ 1 + \text{erf} \left( \frac{x_{ipj}}{\sqrt{2D_j}/2} \right) \right],
\]
(36)

\[
\frac{\partial \delta v_j}{\partial x_{ipj}} = \left\{ \begin{array}{ll}
0.25C_{T,j} u_{\infty,j} (\cos^3 \gamma_j - 2 \cos \gamma_j \sin^2 \gamma_j), & \text{if } j \neq i; \\
0, & \text{otherwise.}
\end{array} \right.
\]
(37)

Equations (20)-(22) are also used, but were omitted in this section to avoid repetition. For
the derivative of the rotor pieces position according to the yaw angle, the distance from
each rotor piece to the rotation center (nacelle position) is needed. Then, using simple trigonometric
rules, the derivatives are calculated.

\[
r_{ipj} = \sqrt{(x_{ipj} - x_{ij})^2 + (y_{ipj} - y_{ij})^2},
\]
(41)

\[
\frac{\partial x_{ipj}}{\partial \gamma_j} = \left\{ \begin{array}{ll}
r_{ipj} \cos \gamma_j, & \text{if } j = i; \\
0, & \text{otherwise;}
\end{array} \right.
\]
(42)

\[
\frac{\partial y_{ipj}}{\partial \gamma_j} = \left\{ \begin{array}{ll}
-r_{ipj} \sin \gamma_j, & \text{if } j = i; \\
0, & \text{otherwise.}
\end{array} \right.
\]
(43)

Some derivatives have cases that bear a specific meaning. For example, the $C_T$ and rotor
pieces’ position derivatives of (30), (42) and (43) only make sense with respect to yaw variations
of the turbine itself, since the turbine $C_T$ or its rotor piece coordinates do not change if another
turbine changes its yaw. Equation (40) is related to variations of the wake centerline of turbine
$j$, which is affecting turbine $i$. Again, since a wind turbine does not affect itself via wakes, the
case where $j = i$ is set to zero.

Similar to the wake parameters calibration, the optimal yaw calculation is done iteratively
for different wind speed and direction bins, using the gradient descent algorithm. On each
iteration, the derivatives are calculated and the yaw values are updated, repeating until the
algorithm converges and the optimal yaw angles are found.

### 3.5. Power Calculation

At this point, the algorithm includes a wake model with calibrated parameters that can calculate
the power production of the wind farm. It also knows the optimal yaw values to be applied if
wake steering is to be considered. The algorithm calculates the wind farm production with
normal operation (yaw=0°) and with wake steering operation (optimal yaw values). Based on
SCADA data, the local wind rose is found, with information about the site-specific wind speed
distribution. With this information, the estimated AEP can be calculated for each operation scenario, and the wake steering gain is found.

4. Case Study
The complete algorithm is run for a small wind farm owned by Voltalia, comprising 5 turbines organized in 2 rows, to assess the potential gain that could come from applying wake steering to this site. Figures 5a and 5b show its layout (where each turbine is represented by a circle) and the local wind rose for the considered time period, calculated from SCADA data. The available data amounts to roughly 6 months of operation.

Figure 5c shows the optimal yaw values, binned by wind direction. It shows the average and the maximum values of the turbines optimal yaws for each bin. It can be seen that the maximum individual yaw values are not high, generally below 10°, which is positive from a mechanical perspective. High values of yaw could lead to mechanical stress on the yaw motors and bearings, since the turbines are usually designed to operate without a yaw misalignment to the wind. It is important to note that not all turbines are yawed on a given situation; there are also direction bins where no turbine is inside the wake region of any other, therefore no yawing is required. On any given direction bin, the back turbines are not yawed, since they would only lose production without benefiting any other turbine.

The possible gain from applying the wake steering technique is shown in Figure 5d for different bins. It is important to note that not all possible bins have corresponding SCADA data, such as the bins for 150° and 160°. Therefore, no yaw or possible gain are calculated for these cases. Considering the available bins, the calculated AEP gain for the wind farm is in the order of 1%. The bins with highest gains (50°, 240°) correspond roughly to the direction of the turbine rows in the wind farm; this makes sense since, in these cases, the turbines are aligned with respect to the incoming wind, which results in higher wake losses. A relevant gain is also observed for the bins around 330°. In this situation, the wind is coming perpendicularly to the turbine rows, therefore turbines in different rows end up aligned with respect to the incoming wind, resulting in high wake losses that are mitigated by the wake steering technique.

5. Conclusions
An algorithm to estimate the possible gain of implementing the wake steering technique on a given wind farm was presented. No extra equipment is needed, and only existing SCADA data is used in the calculations. The algorithm includes the description of a wake model that is able to represent the wake displacement due to turbine yaw; a data treatment process that enables the usage of a different free wind speed value for each turbine as opposed to a single value for the entire wind farm; wake parameters calibration via minimization, to better represent site-specific conditions; an optimizer to calculate the yaw angles that result in maximum generated wind farm power; and a matrix representation of the variables used to perform fast calculations.

A preliminary case study for one of Voltalia’s small wind farms, comprising 5 turbines, was performed. It showed a potential AEP gain in the order of 1%, which is significant over the project’s lifetime. Discussions are ongoing with the turbine’s manufacturer to start an experimental measurement campaign in this wind farm, in order to verify the numerical results and assess the technique’s feasibility. Higher gains are expected for larger wind farms, since the wake interaction among various turbines would be in place, and yawing a given turbine could potentially increase the production of several others.

The model can be further improved by including the wind resource map of the area as an input. The wake parameters calibration could also be improved, with the implementation of sensitivity analyses for the results. Besides that, mechanical and control system studies must also be performed to enable the commercial application of this technique.
Figure 5: Wind farm (a) layout and (b) wind rose for the analyzed wind farm. Results include the (c) turbine’s optimal yaw and (d) the overall wind farm production increase by bin.

References
[1] Stevens R, Hobbs B, Ramos A and Meneveau C 2017 Combining economic and fluid dynamic models to determine the optimal spacing in very large wind farms Wind Energy 20 465–77
[2] Bossanyi E 2018 Combining induction control and wake steering for wind farm energy and fatigue loads optimisation IOP Conf. Series: Journal of Physics 1037 (3)
[3] van der Hoek D, Kanev S, Allin J, Bieniek D and Mittelmeier N 2019 Effects of axial induction control on wind farm energy production - a field test Renewable Energy 140 994-1003
[4] Shapiro C, Gayme D and Meneveau C 2018 Modelling yawed wind turbine wakes: a lifting line approach Journal of Fluid Mechanics 841 R1
[5] Doekmeijer B M, Fleming P A and van Wingerden J W 2018 A tutorial on the synthesis and validation of a closed-loop wind farm controller using a steady-state surrogate model arXiv:1810.12066v2
[6] Raach S, Schlipf D, Borisade F and Cheng P 2016 Wake redirecting using feedback control to improve the power output of wind farms Proc. 2016 American Control Conf.
[7] Bastankhah M and Porte-Agel F 2016 Experimental and theoretical study of wind turbine wakes in yawed conditions J. Fluid Mech. 806 506-41
[8] Howland M, Lele S and Dabiri J 2019 Wind farm power optimization through wake steering Proc. National Academy of Sciences 116 14495-500
[9] Jensen N 1983 A note on wind generator interaction Riso National Laboratory
[10] Frandsen S, Barthelmie R, Pryor S, Rathmann O, Larsen S and Hojstrup J 2006 Analytical Modelling of Wind Speed Deficit in Large Offshore Wind Farms Wind Energy 9 39-53
[11] Boersma S, Doekemeijer B M, Siniscalchi-Minna S and van Wingerden J W 2018 A constrained wind farm controller providing secondary frequency regulation: an LES study Renewable Energy 134 639-52